The Aeronautical Journal (2026), **130**, pp. 3197–3213
doi:10.1017/aer.2026.10205

RESEARCH ARTICLE

A triangular number-based reward function for trajectory planning of wing-in-ground crafts towards fast-response and low-deviation altitude changes

Huan Hu[1], Yajun Shi[1], Guiyong Zhang[2], Qian Zhang[1], Yiheng Wang[1] and Ji Zhang[2]

[1]China Ship Scientific Research Center, China
[2]State Key Laboratory of Structural Analysis, Optimization and CAE Software for Industrial Equipment, School of Naval Architecture Engineering, Dalian University of Technology, Dalian, China
Corresponding author: Yajun Shi; Email: shiyajun@wigcraft.cn

Received: 25 March 2026 UTC; **Revised:** 16 June 2026 UTC; **Accepted:** 17 June 2026 UTC

Keywords: deep reinforcement learning; reward function; rise time; trajectory planning; wing-in-ground craft

Abstract

Altitude adjustment for wing-in-ground (WIG) crafts during obstacle avoidance or mission execution necessitates a rapid response followed by precise stabilisation at the target altitude. Conventional approaches typically treat the minimisation of response time and the minimisation of deviation from the target altitude as two separate objectives. These are assigned weights, and their performance is balanced through iterative trial-and-error adjustment of these weights, ultimately finding a target trajectory after multiple optimisation runs. This paper proposes a novel reward function that yields, via a single deep reinforcement learning pass, a trajectory minimising both the response time and deviation for WIGs' altitude changes. Formally, this function bounds the reward value based on the deviation magnitude and incorporates triangular numbers to assess the response time. Its underlying principle ensures that the reward gain from reducing the response time outweighs the reward loss incurred by the potential increase in accumulated deviation resulting from that same reduction. The validity of this reward function is verified through mathematical derivation. A numerical example involving a WIG descending to a lower altitude is presented, using proximal policy optimisation. Comparative results against traditional multi-objective functions requiring extensive parameter tuning demonstrate that the proposed reward function directly produces the desired trajectory. This reward function provides a valuable reference for trajectory planning in second-order motion systems aiming to achieve minimised rise time and subsequent minimised integrated absolute error.

Nomenclature

WIG	wing-in-ground craft
IAE	integrated absolute error
AE	absolute error
RT	rise time
ST	settling time
OS	overshoot
ITAE	integrated time absolute error
ISE	integrated squared error
ITSE	integrated time squared error
SSE	steady-state error
PID	proportional-integral-derivative
m	the total steps of the trajectory or the number of trajectory points
h^*	the target altitude
c	chord length of the wing

© The Author(s), 2026. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

t_r	the step where the WIG reached the target altitude h^* for the first time
h_t	altitude of the trajectory point in the t-th step
h_0	the initial altitude
$r_r^{(t)}$	the reward of RT for the t-th step
$r_e^{(t)}$	the reward of AE for the t-th step
R_r	the cumulative rewards of RT for the trajectory
R_e	the cumulative rewards of AE for the trajectory
R	the total reward evaluated for the trajectory
$R_r^{(t)}$	the RT reward for the trajectory whose RT$= t$
$R_e^{(t)}$	the IAE reward for the trajectory whose RT$= t$
$R_r^\star$	the cumulative rewards of RT for the trajectory by the multi-objective method
$R_e^\star$	the cumulative rewards of AE for the trajectory by the multi-objective method
$R^\star$	the total reward evaluated for the trajectory by the multi-objective method
k	the varying weight value for the RT part of the multi-objective function
α	the varying weight for the original method to examine its invariance
F^i	the inertial frame
F^f	the WIG-fixed frame
F^b	the body frame
m_w	mass of the WIG
F	force on the WIG in the WIG-fixed frame F^f
M	moment on the WIG in the WIG-fixed frame F^f
u	velocity vector measured in the body frame F^b
$\dot{u}$	acceleration vector measured in the body frame F^b
g^n	acceleration of gravity in the inertial frame F^i
C_g^b	transformation matrix from F^f to F^b
V	velocity vector measured in the inertial frame F^i
I	moment of inertia

Greek symbol

λ	exponential base
ω	angular velocity vector measured in the body frame F^b
$\dot{\omega}$	angular acceleration vector measured in the body frame F^b

1.0 Introduction

As a transitional vehicle between hovercraft and seaplanes, wing-in-ground (WIG) crafts typically maintain low flight altitudes to utilise ground effect, which provides a high lift-to-drag ratio and enhanced payload capacity. This operational mode also offers the militarily valued advantage of low observability. When encountering obstacles, waves or mission-specific requirements, WIG crafts must adjust their altitude. The ability to rapidly reach a target altitude and maintain minimal deviation during subsequent stabilisation demonstrates superior control performance. This manoeuvre is illustrated in Fig. 1.

The manoeuvring of a WIG is similar to that of an aircraft, so its motion response is a typical second-order model. Regarding trajectory planning for flight altitude adjustments, the absolute error (AE) is commonly adopted as the basis for reward functions. AE quantifies the minimum distance between the current position and the target trajectory or altitude. By integrating the AE across multiple trajectory points, the integrated absolute error (IAE) is obtained, which serves as a metric for evaluating the overall deviation between the current and target trajectories. In studies on nonlinear control strategies [1], nonlinear adaptive control systems [2] and feedback control [3] for switching the WIG's flight altitude, there are distinctive second-order motion responses, with different performance of rise time (RT), settling time (ST) and overshoot (OS). There are also some studies on motion planning, where the reward function is designed with the help of integrated absolute error (IAE) to realise the WIG's trajectory planning, such as reducing the OS [4], stabilising the landing control [5, 6] or meeting multiple kinds

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

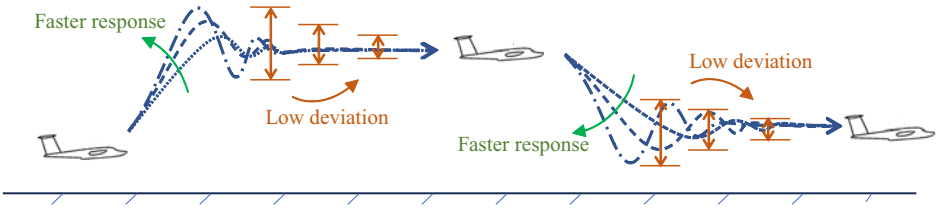


Figure 1. *The WIG's fast-response and low-deviation trajectory during altitude transition.*

of flight requirements [7]. However, taking a closer look at these studies, it can be found that minimising IAE cannot guarantee whether the WIG adjusts its altitude quickly enough. If the change of WIG's altitude is rapid, the WIG will reach the target altitude earlier; in other words, RT of the trajectory will be smaller. Then, the extent to which the WIG's flight exceeds the target altitude will be greater due to its inertia, leading to a larger OS. Next motion is that the WIG oscillates up and down and gradually approaches the target altitude. This process of convergence will be longer, resulting in a greater ST of the trajectory. Generally, IAE will be greater for a more zigzag trajectory. It can be seen that there is a contradictory relationship between RT and full-trajectory IAE.

