A pretrained proximal policy optimization algorithm with reward shaping for aircraft guidance to a moving destination in three-dimensional continuous space

Zhuang Wang¹, Hui Li¹,², Zhaoxin Wu² and Haolin Wu¹

Abstract
To enhance the performance of guiding an aircraft to a moving destination in a certain direction in three-dimensional continuous space, it is essential to develop an efficient intelligent algorithm. In this article, a pretrained proximal policy optimization (PPO) with reward shaping algorithm, which does not require an accurate model, is proposed to solve the guidance problem of manned aircraft and unmanned aerial vehicles. Continuous action reward function and position reward function are presented, by which the training speed is increased and the performance of the generated trajectory is improved. Using pretrained PPO, a new agent can be trained efficiently for a new task. A reinforcement learning framework is built, in which an agent can be trained to generate a reference trajectory or a series of guidance instructions. General simulation results show that the proposed method can significantly improve the training efficiency and trajectory performance. The carrier-based aircraft approach simulation is carried out to prove the application value of the proposed approach.

Keywords
Aircraft guidance, deep reinforcement learning, PPO, reward shaping

Date received: 26 March 2020; accepted: 28 December 2020

Introduction
The guidance of aircraft, including manned aircraft and unmanned aerial vehicles (UAVs), has become the research focus of scholars since it is widely utilized in realistic situations. For example, it can be applied for the guidance of carrier-based aircraft in the process of approach and landing for the maneuvering guidance of aircraft in air combat to reach the position of advantage, for aerial refueling guidance of UAVs, and so on. The guidance method is used to generate a trajectory or a set of instructions to guide a manned aircraft or a UAV to a moving destination in a certain direction in three-dimensional continuous space.

With the development of command and control technologies, various guidance methods have been proposed in aircraft guidance research. A visual/inertial integrated carrier landing guidance algorithm is presented,¹ which can
obtain satisfactory accuracy results with less execution time. Wang et al.\(^2\) establish a pilot behavior model for carrier landing operations based on realistic mechanisms and strategies. There are also some autonomous landing techniques for UAVs.\(^3\) In air-to-air combat research community, many algorithms focus on automating maneuvering guidance to control UAVs or reduce the workload of pilots.\(^4\)–\(^6\) Toubman et al.\(^4\) adopt rule-based dynamic scripting in air combat, which requires hard coding the air combat tactics into a maneuver selection algorithm. Eklund et al.\(^5\) present a nonlinear, online model predictive controller for pursuit and evasion of two fixed-wing autonomous aircraft, which requires a reliance on previous knowledge of the maneuverers. A virtual pursuit point-based combat maneuvering guidance law for an unmanned combat aerial vehicle is presented and is used in X-plane based nonlinear six-degrees-of-freedom air combat simulation.\(^6\) The methods of trajectory optimization for UAV refueling in the air\(^7\) and using a ground mobile refueling unit\(^8\) are proposed.

These classical methods used in previous studies can accomplish the tasks for which they are programmed and require accurate models. However, in realistic situations, the accurate models usually do not exist or can be manually designed by human experts only. In this context, reinforcement learning (RL)\(^9\) has proven to increasingly overcome this limitation and generated high expectations in the research community. RL is a type of artificial intelligence, which allows agents to learn directly from the environment through trial and error without a perfect knowledge of environment in advance. A well-trained RL agent can automatically determine the optimal or suboptimal behavior within a specific context to maximize its performance and it needs less computational time. Deep reinforcement learning (DRL) is a particular type of RL with deep neural networks for continuous state representation. Great achievements have been made in using the advanced DRL algorithms, such as deep Q-network (DQN),\(^10\) deep deterministic policy gradient (DDPG),\(^11\) and proximal policy optimization (PPO).\(^12\) Mnih et al. introduce DQN and ignite the field of DRL. DQN uses experience replay and target networks to address instability issues and achieves outstanding results over many discrete Atari games. DDPG is an actor-critic, model-free algorithm, which is an extension of DQN and DPG.\(^13\) It avoids the optimization of action-value function at every time and is feasible in complex continuous action spaces. PPO is proposed to benefit the stability and reliability from trust region policy optimization (TRPO),\(^14\) and it is much simpler to implement, more general, and has better sample complexity. It achieves good performance on several continuous tasks and on discrete Atari games. DRL has been utilized in many decision-making fields. For example, it has been used in playing board games,\(^15\) video games,\(^10\) and robot control,\(^16,17\) and obtained great achievements of human-level or super-human performance.

Some RL methods have been proposed to solve aircraft guidance problems.\(^18\)–\(^22\) An RL agent is developed for guiding a powered UAV from one thermal location to another quickly and efficiently by controlling bank angle\(^18\) in the x-y plane. To solve guidance problem with many conflicting objectives, a voting Q-learning algorithm is proposed.\(^19\) A model-based path-finding method is proposed using Q-learning to show the efficiency and reliability of the training methodology in solving guidance problems.\(^20\) Rodriguez-Ramos et al.\(^21\) propose a DRL strategy for UAV autonomous landing on a moving platform. However, the orientation of the landing platform is not considered, and the vertical velocity control is not included in the action set. A DQN algorithm is used to generate a trajectory to perform a perched landing on the ground.\(^22\) In this DQN algorithm, noise is added to the numerical model of airspeed in the training environment, which is more in line with the actual scenario.

