

Article

UAV Autonomous Tracking and Landing Based on Deep Reinforcement Learning Strategy

Jingyi Xie ^{1,2}, Xiaodong Peng ^{1,2,*}, Haijiao Wang ³, Wenlong Niu ^{1,2} and Xiao Zheng ^{1,2}

¹ Key Laboratory of Electronics and Information Technology for Space System, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China; xiejingyi18@mails.ucas.ac.cn (J.X.); niuwenlong@nssc.ac.cn (W.N.); zhengxiao@nssc.ac.cn (X.Z.)

² University of Chinese Academy of Sciences, Beijing 100049, China

³ Alibaba Damo Academy, Hangzhou 311121, China; wanghaijiao@cool.com

* Correspondence: pxd@nssc.ac.cn

Received: 17 August 2020; Accepted: 28 September 2020; Published: 1 October 2020



Abstract: Unmanned aerial vehicle (UAV) autonomous tracking and landing is playing an increasingly important role in military and civil applications. In particular, machine learning has been successfully introduced to robotics-related tasks. A novel UAV autonomous tracking and landing approach based on a deep reinforcement learning strategy is presented in this paper, with the aim of dealing with the UAV motion control problem in an unpredictable and harsh environment. Instead of building a prior model and inferring the landing actions based on heuristic rules, a model-free method based on a partially observable Markov decision process (POMDP) is proposed. In the POMDP model, the UAV automatically learns the landing maneuver by an end-to-end neural network, which combines the Deep Deterministic Policy Gradients (DDPG) algorithm and heuristic rules. A Modular Open Robots Simulation Engine (MORSE)-based reinforcement learning framework is designed and validated with a continuous UAV tracking and landing task on a randomly moving platform in high sensor noise and intermittent measurements. The simulation results show that when the moving platform is moving in different trajectories, the average landing success rate of the proposed algorithm is about 10% higher than that of the Proportional-Integral-Derivative (PID) method. As an indirect result, a state-of-the-art deep reinforcement learning-based UAV control method is validated, where the UAV can learn the optimal strategy of a continuously autonomous landing and perform properly in a simulation environment.

Keywords: quadrotor unmanned aerial vehicle; deep reinforcement learning; autonomous tracking and landing

1. Introduction

In recent years, with the rapid development of unmanned aerial vehicle (UAV) technology, the UAV has been widely used in military and civilian fields, such as search, rescue, exploration, and surveillance [1,2]. Autonomous tracking and landing is a key point in UAV application [3–6]. However, compared to landing in a simulated environment or a static platform, autonomous tracking and landing is more difficult because classical techniques have their limitations, in terms of model design and non-linearity approximation [7]. Furthermore, due to the absence of precise sensors and the constraint of the sensors' specific physical motion, autonomous tracking and landing works poorly with high sensor noise [6,8] and intermittent measurements [9,10].

Given the importance and complexity of the UAV tracking and autonomous landing, increasingly more scholars from different fields have shown interest in specific solutions such as perception and relative pose estimation [11,12] or trajectory optimization and control [13,14]. Regarding

the control maneuvers when the relative state of the moving platform is assumed to be known, the Proportional-Integral-Derivative (PID) controller is the mainstream algorithm for aggressive landing from relatively short distances [4,5,15], but a fixed gain of the PID controller cannot provide immediate response to overcome a nonlinear effect, moreover, PID gain tuning is a crucial part and needs a lot of effort for optimal gain [16]. This calls for nonlinear control approaches for more precise control of UAV. By applying a state-dependent coefficient (SDC)-based nonlinear model inversion, the authors of reference [17] eliminated the need for linearization of the aircraft dynamics, but this approach is quite sensitive towards sensor noise. In order to solve the problem of UAV autonomous landing under special circumstances, such as landing on top of a moving inclined platform [14,18], or landing on a platform moving in a figure of eight [3,19] (the task of the competition in Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017), a Model Predictive Controller (MPC) tracker was used to generate UAV feasible trajectories, which could minimize an error of UAV future states over a prediction horizon to fly precisely above the car given the dynamical constraints of the aircraft. MPC holds the ability to anticipate upcoming events and can yield control inputs accordingly, however, the development of accurate prediction models requires a tiresome design effort. Sliding mode methods have been widely used in UAV autonomous tracking and landing control algorithms [20–22]. This approach changes the UAV nonlinear dynamics by the application of a discontinuous control signal, but the main issue with sliding mode control is chattering and a high control demand [7]. Unlike the linear and non-linear controllers mentioned above, a fuzzy logic controller does not depend on a precise mathematical model. Regardless of the good performance by PID controllers, these still need to be adaptive for uncertain conditions. To achieve this purpose, PID control was extended further to fuzzy adaptive PID [23,24], and the authors of reference [16,25] presented a fuzzy logic-based UAV tracking and landing using computer vision.

Different from the above methods in solving the control of a UAV by building a prior model and then making a decision based on a dynamics model, reinforcement learning is an ideal solution to deal with unknown system dynamics in different tracking and landing circumstances [26]. Recently, significant progress has been made by combining deep learning with reinforcement learning, resulting in the deep Q-learning network (DQN) [27], the deterministic policy gradient (DPG) [28], and the asynchronous advantage actor-critic (A3C) [29]. These algorithms have achieved unprecedented success in challenging domains, such as the Atari 2600 [27].

Concerning deep reinforcement learning for UAV autonomous tracking and landing tasks, a hierarchy of DQN was proposed in [30,31], which is used as a high-end control policy for the navigation in different phases. However, the flexibility of the method is limited because the UAV's action space is defined by discrete space rather than continuous space. Furthermore, the authors of reference [32] addressed the full problem with continuous state and actions spaces based on Deep Deterministic Policy Gradients (DDPG) [33]. However, the altitude (z-axis) is not included in the framework, there is a significant design effort during the reward plasticity, and the proposed Gazebo-based reinforcement learning framework has a weak generalization capability, which results in less autonomous agents.

In general, the current research on the problem of a UAV autonomous tracking and landing is mainly focused on the situation in which the platform has a constant speed. However, more sophisticated control is required to operate in unpredictable and harsh environments, such as high sensor noise and intermittent measurement. In terms of the trajectory control or classic controller, e.g., PID, nonlinear, fuzzy logic controllers are the mainstream algorithm, but these techniques are limited to model design, non-linear approximation, and disturbances rejection. Although, in several studies, the deep reinforcement learning (DRL)-controller has performed well in UAV tracking and landing problems, the control problem of UAV tracking and landing on a randomly moving platform becomes intractable, due to the low reliability and weak generalization ability of the mathematical model.

We decompose the problem of landing on a moving platform into two aspects: one is perception and relative pose estimation, the other is trajectory optimization and control. The reader is referred

to [6,34] for algorithms which are used particularly for landing on a platform using visual inputs from UAV. In the present paper, we focus on trajectory planning and control, aiming to solve the control problem of UAV tracking and landing on a moving platform. We build a novel model-free-based tracking and landing algorithm to solve the problem of sensor noise and intermittent measurements. First, by taking the sensor noise, intermittent measurements, and randomness of UAV movement into consideration, a novel dynamic model based on the partially observable Markov decision process (POMDP) [35] is built to describe the autonomous process of UAV tracking and landing. Then, an end-to-end neural network is used to approximate the action controller of UAV autonomous tracking and landing. Finally, a DRL-based algorithm is adopted to train the neural network to learn from the tracking and landing experience.

