Learning-based Lyapunov Analysis for Nonlinear Control Systems

Ya-Chien Chang, Lijing Kuang, Yuet Fung

Abstract

The establishment of Lyapunov theory has provided fundamental methods to determine local stability using Lyapunov functions. How to successfully search Lyapunov functions therefore becomes a critical problem in stabilizing control systems. In this work, we tackle the stability problem in nonlinear control systems using a novel learning-based architecture, which guarantees the satisfaction of Lyapunov constraints in a strict sense. Experiments on various dynamical systems are performed to evaluate the effectiveness of the presented method.

1 Introduction

Evaluating the regions of stability is one of the most fundamental problems in cyber-physical systems and safety-critical systems [1,2], where continuous and unknown system dynamics are often involved. When dealing with continuous dynamics, the common practice is to discretize the space using dynamic programming. This approach, however, struggles to handle discontinuous dynamics and complex domains effectively [3]. Recently, we have seen some promising progresses in the control community to utilize convex optimization for the development of efficient algorithms that are able to search and compute Lyapunov functions in a direct way, so as to stabilize nonlinear control systems. The presence of nonlinearity requires function approximation, where existing algorithms typically refer to sum-of-squares polynomials as Lyapunov functions using semidefinite programming (SDP). It has been shown that SDP facilitates the determination of stability in nonlinear control systems [4,5], which can then be combined with motion planning algorithms for feedback control synthesis in complex dynamical systems [6]. Nevertheless, none of the existing methods provide provable guarantees of Lyapunov stability constraints, which may lead to infeasible solution sets as a result.

To tackle this issue, we take one step forward to solve the stability problem in nonlinear dynamical systems by resorting to learning-based optimization methods using neural network architectures. The presented method is able to provide provable guarantees in constructing Lyapunov functions for the required system dynamics, which in turn allows us to establish regions of stability. To summarize, the main contributions of this work are as follows:

- We formulate the searching of safe regions for arbitrary nonlinear dynamical systems as a convex optimization problem, by dealing with a scalar function of states.
- A neural network learning framework is presented to construct Lyapunov function candidates for computing regions of stability, which are guaranteed to satisfy Lyapunov stability conditions. Our method excels the existing algorithms in providing guarantees of the satisfaction of Lyapunov conditions, which is a key challenge in the control community.
- Several nonlinear dynamical systems are introduced for experimental evaluations. Our presented method is able to find policies that generate larger region of attractions than existing methods.

The rest of this paper is as follows. In Section 2, we discuss the prior works and highlight our contributions. We describe the problem statement in Section 3 under which we introduce the primal problem, dual problem and KKT conditions. We develop a learning-based architecture using neural
networks to discover Lyapunov functions in Section 4. Section 5 presents our experiment results. Task assignment is in Section 6. We conclude our study in Section 7.

2 Related work

Richards et al. [7] have proposed a Lyapunov framework using neural networks to learn safety certificates in his recent work and he has shown that using neural networks is more effective. However, the goal and the approach proposed in this study are different from that of the prior work. Richards et al. focus on discrete-time polynomial systems whereas we focus on learning the control and the Lyapunov function together with a provable guarantee of stability in larger regions. In terms of approaches, Richard et al. use neural networks to learn the region of an attraction of a given controller while our method is capable of handling non-polynomial continuous dynamical systems with only the initialization of control functions. Besides, our architecture uses generic feed-forward network representations with no manual design, while the neural architecture in [7] required special design practices to accommodate the relaxation. As a result, our architecture applies to more nonlinear system and can find new control functions that enlarge provinces of attraction obtainable from standard control methods.

3 Problem Statement

Consider a dynamical system

$$\frac{dx}{dt} = f(x), \quad x(0) = x_0. \quad (1)$$

A point $x^*$ is a stable point of the above system if $f(x^*) = 0$. The equilibrium state $x^*$ is asymptotically stable if $x^*$ is locally stable and, as $t \to \infty$ all solutions starting near $x^*$ tend to $x^*$ (i.e. all trajectories converge to zero).

