Journal Pre-proof

A novel trajectories optimizing method for dynamic soaring based on deep reinforcement learning

Wanyong Zou, Ni Li, Fengcheng An, Kaibo Wang, Changyin Dong

PII: S2214-9147(24)00281-2

DOI: https://doi.org/10.1016/j.dt.2024.12.007

Reference: DT 1546

To appear in: *Defence Technology*

Received Date: 31 August 2024

Revised Date: 20 November 2024

Accepted Date: 8 December 2024

Please cite this article as: Zou W, Li N, An F, Wang K, Dong C, A novel trajectories optimizing method for dynamic soaring based on deep reinforcement learning, *Defence Technology*, https://doi.org/10.1016/j.dt.2024.12.007.

A novel trajectories optimizing method for dynamic soaring based on deep reinforcement learning

Wanyong Zou[a], Ni Li[a,b*], Fengcheng An[a], Kaibo Wang[a], Changyin Dong[a,b]

[a]School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China
[b]National Key Laboratory of Aircraft Configuration Design, Xi'an 710072, China

[*]Corresponding Author: lini@nwpu.edu.cn

A novel trajectories optimizing method for dynamic soaring based on deep reinforcement learning

Abstract

Dynamic soaring, inspired by the wind-riding flight of birds such as albatrosses, is a biomimetic technique which leverages wind fields to enhance the endurance of unmanned aerial vehicles (UAVs). Achieving a precise soaring trajectory is crucial for maximizing energy efficiency during flight. Existing nonlinear programming methods are heavily dependent on the choice of initial values which is hard to determine. Therefore, this paper introduces a deep reinforcement learning method based on a differentially flat model for dynamic soaring trajectory planning and optimization. Initially, the gliding trajectory is parameterized using Fourier basis functions, achieving a flexible trajectory representation with a minimal number of hyperparameters. Subsequently, the trajectory optimization problem is formulated as a dynamic interactive process of Markov decision-making. The hyperparameters of the trajectory are optimized using the Proximal Policy Optimization (PPO2) algorithm from deep reinforcement learning (DRL), reducing the strong reliance on initial value settings in the optimization process. Finally, a comparison between the proposed method and the nonlinear programming method reveals that the trajectory generated by the proposed approach is smoother while meeting the same performance requirements. Specifically, the proposed method achieves a 34% reduction in maximum thrust, a 39.4% decrease in maximum thrust difference, and a 33% reduction in maximum airspeed difference.

Keywords Dynamic soaring; Differential flatness; Trajectory Optimization; Proximal Policy Optimization

1.Introduction

In modern and intelligent battlefield confrontations, unmanned aerial vehicles (UAVs) play a crucial role in defense, being widely used for monitoring, surveillance, and serving as pseudo-satellite communication relays. Typical methods to enhance a UAV's endurance involve optimizing its aerodynamic configurations to amplify lift-to-drag ratios and enhance aerodynamic efficiency, advancing engine or battery energy efficiency, and exploring new energy sources such as solar-powered and dynamic soaring UAVs. Among these methods, leveraging renewable energy sources to enhance flight endurance has emerged as a leading research focus, garnering growing interest from the scientific community.

Solar-powered UAVs are the most studied for long-endurance applications due to the simplicity of using solar panels to convert solar energy into electrical power. A significant amount of in-depth work has already been completed[1–3]. In contrast to solar energy, wind energy stands out as another prominent renewable energy source. The utilization of wind fields for UAV flight, called dynamic soaring, is inspired by the remarkable long-distance flights of albatrosses, which traverse vast distances with minimal energy consumption[4]. Current research on dynamic soaring flight primarily encompasses three key areas: (1) biological observations for the study of gliding behavior, (2) analysis of mechanisms for acquiring energy during gliding, and (3) the planning and optimization of gliding trajectories.

In 1883, British scientist Lord Rayleigh[5] presented a comprehensive study of the dynamic soaring behavior observed in albatrosses. Weimerskirch[6] utilized a GPS receiver and heart rate monitoring equipment to meticulously track albatrosses' long flights and presented a thorough analysis of flight patterns and cardiovascular data. The research unveiled a fascinating revelation: albatrosses eschew anticipated linear trajectories within regions characterized by updrafts along the continental shelf edge, instead engaging in elegantly smooth circular movements.

Richardson, Sachs, and other scholars elucidated the underlying gliding mechanism from the energy perspective and pointed out that gliding is sustained by the gradient wind field. The flight pattern, known as the Rayleigh cycle, consists of four stages in each iteration: ascending into the wind, executing high-altitude turns (from upwind to downwind), descending with the wind, and performing low-altitude turns (from downwind to upwind). In the late 1980s, Sachs[7] pioneered the idea that aircraft could harness energy from gradient wind fields through optimal control, akin to albatross flight. Subsequent studies further demonstrated the feasibility of small gliding UAVs achieving soaring flight and identified the requisite wind conditions[8–10]. Additionally, Mir introduced an innovative approach that integrates dynamic soaring with a morphing UAV, resulting in a 14%–15% reduction in required wind shear for a sustained flight, as well as decreases in drag, lift requirements, and angle of attack requirements by 15%, 11%, and 20%, respectively, while concurrently boosting maximum velocity by 6.2%.

Although we have witnessed exciting progress, achieving sustained unpowered gliding across extended regions continues to pose significant challenges due to the dynamic nature of real-world wind patterns, which vary spatially and evolve over time. Presently, there is a dearth of effective techniques for accurately modeling these wind dynamics. However, within localized areas, wind conditions tend to exhibit relative stability over time, often being primarily

influenced by spatial positioning. Consequently, this study is primarily dedicated to optimizing trajectories for dynamic soaring within such fixed regions. Common approaches currently employed to tackle this issue encompass Gaussian pseudo-spectral methods[11], direct collocation techniques[12], and other variations[13].

In recent years, there has been significant attention on DRL based optimization methods for both linear and nonlinear system design. Vitaly[14] proposed a reinforcement learning approach to estimate the optimal launch time for defenders and the guidance law for targets in real-time. Peng[15] introduced a pioneering model-free control strategy, leveraging integral reinforcement learning techniques, to stabilize highly flexible aircraft under uncertainty. Concurrently, considerable research has explored RL techniques in dynamic soaring. Novati[16] demonstrated that DRL can identify gliding and landing strategies with various optimality criteria regardless of any explicit knowledge of underlying physics. They found that model-free RL leads to more robust gliding compared to model-based optimal control strategies. Zhao[17] introduced a Twin Delayed Deep Deterministic Policy Gradient (TD3) RL algorithm to investigate optimal strategies for unpowered gliders to harness energy from thermal updrafts. Reddy[18] identified an effective navigation strategy, employing RL to make sequential decisions in response to ascending thermal plumes (thermals). They validated the learned flight policy through field experiments, numerical simulations, and estimates of measurement noise due to atmospheric turbulence.

Up to now, DRL methods have been predominantly used on static soaring, where energy is harvested from thermals, while there has been limited exploration into optimizing dynamic soaring through reinforcement learning. Thus, this paper endeavors to tackle the trajectory optimization challenge of dynamic soaring using the reinforcement learning technique with PPO2 algorithm. Initially, it constructs smooth and dynamically feasible flight paths utilizing the Fourier series as foundational functions through the differential flatness method, sidestepping the complexities associated with solving nonlinear system differential equations. Subsequently, a PPO2 based DRL approach is applied to iteratively update the hyperparameters. Compared against the nonlinear programming method, the proposed method offers better accuracy and robustness.

The main contributions of this paper are as follows. (1) Utilizing the differential flatness method to determine optimal flight trajectories, represented through Fourier series as basis functions, enables examination of the flight patterns' characteristics; (2) The reinforcement learning method mitigates the dependence on initial value settings inherent in traditional numerical optimization approaches; (3) Comparing optimization outcomes with those of the nonlinear programming (NP) method showcases the viability of reinforcement learning in dynamic gliding trajectory planning and optimization.

The rest of the paper is organized as follows. The background and preliminaries of this paper are stated in section II, while Section III delves into the trajectory construction method grounded in differential flatness and elucidates the core principles of the PPO2 method. Extensive simulations of the proposed trajectory optimal algorithm are shown in Section IV. Finally, Section V provides a summary of the research and offers insights into future directions.

2. Background and preliminaries

2.1. Windy field

Generally, the wind field is defined as $w = f(x, y, z, t)$, which reflects that the wind changes both with position and time. Meteorological observation studies indicate that under standard conditions, the variation of the wind field in a small area over a short period is minimal and typically negligible[19]. Based on this, to simplify the model, most research considers the wind field as a high shear gradient model occurring at the boundary layer between two regions with significantly different prevailing wind vectors[20]. This approach allows us to focus more on key factors, thereby enhancing the practicality and efficiency of the model. To approximate the wind shear, various mathematic models have been used to describe wind profiles, including linear[21], exponential[22, 23] or logarithmic models[24]. In this study, the exponential model with nonlinear correction parameters, as proposed by Zhao[25] is employed, which is given as:

$$w_x = \beta[Ah + \frac{1-A}{h_{\mathrm{tr}}}h^2] \tag{1}$$

where, w_x represents the wind speed in the x-direction, h is the height, h_{tr}denotes the transition height at which the horizontal wind becomes constant with respect to altitude, β is the average slope of the wind profile, and A is a hyperparameter dependent on environmental conditions. To ensure that the wind component remains within a certain range $[0, w_{x,\max}]$, it is required that $0 < A < 2$. The curves of different A values are illustrated in Fig. 1, where wind shear profiles correspond to logarithmic-like, exponential-like, and straight line wind profiles at $0 < A < 1$, $1 < A < 2$ and $A = 1$, respectively. The value of $\beta = 0.05$, $h_{\mathrm{tr}} = 609.6, A = 1.5$ is chosen for this work.