The rest of this article is organized as follows. In Section 2, we introduce a UAV autonomous tracking and landing model based on the POMDP. In Section 3, we discuss the hybrid strategy included with deep reinforcement learning and heuristic rules in order to calculate the optimal control output and realize UAV autonomous tracking and landing tasks. In Section 4, we present the experimental results of our methods on a Modular Open Robots Simulation Engine (MORSE) simulator. Conclusions and future work are drawn in Section 5.

2. Problem Definition

Generally, UAV autonomous tracking and landing could be described as follows: Given a moving platform, such as a car or truck, and a UAV, the moving state of the UAV and platform could be detected by some kind of sensors; however, only part of the moving state of the platform could be observed. The observation states of both platform and UAV may have some errors and information update delay, so the observation could not be regarded as accurate. The task of UAV autonomous tracking and landing is to control the UAV with a proper speed or accurate speed, so as to land on the platform properly. Unlike previous research [30,32,36], in this paper, incomplete and inaccurate observations are taken into consideration, which makes it more consistent with real situations.

2.1. UAV Autonomous Landing System

As mentioned above, an autonomous tracking and landing system consists of a UAV and platform (shown in Figure 1), and, to keep the example simple, we assume that the sensors attached to the UAV and platform could only observe the speed and position of the UAV and platform. In addition, the observations of both the UAV and platform have a certain frequency, so no real-time information could be observed, such as the state update delay, making autonomous tracking and landing more difficult.

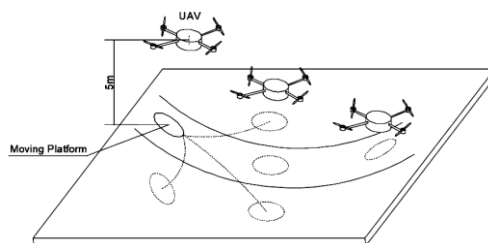


Figure 1. Process of an unmanned aerial vehicle's (UAV) autonomous tracking and landing, which includes a moving platform and a UAV, in which the UAV's task is to achieve the tracking of the moving platform and landing on it.

Naturally, observations of a system have a certain frequency, which means that multiple states with a timeline could be obtained, as shown in Figure 2. At each time a state is observed, a proper decision should be made (which usually means some control parameters, such as speed), and then a new state could be observed and a new decision should be made. Such a process would continue until

the UAV lands successfully or reaches a certain step state. Therefore, the control of UAV autonomous tracking and landing can be regarded as a sequential decision problem.

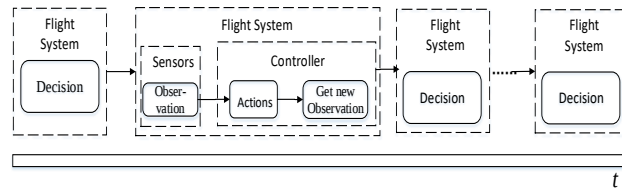


Figure 2. At each moment, the UAV sequential decision process can be divided into three steps: obtaining state information of the current system, performing actions, and updating the state.

2.2. POMDP Mathematical Model

As mentioned above, the UAV autonomous tracking and landing process is partially observable and can be seen as a sequential decision problem. Therefore, the partially observable Markov decision process is adopted to describe this process.

Typically, the POMDP consists of tuples $(S, A, T, R, O, \Omega, \gamma)$, where S is a set of states, A is a set of actions, and Ω is a set of observations. O is a conditional observation probability and the functions T and R and factor γ define a Markov decision process [37], where T is a transition function, R denotes immediate rewards, and γ is a discount factor. Based on the POMDP mathematical model, the UAV autonomous tracking and landing model is described as follows:

1. State S : Represents the state of the system, including the state of the UAV and moving platform; the system coordinates are constructed as in Figure 3. On the basis of this coordinate system, the state information could be described by the respective speeds and positions of the UAV and platform, as in Equation (1):

$$s = \{X_u, Y_u, Z_u, v_{ux}, v_{uy}, v_{uz}, X_t, Y_t, Z_t, v_{tx}, v_{ty}, v_{tz}\} \quad (1)$$

where $v_{ux}, v_{uy} \in [-10, 10]$, $v_{uz} \in [-1, 3]$, $v_{tx}, v_{ty} \in [-5, 5]$, $v_{tz} = 0$.

2. Action A : The speeds of the UAV are used as action parameters defined as $A = \{v_x, v_y, v_z\}$.
3. Transition function T : Represents the dynamic model of the UAV autonomous tracking and landing system, which is difficult to model and describe explicitly, in this work the transition model is not given (model-free).
4. Observation Ω : Owing to the physical limitations of the sensor, one could only obtain the velocity and position information of the UAV and moving platform with sensor noise or intermittent measurements, and the observation is defined as

$$\Omega = \{X_u', Y_u', Z_u', v_{ux}', v_{uy}', v_{uz}', X_t', Y_t', Z_t', v_{tx}', v_{ty}', v_{tz}'\}. \quad (2)$$

5. Reward function R : Because one of our goals is to minimize the distance between the UAV and the moving platform, a positive reward of 10 is given when the distance between the UAV and the platform is less than a certain threshold, and a negative reward of -10 is given if the distance between the UAV and the platform is more than a certain threshold.

Additionally, due to the complexity of the UAV control, when the agent explores its environment in the early stages, the event of randomly reaching the distance between the UAV and the moving platform within a certain threshold is rare, and thus cannot provide the agent with enough information to converge. Conversely, it is the best to receive reward signals at each time step of UAV tracking, so the intermediate reward between a positive and negative reward is designed, as shown in Equation (3),

where the agent maximizes the reward (thereby minimizing distance) in order to track the target as accurately as possible.

$$R = \begin{cases} -10, dist > 6 \\ -0.1 \times dist, 0.8 \leq dist \leq 6 \\ +10, dist < 0.8 \end{cases} \quad (3)$$

where the variable $dist$ is given by

$$dist = \sqrt{(X_u - Y_t)^2 + (X_u - Y_t)^2}. \quad (4)$$

6. Discount factor γ is a parameter used to incorporate the future reward into current action. We used an action-value function to evaluate the learning outcome, which describes the expected accumulated discounted reward after taking an action a_i in state s_i and, thereafter, following policy π :

$$Q^\pi(s_i, a_i) = R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P_{s_i s_{i+1}}^{a_i} v^\pi(s_{i+1}) \quad (5)$$

where S_{i+1} denotes the state information on the next moment, $P_{s_i s_{i+1}}^{a_i}$ is the Markov dynamics model of the system, and v^π is the total expected reward under the strategy π . The goal of this paper is to find the strategy $\pi(a_i|s_i)$ to maximize $Q^\pi(s_i, a_i)$, which means finding an optimal strategy for selecting actions based on the information observed by the UAV to realize UAV autonomous tracking and landing tasks.

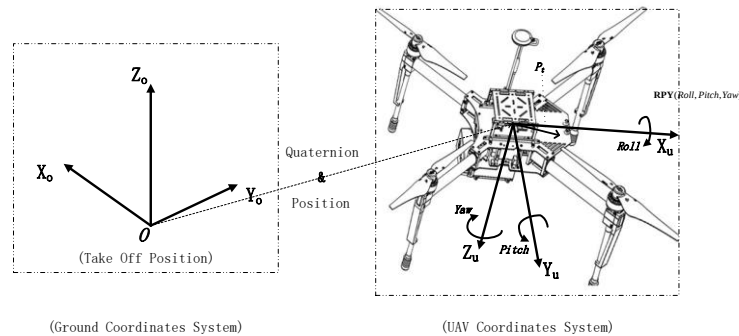


Figure 3. The UAV coordinate system and ground coordinate system are constructed separately, and the pose and velocity of the moving platform and UAV are obtained through coordinates.

