Stability-Constrained Inverse Modeling for Parameter Identification in Dynamical Systems via Neural Surrogates

Daniel Menacho Ordoñez
dmenordo@umich.edu
University of Michigan
Ann Arbor, USA

Abstract

Dynamical systems are governed by complex differential equations that require accurate parameter identification to achieve desired stable behavior. Traditional forward modeling approaches predict system outputs from given parameters but lack mechanisms to ensure that the identified parameters lead to stable long-term dynamics. In this work, we propose a stability-constrained inverse modeling framework that integrates neural surrogate models with Lyapunov-based stability guarantees to delimit admissible parameter spaces, ensuring system stability. Our proposed approach relies on Neural Ordinary Differential Equations as flexible surrogate dynamic models and incorporates a physics-informed learnable Lyapunov function to characterize regions of attraction. This design enables the formulation of the inverse problem with explicit stability constraints, ensuring that all identified parameters lead to stable equilibrium convergence and overcoming the limitations of traditional analytical and forward approaches. The framework is evaluated on two representative dynamical systems: a pendulum–spring mechanical system and the training dynamics of neural networks. Our method demonstrates a reliable proof of concept through a toy experiment and a generalizable application, utilizing learning curves, both of which offer theoretical stability guarantees for safe parameter identification. The code is available at https://github.com/dmenacho/lyapunov_inv, a webpage [1] summarizing the project, and the dataset can be accessed through Drive [2].

Keywords

Inverse Problem, Lyapunov Stability, Surrogate Model, Region of Attraction, Parameter Identification

1 Introduction

Stability analysis constitutes a cornerstone of research in control systems [9, 17]. Ensuring stability guarantees that the system maintains consistent qualitative behavior in its trajectories relative to an equilibrium state [11]. A fundamental tool to guarantee stability is the Lyapunov function, which estimates the Region of Attraction (RoA), a quantitative measure of stability [17], and serves to evaluate whether the system can recover and converge to a stable equilibrium after an abrupt perturbation [9]. Conversely, the absence of convergence may result in unsafe and unstable behavior, which is critical in applications such as autonomous vehicles, robotics, and medical decision-making [4].

The traditional method for stability analysis relies on the manual design of a Lyapunov function. To automate this process, computational approaches based on sums-of-squares (SOS) optimization have gained widespread attention [7]. However, integrating this concept into learned models remains a major challenge due to the abstract nature of stability and the difficulty of expressing Lyapunov inequalities [9] as nonlinear constraints within neural network architectures. Recent advances in Physics-Informed Neural Networks (PINNs) [3, 8, 10] and Neural Ordinary Differential Equations (NeuralODEs) [14, 18, 19] have begun to incorporate stability representations through Lyapunov candidate functions, allowing modeling of well-defined RoA and achieving robust, stable behavior in empirical experiments.

The main drawback of these forward approaches lies in their focus on achieving stability for fixed inputs, without analyzing the true range of parameter variations that lead to stable behavior, inspired by [13]. Within this context, surrogate models provide a promising framework for studying how variations in initial conditions or parameters affect the resulting simulated trajectories [5]. However, these models often produce trajectories that become unstable due to their limited interpretability and lack of physical constraints consistent with real-world systems.

To address these limitations, we propose a stability-constrained inverse modeling framework that integrates neural surrogate models with Lyapunov-based stability constraints. Unlike existing approaches that either certify stability for given parameters (forward problem) or learn surrogates without stability guarantees, our method formulates parameter identification as an optimization problem where candidate solutions must simultaneously match desired outputs and satisfy learned Lyapunov conditions. Our main contributions are summarized as follows:

- We propose an inverse modeling framework constrained by a learnable Lyapunov function to identify parameter sets that induce stability.
- We introduce a generalizable framework adaptable to a wide range of neural surrogate models.
- We validate the reliability of the proposed approach through a proof-of-concept toy-physic experiment, demonstrating its scalability to decision-making tasks for selecting stable parameters during model training.

2 Related Works

2.1 Learning Stability from Dynamic Systems with Neural Models

Proving stability in high-complexity dynamical models with strong nonlinearities and higher-order differential equations remains a

[1] https://dmenacho.github.io/
[2] https://drive.google.com/drive/folders/1dGcmLMgrN1DqZYFVxtohhVEB23_yqTRh?usp=sharing

major challenge. Recent work has increasingly employed PINNs to automate stability analysis. Chu et al. [3] proposed a structure-preserving PINN that incorporates stability constraints directly into the loss function through additional term enforcing Lyapunov function decay or energy conservation, demonstrating robustness gains on nonlinear systems and under adversarial perturbations. Convergence guarantees were established by Drgoňa et al. [4], who parameterized the candidate Lyapunov function using an Input Convex Neural Network (ICNN) followed by a positive-definite layer. This configuration ensures the absence of local minima in the Lyapunov candidate and guarantees convergence to a stable state, with successful applications to robotic controller stabilization. However, these advances often require manual specification of Lyapunov candidates, limiting scalability to new domains. Liu et al. [9, 10] proposed LyZNet, a PINN-Lyapunov approach that treats the Lyapunov function as a learnable object within a PINN framework by solving the Lyapunov and Zubov PDEs, eliminating its inequality formulation. This framework estimates the RoA to ensure system stability [8], and its robustness was later improved through a rollout-based training strategy [17]. Nevertheless, these pipelines remain computationally intensive but provide an interpretable region that guarantees stability, which aligns with one of the main objectives of our work.

