2024 19th Annual System of Systems Engineering Conference (SoSE) | 979-8-3503-6591-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/SOSE62659.2024.10620953

Reinforcement Learning Integrated Nonlinear Controller for Guaranteed Stability

1st Shiqi Nan
Electrical and Computer Engineering Department
University of Texas at San Antonio
San Antonio, Texas
shiqi.nan@utsa.edu

2nd Chunjiang Qian
Electrical and Computer Engineering Department
University of Texas at San Antonio
San Antonio, Texas
chunjiang.qian@utsa.edu

***Abstract*—This paper considers the stabilization of a class of uncertain upper-triangular systems. Reinforcement learning has demonstrated the ability to handle complex control tasks for its model-free property and adaptability. However, the stability cannot be assured. To guarantee the stability of the uncertain system, a reinforcement learning integrated nonlinear controller is introduced. It shows that the proposed controllers provide the adaptability of reinforcement learning and guaranteed stability for the uncertain upper-triangular system.**

***Keywords*—reinforcement learning, nonlinear control, stability, upper-triangular systems**

I. Introduction

In control theory, stabilizing systems with uncertain parameters is a key challenge. This paper considers the problem of local asymptotic stabilization for uncertain upper-triangular systems (1).

$$\dot{x}_i = a_i x_{i+1} + f_i(x_{i+1}, \dots, x_n), i = 1, \dots, n-1,$$

$$\dot{x}_n = a_n u, \tag{1}$$

where $x = (x_1, \dots, x_n)^T \in \mathfrak{R}^n$ represents the system state vector, $u \in \mathfrak{R}$ denotes the control input, a_i, $i = 1, \dots, n-1$, are unknown positive constants and f_i , $i = 1, \dots, n-1$, are unknown functions.

This upper-triangular systems are used to model many practical systems like cart-pendulum system [1] and the ball and beam system [2]. The stabilization problem of upper-triangular systems has been extensively studied in existing research. Most results [3]-[8] are based on known a_i or with known bounds and known bounded high-order nonlinearities f_i. Stabilizing these systems can be challenging when parameters are unknown. For example, it's impossible to design a linear feedback controller to achieve local asymptotic stability when a_i are unknown. In addition, linear controller is sensitive to the upper-triangular system. By increasing the powers of the states, a nested controllers is designed in [9] to overcome the limitations of linear controllers. Although the proposed controller is not as sensitive as linear controller and guarantees the local asymptotic, global asymptotic stabilization is still a challenge.

Reinforcement learning (RL) has been successfully applied to control tasks in various domains, such as walkers[10], robotic manipulators[11], and aerial robots[12]. The utilization of Markov Decision Processes (MDPs) provides a structured framework for RL, enabling the modeling of sequential decision-making under uncertainty. MDPs are characterized by four main components: states, actions, transitions and rewards. During a series of time steps $(t = 0, 1, 2, 3, \dots)$, an agent repeatedly interacts with the environment E. At each time step t, the agent observes the state s_t of the environment, denoted as s_t which belongs to the set of possible states, and randomly chooses a real-valued action, $a_t \in \mathbb{R}^N$, from the set of available actions. After a time step, as a result of its action, the agent receives a numerical reward, r_{t+1}, which is a part of the set of real numbers, and transitions into a new state, s_{t+1}. The MDP and agent together thereby create a sequence that begins as follows:

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, \dots \tag{2}$$

Therefore, the decision rule is a state feedback control law, known as the policy π in RL, which determines the action based on the current state. This action may alter the state of the system, often in unpredictable ways, and the outcome of this change is measured by the reward function $r(s_t, a_t)$. The goal is to maximize the total expected reward over time from each starting state, a concept referred to as the value. RL is particularly appealing for solving control problems for system with uncertainty and disturbances due to its adaptability. In addition, RL is model-free and does not require a model of the system dynamics. While reinforcement learning offers a powerful framework for solving control problems in uncertain environments, it does not ensure stability in learning processes. Due to the dynamic nature of environments and the complexity of learning algorithms, convergence to optimal policies may not always be guaranteed.