Meanwhile, consider a function $V$, $V$ is positive definite if

- $V(x) \geq 0 \quad \forall x$,
- $V(x) = 0 \iff x = 0$, and
- $V(x) \to \infty$ as $x \to \infty$.

Suppose $V(x)$ is the generalized energy function of a dynamical system, as the dynamical system approaches stability, the energy of it decreases with time. As a result, we have $-\nabla V(x) > 0$. Consider the Lyapunov global asymptotic stability theorem:

Suppose there is a function $V$ such that

- $V$ is positive definite,
- $\nabla V(x) < 0 \quad \forall x \neq 0, \nabla V(0) = 0$,

then every trajectory of $\frac{dx}{dt} = f(x)$ converges to zero as $t \to \infty$.

By changing the origin to $x_0 = 0$ and with the above facts, we can restate the stability problem as follows:

Seek a positive definite function $V : D \to \mathbb{R}$ that satisfies the following conditions:

$$V(0) = 0, \text{ and, } \forall x \in D \setminus \{0\}, V(x) > 0 \text{ and } \nabla V(x) < 0, \quad (2)$$

then the equilibrium is asymptotically stable and $V$ is called a Lyapunov function of the dynamical system.

3.1 Primal Problem

Consider the quadratic Lyapunov function candidate $V(x) = x^T Px$. We then have $\nabla V(x) = x^T (A^T P + PA)x$, where $P \in \mathbb{R}^{n \times n}$ is a positive definite matrix and $A \in \mathbb{R}^{n \times n}$. To satisfy above Lyapunov conditions we need $A^T P + PA \prec 0$. Thus, the primal problem is searching a positive definite matrix $P$ subject to $A^T P + PA \prec 0$. 

2
3.2 Dual Problem

Assume a candidate Lyapunov function $V_\theta$ is a multilayered feed-forward networks, then we formulate a cost function to measure the degree of violation of the Lyapunov conditions in (2). The learning process updates the parameters, $\theta$, to improve the likelihood of satisfying the Lyapunov conditions. Simultaneously, we want the safe regions as big as enough. Therefore, the cost function takes the form:

$$\sup_{x \in \mathcal{D}} \inf_{\theta} \left( \max(0, -V_\theta(x)) + \max(0, \nabla V_\theta(x)) + V_\theta^2(0) \right),$$

s.t. $x(i + 1) = f(x(i)), \ x(0) = x_0, \ x(i) \in \mathcal{D}, \forall i \in [0, N].$

3.3 KKT conditions

The corresponding first-order necessary conditions defined by KKT can be represented as follows: at a local optimum $x^*$, under the defined Lyapunov conditions, there exists Lagrange multipliers $\lambda^*_i$ such that:

$$\mathcal{L}(x, \theta, \lambda) = V_\theta(x) + \lambda (A^T P + PA),$$

$$\nabla_\theta \mathcal{L}(x^*, \theta^*, \lambda^*) = 0, \quad A^T P + PA < 0, \quad \lambda^* \geq 0, \quad \lambda (A^T P + PA) = 0.$$  

4 Methodologies

In this section, we introduce a learning-based architecture to seek Lyapunov functions using neural networks, which ensure the stability of control systems. Our learning-based architecture contains two neural-network components: an actor that learns candidate Lyapunov functions, and an evaluator that finds states that violate the Lyapunov conditions specified in (2). During the learning process, the actor network learns a parameterized Lyapunov function, by taking the input of vectorized system state and outputting a scalar value. The objective of the actor network measures the degree of violation of Lyapunov conditions, which takes the following form:

$$L(\theta) = \mathbb{E} \left[ \max(0, -V_\theta(x)) + \max(0, \nabla V_\theta(x)) + V_\theta^2(0) \right].$$

Accordingly, using Monte Carlo estimation, we are able to obtain the surrogate empirical loss function by drawing samples, which has the following form:

$$L_e(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[ \max(0, -V_\theta(x_i)) + \max(0, \nabla V_\theta(x_i)) \right] + V_\theta^2(0),$$

where $x_i$ are random samples drawn from the state vectors of the system in the i.i.d. manner. Throughout each iteration, the above objective function needs to be minimized using stochastic gradient descent (SGD).

