Proceedings of Machine Learning Research vol 211:1–17, 2023 5th Annual Conference on Learning for Dynamics and Control

Hybrid Systems Neural Control with Region-of-Attraction Planner

Yue Meng MENGYUE@MIT.EDU

Chuchu Fan CHUCHU@MIT.EDU

Massachusetts Institute of Technology

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

Hybrid systems are prevalent in robotics. However, ensuring the stability of hybrid systems is challenging due to sophisticated continuous and discrete dynamics. A system with all its system modes stable can still be unstable. Hence special treatments are required at mode switchings to stabilize the system. In this work, we propose a hierarchical, neural network (NN)-based method to control general hybrid systems. For each system mode, we first learn an NN Lyapunov function and an NN controller to ensure the states within the region of attraction (RoA) can be stabilized. Then an RoA NN estimator is learned across different modes. Upon mode switching, we propose a differentiable planner to ensure the states after switching can land in next mode's RoA, hence stabilizing the hybrid system. We provide novel theoretical stability guarantees and conduct experiments in car tracking control, pogobot navigation, and bipedal walker locomotion. Our method only requires 0.25X of the training time as needed by other learning-based methods. With low running time (10∼50X faster than model predictive control (MPC)), our controller achieves a higher stability/success rate over other baselines such as MPC, reinforcement learning (RL), common Lyapunov methods (CLF), linear quadratic regulator (LQR), quadratic programming (QP) and Hamilton-Jacobian-based methods (HJB). The project page is on `https://mit-realm.github.io/hybrid-clf`.

Keywords: Hybrid system, Control Lyapunov functions, Region of Attraction

1. Introduction

Learning how to control hybrid systems is critical in the realm of robotics and artificial intelligence, given the wide variety of hybrid systems in autonomous driving (Ning et al., 2021), locomotion for bipedal robots, and UAVs (Gillula et al., 2011). However, it remains challenging to analyze the stability and design controllers for general nonlinear hybrid systems, due to the intricate dynamics involving both the continuous flows and discrete jumps (Chen et al., 2021).

Various approaches in classic control emerged to analyze the stability for a certain type of hybrid systems, such as piecewise affine (PWA) systems (Johansson, 2003; Pettersson and Lennartson, 1996, 1999; Prajna and Papachristodoulou, 2003) and periodic systems (Poincaré, 1885; Clark et al., 2018; Manchester, 2011; Manchester et al., 2011). However, these methods either work on simple systems in low dimensions or rely on computation-heavy methods for verification and synthesis (Abate et al., 2020; Jarvis-Wloszek et al., 2003; Topcu et al., 2008; Majumdar et al., 2013).

The pivot hurdle for stabilizing general hybrid systems is to properly handle the system at the mode switching (De Schutter et al., 2009). A hybrid system, provided with all its system modes stable, can still be unstable if the mode switching is too fast (Branicky, 1998). Using multiple Lyapunov functions, some works constraint the average dwell time (ADT) (Hespanha and Morse,

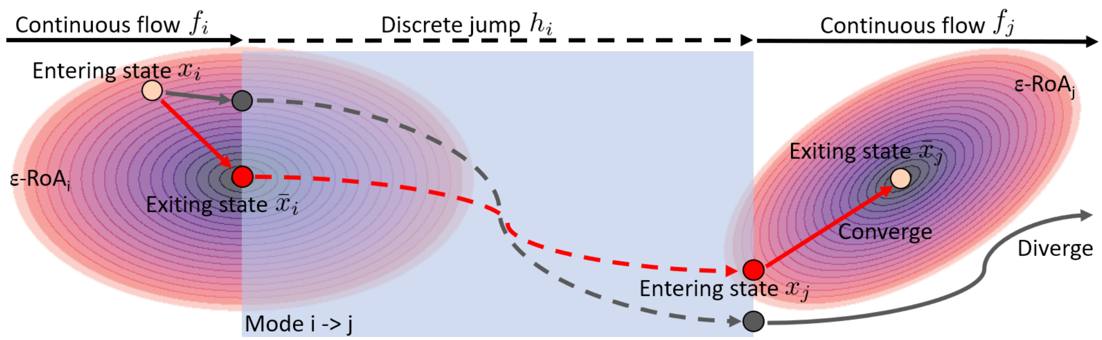


Figure 1: We learn the stabilizing control and the RoA for each system mode. In testing phase, our method (red lines) plans the configuration (the exiting state and the mode) to ensure the next entering state is always within the RoA of the next equilibrium, whereas traditional methods (gray lines) directly tracking the next equilibrium will diverge after the jump.

1999; Zhai et al., 2000) so the system switching is "slow-on-the-average", but they cannot handle discrete jumps. Other methods enforce the sub-sequence of each Lyapunov function at switch-in instants decreasing (Branicky, 1998). Those methods track each Lyapunov function's switching sequence, which is hard to implement when there are many system modes.

Motivated by region of attraction (RoA)-based planning methods (Tedrake et al., 2010), we propose an RoA-based approach to stabilize hybrid systems. The idea is to let the state always enter next mode's RoA after switching. We ensure the stability of the system under each mode by using control Lyapunov functions (CLF), and we use the invariant set provided by the Lyapunov function to represent the RoA for the system under each mode. In this way, we don't need to optimize the whole sequence of the Lyapunov functions but only consider the Lyapunov function (and RoA) in consecutive modes, which is more efficient to implement for stability analysis and controller synthesis. Constructing Lyapunov functions used to be time-consuming and is limited to low-dimension systems (Abate et al., 2020; Jarvis-Wloszek et al., 2003; Topcu et al., 2008; Majumdar et al., 2013). Recently there is a trend in using neural networks to construct control Lyapunov and Barrier certificates (Chang et al., 2019; Dawson et al., 2022a; Qin et al., 2021; Meng et al., 2021; Sun et al., 2020). We follow this direction and further use neural networks for RoA estimation.

The whole pipeline is as follows: we first collect samples under each system mode, use the neural network controller to generate trajectories, and construct neural Lyapunov certificates to ensure the stability for each mode. Then we estimate the RoA by the Lyapunov level-set and learn each system mode's stable level-set using the Neural RoA estimator. Finally, upon deployment, we use a differentiable planner to find the optimal configuration before mode switching to ensure the next state will be within the RoA of the next mode, hence achieving the stability of the hybrid system. Although there might exist a gap between theoretical guarantees and simulation performance due to imperfect learning of neural networks, in practice we observe strong empirical results.