Previous research have shown the benefits of using RL to solve guidance problems while some limitations still exist. First, because of curse of dimensionality, RL is difficult to converge in three-dimensional continuous space, and the quality of generated trajectory does not meet the realistic requirement. Non-RL method is used to achieve the control of altitude\(^18,21\) or trajectory smoothing.\(^23\) Second, previous research did not consider the direction of the aircraft when it arrived at the destination, but only the location.\(^18,21\) which could not completely solve the guidance problem. Last, current environments are mainly for video games\(^24\) or robot control,\(^25,26\) which do not have the ability to train a guidance agent. Previous studies did not have frameworks for solving guidance problems, either developing RL environment based on existing ones\(^21\) or using a two-dimensional grid-world,\(^19,20\) which are discrete and do not have the ability to solve three-dimensional problem.

Reward shaping\(^27\) is usually used to modify reward function to facilitate learning while maintaining optimal policy, which is a manual endeavor. It is widely adopted in RL community and also used in aircraft planning and control. Toubman et al.\(^28\) propose a reward function to remedy the false rewards and punishments for firing air combat missiles, which allow computer-generated forces to generate more intelligent behavior. Tummer and Aggino\(^29\) propose difference reward functions in a multi-agent air traffic system and show that agents can manage effective route selection and significantly reduce congestion. Two types of reward functions are developed to solve ground holding and air holding problems,\(^30\) which assist air traffic controllers in maintaining high standard of safety and fairness between airlines.

The main contributions of this article are listed in the following:

1. A novel reward function is proposed to improve the performance of the generated trajectories and the training efficiency.
2. A pretrained PPO algorithm is presented for different kinds of moving destination guidance tasks.
Using this algorithm, an agent that will be used in a new task can be trained quickly based on the existing agent.

3. An RL framework is built, in which an agent can be trained to guide a manned aircraft or to control a UAV to a moving destination in a certain direction in three-dimensional continuous state space.

This article is organized as follows. The “Problem formulation” section introduces the guidance problem under consideration and shows how to formulate this problem as a DRL task. The proposed reward shaping method and pre-trained PPO algorithm to solve this task are presented in “training optimization” section. The general simulation and the carrier-based aircraft approach simulation of the proposed method are given in “Simulation evaluation” section. The final section concludes the article.

**Problem formulation**

In this section, the problem of aircraft guidance in three-dimensional continuous space is introduced, and the RL-based algorithm and training framework are proposed to solve this problem.

**Problem statement**

The objective of aircraft guidance is to guide a manned aircraft or control a UAV from its current position \((x^f, y^f, z^f, \chi^f)\) to the target position \((x^t, y^t, z^t, \chi^t)\). The superscript \(a\) and \(o\) represent the aircraft and its destination, respectively. The subscript \(t\) and \(f\) represent the current time and the final time, respectively. The kinematic model of the aircraft is shown in Figure 1, and the kinematic equations of the aircraft\(^{31}\) are given by

\[
\begin{align*}
\dot{x} &= v \cos \phi \cos \chi \\
\dot{y} &= v \cos \phi \sin \chi \\
\dot{z} &= v \sin \phi
\end{align*}
\]

where \((x, y, z)\) is the three-dimensional coordinates of the aircraft, \(v, \phi, \) and \(\chi\) are velocity, flight path angle, and heading angle of the aircraft, respectively. The dynamic equations of the aircraft are defined as

\[
\begin{align*}
\dot{v} &= \frac{1}{m} \left( \eta T_{\text{max}} \cos \alpha - D \right) - g \sin \phi \\
\dot{\phi} &= \frac{1}{mv} \left( (L + \eta T_{\text{max}} \sin \alpha) \cos \mu - mg \cos \phi \right) \\
\dot{\chi} &= \frac{1}{mv \cos \phi} \left( (L + \eta T_{\text{max}} \sin \alpha) \sin \mu \right)
\end{align*}
\]

where \(m\) and \(g\) denote the mass of the aircraft and the acceleration due to the gravity, respectively. \(L(.)\) is the lift force, \(D(.)\) is the drag force, and \(T_{\text{max}}(.)\) denotes the maximum available thrust force. The aircraft is controlled by the angle of attack \(\alpha\), the throttle setting \(\eta\), and the bank angle \(\mu\). The lift force and drag force are given by

\[
\begin{align*}
L &= C_L S q \\
D &= C_D S q \\
q &= \frac{1}{2} Q v^2
\end{align*}
\]

where \(S\) is the reference wing area, \(Q\) is the air density, \(C_L\) is the lift coefficient, and \(C_D\) is the drag coefficient.