While it is generally difficult for the existing algorithms to fully satisfy Lyapunov conditions, our method provides guarantees over all states of the system by incorporating an evaluator network. It takes the learned Lyapunov function from the actor network, and performs a global search to find out the state vectors that fail to meet the constraints. Such undesired states are appended to the training set for the subsequent training. Using delta-complete constraint solving introduced in [8], the algorithm assures when no occurrence of violation, Lyapunov conditions can be ensured to be satisfied across the examined domain. The learning process thus becomes an alternating learning process between the actor network and the evaluator network.

By adjusting the cost functions in the framework, it is possible to obtain additional desired properties for the learned Lyapunov functions, such as tuning region of attraction by adopting regularization in term of the speed that Lyapunov function value increases with respect to the radius of the level sets:

$$L(\theta) + \frac{1}{N} \sum_{i=1}^{N} \| x_i \| - \alpha V_\theta(x_i).$$

Compared to the gradient based methods which typically takes the entire training set for learning, the adoption of SGD in our algorithm allows us to achieve much better efficiency in each iteration and eventually, converge to the desired solution much faster.
To demonstrate the effectiveness of our method, we perform experimental evaluations on a nonlinear control problem. We demonstrate that our method successfully finds Lyapunov functions that fully satisfy the Lyapunov criteria. We also show that our framework outperforms the existing algorithms, in the sense to be able to generate significantly larger regions of attraction (ROA) compared to the results presented in [9].

Normalized pendulum. The standard pendulum system with normalized parameters is one of the most standard nonlinear problems. The two state variables in the system, \( x_1 \) and \( x_2 \), represent the angular position and angular velocity respectively. The system dynamics can be described as

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\sin(x_1) - x_2
\end{align*}
\]  

(3)

Our learning procedure finds the following neural Lyapunov function:

\[
V = \tanh(W_2 \tanh(W_1 x + B_1) + B_2),
\]

where \( x = [x_1, x_2]^T \) and

\[
W_1 = \begin{bmatrix}
0.3575 & -0.4428 & 0.2622 & -0.9552 & 0.3827 & 0.4264 \\
0.2368 & -0.2765 & 0.9584 & -0.5441 & -0.5767 & -1.4503
\end{bmatrix}^T,
\]

\[
W_2 = \begin{bmatrix}
-1.1849 & 1.3468 & -1.7934 & -1.6629 & -1.4462 & -0.1228
\end{bmatrix},
\]

\[
B_1 = \begin{bmatrix}
-1.4599 & 1.7899 & 1.3305 & 1.3885 & 1.4987 & -0.8466
\end{bmatrix} \text{ and } B_2 = \begin{bmatrix}
1.8862
\end{bmatrix}.
\]

![Figure 1: (left) Learned Lyapunov function for normalized pendulum. It shows any trajectory starts inside the regions of attraction (ROA) defined by the learned neural Lyapunov function will approach the equilibrium point as } t \to \infty \text{. (right) Comparison of ROA estimated from different Lyapunov functions. Our method enlarges the invariant set (red unit ball) from the previous method in [9] twice.}

6 Task Assignments

The write-ups (outline and report) are finished in teamwork. **Yuet Fung** studies the prior works and formulates the constrained primal and dual problem. **Lijing Kuang** develops a learning-based architecture to approximate Lyapunov functions. **Ya-Chien Chang** conducts experiments to demonstrate the effectiveness of the proposed framework.

7 Conclusions and Future Work

In this paper, we focus on the problem of stabilizing nonlinear control systems using convex optimization techniques. To address the fundamental issue in cyber-physical systems and to learn Lyapunov functions with provable guarantees, we introduce a neural network architecture, which can be used to...
compute regions of stability for achieving equilibrium in nonlinear dynamical systems. As for future work, it is of interest to incorporate the state-of-the-art reinforcement learning methods, such as trust region policy optimization (TRPO), soft actor critic (SAC), so as to search optimal control policies in continuous domains.

References