The RoA has been explored using neural networks (NN). Zhou et al. [19] proposed a dual framework capable of learning unknown dynamics and identifying a valid Lyapunov function, supported by SMT-based verification to ensure stability guarantees. Research on Neural ODEs has also advanced the integration of Lyapunov-based stability properties within neural surrogate models. Stable Neural ODEs [18] introduced a framework capable of supporting multiple attractors. In this approach, the system dynamics are corrected by a Lyapunov term whenever the Lyapunov inequality condition is violated, thereby ensuring stability. Nevertheless, the method remains sensitive to Lyapunov function initialization, where poor initialization may lead to local minima. LyaNet [14], evaluated on noisy vision datasets, proposed a framework that learns stability through a Lyapunov loss formulation, which quantifies the degree of violation of the system dynamics relative to the Lyapunov condition for exponential stability. This introduces structural priors into the learned dynamics, a property typically missing in standard Neural ODEs. Extended for forecasting in chaotic PDEs, Stabilized Neural ODEs [18] decompose the ODE representation into two components. A nonlinear neural network that models the intrinsic system dynamics, and a linear operator that acts as a stabilizer. This formulation improves stability and long-term forecasting accuracy compared to standard Neural ODEs, although it lacks formal Lyapunov guarantees.

These studies show that forward stability modeling has been thoroughly investigated; however, they do not explicitly address how to identify the parameter sets and initial conditions that ensure stability. Our proposed framework bridges this gap by reformulating stability analysis as an inverse optimization problem that integrates Lyapunov-region constraints with reliable surrogate model predictions within a unified learning objective.

2.2 Surrogate Models for Inverse Parameter Identification

The counterfactual component of the surrogate model can be interpreted as an adaptive mechanism that enables a control system to respond to adversarial disturbances. This concept is realized in the Counterfactual Generator framework [16], a generative model designed to enhance a robot controller's ability to recover from counterfactual perturbations, approximating the system's stability boundaries. Furthermore, the interpretability of such models can be strengthened by integrating expert-driven dynamics with Neural ODEs and diffusion-based processes, as demonstrated in the ODE-Diff framework [12].

Parameter estimation has primarily focused on model discovery and uncertainty quantification. The SINDy algorithm [1] leverages sparse regression to select the most relevant terms, successfully recovering governing equations even in chaotic or high-dimensional systems. Fuentes et al. [5] employed sparse Bayesian learning for joint model selection and parameter estimation, providing probabilistic uncertainty estimates for both parameters and predictions. CausalKinetiX [13] improves out-of-sample generalization by identifying invariance as a signature of causal structure, ranking models according to predictive accuracy and invariance across heterogeneous experimental regimes. Despite their strengths, current surrogate-based parameter identification approaches focus mainly on data fitting, model sparsity, or experimental invariance, and rarely impose formal dynamical stability.

3 Problem Definition

Consider a general dynamical system governed by a function f that evolves in continuous time t. The function defines how the state vector $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, composed of n variables, interacts with a parameter vector $\theta \in \Theta \subseteq \mathbb{R}^p$, composed of p parameters, to describe the system's temporal evolution. *Assumption 1: the governing function $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ is continuously differentiable and describe the corresponding ordinary differential equation (ODE) with the initial condition $x(0) = x_0$.*

$$\dot{x} = f(x(t), \theta), \quad x(0) = x_0 \tag{1}$$

The system is assumed to be observable at each time step through the use of a reliable numerical solver. *Assumption 2: The proposed approach employs a surrogate model $\mathcal{M}_{fwd}$ that approximates the solution of the complex dynamical system, providing an output $\hat{y} \in \mathcal{Y}$ at a finite horizon T.*

$$x(T, x_0, \theta) = \hat{y} \approx \mathcal{M}_{fwd}(x_0, \theta) \tag{2}$$

Assumption 3: For every parameter $\theta \in \Theta$, the dynamical system admits at least one equilibrium point $x^*(\theta)$. For simplicity, we denote this equilibrium point as $x^* \in \mathbb{R}^n$, where the state remains constant for all $t \geq 0$ and satisfies $\dot{x} = 0$.

$$f(x^*, \theta) = 0 \tag{3}$$

The stability of the system is demonstrated using a Lyapunov function. *Assumption 4: There exists a continuously differentiable, positive-definite Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$ that certifies the asymptotic stability of the system.* The proposed Lyapunov function must satisfy $V(x, \theta) > 0, : \forall x \neq x^*$, acting as an energy-like function that is positive outside the equilibrium and vanishes at the

equilibrium point $V(x^*, \theta) = 0$. Furthermore, to ensure convergence to the equilibrium, the function must decrease along the system's trajectories.

$$\dot{V}(x, \theta) = \nabla_x V(x, \theta) \cdot f(x, \theta) < 0 \tag{4}$$

The three conditions ensure the asymptotic stability of the system, implying the existence of a RoA for a set of parameters θ, where any initial condition x_0 leads the system trajectory to converge toward x^*.

$$\mathcal{R}(\theta) = \{x_0 : \lim_{t\to\infty} x(t, x_0, \theta) = x^*(\theta)\} \tag{5}$$

However, a Lyapunov function does not explicitly define the Region of Attraction, and a conservative Lyapunov formulation may underestimate the actual region, omitting states where the system remains stable. Following the LyZNet approach [9], we propose a subset of the Region of Attraction, denoted as $\mathcal{S}_{RoA}$:

$$\mathcal{R}(\theta) \supseteq \mathcal{S}_{RoA} = \{x : V(x, \theta) \leq c, \dot{V}(x, \theta) < 0\} \tag{6}$$

Since $\mathcal{S}_{RoA}$ lies within the true RoA, it is asymptotically stable. The objective is to maximize $\mathcal{S}_{RoA}$ by determining the largest non-conservative value of c that preserves stability.

