Contents lists available at ScienceDirect

Chinese Journal of Aeronautics

Journal homepage: www.elsevier.com/locate/cja

Final Accepted Version

A multi-strategy pigeon-inspired optimization approach to active disturbance rejection control parameters tuning for vertical take-off and landing fixed-wing UAV

Hangxuan HE[a], Haibin DUAN[b,*]

[a]*State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100083, China*

[b]*Peng Cheng Laboratory, Shenzhen 518000, China*

Received 21 Jan 2021; revised 20 Mar 2021, 22 Apr 2021; accepted 6 May 2021

Abstract

In this paper, active disturbance rejection control (ADRC) is utilized in the pitch control of a vertical take-off and landing fixed-wing unmanned aerial vehicle (UAV) to address the problem of height fluctuation during the transition from hover to level flight. Considering the difficulty of parameter tuning of ADRC as well as the requirement of accuracy and rapidity of the controller, a modified pigeon-inspired optimization algorithm (MSPIO) containing multiple strategies is employed. Particle swarm optimization (PSO), genetic algorithm (GA), the basic pigeon-inspired optimization (PIO), and an improved PIO algorithm CMPIO are compared. In addition, the optimized ADRC control system is compared with the pure proportional-integral-derivative (PID) control system and the non-optimized ADRC control system. The effectiveness of the designed control strategy for forward transition is verified and the faster convergence speed and better exploitation ability of the proposed MSPIO algorithm are confirmed by simulation results.

Keywords: Pigeon-inspired optimization algorithm; Active disturbance rejection control (ADRC); Vertical take-off and landing; Unmanned aerial vehicle (UAV); Transition mode

*Corresponding author. *E-mail address:* hbduan@buaa.edu.cn

1. Introduction

Recent years have witnessed the growing popularity of morphing unmanned aerial vehicles (UAVs), including bionic UAVs with flapping wings and hybrid UAVs. [1,2] Hybrid UAVs, which combine the merits of fixed-wing and rotor-wing UAVs, have drawn more attention. On the one hand, hybrid UAVs retain advantages, especially the long endurance and fast speed of the fixed-wing UAV. On the other hand, the characteristic of vertical take-off and landing enables the broad application of hybrid UAVs.

By the nature of hybrid UAVs, the flight envelop is usually divided into three regions, the hovering mode, the level flight mode, and the transition mode. This paper concentrates on the control during the transition mode from hover to level flight. Two main strategies used during the transition mode are changing the altitude following a designed trajectory, which is often employed by tail-sitters and maintaining a constant height when flying forward, which is popular among tilt-rotor UAVs as well as vertical take-off and landing fixed-wing UAVs. [3-7] The second strategy is more

commonly used as firstly, lifting is power-consuming and will weaken the long endurance of hybrid UAVs and secondly, it is easy to keep a constant altitude during the transition. In contrast to the tilt-rotors and tail-sitters, the vertical take-off and landing fixed-wing UAV, which is composed of four fixed rotors, is easier to assemble and manipulate.[4,7,8] Therefore, this paper studies the vertical take-off and landing fixed-wing UAV and adopts the fixed-altitude strategy.

In hybrid UAVs, various control methods have been employed. Some works adopted proportional-integral-derivative (PID) control, while others combined different approaches including sliding mode control and adaptive control.[3,7,9-12] In the aspect of transition at a constant altitude, Gu et al. have designed a transition position controller and utilized a rotary-wing attitude controller for the vertical take-off and landing fixed-wing UAV from hover to level flight.[6] Malang has employed PID controllers for the tilt-rotor UAV during the hovering mode and the forward flight mode.[7] During the transition, the mode shift was based on the tilt angle. Yayla et al. have proposed a hierarchical approach where the adaptive control is used in the inner loop, the Lyapunov method is adopted in the outer loop, and proportional-derivative (PD) control is used in the navigation loop.[11] Although extensive research has been carried out on controlling hybrid UAVs, studies focusing on the problem of height fluctuation during mode shift are rare. The height fluctuation is due to that the hybrid UAV pitches down when flying forward in the quadrotor mode whereas in the fixed-wing mode, the aircraft pitches up. The aerodynamic characteristics lead to uncertainties. Moreover, the slow response of the control system also results in the deviation from the trim states.