In this paper, we propose a reinforcement learning integrated nonlinear controller for the upper-triangular system to guarantee local stability. The proposed controller combines the adaptability of reinforcement learning with the robustness of nonlinear control strategies to ensure local asymptotically stability of the upper-triangular systems.

This work was supported in part by the National Science Foundation under grant No. 2134237.

II. Related Work

A. Nonlinear Controller for Upper-triangular system

In [1], a nested nonlinear controller is proposed to overcome the limitation of linear controller to stabilize the upper-triangular system (1). For example, when $n = 3$, the nonlinear controller is designed as

$$u = -(\left(\left(k_1 x_1\right)^{5/3} + k_2 x_2\right)^{33/25} + k_3 x_3) \tag{3}$$

where k_is are any positive numbers.

The nested controller is derived by changing the linear structure of the controller and increasing the powers of the states. A fundamental principle in effectively increasing the powers of states of linear controllers is based on the theory of homogeneous systems [13][14]. Nevertheless, conventional homogeneity fails when addressing the unknown parameters a_i because of the similarity between linear systems and homogeneous systems.

The concept of homogeneity with monotone degrees (HMD) is proposed in [15] and [16] to address the shortcomings of conventional homogeneity. In [17], a linear feedback controller is studied to stabilize a power integrator system which has a homogeneity with strictly decreasing degrees (HSDD) with respect to the homogeneous weight vector $(1,\ 1, \dots,\ 1) \in \Re^n$. Based on the notion of HSDD, we can increase the powers of the states in a linear controller to locally asymptotically stabilize the uncertain upper-triangular system.

B. Reinforcement Learning in Control

Control theory has traditionally relied on mathematical models to predict system behavior and design controllers accordingly. However, with advancements in computational capabilities, researchers began to explore learning-based approaches that do not require a complete understanding of system dynamics. Early work in [18] established the foundation for integrating RL with control theory, emphasizing its potential to optimize control policies directly through system interaction without detailed models. Model-free RL methods, particularly Q-learning and policy gradient techniques, have been adapted for various control applications, from robotics to autonomous vehicles. These approaches allow systems to learn optimal strategies through trial and error, improving their performance in real-time as they interact with the environment [19]. Such techniques are particularly valuable in environments where the system dynamics are either too complex or costly to model accurately.

There has been significant interest in hybrid approaches that combine the robustness of traditional control methods with the adaptability of RL. For example, the work of [20] introduced Adaptive Dynamic Programming (ADP), a method that merges concepts from dynamic programming with RL to create controllers that can adapt to changes and uncertainties in the system [20]. These hybrid techniques aim to provide more reliable control in practical applications where safety and efficiency are critical. The application of RL in real-world control systems has been extensively documented across various domains. In robotics, RL has been used to develop controllers for complex tasks such as manipulation and locomotion, where precise model-based controllers would require prohibitive computational resources [21]. In the automotive industry, RL has been explored for developing advanced driver-assistance systems and fully autonomous driving technologies, demonstrating significant potential for improving safety and efficiency [22]. Despite its successes, the application of RL in control theory faces several challenges, including the need for improved sample efficiency, ensuring stability during the learning process, and addressing safety concerns inherent in deploying learning-based controllers in critical systems. Future research directions include the development of safer learning algorithms, methods to combine learning with robust control techniques, and strategies to reduce the data requirements of RL algorithms.

III. Reinforcement Learning and Nonlinear Integrated Controller

The proposed methodology integrates RL controller with a nonlinear controller to ensure both global and local stability in dynamic systems. In reinforcement learning, the policy π decides the behavior of an agent by mapping states to a probability distribution over the actions $\pi: \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$. Here, $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. The return from a state is defined as the sum of discounted future reward $R_t = \sum_{i=t}^{T} \gamma^{(i-t)}\, r(s_i, a_i)$ with a discounting factor $\gamma \in [0,1]$. The goal is to learn a policy which maximizes the expected return from the start distribution $J = \mathbb{E}_{r_i, s_i \sim E, a_i \sim \pi}[R_1]$.