With the observed output y and the delimited Region of Attraction (RoA), we can formulate the inverse stability-constrained problem. Therefore, there exist a perturbed initial state $x_0' = x_0 + \Delta x_0$ and perturbed parameters $\theta' = \theta + \Delta\theta$. These perturbed variables (x_0', θ') are mapped through the inverse problem, which requires that they remain within a RoA guaranteeing asymptotic stability:

$$\mathcal{R}(\theta') = \{x_0' : V(x_0', \theta') \leq c, \dot{V}(x_0', \theta') < 0\} \tag{7}$$

Since the output y is fixed, the surrogate model evaluated at these perturbed variables must reproduce the same output, $\mathcal{M}_{fwd}(x_0', \theta') \approx y$. This formulation leads to the following optimization problem:

$$\begin{aligned} &\min_{\Delta x_0, \Delta\theta} \|\mathcal{M}_{fwd}(x_0 + \Delta x_0, \theta + \Delta\theta) - y\|^2, \\ &\textit{constrained by } x_0 + \Delta x_0 \in \mathcal{R}(\theta + \Delta\theta) \end{aligned} \tag{8}$$

Finally, to ensure exponential stability and convergence without overshoot, we adopt the stability constraint proposed in Eq. (9) from [18] and [3], where α is a positive hyperparameter:

$$\dot{V}(x_0', \theta') \leq -\alpha V(x_0', \theta'), \;\; \alpha > 0 \tag{9}$$

$$\nabla_x V(x', \theta') \cdot \mathcal{M}_{fwd}(x', \theta') \leq -\alpha V(x_0', \theta') \tag{10}$$

4 Method

Intuition:

The proposed method is motivated by a common physics-class scenario: *when studying a pendulum, students observe its trajectory over time and see its initial and final states.* A natural question is, *can we infer the system parameters solely from observing the trajectory?* and, more deeply, *can we ensure that these parameters guarantee convergence to equilibrium?* We translate this intuition into an inverse problem by replacing the student with a machine learning model. First, the model must learn the dynamics of the underlying system, which requires a surrogate model capable of capturing both stable and unstable trajectories. Second, enforcing stability is challenging; the model must learn a Lyapunov function that constrains the system under its associated parameters. Existing approaches introduce stability by providing a manually designed Lyapunov candidate [3, 14], which limits generality. Our method instead uses a PINN to solve the Zubov equation and automatically learn the Lyapunov structure, following the ideas of [8, 9, 17]. Once the model learns both dynamics and stability, we use these pretrained components to solve the inverse task. This approach advances the state-of-the-art by extending scientific machine learning models beyond state prediction toward understanding whether the inferred parameters meaningfully reflect the underlying physics. Our method strengthens parameter identification by integrating both physical constraints and stability. The complete pipeline is divided into two stages: the pretraining stage and the inverse parameter-identification stage. Within this pipeline, the method includes three main components. The overall structure is shown in Figure 1 and described in the pseudocode section.

Neural Surrogate Model:

When a dynamical system is too complex to solve analytically or is computationally intractable, a surrogate model is employed to approximate its temporal evolution. The most flexible and widely used surrogate in this context is the Deep Operator Network with trainable weights ϕ, which parameterizes the system dynamics.

$$\dot{x} = \mathcal{M}_{fwd_\phi}(x_0, \theta) \tag{11}$$

In this setting, the surrogate model is trained to directly predict the output $\hat{y}$ at a specific time horizon.

$$\hat{y} = \mathcal{M}_{fwd_\phi}(x_0, \theta) \tag{12}$$

Neural Lyapunov Function:

In traditional Lyapunov-based approaches, the Lyapunov function is manually designed and tested as a candidate for stability analysis. To automate this process, we adopt an autonomous learning scheme inspired by [14, 17], where Physics-Informed Neural Networks (PINNs) are employed to learn neural Lyapunov functions for nonlinear dynamical systems. These methods leverage the Zubov partial differential equation (PDE), which provides a continuous formulation for constructing Lyapunov functions.

$$\dot{W} := \nabla W(x) \cdot f(x) = -\Psi(x)(1 - W(x)) \tag{13}$$

Where W denotes the Lyapunov function, $f(x)$ represents the system dynamics, and $\Psi(x)$ is a positive-definite function. The neural network is trained to minimize the residual of the Zubov PDE across the defined domain and its boundary, enforcing the Lyapunov condition. The RoA is then estimated from the learned Lyapunov function W_{NN} as $\mathcal{R} = \{x : W_{NN}(x) < 1\}$. This definition implies that any initial condition where learnerd Lyapunov function is less than 1 lies within the RoA. The learning objective is to maximize this region by extending the learned boundary to 1 to covering the whole RoA.

Inverse Model:

The aim of this block is to find the set of parameters (x', θ') that best match the desired stable outcome of the system while maintaining stability enforced by the learned Lyapunov function. The minimization of the inversion model loss yields this optimal set.

$$x_0', \theta' = arg \min_{x_0 \in X, \theta \in \Theta} \mathcal{L}_{total}(x_0, \theta) \tag{14}$$

The total loss penalizes both the deviation from the desired stable output and violations of the Lyapunov stability constraints within

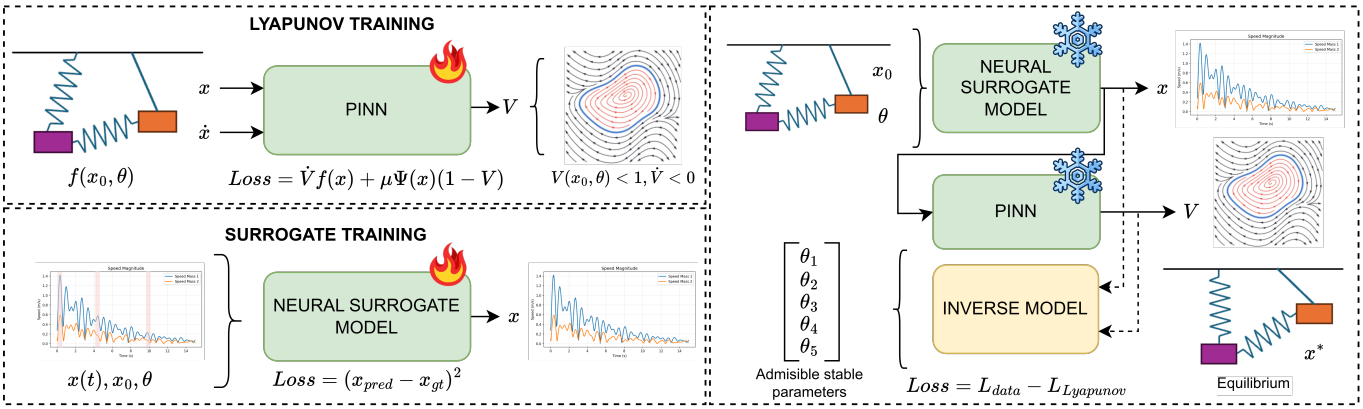

Figure 1: Overview of the stability-constrained inverse design pipeline.