We conduct experiments on three challenging scenarios (car tracking control on different road conditions, pogobot navigation, and bipedal walker locomotion). We achieve the best performance (mean square error, failure rate, etc) compared to other baselines (MPC, RL, LQR, CLF, QP, HJB).

Our learned method requires less training time than RL-based methods or the HJB approach, and at the evaluation stage, the runtime is only 1/50X∼1/10X the time for MPC.

Our contributions are: (1) we are the first to use a neural-network RoA estimator, planner and controller to stabilize hybrid systems within certain RoA. (2) we define a new stability for hybrid systems and derive sufficient conditions to enforce the stability with theoretical guarantees (3) our approach can be applied to different hybrid systems even if the dynamics is unknown, where the controller can be state-feedback control or apex-to-apex control for periodic systems. (4) we conduct challenging experiments that involve complicated hybrid systems and outperform other baselines.

2. Related Work

Controlling hybrid systems is challenging and has been studied for decades. Various formulations for system modelling and control strategies have been extensively presented in (DeCarlo et al., 2000; De Schutter et al., 2009; Davrazos and Koussoulas, 2001; Sanfelice, 2013). We only list a few closely related works in stability analysis and controller synthesis.

Lyapunov stability analysis: The common Lyapunov function is used to prove hybrid system stability in (Dogruel and Ozguner, 1994; Fierro, 1997) . Multiple Lyapunov functions are proposed in (Peleties and Decarlo, 1992) for linear systems and (Branicky, 1998; Michel and Hu, 1999) for more general cases where the monotonous decreasing condition is relaxed. In (Malmborg, 1998), non-smooth Lyapunov functions are used for hybrid controller synthesis. For piecewise affine (PWA) systems, linear matrix inequality based approaches are proposed to construct Lyapunov certificates (Johansson, 2003; Pettersson and Lennartson, 1996). For switched systems, average dwell time (ADT) is introduced in (Hespanha and Morse, 1999; Zhai et al., 2000) to tie the stability with the ratio between stable and unstable modes. For periodic systems, Poincaré map (Clark et al., 2018) and transverse Lyapunov functions (Manchester, 2011) are used to analysis the limit cycle stability. Although equipped with theoretical guarantees for the stability, the methods above are either only suited for a specified type of systems (linear or PWA) or require expert knowledge to design the Lyapunov functions, or rely on tedious numerical methods such as Satisfiability Modulo Theories (SMT) solvers (Abate et al., 2020) and sum-of-squares (SoS) optimization (Jarvis-Wloszek et al., 2003; Topcu et al., 2008). For complicated systems, recently there is a growing trend to approximate the Lyapunov functions and the controllers using data-driven methods like Gaussian Process (Zhai and Nguyen, 2019; Xiao et al., 2020), SVM (Sun et al., 2005) and neural networks (Richards et al., 2018; Chang et al., 2019; Mehrjou et al., 2020; Dawson et al., 2022a,b). We are aligned with this and further study to stabilize hybrid systems using neural networks.

Hybrid system control designs: Upon grounding work on Lyapunov theory (Sontag, 1983; Artstein, 1983), a wide body of literature exists on synthesizing feedback controllers for switched linear and affine systems (Wicks and DeCarlo, 1997; Johansson, 1999; Mignone et al., 2000). There are also many works beyond Lyapunov controllers for more general hybrid systems, such as optimal control (Cassandras et al., 2001; Cho et al., 2001), model predictive control (MPC) (Slupphaug and Foss, 1997; Lazar, 2006; Camacho et al., 2010; Marcucci and Tedrake, 2020) and Hamilton-Jacobian reachability-based (HJB) methods (Choi et al., 2022), and region-of-attraction (RoA) based approaches (Tedrake et al., 2010; Bhounsule et al., 2018; Zamani et al., 2019). However, these methods are often computational expensive for high dimension hybrid systems. There exist reinforcement learning (RL) methods to control hybrid systems like legged robots (Benbrahim and Franklin, 1997; Morimoto et al., 2004; Neunert et al., 2020; Mastrogeorgiou et al., 2020), but finding the appropriate RL methods and rewards are extremely challenging and may cause undesired behaviors learnt to hack for high returns (Clark and Amodei, 2016). Our method shares similar philosophy with RoA-based works (Tedrake et al., 2010; Bhounsule et al., 2018; Zamani et al., 2019), whereas ours is computation-efficient, suits general nonlinear hybrid systems and can represent RoAs for arbitrary number of modes.

3. Problem Formulation

Definition 1 (Controlled Hybrid Systems) *A controlled hybrid system is defined as:*

$$\begin{cases} \dot{x} = f_i(x, u; p_i), & x \in \mathcal{C}_i \\ x^+ = h_i(x, u; p_i, p_j), & x \in \mathcal{D}_{i,j} \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the system state, $u \in \mathbb{R}^m$ denotes the control input, and $p_i \in \mathcal{P}$ denotes the system configuration (e.g. the set point, reference velocity). The index $i = 1, ..., I$ denotes the system mode. $\mathcal{C}_i$ is the flow set where states follow continuous flow map $f_i : \mathbb{R}^n \times \mathbb{R}^m \times \mathcal{P} \to \mathbb{R}^n$, and $\mathcal{D}_{i,j}$ is the jump set where states follow discrete jump map $h_i : \mathbb{R}^n \times \mathbb{R}^m \times \mathcal{P} \times \mathcal{P} \to \mathbb{R}^n$.

We aim to stabilize the hybrid system in Def. 1. For each mode of system $\dot{x} = f_i(x, u; p_i)$ with equilibrium x_i^*, we consider the stability defined in (Khalil, 2009). Then the sufficient conditions for each mode system stability are as follows (we omit the index i and the configuration p for brevity).