Studies in recent years have incorporated various metrics, including IAE, RT and ST, as partial terms in trajectory adjustment. These terms are typically incorporated linearly into the reward function as penalty terms, with coefficients or exponents used to adjust the trajectory towards desired behaviours. For wind energy maximum power point tracking (MPPT), multi-objective genetic algorithms (GA) simultaneously minimised integrated time absolute error (ITAE), integrated squared error (ISE), IAE and integrated time squared error (ITSE) [8], while GA and ant colony optimisation were also employed to minimise ITAE [9, 10]. ITAE weights AE by time t. Formally, it assigns a weight of t to the AE at each time step. In essence, it treats AEs at different time steps as distinct objectives to be coordinated, with the importance of subsequent AEs increasing progressively with the elapsed time. This is equivalent to formulating a multi-objective function. In load frequency control, sunflower optimisation, teaching–learning-based optimisation and GA were compared with ITAE as the cost [11], and particle swarm optimisation (PSO)-tuned fuzzy controllers were evaluated via ISE, IAE, mean squared error (MSE) and ITAE [12]. To more explicitly balance transient specifications, many authors constructed composite objective functions. Ghosh et al. [13] built an objective combining ISE, OS and ST, via PSO; Samuk and Çakr [14] employed radial movement optimisation with a multi-variable objective that combined classical error integrals (ITAE, ISE, IAE, ITSE) with transient parameters (maximum OS and ST) using weight coefficients. Cárdenas et al. [15] compared four multi-objective PSO cost functions merging ST, OS and steady-state error (SSE). Weighted sums of IAE, ISE and ITAE were minimised via enhanced GA [16], while animated oat optimisation and PSO used IAE, ISE, ITAE and ITSE for inverter control [17]. Multi-objective PSO also simultaneously reduced ST and torque ripple [18]. Further works explicitly integrated OS, ST and RT into the fitness function for multi-objective GA [19], or modified classical error integrals by adding weighted ST, RT, OS percentage and SSE [20], by the method of GA. Time-domain objective functions aggregating maximum OS, SSE, RT and ST have been optimised by opposition-based hybrid manta ray foraging and simulated annealing [21], by PSO [22] and by an opposition-based weighted mean of vectors algorithm [23]. The Zwee-Lee Gaing (ZLG) function, which combines SSE, OS, ST and RT, was employed with the artificial rabbits optimiser [24] and with the equilibrium optimiser [25]. A custom fitness function comprising OS, RT, ST and steady-state error was designed for PSO and differential evolution [26]. PSO combined with gravitational search algorithm optimised fall time, symmetrical rise/fall times and propagation delays [27], and the rime optimisation algorithm tuned a delay-affected controller using a composite objective integrating IAE, ITAE, ISE, SSE, RT, ST and OS [28].

A similar weighted objective philosophy appears in reinforcement learning, where reward functions are designed as weighted sums of multiple terms to provide step-by-step feedback for learning an adaptive policy. Deep deterministic policy gradient (DDPG) has been applied to missile control with

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

a reward comprising minimum error (effectively IAE), excessive error and excessive deflection rate [29]. For an optoelectronic stabilisation platform, DDPG combined with a fractional-order proportional-integral-derivative (PID) controller used a reward considering accuracy (IAE-related), stability, response speed and energy consumption [30]. A comparative study of proximal policy optimisation (PPO), DDPG, advantage actor-critic (A2C) and twin-delayed deep deterministic policy gradient (TD3) for spacecraft attitude recovery designed an exponential signal reward with an accuracy term corresponding to an IAE-like measure and a rate penalty [31]. In reinforcement learning-based model predictive control, TD3 was embedded with a cost function whose first term stabilises stator currents (IAE minimisation) and second term penalises control increments [32]. TD3 was also used to tune cascaded proportional-integral (PI) controllers for a five-phase motor drive, with the reward implicitly minimising speed and torque errors, which essentially correspond to an IAE metric [33].

However, prior research lacks specific guidance on how to determine appropriate values for these coefficients or exponents. While most studies qualitatively analyse how these penalty terms influence the trajectory, there has been limited quantitative assessment of their effect. As a result, no precise method exists for determining the optimal coefficient or exponent values. In conventional approaches, metrics such as RT and IAE are combined into a multi-objective reward, with each term assigned an individual weight. The relative magnitudes of these weights are typically adjusted through trial and error to align the trajectory with the desired outcome. This process aims to reduce IAE or RT, ultimately bringing the trajectory closer to the target altitude or allowing the WIG to reach the target altitude more quickly. However, this method lacks precision and requires iterative adjustments. For trajectory planning, especially in obstacle scenarios, the goal is to minimise RT first and then approach the target altitude as closely as possible. Designing a reward function that directly achieves this priority remains a significant engineering challenge.

The innovation of this paper is to propose a reward function for trajectory planning, which satisfies the preferential minimisation of RT and subsequent minimisation of IAE. The engineering contribution is directly guiding the WIG to change its altitude as quickly as possible, and then approach the target altitude as fast as possible. When evaluating a trajectory where there are remaining n steps after the WIG reaches the target altitude, the IAE-related part of the reward function evaluates IAE for the last n steps, while the RT-related part is assigned based on the maximum value of IAE. If the RT can be reduced by one step, in other words, if the WIG reach the target altitude one step ahead, the RT-related reward will increase the maximum value of IAE, so as to drive the WIG to the target altitude in advance. In this way, the RT-related item can be expressed as a triangular number, i.e. $n^*(n+1)/2$, and the feasibility of the method is demonstrated by mathematical reasoning.

In the following sections, the construction of the proposed reward function is demonstrated; then the model and optimisation methods which are used for its validation are introduced. Next, based on the results obtained by the proposed reward function and the conventional multi-objective reward function, comparison and further analysis are conducted. Finally, the conclusion is drawn.

2.0 Reward function

2.1 Form

Assume that the WIG traverses a trajectory consisting of m steps and has not reached the target altitude h^* by the t_r-th step. The whole trajectory can be divided into two parts by the step t_r, as illustrated in Fig. 2. To evaluate an optional trajectory, two kinds of reward are calculated in each step, which are named as $r_r^{(t)}$ representing the reward of RT and $r_e^{(t)}$ representing the reward of AE, expressed as:

$$r_e^{(t)} = \begin{cases} 0, & \text{if } t < t_r \\ 1 - \left| \dfrac{h_i - h^*}{h^* - h_0} \right|, & \text{if } t \geqslant t_r \end{cases} \tag{1}$$

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

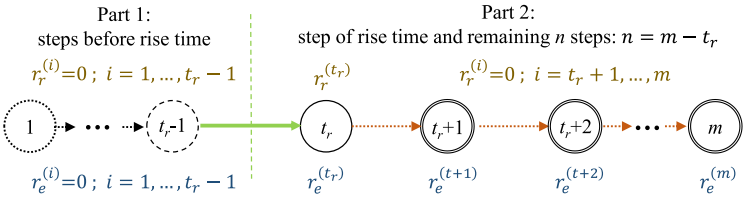


Figure 2. *The division of the whole trajectory and reward for different steps.*

$$r_r^{(t)} = \begin{cases} 0, & \text{if } t \neq t_r \\ 1 + \dfrac{n * (n+1)}{2}, \ n = m - t, & \text{if } t = t \end{cases} \tag{2}$$

where h_i is the WIG's flight altitude at the i-th step. h_0 is the initial altitude at the beginning of the trajectory, where the WIG starts to operate. n is the number of remaining steps after RT occurs, i.e., steps $(t+1)$ to m, so $n = m - t$. For the whole trajectory, these two kinds of corresponding cumulative rewards are R_r and R_e, respectively, and their sum is the reward evaluated for the trajectory, given as:

$$\begin{aligned} R_r &= \sum_{t=1}^{m} r_r^{(t)} \\ R_e &= \sum_{t=1}^{m} r_e^{(t)} \\ R &= R_e + R_t \end{aligned} \tag{3}$$

Rewards for parts of a trajectory are also shown in Fig. 2. $r_e^{(t)}$ are assigned as 0 for steps from the first to the (t_r-1)-th, where RT has not occurred. With the exception of the t_r-th step, $r_t^{(t)}$ of all other steps are set to 0.

The process by which the reward function assesses the reward of an available trajectory where the WIG adjusts its altitude to a higher one is as follows, and the algorithm operates as 9(')@, whose flowchart is shown in Fig. 3. It proceeds in the forward direction from the initial state to the end of the trajectory, with the RT initially assumed to occur at the final step, i.e. m-th step. When the WIG's flight altitude exceeds the target altitude for the first time at an intermediate step, the occurrence of the RT is identified, and the corresponding t_r for that trajectory is thereby determined. In the current study, which considers the case of the WIG reducing its flight altitude, the condition $h* > h_t$ is used to determine whether the RT has occurred.

(1) Give the trajectory with the altitude of each trajectory point h_t, $t = 1, 2 \ldots m$.

