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

Daniel Menacho Ordoñez

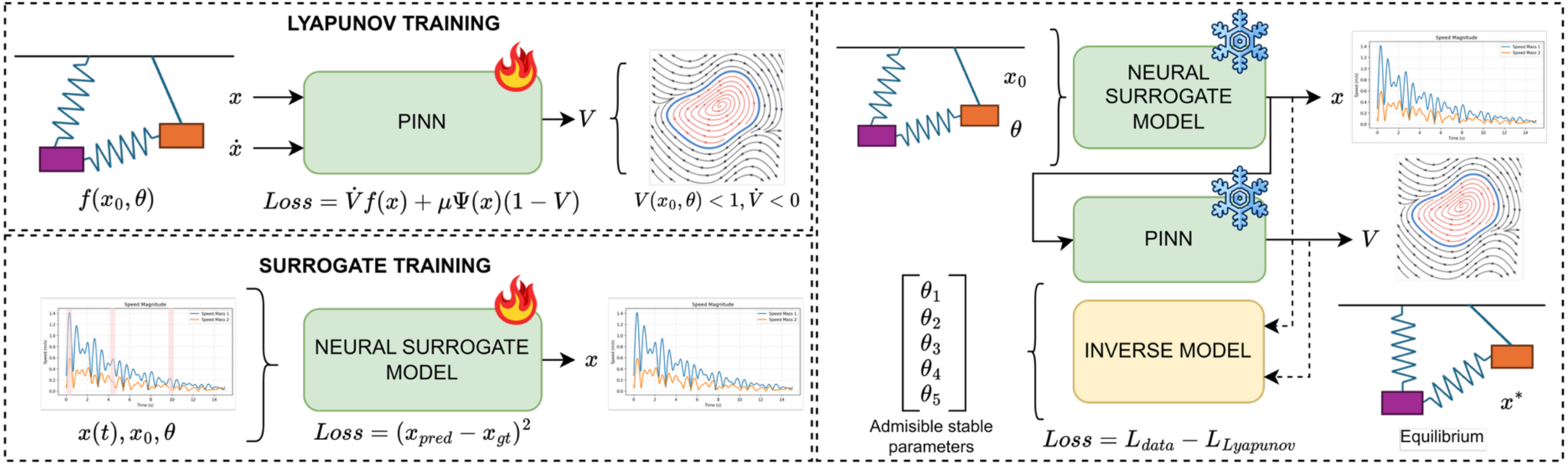
Problem Statement

- Dynamical systems are governed by complex differential equations that require **accurate parameter** values to achieve the desired stable behavior.
  - An unstable system doesn't converge to the equilibrium and could provoke chaotic behaviors.
  - The traditional method for stability analysis relies on the manual design of a Lyapunov function and its inequalities.
  - PINN and Neural ODEs analyze stability through Lyapunov

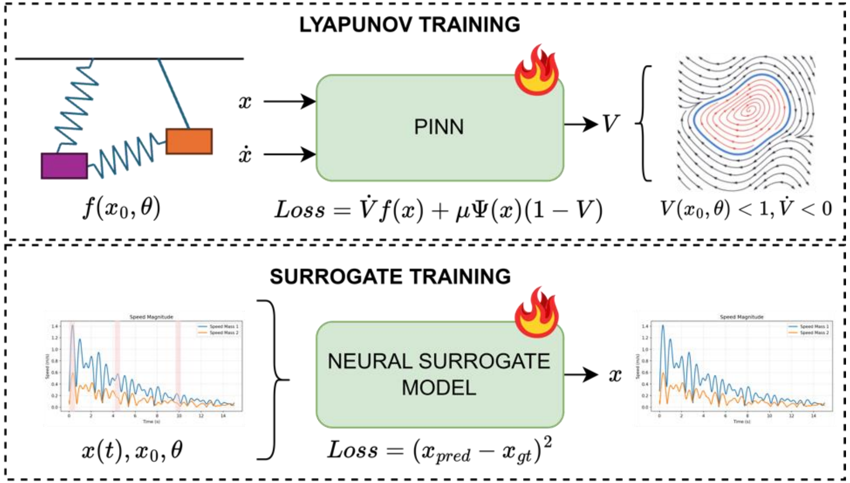
Contribution

- **AIM:** Design an inverse modeling framework constrained by a learnable Lyapunov function to identify parameter sets that induce stability
  - Introducing a generalizable framework adaptable to a wide range of neural surrogate models.
  - Enhancing the decision-making task to reach a safe and stable dynamic behavior