Therefore, active disturbance rejection control (ADRC) is adopted in this paper. ADRC regards all uncertainties as one term and uses an extended state observer (ESO) to estimate disturbances. Besides, ADRC keeps a better balance between rapidity and overshoot compared with PID owing to the tracking differentiator (TD).[13] For the merit of flexibility and simplicity, ADRC has been applied to various domains.[14-18] Nevertheless, tuning the coupling parameters in ADRC by trial and error is time-consuming and the controller needs to work in its best condition. Thus, an improved optimization algorithm is proposed to solve the problem.

Pigeon-inspired optimization (PIO) developed by Duan and Qiao is a novel swarm intelligence algorithm imitating the behavior of homing pigeons guided home by different tools during different stages.[19] With fast convergence speed and high efficiency, this algorithm and the variants have been successfully applied to different fields.[20] For example, image restoration, UAV or spacecraft path planning, formation control, parameter tuning, and data clustering.[21-29] However, PIO may be trapped in the local optimum especially when dealing with multi-dimensional optimization problems. To improve the search ability of PIO, a modified algorithm is presented in this paper by adopting three designed strategies, namely, Dynamic Inheritance, Hovering and Approaching, and Random Opposite Learning.

This paper aims to propose a control method adopting ADRC to address the problem of height fluctuation during forward transition for vertical take-off and landing fixed-wing UAVs and to develop an improved PIO named multi-strategy PIO (MSPIO) to solve multi-dimensional optimization problems and tune the parameters in the controller of the hybrid UAVs.

The rest of the paper is organized as follows. Section 2 presents the longitudinal mathematical model of a vertical take-off and landing fixed-wing UAV. Section 3 shows two ADRC controllers for the quadrotor mode and the fixed-wing mode respectively. Section 4 offers the detail of the MSPIO algorithm. Section 5 presents the implementation of the algorithm on the proposed controller and a pure PID controller. Simulation and comparison are conducted to demonstrate the effectiveness of the designed controller and the algorithm. The conclusion is in section 6.

2. Longitudinal modeling of vertical take-off and landing fixed-wing UAV

The vertical take-off and landing fixed-wing UAV is modeled as a six-degree-of-freedom dynamics.[30] Suppose the hybrid UAV is a rigid body with mass m and the products of inertia $I_{xy}=I_{yz}=0$. $[x, y, h, \varphi, \theta, \psi]$ is in the inertial frame O_g-$x_g y_g z_g$ fixed to the ground. $[u, v, w, p, q, r]$ is described in the body frame O_b-$x_b y_b z_b$ attached to the center of mass. The platform and the frames are shown in Fig. 1. The transition primarily takes place in the longitudinal motion, and the corresponding dynamics is shown in detail as follows.

$$\begin{aligned}
\dot{u} &= vr - wq - g\sin\theta + \frac{F_x}{m} \\
\dot{w} &= uq - vp + g\cos\theta\cos\varphi + \frac{F_z}{m} \\
\dot{q} &= \frac{I_z - I_x}{I_y} pr - \frac{I_{xz}}{I_y}\left(p^2 - r^2\right) + \frac{M}{I_y}
\end{aligned} \tag{1}$$

where F_x and F_z are the forces along the x_b and z_b axes, respectively. M is the pitch moment, including torques created by both rotors and wings. The angular kinematic equation of the motion is

$$\dot{\theta} = q\cos\varphi - r\sin\varphi \tag{2}$$

It is assumed that the effect of the lateral-directional motion is negligible during the transition, then Eq. (2) can be linearized as