This wind field has three typical characteristics: ① it remains constant over time; ② it aligns with the positive direction of the *x*-axis; ③ wind speeds are uniform within the same horizontal plane.

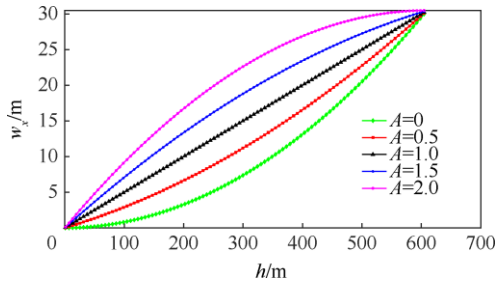


Fig. 1. Wind shear profiles ($\beta = 0.05$, $h_{\mathrm{tr}} = 609.6$).

2.2. Aerodynamics in windy environment

Both the 3 degree-of-freedom (DOF) point-mass model and the 6 DOF rigid body dynamics model have been used for studying dynamic soaring. A comparison of the two models was carried out and it suggests that these two models offer comparable performance [21]. Due to its computational advantages, the 3 DOF point-mass model is used. This model is described in an inertial reference frame $\mathcal{F}_{\mathrm{I}}$ and a body-fixed reference frame $\mathcal{F}_{\mathrm{B}}$, under the assumption of a flat, non-rotating Earth.

As Fig. 2, the inertia frame $\mathcal{F}_{\mathrm{I}}$ can be defined with three unit vectors $\hat{\boldsymbol{n}}_1$ point to the East, $\hat{\boldsymbol{n}}_2$ to the North, and $\hat{\boldsymbol{n}}_3$ upward from a fixed point on the earth surface. The body-fixed reference frame $\mathcal{F}_{\mathrm{B}}$ with the three unit vectors $\hat{\boldsymbol{e}}_1$ pointing through the nose, $\hat{\boldsymbol{e}}_2$ pointing to the left of the aircraft, and $\hat{\boldsymbol{e}}_3$ pointing upward in perpendicular to the other two axes following the right-hand rule.

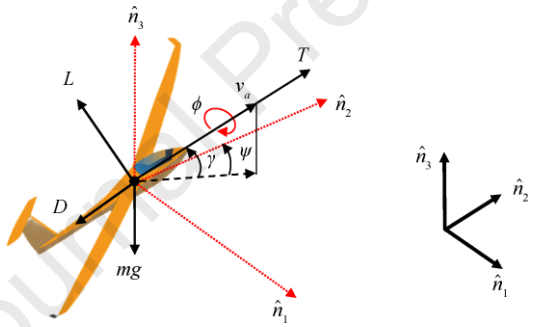


Fig. 2. Inertia frame $\mathcal{F}_{\mathrm{I}}$ and forces.

The transformation from reference frame $\mathcal{F}_{\mathrm{I}}$ to reference frame $\mathcal{F}_{\mathrm{B}}$ can be realized through the following rotation matrix,

$$R_{\mathcal{F}_{\mathrm{I}}}^{\mathcal{F}_{\mathrm{B}}} = R_{1,\phi} R_{2,\gamma} R_{3,\psi} \tag{2}$$

where $R_{1,\phi}$, $R_{2,\gamma}$, and $R_{3,\psi}$ are elementary rotation matrices, γ is the flight path angle, ψ is the heading angle, ϕ is the bank angle. Here, $R_{1,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$, $R_{2,\gamma} = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix}$, and $R_{3,\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$. When wind is considered, the absolute velocity of the aircraft in the inertial frame can be written as the summation of air-relative velocity and the wind velocity,

$$\boldsymbol{V}|_{\mathcal{F}_I} = \left(R_{\mathcal{F}_{\mathrm{I}}}^{\mathcal{F}_{\mathrm{B}}}\right)^{\mathrm{T}} \boldsymbol{V_a}|_{\mathcal{F}_{\mathrm{B}}} + \boldsymbol{W}|_{\mathcal{F}_{\mathrm{I}}} \tag{3}$$

where $\boldsymbol{V}|_{\mathcal{F}_{\mathrm{I}}} = [\dot{x}, \dot{y}, \dot{z}]^{\mathrm{T}}$ is the velocity of the aircraft in the inertial frame, x, y and z are coordinate along the $\hat{\boldsymbol{n}}_1$, $\hat{\boldsymbol{n}}_2$ and $\hat{\boldsymbol{n}}_3$, respectively. $\boldsymbol{V}_{\mathrm{a}}|_{\mathcal{F}_{\mathrm{B}}} = V_{\mathrm{a}}\hat{\boldsymbol{e}}_1$ is the aircraft velocity relative to the wind. Then, we can use the Newton's 2nd law to obtain its dynamics in the body frame,

$$m\dot{\boldsymbol{V}}\big|_{\mathcal{F}_{\mathrm{B}}} = \sum \boldsymbol{F}|_{\mathcal{F}_{\mathrm{B}}} \tag{4}$$

where m is the mass of the aircraft, and $\boldsymbol{F}$ is the force given by,

$$\sum \boldsymbol{F} = (T - D + mg\sin\gamma)\hat{\boldsymbol{e}}_1 - mg\cos\gamma\sin\phi\,\hat{\boldsymbol{e}}_2 + (L - mg\cos\gamma\cos\phi)\hat{\boldsymbol{e}}_3 \tag{5}$$

where L and D are the lift and drag, respectively, and are described by:

$$L = \frac{1}{2}\rho {v_{\mathrm{a}}}^2 S C_{\mathrm{L}} \tag{6}$$

$$D = \frac{1}{2}\rho {v_{\mathrm{a}}}^2 S C_{\mathrm{D}} \tag{7}$$

where ρ is the atmospheric density, S is the wing area, and C_{D} and C_{L} are the coefficients of drag and lift respectively. The drag coefficient is given by the following equation:

$$C_{\mathrm{D}} = C_{\mathrm{D}_0} + K {C_{\mathrm{L}}}^2 \tag{8}$$

in which C_{D_0} is the zero lift drag coefficient, and K is the induced drag coefficient.
Combining Eqs.(3)–(8), we can obtain the dynamics equations as follows[26],

$$\begin{cases} \dot{x} = V_{\mathrm{a}} \cos\gamma \cos\psi + w_x \\ \dot{y} = V_{\mathrm{a}} \cos\gamma \sin\psi \\ \dot{z} = -V_{\mathrm{a}} \sin\gamma \\ \dot{V_a} = \dfrac{T}{m} - \dfrac{D}{m} + g\sin\gamma - \dfrac{\partial w_x}{\partial h}\dot{h}\cos\gamma\cos(\psi) \\ \dot{\gamma} = \dfrac{-L\cos\phi + mg\cos\gamma}{mV_{\mathrm{a}}} + \dfrac{1}{V_{\mathrm{a}}}\dfrac{\partial w_x}{\partial h}\dot{h}\sin\gamma\cos(\psi) \\ \dot{\psi} = -\dfrac{L\sin\phi}{mV_{\mathrm{a}}\cos\gamma} + \dfrac{1}{V_{\mathrm{a}}\cos\gamma}\dfrac{\partial w_x}{\partial h}\dot{h}\sin(\psi) \end{cases} \tag{9}$$

Similar to the treatment in many existing studies[25, 27, 28], the lift coefficient, and bank angle will be treated as control inputs.

3. Method design

3.1. Optimization via differential flatness

The trajectory optimization is formulated by using Differential Flatness. Flatness is a property of systems that extends the controllability concept of linear systems to nonlinear systems. In a flat system, both state and control can be represented by a set of flat outputs (which can be hypothetical) and their derivatives. The advantage of the differential flatness approach lies in its ability to simplify complex nonlinear control problems into relatively more manageable trajectory tracking problems.