Proposition 2 (Control Lyapunov Conditions for System Stability) *The system $\dot{x} = f(x, u)$ with equilibrium x^* is asymptotically stable at x^* if there exist a differentiable function $V : \mathcal{C} \to \mathbb{R}$ and a control policy $u = \pi(x)$ such that: $V(x^*) = 0$; and $\forall x \in \mathcal{C} \backslash \{x^*\}$, $V(x) > 0$, and $\frac{dV}{dx} f(x, u) < 0$. The V is called a control Lyapunov function (CLF). The system is exponentially stable at x^* if the V further satisfies: $\exists \gamma > 0$, $\forall x \in \mathcal{C} \backslash \{x^*\}$, $\frac{dV}{dx} f(x, u) < -\gamma V$.*

The proof can be found in (Isidori, 1985)[Theorem 9.4.1]. If each system mode is stable with valid Lyapunov functions, will the hybrid system converge? Unfortunately, the answer is no with a counter-example in (Branicky, 1998). The culprit is that the system switches too fast: although the Lyapunov value is decreasing in each mode, the distance toward equilibrium is not decreasing yet. For systems with jumps, it is more unlikely to ensure the stability, as there might be jumps making $||x(t) - x^*|| \geq \epsilon$ infinitely often. Thus, we propose a new stability for the hybrid systems.

Definition 3 (Hybrid System Stability) *Given $\epsilon = \{\epsilon_i\}$, a hybrid system is ϵ-asymptotically stable (exponentially stable), if each system mode is asymptotically stable (exponentially stable), and all states $\bar{x}_i$ exiting the mode i are within ϵ_i-ball of the x^*, i.e., $||\bar{x}_i - x^*|| \leq \epsilon_i$. We call $\bar{x}_i$ as exiting state (and call state x_i that enters the mode i as entering state).*

If ϵ is constant, the exiting states will be at most ϵ-far away from the equilibrium. If ϵ converges to zero, the sequence of the exiting states will converge to the equilibrium hence asymptotic stability is achieved. In this paper, we consider the former case. With constant ϵ, we define ϵ-RoA as follows.

Definition 4 (ϵ-Region of Attraction) *The ϵ-RoA for x^* in mode i is defined as: $\mathcal{R}^\epsilon = \{x_0 |, x(0) = x_0, ||\bar{x}_i - x^*|| \leq \epsilon\}$, where $\bar{x}_i$ is the exiting state (for mode i) starting at x_0.*

Using ϵ-RoA, we do not need to check dwell time conditions like (Hespanha and Morse, 1999). To efficiently check if $x \in \epsilon$-RoA, we seek a set representation for ϵ-RoA and a scalar function/index to tell whether x is in the set. We use Lyapunov level set to (conservatively) represent ϵ-RoA.

Definition 5 (Maximum ϵ-Stable Level Set) *For mode i and configuration p_i, the largest level set $\mathcal{S}^{c_i} = \{x | V_i(x, p_i) \leq c_i(p_i)\}$ within ϵ-RoA is called Maximum ϵ-Stable Level Set, and $c_i(p_i)$ is called Maximum ϵ-Stable Level Set index.*

For an entering state x_j, we have $V_j(x_j, p_j) \leq c_j(p_j) \to x_j \in \epsilon$-RoA. One step ahead, at mode i with switching $i \to j$, it requires $V_j(h_i(\bar{x}_i, u; p_i, p_j)) \leq c_j(p_j)$. Propagating from $\bar{x}_i$ to x^*, we derive sufficient conditions for hybrid system stability proved in [1](Meng and Fan, 2022)[Appx. A].

Theorem 6 (Lyapunov Conditions for Hybrid System ϵ-Stability) *A hybrid system in Def. 1 is ϵ-exponentially stable, if there exists a Lyapunov function V_i for each mode i (and configuration p_i) satisfying all conditions in Prop. 2 and $\alpha||x - x^*|| \leq V_i(x, p_i) \leq \beta||x - x^*||$, and for each entering state x_i that moves toward the mode j, we have the p_i, c_i, p_j, c_j to satisfy:*

$$V_i(x_i, p_i) \leq c_i(p_i) \text{ and } V_j(h_i(x^*, u; p_i, p_j), p_j) \leq \Upsilon \tag{2}$$

where $\Upsilon = \frac{\alpha_j}{\beta_j} c_j(p_j) - \alpha_j K_i \epsilon$, and c_i, c_j are the maximum ϵ-stable level set indices for modes i and j, and K_i is the local Lipschitz constant for function h_i within ϵ-ball of x^*.

In short, Theo. 6 guarantees each system mode is exponentially stable, and at switching $i \to j$, the entering state x_i is within ϵ-RoA for mode i, and the next entering state x_j is also inside ϵ-RoA for mode j. The Lipschitz constant K_i can be approximated by $|\frac{\partial h_i}{\partial x}|$ at $(x^*, u; p_i, p_j)$ for small ϵ. In the next section, we will use neural networks to satisfy Theo. 6 for hybrid system stability.

4. Methodology

To control the system in Def. 1, we propose a learning framework in Algor. 1. Guided by Theo. 6, we first learn the controller and the CLF to stabilize the system within each mode. Then we perform the RoA estimation, and finally, we bring in the differentiable planner that leverages the controller and RoA estimator to enforce the hybrid system stability.

4.1. Learning neural Lyapunov functions and controllers

We train neural networks (NN) to synthesize the control Lyapunov functions $V(x, p)$ and controllers $u = \pi(x, p)$ for each system mode. During training, we use the controller to roll out trajectories from uniformly sampled initial states and compute the CLF values. To satisfy the CLF conditions in Theo. 6, we design the Lyapunov function as $V(x, p) = ||P_{\text{NN}}(p)(x - x^*)|| + V_{\text{NN}}(x, p)^2 ||x - x^*||^2$, where $P_{\text{NN}}(p)$ is an NN that takes p as input and outputs a matrix, and $V_{\text{NN}}(x, p)$ is an NN taking x, p as inputs and outputs a scalar. In this way, the positive definiteness and the norm inequalities of the Lyapunov functions are naturally satisfied. Finally, to enforce the Lyapunov value decreasing (exponentially) along the trajectories $\mathcal{S}$, we design the loss:

$$\mathcal{L}_{clf} = \sum_{x \in \mathcal{S}, p \in \mathcal{P}} \text{ReLU}\Big(\gamma V(x, p) + \frac{\partial V(x, p)}{\partial x} f(x, u)\Big) \tag{3}$$

where $u = \pi(x, p)$ and we compute the derivative term $\frac{\partial V(x_t, p)}{\partial x_t} f(x_t, u) \approx (V_{t+1} - V_t)/dt$.