The action-value function is used in many reinforcement learning algorithms. It describes the expected return after taking an action a_t in state s_t and thereafter following policy π:

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{r_{i \geq t}, s_{i > t} \sim E, a_{i > t \sim \pi}}[R_t | s_t, a_t] \tag{4}$$

Numerous methods in reinforcement learning utilize the recursive formula referred to as the Bellman equation.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi}[Q^{\pi}(s_{t+1}, a_{t+1})]] \tag{5}$$

When the target policy is deterministic, it can be represented as a function $\mu: S \leftarrow A$, thereby eliminating the need for the inner expectation.

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi}[Q^{\mu}(s_{t+1}, \mu(s_{t+1}))] \tag{6}$$

In this paper, we consider function approximators parameterized by θ^Q, which we optimize by minimizing the loss:

$$L(\theta^Q) = \mathbb{E}_{s_t \sim \rho^{\beta}, a_t \sim \beta, r_t \sim E}[(Q(s_t, a_t | Q^{\pi}) - y_t)^2] \tag{7}$$

where

$$y_t = r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, \mu(s_{t+1}) | \theta^Q) \tag{8}$$

while y_t is also dependent on θ^Q.

A. Controller Architecture

1) DDPG Reinforcement Learning Framework

The actor-critic structure in RL consists of two main components: the actor and the critic. The actor $\mu(s|\theta^\mu)$ is responsible for generating actions given the current state of the environment. It deterministically maps states to actions, aiming to maximize the expected reward. The critic $\mathrm{Q}(s,a)$ evaluates these actions by computing the value function, which estimates the expected return from the current state under the policy dictated by the actor. The actor updates its policy using a policy gradient method directed by the critic's value function gradient.

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s_t\sim\rho^\beta}\left[\nabla_{\theta^\mu} Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t|\theta^\mu)}\right] = \mathbb{E}_{s_t\sim\rho^\beta}\left[\nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)}\nabla_{\theta_\mu\mu}(s|\theta^\mu)|_{s=s_t}\right] \tag{9}$$

In this paper, we implemented the DDPG algorithm which uses neural network to approximate the actor and critic function.

Reinforcement Learning Integrated with Nonlinear Controller

a) Nonlinear Controller Design

In this section, we introduce the concept of homogeneity with strictly decreasing degrees (HSDD) and then show that the nested nonlinear controller guarantees the local asymptotic stability of the equilibrium $x = 0$ of system (1).

Assumption 1. There exists positive constants $\delta_i > 0$ and $c_i \geq 0$ such that the functions f_i satisfy $|f_i(x_{i+1},\dots,x_n)| \leq c_i\left(|x_{i+2}|^{1+\delta_i} + \sum_{j=i+2}^{n}|x_j|\right), 1 \leq i \leq n-1$, in a neighborhood of the origin, where c_i and δ_i do not need to be known.

Definition 2. A continuous vector field $v: \Re^n \to \Re^n$ with $v = [v_1,\dots,v_n]^T$ is said to satisfy homogeneity with strictly decreasing degrees (HSDD), if we can find positive real numbers $(r_1,\dots,r_n)$ and real numbers $\mu_1 > \mu_2 > \cdots > \mu_n$ such that $v_i(\epsilon^{r_1}x_1,\dots,\epsilon^{r_n}x_n) = \epsilon^{r_i+\mu_i}v_i(x)$ for all $x \in \Re^n, \epsilon > 0$ and $i = 1,2,\dots,n$. The constants r_i are the homogeneous weights and μ_i are homogeneous degrees.

Homogeneity with Strictly Decreasing Degrees (HSDD) is a subset of the broader concept of homogeneity, which includes systems with monotonically decreasing degrees (HMD). For HMD, the condition is that the sequence of degrees must satisfy $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ [25]. When all the values are identical, it becomes the classical notion of homogeneity. It's noted that linear controllers don't guarantee local asymptotic stability at the equilibrium $x = 0$ for the entire range of unknown parameters a_i. To address this, we select $r_1 \geq 1$ as a ratio of two positive odd integers, and $\mu_i, i = 1,\dots,n-1$ values are chosen as ratios of an even integer over an odd integer such that $\mu_1 > \mu_2 > \cdots > \mu_{n-1} > 0 =: \mu_n$, we define the powers r_i as $r_{i+1} = r_i + \mu_i$, $i = 1,\dots,n$. The linear controller $u = -(k_1x_1 + k_2x_2 + \cdots + k_nx_n)$ can be replaced as

$$u = -((\dots((k_1{x_1}^{\frac{r_2}{r_1}} + k_2x_2)^{\frac{r_3}{r_2}} + k_3x_3)^{\frac{r_4}{r_3}} + \cdots, +k_{n-1}x_{n-1})^{\frac{r_n}{r_{n-1}}} + k_nx_n) =: -\phi_n(x_1,\dots,x_n). \tag{10}$$