$$\dot{\theta} = q \tag{3}$$

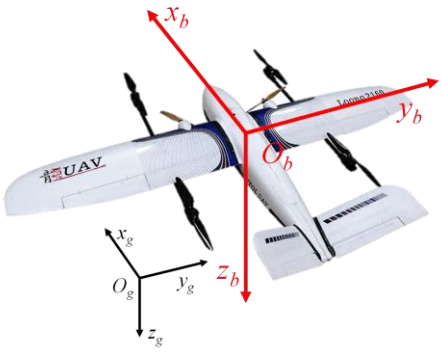


Fig. 1 Model of vertical take-off and landing fixed-wing UAV

3. ADRC design for the model

Considering the configuration of the vertical take-off and landing fixed-wing UAV, two ADRC controllers for the quadrotor mode and the fixed-wing mode are designed respectively. For simplicity, only second-order ADRC is employed.

3.1. Structure of ADRC

ADRC consists of a tracking differentiator (TD), an extended state observer (ESO), and a structure of nonlinear state error feedback (NLSEF), as shown in Fig. 2.

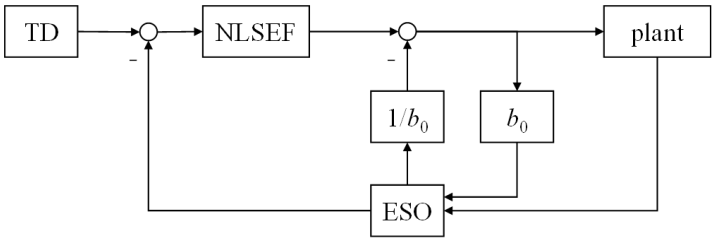


Fig. 2 Structure of ADRC

(1) Tracking Differentiator

The input signal is supposed as v. v_1 is the tracking signal of v and v_2 is the differential of v_1. The equations of the tracking differentiator can be given as

$$\begin{cases} e = v_1 - v \\ fh = fhan(e, v_2, r_0, h) \\ \dot{v}_1 = v_2 \\ \dot{v}_2 = fh \end{cases} \tag{4}$$

where e represents the error between the observer and the state. *fhan* is an optimal control synthesis function designed according to discrete optimization theory and is given as

$$\begin{cases} d = r_0 h \\ d_0 = r_0 h^2 \\ y = e + h v_2 \\ a_0 = \sqrt{d^2 + 8 r_0 \mid y \mid} \\ a = \begin{cases} v_2 + \dfrac{(a_0 - d)}{2} \operatorname{sign}(y), & \mid y \mid > d_0 \\ v_2 + \dfrac{y}{h}, & \mid y \mid \le d_0 \end{cases} \\ fhan = -\begin{cases} r_0 \cdot \operatorname{sign}(a), & \mid a \mid > d \\ r_0 \dfrac{a}{d}, & \mid a \mid \le d \end{cases} \end{cases} \tag{5}$$

where r_0, h are the parameters to be tuned.

(2) Extended State Observer

The extended state observer estimates the general disturbance and transforms the system into an integral-chain system. The equations are given as

$$\begin{cases} e = z_1 - y \\ \dot{z}_1 = z_2 - \beta_{01} e \\ \dot{z}_2 = z_3 - \beta_{02}\, fal(e, \alpha_{01}, \delta) + b_0 u \\ \dot{z}_3 = -\beta_{03}\, fal(e, \alpha_{02}, \delta) \end{cases} \tag{6}$$

$$fal(e, \alpha, \delta) = \begin{cases} \dfrac{e}{\delta^{\alpha - 1}}, & |e| \le \delta \\ |e|^{\alpha} \operatorname{sign}(e), & |e| \ge \delta \end{cases} \tag{7}$$

where z_1, z_2, z_3 are the estimated values; α_{01}, α_{02} are the factors deciding the non-linearity of the function *fal*; δ represents the nonlinear section of *fal*; β_{01}, β_{02}, β_{03} are the observer gains; b_0 is the compensation factor and u is the control input; α is in (0,1) and is often set to be 0.5.