1. The appendix is at https://mit-realm.github.io/hybrid-clf/static/appendix.pdf

4.2. Learning neural RoA estimator

After training for the CLF and controllers, we estimate the maximum ϵ-stable level set in Def. 5 for any given p using: $c^*(p) = \max \left\{ c \,\middle|\, \forall x_0 \in \mathcal{C}_i,\ V(x_0, p) \leq c \rightarrow x_0 \in \mathcal{R}^\epsilon \right\}$, where $\mathcal{R}^\epsilon$ is the ϵ-RoA in Def. 4. For each p_i, we set $\epsilon = 10^{-2}$ and uniformly sample $10^3 \sim 10^4$ initial states, roll out trajectories and find the largest Lyapunov value c_i^* for all the initial states that have exiting states within the ϵ-ball of the equilibrium. We train the NN RoA Estimator $R_{\text{NN}}(p)$ with:

$$\mathcal{L}_{roa} = \sum_{(p_i, c_i^*) \sim \mathcal{Z}} \Big(R_{\text{NN}}(p_i) - c_i^* \Big)^2 \tag{4}$$

where $\mathcal{Z}$ is the set for simulated trajectories and maximum ϵ-Stable level set indices. In this way, the RoA under configuration p can be approximated by $\big\{ x \mid V(x, p) \leq R_{\text{NN}}(p) \big\}$.

4.3. Differentiable configuration planner

We use a differentiable planner to ensure switching stability by satisfying the last condition in Eq. 2 of Theo. 6. Assume p_j is given at mode i. We use gradient descent to find p_i to minimize:

$$\mathcal{L}_e = \text{ReLU}\Big(V_i(x_i, p_i) - R_{\text{NN}}(p_i) \Big) + \text{ReLU}\Big(V_j(h_i(x^*, u, p_i), p_j) - \eta R_{\text{NN}}(p_j) + \kappa \Big) \tag{5}$$

where $\eta = \frac{\alpha_j}{\beta_j}$ and $\kappa = \alpha_j K_i \epsilon$. It is time-consuming to evaluate α_j, β_j and K_i for each p_i and p_j. Empirically, we set η=0.9 and κ=10^{-2} for ϵ=10^{-2} (ablation in (Meng and Fan, 2022)[Appx. F]). The optimization converges after 3∼10 steps, adding only little runtime overhead. We can also use a new loss $\mathcal{L}_e^+$ for periodic systems with decomposable jump map (common in error-state dynamics):

$$h_i(x, u; p_i, p_j) = h^+(x, p_i) + p_i - p_j,\ h^+(x^*, p_i) = x^* \tag{6}$$

where the heuristic is to guide the configuration gradually approaches the target configuration p_j:

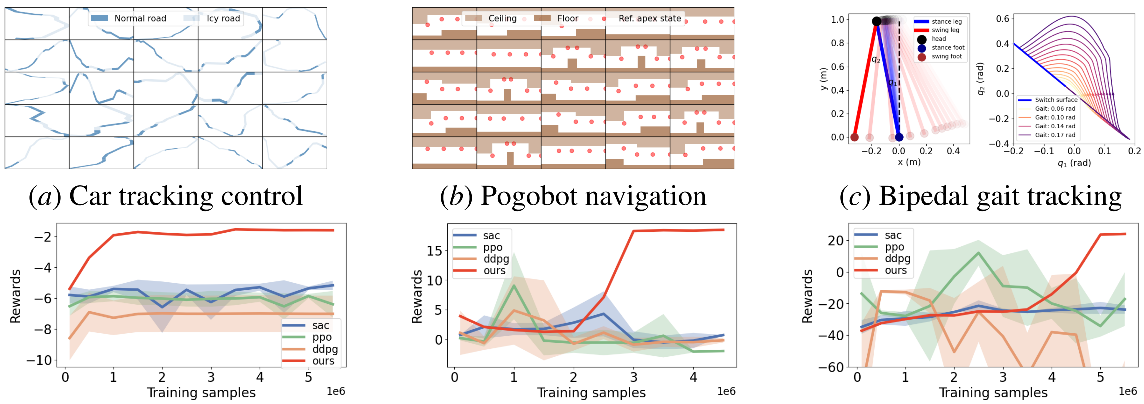
$$\mathcal{L}_e^+ = \text{ReLU}\Big(V_i(x_i, p_i) - R_{\text{NN}}(p_i) \Big) + \lambda ||p_j - p_i|| \tag{7}$$

We guarantee it converges to the target p_j in finite steps (proved in (Meng and Fan, 2022)[Appx. B]):

Theorem 7 (Finite-step converging toward the target configuration) *For special hybrid systems with h_i defined in Eq. 6. If $c_m(p_m) \geq \beta_m K_m \epsilon$ for all modes m and p_m, and assume the first term in Eq. 7 is zero for the optimal p, then the system is ϵ-stable and the configuration is guaranteed to reach the target configuration p_j in finite switches $N \leq \left\lceil \frac{||p_j - p_i||}{\min\limits_m \frac{c_m}{\beta_m} - K_m \epsilon} \right\rceil$.*

Algorithm 1 Hybrid System Control Algorithm

Sample $\{f_i, p_i\}_{i=1}^N$ from the hybrid system in Def. 1 ; `// Training phase`
foreach f_i, p_i **do**
  Train V_i and π_i with the loss $\mathcal{L}_{clf}$ in Eq. 3
  Find c_i^* for V_i and π_i (according to Sec. 4.2)
Train R_{NN} on $\{(p_i, c_i^*)\}_{i=1}^N$ with the loss $\mathcal{L}_{roa}$ in Eq. 4
Initialize state $x(0) \leftarrow x_0$, mode k and time $t \leftarrow 0$; `// Testing phase`
while $t \leq T$ **do**
  if *right before entering mode i (with next mode j)* **then**
    Optimize p_i with the loss $\mathcal{L}_e$ ($\mathcal{L}_e^+$) in Eq. 5 (Eq. 7)
    $x(t) \leftarrow h_k(x(t), u; p_k, p_i),\ \ k \leftarrow i$
  else
    $x(t + \Delta t) \leftarrow x(t) + f_k(x(\tau), \pi_k(x(\tau), p_k), p_k)\Delta t, \quad t \leftarrow t + \Delta t$