The closed-loop system (1) with controller (10) can be written as

$$\dot{x}_i = a_ix_{i+1} + f_i(x_{i+1},\dots,x_n), i = 1,2,\dots,n-1,$$

$$\dot{x}_n = -a_n\phi_n(x_1,\dots,x_n). \tag{11}$$

Let $e = [e_1,e_2,\dots,e_n]^T = \phi(x)$, we obtain

$$x_1 = k_1^{-1}e_1, x_i = k_i^{-1}(e_i - e_{i-1}^{r_i/r_{i-1}}), i = 2,\dots,n \tag{12}$$

Then:

$$\dot{e}_1 = \frac{k_1a_1}{k_2}(e_2 - e_1^{\frac{r_2}{r_1}}) + k_1\tilde{f}_1(e) = -\frac{k_1a_1}{k_2}e_1^{\frac{r_2}{r_1}} + g_1(e) \tag{13}$$

$$\dot{e}_i = \frac{k_ia_i}{k_{i+1}}(e_{i+1} - e_i^{\frac{r_{i+1}}{r_i}}) + k_i\tilde{f}_i(e)$$

$$+\frac{r_i}{r_{i-1}}e_{i-1}^{\frac{r_i}{r_{i-1}}-1}(-\frac{k_{i-1}a_{i-1}}{k_i}e_{i-1}^{\frac{r_i}{r_{i-1}}} + g_{i-1}(e)$$

$$= -\frac{k_ia_i}{k_{i+1}}e_i^{\frac{r_{i+1}}{r_i}} + g_i(e),$$

$$i = 2,3,\dots,n-1, \tag{14}$$

$$\dot{e}_n = -k_na_ne_n + \frac{r_n}{r_{n-1}}e_{n-1}^{\frac{r_n}{r_{n-1}}-1}(-\frac{k_{n-1}a_{n-1}}{k_n}e_{n-1}^{\frac{r_n}{r_{n-1}}} + g_{n-1}(e))$$

$$= -k_na_ne_n + g_n(e) \tag{15}$$

Where $\tilde{f}_i = f_i \circ \phi^{-1}, i = 1,2,\dots,n-1$, and g_i are recursively defined by

$$g_1(e) = \frac{k_1a_1}{k_2}e_2 + k_1\tilde{f}_1(e), \tag{16}$$

$$g_i(x) = \frac{k_ia_i}{k_{i+1}}e_{i+1} + k_i\tilde{f}_i(e) + \frac{r_i}{r_{i-1}}e_{i-1}^{\frac{r_i}{r_{i-1}}-1}(-\frac{k_{i-1}a_{i-1}}{k_i}e_{i-1}^{\frac{r_i}{r_{i-1}}} + g_{i-1}(e)), \tag{17}$$

$$g_n(e) = \frac{r_n}{r_{n-1}}e_{n-1}^{\frac{r_n}{r_{n-1}}-1}(-\frac{k_{n-1}a_{n-1}}{k_n}e_{n-1}^{\frac{r_n}{r_{n-1}}} + g_{n-1}(e)). \tag{18}$$

Theorem 3. Given that all functions f_i satisfy Assumption 1, the gains k_i are nonzero, and the ratios r_i are positive odd integers defined by $r_{i+1} = r_i + \mu_i, i = 1,\dots,n$, where $r_1 > 0$ and $\mu_1 > \mu_2 > \cdots > \mu_{n-1} > \mu_n = 0$. Under these conditions, the uncertain system is locally asymptotically stable for all $a_i > 0$ and f_i satisfying Assumption 1 if and only if $k_i > 0$ for all $i = 1,\dots,n$.