the RoA:

$$\mathcal{L}_{total}(x_0, \theta) = \mathcal{L}_{data}(x_0, \theta) + \mathcal{L}_{lyapunov}(x_0, \theta) \tag{15}$$

The first term corresponds to the surrogate model and penalizes deviations of the predicted output from the desired stable state of the system.

$$\mathcal{L}_{data}(x_0, \theta) = \|\mathcal{M}_{fwd_\phi}(x_0, \theta) - y\|^2 \tag{16}$$

The second term enforces Lyapunov stability by penalizing violations of the RoA conditions.

$$\begin{aligned} \mathcal{L}_{Lyapunov}(x_0, \theta) &= \max(0, -V_\psi(x_0, \theta)) \\ + \max(0, \nabla_x V_\psi(x_0, \theta) \cdot \mathcal{M}_{fwd_\phi}(x_0, \theta) &+ \alpha V_\psi(x_0, \theta)) \end{aligned} \tag{17}$$

5 Experiments

In this section, we assess the physical interpretability of the proposed method. We conduct a experiments to answer the following research questions: **RQ1.** Does the Lyapunov-based PINN correctly represent the system's stability in accordance with Lyapunov's stability conditions? **RQ2.** Does the inverse model return parameter values that enforce system stability? **RQ3.** How can we interpret the role of the Lyapunov constraint in guiding the search for stable parameters?

Dataset

We evaluate the parameter identification model on a damped double-pendulum system that produces complex two-dimensional trajectories. This 2D configuration is selected to limit computational cost, which grows with system dimensionality.The system corresponds to a damped double-pendulum configuration in which two bodies are connected by springs. Its parameters consist of the masses (m_1, m_2), spring constants (k_1, k_2), and damping coefficients (b_1, b_2). The state vector includes the positions $(u_{x_1}, u_{y_1}, u_{x_2}, u_{y_2})$ and velocities $(v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2})$ along both spatial dimensions.

The data was generated using code inspired by the Double 2D spring simulator from MyPhysicsLab [3]. The state vector consists of 8 elements perturbed from the equilibrium state, and the parameter vector contains 6 elements randomly selected from the following ranges: mass $[0.3, 0.7]$, spring constant $[4.0, 8.0]$, and damper $[-0.2, 0.5]$. Using these values, we created 50,000 trajectories sampled every 0.1 seconds over a 15-second window. The data generator was adjusted to include unstable behaviors by allowing negative damping values. All trajectories were computed using the SciPy Runge–Kutta ODE solver.

Furthermore, the dataset was labeled as stable or unstable based on the energy variation along the trajectory, as well as the position and speed error tolerances, defined as the mean deviation of the state from the equilibrium. These criteria are sensitive to overshooting and trajectories that diverge from equilibrium. Since the dataset was collected over a short time window, some systems that would eventually converge were labeled as unstable due to their short-term behavior. Treating these samples as unstable makes the model more robust. The final dataset contains 60% unstable samples and 40% samples that show asymptotic stability. Figure 2 shows the dynamical system simulation extracted from MyPhysicsLab and presents one stable and one unstable sample generated in our dataset.

Implementation Details

The surrogate model was a basis operator learning model composed of three hidden layers, with an input that receives the state vector and parameter vector to approximate the solution of the dynamical system. Training is performed using a mean squared error (MSE) loss. The Lyapunov-based PINN model used two hidden layers and computed the Lyapunov value at each step along the trajectory. The Zubov loss used in the PINN training has a dissipation-rate hyperparameter of $\mu = 0.1$. Both models were trained using the Adam optimizer for 100 epochs with a learning rate of 10^{-3}. The loss of the inverse problem consisted of two components: the loss of data, defined as the MSE of the surrogate model evaluated at equilibrium in the velocity states, and the loss of Lyapunov, inspired by the principle of the region of attraction, which uses an exponential decay rate of $\alpha = 0.1$ to penalize positive Lyapunov derivatives indicating lack of convergence. The entire pipeline was implemented in the PyTorch framework (version 2.7.1+cu11). Experiments were conducted on Ubuntu 22.04.5 LTS using an NVIDIA GTX 1060 GPU.

[3] https://github.com/myphysicslab/myphysicslab

Algorithm 1: Stability-Constrained Inverse Design

Input: Baseline x_0, θ, desired output y^*, learning rates η_1, η_2, η_3, penalty α, tolerance $\varepsilon_{\text{total}}$, number of restarts N_{restarts}, number of epochs N_{epochs}

Output: Admissible set $\mathcal{A} = \{(x_0, \theta)\}$

Train Deep Operator Network:

for $i = 1$ **to** N_{epochs} **do**

Sample batch (x_i, θ_i, y_i) from simulator

Compute $\hat{y}_i \leftarrow M_{\text{fwd}_\phi}(x_i, \theta_i)$

Compute data loss $\mathcal{L}_{\text{data}} \leftarrow \|\hat{y}_i - y_i\|^2$

Update $\phi \leftarrow \phi - \eta_1 \nabla_\phi \mathcal{L}_{\text{data}}$

Train Neural Lyapunov Function:

for $i = 1$ **to** N_{epochs} **do**

Sample x in state space

Compute value: $V_\psi(x)$

Compute gradient $\nabla_x V_\psi(x)$

Compute Lie derivative: $\dot{V}_\psi(x) \leftarrow \nabla_x V_\psi(x) \cdot f(x, \theta)$

