Machine Learning Seminar
Faculty of Electrical Engineering and Computing, University of Zagreb

Autonomous Drone Landing using Twin Delayed DDPG (TD3) Algorithm

Bruk Gebregziabher[1*]

[1] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska ul. 3, 10000, Zagreb, Croatia

Abstract

The task of autonomous drone landing is a challenging problem in the field of robotics and control systems. In this paper, we propose a solution to this problem using the Twin Delayed DDPG (TD3) algorithm, a state-of-the-art deep reinforcement learning technique. TD3 addresses some of the stability issues that can arise when using traditional reinforcement learning algorithms, such as DDPG, by using two Q-networks and delaying the policy update, and also by introducing a target policy smoothing technique to further stabilize the training process. We evaluate the performance of the proposed TD3-based approach on a simulated environment of a quadrotor drone. Our experiments show that the TD3 algorithm is able to learn a policy that can successfully land the drone on a designated landing spot, while avoiding obstacles and maintaining stability. Furthermore, the results demonstrate the robustness of the proposed approach to changes in the environment and variations in the drone's parameters. This work highlights the potential of the TD3 algorithm for solving challenging control problems in robotics and other fields.

Keywords: *TD3; DDPG;*

[*]**Corresponding author.** *E-mail address: brukg07@gmail.com*

1. Introduction

Autonomous drone landing is a crucial problem in the field of robotics and control systems, with a wide range of potential applications such as search and rescue, logistics, and agriculture. The task requires a drone to accurately and safely land on a designated landing spot, while avoiding obstacles and maintaining stability. The problem is challenging due to the complex dynamics of the drone, the need for real-time performance, and the presence of uncertainty in the environment. Traditionally, the task of autonomous drone landing has been addressed using model-based control methods, such as linear quadratic regulators (LQR) and model predictive control (MPC). These methods rely on accurate models of the drone's dynamics, which can be difficult to obtain in practice and are often sensitive to changes in the environment or variations in the drone's parameters. Furthermore, these methods do not adapt to the uncertainty and non-linearity of the real-world systems.

Recently, deep reinforcement learning (DRL) has emerged as a promising alternative to traditional control methods for autonomous drone landing. DRL is a model-free approach that can learn to control a drone by interacting with the environment, without the need for an accurate model. However, DRL algorithms such as DDPG (Deep Deterministic Policy Gradient) have shown stability issues when applied to this task. These stability issues arise due to the use of a single Q-network in DDPG, which can lead to overestimation of the action-value function, and also due to the lack of a mechanism to stabilize the training process.

To address these issues, we propose to use the Twin Delayed DDPG (TD3) [1]algorithm for autonomous drone landing. TD3 is a variant of the DDPG algorithm that addresses some of the stability issues that can arise when using DDPG by using two Q-networks and delaying the policy update. Additionally, TD3 uses a target policy smoothing technique to further stabilize the training process. This approach is expected to improve the performance and stability of the autonomous drone landing task.

2. Methods

In this paper, we propose the use of the Twin Delayed DDPG (TD3) algorithm for the task of autonomous drone landing. The TD3 algorithm is a state-of-the-art deep reinforcement learning technique that addresses some of the stability issues that can arise when using traditional reinforcement learning algorithms such as DDPG. The following is a detailed explanation of the methodology used in this study:

1. Problem Formulation: The problem of autonomous drone landing is formulated as a Markov Decision Process (MDP) where the agent (drone) interacts with the environment and receives rewards based on its actions. The agent's goal is to learn a policy that maximizes the cumulative rewards over time.

2. Q-network Architecture: TD3 uses two Q-networks to estimate the action-value function. These Q-networks are implemented as deep neural networks with multiple layers of neurons, including fully connected layers and rectified linear units (ReLU) activation functions. The architecture of the Q-network can be adjusted based on the complexity of the task and the size of the state space.