In the scenario of UAV guidance, a continuous control vector \([\alpha^f, \eta^f, \mu^f]\) is generated at each time step to guide the UAV to the target position. In manned aircraft guidance situations, many factors need to be considered. For example, instructions should not be changed frequently, and the generated trajectory should be smooth, not irregular, and so on. These factors are analyzed in the next section using reward shaping. Aircraft dynamics can be represented by overloading,\(^{32}\) and then, the dynamic equations become the following description

\[
\begin{align*}
\dot{v} &= g (n_x - \sin \phi) \\
\dot{\phi} &= \frac{g}{v} (n_y - \cos \phi) \\
\dot{\chi} &= \frac{-g}{v \cos \phi} n_z
\end{align*}
\]
where \( n_x, n_y, n_z \) are defined as
\[
\begin{align*}
    n_x &= \frac{\eta T_{\max} \cos \alpha - D}{mg} \\
    n_y &= \frac{(L + \eta T_{\max} \sin \alpha) \cos \mu}{mg} \\
    n_z &= \frac{(L + \eta T_{\max} \sin \alpha) \sin \mu}{mg}
\end{align*}
\]  
(5)

The continuous control variables are replaced with seven discrete control alternatives,\(^{32}\) including steady flight, max load factor left turn, max load factor right turn, max long acceleration, max long deceleration, max load factor pull-up, and max load factor pushover, which are also the behaviors of the aircraft in realistic situations.

### Proximal policy optimization-based algorithm for guidance agent training

In this article, the PPO-based algorithm is adapted to train an aircraft guidance agent, since it can be used in both discrete and continuous action spaces. The objective of PPO,\(^{12}\) which is maximized at each iteration, is defined as follows
\[
L_C^{V_{\text{FF}}+P}(\theta) = \hat{E}_t[L_C^V(\theta) - c_1 L_C^{FF}(\theta) + c_2 B[\pi_\theta](S_t)]
\]  
(6)

where \( \hat{E}_t[...] \) indicates the empirical average over a finite batch of samples, \( c_1 \) and \( c_2 \) are coefficients, \( B[\pi_\theta](S_t) \) denotes an entropy bonus, and \( L_C^{FF}(\theta) \) is a squared-error loss \( (V_\theta(S_t) - V^\text{arg})^2 \). \( L_C^V(\theta) \) is the clipped surrogate objective, which is defined as
\[
L_C^V(\theta) = \hat{E}_t[\min(r_t(\theta)\hat{a}_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon)\hat{a}_t)]
\]  
(7)

where \( \varepsilon \) is a hyperparameter and \( r_t(\theta) \) denotes the probability ratio
\[
r_t(\theta) = \frac{\pi_\theta(A_t|S_t)}{\pi_{\theta_{\text{old}}}(A_t|S_t)}
\]  
(8)

where \( \pi_\theta(A_t|S_t) \) and \( \pi_{\theta_{\text{old}}}(A_t|S_t) \) denote the probability to generate action \( A_t \) under situation \( S_t \) under new guidance strategy and old guidance strategy. The guidance strategy can be steadily improved in the training process using the clipped surrogate objective. The neural network is updated every \( T \) time steps. The advantage estimator \( \hat{a}_t \) used is as follows
\[
\hat{a}_t = \sigma_t + \gamma \sigma_{t+1} + \cdots + \gamma^{T-t} \sigma_T - 1
\]  
(9)

PPO algorithm is executed according to the training time step. However, there are not only training time step but also simulation time step. Because of the inconsistency in them, PPO algorithm needs to be used to train a guidance agent. The pseudocode of PPO for aircraft guidance agent training is shown in Algorithm 1. The simulation results and their comparison with the proposed method are shown in “Simulation evaluation” section.

### Algorithm 1. PPO for aircraft guidance agent training

1. Set aircraft and destination parameters.
2. Set training time step size \( \Delta t \), simulation time step \( \delta t \), and maximum number of control times.
3. Initialize policy parameter \( \theta_0 \).

\[ \text{for iteration}=1,2,... \text{ do} \]
\[ \text{for timestep}=1,2,...,T \text{ do} \]
4. Compute advantage estimators with Equation.9.
5. Obtain the positions of aircraft and destination.
6. Obtain the next state and reward.
7. Obtain the next state and reward.
8. Compute advantage estimators with Equation.9.
9. Optimize objective function \( \text{w.r.t } \theta \), with \( K \) epochs and minibatch size \( M \).
10. \( \theta_{\text{old}} <- \theta \).

\[ \text{end for} \]
\[ \text{end for} \]

### Reinforcement learning training framework

RL can be treated as Markov decision process (MDP).\(^{33}\) An MDP is defined as a tuple \( < S, A, T, R, \gamma > \), where \( S \) is the set of states, \( A \) is the set of actions that the agent can use to interact with the environment, \( T \) is the transition function that defines the probability from one state to another, \( R \) is the immediate reward function, and \( \gamma \) is the discount factor.

Unlike supervised learning agent, RL agents improve their abilities through continuous interaction with the environment. An aircraft guidance training framework is designed, as shown in Figure 2. Through training in the environment, the agent has the ability of guidance and then can be validated in special guidance simulation software or realistic situations.

In the training process, at each time step \( t \), the agent receives a state \( S_t \) in a state space \( S \) and selects an action \( A_t \) from an action space \( A \), following a policy \( \pi: S \rightarrow A \). The agent receives a scalar reward \( R_t \) and transitions to the next state \( S_{t+1} \) according to the environment dynamics. The agent aims to maximize the long-term reward.