3. UAV Autonomous Tracking and Landing Method Based on Hybrid Strategy

If the UAV autonomous landing problem can be described by an accurate mathematical model, then we can solve the objective function of the Markov decision model (shown in Equation (5)) by an iterative solution based on the direct method or indirect method of the optimal control theory and then directly obtain the above optimal decision strategy of the Markov decision process. However, as described in Section 2, the model of the target system is unknown and difficult to describe. Much of the current research [38,39] uses a reduced model of the UAV for generating the optimal landing trajectory for landing on a moving target. However, such an approximation may not be accurate, and, consequently, the landing performance is decreased. Besides, most methods take a state as inputs and assume that the observation of the state is accurate, which is unlikely to be true. Different from the previous definition of optimal control [40], the UAV optimal control method we propose refers to the control strategy that maximizes the action value function (shown in Equation (5)) of the UAV. Aided by reinforcement learning, a model-free method is adopted, and a novel network is used to map the observation to optimal action in an end-to-end way. In addition, we use the tracking

and landing experience history to train the network to learn the optimal tracking strategy. Once the platform is tracked, a rule-based landing strategy is used to land the UAV.

3.1. Hybrid Strategy Method

The hybrid strategy adopted in this paper is shown in Figure 4. The strategy consists of two parts: tracking and landing modules. The tracking module introduces the reinforcement learning method to adjust the speed of the UAV in the horizontal direction, aiming to achieve the stable tracking of the moving platform. The landing module adjusts the height of the UAV in the vertical direction based on heuristic rules, so as to land the UAV on the platform.

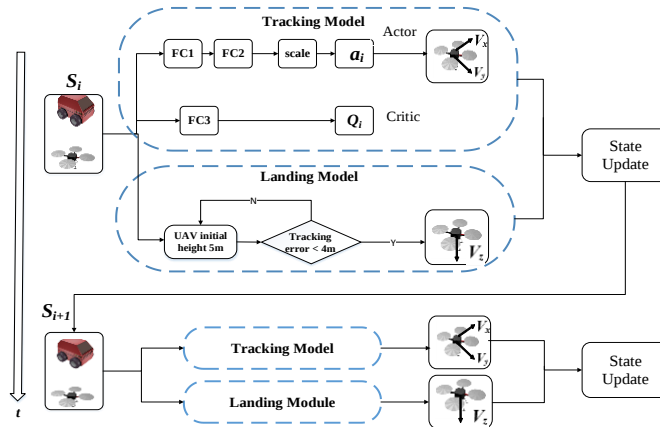


Figure 4. Hybrid strategy consisting of two parts: tracking and landing modules. The tracking module introduces the deep reinforcement learning method, and the landing module adopts heuristic rules, so as to land the UAV on the platform.

Tracking module: The UAV autonomous tracking and landing problem has a continuous state and decision space, and the DDPG, which combines the DQN and DPG, is a deterministic strategy for a continuous action space. This method combines reinforcement learning with deep learning and has good potential for dealing with complex tasks. Thus, this network is introduced in this paper for mapping the observation to proper action in an end-to-end way.

Details of the decision network structure are shown in Figure 4. The network adopts the actor-critic architecture [41], in which the actor network input is the system state, mainly including the motion state information of the UAV and moving platform in the system. The output layer is a two-dimensional continuous action space, which corresponds to the speed value of the UAV in the longitudinal and lateral directions after scale conversion. The critic network estimates the action-value function that describes the expected reward after following policy π . Since the decision is a deterministic action, to ensure that the environment is fully explored during the training process, we constructed a random action by adding noise sampled from a noise process N_i , in which the noise is only needed in the training process:

$$a_i = \mu(s_i|\theta^\mu) + N_i \quad (6)$$

where a_i is the output action, s_i represents the current state information, $\mu(s_i|\theta^\mu)$ represents the decision taken when the policy parameter is θ^μ in state s_i , and N_i is the artificially added Gaussian noise attenuated over time.

As shown in Figure 4, the network consists of three fully connected layers, the FC1 and FC3 layers are followed with the relu activation function, and the FC2 layer is followed with the tanh layer. The parameters of each layer in the network are shown in Table 1. Furthermore, the effects of different network parameters on the UAV tracking performance are shown in Appendix C.

Table 1. Network parameters.

Network Name	Number of Hidden Layers	Activation Function
FC1	30	relu
FC2	2	tanh
FC3	30	relu

Landing module: The landing module adjusts the height of the UAV in the vertical direction based on heuristic rules (shown in Table 1). As shown in Table 2, $dist$ has been defined in Equation (4), and $height$ is defined as $height = Z_u - Z_t$. According to the rules table, the speed of the UAV in the vertical direction depends on the distance and height between the UAV and the moving platform. When the distance between the UAV and moving platform is less than 4 m, the UAV should gradually reduce its height while ensuring stable target tracking. When the relative height between the UAV height and moving platform is less than 0.1 m, and the distance error of the horizontal direction is less than 0.8 m, it is then considered that the landing task is successful. When the target is lost during landing, the UAV would stop landing and gradually restore the initial height and re-plan the landing trajectory.

Table 2. The vertical speed generated by heuristic rules.

Velocity (m/s)		Dist		
		$0 < dist < 0.8$	$0.8 \leq dist \leq 4$	$dist > 4$
Height (m)	$0 \leq height \leq 0.1$	0	$0.5 \times height$	-1
	$0.1 < height \leq 3.5$	$0.5 \times height$	$0.5 \times height$	-1
	$ height > 3.5$	$0.5 \times height$	$0.5 \times height$	0

3.2. Network Model Training

To train the UAV tracking neural network to learn the optimal tracking strategy, we adopted the reinforcement learning process (shown in Figure 5). At each step, the UAV observes the states' partially observable information and then interacts with the environment through actions while receiving immediate reward signals. After multi-step decisions, the agent gains decision-making experience, so as to obtain more cumulative rewards and maximize the action-value function (defined in Section 2).

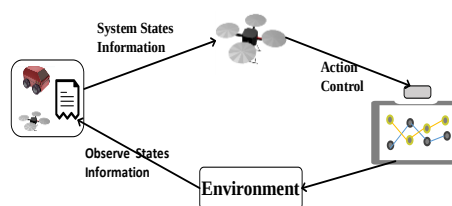


Figure 5. At each step, the UAV observes the environment and interacts with it through actions while receiving immediate signals and updating information of the environment. After multi-step decisions, the UAV acquires decision-making experience and then optimizes the entire task sequence.

One challenge when training a neural network is that there is a correlation between the data generated by sequential exploration in the UAV autonomous tracking and landing environment. To address these issues, a replay buffer is used to define a control experience tuple: $D_i = \{s_i, a_i, r_i, s_{i+1}\}$, indicating the UAV input state at time i , outputting the control action, receiving the reward, and obtaining the state at the next time $i + 1$. The tuple is stored in the replay buffer, and the neural network is updated by uniform random sampling of the mini-batch data in the replay buffer.