Proof. Construct the Lyapunov function as $V(e) = \sum_{i=1}^{n}\frac{l_i}{\alpha_i}{e_i}^{\alpha_i}$, where

$$l_i = -k_{i+1}k_i^{-1}a_i^{-1}, i = 1,2,\dots,n-1, l_n = -k_n^{-1}a_n^{-1}, \tag{19}$$

and $\alpha_i = 2r_n r_i^{-1} \geq 2, i = 1, 2, \dots, n$. The derivative of $V(e)$ can be calculated as

$$\dot{V}(e) = \sum_{i=1}^{n} e_i^{\alpha_i - 1} e_i^{\frac{r_{i+1}}{r_i}} + \sum_{i=1}^{n} l_i e_i^{a_i - 1} g_i(e), \tag{20}$$

where $r_{n+1} = r_n$ and $\alpha_n = 2$. From (21), it follows that

$$\dot{V}(e) \geq \sum_{i=1}^{n} e_i^{m_i} - \sum_{i=1}^{n} |l_i||e_i|^{\alpha_i - 1} |g_i(e)|, \tag{21}$$

where $m_i = \alpha_i - 1 + \frac{r_{i+1}}{r_i} = \frac{\mu_i + 2r_n}{r_i}, i = 1, 2, \dots, n$ are ratios of an even integer and an odd one. Applying (18) to (21), we can find constants ρ_i, $\hat{\rho}_i$ and $\tilde{\rho}_i$ such that

$$\dot{V}(e) \geq \sum_{i=1}^{n} e_i^{m_i} - \sum_{i=1}^{n} \rho_i |e_i|^{\alpha_i - 1 + \frac{\mu_i + 2r_i}{r_i}(1 + \hat{\delta}_i)} - \sum_{i=1}^{n} \sum_{j=1}^{i-1} \tilde{\rho}_i |e_i|^{\alpha_i - 1} |e_j|^{\frac{\mu_j + r_i}{r_j}} - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \hat{\rho}_i |e_i|^{\alpha_i - 1} |e_j|. \tag{22}$$

Let $\zeta_i = |e_i|^{m_i}$ for $i = 1, \dots, n$, rewrite the (22) as

$$\dot{V}(e) \geq \sum_{i=1}^{n} \zeta_i - \sum_{i=1}^{n} \rho_i \zeta_i^{\frac{\alpha_i - 1}{m_i} + \frac{r_{i+1}}{m_i r_i}(1 + \hat{\delta}_i)} - \sum_{i=1}^{n} \sum_{j=1}^{i-1} \tilde{\rho}_i |\zeta_i|^{\frac{\alpha_i - 1}{m_i}} |\zeta_j|^{\frac{\mu_j + r_i}{m_j r_j}} - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \hat{\rho}_i \zeta_i^{\frac{\alpha_i - 1}{m_i}} \zeta_i^{\frac{1}{m_j}} =: \sum_{i=1}^{n} \zeta_i - H(\zeta), \tag{23}$$

where $\zeta = [\zeta_1, \zeta_2, \dots, \zeta_n]$.

In the following analysis, we demonstrate that $H(\zeta)$ consists of higher-order terms with respect to ζ. By the expression of m_i below, it becomes evident that

$$\frac{\alpha_i - 1}{m_i} + \frac{r_{i+1}}{m_i r_i}(1 + \hat{\delta}_i) = 1 - \frac{r_{i+1}}{m_i r_i} + \frac{r_{i+1}}{m_i r_i}(1 + \hat{\delta}_i) = 1 + \frac{r_{i+1}}{m_i r_i}\hat{\delta}_i > 1. \tag{24}$$

$$\frac{\alpha_i - 1}{m_i} + \frac{\mu_j + r_i}{m_j r_j} = 1 - \frac{r_{i+1}}{\mu_i + 2r_n} + \frac{\mu_i + r_i}{\mu_j + 2r_n}. \tag{25}$$

Because $\mu_j > \mu_i$ for $j < i$ and $2r_n > r_i$, we obtain

$$\frac{\alpha_i - 1}{m_i} + \frac{\mu_j + r_i}{m_j r_j} > 1 - \frac{r_{i+1}}{\mu_i + 2r_n} + \frac{\mu_i + r_i}{\mu_j + 2r_n} = 1. \tag{26}$$