Compute Lyapunov loss: $\mathcal{L} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{origin}}$

Zubov Residual: $\mathcal{L}_{\text{PDE}} \leftarrow \left(\dot{V}_\psi(x) + \mu h(x)(1 - V_\psi)\right)^2$

Boundary Condition: $\mathcal{L}_{\text{origin}} \leftarrow V_\psi(0)^2 + \nabla_x V_\psi(0)^2$

Update parameters: $\psi \leftarrow \psi - \eta_2 \nabla_\psi$

Inverse Optimization (Admissible Set Discovery):

Initialize admissible set $\mathcal{A} \leftarrow \emptyset$

for $k = 1$ **to** $N_{restarts}$ **do**

Draw random $x_0' \leftarrow x_0 + \text{Uniform}(-\delta_x, \delta_x)$

Draw random $\theta' \leftarrow \theta + \text{Uniform}(-\delta_\theta, \delta_\theta)$

while *not converged* ***and*** $\mathcal{L}_{\text{total}} > \varepsilon_{total}$ **do**

Compute predicted output: $\hat{y} \leftarrow M_{\text{fwd}_\phi}(x_0', \theta')$

Compute data loss: $\mathcal{L}_{\text{data}}(x_0', \theta') \leftarrow \|\hat{y} - y^*\|^2$

Compute Lyapunov penalty:

$\mathcal{L}_{\text{Lyapunov}}(x_0', \theta') \leftarrow \max(0, -V_\psi(x_0', \theta')) + \max(0, \nabla_x V_\psi(x_0', \theta') \cdot M_{\text{fwd}_\phi}(x_0', \theta') + \alpha V_\psi(x_0', \theta'))$

Compute total loss: $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{Lyapunov}}$

Update:

$x_0' \leftarrow x_0' - \eta_3 \nabla_{x_0'} \mathcal{L}_{\text{total}}$

$\theta' \leftarrow \theta' - \eta_3 \nabla_{\theta'} \mathcal{L}_{\text{total}}$

if $\mathcal{L}_{\text{total}} < \varepsilon_{total}$ **then**

$\mathcal{A} \leftarrow \mathcal{A} \cup \{(x_0', \theta')\}$

return $\mathcal{A}$

5.1 Lyapunov stability constraint

During the initial training stage of the neural Lyapunov function, the model was trained on the complete dataset, which resulted in training difficulties. Specifically, unstable trajectories deviate significantly from equilibrium states, producing large residuals in the Zubov loss and degrading overall model performance. This motivated the decision to restrict training to stable samples only, which exhibit convergence behavior. The Zubov loss seeks to recover the shape and gradient of the Lyapunov function using the system dynamics and a helper function Ψ that minimizes the Zubov

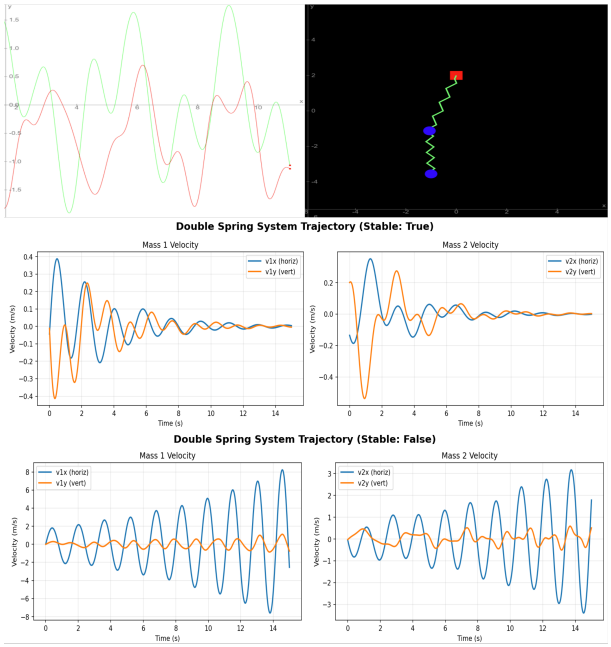

Figure 2: Simulator of Double 2D Spring from MyPhysicLab and Stable/Unstable sample from our dataset

PDE residual (in this work, $\Psi = x^2$). Training on stable trajectories allows the network to learn a consistent Lyapunov function for the subset of the parameter space where stability guarantees hold. Stable samples satisfy the implicit assumption that trajectories converge to the origin, enabling the Zubov residual to remain bounded and tractable. Conversely, unstable samples violate the assumptions of the Zubov formulation, which entails convergence to equilibrium. The inclusion of unstable data during training produces inconsistent gradient signals, where the method attempts to impose stability conditions on diverging trajectories, resulting in unreliable loss terms. Therefore, just training on stable samples ensures coherence between the Zubov formulation and the underlying system behavior.