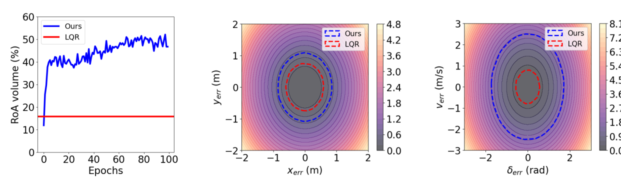
(*a*) Car tracking control (*b*) Pogobot navigation (*c*) Bipedal gait tracking

(*d*) Car control reward (*e*) Pogobot reward (*f*) Bipedal walker reward

Figure 2: Simulation environment visualization and reward comparisons under varied training sizes: (a) car control on icy roads (25 maps) (b) pogobot navigation in 2D mazes (25 maps) (c) Bipedal walker control. The RL reward is averaged across 3 random seeds.

5. Experiments

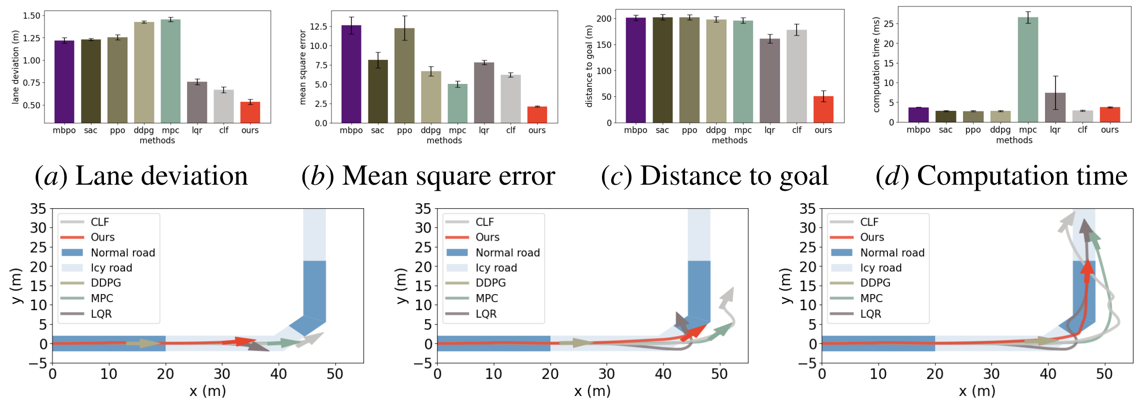
We conduct three challenging experiments shown in Fig. 2. Our method achieves the best performance (success rate, RMSE, etc) than most baselines and is 10∼50X faster than MPC. Our approach also results in shorter training time (0.5X of RL methods and 0.1X of HJB approaches). **Baselines:** For all cases we compare with: model-base RL (MBPO (Janner et al., 2019)), model-free RL(SAC (Haarnoja et al., 2018), PPO (Schulman et al., 2017) and DDPG (Lillicrap et al., 2015)) and model predictive control (MPC). Besides, for the car case, we compare with Linear Quadratic Regulator (LQR) and single CLF (Chang et al., 2019). For the bipedal we compare with quadratic program (QP) and Hamilton-Jacobian (HJB) (Choi et al., 2022). We did not compare with HJB for (9-dim) car or (8-dim) pogobot as the state dimension is too high for HJB to handle. **Implementation details:** For CLF, the controller and the RoA estimator, we use 2-layer NNs with 256 units in each layer and ReLU (Nair and Hinton, 2010) in intermediate layers. The controller uses TanH (LeCun et al., 2012) in the last layer to clip the output in a reasonable range. We implement our method in PyTorch (Paszke et al., 2019). The training takes 3∼6 hours on an RTX2080 Ti GPU. More details are in (Meng and Fan, 2022)[Appx. E].



(*a*) RoA (*b*) x-y plane (*c*) v-δ plane

Figure 3: RoA comparison with LQR.

Remarks on sample efficiency: As shown in Fig. 2, compared to RL under the same sample size, we achieve the highest rewards. This is because RL directly interacts with the hybrid systems, whereas ours learns to control the system under each mode, which is easier.



(*a*) Lane deviation (*b*) Mean square error (*c*) Distance to goal (*d*) Computation time

(*e*) T=2.5s (*f*) T=3.8s (*g*) T=5.8s

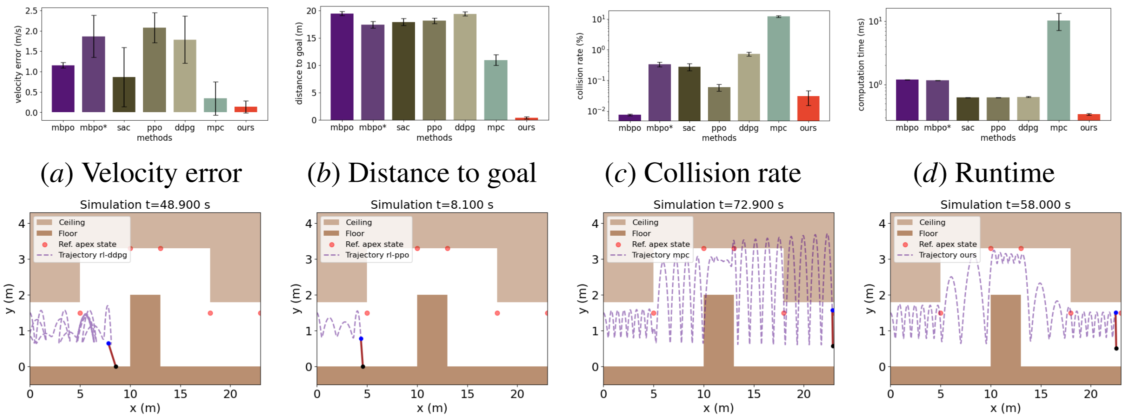
Figure 4: Quantitative ((a)∼(d)) and qualitative ((e)∼(g)) comparison for car control experiment. At T=2.5s, our approach learns to decelerate and turn left a bit to prepare for the incoming turn at icy road. At T=3.8s, our method is on the road, whereas other baselines diverge.