(2) Initially, the order of the current step, t=1. Initially, the total RT-related and total IAE-related reward, $R_r = 0$ and $R_e = 0$.

(3) Assume the rise time occurs at the last step, as $t_r = m$.

(4) If the current step t is after t_r, meaning that $t > t_r$, the current step RT reward $r_r^{(t)}$ is assigned 0, and the current step AE reward $r_e^{(t)}$ is calculated by Equation (1) based on the current flight altitude h_t.

(5) If $t < t_r$ and h_t is higher than the target altitude h^*, t_r is replaced by t. $r_r^{(t)}$ and $r_e^{(t)}$ are calculated by Equations (1) and (2), respectively.

(6) If $t < t_r$ and $h_t < h^*$, $r_r^{(t)}$ and $r_e^{(t)}$ are both assigned 0.

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

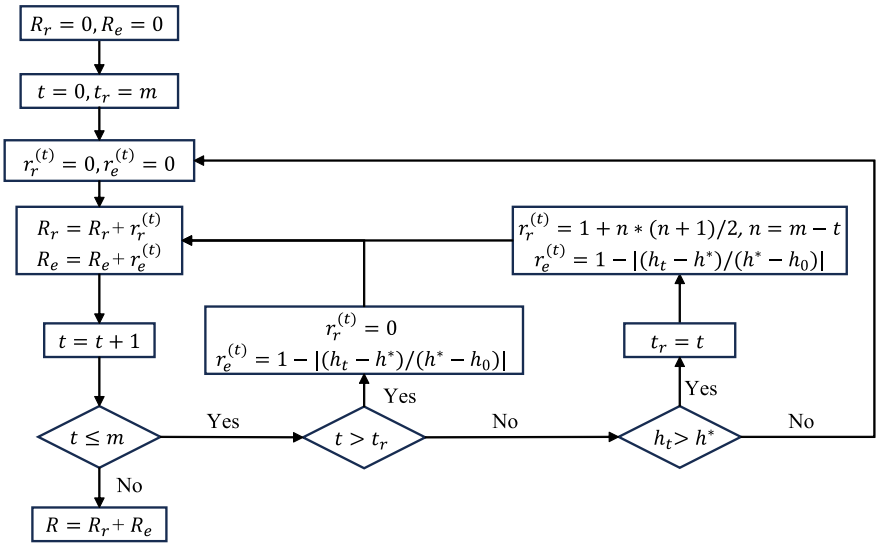


Figure 3. *Workflow of reward evaluation on the trajectory.*

(7) Accumulate $r_r^{(t)}$ and $r_e^{(t)}$ to the total RT-related reward R_r and total IAE-related reward R_e, respectively.

(8) Proceed one step forward, and loop through steps (4) to (7), until the current step reaches the end.

(9) Summarise R_r and R_e as the total reward R of the trajectory.

2.2 Proof

Assuming there are multiple trajectories for the WIG to manoeuvre during altitude change, consider one trajectory T with RT as t_r. For comparison, there exists another trajectory T' with a smaller RT denoted as t_r', assuming $t_r' = t_r - \delta$. The respective rewards for the two trajectories are labeled as R and R', which can be combined using Equations (1), (2) and (3), then expressed as:

$$R = \sum_{t=t_r}^{m} r_e^{(t)} + \frac{n*(n+1)}{2} + 1\,,\ n = m - t \tag{4}$$

$$R' = \sum_{t=t_r'}^{m} r_e'^{(t)} + \frac{n'*(n'+1)}{2} + 1\,,\ n' = m - t' \tag{5}$$

where $r_e'^{(t)}$ and n' are the AE reward of each step and the number of subsequent steps after RT for T', $n' = n + \delta$. R' minus R gives:

$$\Delta R = R' - R = \sum_{t=t_r'}^{m} r_e'^{(t)} - \sum_{t=t_r}^{m} r_e^{(t)} + \delta * \left(n + \frac{\delta + 1}{2}\right) \tag{6}$$

To explore the minimum value of ΔR, note that the per-step AE reward lies within [0,1] for $r_e^{(t)}$ and $r_e'^{(t)}$. Since $\sum_{t=t_r'}^{m} r_e'^{(t)}$ appears as an additive term, its minimum value should be taken as 0. In contrast, $\sum_{t=t_r}^{m} r_e^{(t)}$ appears as a subtractive term, and thus its maximum, $n+1$, should be used. Consequently, ΔR can be derived as:

$$\Delta R = \delta * \left(n + \frac{\delta + 1}{2}\right) - (n + 1) \tag{7}$$

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

Algorithm 1

Input The altitude of each trajectory point h_t, t=1, 2 . . . m for the trajectory T
Initialise $R_r \leftarrow 0$, $R_e \leftarrow 0$, $t = 0$
Assume the time step of rise time as $t_r = m$
while $t \leq m$ **for**
    if $t > t_r$ **then**
        $r_r^{(t)} \leftarrow 0$, $r_e^{(t)} \leftarrow 1 - |(h_t - h^*) \,/\, (h^* - h_0)|$
    else if $h_t > h^*$ **then**
        $t_r \leftarrow t$
        $r_r^{(t)} \leftarrow 1 + n * (n+1)\,/2\,,\ \ n = m - t$
        $r_e^{(t)} \leftarrow 1 - |(h_t - h^*) \,/\, (h^* - h_0)|$
    else
        $r_r^{(t)} \leftarrow 0$, $r_e^{(t)} \leftarrow 0$
    end if
    $R_r \leftarrow R_r + r_r^{(t)}$, $R_e \leftarrow R_e + r_e^{(t)}$
    $t = t + 1$
end while
$R = R_r + R_e$
output R

Given that δ is a positive integer, $\delta \geqslant 1$, the value of ΔR is therefore never less than 0, $\Delta R \geqslant 0$. Moreover, since the boundary value cannot be reached in practice, ΔR is greater than 0, $\Delta R > 0$. As the optimisation algorithm seeks trajectories with larger reward, it will prefer T' over due to its higher reward, thereby driving RT to decrease, as:

$$R < R' \Rightarrow T \rightarrow T' \Rightarrow t_r \rightarrow t_r' \tag{8}$$

Meanwhile, the smaller the deviation, the larger the IAE reward becomes.

2.3 Interpretation

For comparison, trajectories where RT occurs in the t_r-th and t_r'-th step is selected, assuming t_r'=t_r-1. Figure 4 illustrates both typical and extreme cases for these two types of trajectories. T_1 represents a typical possible trajectory. After RT, if there is some deviation from the target altitude that does not reach the maximum, the reward R_e will not be maximised. If no deviation from the target altitude occurs after RT, as in trajectory T_2, R_e reaches its maximum value, and this trajectory achieves the highest possible reward for RT=t. When RT decreases, if the deviation after RT reaches its maximum, as in trajectory T_3, the reward $R_{e'}$ drops to its minimum of zero, representing the lowest possible reward for RT=t'. A more common case with sub-maximum deviation is trajectory T_4, where $R_{e'}$ is greater than zero. To drive the transition from trajectories with RT=t_r to those with RT=t'_r feasibly through reward shaping, it must be ensured that the lower bound of the reward evaluation for RT=t'_r trajectories is no less than the reward value for RT=t_r trajectories. Therefore, the following equation must be satisfied:

$$R_{r'} = R_r + \max\ (R_e) \tag{9}$$

The RT reward after a one-step reduction must equal the sum of the RT reward and the maximum achievable IAE reward before the reduction. This condition enables the transition from trajectory T_2 to T_3 and makes it possible to drive the reduction in RT. This relationship can be formally expressed as

$$R_{r'} - R_r \geqslant R_e \tag{10}$$

Physically, this equation implies that the incremental gain in RT reward resulting from a one-step decrease in RT should be greater than the loss of IAE reward caused by the RT reduction. In more