Focusing on the characterization of the flight trajectories, we use Fourier series as basis functions to represent the flight trajectories, which are infinitely differentiable. There are several advantages of using Fourier series as basis functions: 1) it allows us to use a very limited number of parameters (i.e., magnitudes and phases of basis functions) to represent a complex trajectory; 2) these parameters have physical meaning related to the overall shape of the trajectory; and 3) they are continuously differentiable, which ensures smoothness of optimal trajectories and makes it easy to find their derivatives. A cyclic loitering trajectory can be represented as:

$$\begin{cases} x = a_{x,0} + \sum_{i=1}^{M} a_{x,i} \sin\left(\dfrac{2\pi i t}{t_{\mathrm{f}}} + \theta_{x,i}\right) \\ y = a_{y,0} + \sum_{i=1}^{M} a_{y,i} \sin\left(\dfrac{2\pi i t}{t_{\mathrm{f}}} + \theta_{y,i}\right) \\ z = a_{h,0} + \sum_{i=1}^{M} a_{z,i} \sin\left(\dfrac{2\pi i t}{t_{\mathrm{f}}} + \theta_{z,i}\right) \end{cases} \tag{10}$$

where t_{f} is the time for one cycle of flight, $a_{k,i}$ are the coefficients represent the magnitude of the corresponding frequency components with $k = x, y, z$, and $i = 0, \cdots, M$, $\theta_{k,i}$ are the phase angles of the different frequency components. The parameters to be optimized form a decision vector given as:

$$X = \left[a_{x,0}, \cdots, a_{x,M}, a_{y,0}, \cdots, a_{y,M}, a_{z,0}, \cdots, a_{z,M}, \theta_{x,0}, \cdots, \theta_{x,M}, \theta_{y,0}, \cdots, \theta_{y,M}, \theta_{z,0}, \cdots, \theta_{z,M}, t_f\right] \tag{11}$$

According to Eq.(10), the derivatives of x, y, and z can be obtained as:

$$\begin{cases} \dot{x} = \dfrac{2\pi i a_{x,i}}{t_\mathrm{f}} \cos\left(\dfrac{2\pi i t}{t_\mathrm{f}} + \theta_{x,i}\right) \\ \dot{y} = \dfrac{2\pi i a_{y,i}}{t_\mathrm{f}} \cos\left(\dfrac{2\pi i t}{t_\mathrm{f}} + \theta_{y,i}\right) \\ \dot{z} = \dfrac{2\pi i a_{h,i}}{t_\mathrm{f}} \cos\left(\dfrac{2\pi i t}{t_\mathrm{f}} + \theta_{z,i}\right) \end{cases} \tag{12}$$

Using $\{x, y, z\}$ in Eq.(12) as flat output, then we can solve for the other states and controls from Eq.(9) as:

$$V_\mathrm{a} = \sqrt{(\dot{x} - w_x)^2 + \dot{y}^2 + \dot{z}^2} \tag{13}$$

$$\gamma = \mathrm{asin}\left(-\frac{\dot{z}}{v_\mathrm{a}}\right) \tag{14}$$

$$\psi = \mathrm{atan}\left(\frac{\dot{y}}{\dot{x} - w_x}\right) \tag{15}$$

$$\phi = a\mathrm{tan}\left(\frac{v_\mathrm{a}\cos\gamma\,\dot{\psi} - \dot{w}_x \sin\psi}{v_\mathrm{a}\dot{\gamma} - g\cos\gamma - \dot{w}_x \sin\gamma\cos\psi}\right) \tag{16}$$

$$C_\mathrm{L} = \frac{-m(v_\mathrm{a}\dot{\gamma} - g\cos\gamma - \dot{w}_x \sin\gamma\cos\psi)}{0.5\rho S V_\mathrm{a}^2 \cos\phi} \tag{17}$$

It worth noting that, V_a, γ, and ψ can be exactly solved from the first three equations of Eq.(9); however, the problem is over constraint when we solve for two controls from there equations. Here, a virtual thrust T is defined such that:

$$T = m\dot{V}_\mathrm{a} + D - mg\sin\gamma + m\dot{w}_x \cos\gamma\cos\psi \tag{18}$$

This virtual thrust controls the speed of the aircraft. Constraints will be imposed on this thrust force to prevent the aircraft from changing speed too fast.

After solving the inverse dynamics problem, we can establish an optimization problem for dynamic soaring trajectories, with the objective of minimizing the energy expended by virtual thrust while adhering to a set of constraints. These constraints encompass various considerations, such as parameter limits,

$$\begin{cases} a_{x,\min} \le a_{x,i} \le a_{x,\max} \\ a_{y,\min} \le a_{y,i} \le a_{y,\max} \\ a_{z,\min} \le a_{z,i} \le a_{z,\max} \\ \theta_{x,\min} \le \theta_{x,i} \le \theta_{x,\max} \\ \theta_{y,\min} \le \theta_{y,i} \le \theta_{y,\max} \\ \theta_{z,\min} \le \theta_{z,i} \le \theta_{z,\max} \\ t_{\mathrm{f},\min} \le t_\mathrm{f} \le t_{\mathrm{f},\max} \end{cases} \tag{19}$$

State constraints:

$$\begin{cases} x_\min \le x \le x_\max \\ y_\min \le y \le y_\max \\ z_\min \le z \le z_\max \\ V_{\mathrm{a},\min} \le V_\mathrm{a} \le V_{\mathrm{a},\max} \\ \gamma_\min \le \gamma \le \gamma_\max \\ \psi_\min \le \psi \le \psi_\max \end{cases} \tag{20}$$

Control constraints:

$$\begin{cases} C_{\mathrm{L},\min} \le C_\mathrm{L} \le C_{\mathrm{L},\max} \\ \phi_\min \le \phi \le \phi_\max \\ T_\min \le T \le T_\max \end{cases} \tag{21}$$

where $[\cdot]_\min$ and $[\cdot]_\max$ represent the minimum and maximum values of the variables. Both the state and control and virtual thrust constraints are enforced for the whole trajectory. The other type of constraints depends on the problem of interest and will be introduced in section IV. There are many algorithms and software packages (e.g., OptimTraj [29], FALCON.m [30], and GPOPS-II [31]) that can help solve nonlinear optimal control problems. In this paper, the optimization problem was tackled using DRL with PPO2 algorithm, while also comparing the results with those obtained using NP method with interior point algorithm. Further details will be provided in Section IV. Next, a brief overview of the PPO2 method will be presented.

3.2. Optimization via PPO2

PPO2 is a policy-based approach with an actor-critic architecture. Here, the actor and critic are two neural networks characterized by parameters θ^u and θ^Q, respectively. The actor network's parameters θ^u are utilized to approximate the policy function $\pi(\mathrm{a}|\mathrm{s},\theta^u)$, while the critic network's parameters θ^Q are employed to estimate the value function V_s. Specifically, the actor's policy function, is modeled as a normal distribution, yielding mean and variance outputs from the actor network, denoted as u and δ. During policy execution, actions are stochastically sampled from this normal distribution, and the agent executes the corresponding action. Subsequently, the critic network assesses the state, represented as a scalar, to evaluate the state post-execution of the action.

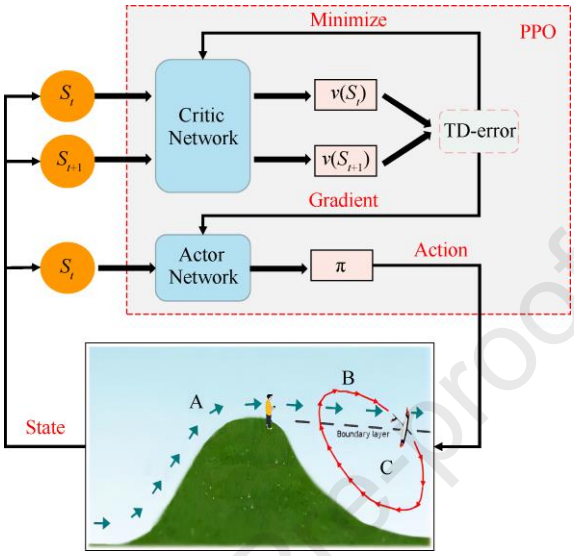


Fig. 3. Reinforcement learning trajectory optimization.

This paper proposes a method for formulating the gliding path optimization problem as a Markov Decision Process (MDP) that accounts for environmental uncertainties, and employs the PPO2 algorithm—a sophisticated dynamic interactive learning technique—to address the optimization of gliding paths in localized small areas. This approach is detailed and explained in Fig. 3. Assuming that the agent's next state depends solely on its current state, not its previous state (Markov property) denoted as $\mathcal{P}[s_{t+1}|s_t] = \mathcal{P}[s_{t+1}|s_1, s_2, \cdots, s_t]$, where s_i is the state of time i, $i \in [1,2,\cdots,t+1]$. The transition between states is governed by a state transition probability matrix $\mathcal{P}$, and the efficacy of decisions is gauged by rewards R. The parameters of the target deep neural network are periodically updated. Under policy π, the actor network is trained to maximize cumulative rewards by updating its policy parameters θ^u using the gradient ascent method with a weighting coefficient of $TD-\mathrm{error}$, while the critic network parameters are updated with the objective of minimizing $TD-\mathrm{error}$. To circumvent the "positive number trap" issue of the Policy Gradient (PG) algorithm, we introduce a weighting coefficient, which is formulated as follows,

$$TD-\mathrm{error} \;=\; \gamma v(s_{t+1}) + R_{t+1} - v(s_t) \tag{22}$$

$$\begin{aligned} v_\pi(s) = E_{s_t s_{t+1},\ldots}\big(R_{t+1} + \gamma G(s_{t+1})\big) \;&= E_{s_t}\Big(R_{t+1} + \gamma E_{s_{t+1},\ldots}\big(G(s_{t+1})\big)\Big) \;= E\big(R_{t+1} + \gamma v(s_{t+1})\big) \\ &= \sum_{a\in A}\pi(a|s)\Big(R_s^a + \gamma \sum_{s_{t+1}\in S} P_{s_t s_{t+1}}^a v_\pi(s_{t+1})\Big) \end{aligned} \tag{23}$$