The decision network training method is now presented. The critic network uses the method of minimizing the loss function to approximate the value function, which is defined as

$$L(\theta^Q) = \frac{1}{N} \sum_i [Q(s_i, a_i | \theta^Q) - r_i - \gamma Q'(s_{i+1}, \mu'(s_{i+1} | \theta^{\mu'})) | \theta^{Q'}]^2 \quad (7)$$

where $L(\theta^Q)$ is the loss function, $Q(s_i, a_i | \theta^Q)$ is the estimate of the Q value at time i (the latter two values are the actual Q values after the action a_i at time i), γ is the discount factor, and r_i is an immediate reward.

In the actor network, the neural network is also used to approximate the strategy function, and the actor policy is updated to output the optimal decision on the basis of the current state. The updated formula is

$$\nabla J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) \Big|_{s=s_i, a=a_i} \nabla_{\theta^\mu} \mu(s | \theta^\mu) \Big|_{s=s_i} \quad (8)$$

where ∇J represents the gradient direction of the Q value caused by the strategy μ , thereby updating the policy parameter $\mu(s | \theta^\mu)$, $\nabla_a Q(s, a | \theta^Q) \Big|_{s=s_i, a=a_i}$ represents the change in the Q value generated by the action a_i in the current state, and $\nabla_{\theta^\mu} \mu(s | \theta^\mu)$ represents the current policy gradient direction.

To ensure the stability of the learning process, in this paper, the networks $\mu'(s | \theta^{\mu'})$ and $Q'(s, a | \theta^{Q'})$ are created to be the same as the actor and critic network, respectively, which are then used for calculating the target values, and the parameter-updated formula is

$$\begin{aligned} \theta_{i+1}^{Q'} &= \tau \theta^Q + (1 - \tau) \theta_i^{Q'} \\ \theta_{i+1}^{\mu'} &= \tau \theta^\mu + (1 - \tau) \theta_i^{\mu'} \end{aligned} \quad (9)$$

4. Simulation Results

4.1. Simulation Environment

To evaluate the behavior of the control system approach, experiments were performed on the Modular Open Robots Simulation Engine (MORSE) simulation platform (<https://github.com/morse-simulator/morse>), which can perform accurate dynamic simulations based on the state-of-the-art Bullet library. The simulation environment was established using a UAV and ground vehicle. The UAV speed was controlled by the algorithm proposed in this paper, and the orientation of the ground vehicle (robot) was able to be changed at any time from the command line. The experimental computer was an AMD Ryzen 71,700 (eight-core processor, main frequency 3.0 GHz, 8 GB DDR4, 2400 MHz memory, operating system Ubuntu 16.04) running Python v3.6.0, anaconda v4.2.0, and TensorFlow v1.4.0.

The flowchart of the simulation experiment for the proposed method is shown in Figure 6. The flowchart defines two different types of tests. The first was to test the performance of the UAV tracking model by tracking moving platform with a fixed height of 5 m. Once the tracking error was less than a certain threshold, a second experiment could be carried out to evaluate the ability of the hybrid strategy method to automatically land from a height of 5 m. Both tracking and landing tests have the same environment: The tracking and landing phases were trained in simulation throughout approximately 60 k training steps. The trajectory of the moving platform could be linear, circular, or random. The maximum velocity of UAV and moving platform were defined in Section 2. The permitted horizontal area for the moving platform to move is a rectangle of 37 m $\times$ 85 m. The velocity controller frequency of UAV is 20 Hz in a real-time MORSE simulation.

In the simulation environment, in order to ensure the flight safety of the UAV, on the one hand, during the model training phase, we restricted the speed data output by the model, and stipulated that the horizontal velocity of the UAV v_x, v_y was less than 10 m/s, and v_z was less than 3 m/s. On the other hand, we introduced heuristic rules to prevent the UAVs from crashing and keep them at a safe flying height, specifically, when the height of the drone was less than 3.5 m, $v_z = 0.5 \times \text{height}$. When the

distance between the UAV and the moving platform was greater than 6 m, the flying height of the UAV must gradually return to the initial height of 5 m.

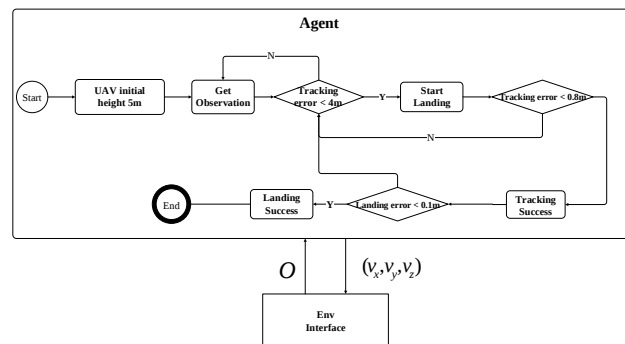


Figure 6. The simulation experiment includes two parts: the tracking and landing experiment. Once the UAV can track the moving target stably, it will adjust the altitude and land on the moving platform.

4.2. Autonomous Tracking Tests

In the autonomous tracking tests, the UAV and ground vehicle were randomly placed at an initial position. The behavior of the control system was evaluated using the root-mean-square error (RMSE) and tracking success rate (TSR) during the tracking tests. In Equation (10), the tracking success is defined as the horizontal distance between the UAV and moving platform, which is less than 3 m, and the definition of TSR is shown as follows:

$$TSR = \sum_{i=0}^N \frac{D_i}{N} \times 100\%, D_i = \begin{cases} 1, dist < 3 \\ 0, dist \geq 3 \end{cases} \quad (10)$$

where $D_i = 1$ refers to the success of the UAV tracking moving platform at step i , $D_i = 0$ refers to tracking failure, and N refers to stopping the tracking test after updating fixed N time steps.

Table 3 shows the RMSE and TSR values of the tracking modules compared with the PID controllers (a complete description of PID controllers and parameters are shown in Appendix A). The moving platform has four different types of motion trajectories, including linear, circular, and random motion. The results show that, although the tracking performance of the PID method is more stable, when encountering complex motion, the RMSE of the proposed method is smaller. Compared with the VIEW percent indicator, the PID method is more likely to cause the loss of the tracking target. Figure 7 shows the trajectory of the UAV in the x-y plane while following a moving platform. Figure 7c,d show that the PID method could fit the motion trajectory of the platform, but it has a large response hysteresis and response error; therefore, compared with the PID method, the proposed method is more effective in solving the problem of tracking large-scale changes in moving platforms.

Table 3. Root-mean-square error and tracking success rate (TSR) of the unmanned aerial vehicle (UAV) for tracking a moving platform.