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

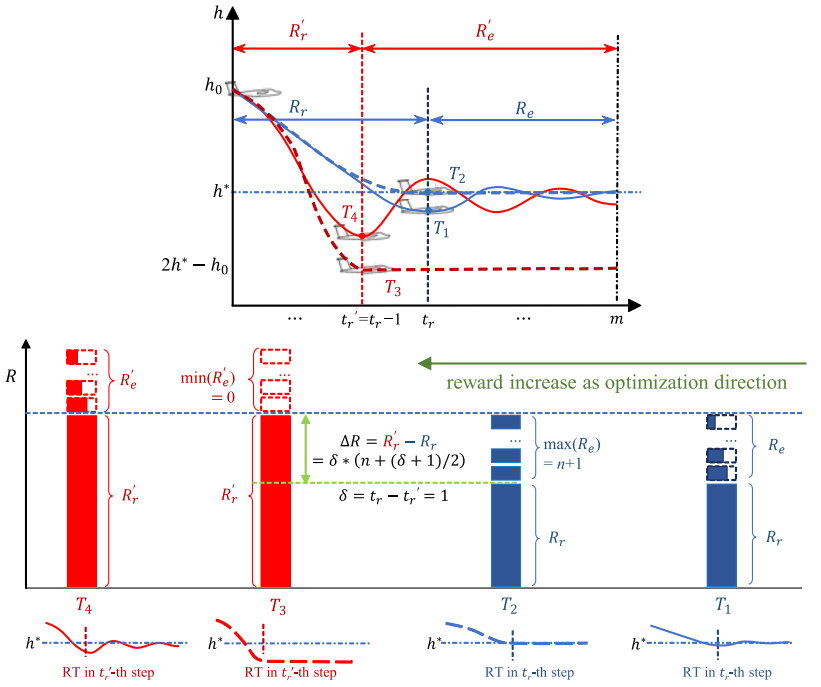


Figure 4. *The comparison of the reward for trajectories whose RT occurs at t-th step and (t′)-th step, respectively.*

general cases, such as trajectories T_1 and T_4, the RT reward gain exceeds the corresponding IAE loss. Moreover, the trajectory after RT reduction still retains a residual IAE reward. These effects collectively promote the transition from trajectories like T_1 toward those like T_4.

The above reasoning applies to adjacent RT differing by exactly one time step. This concept can be generalised to multi-step RT differences through iterative computation, allowing derivation of the RT reward for RT occurring at any arbitrary step. For notational clarity, $R_r^{(t)}$ and $R_e^{(t)}$ represent the RT reward and IAE reward, respectively, when RT=t, that can be derived step-by-step backwards, as:

$$\begin{aligned} R_r^{(t)} &= R_r^{(t+1)} + \max\left(R_e^{(t+1)}\right) \\ &= R_r^{(t+2)} + \max\left(R_e^{(t+2)}\right) + \max\left(R_e^{(t+1)}\right) \\ &\quad\vdots \\ &= R_r^{(m)} + \max\left(R_e^{(m)}\right) + \max\left(R_e^{(m-1)}\right) \cdots + \max\left(R_e^{(t+2)}\right) + \max\left(R_e^{(t+1)}\right) \\ &= R_r^{(m)} + 1 + 2 + 3 + \cdots + n \\ &= 1 + \frac{n\,(n+1)}{2} \end{aligned} \tag{11}$$

where $R_r^{(m)}$ is defined as the initial RT reward for the situation where RT occurs at the last step, m-th step. To prioritise achieving RT in the final step of the trajectory, the RT reward value should not exceed the maximum AE, i.e. 1. From the end to the beginning, the reverse analysis of RT reward for the step where RT happens is conducted to define its magnitude, illustrated in Fig. 5.

Table 1 shows the upper and lower bounds of the reward for trajectories whose RT occurs at different steps. Moving from left to right, the RT becomes progressively smaller. Although R_r and R_e is greater and greater, the maximum of the total reward is the same as the minimum of that when RT occurs in

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

Table 1. *Bounds of reward for the trajectory whose rise time occurs in different steps*

Rise time in t-th step	$t=m$		$t=m$-1		$t=m$-2		$t=m$-3		$\cdots$	$t=m$-n	
Reward	R_r	R	R_r	R_e	R_r	R_e	R_r	R_e	$\cdots$	R_r	R_e
Magnitude	1	≤ 1	2	≤ 2	4	≤ 3	7	≤ 4	$\cdots$	$1+\frac{n*(n+1)}{2}$	$\leq (n+1)$
Total reward	$[1, 2]$		$[2, 4]$		$[4, 7]$		$[7, 11]$		$\cdots$	$[1+\frac{n*(n+1)}{2}$, $1+\left(\frac{n}{2}+1\right)*(n+1)]$	

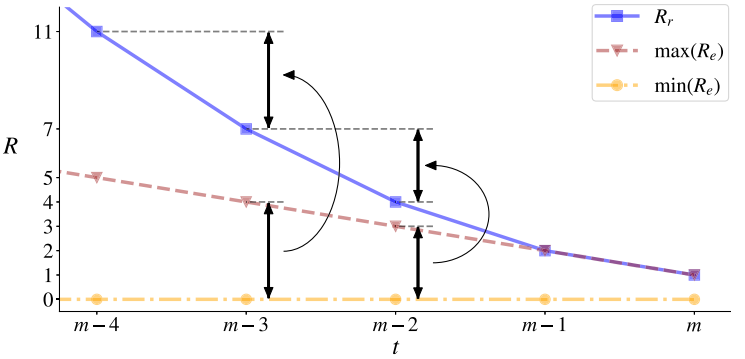


Figure 5. *RT reward R_r and the bounds of IAE reward R_e if rise time occurs in the last five steps.*

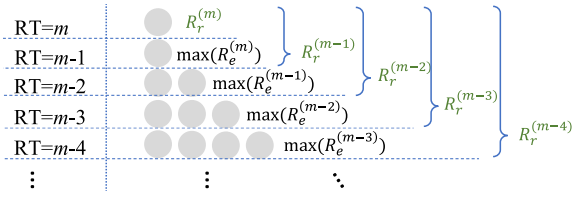


Figure 6. *The form of a triangular number for the construction of the RT reward R_r based on IAE reward R_e.*

one earlier step. The staircase-like pattern strictly drives the trajectory to shift towards the direction of RT's decrease.

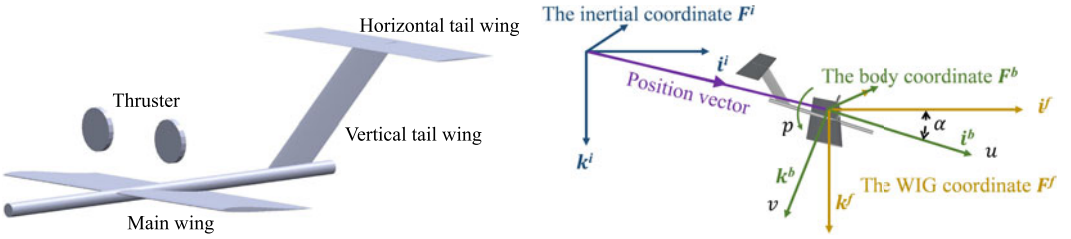
The RT reward value exhibits a form similar to the sum of an arithmetic sequence, i.e. $n^*(n+1)/2$ concerning the remaining steps, which is the triangular number shown in Fig. 6.