(3) Nonlinear State Error Feedback

In this part, the function *fal* in Eq. (7) is utilized to generate initial control input u_0 as shown below.

$$\begin{cases} e_1 = v_1 - z_1 \\ e_2 = v_2 - z_2 \\ u_0 = \beta_{11}\, fal(e_1, \alpha_{11}, 0.01) + \beta_{12}\, fal(e_2, \alpha_{12}, 0.01) \end{cases} \tag{8}$$

where β_{11} and β_{12} are the feedback gains, α_{11} and α_{12} are set to be 0.75 and 1.5 in the quadrotor mode controller and set as 0.25 and 1.5 in the fixed-wing mode controller respectively after some trials.

Finally, the control input is calculated by

$$u = u_0 - \frac{z_3}{b_0} \tag{9}$$

3.2. ADRC for the quadrotor mode

It is supposed that the motors are modeled as the first order systems. The second order ADRC is adopted to build a pitch rate controller and the system model can be described in Eq. (10) by setting $\dot{\theta} = x_1, \ddot{\theta} = x_2$.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, \omega(t), t) + b_0 u \\ y = x_1 \end{cases} \tag{10}$$

where $\omega(t)$ is the disturbance. By treating $f(x_1,x_2,\omega(t),t)$ as the third state in the extended state observer, the observer is modeled as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + b_0 u \\ \dot{x}_3 = \omega(t) \\ y = x_1 \end{cases} \tag{11}$$

3.3. ADRC for the fixed-wing mode

In the fixed-wing mode, the ADRC is applied to the pitch control. The extended state observer design for the fixed-wing mode is the same as that in the quadrotor mode except that $\theta=x_1,\dot{\theta}=x_2$. Besides, since a flight mission is mainly composed of the level flight that requires a faster response, the tracking differentiator is not included in the ADRC of the fixed-wing mode. The whole longitudinal control structure is shown in Fig. 3.

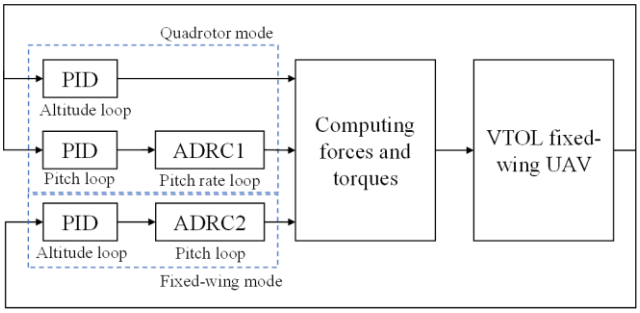


Fig. 3 Longitudinal control structure

When determining forward airspeed, firstly, the constraints of the overall actuators should be considered. Since the transition and the level flight primarily take place in the longitudinal motion, the airspeed of level flight must be faster than 10m/s to make sure the elevon is within ±45°. Secondly, according to the model adopted, the stalling angle of attack is 25° where the stalling speed is 14m/s and the designed airspeed needs to be greater than that.[30] Thirdly, lift generated by the wings after the transition must balance the UAV's gravity. The equation is given in Eq. (12).

$$\frac{1}{2}\rho V^2 S C_L = mg \tag{12}$$

In Eq. (12), ρ is the air density; S is the planform area of the UAV wing; C_L is the nondimensional lift coefficient; m and g are the mass of the UAV and the acceleration of gravity respectively.

The transition consists of two stages. The quadrotor mode controller firstly holds the altitude while the hybrid UAV speeding up forward. Then two controllers start transferring after the first stage according to the controlling weight in following Eqs. (13) and (14).

$$k_{fix} = \frac{Va - v_0}{v_1 - v_0}, v_0 \le v \le v_1 \tag{13}$$

$$k_{quad} = 1 - 0.4(t - t_0) \tag{14}$$

where v_0, v_1 are the minimum and the maximum speed respectively; t_0 is the start time of the second stage; t is the actual flight time.