Controller	PID Method					Proposed Method			Units
Movement	Linear	Circular	Random1	Random2	line	Circular	Random1	Random2	
x axis (RMSE)	0.2969	2.4204	3.3479	3.7198	0.3896	1.7466	2.8444	2.6720	m
y axis (RMSE)	0.8809	1.9919	2.7271	3.6738	1.1834	2.0197	1.9053	2.0395	m
TSR (%)	95.0	65.5	27.5	20.3	94.8	63.8	45.3	37.2	–

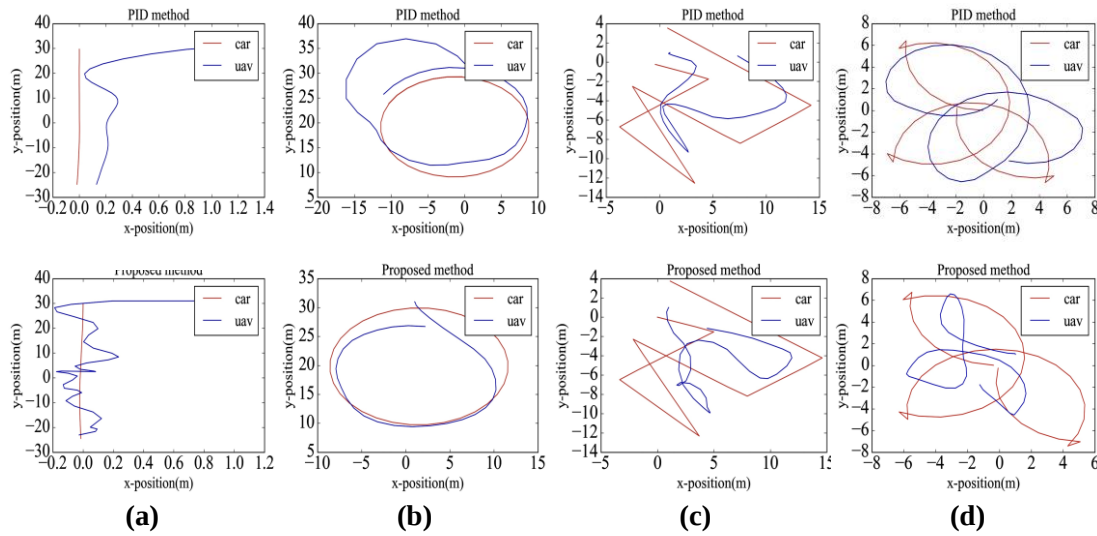


Figure 7. Trajectory of UAV while tracking a moving platform: moving platform for (a) linear motion, (b) circular motion, (c) random motion, and (d) random motion.

In order to test the capability of the generalization and robustness of the proposed method, experiments were conducted to analyze the effects of sensor noise on the tracking model. These experiments simulated the measurement noise of the sensors, which limits the ability to acquire the ground-truth motion information of the UAV and moving platforms. We added Gaussian noise ($\mu = 0$ and $\sigma = 1, 2, 3, 4$) to observations obtained from the UAV, and the results of the noise influence are shown in Figure 8, and the confidence intervals of the data points are specified in Appendix B. Both models degrade in performance as more noise is applied to the UAV observation. However, the performance of the proposed model does not degrade as quickly as that of the PID algorithms and even performed tracking tasks better than the PID algorithms.

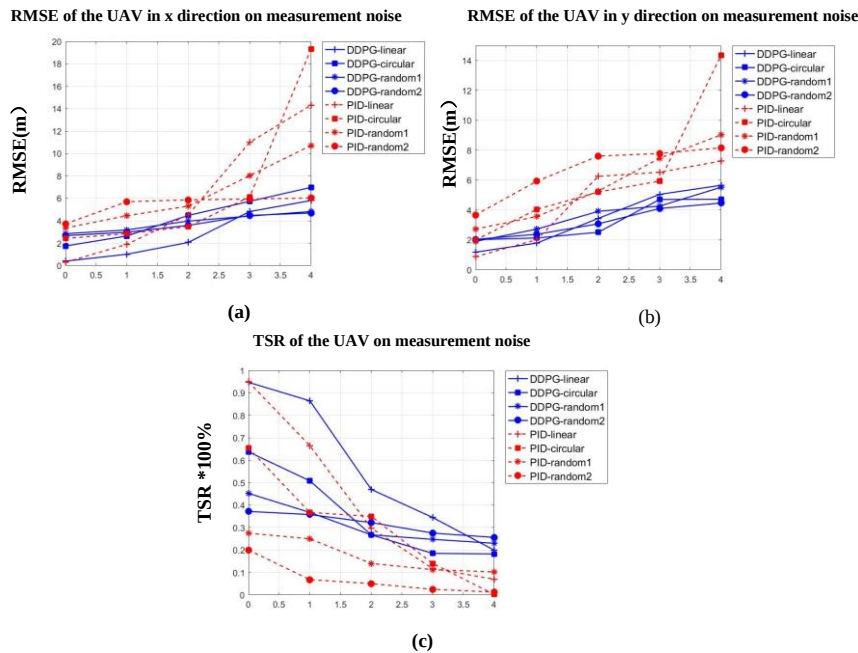


Figure 8. Tracking results on measurements error: (a) root-mean-square error (RMSE) of UAV in the x-direction, (b) RMSE of UAV in the y-direction, and (c) tracking success rate (TSR) of UAV.

On the other hand, we tested the effects of intermittent measurements on the tracking method. In the process of the UAV tracking the moving platform, the initial state of the system is S_0 , and then the system needs to update the state at a fixed frequency and stop the tracking process after updating the fixed N steps (arriving final state S_N). Therefore, we set time steps L during which the observations were not updated. Furthermore, the random starting interval S_i of $[S_0, S_{N-L}]$ was generated by a random number generation algorithm, namely the linear congruential method [42], and then the intermittent measurements interval $[S_i, S_{i+L})$ was obtained. This simulated the incomplete information acquired by the sensors—the tracking results for the loss of motion information in an information acquired by the sensors. The tracking results for the loss of motion information in an intermittent measurement environment are shown in Table 4. The results show that, in most cases, the performance of the proposed method is better than that of the PID algorithms.

Table 4. Tracking results in an intermittent measurement environment. Root-mean-square error and percentage of steps of the moving platform in the field of view of the UAV are shown in this table.

Controller Movement	Values	PID Method			Proposed Method		
		x axis (m) (RMSE)	y axis (m) (RMSE)	TSR (%)	x axis (m) (RMSE)	y axis (m) (RMSE)	TSR (100%)
linear		0.3991	2.4481	78.60	0.3764	2.0297	90.50
circular		6.7256	8.2300	15.50	2.1986	2.5640	59.75
random1		10.3374	10.1190	15.00	3.1644	2.3531	43.00
random2		6.2291	6.0182	12.00	2.6294	3.1471	35.00

The results of these three tracking experiments indicate that our algorithms are outperformed by PID controllers in some conditions and are more tolerant of noisy and intermittent measurements than PID method.

4.3. Autonomous Landing Tests

Once the tracking methods and landing methods are obtained, they can be used together to control the autonomous landing of the virtual UAV. The moving platform changes its position in the x - y plane to achieve the four different movement types. The initial position of the UAV is an altitude of 5.0 m with the vertical velocity controller constrained by the x and y axis errors, and this strategy retains the UAV's ability to carry out autonomous tracking.

To evaluate the UAV landing method proposed in this paper, we introduced the landing success rate as the evaluation indicator of the landing algorithm. A landing trial is considered successful when the horizontal and vertical distance between the UAV and the moving platform are less than the fixed threshold.

Table 5 shows the success rate based on 100 times landing test episodes, compared with the PID algorithm, the proposed algorithm reduces the success rate by 2~3% when the moving platform performs a linear or circular motion, however, the landing success rate of the proposed algorithm is significantly improved, such as when the moving platform performs random motion. In addition, Figure 9 shows the trajectory of the UAV in the x - y - z plane while landing on a moving platform. The success rate and landing trajectory suggest that the PID control could not explore the unknown nonlinearity in its control architecture and thus, usually leads to a suboptimal performance, but the proposed approach succeeded in learning the landing maneuver in complex environments.