3.0 Simulation and method

3.1 Flight simulation

Validation is carried out using a dynamic and kinematic model of a small WIG operating in the longitudinal plane. The model has three degrees of freedom, with control surfaces consisting of an elevator and a propulsion [4]. Its geometrical shape and coordinate frames are shown in Fig. 7. The inertial frame F^i is a fixed Earth-based reference frame, with its origin located at a specified home position. Its unit vector i^i points forward, and k^i points downward. The WIG-fixed frame F^f is attached to the vehicle's centre of gravity (CG), and its axes are aligned with those of the inertial frame. The body frame F^b also originates at CG, but its axes move with the vehicle: i^b points toward the nose, and k^b points downward along the belly direction [34]. The WIG's state quantities are displacements and angles along with their

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press



(a) Geometry [4] (b) Coordinate frames

Figure 7. *The WIG's geometry and coordinate frames for describing the WIG's motion.*

first and second derivatives, and the flight altitude can be extracted from the state. Aerodynamic forces applied to the WIG are computed using the vortex lattice method and image method [4], and then fed into the kinematic model to calculate the WIG's motion response.

$$\dot{\boldsymbol{u}} = \boldsymbol{u} \times \omega + \boldsymbol{C}_{\boldsymbol{g}}^{\boldsymbol{b}} \cdot \left(\frac{\boldsymbol{F}}{m_w} + \boldsymbol{g}^{\boldsymbol{n}} \right) \tag{12}$$

$$\dot{\omega} = \boldsymbol{I}^{-1} \left(-\omega \times \boldsymbol{I}\omega + \boldsymbol{M} \right) \tag{13}$$

where m_w is mass of the WIG. $\boldsymbol{u} = [u, 0, w]^T$ and $\omega = \left[0, q, 0\right]^T$ denote the translational and angular velocity vectors in the body frame F^b. The aerodynamic force and moment vectors are given by $\boldsymbol{F} = \left[F_x, 0, F_z\right]^T$ and $\boldsymbol{M} = \left[0, M_y, 0\right]^T$, respectively. Gravity in the inertial frame is represented as $\boldsymbol{g}^{\boldsymbol{n}} = \left[0, 0, g\right]^T$, and m is the vehicle mass. Velocities u and w are along the body axes i^b and k^b, with their derivatives $\dot{u}$ and $\dot{w}$ indicating linear accelerations. The pitch rate q is the angular velocity about j^b. Forces F_x and F_z act along i^f and k^f in the WIG-fixed frame F^f, while g denotes gravitational acceleration. The transformation matrix $\boldsymbol{C}_{\boldsymbol{g}}^{\boldsymbol{b}}$ converts vectors from F^f to F^b. The inertial velocity $\boldsymbol{V} = \left[V_x, 0, V_z\right]^T$ is obtained via $\boldsymbol{V} = \boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{g}} \cdot \boldsymbol{u}$, where V_x and V_z are velocity components along i^i and k^i, and $\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{g}} = \left(\boldsymbol{C}_{\boldsymbol{g}}^{\boldsymbol{b}}\right)^T$. The inertia tensor is denoted by $\boldsymbol{I}$.

3.2 Optimisation method

The reward function directs the trajectory towards the extreme value, and therefore, optimisation algorithms are employed to identify this trajectory. Consequently, the identified trajectory is analysed to determine whether it aligns with the design expectations. Deep reinforcement learning (DRL) excels in solving complex, high-dimensional and dynamic decision-making problems by autonomously learning sequential policies. Thus, DRL is utilised as an optimisation tool in this research to search for the trajectory corresponding to the reward function. The basis structure of deep reinforcement learning for WIG's trajectory planning is shown in Fig. 8. Specifically, the PPO algorithm is employed, and details such as hyperparameters can be found at the link mentioned in the final section ***Data availability statement***. PPO is a policy gradient method that improves stability by limiting the policy update step size. The mathematical formulation of PPO and the corresponding importance sampling mechanism are given in [35]:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{14}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta\text{old}}(a_t|s_t)} \tag{15}$$

where θ denotes the parameters of the policy π_θ, which defines the probability of taking action a_t given state s_t at timestep t. The term $\hat{A}_t$ represents the estimated advantage at time t, quantifying how favourable action a_t is in state s_t compared to the expected baseline. It is defined as $\hat{A}_t(s,a) = \mathbb{E}[r(s,a) - r(s)]$,

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

Table 2. *Hyperparameters of DRL for the adjustment of altitude*

Hyperparameters	Magnitude
Learning rate *lr* for policy optimiser and value function optimiser	$10^{-4}, 10^{-5}, 10^{-6}$
Discount factor	$1 - 1/m$
Number of steps of interaction per epoch/replay buffer	$1,024\,m$
Batch size	$2\,m$
Repeating times for each batch to be trained on per training iteration	16
Clipping ratio	0.15
Entropy coefficient λ	0.01, 0.005, 0.0025
Consecutive epochs if no growth in episode reward for the next *lr* and λ	200

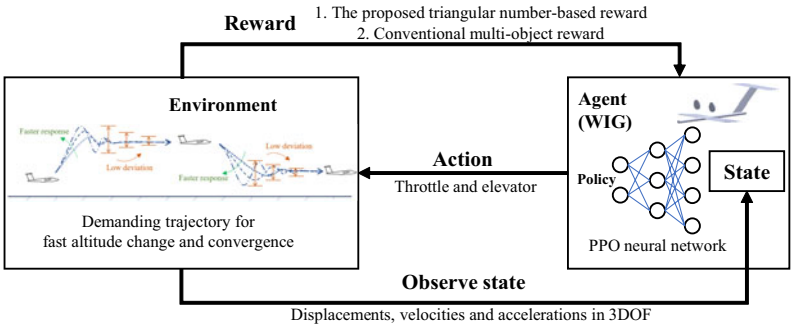


Figure 8. *The structure of deep reinforcement learning for WIG's trajectory planning.*

where $r\,(s, a)$ denotes the expected reward for taking action a in state s, and $r\,(s)$ is the expected reward for state s regardless of action. The objective L is used to guide gradient ascent updates. The expectation operator $\hat{\mathbb{E}}_t$ indicates an empirical average over a batch of collected data in the alternating sampling and optimisation process. The ratio r_t is the importance sampling weight, representing the relative likelihood of the action under the new policy compared to the old one and is essential for stable updates. The function clip $(\cdot)$ constrains the ratio within a predefined interval to prevent overly large updates; for instance, values below 0 become 0, and those above 1 become 1 when clipped to [0, 1]. The hyperparameter ε controls the clipping threshold in the objective function, limiting how far the policy is allowed to deviate during each update. This work implements PPO using ElegantRL [36], an open-source DRL framework.

Each DRL case is trained with eight different random seeds, and the training result with the highest reward value is selected as the final outcome for that case. The training hyperparameters are set as follows: each policy iteration collects 1,024 complete trajectory steps of data, with the clip ratio set to 0.15. During training, when the reward showed no improvement after 200 consecutive iterations, the learning rate is reduced from the initial 10^{-4} to 10^{-5}, and the entropy coefficient is decreased from 0.01 to 0.005. Training then continued until the reward again showed no improvement after another 200 iterations, at which point the learning rate is further reduced to 10^{-6} and the entropy coefficient to 0.0025. The training is terminated when no reward improvement is observed for a final 200-iteration period. Table 2 lists hyperparameters of the DRL algorithm.

4.0 Cases and analysis

To validate the effectiveness of the proposed reward function, the manoeuvre is designed as the switch of the flight altitude from the initial h_0=0.24 m to the end h^*=0.12 m (Fig. 9), and the planned trajectory is compared with those obtained by a conventional multi-objective reward function. The trajectory consists

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

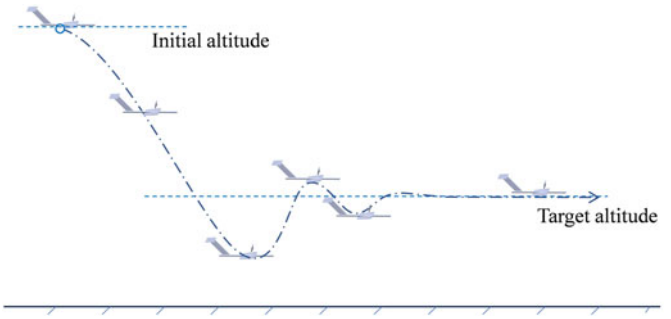


Figure 9. *Validation case of a WIG's flight altitude switch.*

of 24 steps, m= 24. The conventional multi-objective reward function is expressed by

$$R_e^\star = \sum_{i=1}^{m} \left(1 - \left|\frac{h_i - h^*}{h^* - h_0}\right|\right) \tag{16}$$

$$R_r^\star = m - t_r \tag{17}$$

$$R^\star = R_e^\star + k \cdot R_r^\star \tag{18}$$

where t_r is the order of the step where RT occurs in the trajectory, and k is the weight of the RT-related part, which needs to be tuned step by step to roughly obtain the expected trajectory. Likewise, $R_e^\star$ and $R_r^\star$ are the RT and IAE parts of total reward, $R^\star$. Their evaluation is based on all steps, which is the most significant difference from the proposed reward function.

