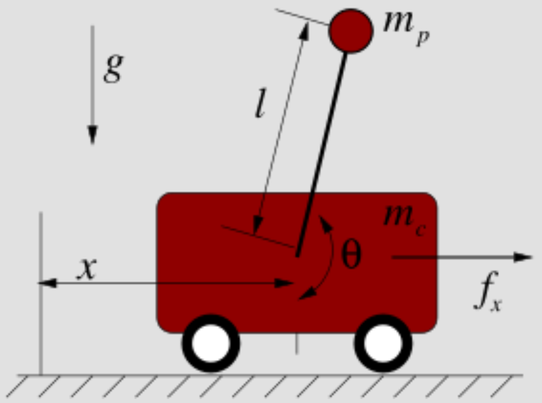
Neural Lyapunov Control for Discrete-Time Systems

Junlin Wu, Andrew Clark, Yiannis Kantaros and Yevgeniy Vorobeychik

Washington University in St. Louis

Learning Stable Policies

- Discrete-time <u>nonlinear dynamics</u>: $x_{t+1} = f(x_t, u_t)$, where x_t, u_t are state and control at time t.
- **Our Goal:** learn a provably stable policy $u_t = \pi(x_t)$.
- **Stability:** a dynamical system converges to a nominal state whenever the starting state is in a "region of attraction (RoA)".



Nominal state: $(x, \dot{x}, \theta, \dot{\theta}) = (0,0,\pi,0)$

Lyapunov Stability (Conventional)

- Lyapunov Stability (discrete-time controlled dynamical systems): If policy $u_t = \pi(x_t)$ and Lyapunov function $V(x)$ satisfy the below conditions, then $x = 0$ is stable.

Lyapunov conditions

1) $V(0) = 0,\ V(x) > 0\ \forall x \neq 0$
2) $V(f(x_t, u_t)) < V(x_t)\ \forall x, u_t = \pi(x_t)$

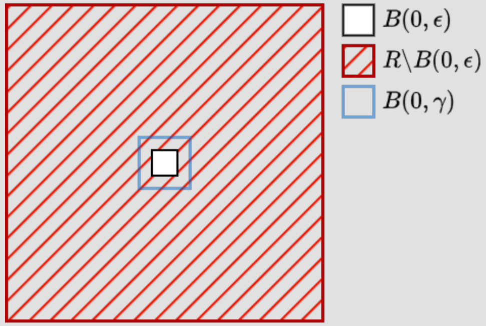
- **Goal:** synthesize a Lyapunov function $V(x)$ and policy (controller) $u = \pi(x)$ over a region R such that conditions 1) and 2) hold on R.
- Sub-level set $D = \{x \in R \mid V(x) \leq \beta\}$ is RoA.

Approximately Lyapunov Stability

- Key Challenge: verification does not work near the origin (numerical instability, precision limits, etc).

ϵ-Lyapunov conditions over R

1) $V(0) = 0,\ V(x) > 0\ \forall x \in R \backslash B(0,\epsilon)$
2) $\exists \eta > 0 : V(f(x, \pi(x))) \leq V(x) - \eta\ \forall x \in R \backslash B(0,\epsilon), u = \pi(x)$



- Theorem [informal]: Under these conditions, if we start in RoA D:

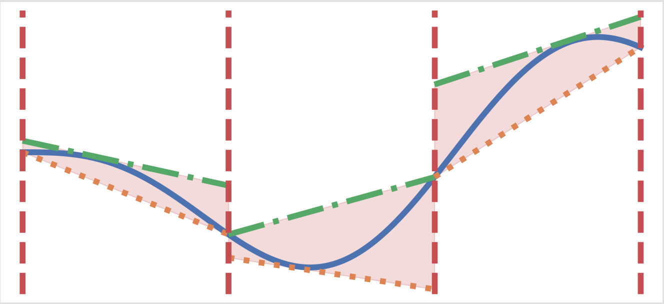
a) we will reach $B(0,\epsilon)$ in finite time,

b) we will reach $B(0,\epsilon)$ infinitely often, and

c) for any γ, there is ϵ such that ϵ-stability implies that we converge to $B(0,\gamma)$ in finite time.

Verification Algorithm (MILP)

- We represent policies $\pi_\beta(x)$ and Lyapunov functions $V_\theta(x)$ as NNs with ReLU activation function.
- The main challenge is to verify the term $V(f(x, \pi(x))) \leq V(x) - \eta$ $\forall x \in R \backslash B(0, \epsilon)$, where the dynamics $f(x_t, u_t)$ is nonlinear.
- We split the region R into grid, and use linear function to upper/lower bound $f(x_t, u_t)$ within each sub-region.
  - The problem can now be written into MILP.
  - Automatically refine grid for tighter bounds.