where A is the action space, S is the state space, s is the input observation, $u(s) \in A$ is the model's predicted mean action, and $\delta(s) \in \mathbb{R}^A$ is the standard deviation. The probability density function of the diagonal Gaussian distribution is expressed as follows,

$$\mathcal{P}(a;u,\delta) = \frac{1}{\sqrt{(2\pi)^{|A|}\prod_{i=1}^{|A|}\delta_i^2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{|A|}\frac{(a_i-u_i)^2}{\delta_i^2}\right) \tag{24}$$

where,$a \in A$, $u = (u_1, u_2, \cdots, u_{|A|})$. Consequently, the logarithmic probability of the action a is:

$$\begin{aligned}&\log\mathcal{P}(a;u,\delta)\\&=\log\frac{1}{\sqrt{(2\pi)^{|A|}\prod_{i=1}^{|A|}\delta_i^2}}-\frac{1}{2}\sum_{i=1}^{|A|}\frac{(a_i-u_i)^2}{\delta_i^2}=-\frac{1}{2}\log\left((2\pi)^{|A|}\right)-\frac{1}{2}\sum_{i=1}^{|A|}\log(\delta_i^2)-\frac{1}{2}\sum_{i=1}^{|A|}\frac{(a_i-u_i)^2}{\delta_i^2}\\&=-\frac{1}{2}|A|\log(2\pi)-\frac{1}{2}\sum_{i=1}^{|A|}\log(\delta_i^2)-\frac{1}{2}\sum_{i=1}^{|A|}\frac{(a_i-u_i)^2}{\delta_i^2}\end{aligned}\tag{25}$$

The rate of data acquisition fails to match the training pace of the network is a bottleneck issue for reinforcement learning. To address this, an experience replay mechanism is proposed. It involves constructing a dataset $(s_t, a, s_{t+1}, R_{t+1}, info)$, where s_t and s_{t+1} represent the states at respective time steps t and $t+1$, while a and R_{t+1} signify the actions and rewards executed from one state to the next, along with a termination flag $info$ for each training episode. During each interaction between the agent and the environment, these datasets are stored in a replay buffer. Once a certain amount of data is collected, a batch of data is extracted for neural network training. Following training, the game continues, and the newly generated data is added to the replay buffer. This enables the utilization of previously generated data for training, significantly enhancing training speed while reducing over-fitting issues caused by excessive reliance on local experiences during the training process.

The experience replay mechanism effectively addresses the mismatch between data collection rates and the training speed of deep learning networks, significantly improving the efficiency of data utilization. However, it also introduces a challenge: ensuring consistency between data generated by an old policy and the updated current policy when the former is used to inform the latter. To address this challenge, this paper introduces an important sampling technique. This technique adjusts the weight of samples by calculating the ratio of probabilities of taking a specific action under different policies, thereby reducing sampling bias, improving the accuracy of policy estimation, and ensuring the compatibility of data across different policies.

To update policy $\mathcal{P}$ using data sampled from policy B, the weight coefficient can be calculated with the following formula:

$$iw = \mathcal{P}(a)/B(a)\tag{26}$$

Here, policy $\mathcal{P}$ and policy B represent the current policy and the old policy used to generate data, respectively, while action a is the action considered under both policies.

During the interaction between the agent and the environment, reinforcement learning relies on a reward feedback mechanism to identify and select the optimal actions. This mechanism guides the achievement of predetermined goals by maximizing the average reward. Designing an appropriate reward function is crucial for reinforcement learning, as it directly determines the effectiveness of training and the precision of the final outcomes. In this study, the design of the reward function has focused on two key priorities: first, ensuring that the aircraft's flight status complies with specific boundary constraint conditions; and second, minimizing the thrust work of the unmanned aerial vehicles (UAVs). Taking into account these considerations, the reward function is designed as follows:

$$R = R_{\text{over}} + R_{\text{power}}\tag{27}$$

where R_{over} ensures that the optimized trajectory meets the boundary constraint conditions, and R_{power} minimizes the thrust work through these constraints.

Given the transmission rate limitation of command sending during flight, the trajectory $\mathcal{L} = f(x,y,z)$ is first discretized with a time interval of Δt. After discretization, the number of discrete points within one cycle is denoted as $N = \frac{t_{\text{f}}}{\Delta t}$. To ensure that the state of the UAVs remains within the constraint range at each discrete point, the article uses the quantity of states exceeding the constraints, denoted as N_{over}, as a penalty term incorporated into the reward function：

$$R_{\text{over}} = \frac{t_{\text{f}}}{\Delta t} - N_{\text{over}}\tag{28}$$

Assuming that the motion of the UAVs within each time interval is uniform, the approximation of the thrust doing work can be expressed as：

$$E = E_1 + E_2 + \cdots + E_{N-1} = \sum_{i=1}^{N} T_i v_{a_i} \Delta t\tag{29}$$

where T_i, E_i and v_{a_i} are thrust, power of thrust and airspeed at the step i,respectively. Δt is a constant time interval.

Within the same time interval, even if the thrust work of the UAVs remains constant, the energy efficiency may vary due to differences in the UAVs' flight distances. Therefore, in this paper, to provide a more precise description of energy consumption, a parameter $P = \frac{E}{\Delta \mathcal{L}}$ is introduced to denote the energy consumption rate per unit distance. Assuming

the UAVs undergoes uniform motion within each time interval, denoted as $S = v_{a_i}\Delta t$, P can be derived as,

$$P = \frac{E_1}{\Delta\mathcal{L}_1} + \frac{E_2}{\Delta\mathcal{L}_2} + \cdots + \frac{E_{N-1}}{\Delta\mathcal{L}_{N-1}} = \sum_{i=1}^{N-1} \frac{T_i v_{a_i}\Delta t}{v_{a_i}\Delta t} = \sum_{i=1}^{N} T_i \tag{30}$$

where $\Delta\mathcal{L}_i$ is the length of the flight at time step i.

Summing up, the constraint conditions R_{power} is designed as follows：

$$R_{\text{power}} = \xi \times \mathcal{A}^P \tag{31}$$

where, R_{power} is a decreasing function that diminishes as the thrust work increases, while ξ and $\mathcal{A}$ are two hyperparameters, $\xi > 0$ and $0 < \mathcal{A} < 1$.

4. Simulation

In this section, the performance of the proposed method is evaluated by numerical simulations based on the small glider SBXC[32]. The parameters of the aircraft are given in Table 1. Similar to the albatross parameters, those of the glider pertain solely to the simplified 3-degree-of-freedom (DOF) motion equations, utilized for trajectory optimization and energy harvesting analysis.

Table 1
The parameters of glider.

Parameter	Explanation	Value	Units
m	Vehicle mass	5.443	kg
b	Wing span	4.32	m
S	Wing reference area	0.957	m^2
C_{D_0}	Parasitic drag coefficient	0.017	-
K	Induced drag factor	0.0192	-
$n_{\max}$	Maximum lift-drag ratio	3	-

The state vector encompasses position coordinates, airspeed, flight path angle, heading angle, and wind speed, represented by $s_t = [x_t, y_t, z_t, v_{a_t}, \gamma_t, \psi_t, w_{x_t}]$. And the constrains related to these variables are summarized in Table 2 Notice that for the DRL, the initial state of the UAV is randomly sampled from a uniform distribution within the specified ranges in Table 2.

Table 2
The constrains of the variables.

Parameter	Description	Value or interval	Units
x	Position in the *X*-axis direction	$[-1000, 1000]$	m
y	Position in the *Y*-axis direction	$[-1000, 1000]$	m
z	Position in the *Z*-axis direction	$[10, 360]$	m
a_x	The amplitude coefficients of the x	$[-1000, 1000]$	-
a_y	The amplitude coefficients of the y	$[-1000, 1000]$	-
a_z	The amplitude coefficients of the z	$[-1000, 1000]$	-
θ_x	The phase position coefficients of the x	$[-180, 180]$	deg
θ_y	The phase position coefficients of the y	$[-180, 180]$	deg
θ_z	The phase position coefficients of the z	$[-180, 180]$	deg
t_f	Cycle time	$[0, 200]$	s
v_a	Airspeed	$[9.54, 73.2]$	m/s
ψ	Flight path angle	$[-180, 180]$	deg
γ	Heading angle	$[-60, 60]$	deg
ϕ	Bank angle	$[-60, 60]$	deg
C_L	Lift coefficient	$[-0.3, 1.2]$	

The hyper-parameters used in DRL training in this work are outlined in Table 3. It's important to note that the

adjustment of hyper-parameters significantly impacts the performance of PPO2 algorithm. However, there is currently no unified rule for adjusting hyper-parameters. Therefore, we fine-tuned these hyper-parameters through several trial and error test to solve the trajectory optimization problem addressed in this paper.

Table 3
The hyper-parameters of DRL.

Parameter	Description	Value
ζ	Discount factor	0.99
ϵ	Entropy coefficient	0.01
λ	Learning-rate	2.5e-4
V_{coef}	Value function coefficient	0.5
η	Discount factor	0.95
ϖ	Ration clipping	0.2

The following are the results for the initial guesses mentioned in the problem description for the Fourier Harmonics of $M = 3$ and nodes of $N = 1500(N = 500\ M)$. The convergence pattern of the average reward and the sum of thrust per cycle in training the DRL is shown in Fig. 4. From this paper, it can be clearly observed that the average reward converges to the steady-state within 1×10^6, meanwhile the sum of thrust per cycle remains stable around $3000\ N$.