5.1. Car control under different road conditions

We consider the tracking control problem for the single-track model in (Althoff et al., 2017) under varied frictions. The system is hybrid as varied frictions and tracking velocities lead to different dynamics. The challenge is that when the friction is low (e.g., on icy roads), the road cannot provide enough traction to keep the car on the lane. To stabilize the system, the controller needs to track proper configurations (speeds and waypoints). Our method ensures the entering state at the connection of segments is always in the RoA of the next segment. **Setups:** We generate 25 maps with 10 randomly sampled segments. Each segment has friction $\in \{0.1, 1\}$ and length $\in [7.5\text{m}, 37.5\text{m}]$. The angle between consecutive segments $\in [\frac{3\pi}{4}, \pi]$. **Metrics:** We measure average lane deviation, mean square error (MSE) to the reference trajectory, distance to the goal (before driving out of lane), and the run time per step.

As shown in Fig. 3(a), the RoA of our approach surpasses the LQR method in just a few epochs and converges in 100 epochs, becoming nearly 2X larger than the LQR RoA. From the RoA visualization in Fig. 3(b, c), we show that the gain is mainly from velocity error, tire angle error, longitudinal and lateral errors. This shows our method can enlarge the RoA to stabilize more states. The rest visualizations can be found in (Meng and Fan, 2022)[Appx. I].

Next, we compare with other baselines. As shown in Fig. 4, we achieve the lowest lane deviation and MSE, 67%∼75% reduction in the distance to goal, and low run time on par of RL, which is less than 0.1X of the time used by MPC. Qualitatively, as shown in Fig. 4 (here we omit MBPO, SAC and PPO because they cannot provide reasonable trajectories), our approach is the only method that can keep the car on the road. The reason is that our method learns to decelerate and turn left to prepare for the next icy road segment (Fig. 4(a)), so that the car can gain more traction on the icy road for a normal left turn (Fig. 4(b)) where other methods fail to keep the car on the road (Fig. 4(c)).



(*a*) Velocity error (*b*) Distance to goal (*c*) Collision rate (*d*) Runtime

(*e*) RL method (DDPG) (*f*) RL method (PPO) (*g*) MPC method (*h*) Our approach

Figure 5: Quantitative ((a)∼(d)) and qualitative ((e)∼(h)) results for the pogobot. Our method can safely jump through the maze. DDPG/PPO jump to the left and MPC results in collisions.

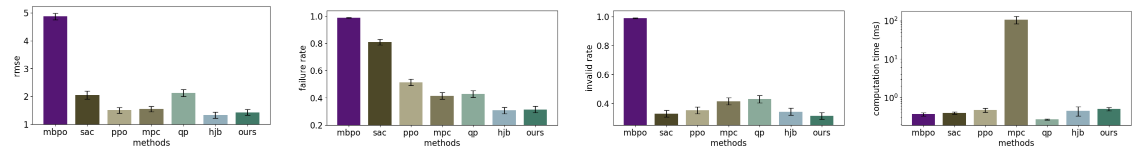
5.2. Pogobot navigation

We control a pogobot (with the model in (Zamani et al., 2019)) to jump through 2d mazes shown in Fig. 5. The pogobot alternates between flight and stance phases. The apex state is at the top of a trajectory in the flight phase. Given the reference apex states, the goal is to jump through the maze safely. Our method first learns the unknown apex state dynamic then plans configurations (reference apex state) to ensure apex states are within the RoA of the next reference apex state. **Setups:** We generate 25 maps by randomly sampling 3∼5 segments with a segment length $\in [3m, 6m]$, a floor height $\in [-0.5m, 2m]$, a ceiling height $\in [1.5m, 3.5m]$, and a reference speed $\in [0.5m/s, 1.5m/s]$. **Metrics:** We measure velocity error, distance to goal, collision rate, and computation runtime.

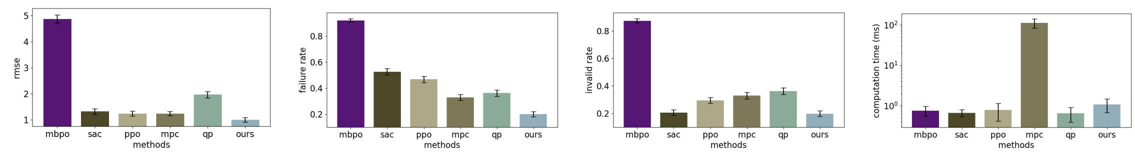
Since we learn dynamics here, we also compare with MBPO with pretrained dynamic model in our method (denoted as MBPO*). As shown in Fig. 5, we achieve the lowest velocity error, distance to goal, low collision rate and runtime (MBPO has the lowest collision rate as it often jumps out of valid region before crashing - which explains its high distance-to-goal metric). We use just 1/70X of the computation time as needed for MPC. Fig. 5 (e)-(h) show that our method can safely jump through the 2d maze, whereas DDPG and PPO jump to the left (we omit SAC as it cannot plan for one cycle), and MPC results in collisions. More details are in (Meng and Fan, 2022)[Appx. C∼D].

5.3. Bipedal walker locomotion

We control the bipedal robot (Choi et al., 2022) to converge to a target gait. The configuration is the leg angle q_1 of the next gait at the switch surface shown in Fig. 2(c). It is hard to converge directly to the goal gait when the initial gait is far. Thus, we use Eq. 7 in planning to ensure the gaits are gradually closer to the target. **Setups:** We calculate the gaits using Frost Library (Hereid and Ames, 2017), uniformly sample the initial states around each gait with angle $\in [0.04, 0.18]$ and target gait $\in [0.04, 0.18]$. When compared to HJB, we set the target gait=0.13rad since HJB only



(*a*) RMSE (*b*) Failure rate (*c*) Invalid rate (*d*) Computation time



(*e*) RMSE (*f*) Failure rate (*g*) Invalid rate (*h*) Computation time

Figure 6: Bipedal comparison with same ((a)∼(d)) and different target gaits ((e)∼(h)).We achieve close results to HJB on the same gait, and much better than others on different gaits setup.

handles that gait. We compare other baselines with different target gaits. **Metrics:** We measure RMSE towards reference gaits, failure rate for convergence, invalid trajectories rate and run time.