For UAVs, the state set in this environment is a 10-dimensional vector, which is composed of the three-
The dimensional position and the heading angle of the aircraft as well as the destination, the flight path angle of the aircraft, and the number of control times remained. The action set is a three-dimensional continuous vector, which is defined as $A = \{a, h, \mu\}$. For manned aircraft, the state set is a 11-dimensional vector, adding the previous action as a dimension of the state vector. The action set includes steady flight, max load factor left turn, max load factor right turn, max long acceleration, max long deceleration, max load factor pull-up, max load factor pushover, keeping the previous action unchanged.

It is called an episode from the time when the aircraft receives the start instruction to the time when it reaches the destination successfully or fails. Every episode starts with initialization. In the initialization process, the information of the aircraft and destination are initialized, including their positions and moving patterns, and the reward shaping value is initialized, which are explained in detail in the next section. After the parameter setting is completed, the situation information is sent to the guidance agent as a tuple of state, and the initialization is completed. In each training time step, the agent generates an action and sends it to the environment. The action set includes steady flight, max load factor left turn, max load factor right turn, max long acceleration, max long deceleration, max load factor pull-up, max load factor pushover, keeping the previous action unchanged.

In this study, there are three rules to follow in reward shaping:

**Figure 2. Aircraft guidance training framework.**

**Training optimization**

In this section, a reward shaping method is adopted to solve the slow convergence problem and improve the performance of the generated trajectories. In addition, a pre-trained PPO algorithm is presented to further improve the training efficiency handling different kinds of moving destination guidance tasks.

**Reward shaping**

There are two problems when using RL method to train a guidance agent. One is that the only criterion to evaluate the guidance is the successful arrival to the destination, which is a sparse reward problem, leading to the slow convergence rate of training. The other is that different sequences which consist of the same actions have the same results in the training task. However, in realistic situations, it is very important to provide a smooth trajectory or instruction set with fewer changes for manned aircraft guidance. A reward shaping method is proposed to solve these two problems.

In this study, there are three rules to follow in reward shaping:
1. Limited by sector scope and maximum number of control times, the aircraft is guided to the destination in a certain direction.
2. The aircraft should be as close to the destination as possible and in the same direction as the destination.
3. For manned aircraft guidance, the generated action should not be changed frequently.

According to the above rules, the reward function is defined as follows

\[ R(S_t, A_t, S_{t+1}) = T(S_{t+1}) + F(S_t, A_t, S_{t+1}) \]  

where \( T(S_{t+1}) \) is the termination reward and \( F(S_t, A_t, S_{t+1}) \) is a bounded real-value function called position reward.

In this study, there are three kinds of termination states: the aircraft arrives at the destination with the same direction; the aircraft moves out of the sector; the maximum number of control times has been reached, and the aircraft is still in the sector and does not reach the destination. Termination reward is the reward obtained when the next state is termination, which is often used in the standard RL training. It is defined as

\[
T(S_{t+1}) = \begin{cases} 
  c_3, & \text{if successful arrived} \\
  c_4, & \text{if out of the sector} \\
  c_5, & \text{if no control times left} \\
0, & \text{else}
\end{cases}
\]  

where \( c_3, c_4, \) and \( c_5 \) are negative values.

\[ F(S_t, A_t, S_{t+1}) \text{ has the form} \]

\[
F(S_t, A_t, S_{t+1}) = \gamma \Phi(S_{t+1}) - \Phi(S_t)
\]

where \( \Phi(S_t) \) is a real-valued function over states. It can be proven that the final policy after using reward shaping is equivalent to the final policy without it. \( \phi(S_t) \) is defined as

\[
\phi(S_t) = C(S_t) + P(S_t)
\]

where \( C(S_t) \) is the continuous action reward and \( P(S_t) \) is the position reward, which is given by

\[
P(S_t) = aD(S_t) + bO(S_t) + cH(S_t)
\]

where \( D(S_t) \) is the horizontal distance reward function, \( O(S_t) \) is the direction reward function, and \( H(S_t) \) is the altitude reward function. \( a, b, \) and \( c \) are coefficients of \( D(S_t), O(S_t), \) and \( H(S_t) \), respectively.

To alleviate the pressure of the controller, the guidance task should be finished with the least number of control times and the action should not be changed frequently. As mentioned in “Proximal policy optimization-based algorithm for guidance agent training” section, an action which is the same as the previous one is added and defines the continuous action reward as

\[
C(S_t) = \begin{cases} 
  0, & \text{if UAVs} \\
  c_6, & \text{if manned aircraft and current action} \\
  & \text{is the same as the previous one} \\
  c_7, & \text{else}
\end{cases}
\]

Both \( c_6 \) and \( c_7 \) are negative values, and \( c_6 \) is larger than \( c_7 \).

For general training process, the position reward functions are given by

\[
\begin{align*}
D(S_t) &= -\sqrt{(x_t^a - x_t^f)^2 + (y_t^a - y_t^f)^2} \\
O(S_t) &= \cos(\chi_t^a - \chi_t^f) \\
H(S_t) &= |z_t^a - z_t^f|
\end{align*}
\]

The training speed can be improved using the shaping reward function proposed above. For different tasks in realistic situations, training efficiency can be improved by adjusting one or more reward functions or coefficients.

**Pretrained proximal policy optimization algorithm**

For different tasks, the destinations have different moving patterns, such as straight-moving or curve-moving. In spite of reward shaping, there is still a problem of slow training speed. For complex scenarios, it will not even be able to train guidance agents successfully. Therefore, to speed up the training process and to train an agent quickly for the new task, it is necessary to propose an improved algorithm.