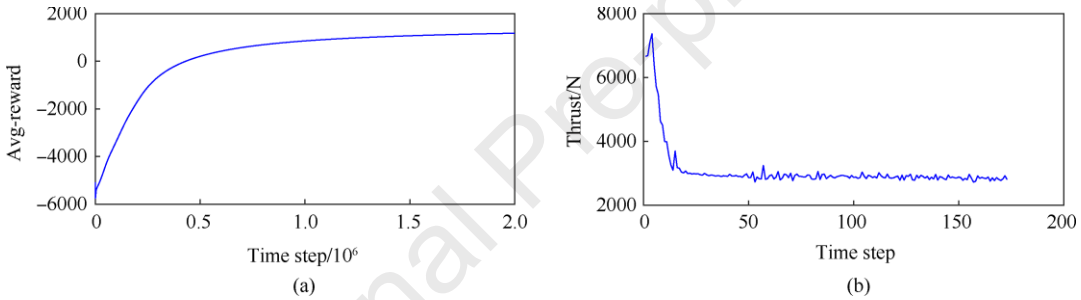


Fig. 4. Learning curve of the DRL.

As shown in Fig. 5, the variation of thrust over time during the last 10 training sessions of the DRL method is illustrated. It can be observed that the thrust within each cycle achieves good stability and exhibits a symmetrical distribution. Besides，a comparison is made between DRL and NP method. the positional results over time during one cycle are shown in Fig. 6. It is evident that while both methods display overall consistent trends in positional coordinates within one cycle, there are differences, primarily in the extent of the flight space. As Table 4, the height differences between the periodic trajectories obtained by the two methods are respectively 31.5973 m and 33.6542 m, indicating minor disparities. Additionally, it is apparent that the flight trajectory within one cycle can be divided into four phases: upwind climb, high-altitude turn (from upwind to downwind), downwind descent, and low-altitude turn (from downwind to upwind). This observation is consistent with the findings of Richardson [33], Sachs[34], and other scholars.

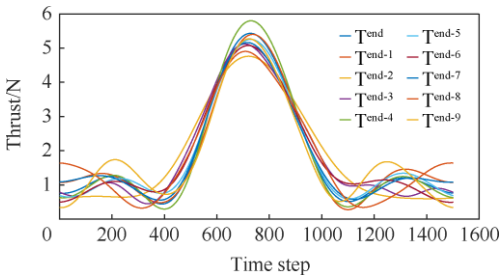


Fig. 5. Thrust of the last 10 episodes.

Table 4
Comparison results of position.

Parameter	NP	DRL
Distance along the *X*-axis	206.0883 m	76.7015 m
Distance along the *Y*-axis	400.0075 m	57.1816 m
Distance along the *Z*-axis	33.6542 m	31.5973 m

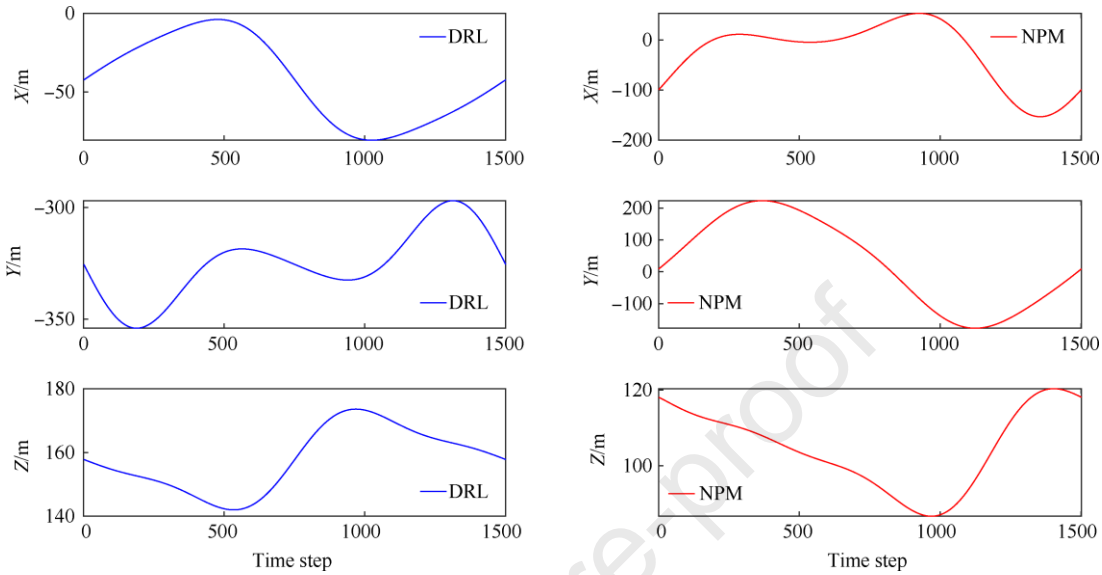


Fig. 6. Time history of position.

In Fig. 7, we present a comparative analysis of the temporal variation of airspeed, flight path angle, and heading angle over the course of a single cycle for the two methodologies in question. This visualization elucidates the dynamic behavior of these parameters, offering insights into the performance characteristics of each method. From the graph, it can be observed that the overall trends of airspeed and track angle are consistent, but there are significant differences in the heading angle. Table 5 shows that the maximum values of these parameters within one cycle are respectively $[12.9399\ \mathrm{m/s}, 1.9530°, 10.4976°]$ and $[17.4243\ \mathrm{m/s}, 3.7454°, 70.1324°]$, with corresponding ranges of variation being $[3.1893\ \mathrm{m/s}, 6.0645°, 17.3776°]$ and $[8.7217\ \mathrm{m/s}, 9.0504°, 158.8662°]$. Compared to the NP method, the DRL method results in reductions of 63.4%, 33%, and 89.1% in the ranges of variation, respectively.

Table 5
Comparison results of flight parameters.

Parameter	NP	DRL
v_a difference	$8.7217\ \mathrm{m/s}$	$3.1893\ \mathrm{m/s}$
γ difference	$9.0504°$	$6.0645°$
ψ difference	$158.8662°$	$17.3776°$
Maximum v_a	$17.4243\ \mathrm{m/s}$	$12.9399\ \mathrm{m/s}$
Maximum γ	$3.7454°$	$1.9530°$
Maximum ψ	$70.1324°$	$10.4976°$

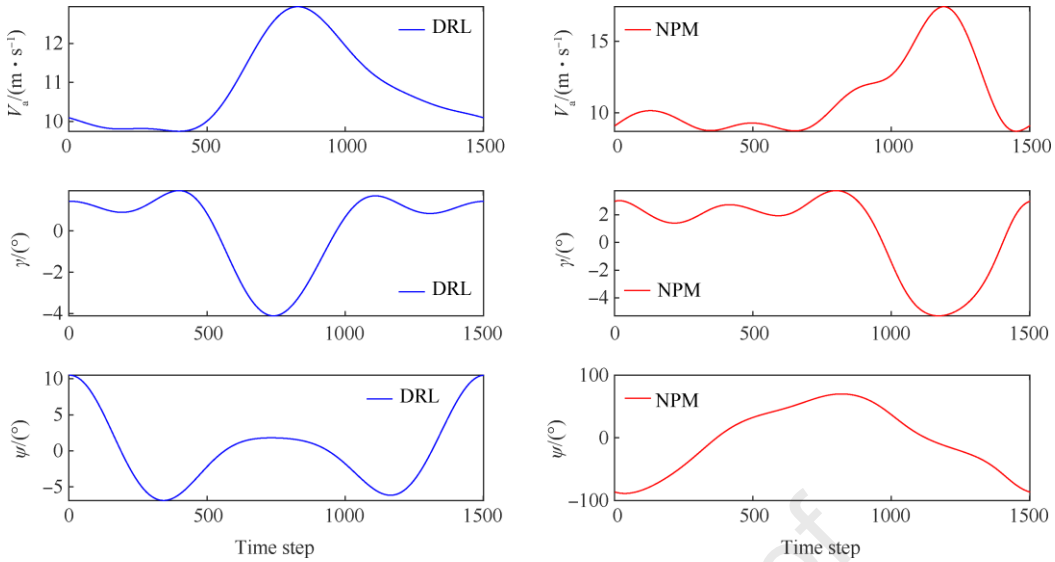


Fig. 7. Time history of flight parameters.

As depicted in Fig. 8 and Table 6, the lift coefficient and thrust show consistent variations between the two methods within a single cycle. Notably, each method's thrust and lift coefficient demonstrate opposing trends, in line with conventional expectations. Furthermore, the maximum control inputs for each method within one cycle are $[0.6280°, 0.9569, 5.4309N]$ and $[5.3335°, 1.1952, 8.2322N]$, with maximum ranges of control inputs being $[1.2466°, 0.4155, 4.9879N]$ and $[9.8184°, 0.8956, 8.2269N]$. Compared to the NP method, the DRL method reduces the ranges of variation in control inputs by 87.3%, 53.6%, and 39.4%, respectively, and decreases the required maximum thrust by 34%. It is evident that the control inputs obtained from the proposed method are more stable within one cycle.

Table 6
Comparison results of control parameters.