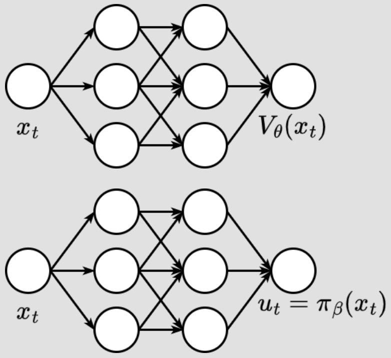
1d illustration

Learning Algorithm

- The goal now is to jointly learn (π_β, V_θ) to provably satisfy the ϵ-Lyapunov stability conditions.

Lyapunov loss function

$$\min_{\theta,\beta} \sum_{S} L(x; V_\theta, \pi_\beta)$$



- Counterexamples in set S comes from: 1) a novel MILP-based verifier (slow); 2) a novel gradient-based approach (fast).

Gradient-Based Approach (Counterexamples)

- We use *projected gradient descent* (PGD) to solve two optimization problems which enables faster counterexample generation:

$$\min_{x \in R} V_\theta(x)$$

$$\min_{x \in R} V_\theta(x) - V_\theta(f(x, \pi_\beta(x)))$$

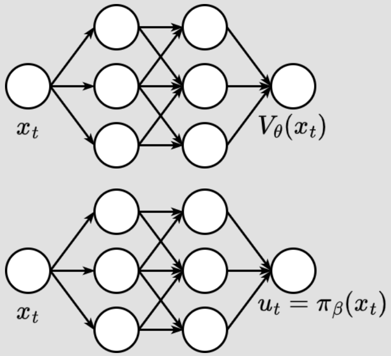
- PGD: $x_{k+1} = \Pi\{x_k - \alpha_k \mathrm{sgn}(\nabla F_\theta(x_k))\}$, where $F(\cdot)$ is the objective for the minimization problem.

Learning Algorithm

- The goal now is to jointly learn (π_β, V_θ) to provably satisfy the ϵ-Lyapunov stability conditions.

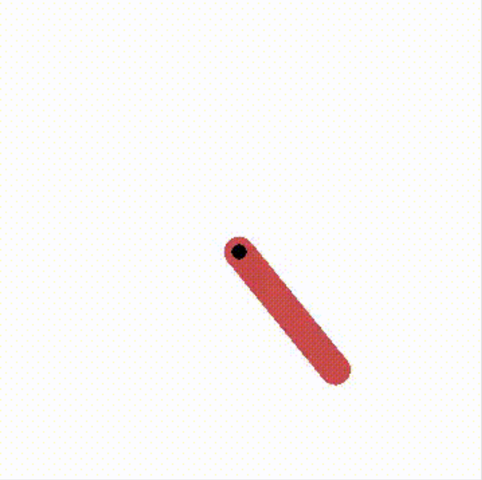
Lyapunov loss function

$$\min_{\theta,\beta} \sum_{S} L(x; V_\theta, \pi_\beta)$$

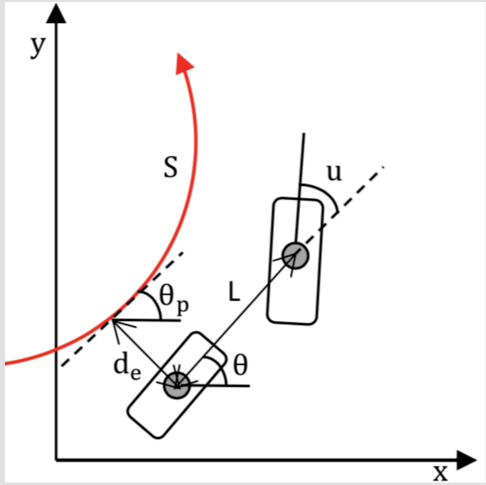


- Counterexamples in set S comes from: 1) a novel MILP-based verifier (slow); 2) a novel gradient-based approach (fast).
- Keep training until $\pi_\beta(x)$ and $V_\theta(x)$ pass verification.