4. MSPIO algorithm

4.1. PIO algorithm

PIO algorithm consisting of the map and compass operator and the landmark operator imitates the homing behavior of a flock of pigeons. It is assumed that there are N pigeons in the D-dimensional space with each being given the position $\boldsymbol{X}_i=[x_{i1}, x_{i2}, \ldots, x_{iD}]$ and the velocity $\boldsymbol{V}_i=[v_{i1}, v_{i2}, \ldots, v_{iD}]$. In the map and compass operator, the update equation is described as

$$\boldsymbol{V}_i(t)=\boldsymbol{V}_i(t-1)\cdot e^{-Rt}+\text{rand}\cdot\left(\boldsymbol{X}_{gbest}-\boldsymbol{X}_i(t-1)\right) \tag{15}$$

$$\boldsymbol{X}_i(t)=\boldsymbol{X}_i(t-1)+\boldsymbol{V}_i(t) \tag{16}$$

where R is the map and compass factor; $\boldsymbol{X}_{gbest}$ represents the global best location.

In the landmark operator, homing pigeons decrease by half at each iteration because of unfamiliarity with landmarks and the remaining update locations according to the center of the flock, which can be written as below.

$$N_p(t)=\frac{N_p(t-1)}{2} \tag{17}$$

$$\boldsymbol{X}_{center}(t)=\frac{\sum\boldsymbol{X}_i(t)\cdot F\left(\boldsymbol{X}_i(t)\right)}{\sum F\left(\boldsymbol{X}_i(t)\right)} \tag{18}$$

$$\boldsymbol{X}_i(t)=\boldsymbol{X}_i(t-1)+\text{rand}\cdot\left(\boldsymbol{X}_{center}(t-1)-\boldsymbol{X}_i(t-1)\right) \tag{19}$$

where N_p is the number of pigeons; $\boldsymbol{X}_{center}$ is the center of the flock; F ($\boldsymbol{X}_i$(t)) is the quality of each pigeon calculated from fitness value.

4.2. MSPIO

Four aspects can be explored to improve the PIO algorithm. **a**. Limiting the search space. It is better to set the search space around the optimal value because of the certain search bias in the iterative process of the pigeon flock. **b**. Modifying the inheritance manner. The pigeons' evolution is based on the last position or velocity and thus by altering the reliance of the known information, the search ability of the swarm can be enhanced or weakened at different stages of the iteration. **c**. Adjusting the search methodology. The PIO algorithm reduces the number of pigeons by half in the landmark operator to further increase the speed of gathering to the center. There are other search approaches, for example, setting the proportion of different operators and the acceptance of the local optimum. **d**. Reconfiguring the topology. The information flow in PIO is mainly transferred from the heading pigeon at the best position to others. In effect, the search ability can also be strengthened by defining the neighborhood of each pigeon and changing the number of interactions between neighbors. Aiming at enhancing the capability of solving the multi-dimensional search problem of the PIO algorithm, this paper proposes three kinds of strategies by changing the inheritance manner and the search methodology.

(1) Dynamic Inheritance

A nonlinear weight factor is proposed in Eq. (20) to dynamically change the ability of pigeons' position inheritance. Fig. 4 is the map of β_0 indicating that the position can be updated bidirectionally. The modified position update equation is written as Eq. (21).

$$\beta_0=\begin{cases}\sqrt{2\cdot\text{rand}}-1 & \text{rand}<0.5\\ 1-\sqrt{2\cdot(1-\text{rand})} & \text{rand}>0.5\end{cases} \tag{20}$$

$$\boldsymbol{X}_i(t)=\boldsymbol{X}_i(t-1)\cdot\beta_0-\text{rand}\cdot\left(\boldsymbol{X}_{gbest}-\boldsymbol{X}_i(t-1)\right) \tag{21}$$

(2) Hovering and Approaching

Inspired by Mirjalili, the hovering and the approaching will enhance the pigeons' exploration ability. [31]

In the Hovering Behavior, a factor is introduced as follows.

$$\omega=2\pi le^{bl} \tag{22}$$

where l=2rand-1, and b is a constant.