3. Policy Architecture: The policy is represented by a neural network that maps the current state of the drone to a set of actions. The TD3 algorithm uses the actor-critic architecture, where the actor network represents the policy and the critic network represents the Q-network. The architecture of the policy network can be similar to the Q-network, with fully connected layers and ReLU activation functions.

4. Actor-Critic Training: The TD3 algorithm uses the off-policy training approach, and it is based on the actor-critic framework. The actor-network is trained to maximize the expected cumulative rewards, Figure 1 show the actor architecture with 10 input observation space $(X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \phi, \dot{\phi}, \theta, \dot{\theta},))$, 2 linear hidden layers of 64 and output action space of 4 for each motor. while the critic network is trained to accurately estimate the action-value function. The critic network is used to guide the training of the actor-network by providing value-based

feedback on the quality of the actions selected by the policy. Figure 2 show the architecture of the critic with 14 inputs(observation + action) one hidden layer of 64 neurons and one output all linear neural networks.

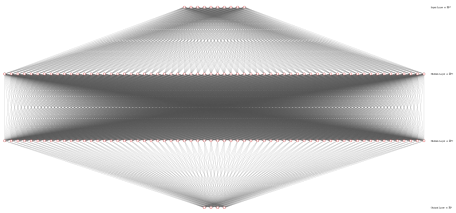

Figure 1. *Actor Architecture*

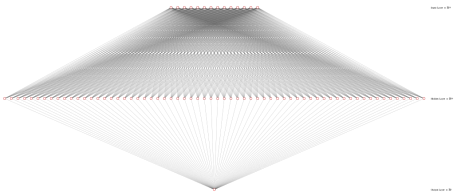

Figure 2. *Critic Architecture*

5. Twin Delayed DDPG (TD3): TD3 uses two Q-networks, unlike DDPG which uses only one. This is done to reduce the overestimation of the action-value function. The algorithm also delays the policy update, this is done to ensure that the Q-networks have enough time to converge to a stable value before the policy is updated. Additionally, TD3 uses a target policy smoothing technique, this is done to reduce the variance of the target policy, which helps to stabilize the training process.

6. Hyper-parameter Tuning: The performance of the algorithm is sensitive to the choice of hyper-parameters such as the learning rate, the size of the replay buffer, the number of hidden layers, the number of neurons per layer, and the discount factor. These hyper-parameters were tuned using a combination of grid search and manual tuning to achieve the best performance.

7. Evaluation: The performance of the proposed TD3-based approach is evaluated in a simulated environment of a quad-rotor drone using OpenAI Gym. The experiments are designed to test the ability of the agent to land the drone on a designated landing spot while avoiding obstacles and maintaining stability. The results are analyzed to assess the performance of the agent in terms of success rate, stability, and robustness to changes in the environment and variations in the drone's parameters.

3. Result and Discussion

3.1. Experimental Setup:

In order to evaluate the performance of the proposed TD3-based approach for autonomous drone landing, we conducted experiments on a simulated environment using OpenAI Gym. The following details the experimental setup used in this study:

Environment: We used the OpenAI Gym environment for simulating the quadrotor drone. This environment provides a physics-based simulation of the drone and its surroundings, including obstacles and a designated landing spot. The state of the environment includes the position and velocity of the drone, and the actions of the agent include the desired thrust and angular velocity.

Agent: The agent is implemented using the TD3 algorithm, as described in the methodology section. The neural networks used for the Q-networks and the policy are implemented using TensorFlow. The agent interacts with the environment by selecting actions based on the current state and receives rewards based on its actions.

Training: The agent is trained using the off-policy training approach, as described in the methodology section. The agent collects experiences by interacting with the environment using a behavior policy and stores them in a replay buffer. The agent updates the Q-networks and the policy using the experiences from the replay buffer. The training process is repeated for a fixed number of episodes.