Problem Definition



Proposed Method



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

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

Proposed Method

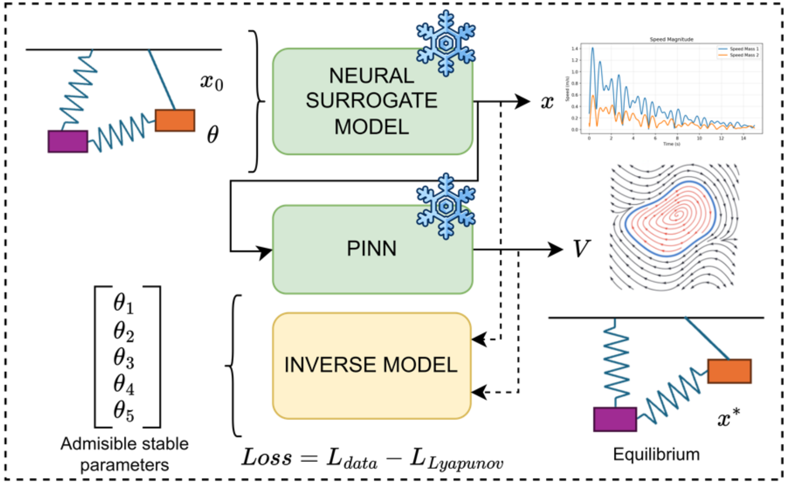
Inverse Optimization (Admissible Set Discovery):

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

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

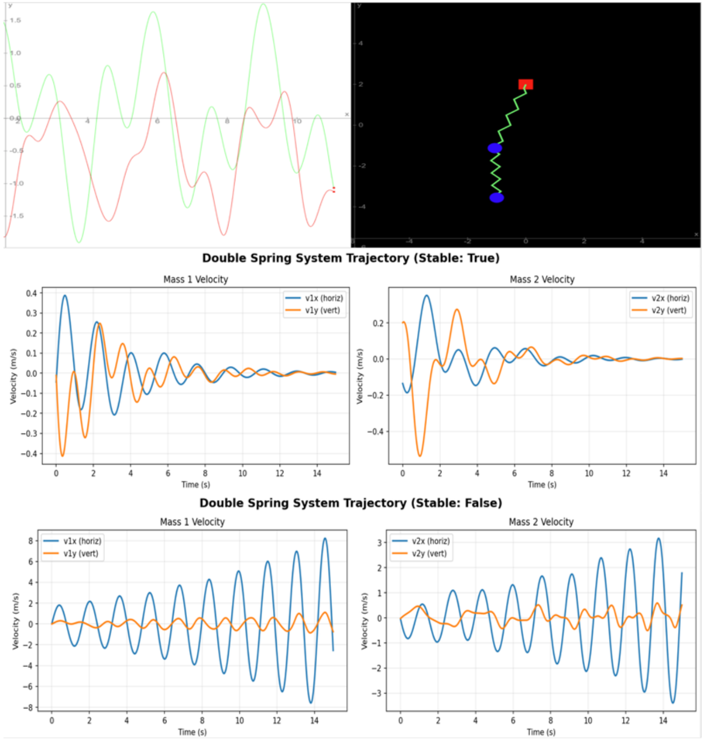
- Draw random $x_0' \leftarrow x_0 + \text{Uniform}(-\delta_x, \delta_x)$
- Draw random $\theta' \leftarrow \theta + \text{Uniform}(-\delta_\theta, \delta_\theta)$
- **while** *not converged* ***and*** $\mathcal{L}_{\text{total}} > \varepsilon_{total}$ **do**
  - Compute predicted output: $\hat{y} \leftarrow M_{\text{fwd}_\phi}(x_0', \theta')$
  - 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 $\mathcal{A}$