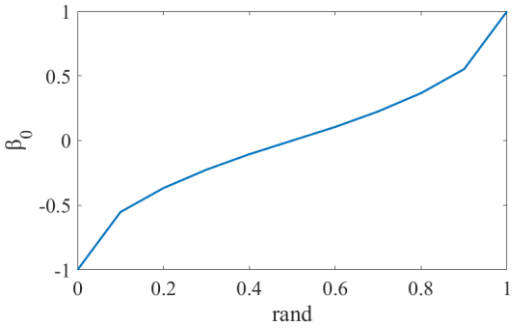


Fig. 4 Map of β_0

In the map and compass operator, another weight coefficient c is added before the social term and the velocity update equation is written as

$$\boldsymbol{V}_i(t) = \boldsymbol{V}_i(t-1) \cdot 2\pi l e^{bl} + c \cdot \text{rand} \cdot \left(\boldsymbol{X}_{gbest} - \boldsymbol{X}_i(t-1) \right) \tag{23}$$

The Approaching Behavior is used in the landmark operator as follows.

$$\boldsymbol{X}_i(t) = \boldsymbol{X}_{center}(t-1) - P \cdot \left(Q \cdot \boldsymbol{X}_{center}(t-1) - \boldsymbol{X}_i(t-1) \right) \tag{24}$$

where P=a(2rand-1), Q=2rand, a=2(1-n/N), n is the current iteration and N is the total iteration.

(3) Random Opposite Learning

As the pigeons may be in stagnation in local optimum, this paper proposes a random opposite learning method in Eq. (25).

$$\boldsymbol{X}_i(t) = \boldsymbol{Max} + \boldsymbol{Min} - \xi \cdot \boldsymbol{X}_i(t-1) \tag{25}$$

where $\xi = \left(2 \cdot \sqrt{\text{rand}} - 1\right)\left(1 + \text{rand}\right) / \text{rand}$; ***Max*** and ***Min*** are the upper bound and lower bound of the position, respectively.

The flow chart of MSPIO is shown in Fig. 5. The detailed implement procedure of MSPIO in the controller parameters optimization is described as follows.

Step 1: Initialize parameters of MSPIO, such as the pigeon population N_p, the number of dimensions D, iteration time N, etc. Set the position and the velocity of each pigeon randomly.

Step 2: Substitute parameters in the controller with the positions of pigeons and simulate the transition from hover to level flight to obtain the values of the pitch angle and the altitude within a fixed period.

Step 3: Evaluate the current best position by computing the designed fitness function and comparing the fitness value with the last optimal one.

Step 4: If the iteration time is less than the maximum number of the map and compass operator, evaluate whether the fitness value is in stagnation and selectively perform Random Opposite Learning.

Step 5: If the iteration time is less than the maximum number of the map and compass operator, conduct Hovering Behavior or Dynamic Inheritance to update the position and velocity of each pigeon. Otherwise, go to the landmark operator and perform Approaching or Hovering Behavior.

Step 6: Repeat steps 2-5. Terminate the optimization once the iteration time reaching the maximum value.

4.3. Parameters and fitness function in MSPIO

(1) Parameters

The performance of the MSPIO algorithm depends on parameter settings. There are three adjustable parameters: the probability terms P_1 and P_2 in Fig. 5, and the weight coefficient term c. Table 1 shows six items of parameters tested in the simulation correlating with Fig. 6. Because P_2 has little effect on the results according to the experiments, this term is not taken into consideration. Each flock of pigeons is initialized to the same position. It can be observed from Fig. 6 that the 3rd item performs the best with the lowest fitness value, indicating that the equal probability of the Dynamic Inheritance and the Hovering and Approaching Behavior achieves the best search capability.