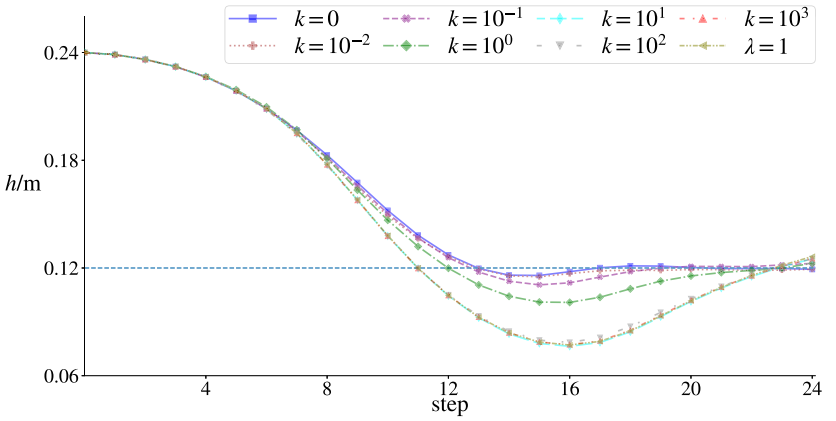
4.1 Effectiveness

To validate the effectiveness of the proposed method, we employed a traditional multi-objective function configured with a sequence of seven continuously increasing weight values. This setup simulates the practical process of iteratively tuning weights in multi-objective optimisation until convergence to the target trajectory is achieved. The proposed method is implemented under the same conditions for comparison. The traditional approach uses seven distinct weight values: $k = 0$, $k = 10^{-2}$, $k = 10^{-1}$, $k = 10^{0}$, $k = 10^{1}$, $k = 10^{2}$ and $k = 10^{3}$, while the proposed method is represented by 'Proposed'.

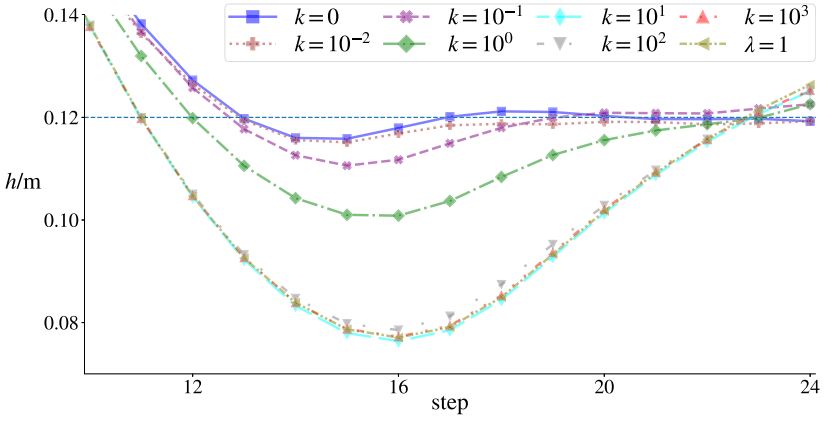
Figure 10 shows the results of the multi-objective reward function with different weight schemes. Figure 10(c) demonstrates that as the RT-related weight increases from $k = 0$ to $k = 10^{1}$, the moment when the WIG craft reaches the target altitude gradually advances, until the rise time is minimised at the 11-th step. In the latter part of the trajectory, Fig. 10(a) shows that after reaching the target altitude, the concavity of the trajectory becomes increasingly pronounced, accompanied by a corresponding growth in deviation from the target altitude. The degree of concavity reaches its maximum when the RT-related weight increases to $k = 10^{1}$. Figure 10(b) further reveals that although trajectory deviation increases with the RT-related weight, this trend saturates once the weight exceeds $k = 10^{1}$, beyond which further increases yield no significant change in deviation. The trajectory corresponding to the proposed reward function directly achieves the minimum RT and subsequent approach to the target altitude. The proposed reward function does not introduce any part that needs iterative adjustments of weights through trial and error, and it directly results in the expected trajectory with the maximum concavity.

Figure 11 displays the RT and IAE values for the aforementioned cases. It can be observed that as the RT-related weight in the multi-objective function increases exponentially, the trajectory's RT decreases while the IAE increases. Beyond $k = 10^{1}$, both RT and IAE have plateaued, and the proposed method directly achieves identical RT and IAE values. The observed behaviour aligns with the design intent: a relatively large RT-related weight in the reward function effectively reduces RT. The proposed method

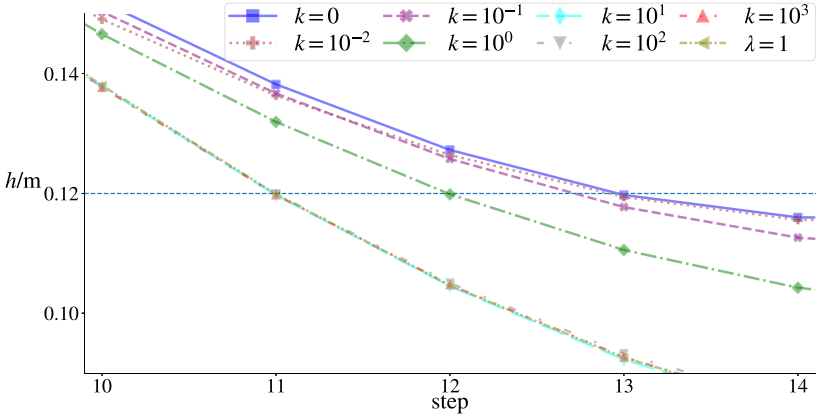
https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press



(a) Overall view



(b) Intermediate view



(c) Close-up view

Figure 10. *Trajectories of multi-objective reward functions assigned with different weight k and the proposed reward function.*

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

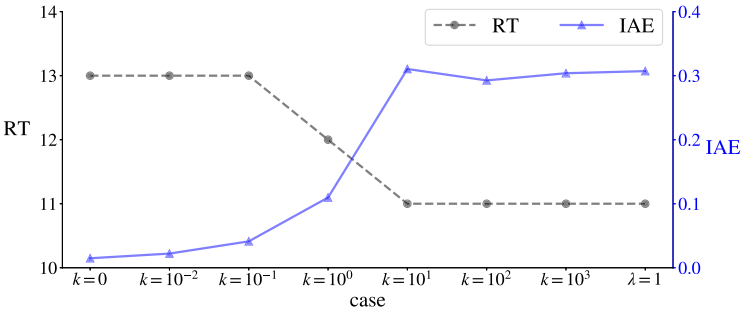


Figure 11. *Performance of RT and IAE for the multi-objective reward function with different weight k and the proposed reward function*

achieves the same RT and IAE as the multi-objective approach after full convergence, demonstrating that it successfully minimises RT while preventing further increase in IAE, thereby reaching the minimum IAE achievable under the constraint of minimised RT. With increasing RT-related weight, its proportion in the total reward becomes more substantial. Since the WIG tends to maximise reward, its trajectory shows a tendency toward reduced RT, though this comes at the cost of greater deviation from the target altitude. When the RT-related weight is sufficiently large, the reward gain from reducing RT exceeds the loss caused by trajectory deviation, thereby promoting a faster response. This indicates that the earlier arrival at the target altitude is achieved at the expense of trajectory precision during the approach.