Table 5. Landing success rate for several movement types.

Movement Type	Linear	Circular	Random1	Random2
PID method	97%	85%	56%	41%
Proposed method	94%	83%	70%	74%

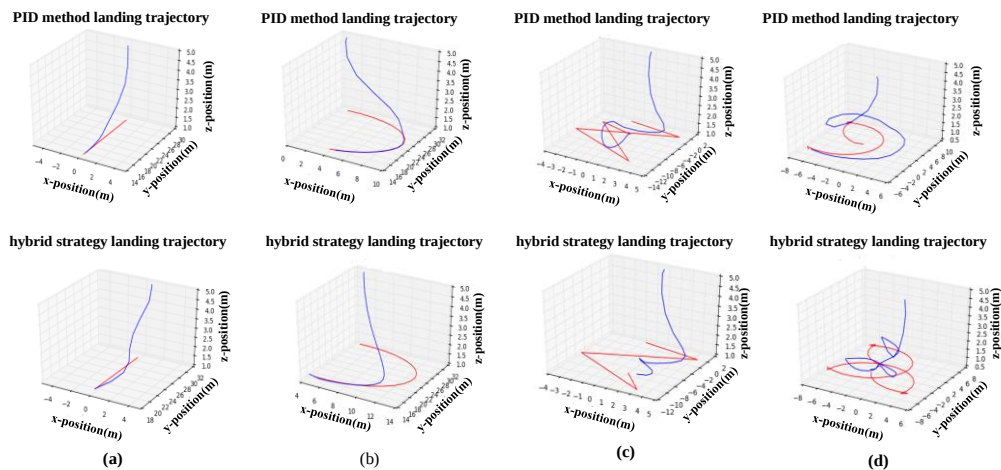


Figure 9. Trajectory of UAV while landing on a moving platform for (a) linear motion, (b) circular motion, (c) random motion, and (d) random motion.

Additionally, in order to further validate the performance of our proposed model, we compare the landing success rate of the proposed method with other deep reinforcement learning methods. The results of the test phase are summarized in Figure 10. The bar chart compares the performances of the DQN-single [30], DQN-multi [30], DDPG (two-dimensional) [5,32], and the proposed method. In all tests, the moving platform performed a uniform straight-line motion at a speed of 3 m/s. In this case, the proposed method has the highest success rate with an accuracy of 93%, the DDPG (two-dimensional) has the similar performance (89%). The DQN-single follows with a score of (85%). The DQN-multi score is significantly lower (80%). The results show that the method proposed in this paper can more effectively solve the problem of a UAV's autonomous landing in high-dimensional and continuous action space.

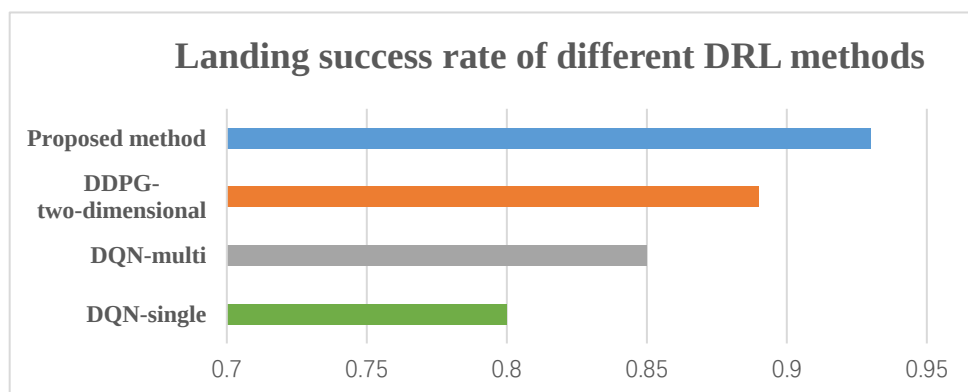


Figure 10. The results of landing simulation with different deep reinforcement learning (DRL) methods. The moving platform performed a uniform straight-line motion at a speed of 3 m/s.

5. Conclusions and Future Work

A hybrid strategy-based autonomous tracking and landing method is proposed. With the help of reinforcement learning and an end-to-end neural network, the proposed method requires no prior information of the moving platform and the UAV can work well in a noisy measurement and intermittent measurements. Compared with the PID methods, the proposed method shows good performance when the platform moves in a complex trajectory. Considering the control strategy of the UAV landing (or other types of robot complex tasks) in unpredictable and harsh environments, other reinforcement learning algorithms can also be applied to our neural network training process, which improves the generality of our control framework. The developed algorithm also has some

limitations. For example, due to the complexity and continuity of the state and action space of the UAV landing problem, the z-axis is based on heuristic rules and is not included in the solution space of reinforcement learning. Therefore, we would consider including the z-axis in the action space in future work. At the same time, we would use the dji-m210 UAV to further verify the robustness and effectiveness of the proposed method, and some offline training controller and online optimizer solutions would be tested.

Author Contributions: Conceptualization, J.X. and H.W.; methodology, X.P.; software, X.Z.; validation, H.W. and W.N.; formal analysis, J.X.; investigation, J.X. and X.P.; resources, X.P.; data curation, X.Z.; writing—original draft preparation, J.X. and H.W.; writing—review and editing, X.P.; visualization, W.N.; supervision, X.P.; project administration, X.Z.; funding acquisition, W.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Key Research Program of Frontier Sciences, Chinese Academy of Sciences.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Combined with the velocity control method of the UAV in the MORSE platform, the PID control law of velocity can be given as follows:

$$V = k_I \cdot \Delta P + k_P \cdot \Delta V + k_D \cdot \Delta a \quad (\text{A1})$$

where V is the velocity output, and ΔP , ΔV , Δa , respectively, correspond to the position difference, velocity difference, and acceleration difference between the UAV and the moving platform. K_I is the integral term parameter, K_P is the proportional term parameter, and K_D is the differential term parameter. Parameter setting is the key step of the PID controller design, and the PID parameters during the UAV's moving tracking and landing are displayed in Table A1. In addition, in this paper, we used the critical proportion method and engineering experience to adjust the PID parameters.

Table A1. Parameters of the Proportional-Integral-Derivative (PID) controller.

Movement Type	Linear	Circular	Random1	Random2
K_P	5.0	8.0	2.5	4.0
K_D	3.0	3.0	5.0	3.5
K_I	3.5	3.5	0.3	1.0

Appendix B

This paper considers that the RMSE and TSR data obtained by measurement and calculation obey a normal distribution, and the confidence probability of the data is 99.7%. The confidence interval of the data obtained in the sensor noise environment is shown in Tables A2–A4.

Table A2. Tracking results of UAV in the sensor noise environment. The root-mean-square error in the x -axis direction and data confidence interval are shown in this table.