EXP1. Lyapunov performance:

The PINN evaluation confirms a strong correspondence between the model's predictions and the training loss behavior, forehead mentioned. In this experiment, we assess how well the PINN predicts the Lyapunov value V and its derivative $\dot{V}$. Two samples were selected from the dataset, and their Lyapunov trajectories over time are shown in Fig. 3. For the stable sample $[m_1 = 0.56, m_2 = 0.53, k_1 = 4.07, k_2 = 4.91, b_1 = 0.25, b_2 = 0.47]$, the model satisfies Lyapunov conditions, $\dot{V}$ converges to zero with consistently negative values, and V remains below 1 while steadily decreasing. For the unstable sample $[m_1 = 0.56, m_2 = 0.66, k_1 = 7.04, k_2 = 5.10, b_1 = -0.17, b_2 = 0.19]$, the predicted behavior diverges from expectations. Portions of the trajectory show $V > 1$ indicating instability, but the curve shows a decreasing trend, likely due to the Zubov loss component enforcing asymptotic convergence. A good point is the he derivative $\dot{V}$, which does not converge to zero and oscillates with positive values toward the end, pointing instability. We then

evaluate PINN-based stability predictions using Lyapunov rules and compare them to ground-truth labels through a confusion matrix. The model correctly identifies stable cases but struggles with unstable ones, which is expected because it was trained only on stable trajectories. Adjusting the decision boundary improves classification but requires further study. Overall, the Lyapunov PINN is reliable for stable samples but fails for most unstable ones (**RQ.1**)

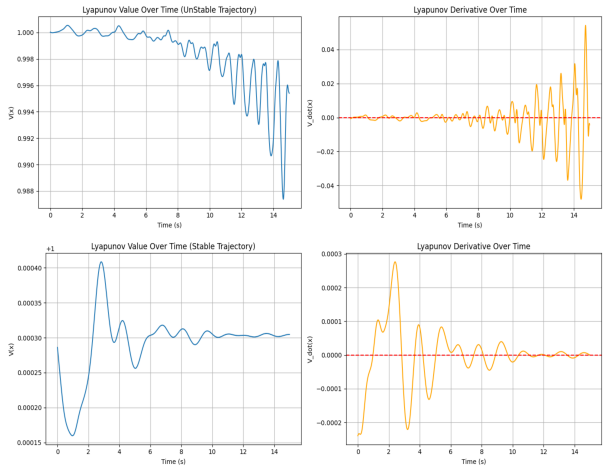

Figure 3: Lyapunpov value and derivative behaviour over the time for unstable and stable sample.

5.2 Stable parameters

EXP2. Inverse performance:

The inverse problem must return parameter values that lead to system stability. To validate this behavior, we design an experiment inspired by the physical intuition behind dampers and springs. Specifically, we check whether the inverse model avoids negative spring constants (physically impossible) and negative damping values (commonly induce instability). The pipeline is evaluated using a fixed initial position and 1000 randomly sampled parameter sets from the ranges: mass $[0.3, 2.0]$, spring constant $[-2, 8]$, and damping coefficient $[-0.5, 2.0]$. Figure 4 presents two fixed initial positions used in the experiment. The purple configuration produces more complex motion due to the large distance between the bodies and from the pivot of the pendulum, whereas the blue configuration, located near the center line and close to the natural spring length, yields simpler dynamics. Each evaluation takes approximately 1 hour and 30 minutes. Observing the frequency plot of the admissible parameters, we notice a clear tendency to avoid negative values, which confirms that the model understands the physical meaning of the system and naturally excludes negative values when searching for stability (**RQ2**). Thus, this trend suggests that with better tuning and improvements to the Lyapunov-based PINN component, the proposed inverse model could produce more reliable results.

EXP3. Lyapunov influence:

However, analyzing in depth the inverse model optimization loss, we observe that it merges the data loss with the Lyapunov loss. The data loss evaluates the difference between the surrogate model's predicted states and the equilibrium, particularly enforcing zero velocity, which always corresponds to the equilibrium state. Meanwhile, the Lyapunov loss is designed to capture the region-of-attraction behavior. A reasonable concern is that the inverse model may rely too strongly on the surrogate model when selecting parameters that lead to stability, since a high velocity at the end of the trajectory could indicate an unstable system. Therefore, the next experiment consists of an ablation study where the same conditions are tested with and without the Lyapunov loss to evaluate its real influence. Figure 4, in the third and fourth rows, shows that when the inverse model relies only on the data loss from the surrogate model, it produces more cases where negative spring and damper values are treated as admissible. The blue squares in Figure 4 highlight the differences in the frequency scale for b_1 and b_2 when the Lyapunov loss is included (rows 1–2) compared to when it is omitted (rows 3–4). These results show that incorporating the Lyapunov term reduces the number of negative parameter values in the corresponding experiments. Furthermore, focusing on the blue case for the damper b_2 (orange circle), the total optimizer loss yields fewer negative values, whereas removing the Lyapunov loss allows the inverse model to extend the admissible range down to -0.5 and select values between $[-0.125, 0]$ as stable. Thus, the Lyapunov loss contributes substantially to obtaining a stable parameter set, reducing the number of incorrect selections that do not lead to stability (**RQ3**).

EXP4. Stability:

Finally, we evaluate the stability objective by isolating the damper coefficient as the parameter of interest, while keeping all remaining parameters positive. To ensure a fair comparison across the parameter space, the damper range is restricted to $[-0.5, 0.5]$. As shown in Figure 5, the inferred parameter distribution exhibits a consistent preference for positive damping values and avoids negative ones, aligning with the physical requirement for stability. Moreover, the model does not concentrate around zero, which typically corresponds to marginal or unstable dynamics. These results indicate that the inverse model systematically rejects negative damping values and selects stable parameter regions (**RQ2**).

6 Discussion and Conclusion

After conducting the experiments and analyzing the losses, we use this section to discuss key findings and outline future directions for refining the proposed method.

In the experiments related to the Lyapunov constraint **EXP1**, the model captures meaningful stability-related insights. However, it fails to satisfy the Lyapunov criteria for identifying instability ($V > 1$ or $\dot{V} > 0$). To address this, we propose an adaptive Zubov loss that trains with the standard Zubov equation for stable trajectories and applies a normalization factor when the trajectory is unstable. As shown in Figure 6, the Lyapunov curves improve for unstable cases, achieving values above 1 for V and significantly positive values for $\dot{V}$, which match the Lyapunov instability conditions. Although this modification introduces drawbacks for stable trajectories, the results indicate that it is possible to train the Lyapunov-based PINN to properly characterize unstable dynamics with appropriate tuning.