Since $r_j \geq r_{i+1}$ and $\mu_i > \mu_j$ for $j \geq i + 1$,

$$\frac{\alpha_i - 1}{m_i} + \frac{1}{m_j} = 1 - \frac{r_{i+1}}{m_i r_i} + \frac{1}{m_j} = 1 - \frac{r_{i+1}}{\mu_i + 2r_n} + \frac{r_j}{\mu_j + 2r_n} > 1. \tag{27}$$

So, we can conclude that $H(\zeta)$ has an order greater than 1. Therefore, there exists a sufficiently small neighborhood around the origin such that

$$\dot{V}(e) > \varepsilon \sum_{i=1}^{n} \zeta_i = \varepsilon \sum_{i=1}^{n} e_i^{m_i}, \forall e \in D \tag{28}$$

for a constant $\varepsilon > 0$. From (19), we obtain

$$l_i < 0 (i = 1, 2, \dots, n) \Leftrightarrow k_i > 0 (i = 1, 2, \dots, n). \tag{29}$$

If $k_i > 0 (i = 1, 2, \dots, n)$, $V(e)$ is negative definite. This implies that the zero solution of (11) is asymptotically stable according to Lyapunov Stability Theorem. Therefore, the positivity of k_i is sufficient to guarantee the local asymptotic stability of (11).

b) Integration Mechanism

The objective of integrating a RL method with a nonlinear controller is to leverage the adaptive learning capability of RL for global system behaviors while utilizing the precision and stability of nonlinear control techniques for local scenarios. The integrated controller consists of a global RL controller and a local nonlinear controller. The global controller operates across the entire state space, adapting to complex, non-linear dynamics and learning optimal control strategies through interaction with the environment. The local controller provides precise control in critical regions, such as near setpoints or within stability boundaries, where traditional control methods are effective and reliable. When the tracking error or the system's response reach to a pre-set threshold, control switches from the RL controller to the nonlinear controller.

B. Training Process

The parameters of the actor and critic networks are initialized with random weights. The target networks are initialized to the weights of their corresponding original networks. An experienced replay buffer $\mathcal{D}$ is introduced to store transition tuples (s_t, a_t, r_t, s_{t+1}), where s_t is the current state, a_t is the action taken, r_t is the reward received, and s_{t+1} is the next state. During training, the actor's deterministic policy is supplemented with noise for exploration purposes. At each training step, a mini batch of N transitions are sampled from $\mathcal{D}$. The sampled batch is used to compute the loss for the critic network, using the Bellman equation as the target for the temporal difference (TD) error. The policy is updated using the sampled policy gradient, estimated from the critic's Q-values with respect to the actions. After each learning step, the target networks' weights are updated to slowly track the learned networks' weights, employing a soft update strategy controlled by. The process repeats with the agent interacting with the environment according to the current policy and exploration noise, storing the experiences, and periodically updating the actor and critic networks using mini batches from the replay buffer.

Algorithm 1 DDPG algorithm

```
Initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights $\theta^Q$, and $\theta^\mu$ respectively
Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer $R$
for each episode
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial observation state $s_1$
    for each step $t$
        Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
        Execute action $a_t$ in the environment
        Observe reward $r_t$ and observe new state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $R$
        Sample a random minibatch of $N$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $R$
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$ for updating the critic
        Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
        Update the actor policy using the sampled policy gradient:
            $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
        Update the target networks:
            $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
            $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
```

IV. Example

In this section, we compare the proposed controllers to standard trained DDPG controllers. Here, we implement the two controllers to the dynamics of orientation of a car [25]

$$\dot{\theta} = \frac{v}{l} tan\phi + unknown\ dynamics, \dot{\phi} = \omega, \dot{\omega} = \frac{1}{J} u \qquad (30)$$