4.2 Invariance

The proposed reward function is highly effective in achieving both reduced RT and lower deviation, and its performance is unlikely to be significantly improved through further modifications. To demonstrate the advantage of the proposed reward function over traditional multi-objective approaches that require parameter tuning, an additional coefficient λ is introduced to examine its invariance, as:

$$R_r = 1 + \frac{n * (n+1)}{2}, \quad n = m - t_r \tag{19}$$

$$R_e = \sum_{i=t_r}^{m} \left(1 - \left|\frac{h_i - h^*}{h^* - h_0}\right|\right) \tag{20}$$

$$R = R_e + \lambda \cdot R_r \tag{21}$$

Four coefficients – $\lambda = 1$, $\lambda = 10$, $\lambda = 10^1$ and $\lambda = 10^2$ – are selected to compare the resulting trajectories, as shown in Fig. 12. It can be observed that the trajectories are similar. Figure 13 presents the corresponding RT and IAE values, which show only minor differences. Slight variations can be seen in the concavity of the latter trajectory segments, which may be attributed to optimisation errors. This consistency occurs because when $\lambda \geqslant 1$, the reward gain from reducing RT is guaranteed to be sufficiently large to outweigh the penalty caused by increased deviation. Since the quantitative condition of the reward function is satisfied, essentially identical trajectories are obtained.

Validation is performed by attaching a coefficient greater than 1 to the RT-related term and comparatively evaluating a series of coefficient values. The results confirm that the proposed reward function requires no custom development for performance adjustment. Unlike conventional multi-objective reward functions, this method eliminates the need for iterative coefficient tuning and directly generates the ideal trajectory that achieves minimum rise time while maintaining maximum proximity to the target altitude.

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

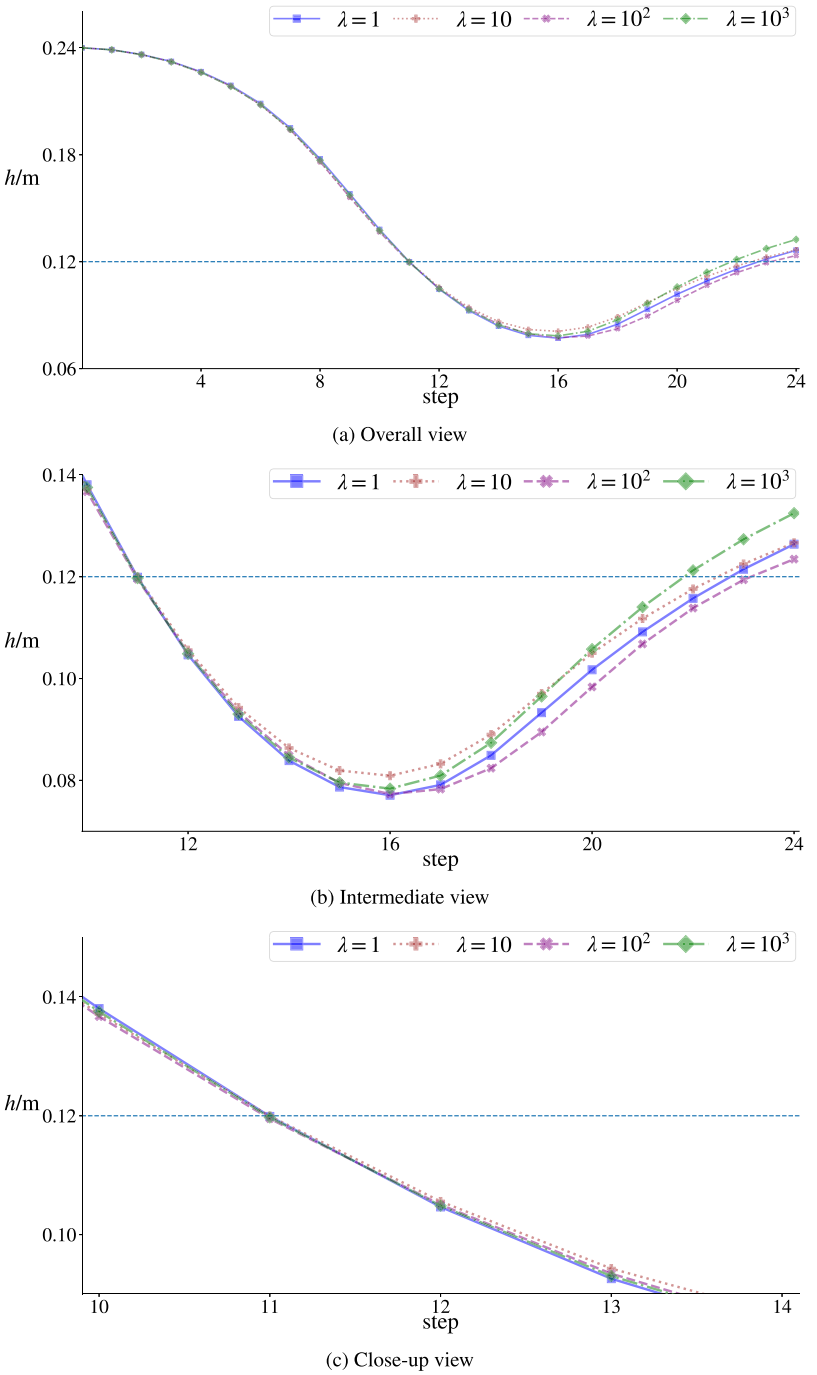


Figure 12. *Trajectories of the proposed reward functions assigned with different coefficient α.*

5.0 Conclusion

A reward function is proposed to directly achieve rapid response by minimising RT during state transition, allowing the WIG to quickly adjust its flight altitude, and then sufficiently approach the target altitude. The core idea is to set the reward for early RT higher than the penalty for deviating from

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

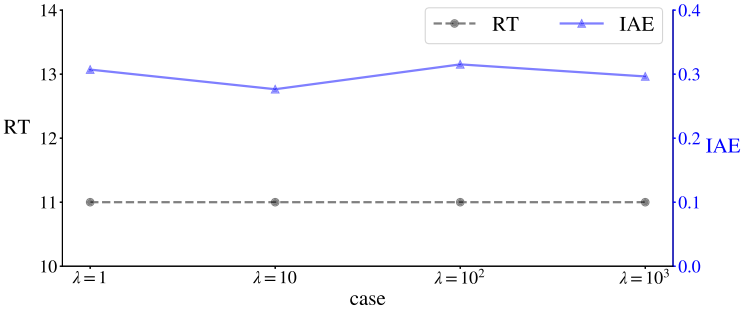


Figure 13. *Performance of RT and IAE for the proposed reward function with different coefficient λ.*

the target altitude, thus encouraging the WIG to reach the target altitude more quickly. Its effectiveness is validated through trajectory planning of a WIG's change of flight altitude. By comparison with conventional multi-objective reward functions that adjust weights, the straightforward advantage of the proposed method is highlighted. It can guide the WIG to exert itself to quickly change and converge to the target altitude.

Acknowledgements. The authors gratefully acknowledge the constructive comments from the anonymous reviewers and the editor, whose insights and suggestions significantly improved the quality of this manuscript.

Data availability statement. Source codes are available at the link that is https://github.com/HuanHu2019/Triangular-Numbers-Based-Objective-Function-for-Fast-Altitude-Change-and-Convergence.

Competing 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.