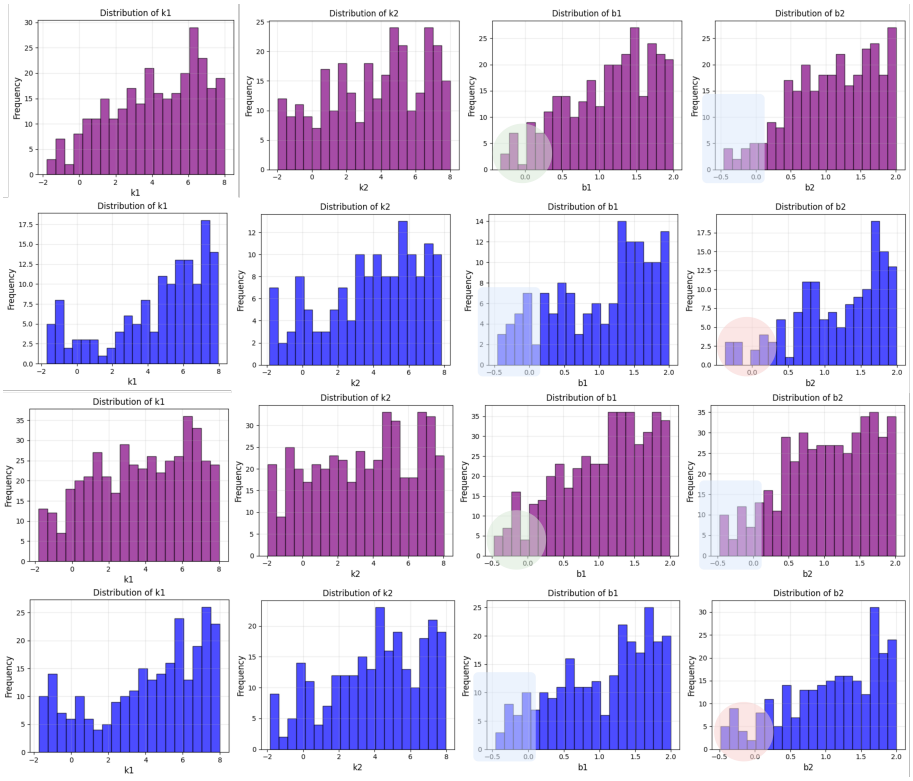

Figure 4: Parameter distribution for reaching stable trajectories. The fixed initial conditions are represented by $[u_{x_1}, u_{y_1}, u_{x_2}, u_{y_2}]$. The purple and blue experiments use the initial conditions $[0.5, 1.0, -0.5, -4.0]$ and $[0.25, 0.5, -0.2, -1.0]$, respectively. The first two rows correspond to the total optimization loss in the inverse model, while the remaining rows use only the data loss.

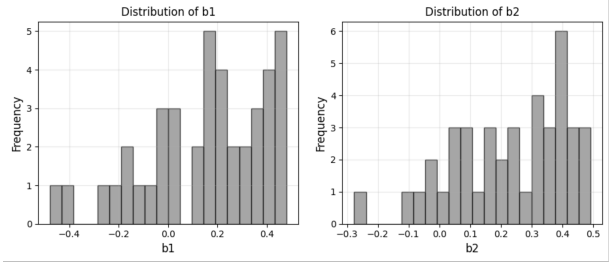

Figure 5: Parameter distribution to reach stable trajectory for the initial condition $[0.25, 0.5, -0.2, -1.0]$.

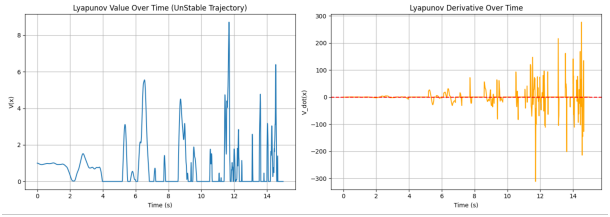

Figure 6: Normalization approach to train Lyapunov for unstable samples

EXP2 highlights the need for additional fixed initial positions to strengthen the conclusion that the inverse model consistently avoids negative parameter values and selects stable parameter ranges. An interesting observation arises from the frequency distribution of the parameters: the blue experiment exhibits a lower overall scale than the purple one, likely because the blue initial position is closer to equilibrium. This suggests that the inverse model may be sensitive to the choice of initial conditions. Moreover, extending the stability analysis to more complex systems (fluid dynamics governed by the Navier–Stokes equations) would provide a more rigorous test, as these systems do not rely on negative parameters to indicate instability. Such systems could offer valuable insight into the model's functionality and scalability.

EXP3 and **EXP4** demonstrate the importance of incorporating a stability loss guided by Lyapunov theory. However, since this area is unexplored, there are no established metrics or baseline models for comparison. For this reason, we evaluate the frequency distribution of parameters identified as admissible. Using this approach, we observe that the model tends to avoid unstable or physically inconsistent parameter values. Additionally, to expand the ablation analysis, we could test how the inverse model performs with different surrogate models, ranging from simple to more advanced architectures. This would allow us to examine whether the Lyapunov loss consistently contributes to identifying reliable stability-inducing parameters. We hypothesize that the inverse model would

achieve better performance if a more accurate surrogate model were used, since it is currently tested with a simple DeepONet. Despite this simplicity, the Lyapunov loss already enables the surrogate to identify stable parameter ranges, indicating promising potential for more advanced architectures.