Considering the state space and action space, a pretrained PPO algorithm is proposed. The state space can be divided into two parts: the state of the aircraft and the state of the destination. The speed of the destination is far less than that of the aircraft, so the influence of the aircraft state on the agent is greater than that of the destination. Therefore, it can be assumed that the position of the destination is fixed, which means that four dimensions of the state space are unchanged, greatly reducing the complexity of the state space.

The pretrained PPO algorithm is adopted. By designing a greedy exploration policy, an agent can be trained in the fixed position destination environment and set as a baseline agent. Based on this agent, a conservative exploration strategy is set up to train an agent with guidance ability in the moving destination scenarios. This algorithm is more efficient than directly using PPO in moving destination scenarios. The pseudocode of pretrained PPO is shown in Algorithm 2.

**Simulation evaluation**

In this section, first of all, the feasibility and performance of the proposed method are verified by general simulations. Furthermore, the approach guidance simulation of carrier-
Simulation setup

General simulation setup. Six guidance simulations are carried out and compared, including PPO for manned aircraft, PPO with continuous action reward function for manned aircraft, PPO with position reward function for manned aircraft, PPO with both reward functions for manned aircraft, PPO for UAV, and PPO with position reward function for UAV. Then, PPO with position reward function for UAV and PPO with both reward functions for manned aircraft using pretrained PPO algorithm are applied, and the simulation results are compared with that of PPO without pretraining.

In actual or simulated air guidance, an aircraft perceives the situation through multisensors. In this article, an assumption is made that a sensor with full situation perception ability is used, through which an aircraft can obtain the position and heading angle of itself and destination without error. The specific aircraft considered is the F-4 Phantom. The simulation parameters are given in Table 1.

The Adam optimizer is used for learning the neural network parameters with a learning rate of $5 \times 10^{-4}$. The actor has three hidden layers with 512, 512, and 1024 units, respectively. The critic has three hidden layers with 256, 256, and 512 units, respectively. The time step $T$ is 10,240 and the minibatch size $M$ is 32. The number of epochs $K$ is 4, and the discount factor $\gamma$ is 0.99.

For each simulation, the reward shaping parameters $c_3$, $c_4$, and $c_5$ are set to 20, $-20$, $-10$, respectively. $c_6$ and $c_7$ are set to 0.01 and 0.1, respectively. $a$, $b$, and $c$ are set to 0.0001, 0.5, and 0.1, respectively.

Carrier-based aircraft approach guidance simulation setup. Guiding aircraft to approach and land on the deck is one of the most important tasks in the use of aircraft carriers. However, it may lead to a landing failure because of the carrier motion, small landing area, and low-visibility conditions. Under different visibility conditions, there are different approach guidance methods. This section mainly studies approach guidance under low-visibility conditions, which cannot be solved by a visual approach.

In the previous approach process, the aircraft should fly a holding pattern with fixed relative direction and
position.\textsuperscript{37} It is assumed that the carrier moves in a straight line at a constant velocity. The approach process of an aircraft is shown in Figure 3. All distances are in nautical miles, and the distance measurement equipment is radar\textsuperscript{37} or satellite navigation system.\textsuperscript{38}

Figure 3(a) shows the aircraft approach from the stern of the carrier. The aircraft is flying in a holding pattern, and it leaves the pattern and flies behind the carrier in a straight line with the runway after it receives the approach instruction. From 20 distance measuring equipment (DME) to 10 DME, the aircraft lowers its altitude from 5000 ft to 1200 ft and then flies straight to the final approach fix which is 3 DME from the carrier.

Figure 3(b) shows the aircraft approach in front of the carrier bow. The aircraft is flying in a holding pattern, and after receiving the approach instruction, it leaves the pattern and flies over the carrier and behind it. The aircraft began to descend after it left the carrier stern 4 to 5 DME. When it is 10 DME away from the carrier, the aircraft dropped to 1200 ft and flies straight to the final approach fix, which is 3 DME from the carrier.

The current holding pattern has not met the approach requirements due to the complex airspace around the carrier. Therefore, it is necessary to propose a new approach process so that the aircraft can approach at a moving carrier in arbitrary holding pattern. The method proposed in this section is to guide a manned aircraft from arbitrary arrival holding pattern to the final approach fix of a moving carrier. In actual approach process, altitude should not change frequently or intermittently. Therefore, the altitude reward shaping is needed to ensure that a reasonable flight trajectory can be generated. The altitude reward function is defined as in equation (17)

\[
H(s_t) = \begin{cases} 
-u(1-h/h_{\text{max}}), & \text{if } \text{dis} \geq 20\text{DME} \\
-uv(1-h/h_{\text{max}}) - (1-v)h_{\text{min}} + \Delta h(\text{dis} - 10) - h/h_{\text{max}}, & \text{if } 10 \text{ DME} \leq \text{dis} \leq 20 \text{ DME} \\
-uv(1-h/h_{\text{max}}) - (1-v)(h/h_{\text{min}} - 1), & \text{if } \text{dis} \leq 10 \text{ DME}
\end{cases}
\]  \ (17)

where \(h\) is the altitude of the aircraft, \(h_{\text{max}}\) is 5000 ft, \(h_{\text{min}}\) is 1200 ft, \(\Delta h\) is the altitude changed for a climb or descend action. \(u\) and \(v\) are defined as

\[
u = \begin{cases} 
1, & \text{if the aircraft is ahead of the carrier} \\
0, & \text{else}
\end{cases}
\]  \ (18)

\[
u = \begin{cases} 
1, & \text{if } h \leq h_{\text{max}} \\
0, & \text{else}
\end{cases}
\]  \ (19)