Dataset

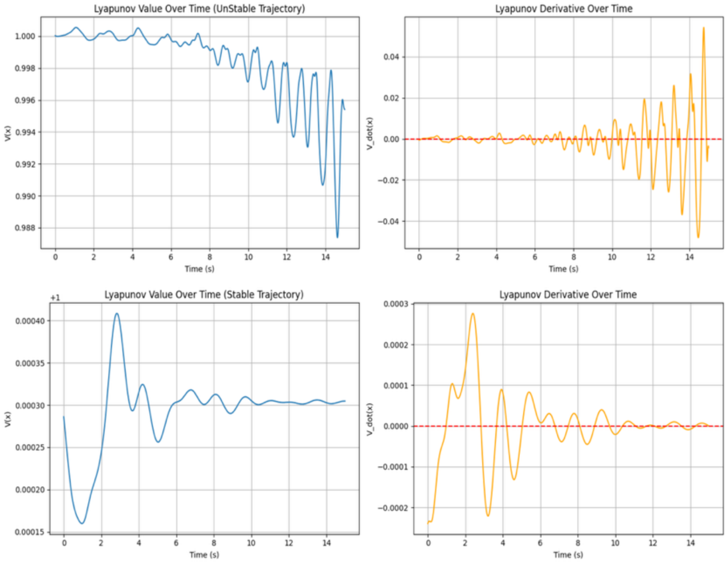
- Damped double-pendulum configuration.
  - 50,000 trajectories sampled every 0.1s
  - 60% unstable and 40% stable samples
  - Parameters: masses [0.3, 0.7], spring constant [-4.0, 8.0], damping coefficient [-0.2, 0.5]
  - State vector: position and velocities in 2D
  - Runge-Kutta ODE Solver

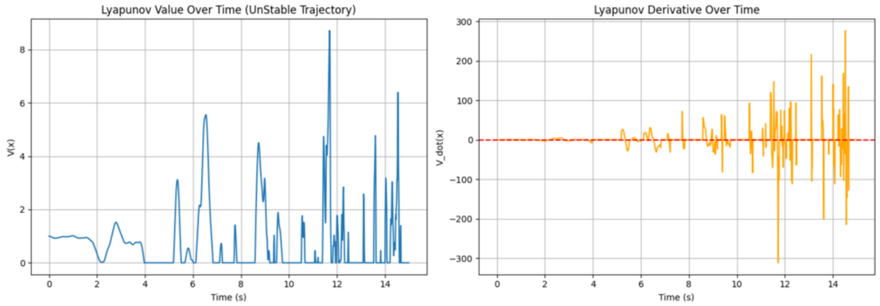


UNIVERSITY OF MICHIGAN

Results

- **RQ1:** Does the Lyapunov-based PINN correctly represent the system’s stability in accordance with Lyapunov’s stability conditions?