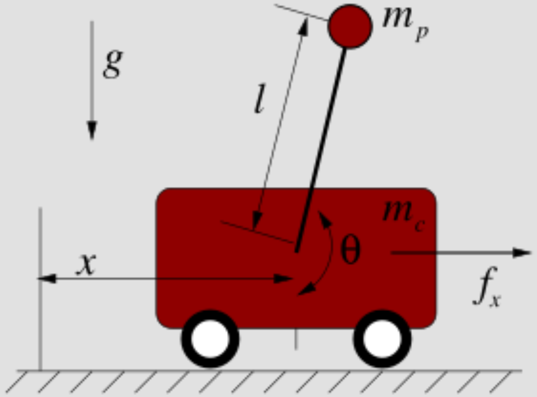
Experiments



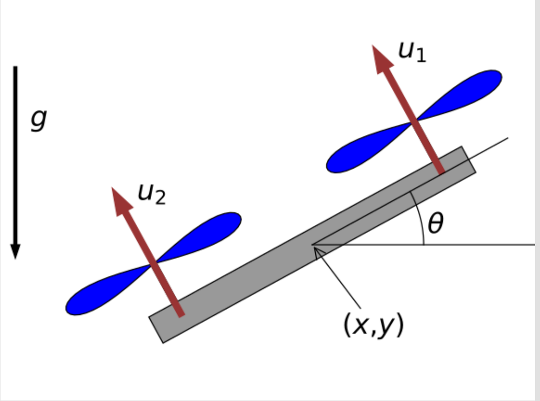
Inverted Pendulum



Path Tracking



CartPole



PVTOL

Experiments

Table 1: Inverted Pendulum

	Valid Region	Runtime (s)	ROA	Max ROA	Success Rate
NLC (free)	$\|\|x\|\|_2 \leq 6.0$	28 ± 29	11 ± 4.6	22	100%
NLC (max torque 6.0)	$\|\|x\|\|_2 \leq 6.0$	519 ± 184	13 ± 27	66	20%
UNL (max torque 6.0)	$\|\|x\|\|_2 \leq 4.0$	821 ± 227	1 ± 2	7	30%
LQR	$\|\|x\|\|_\infty \leq 5.8$	$< \mathbf{1}$	14	14	success
SOS	$\|\|x\|\|_\infty \leq 1.7$	$< \mathbf{1}$	6	6	success
DITL	$\|\|x\|\|_\infty \leq 12$	8.1 ± 4.7	$\mathbf{61 \pm 31}$	**123**	100%

- Key observations: our approach is both much faster, and much more effective than prior art for learning provably stable policies.