The simulation is performed in a commercial software.\textsuperscript{39} In this simulation, the range of the horizontal plane of the sector is 40\text{nmi} \times 40\text{nmi}, and the vertical range is 0–5000 ft. The carrier is in the center of the sector at a height of 0 and the aircraft is randomly located 20 nmi away from the carrier. The turning radius of the aircraft is 1.5 nmi; the horizontal and vertical velocities are 3.3 nmi/min and 380 ft/min. The horizontal speed of the carrier is 0.5 nmi/min. Since the direction of the aircraft is randomly initialized, the direction of the carrier does not affect the result and is
set to northward. The control time interval is 20 s and the maximum number of control times is 100.

**General simulation results**

**Simulation with reward shaping.** Figure 4 shows the success rate in the training process using four kinds of reward functions. The training process of manned aircraft guidance training is shown in Figure 4(a). Using PPO, the system converges after training 200 iterations. The training speed of PPO with continuous action reward function is the slowest, and the system converges after more than 400 iterations. PPO with position reward function and with both reward functions has the fastest training speed. After about 120 iterations of training, the systems converged. For UAV guidance training, it is almost impossible to train successfully without reward shaping. Using reward shaping, the system converges after about 400 iterations, and the success rate can reach more than 99%, as shown in Figure 4(b).

Each well-trained agent is tested for 1000 simulations. The success rate, average number of control times, and average computational time to generate an instruction are given in Table 2. Typical trajectories are shown in Figure 5. The number of control times is an important parameter for evaluating the performance of an agent in the guidance of manned aircraft. The less the number of control times, the less pressure the pilot will have, and the smoother the flight trajectory will be. As shown in Figure 5(b), the aircraft only needs to be controlled four times to successfully reach the destination. For the flight trajectory shown in Figure 5(a), the aircraft is controlled more than 10 times to reach the destination, which will bring greater pressure to the pilot.

| Algorithm                  | Success rate | Average control times | Average computational time |
|----------------------------|--------------|-----------------------|----------------------------|
| Manned aircraft guidance training |              |                       |                            |
| PPO                        | 93.7%        | 10.68                 | 7.23 ms                    |
| PPO with continuous action reward | 97.9%        | 7.74                  | 9.57 ms                    |
| PPO with position reward    | 97.8%        | 9.79                  | 7.77 ms                    |
| PPO with both reward functions | 98.2%        | 7.97                  | 9.35 ms                    |
| UAV guidance training       |              |                       |                            |
| PPO                        | 0.3%         | —                     | 18.9 ms                    |
| PPO with position reward    | 99.7%        | —                     | 18.1 ms                    |
| NMPC                       | 99.8%        | —                     | 287.3 ms                   |

UAV: unmanned aerial vehicle; PPO: proximal policy optimization; NMPC: nonlinear model predictive control.

![Figure 5](image_url)
For manned aircraft guidance training, using the continuous action reward function, the quality of the trajectory has been greatly improved, as shown in Figure 5(b). The number of control times is reduced by 27.5% while the training speed slowed down. The position reward function can be used to improve the training speed by 40%, but it still needs a lot of number of control times and the trajectory quality remains low performance, as shown in Figure 5(c). Using both reward functions can improve the success rate by 4.5%, accelerate the training speed by 40% and reduce the number of control times by 25.4% to make the generated trajectory smoother, as shown in Figure 5(d). It takes less than 10 ms to generate an instruction using an agent, which is very efficient.

For manned aircraft guidance training, using the continuous action reward function, the quality of the trajectory has been greatly improved, as shown in Figure 5(b). The number of control times is reduced by 27.5% while the training speed slowed down. The position reward function can be used to improve the training speed by 40%, but it still needs a lot of number of control times and the trajectory quality remains low performance, as shown in Figure 5(c). Using both reward functions can improve the success rate by 4.5%, accelerate the training speed by 40% and reduce the number of control times by 25.4% to make the generated trajectory smoother, as shown in Figure 5(d). It takes less than 10 ms to generate an instruction using an agent, which is very efficient.

For UAV guidance agent training, simulation results are compared with the mainstream traditional algorithm NMPC. Using standard PPO, it is impossible to train successfully, as shown in Figure 5(e). Using PPO with reward shaping, the success rate can reach 99.7%, as shown in Figure 5(f), which is almost the same as that of using NMPC, 99.8%. However, using the proposed method, the instruction generation time can be reduced from nearly 300 ms to within 20 ms.

**Simulation using pretrained proximal policy optimization.** Using reward shaping, agents can be trained for different destinations to generate reasonable trajectories. However, destinations under different tasks have a variety of moving patterns. It takes a lot of time to train different agents from scratch for different tasks. Using pretrained PPO algorithm, based on the agent of static destination, the training efficiency for a new agent can be greatly improved.