θ represents the car's angular deviation from the x-axis, φ denotes the angle of the steering wheel relative to the car's longitudinal axis, ω signifies the angular velocity of the steering wheels, v indicates the linear velocity of the rear axle's center during the cruising phase, l stands for the distance between the steering wheels and the rear wheels, J denotes the moment of inertia, and u represents the external torque. Assuming v, l, and J are unknown constants, system (30) can be expressed in the format of (1) while satisfying Assumption 1. We choose $r_1 = 3, r_2 = 5$, and $r_3 = 33$. From (3), we derive the following nonlinear controller as a local controller:

$$u = -\left(\left((k_1 x_1)^{5/3} + k_2 x_2\right)^{33/25} + k_3 x_3\right). \qquad (31)$$

In this example, we implemented controllers to systems with initial values $(0.3, 0.2, 0.1)$ and $(2\pi, 1, 1)$, and the objective is to direct the initial value to zero. We choose v, l, j as $2, 3, 4$ separately, and k_1, k_2, k_3 as $1, 2, 1$. The unknown dynamics is set as ϕ^2.

As shown in the Fig. 3 and Fig. 4, the blue line represents the system dynamic with reinforcement learning controller, and the green line represents system dynamic with integrated controller. Both the RL controller and integrated controller stabilize the uncertain upper-triangular system. The RL controller converges faster than the integrated controller. However, the integrated controller stabilizes the system with smaller error.

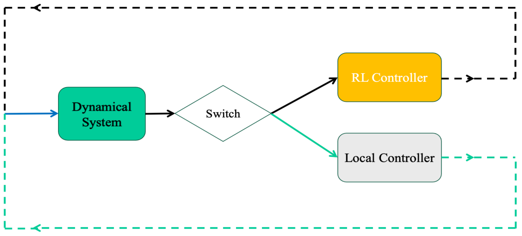

Fig. 1. Integration mechanism

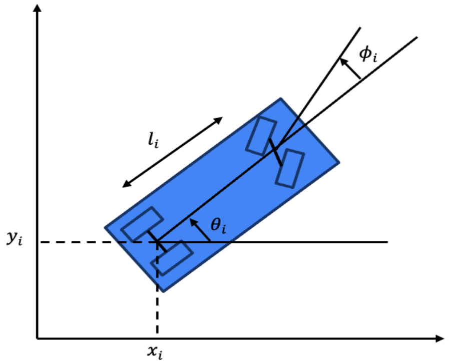

Fig. 2. Dynamic of orientation of a car.

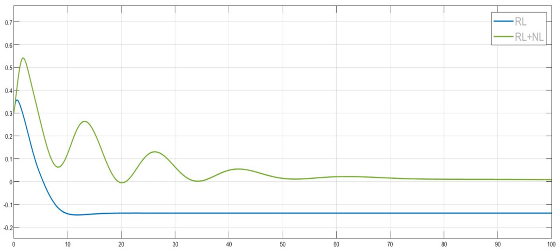

Fig. 3. Integrated Controller vs. RL Controller with initial value $(0.3, 0.2, 0.1)$.

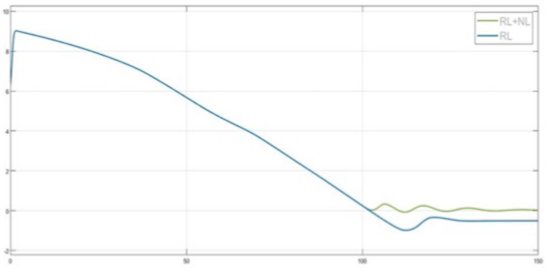

Fig. 4. Integrated Controller vs. RL Controller with initial value $(2\pi, 1, 1)$.

V. Conclusion

In this paper, we proposed a reinforcement learning integrated nonlinear controller for stabilizing a class of uncertain upper-triangular system. A Lyapunov-based actor-critic reinforcement learning framework is introduced for guaranteed stability. We compared the RL controller and integrated controller in simulation. It shows that both controllers stabilize the uncertain upper-triangular system. The proposed integrated nonlinear controller demonstrates improved precision.

References

[1] Mazenc, F., & Bowong, S. (2003). Tracking trajectories of the cart-pendulum system. Automatica, 39(4), 677–684.

[2] Barbu, C., Sepulchre, R., Lin, W., & Kokotovic, P. V. (1997). Global asymptotic stabilization of the ball-and-beam system. In Proceedings of the 36th IEEE conference on decision & control (pp. 2351–2355).