Experiments

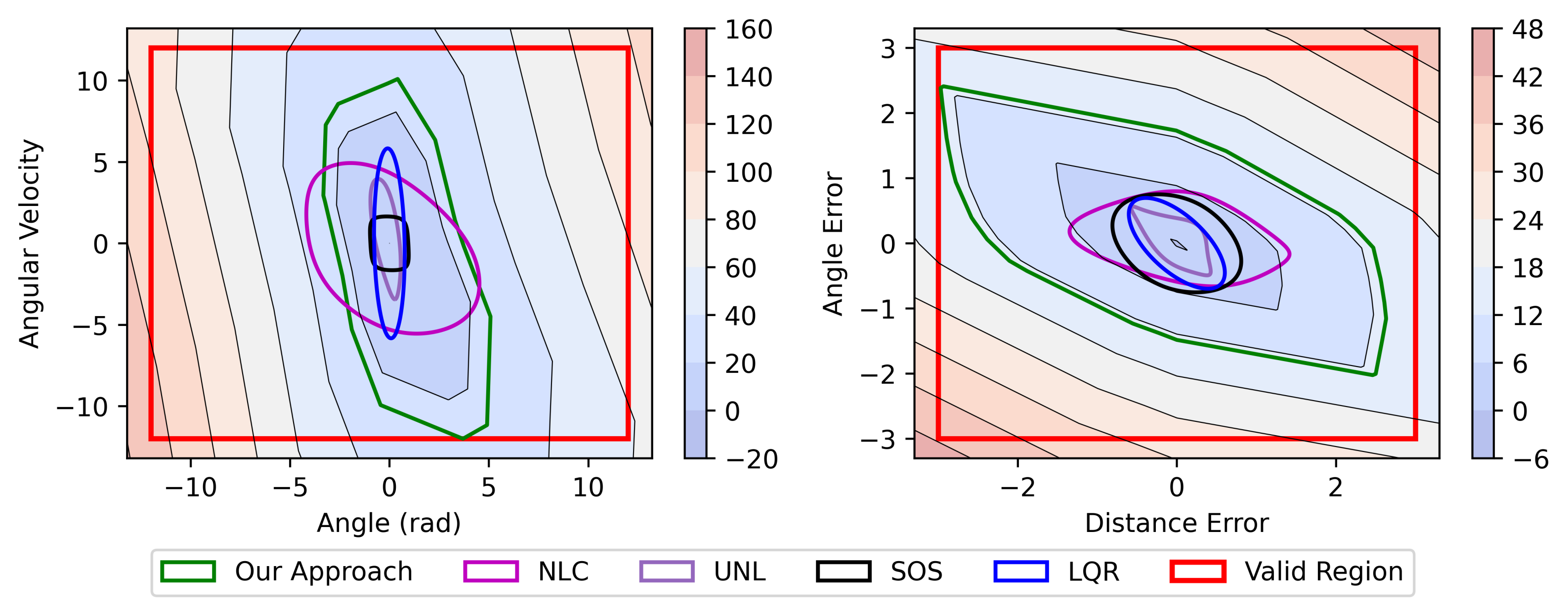
Table 2: Path Tracking

	Valid Region	Runtime (s)	ROA	Max ROA	Success Rate
NLC	$\|\|x\|\|_2 \leq 1.0$	109 ± 81	0.5 ± 0.2	0.76	100%
NLC	$\|\|x\|\|_2 \leq 1.5$	151 ± 238	1.4 ± 0.9	2.8	80%
UNL	$\|\|x\|\|_2 \leq 0.8$	925 ± 110	0.1 ± 0.2	0.56	10%
LQR	$\|\|x\|\|_\infty \leq 0.7$	$< \mathbf{1}$	1.02	1.02	success
SOS	$\|\|x\|\|_\infty \leq 0.8$	$< \mathbf{1}$	1.8	1.8	success
DITL (LQR)	$\|\|x\|\|_\infty \leq 3.0$	9.8 ± 4	8 ± 3	12.5	100%
DITL (RL)	$\|\|x\|\|_\infty \leq 3.0$	14 ± 11	$\mathbf{9 \pm 3.5}$	**16**	100%

- Key observations: our approach is both much faster, and much more effective than prior art for learning provably stable policies.

Experiments

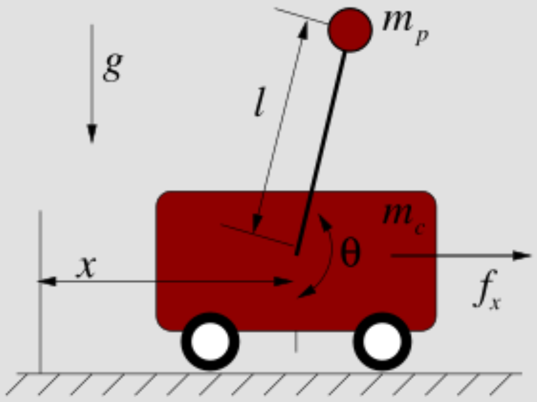
ROA plot of inverted pendulum (left) and path tracking (right)



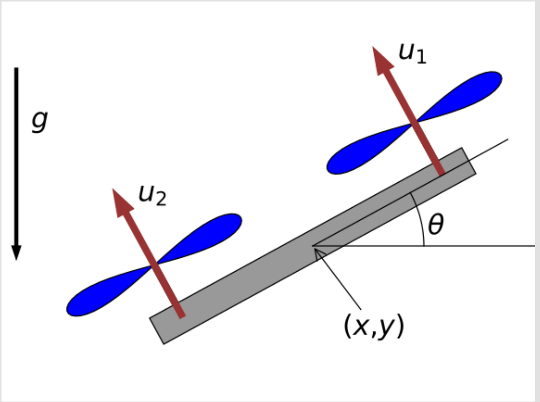
- Key observations: our approach is both much faster, and much more effective than prior art for learning provably stable policies.

Experiments

- In two more complex domains (CartPole and PVTOL), ours is the first automated approach to achieve provable stability for actual underlying nonlinear dynamics.



$(x, \dot{x}, \theta, \dot{\theta}) = (0,0,\pi,0)$



$(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}) = (0,0,0,0,0,0)$

Takeaways

- We utilize the structure of the Lyapunov condition in discrete-time nonlinear systems to enhance verification efficiency.
- We introduce a gradient-based algorithm for rapid counterexample generation, accelerating the model training process.
- We propose “approximately Lyapunov stability”, which formalizes the impact of numerical instability issues of verifying near the origin.
- Our approach outperforms SOTA methods.

Thank you!

Paper

Code