Movement Type	Noise Method	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$
linear	Proposed Method	0.39 ± 0.21	1.00 ± 1.34	2.06 ± 1.96	4.81 ± 2.56	5.80 ± 3.41
	PID Method	0.30 ± 0.32	1.89 ± 1.23	4.52 ± 2.19	10.99 ± 3.13	14.31 ± 5.73
circular	Proposed Method	1.75 ± 0.13	2.63 ± 0.87	4.47 ± 1.95	5.73 ± 2.14	6.98 ± 3.43
	PID Method	2.42 ± 0.31	2.84 ± 1.53	3.47 ± 1.87	6.12 ± 2.06	19.32 ± 4.54
random1	Proposed Method	2.84 ± 0.17	3.17 ± 1.68	3.97 ± 1.93	4.41 ± 2.27	4.82 ± 2.96
	PID Method	3.35 ± 0.27	4.44 ± 1.54	5.30 ± 2.01	8.02 ± 3.84	10.71 ± 4.25
random2	Proposed Method	2.67 ± 0.01	2.97 ± 0.99	3.58 ± 1.95	4.47 ± 2.15	4.69 ± 2.73
	PID Method	3.71 ± 0.53	5.70 ± 1.36	1.86 ± 2.81	5.93 ± 3.72	6.02 ± 4.06

Table A3. Tracking results of UAV in the sensor noise environment. The root-mean-square error in the y -axis direction and data confidence interval are shown in this table.

Movement Type	Noise Method	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$
linear	Proposed Method	1.18 ± 0.17	1.78 ± 1.45	3.44 ± 2.23	5.04 ± 2.47	5.65 ± 3.75
	PID Method	0.88 ± 0.38	2.01 ± 1.65	6.24 ± 2.47	6.52 ± 2.65	7.27 ± 3.06
circular	Proposed Method	2.02 ± 0.22	2.12 ± 1.06	2.51 ± 1.53	4.71 ± 1.97	4.72 ± 3.16
	PID Method	2.00 ± 0.17	4.04 ± 1.47	5.20 ± 1.99	5.93 ± 2.15	14.34 ± 3.69
random1	Proposed Method	1.95 ± 0.13	2.72 ± 1.42	3.91 ± 1.87	4.26 ± 2.43	5.52 ± 3.06
	PID Method	2.72 ± 0.25	3.56 ± 1.56	5.26 ± 2.45	7.47 ± 3.24	9.02 ± 3.99
random2	Proposed Method	2.04 ± 0.02	2.38 ± 1.09	3.08 ± 1.90	4.10 ± 2.07	4.47 ± 2.65
	PID Method	3.67 ± 0.65	5.93 ± 1.10	7.60 ± 2.07	7.78 ± 3.86	8.16 ± 4.67

Table A4. Tracking results of UAV in the sensor noise environment. The TSR and data confidence interval are shown in this table.

Movement Type	Noise Method	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$
linear	Proposed Method	94.8 ± 2.5	86.5 ± 6.5	47.08 ± 7.6	34.5 ± 8.5	20.0 ± 4.3
	PID Method	63.8 ± 2.5	50.8 ± 2.6	29.8 ± 4.2	11.8 ± 1.7	7.0 ± 4.1
circular	Proposed Method	63.8 ± 1.5	50.8 ± 2.1	26.8 ± 4.3	18.5 ± 2.3	18.3 ± 4.2
	PID Method	65.5 ± 0.2	36.8 ± 2.3	35.0 ± 2.9	14.0 ± 10.5	0.25 ± 0.01
random1	Proposed Method	45.3 ± 1.2	36.8 ± 2.5	26.8 ± 3.6	24.8 ± 4.3	23.0 ± 5.2
	PID Method	27.5 ± 0.2	25.0 ± 3.5	14.0 ± 4.8	11.3 ± 6.3	10.3 ± 6.4
random2	Proposed Method	37.2 ± 1.8	35.8 ± 2.4	32.2 ± 3.5	27.6 ± 3.7	25.6 ± 4.3
	PID Method	20.3 ± 0.1	6.8 ± 0.1	5.0 ± 0.1	2.5 ± 0.13	1.25 ± 0.2

Appendix C

In this paper, the number of neurons is one of the important parameters that affect the UAV tracking performance. Therefore, we analyze the model sensitivity of the proposed method from the perspective of the neuron parameter settings. All tracking tests have the same environment: the tracking phases were trained in simulation throughout approximately 60 k training steps. Table A5 shows the RMSE and TSR values of the tracking module with different neural parameters, from which we can see that the result is acceptable, and the performance of the models are better than that of the PID method.

Table A5. Tracking results of UAV in the sensor noise environment with different neural parameters. The RMSE and TSR are shown in this table.

Network Parameters	Noise		$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$
	Method						
Cells (FC1) = 36 Cells (FC3) = 36	x-axis		2.83 $\pm$ 0.17	3.43 $\pm$ 1.57	4.18 $\pm$ 1.98	4.78 $\pm$ 3.99	4.08 $\pm$ 3.74
	y-axis		1.97 $\pm$ 0.12	3.14 $\pm$ 1.89	3.56 $\pm$ 3.41	4.08 $\pm$ 3.13	4.95 $\pm$ 3.74
	TSR		40.0 $\pm$ 1.6	34.6 $\pm$ 2.7	22.2 $\pm$ 3.9	18.0 $\pm$ 4.2	16.8 $\pm$ 2.7
Cells (FC1) = 30 Cells (FC3) = 30	x-axis		2.67 $\pm$ 0.01	2.97 $\pm$ 0.99	3.58 $\pm$ 1.95	4.47 $\pm$ 2.15	4.69 $\pm$ 2.73
	y-axis		2.04 $\pm$ 0.02	2.38 $\pm$ 1.09	3.08 $\pm$ 1.90	4.10 $\pm$ 2.07	4.47 $\pm$ 2.65
	TSR		37.2 $\pm$ 1.8	35.8 $\pm$ 2.4	32.2 $\pm$ 3.5	27.6 $\pm$ 3.7	25.6 $\pm$ 4.3
Cells (FC1) = 24 Cells (FC3) = 24	x-axis		2.04 $\pm$ 0.06	3.11 $\pm$ 1.59	4.06 $\pm$ 1.63	5.02 $\pm$ 3.72	5.41 $\pm$ 4.18
	y-axis		3.31 $\pm$ 0.09	4.05 $\pm$ 1.15	5.22 $\pm$ 2.80	5.43 $\pm$ 1.92	6.07 $\pm$ 3.48
	TSR		37.5 $\pm$ 1.7	22.8 $\pm$ 2.4	12.4 $\pm$ 1.5	9.4 $\pm$ 2.3	7.2 $\pm$ 0.7