This project was designed to demonstrate the influence of Lyapunov constraints on parameter identification in dynamical systems, with the goal of obtaining parameter sets that promote stable behavior. Stability is characterized by reduced overshooting and convergence to equilibrium—both captured by the Lyapunov function. To illustrate this idea, we proposed a toy physics experiment to validate the assumptions and the overall pipeline. The results are encouraging and support the hypothesis that the method can recover parameters that drive the system toward stability. Nevertheless, assumptions such as fixed initial conditions and complete knowledge of the trajectory of the system restrict the applicability of the method. Parameter identification is commonly studied under the framework of parameter identifiability analysis, which includes both structural and practical identifiability. Structural identifiability relies on noise-free data, while practical identifiability incorporates uncertainty and estimates parameters through likelihood-based criteria using incomplete or noisy observations [15]. These uncertainty methodologies are widely applied in real-world scenarios, including hydrological modeling under climate-change impacts [6] and stochastic inverse modeling for dam-displacement analysis [2]. Future work will extend our framework toward a probabilistic formulation, enabling uncertainty quantification over both parameter values and the resulting stability properties of the system.

In conclusion, this project introduces a functional and modular pipeline capable of identifying parameter ranges that promote stability in a dynamical system. The framework allows the surrogate model to be replaced with more advanced architectures, avoiding physically invalid negative parameter values that would lead to unstable trajectories. The experiments further demonstrate that the inverse model is effectively guided by the learned Lyapunov function, which imposes the stability constraints required for reliable parameter identification.

References

[1] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 113, 15 (2016), 3932–3937. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113 doi:10.1073/pnas.1517384113

[2] Siyu Chen Tongchun Li Xinbo Zhou P. H. A. J. M. van Gelder Chaoning Lin, Xiaohu Du. 2024. On the multi-parameters identification of concrete dams: A novel stochastic inverse approach. *International Journal for Numerical and Analytical Methods in Geomechanics* 48, 16 (Aug 2024), 3792–3810. doi:10.1002/nag.3812

[3] Haoyu Chu, Yuto Miyatake, Wenjun Cui, Shikui Wei, and Daisuke Furihata. 2024. Structure-preserving physics-informed neural networks with energy or Lyapunov structure. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence* (Jeju, Korea) *(IJCAI '24)*. Article 428, 9 pages. doi:10.24963/ijcai.2024/428

[4] Ján Drgoňa, Truong X. Nghiem, Thomas Beckers, Mahyar Fazlyab, Enrique Mallada, Colin Jones, Draguna Vrabie, Steven L. Brunton, and Rolf Findeisen. 2025. Safe Physics-informed Machine Learning for Dynamics and Control. In *2025 American Control Conference (ACC)*. 591–606. doi:10.23919/ACC63710.2025.11107836

[5] R. Fuentes, N. Dervilis, K. Worden, and E.J. Cross. 2019. Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning. *Journal of Physics: Conference Series* 1264, 1 (jul 2019), 012050. doi:10.1088/1742-6596/1264/1/012050

[6] Paulo A. Herrera, Miguel Angel Marazuela, and Thilo Hofmann. 2022. Parameter estimation and uncertainty analysis in hydrological modeling. *WIREs Water* 9, 1 (2022), e1569. arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wat2.1569 doi:10.1002/wat2.1569

[7] Morgan Jones and Matthew M. Peet. 2021. Converse Lyapunov Functions and Converging Inner Approximations to Maximal Regions of Attraction of Nonlinear Systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*. 5312–5319. doi:10.1109/CDC45484.2021.9682949

[8] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. 2023. Towards Learning and Verifying Maximal Neural Lyapunov Functions. In *2023 62nd IEEE Conference on Decision and Control (CDC)*. 8012–8019. doi:10.1109/CDC49753.2023.10383299

[9] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. 2024. TOOL LyZNet: A Lightweight Python Tool for Learning and Verifying Neural Lyapunov Functions and Regions of Attraction. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control* (Hong Kong SAR, China) *(HSCC '24)*. Association for Computing Machinery, New York, NY, USA, Article 25, 8 pages. doi:10.1145/3641513.3650134

[10] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. 2025. Physics-informed neural network Lyapunov functions: PDE characterization, learning, and verification. *Automatica* 175 (May 2025), 112193. doi:10.1016/j.automatica.2025.112193

[11] Anthony N Michel, Ling Hou, and Derong Liu. 2008. *Stability of dynamical systems*. Springer.

[12] Wenhao Mu, Zhi Cao, Mehmed Uludag, and Alexander Rodríguez. 2025. Counterfactual Probabilistic Diffusion with Expert Models. arXiv:2508.13355 [cs.LG] https://arxiv.org/abs/2508.13355

[13] Niklas Pfister, Stefan Bauer, and Jonas Peters. 2019. Learning stable and predictive structures in kinetic systems. *Proceedings of the National Academy of Sciences* 116, 51 (2019), 25405–25411. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1905688116 doi:10.1073/pnas.1905688116

[14] Ivan Dario Jimenez Rodriguez, A. Ames, and Yisong Yue. 2022. LyaNet: A Lyapunov Framework for Training Neural ODEs. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:246634091

[15] Matthew J Simpson and Ruth E Baker. 2025. Parameter identifiability, parameter estimation and model prediction for differential equation models. arXiv:2405.08177 [stat.ME] https://arxiv.org/abs/2405.08177

[16] Simón C. Smith and Subramanian Ramamoorthy. 2020. Counterfactual Explanation and Causal Inference In Service of Robustness in Robot Control. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. 1–8. doi:10.1109/ICDL-EpiRob48136.2020.9278061

[17] Junkai Wang, Yuxuan Zhao, Mi Zhou, and Fumin Zhang. 2025. Learning Robust Regions of Attraction Using Rollout-Enhanced Physics-Informed Neural Networks with Policy Iteration. arXiv:2508.19398 [eess.SY] https://arxiv.org/abs/2508.19398

[18] Alistair White, Niki Kilbertus, Maximilian Gelbrecht, and Niklas Boers. 2023. Stabilized neural differential equations for learning dynamics with explicit constraints. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 568, 22 pages.

[19] Ruikun Zhou, Thanin Quartz, Hans De Sterck, and Jun Liu. 2022. Neural Lyapunov Control of Unknown Nonlinear Systems with Stability Guarantees. arXiv:2206.01913 [eess.SY] https://arxiv.org/abs/2206.01913