As shown in Fig. 6(a)-(d), we achieve close RMSE, failure rate, and invalid trajectory rate to HJB. The third best method for failure rate is MPC, which takes a much longer computation time. MBPO has the worst performance as the dynamics might be too complicated to be learnt. Compared to HJB, our advantage is quickly adapting to other targeted gaits. Trained in less than 6 hours, ours can also learn to converge to other target gaits, whereas the learning time for the HJB method to converge to one target gait is 36 hours. We compare with other non-HJB approaches for different gaits in Fig. 6(e)-(h), and we achieve the lowest RMSE, failure and invalid rate and low runtime.

5.4. Limitations

Firstly, the RoA estimation is after the controller training, thus an RoA refinement is needed if the controller is updated. Besides, the certificates are NNs learned from samples thus we cannot guarantee they hold in the whole state space. Also we might bring in errors when numerically approximate the continuous dynamic, though we show in (Meng and Fan, 2022)[Appx. F] that the performance is consistent across varied Δt. Lastly, we assume the system state is known without noises, which might not hold in real-world experiments. We aim to solve these in future works.

6. Conclusion

We propose a learning-based hierarchical approach to stabilize nonlinear hybrid systems. The learned low-level controller can stabilize the system states within each mode. Upon switching, the high-level planner finds the optimal configuration that guarantees the next entering state is within the RoA of the next mode. Experimental results show that our approach achieves the best overall performance on multiple hybrid systems compared to other approaches. For future works, we aim to tackle some of the limitations discussed in Sec. 5.4, and extend this framework to further control hybrid systems with perception modules and the presence of estimation error and disturbances.

Acknowledgement

The Defense Science and Technology Agency in Singapore and the C3.ai Digital Transformation Institute provided funds to assist the authors with their research. However, this article solely reflects the opinions and conclusions of its authors and not DSTA Singapore, the Singapore Government, or C3.ai Digital Transformation Institute.

References

Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2020.

Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726. IEEE, 2017.

Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

Zvi Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods & Applications*, 7(11):1163–1173, 1983.

Hamid Benbrahim and Judy A Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22(3-4):283–302, 1997.

Pranav A Bhounsule, Ali Zamani, and Jason Pusey. Switching between limit cycles in a model of running using exponentially stabilizing discrete control lyapunov function. In *2018 Annual American Control Conference (ACC)*, pages 3714–3719. IEEE, 2018.

Michael S Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on automatic control*, 43(4):475–482, 1998.

Eduardo F Camacho, Daniel R Ramírez, Daniel Limón, D Muñoz De La Peña, and Teodoro Alamo. Model predictive control techniques for hybrid systems. *Annual reviews in control*, 34(1):21–31, 2010.

Christos G Cassandras, David L Pepyne, and Yorai Wardi. Optimal control of a class of hybrid systems. *IEEE Transactions on Automatic Control*, 46(3):398–415, 2001.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

Shaoru Chen, Mahyar Fazlyab, Manfred Morari, George J Pappas, and Victor M Preciado. Learning lyapunov functions for hybrid systems. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2021.

Young C Cho, Christos G Cassandras, and David L Pepyne. Forward decomposition algorithms for optimal control of a class of hybrid systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 11(5):497–513, 2001.

Jason J Choi, Ayush Agrawal, Koushil Sreenath, Claire J Tomlin, and Somil Bansal. Computation of regions of attraction for hybrid limit cycles using reachability: An application to walking robots. *arXiv preprint arXiv:2201.08538*, 2022.

Jack Clark and Dario Amodei. Faulty reward functions in the wild. *Internet: https://blog. openai. com/faulty-reward-functions*, 2016.

William Clark, Anthony Bloch, and Leonardo Colombo. Poincar\'e-bendixson theorem for hybrid systems. *arXiv preprint arXiv:1801.09014*, 2018.

G Davrazos and NT Koussoulas. A review of stability results for switched and hybrid systems. In *Mediterranean Conference on Control and Automation*. Citeseer, 2001.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods. *arXiv preprint arXiv:2202.11762*, 2022a.

Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, pages 1724–1735. PMLR, 2022b.

Bart De Schutter, WPMH Heemels, Jan Lunze, Christophe Prieur, et al. Survey of modeling, analysis, and control of hybrid systems. *Handbook of Hybrid Systems Control–Theory, Tools, Applications*, pages 31–55, 2009.

Raymond A DeCarlo, Michael S Branicky, Stefan Pettersson, and Bengt Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88(7): 1069–1082, 2000.

Murat Dogruel and Umit Ozguner. Stability of hybrid systems. In *Proceedings of 1994 9th IEEE International Symposium on Intelligent Control*, pages 129–134. IEEE, 1994.

Rafael Olmedo Fierro. *A hybrid system approach to a class of intelligent control systems*. The University of Texas at Arlington, 1997.

Jeremy H Gillula, Gabriel M Hoffmann, Haomiao Huang, Michael P Vitus, and Claire J Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *The International Journal of Robotics Research*, 30(3):335–354, 2011.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Ayonga Hereid and Aaron D Ames. Frost*: Fast robot optimization and simulation toolkit. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 719–726. IEEE, 2017.

Joao P Hespanha and A Stephen Morse. Stability of switched systems with average dwell-time. In *Proceedings of the 38th IEEE conference on decision and control (Cat. No. 99CH36304)*, volume 3, pages 2655–2660. IEEE, 1999.

Alberto Isidori. *Nonlinear control systems: an introduction*. Springer, 1985.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard. Some controls applications of sum of squares programming. In *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*, volume 5, pages 4676–4681. IEEE, 2003.

Mikael Johansson. *Piecewise linear control systems*. PhD thesis, Ph. D. Thesis, Lund Institute of Technology, Sweden, 1999.

Mikael K-J Johansson. *Piecewise linear control systems: a computational approach*, volume 284. Springer, 2003.

Hassan K Khalil. Lyapunov stability. *Control Systems, Robotics and AutomatioN*, 12:115, 2009.