[3] Arcak, M., Teel, A. R., & Kokotovic, P. V. (2001). Robust nonlinear control of feedforward systems with unmodeled dynamics. Automatica, 37(2), 265–272.

[4] Ding, S., Qian, C., & Li, S. (2010). Global stabilization of a class of feedforward systems with lower-order nonlinearities. IEEE Transactions on Automatic Control, 55(3), 691–696.

[5] Krishnamurthy, P., & Khorrami, F. (2004). A high-gain scaling technique for adaptive output feedback control of feedforward systems. IEEE Transactions on Automatic Control, 49(12), 2286–2292.

[6] Qian, C., & Lin, W. (2012). Homogeneity with incremental degrees and global stabilisation of a class of high-order upper-triangular systems. International Journal of Control, 85(12), 1851–1864.

[7] Teel, A. R. (1992). Feedback Stabilization: Nonlinear Solution to Inherently Nonlinear Problems (Ph.D. thesis), Berkeley: University of California.

[8] Ye, X., & Jiang, J. (1998). Adaptive nonlinear design without a priori knowledge of control directions. IEEE Transactions on Automatic Control, 43(11), 1617–1621.

[9] Jiandong Zhu, Chunjiang Qian, Local asymptotic stabilization for a class of uncertain upper-triangular systems, Automatica, Volume 118, 2020, 108954, ISSN 0005-1098.

[10] Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, et al. 2017. Emergence of Locomotion Behaviours in Rich Environments. arXiv preprint arXiv:1707.02286 (2017).

[11] Jungdam Won, Jongho Park, Kwanyu Kim, and Jehee Lee. 2017. How to Train Your Dragon: Example-guided Control of Flapping Flight. ACM Trans. Graph. 36, 6, Article 198 (Nov. 2017), 13 pages.

[12] Jungdam Won, Jungnam Park, and Jehee Lee. 2018. Aerobatics Control of Flying Creatures via Self-regulated Learning. ACM Trans. Graph. 37, 6, Article 181 (Dec. 2018), 10 pages.

[13] Hermes, H. (1995). Homogeneous feedback controls for homogeneous systems. Systems & Control Letters, 24(1), 7–11.

[14] Rosier, L. (1992). Homogeneous Lyapunov function for homogeneous continuous vector field. Systems & Control Letters, 19(6), 467–473.

[15] Polendo, J., & Qian, C. (2007). A generalized homogeneous domination approach for global stabilization of inherently nonlinear systems via output feedback. International Journal of Robust and Nonlinear Control, 17(7), 605–629.

[16] Zhang, C., Qian, C., & Li, S. (2013). Global smooth stabilization of a class of feedforward systems under the framework of generalized homogeneity with monotone degrees. Journal of the Franklin Institute, 350(10), 3149–3167.

[17] Zhu, J., & Qian, C. (2018). A necessary and sufficient condition for local asymptotic stability of a class of nonlinear systems in the critical case. Automatica, 96, 234–239.

[18] Sutton, R. S., Barto, A. G., 2018. Reinforcement Learning: An Introduction, 2nd Edition. MIT Press

[19] Kober, J., Bagnell, J. A., Peters, J., 2013. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research 32 (11), 1238–1274.

[20] Lewis, F., Liu, D. (Eds.), 2012. Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control. Wiley.

[21] Deisenroth, M., Neumann, G., Peters, J., 2011. A survey on policy search for robotics. Foundations and Trends in Robotics 2 (1–2), 1–141.

[22] Borrelli, F., Bemporad, A., Morari, M., 2017. Predictive Control for Linear and Hybrid Systems. Cambridge University Press, Cambridge, UK.

[23] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, A Lyapunov-based approach to safe reinforcement learning, arXiv preprint arXiv: 1805.07708, 2018

[24] Y. Chow, O. Nachum, A. Faust, M. Chavamzaded, and E. Duenez-Guzman, Lyapunov-based safe policy optimization for continuous control, arXiv preprint arXiv:1901.10031, 2019

[25] Polendo, J., & Qian, C. (2008). An expanded method to robustly stabilize uncertain nonlinear systems. Communications in Information and Systems, 8(1), 55–70.