Parameter	NP	DRL
ϕ difference	9.8184°	1.2466°
C_{L} difference	0.8956	0.4155
T difference	8.2269N	4.9879N
Maximum ϕ	5.3335°	0.6280°
Maximum C_{L}	1.1952	0.9569
Maximum T	$8.2322N$	$5.4309N$

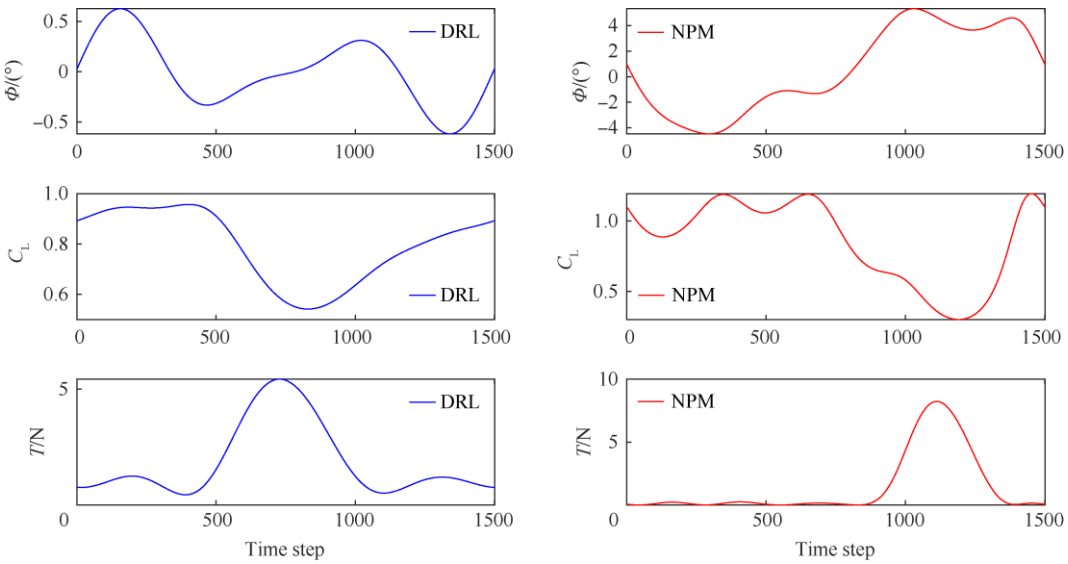


Fig. 8. Time history of control parameters.

As depicted in Table 7, Fig. 9 and Fig. 10, both methods produce trajectories resembling a figure "8" shape, meanwhile the thrust output and altitude changes within a single cycle are compared in Fig. 11, reveals that the thrust from each method reaches a peak, albeit at different times. The peak thrust from the DRL method is lower than that from the NP method and shows an approximate cyclic symmetry. Additionally, both methods display a consistent trend in altitude variations, but the trajectories of DRL shows a higher central height of 157.7954 mcompared to 103.5278 m for NP method which indicates that it operates in a more intense wind environment. The wind speed range for the two methods recorded in one cycle are 9.8229– 11.7837 m/s and 6.1943– 8.4326 m/s, respectively.

Table 7

Comparison results with NP.

Parameter	DRL	NP
Height center	157.7954 m	103.5278 m
Average thrust	1.8630N	1.4468N
Maximum wind	11.7837 m/s	8.4326 m/s
Minimum wind	9.8229 m/s	6.1943 m/s

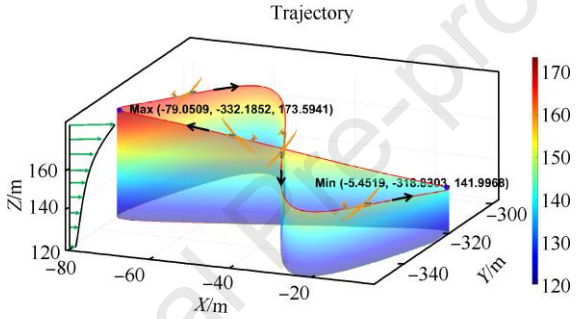


Fig. 9. The trajectory of DRL.

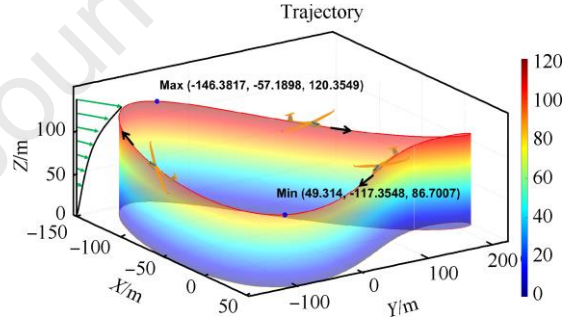


Fig. 10. The trajectory of NP method.

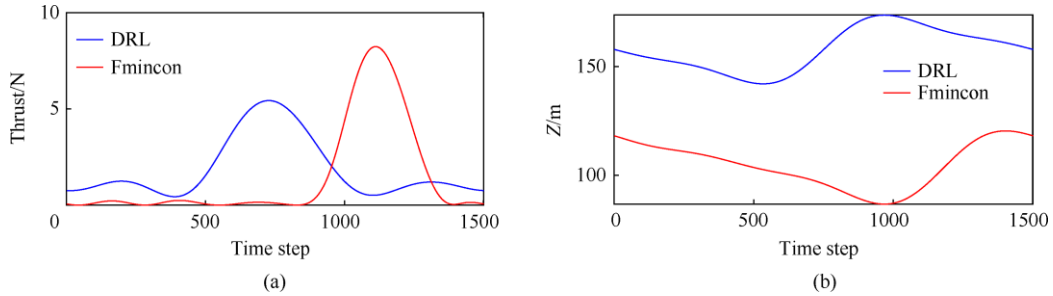


Fig. 11. Comparison of thrust and height.

The relevant research in this paper is based on the assumption that the wind field only exhibits gradient changes in the vertical direction and remains constant within the same horizontal plane regardless of position changes. To facilitate comparison, the paper adjusts the center points of trajectory projections on the horizontal plane to the origin [0,0] for each cycle. The relevant operational process is outlined below:

$$\begin{aligned} \hat{X}_i &= (\max(\boldsymbol{x_i}) + \min(\boldsymbol{x_i}))/2 \\ \breve{\boldsymbol{x}}_i &= \boldsymbol{x_i} - \hat{X}_i \\ \hat{Y}_i &= (\max(\boldsymbol{y_i}) + \min(\boldsymbol{y_i}))/2 \\ \breve{\boldsymbol{y}}_i &= \boldsymbol{y_i} - \hat{Y}_i \end{aligned} \tag{32}$$

where, max[·] and min[·] denote the maximum and minimum values of a variable, respectively.Additionally, $\hat{X}_i$ and $\hat{Y}_i$ represent the midpoints of the coordinate vector $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ in *x*-direction and *y*-direction, respectively, for one cycle of i episode. $\breve{\boldsymbol{x}}_i$ and $\breve{\boldsymbol{y}}_i$ are the coordinate vector after transforming. Through the aforementioned processing, the influence of varying initial positions within the initial horizontal plane on the comparison results is mitigated. Fig. 12 illustrates trajectories obtained under ten different initial states. It is evident that the trajectory planning method proposed in this paper can effectively accommodate diverse initial conditions and yield favorable optimization outcomes, thereby addressing the limitations arising from the heavy reliance of conventional numerical optimization methods on initial values.

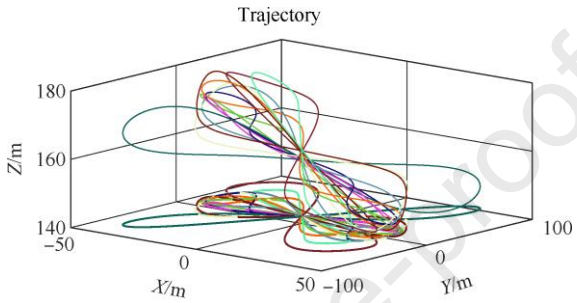


Fig. 12. The trajectory of DRL with different initial state.

5. Conclusions

In this paper, a differential flat model based deep reinforcement learning approach is proposed to iteratively optimizing and maintaining dynamic soaring trajectories. Initially, this method reformulates the traditional global optimization problem as iterative optimization through interactions between an agent and uncertain environment following Markov processes. Leveraging the principles of differential flatness, we utilize Fourier series base functions for trajectory modeling, aiming to minimize the number of parameters for optimization. Subsequently, a trajectory hyper-parameters solver utilizing DRL is proposed, with the goal of minimizing thrust work. The resulting trajectory agrees well with those in the published work, where a soaring trajectory consists of four stages: windward ascent, high-altitude turns, leeward descent, and low-altitude turns.

Compared to the results from the conventional nonlinear optimization, it reveals that the precision of the proposed method is on par with that of traditional nonlinear programming approaches. This observation lends support to the feasibility of employing reinforcement learning techniques for dynamic soaring trajectory planning. Notably, the proposed method is less sensitive to choices for initial values. This establishes a robust groundwork for further exploration into real-time navigation control of unmanned aerial vehicles during dynamic soaring.