Mircea Lazar. Model predictive control of hybrid systems: Stability and robustness, 2006.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. In *2013 IEEE International Conference on Robotics and Automation*, pages 4054–4061. IEEE, 2013.

Jörgen Malmborg. Analysis and design of hybrid control systems. *Department of Automatic Control, Lund Institute of Technology*, 1998.

Ian R Manchester. Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. *IFAC Proceedings Volumes*, 44(1):6285–6290, 2011.

Ian R Manchester, Mark M Tobenkin, Michael Levashov, and Russ Tedrake. Regions of attraction for hybrid limit cycles of walking robots. *IFAC Proceedings Volumes*, 44(1):5801–5806, 2011.

Tobia Marcucci and Russ Tedrake. Warm start of mixed-integer programs for model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, 66(6):2433–2448, 2020.

Athanasios S Mastrogeorgiou, Yehia S Elbahrawy, Andrés Kecskeméthy, and Evangelos G Papadopoulos. Slope handling for quadruped robots using deep reinforcement learning and toe trajectory planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3777–3782. IEEE, 2020.

Arash Mehrjou, Mohammad Ghavamzadeh, and Bernhard Schölkopf. Neural lyapunov redesign. *arXiv preprint arXiv:2006.03947*, 2020.

Yue Meng and Chuchu Fan. Supplementary material for l4dc2023 submission: "hybrid systems neural control with region-of-attraction planner". https://mit-realm.github.io/hybrid-clf/static/appendix.pdf, 2022.

Yue Meng, Zengyi Qin, and Chuchu Fan. Reactive and safe road user simulations using neural barrier certificates. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6299–6306. IEEE, 2021.

Anthony N Michel and Bo Hu. Towards a stability theory of general hybrid dynamical systems. *Automatica*, 35(3):371–384, 1999.

Domenico Mignone, Giancarlo Ferrari-Trecate, and Manfred Morari. Stability and stabilization of piecewise affine and hybrid systems: An lmi approach. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, volume 1, pages 504–509. IEEE, 2000.

Jun Morimoto, Gordon Cheng, Christopher G Atkeson, and Garth Zeglin. A simple reinforcement learning algorithm for biped walking. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 3030–3035. IEEE, 2004.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

Michael Neunert, Abbas Abdolmaleki, Markus Wulfmeier, Thomas Lampe, Tobias Springenberg, Roland Hafner, Francesco Romano, Jonas Buchli, Nicolas Heess, and Martin Riedmiller. Continuous-discrete reinforcement learning for hybrid control in robotics. In *Conference on Robot Learning*, pages 735–751. PMLR, 2020.

Huansheng Ning, Rui Yin, Ata Ullah, and Feifei Shi. A survey on hybrid human-artificial intelligence for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

P Peleties and Raymond Decarlo. Asymptotic stability of m-switched systems using lyapunov functions. In *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, pages 3438–3439. IEEE, 1992.

Stefan Pettersson and Bengt Lennartson. Stability and robustness for hybrid systems. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 2, pages 1202–1207. IEEE, 1996.

Stefan Pettersson and Bengt Lennartson. Exponential stability of hybrid systems using piecewise quadratic lyapunov functions resulting in lmis. *IFAC Proceedings Volumes*, 32(2):4810–4815, 1999.

Luis Pineda, Brandon Amos, Amy Zhang, Nathan O Lambert, and Roberto Calandra. Mbrl-lib: A modular library for model-based reinforcement learning. *arXiv preprint arXiv:2104.10159*, 2021.

Henri Poincaré. Sur les courbes définies par les équations différentielles. *J. Math. Pures Appl.*, 4: 167–244, 1885.

Stephen Prajna and Antonis Papachristodoulou. Analysis of switched and hybrid systems-beyond piecewise quadratic methods. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 4, pages 2779–2784. IEEE, 2003.

Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436*, 2021.

Spencer M Richards, Felix Berkenkamp, and Andreas Krause. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, pages 466–476. PMLR, 2018.

Ricardo G Sanfelice. *Control of hybrid systems: An overview of recent advances*. UC Santa Cruz Previously Published Works, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Aman Sinha, Matthew O'Kelly, Russ Tedrake, and John C Duchi. Neural bridge sampling for evaluating safety-critical autonomous systems. *Advances in Neural Information Processing Systems*, 33:6402–6416, 2020.

Olav Slupphaug and Bjarne A Foss. Model predictive control for a class of hybrid systems. In *1997 European Control Conference (ECC)*, pages 3095–3100. IEEE, 1997.

Eduardo D Sontag. A lyapunov-like characterization of asymptotic controllability. *SIAM journal on control and optimization*, 21(3):462–471, 1983.

Dawei Sun, Susmit Jha, and Chuchu Fan. Learning certified control using contraction metric. *arXiv preprint arXiv:2011.12569*, 2020.

Zonghai Sun, Liangzhi Gan, and Youxian Sun. Support vector machine adaptive control of nonlinear systems. In *International Conference on Intelligent Computing*, pages 159–168. Springer, 2005.

Russ Tedrake, Ian R Manchester, Mark Tobenkin, and John W Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29 (8):1038–1052, 2010.

Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008.

Mark Wicks and Raymond DeCarlo. Solution of coupled lyapunov equations for the stabilization of multimodal linear systems. In *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, volume 3, pages 1709–1713. IEEE, 1997.

Wenxin Xiao, Armin Lederer, and Sandra Hirche. Learning stable nonparametric dynamical systems with gaussian process regression. *IFAC-PapersOnLine*, 53(2):1194–1199, 2020.

Ali Zamani, Joseph D Galloway, and Pranav A Bhounsule. Feedback motion planning of legged robots by composing orbital lyapunov functions using rapidly-exploring random trees. In *2019 international conference on robotics and automation (ICRA)*, pages 1410–1416. IEEE, 2019.

Chao Zhai and Hung D Nguyen. Region of attraction for power systems using gaussian process and converse lyapunov function–part i: Theoretical framework and off-line study. *arXiv preprint arXiv:1906.03590*, 2019.

Guisheng Zhai, Bo Hu, Kazunori Yasuda, and Anthony N Michel. Piecewise lyapunov functions for switched systems with average dwell time. *Asian Journal of Control*, 2(3):192–197, 2000.