Two scenarios of straight and curve moving destinations are set up for manned aircraft and UAV guidance agent training using pretrained PPO algorithm with the proposed reward functions. The success rates in the training process are shown in Figure 6. Using well-trained agents, the success rate, average number of control times, and average computational time to generate an instruction are presented in Table 3. The typical trajectories are shown in Figure 7.

For straight-moving destination, the performance of agents with or without pretraining is comparable. The number of iterations needed to train a static destination guidance agent (baseline) and a straight-moving destination guidance agent based on it is slightly less than that required for direct training. However, based on this baseline, only less than 10 iterations are needed to make the training successful, which greatly improves the training efficiency. Meanwhile, the results show that based on this agent, for other types of tasks, such as reaching the curve-moving destination, high-performance agents can also be trained.
efficiently. Compared with NMPC, the time required to generate an instruction is greatly reduced with a small reduction in the success rate.

Figure 7. Typical trajectory results of pretrained PPO: (a) manned aircraft, straight-moving destination; (b) manned aircraft, circle-moving destination; (c) UAV, straight-moving destination; and (d) UAV, circle-moving destination. UAV: unmanned aerial vehicle; PPO: proximal policy optimization.

Table 4. Success rate and average number of control times of aircraft approach guidance agent.

| Scenario                        | Success rate (%) | Average control times |
|---------------------------------|------------------|-----------------------|
| Approach from the stern of the carrier | 99.7             | 9.6                   |
| Approach in front of the carrier bow | 99.4             | 12.7                  |

Figure 8. Success rate during the process of aircraft approach guidance training.

Carrier-based aircraft approach guidance simulation results

The success rate during the process of carrier-based aircraft guidance training is shown in Figure 8. An agent for static carrier approach guidance is gained through 100 iterations of training. Based on this agent, after 20 iterations, an agent for straight-moving carrier approach guidance is gained. The well-trained agent is tested in the scenario of an aircraft with random initial position and a carrier in the map center. The success rate and average number of control times are presented in Table 4.
The well-trained agent is applied in a general combat simulation platform, which can be used for individual and tactical simulation. The typical simulation results are shown in Figure 9.

The simulation results show that the trajectory can be generated to guide a manned aircraft approaching at a moving carrier. The generated trajectory satisfies the approach requirements by the proposed method, and the aircraft can approach from anywhere without waiting in the unique holding pattern, which is used now.

Conclusion
This article proposed a pretrained PPO algorithm for aircraft guidance to reach a moving destination in a certain direction in three-dimensional continuous space. An RL-based framework is built, which can be used to train a manned aircraft guidance agent or a UAV control agent. Continuous action reward function and position reward function are proposed to improve the performance of the generated trajectories and the training efficiency. For different kinds of moving destination guidance tasks, an agent can be trained quickly based on the existing agent using pretrained PPO algorithm. The general simulation and carrier-based aircraft approach guidance simulation results show that the proposed approach can fulfill guidance tasks and has high training efficiency as well as performance.

Future work could focus on two different aspects to improve the proposed approach. One is to train agents for destinations with multiple moving patterns. The other is to improve the proposed algorithm that can be used to train a general agent to fit most of the application conditions.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by the National Natural Science Foundation of China under grant no. 91338107.