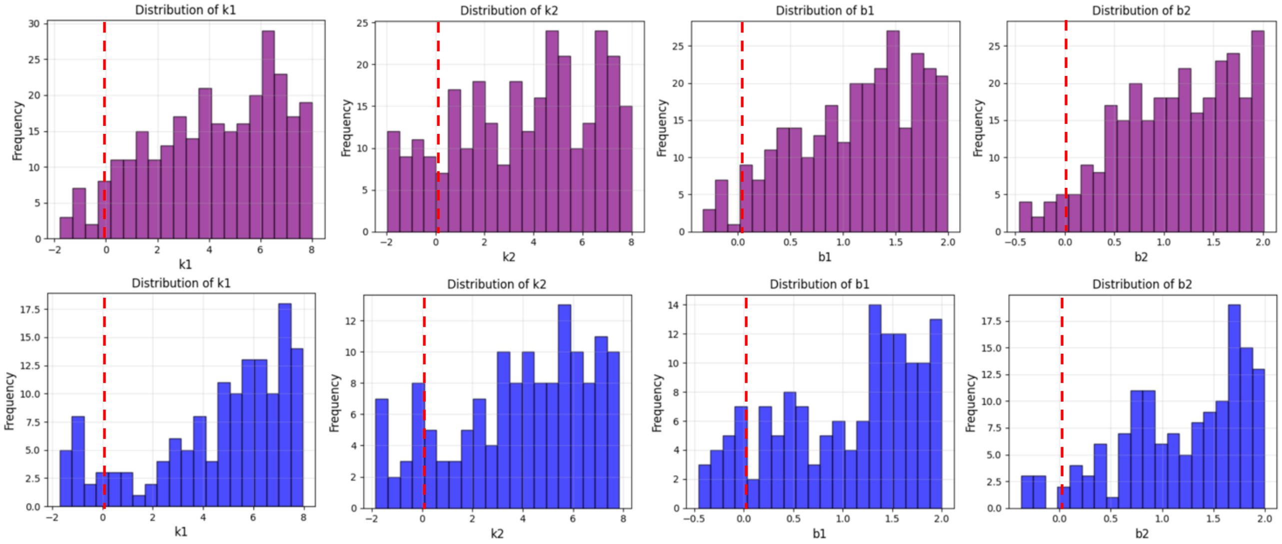


Loss Normalization for

unstable samples

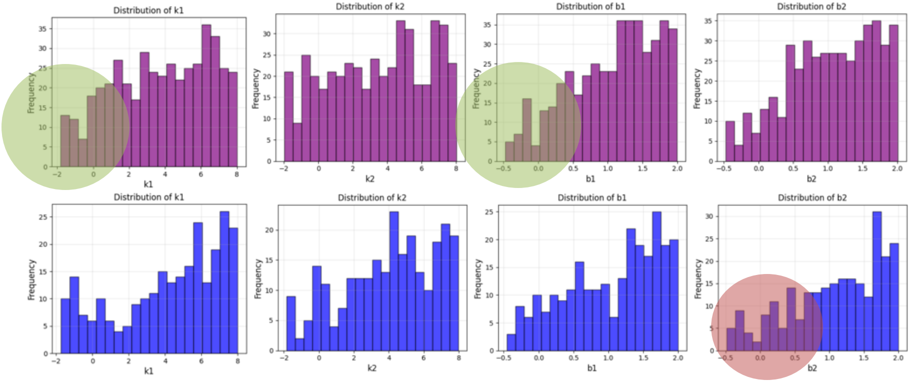
Results

- **RQ2:** Does the inverse model return parameter values that enforce system stability?

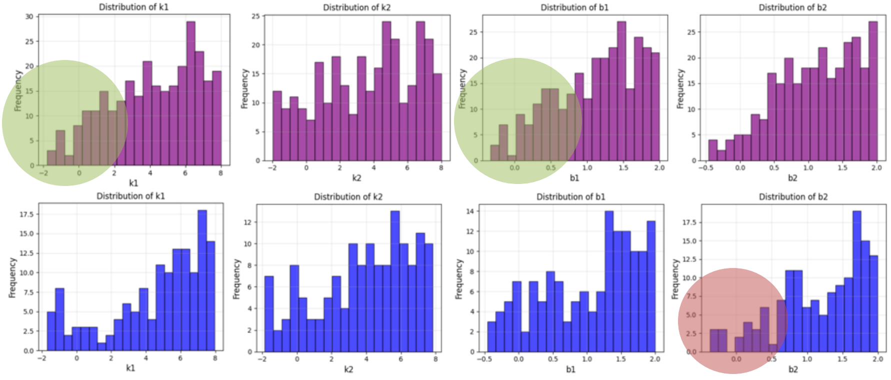


Results

- **RQ3:** How can we interpret the role of the Lyapunov constraint in guiding the search for stable parameters?



Loss without the Lyapunov term



Loss with the Lyapunov term

Conclusion

- This project introduces a functional and modular pipeline
- The inverse model is effectively guided by the learned Lyapunov function, which imposes the stability constraints required for reliable parameter identification.
- Future work:
  - Adaptive Lyapunov training for stable and unstable samples.
  - Test how the inverse model performs with different surrogate models, ranging from simple to more advanced architectures.