References

[1] Patria, D., Rossi, C., Fernandez, R.A.S. and Dominguez, S. Nonlinear control strategies for an autonomous wing-in-ground-effect vehicle, *Sensors*, 2021, **21**, p 4193.
[2] Panferov, A., Nebylov, A. and Brodsky, S. Features of designing control systems for WIG-craft, *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), IEEE*, Torino, Italy, 2019, pp 136–141.
[3] Daniel, D. Development of an automatic height control system for WIG crafts, *IFAC Proc.*, 2008, **41**, pp 1088–1092.
[4] Hu, H., Zhang, G., Ding, L., Jiao, K., Zhang, Z. and Zhang, J. Iterative reward shaping for non-overshooting altitude control of a wing-in-ground craft based on deep reinforcement learning, *Rob. Auton. Syst.*, 2023, **163**, p 104383.
[5] Zhang, J., Hu, H., Zhang, G., Zhang, Z. and Wang, Z. Deep reinforcement learning-based ditching vertical velocity control for a multi-fault wing-in-ground craft, *Appl. Ocean Res.*, 2023a, **136**, p 103585.
[6] Zhang, J., Hu, H., Zhang, G. and Zhang, Z. Landing longitudinal control of wig-craft based on deep reinforcement learning, *Sh. Build. China*, 2023b, **64**, pp 215–223.
[7] Hu, H., Li, D., Zhang, G. and Zhang, Z. Multi-objective reward shaping for global and local trajectory planning of wing-in-ground crafts based on deep reinforcement learning, *Aeronaut. J.*, 2024, **128**, pp 371–397.
[8] Mrabet, N., Benzazah, C., El Akkary, A., Ennaciri, Y. and Lahlouh, I. Maximum power optimization of a direct-drive wind turbine connected to PMSG using multi-objective genetic algorithm, *Int. J. Appl. Power Eng. (IJAPE)*, 2024a, **13**, p 453.
[9] Mrabet, N., Benzazah, C., Mohssine, C., Ahmed, E.A., Driss, K., Badr, R. and Ilyas, L. Enhancing wind energy conversion efficiency: a novel MPPT approach using P&O with ADRC controllers versus PI controllers with Kp and Ki optimization via genetic algorithm and ant colony optimization, *Clean. Energy Syst.*, 2024b, **9**, p 100159.
[10] Mrabet, N., Benzazah, C., El Akkary, A., Chakib, M. and Lahlouh, I. Optimizing PID controller parameters in wind energy conversion systems using a metaheuristic technique that relies the integrated time absolute error (ITAE), *EPJ Web. Conf.*, 2025, **330**, p 02001.
[11] Nayak, P.C., Nayak, B.P., Prusty, R.C. and Panda, S. Sunflower optimization based fractional order fuzzy PID controller for frequency regulation of solar-wind integrated power system with hydrogen aqua equalizer-fuel cell unit, *Energy Sources Part A*, 2025, **47**, pp 9550–9568.
[12] Nasir, N.M., Ghani, N.M.A., Nasir, A.N.K., Ahmad, M.A. and Tokhi, M.O. Neuro-modelling and fuzzy logic control of a two-wheeled wheelchair system, *J. Low Freq. Noise Vib. Act. Control.*, 2025, **44**, pp 588–602.
[13] Ghosh, A., Ray, A.K., Nurujjaman, M. and Jamshidi, M. Voltage and frequency control in conventional and PV integrated power systems by a particle swarm optimized Ziegler–Nichols based PID controller, *SN Appl. Sci.*, 2021, **3**, p 314.

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press

[14] Samuk, D.C. and Çakr, O. Analysis of multivariable objective functions for the PID controller tuned by a radial movement optimization, *Pamukkale U. J. Eng. Sc.*, 2023, **29**, pp 23–29.
[15] Cárdenas, J.A., Carrero, U.E., Camacho, E.C. and Calderón, J.M. Optimal PID ø axis Control for UAV quadrotor based on multi-objective PSO, *IFAC-PapersOnLine*, 2022, **55**, pp 101–106.
[16] Kushary, I. and Barai, R.K. Effective vehicle battery charging station based on dual active full bridge converter using enhanced genetic algorithm, *Electr. Power Syst. Res.*, 2026, **250**, p 112147.
[17] Thampi, A. and Mahendran, N. Tuning of fractional order PID controller using the animated oat optimization technique for capacitor balancing in packed U cell multilevel inverter, *Eng. Res. Exp.*, 2026, **8**, p 065326.
[18] Shashibhushan, G. and Sonoli, S. Multi-objective solution with PSO algorithm for minimization of torque ripple and speed settling time by using solar-fed 11,9 and 3-level multi-level inverter with vector control of induction motor, *Int. J. Power Electron. Drive Syst. (IJPEDS)*, 2020, **11**, 827.
[19] Cojuhari, I., Fiodorov, I., Izvoreanu, B. and Moraru, D. Synthesis of PID Controller for the Automatic Control System with Imposed Performance based on the Multi-Objective Genetic Algorithm, *2020 International Conference and Exposition on Electrical And Power Engineering (EPE)*, IEEE, Iasi, Romania, 2020, pp 598–603.
[20] Zahir, A., Alhady, S.S.N., Wahab, A. and Ahmad, M. Objective functions modification of GA optimized PID controller for brushed DC motor, *Int. J. Electr. Comput. Eng. (IJECE)*, 2020, **10**, p 2426.
[21] Ekinci, S., Izci, D. and Hekimoğlu, B. Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm, *Arab. J. Sci. Eng.*, 2021, **46**, pp 1395–1409.
[22] Rahayu, E.S., Ma'arif, A. and Cakan, A. Particle swarm optimization (PSO) tuning of PID control on DC motor, *Particle Swarm Optimization*, 2022, **2**, pp 435–447.
[23] Ekinci, S., Can, T. and Izci, D. Controller design for automatic voltage regulator system using modified opposition-based weighted mean of vectors algorithm, *Int. J. Model. Simul.*, 2025, **45**, pp 1322–1339.
[24] Abualigah, L., Izci, D., Ekinci, S. and Zitar, R.A. Optimizing aircraft pitch control systems: a novel approach integrating artificial rabbits optimizer with PID-F controller, *Int. J. Robot. Control Syst.*, 2024, **4**, pp 354–364.
[25] Micev, M., C′alasan, M. and Oliva, D. Design and robustness analysis of an automatic voltage regulator system controller by using equilibrium optimizer algorithm, *Comput. Electr. Eng.*, 2021, **89**, p 106930.
[26] Can, M.S. and Ercan, H. Real-time tuning of PID controller based on optimization algorithms for a quadrotor, *Aircr. Eng. Aerosp. Technol.*, 2022, **94**, pp 418–430.
[27] Fantin, V. and Mohanbabu, G. Symmetrical and high-speed CMOS inverter design using hybrid PSOGSA optimization algorithm, *Tehnički vjesnik*, 2026, **33**, pp 126–133.
[28] Yüksek, G. A novel delay-affected two degree of freedom PID controller using physics-inspired optimization for robust control applications, *Sci. Rep.*, 2025, **16**, p 3453.
[29] Taligatos, R., Nguyen, D. and Lowenberg, M.H. Investigating Missile Feedback Control: Reinforcement Learning, PID, or Both?, *AIAA SCITECH 2026 Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2026.
[30] Zhao, J. and Sun, H. Based on DDPG-FOPID control algorithm for high-precision motion control of optoelectronic platform, in Shi, Z. and Tiwari, S. (Eds), *International Conference on Image, Signal Processing, and Machine Learning (ISPML 2025)*, SPIE, Xi'an, 2026, China, p 20.
[31] Ahmadi, S., Mirji, C., Peng, H., Luo, S. and Ahmadi, S. Comparative Analysis of Reinforcement Learning Models for Satellite Attitude Recovery Under Unknown Failures, *AIAA SCITECH 2026 Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2026.
[32] Delelew, E.Y., Dulecha, K.A., Ararso, Z.T. and Abdissa, C.M. Reinforcement learning-driven model predictive control for optimizing counter-rotating permanent magnet synchronous motor in submarine propulsion system, *Sci Rep*, 2026, **16**, p 5277.
[33] Hassan, A.M., Ababneh, J., Attar, H., Shamseldin, T., Abdelbaset, A. and Metwally, M.E. Reinforcement learning algorithm for improving speed response of a five-phase permanent magnet synchronous motor based model predictive control, *PLOS ONE*, 2025, **20**, p e0316326.
[34] Beard, R. and McLain, T. Small unmanned aircraft: theory and practice, 2012.
[35] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. Proximal policy optimization algorithms, 2017, CoRR abs/1707.06347.
[36] Xiaoyang, L., Zechu, L., Zhaoran, W. and Jiahao, Z. ElegantRL: Massively Parallel Framework for Cloud-native Deep Reinforcement Learning, 2021.

Cite this article: Hu H., Shi Y., Zhang G., Zhang Q., Wang Y. and Zhang J. (2026). A triangular number-based reward function for trajectory planning of wing-in-ground crafts towards fast-response and low-deviation altitude changes. *The Aeronautical Journal*, **130**, 3197–3213. https://doi.org/10.1017/aer.2026.10205

https://doi.org/10.1017/aer.2026.10205 Published online by Cambridge University Press