References

- Kim, J.; Jung, Y.; Lee, D.; Shim, D.H. Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera. In Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 1243–1252.
- Li, K.; Liu, P.; Pang, T.; Yang, Z.; Chen, B.M. Development of an unmanned aerial vehicle for rooftop landing and surveillance. In Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 832–838.
- Baca, T.; Stepan, P.; Spurny, V.; Hert, D.; Penicka, R.; Saska, M.; Thomas, J.; Loianno, G.; Kumar, V. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *J. Field Robot.* **2019**, *36*, 874–891. [\[CrossRef\]](#)
- Araar, O.; Aouf, N.; Vitanov, I. Vision based autonomous landing of multirotor UAV on moving platform. *J. Intell. Robot. Syst.* **2017**, *85*, 369–384. [\[CrossRef\]](#)
- Borowczyk, A.; Nguyen, D.-T.; Nguyen, A.P.; Nguyen, D.Q.; Saussié, D.; Le Ny, J. Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle. *IFAC PapersOnLine* **2017**, *50*, 10488–10494. [\[CrossRef\]](#)
- Xu, Y.; Liu, Z.; Wang, X. Monocular Vision based Autonomous Landing of Quadrotor through Deep Reinforcement Learning. In Proceedings of the 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 10014–10019.
- Mo, H.; Farid, G. Nonlinear and adaptive intelligent control techniques for quadrotor uav—A survey. *Asian J. Control.* **2019**, *21*, 989–1008. [\[CrossRef\]](#)
- Yang, T.; Ren, Q.; Zhang, F.; Xie, B.; Ren, H.; Li, J.; Zhang, Y. Hybrid camera array-based uav auto-landing on moving ugv in gps-denied environment. *Remote Sens.* **2018**, *10*, 1829. [\[CrossRef\]](#)
- Yang, B.; Dutta, P. Cooperative Navigation for Small UAVs in GPS-Intermittent Environments. In Proceedings of the AIAA Aviation 2019 Forum, Dallas, TX, USA, 17–21 June 2019; p. 3515.
- Van den Meijdenberg, J.; Totu, L.; Schioler, H.; Leth, J. Stochastic Controller Design for multi-rotor UAV under Intermittent Localization. In Proceedings of the IEEE Australian & New Zealand Control Conference (ANZCC), Melbourne, VIC, Australia, 6–8 December 2018; pp. 56–61.
- Arora, S.; Jain, S.; Scherer, S.; Nuske, S.; Chamberlain, L.; Singh, S. Infrastructure-free ship deck tracking for autonomous landing. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 323–330.
- Skoczylas, M. Vision analysis system for autonomous landing of micro drone. *Actamech. Automat.* **2014**, *8*, 199–203. [\[CrossRef\]](#)
- Rucco, A.; Sujit, P.; Aguiar, A.P.; De Sousa, J.B.; Pereira, F.L. Optimal rendezvous trajectory for unmanned aerial-ground vehicles. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 834–847. [\[CrossRef\]](#)
- Vlantis, P.; Marantos, P.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Quadrotor Landing on an Inclined Platform of a Moving Ground Vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2202–2207.
- Wenzel, K.E.; Masselli, A.; Zell, A. Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. *J. Intell. Robot. Syst.* **2011**, *61*, 221–238. [\[CrossRef\]](#)

16. Talha, M.; Asghar, F.; Rohan, A.; Rabah, M.; Kim, S.H. Fuzzy Logic-Based Robust and Autonomous Safe Landing for UAV Quadcopter. *Arab. J. Sci. Eng.* **2019**, *44*, 2627–2639. [[CrossRef](#)]
17. Prach, A.; Gürsoy, G.; Yavrucuk, L. Nonlinear Controller for a Fixed-Wing Aircraft Landing. In Proceedings of the IEEE American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 2897–2902.
18. Rossi, E.; Bruschetta, M.; Carli, R.; Chen, Y.; Farina, M. Online nonlinear model predictive control for tethered uavs to perform a safe and constrained maneuver. In Proceedings of the IEEE European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 3996–4001.
19. Beul, M.; Houben, S.; Nieuwenhuisen, M.; Behnke, S. Fast autonomous landing on a moving target at MBZIRC. In Proceedings of the European Conference on Mobile Robots, Paris, France, 6–8 September 2017; pp. 1–6.
20. Kamath, A.K.; Tripathi, V.K.; Yogi, S.C.; Behera, L. Vision-based Fast-terminal Sliding Mode Super Twisting Controller for Autonomous Landing of a Quadrotor on a Static Platform. In Proceedings of the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, 14–18 October 2019; pp. 1–6.
21. Huang, Y.; Zheng, Z.; Sun, L.; Zhu, M. Saturated adaptive sliding mode control for autonomous vessel landing of a quadrotor. *IET Control. Theor. Appl.* **2018**, *12*, 1830–1842. [[CrossRef](#)]
22. Fei, Q.; Zhang, J.; Wang, Z.; Huang, X. Sliding Mode Control with Uncertain Model for a Quadrotor UAV's Automatic Visual Landing Problem. In *Proceedings of the Chinese Intelligent Automation Conference*; Springer: Cham, Switzerland, 2019; pp. 226–233.
23. He, S.; Wang, H.; Zhang, S. Vision Based Autonomous Landing of the Quadrotor Using Fuzzy Logic Control. In Proceedings of the IEEE Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 1943–1948.
24. Rabah, M.; Rohan, A.; Mohamed, S.A.; Kim, S.-H. Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-PI. *IEEE Access* **2019**, *7*, 38407–38419. [[CrossRef](#)]
25. De Souza, J.P.C.; Marcato, A.L.M.; De Aguiar, E.P.; Jucá, M.A.; Teixeira, A.M. Autonomous landing of UAV based on artificial neural network supervised by fuzzy logic. *J. Control Automat. Electr. Syst.* **2019**, *30*, 522–531. [[CrossRef](#)]
26. Ananthakrishnan, U.; Akshay, N.; Manikutty, G.; Bhavani, R.R. Control of Quadrotors Using Neural Networks for Precise Landing Maneuvers. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*; Springer: Singapore, 2017; pp. 103–113.
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Hiedmiller, M.; Fiedjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
28. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Hiedmiller, M. Deterministic Policy Gradient Algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
29. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
30. Polvara, R.; Patacchiola, M.; Sharma, S.; Wan, J.; Manning, A.; Sutton, R.; Cangelosi, A. Toward End-to-End Control for UAV Autonomous Landing via Deep Reinforcement Learning. In Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 115–123.
31. James, S.; Wohlhart, P.; Kalakrishnan, M.; Kalashnikov, D.; Irpan, A.; Ibarz, J.; Levine, S.; Hadsell, R.; Bousmalis, K. Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping via Randomized-to-Canonical Adaptation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 16–20 June 2019; pp. 12627–12637.
32. Rodriguez-Ramos, A.; Sampeder, C.; Bavle, H.; De la Puente, P.; Campoy, P. A deep reinforcement learning strategy for UAV autonomous landing on a moving platform. *J. Intell. Robot. Syst.* **2019**, *93*, 351–366. [[CrossRef](#)]
33. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.

34. Falanga, D.; Zanchettin, A.; Simovic, A.; Delmerico, J.; Scaramuzza, D. Vision-based autonomous quadrotor landing on a moving platform. In Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics, Shanghai, China, 11–13 October 2017; pp. 200–207.
35. Sondik, E.J. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Oper. Res.* **1978**, *26*, 282–304. [[CrossRef](#)]
36. Gautam, A.; Sujit, P.; Saripalli, S. A survey of autonomous landing techniques for UAVs. In Proceedings of the IEEE 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 1210–1218.
37. Puterman, M.L. Markov decision processes: Discrete stochastic dynamic programming. *J. Oper. Res. Soc.* **1995**, *46*, 792.
38. Tan, L.; Wu, J.; Yang, X.; Song, S. Research on Optimal Landing Trajectory Planning Method between an UAV and a Moving Vessel. *Appl. Sci.* **2019**, *9*, 3708. [[CrossRef](#)]
39. Forsmo, E.J. Optimal Path Planning for Unmanned Aerial Systems. Master's Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2012.
40. Kirk, D.E. *Optimal Control Theory: An Introduction*; Courier Corporation; Dover Publications: Mineola, NY, USA, 2004.
41. Degris, T.; Pilarski, P.M.; Sutton, R.S. Model-Free Reinforcement Learning with Continuous Action in Practice. In Proceedings of the IEEE American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; pp. 217–218.
42. Von Neumann, J. Various techniques used in connection with random digits. *Appl. Math. Ser.* **1951**, *12*, 36–38.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).