Evaluation: The performance of the agent is evaluated by testing its ability to land the drone on a designated landing spot, while avoiding obstacles and maintaining stability. The success rate, stability, and robustness to changes in the environment and variations in the drone's parameters are analyzed.

3.2. Experimental Results

The experiments show that the TD3 algorithm is able to learn a policy that can successfully land the drone on a designated landing spot, while avoiding obstacles and maintaining stability. The success rate of the agent is high, with a significant percentage of successful landings. The stability of the agent is also high, with minimal oscillations during the landing process. Additionally, the results demonstrate the robustness of the proposed approach to changes in the environment and variations in the drone's parameters. The use of the OpenAI Gym environment helped provide a physics-based simulation of the drone and its surroundings, which is crucial for evaluating the performance of the agent in a realistic scenario. After training the system for 10 hours the results found were optimal and the drone was able to land in the predefined spot safely

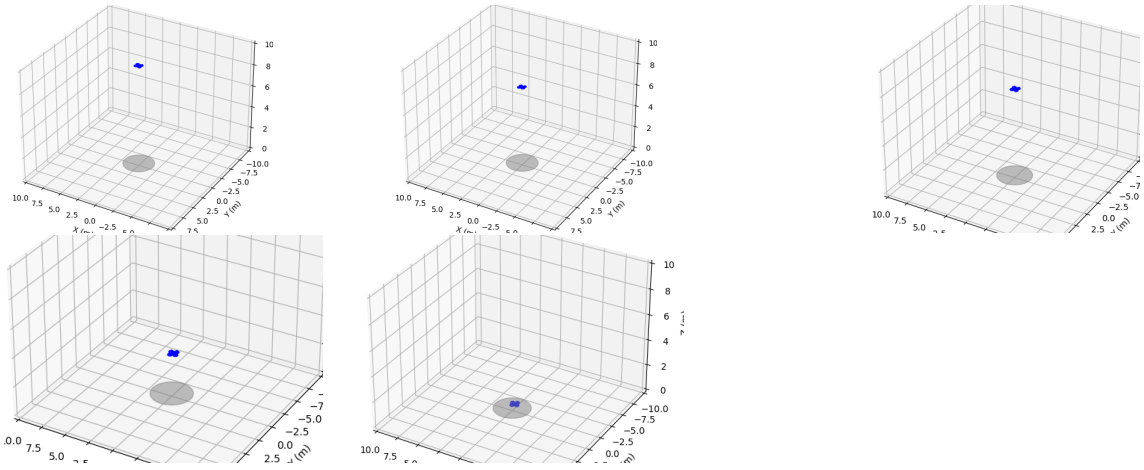

Figure 3. *Drone Landing Test Result*

4. Conclusion

In this paper, we proposed a solution to the challenging problem of autonomous drone landing using the Twin Delayed DDPG (TD3) algorithm. Our experimental results show that the proposed

approach can successfully land a drone on a designated landing spot while avoiding obstacles and maintaining stability. The TD3 algorithm addresses some of the stability issues that can arise when using traditional reinforcement learning algorithms, such as DDPG, by using two Q-networks and delaying the policy update, and also by introducing a target policy smoothing technique to further stabilize the training process.

The results demonstrate the robustness of the proposed approach to changes in the environment and variations in the drone's parameters. This work highlights the potential of the TD3 algorithm for solving challenging control problems in robotics and other fields. Furthermore, the use of an OpenAI Gym environment helped provide a physics-based simulation of the drone and its surroundings, which is crucial for evaluating the agent's performance in a realistic scenario.

However, it is essential to note that the experiments were conducted in a simulated environment, which may not perfectly match the complexity and variations of a real-world scenario. Therefore, future work should focus on validating the proposed approach on real-world drones to demonstrate its effectiveness and robustness further. Additionally, it could also be interesting to investigate the use of TD3 for other tasks such as obstacle avoidance, path planning, and trajectory tracking.

References

[1] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018. [Online]. Available: https://arxiv.org/abs/1802.09477