ORCID iD
Hui Li https://orcid.org/0000-0002-7333-6942

References
1. Meng Y, Wang W, and Ding Z. Research on the visual/inertial integrated carrier landing guidance algorithm. Int J Adv Robot Syst 2018; 15: 1–9.
2. Wang L, Zhu Q, Zhang Z, et al. Modeling pilot behaviors based on discrete-time series during carrier-based aircraft landing. J Aircr 2016; 53: 1922–1931.
3. Gautam A, Sujit PB, and Saripalli S. A survey of autonomous landing techniques for UAVs. In: IEEE international conference on unmanned aircraft systems, Orlando, FL, USA, May 2014, pp. 1210–1218. Piscataway, NJ: IEEE.
4. Toubman A, Roessingh JJ, Spronck P, et al. Transfer learning of air combat behavior. In: IEEE international conference on machine learning and applications, Miami, FL, USA, December 2015, pp. 226–231. Piscataway, NJ: IEEE.
5. Eklund JM, Sprinkle J, and Sastry S. Switched and symmetric pursuit/evasion games using online model predictive control with application to autonomous aircraft. IEEE Trans Contr Syst Technol 2012; 20: 604–620.
6. You DI and Shim DH. Design of an aerial combat guidance law using virtual pursuit point concept. Proc Inst Mech Eng Part G 2014; 229: 792–813.
7. Burns BS, Blue PA, and Zollars MD. Simulation of a real-time trajectory generator for automated aerial refueling with a required time of arrival. In: AIAA modeling and simulation technologies conference and exhibit, Hilton Head, South Carolina, August 2007, pp. 1–13. Reston, VA: AIAA.
8. Rucco A, Sujit PB, Aguiar AP, et al. Optimal rendezvous trajectory for unmanned aerial-ground vehicles. *IEEE Trans Aerosp Electron Syst* 2018; 54: 834–847.
9. Sutton RS and Barto AG. *Reinforcement learning: an introduction*. 2nd ed. London: MIT Press, 2017, p. 1.
10. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015; 7540: 529–533.
11. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. In: *International conference on learning representations*, San Juan, Puerto Rico, May 2016. Amherst, MA: ICLR.
12. Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017; ArXiv preprint:1707.06347.
13. Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. In: *International conference on machine learning*, Beijing, China, June 2014, pp. 387–395. New York, NY: ACM.
14. Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization. In: *International conference on machine learning*, Lille, France, February 2015. New York, NY: ACM.
15. Silver D, Huang A, Maddison CJ, et al. Mastering the game of go with deep neural networks and tree search. *Nature* 2016; 529: 484–503.
16. Koer J, Bagnell JA, and Peters J. Reinforcement learning in robotics: a survey. *Int J Robot Res* 2013; 97: 1–38.
17. Kim M, Han D, Park J, et al. Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay. *Appl Sci* 2020; 10: 1–15.
18. Dunn C, Valasek J, and Kirkpatrick K. Unmanned air system search and localization guidance using reinforcement learning. In: *AIAA infotech at aerospace*, Garden Grove, California, June 2012. Reston, VA: AIAA.
19. Tozer B, Mazzuch T, and Sarkani S. Many-objective stochastic path finding using reinforcement learning. *Expert Syst Appl* 2017; 72: 371–382.
20. Sharma A, Gupta K, Kumar A, et al. Model based path planning using Q-learning. In: *IEEE international conference on industrial technology*, Toronto, ON, Canada, March 2017, pp. 837–842. Piscataway, NJ: IEEE.
21. Rodriguez-Ramos A, Sampedro C, Bavle H, et al. A deep reinforcement learning strategy for UAV autonomous landing on a moving platform. *J Intell Robot Syst* 2019; 93: 351–366.
22. Waldoek A, Greatwood C, Salama F, et al. Learning to perform a perched landing on the ground using deep reinforcement learning. *J Intell Robot Syst* 2018; 92: 685–704.
23. Yoo B and Kim J. Path optimization for marine vehicles in ocean currents using reinforcement learning. *J Mar Sci Technol* 2016; 21: 334–343.
24. Bellmear MG, Naddaf Y, Veness J, et al. The arcade learning environment: an evaluation platform for general agents. *J Artif Intell Res* 2013; 47: 253–279.
25. Brockman G, Cheung V, Pettersson L, et al. OpenAI gym. 2016; ArXiv preprint:1606.01540.
26. Zamora I, Lopez NG, Vilches VM, et al. Extending the OpenAI gym for robotics: a toolkit for reinforcement learning using ROS and gazebo. 2016; ArXiv preprint:1608.05742.
27. Ng A, Harada D, and Russell S. Policy invariance under reward transformations: theory and application to reward shaping. In: *International conference on machine learning*, Bled, Slovenia, June 1999. New York, NY: ACM.
28. Toubman A, Roessingh JJ, Sprock P, et al. Rewarding air combat behavior in training simulations. In: *IEEE international conference on systems, man, and cybernetics*, Kowloon, China, 2015, pp. 1397–1402. Piscataway, NJ: IEEE.
29. Tumm K and Agogino A. Agent reward shaping for alleviating traffic congestion. In: *International conference on autonomous agents and multiagent systems*, Hakodate, Hokkaido, Japan, May 2006. Berlin, German: Springer.
30. Cruciol L, de Arruda AC, Li W, et al. Reward functions for learning to control in air traffic flow management. *Transp Res C Emerg Technol* 2013; 35: 141–155.
31. Virtanen K, Karelhhti J, and Raivio T. Modeling air combat by a moving horizon influence diagram game. *J Guid Control Dyn*. 2006; 29: 1080–1091.
32. Austin F, Carbone G, Falco M, et al. Game theory for automated maneuvering during air-to-air combat. *J Guid Control Dyn* 1990; 13: 1143–1149.
33. Li Y. Deep reinforcement learning. 2018; ArXiv preprint:1810.06339.
34. Williams P. Three-dimensional aircraft terrain-following via real-time optimal control. *J Guid Contr Dyn* 2007; 30: 1201–1205.
35. Kingma DP and Ba J. Adam: a method for stochastic optimization. In: *International conference on learning representations*, San Diego, CA, USA, May 2015. Amherst, MA: ICLR.
36. Wang L and Bax I. Quadrotor autonomous approaching and landing on a vessel deck. *J Intell Robot Syst* 2018; 92: 125–143.
37. Prickett AL and Parkes CJ. Flight testing of the F/A-18E/F automatic carrier landing system. In: *IEEE aerospace conference*, Big Sky, MT, USA, March 2001. pp. 2593–2612. Piscataway, NJ: IEEE.
38. Petrovello MG, O’Keefe K, Lachapelle G, et al. Field results of a GPS/INS-based approach to measuring ship flexure onboard an aircraft carrier. In: *International technical meeting of the satellite division of the institute of navigation*, Long Beach, CA, USA, September 2005, pp. 2039–2048. Manassas, VA: ION.
39. Rong Ao Technology Co., Ltd. Commercial air combat simulation software FG_SimStudio. http://www.eyeextent.com/producttail/11/7 (accessed 20 March 2020).
40. Grüne L and Pannek J. *Nonlinear model predictive control*. London: Springer Press, 2011.