Acknowledgments

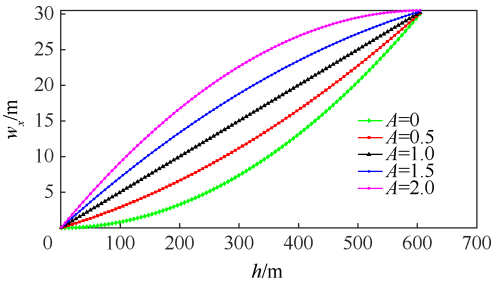
The authors wish to thank the support received by the National Natural Science Foundation of China (Nos. 52372398 & 62003272).

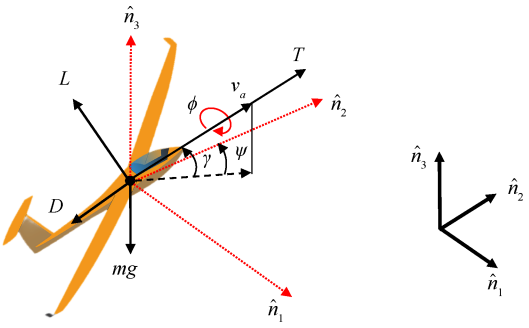
References

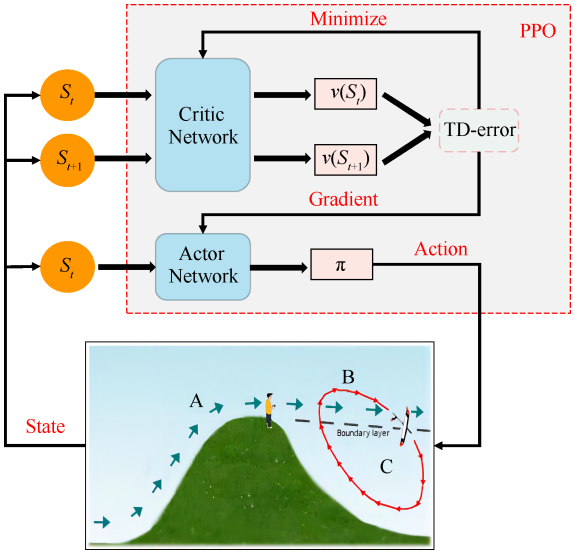
[1] Y. Huang, J. Chen, H. Wang, G. Su, A method of 3D path planning for solar-powered UAV with fixed target and solar tracking, Aerospace Science and Technology 92 (2019) 831-838.
[2] P. Panagiotou, I. Tsavlidis, K. Yakinthos, Conceptual design of a hybrid solar MALE UAV, Aerospace Science and Technology 53 (2016) 207-219.

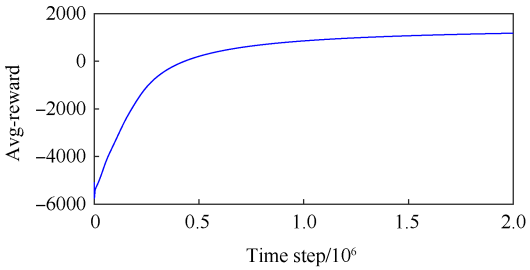
[3] Y. Yuan, Y.M. Deng, S.D. Luo, H.B. Duan, C. Wei, Hybrid formation control framework for solar-powered quadrotors via adaptive fission pigeon-inspired optimization, Aerospace Science and Technology 126 (2022).
[4] P.L. Richardson, How do albatrosses fly around the world without flapping their wings?, Prog Oceanogr 88(1-4) (2011) 46-58.
[5] H. Airy, The soaring of birds, Nature 28(709) (1883) 103-103.
[6] H. Weimerskirch, F. Bonadonna, F. Bailleul, G. Mabille, G. Dell'Omo, H.P. Lipp, GPS tracking of foraging albatrosses, Science 295(5558) (2002) 1259-1259.
[7] G. Sachs, K. Lesch, A. Knoll, Optimal control for maximum energy extraction from wind shear, Guidance, Navigation and Control Conference, 1989.
[8] J.P. Barnes, How flies the albatross–the flight mechanics of dynamic soaring, SAE Technical Paper, 2004.
[9] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, M. Goodrich, Autonomous Vehicle Technologies for Small Fixed-Wing UAVs, Journal of Aerospace Computing, Information, and Communication 2(1) (2005) 92-108.
[10] R. Bencatel, P. Kabamba, A. Girard, Perpetual Dynamic Soaring in Linear Wind Shear, Journal of Guidance, Control, and Dynamics 37(5) (2014) 1712-1716.
[11] Y. Pan, K. Wang, W. Zou, S. Bu, M. Zhou, N. Li, Dynamic Soaring Trajectory Optimization and Tracking with Adaptive Non-singular Fast Terminal Sliding Mode Control, Proceedings of 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022)2023, pp. 3114-3127.
[12] W.E. Shaw-Cortez, E. Frew, Efficient Trajectory Development for Small Unmanned Aircraft Dynamic Soaring Applications, J Guid Control Dynam 38(3) (2015) 519-+.
[13] C. Gao, H.H.T. Liu, Dubins path-based dynamic soaring trajectory planning and tracking control in a gradient wind field, Optimal Control Applications and Methods 38(2) (2016) 147-166.
[14] V. Shalumov, Cooperative online Guide-Launch-Guide policy in a target-missile-defender engagement using deep reinforcement learning, Aerospace Science and Technology 104 (2020).
[15] C. Peng, J. Ma, Online integral reinforcement learning control for an uncertain highly flexible aircraft using state and output feedback, Aerospace Science and Technology 109 (2021).
[16] G. Novati, L. Mahadevan, P. Koumoutsakos, Controlled gliding and perching through deep-reinforcement-learning, Physical Review Fluids 4(9) (2019).
[17] J. Zhao, J. Li, L. Zeng, Energy-Harvesting Strategy Investigation for Glider Autonomous Soaring Using Reinforcement Learning, Aerospace 10(10) (2023).
[18] G. Reddy, J. Wong-Ng, A. Celani, T.J. Sejnowski, M. Vergassola, Glider soaring via reinforcement learning in the field, Nature 562(7726) (2018) 236-239.
[19] Z.Y. Xi, D. Wu, W.N. Ni, X.P. Ma, Energy-Optimized Trajectory Planning for Solar-Powered Aircraft in a Wind Field Using Reinforcement Learning, Ieee Access 10 (2022) 87715-87732.
[20] R. Bencatel, A. Girard, M. Abdelhafiz, J. Sousa, Shear Wind Estimation, AIAA Guidance, Navigation, and Control Conference, 2011.
[21] W.J. Akhtar N, Cooke A, Wind shear energy extraction using dynamic soaring techniques, 47th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition, 2009.
[22] X. Shen, X. Zhu, Z. Du, Wind turbine aerodynamics and loads control in wind shear flow, Energy 36(3) (2011) 1424-1434.
[23] R.J. Gordon, Optimal dynamic soaring for full size sailplanes, (2006).
[24] I. Mir, A. Maqsood, S.A. Eisa, H. Taha, S. Akhtar, Optimal morphing – augmented dynamic soaring maneuvers for unmanned air vehicle capable of span and sweep morphologies, Aerospace Science and Technology 79 (2018) 17-36.
[25] Y.J. Zhao, Optimal patterns of glider dynamic soaring, Optimal Control Applications and Methods 25(2) (2004) 67-89.
[26] L.A. Weitz, Derivation of a Point-Mass Aircraft Model used for Fast-Time Simulation, MITRE Corporation (2015).
[27] G. Sachs, Minimum shear wind strength required for dynamic soaring of albatrosses, Ibis 147(1) (2004) 1-10.
[28] M. Deittert, A. Richards, C.A. Toomer, A. Pipe, Engineless Unmanned Aerial Vehicle Propulsion by Dynamic Soaring, Journal of Guidance, Control, and Dynamics 32(5) (2009) 1446-1457.
[29] M. Kelly, An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation, SIAM Review 59(4) (2017) 849-904.
[30] M.a.B. P. Rieck, Matthias and Gruter, Benedikt and Diepolder, Johannes and Piprek, "FALCON.m User Guide." nstitute of Flight System Dynamics, Technical University of Munich (2018).
[31] M.A. Patterson, A.V. Rao, GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming, ACM Transactions on Mathematical Software 41(1) (2014) 1-37.

[32] L.N.R. J, Autonomous soaring flight for unmanned aerial vehicles, 2011.
[33] P.L. Richardson, Upwind dynamic soaring of albatrosses and UAVs, Prog Oceanogr 130 (2015) 146-156.
[34] G.P. Sachs, Maximum Travel Speed Performance of Albatrosses and UAVs Using Dynamic Soaring, AIAA Scitech 2019 Forum, 2019.

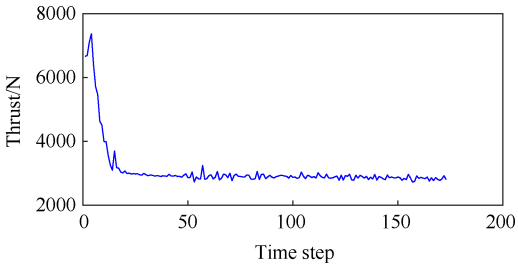




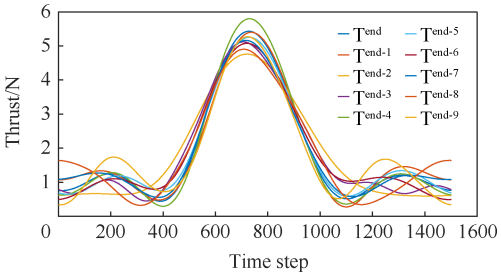


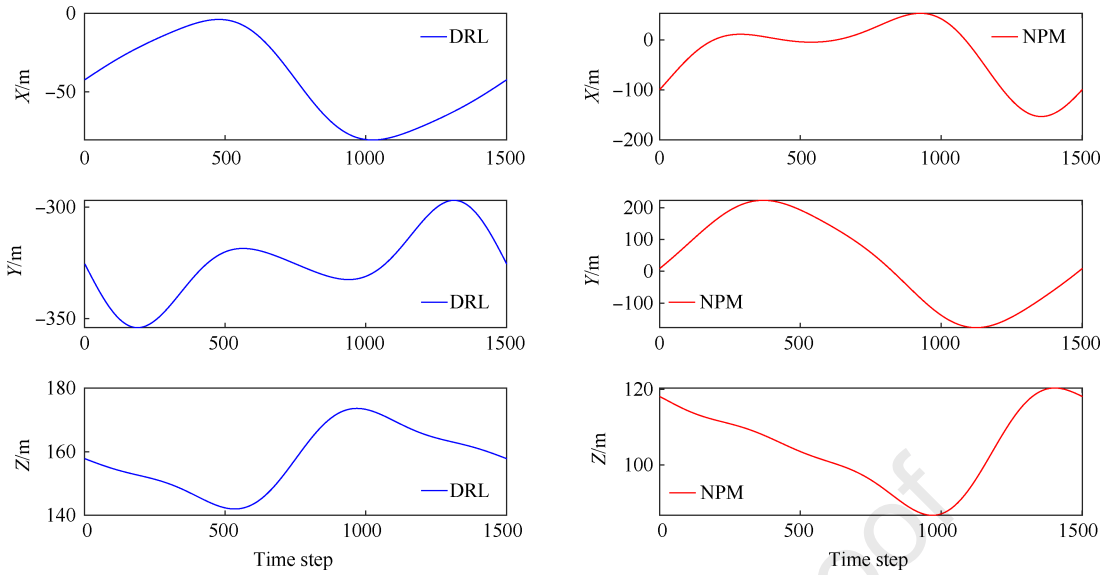


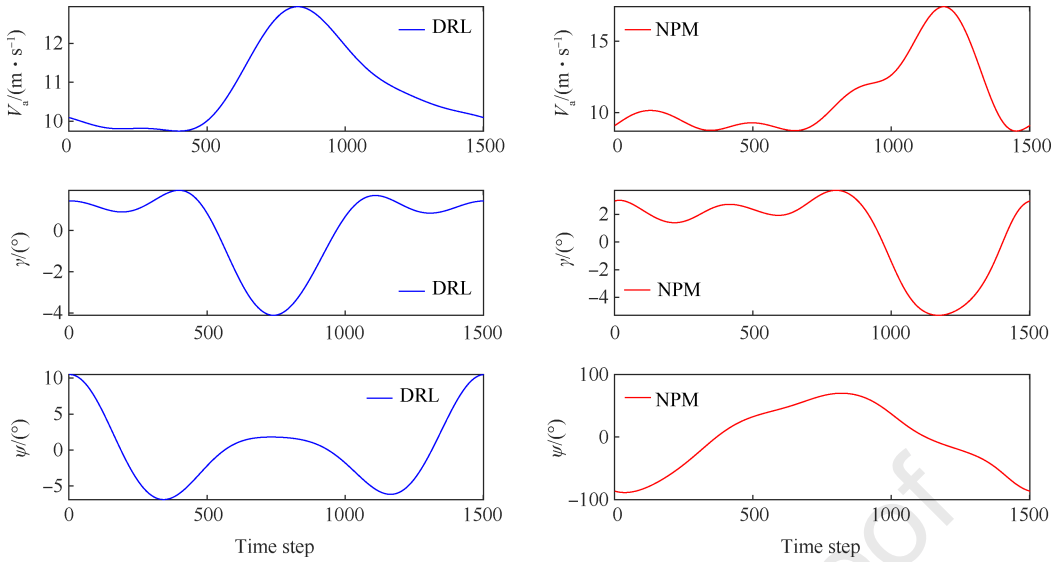
(a)

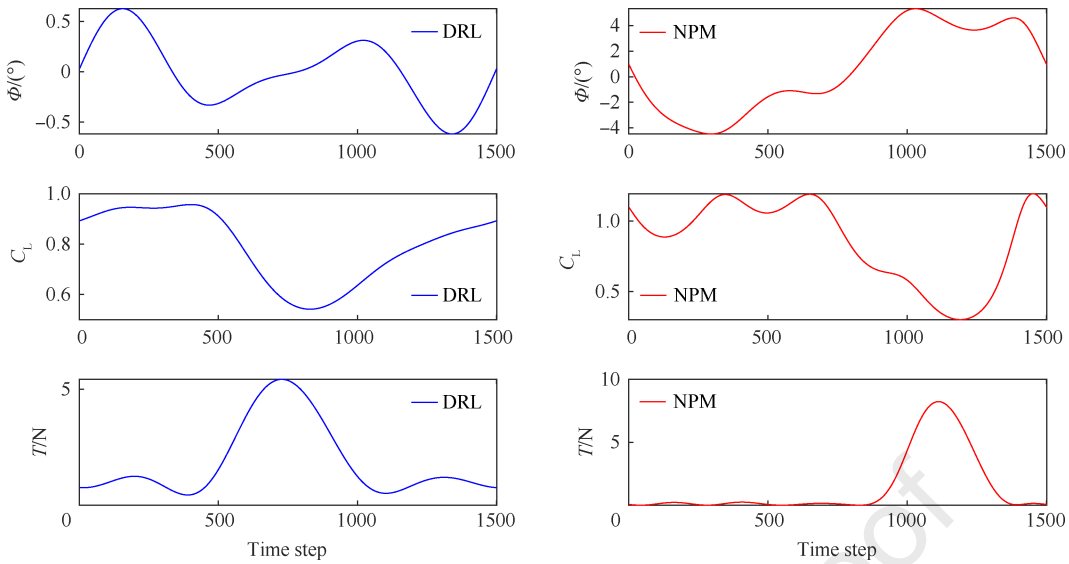


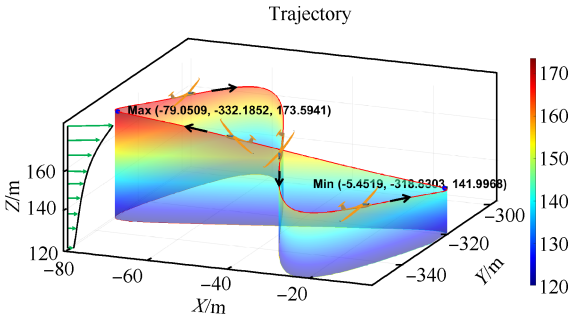
(b)

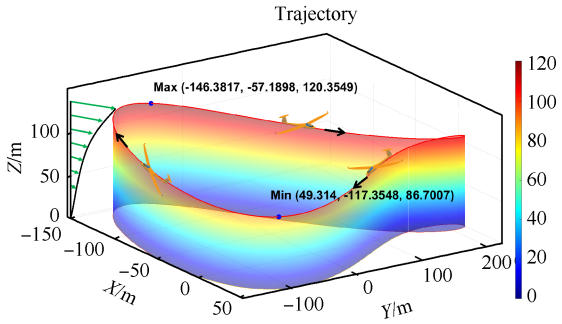


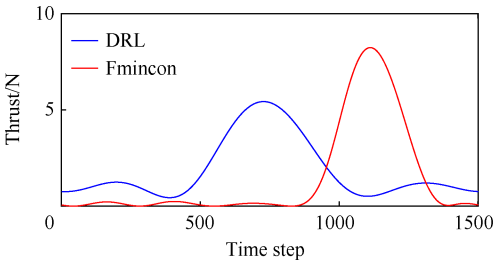




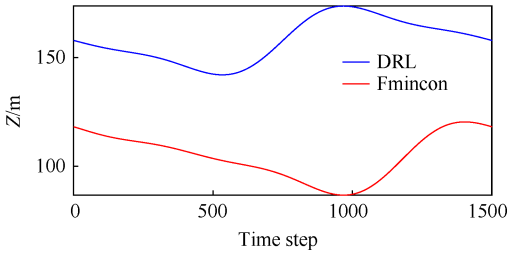




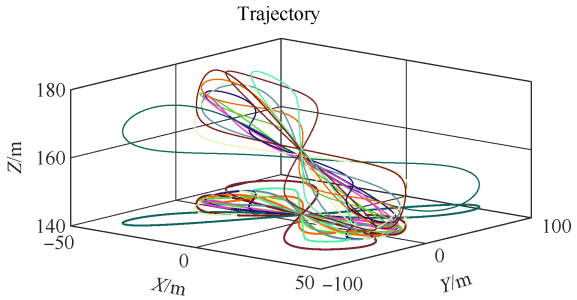




(a)



(b)



Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Ni Li reports financial support was provided by The authors wish to thank the support received by the National Natural Science Foundation of China (Nos. 52372398 & 62003272). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.