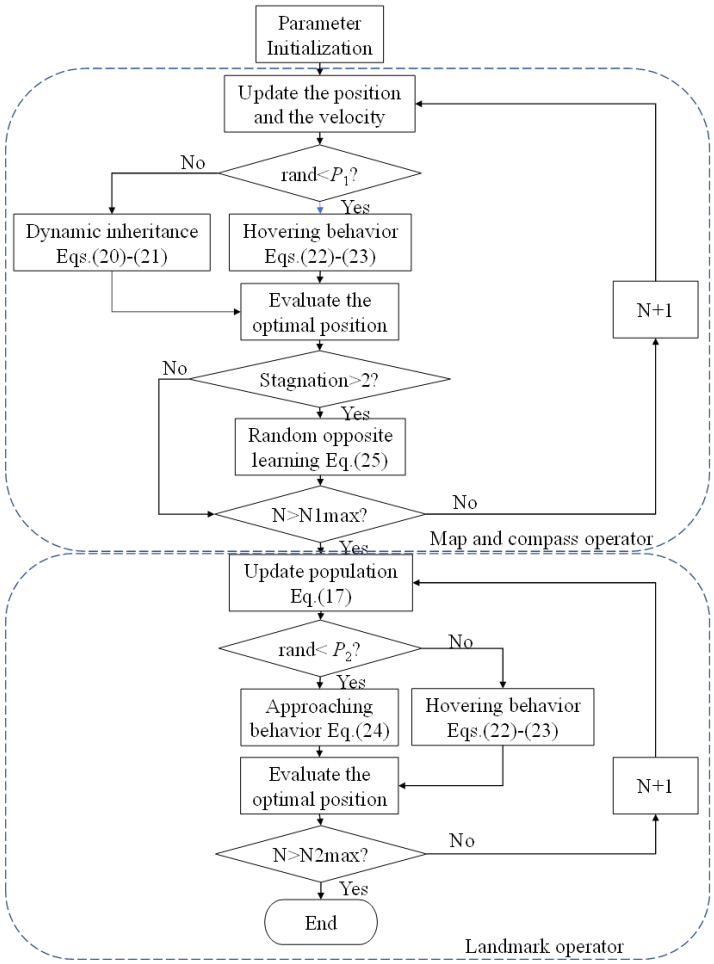


Fig. 5 The flow chart of MSPIO

Table 1 Six items tested in the simulation

	P_1	c	J
num1	0.1	1.3	1762.4
num2	0.1	2	1760.3
num3	0.5	1.3	1756.9
num4	0.5	2	1764.2
num5	0.9	1.3	1764.4
num6	0.9	2	1764.1

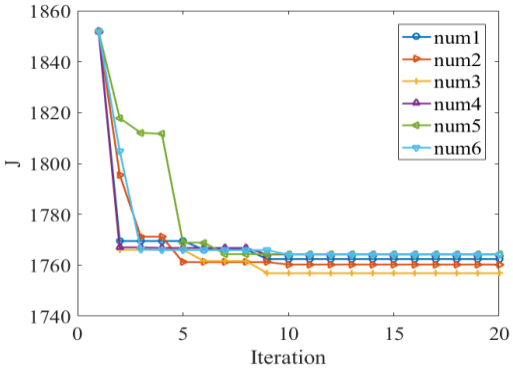


Fig. 6 Iteration of different items

(2) Fitness function

To achieve fast response, suppress the effect of chattering in the pitch angle θ and reduce the altitude fluctuation, the fitness function is set as

$$J = w_1 \cdot I_\theta + w_2 \cdot I_H \tag{26}$$

where w_1=1, w_2=3; I_θ and I_H are the integrated error of the states, pitch angle θ and altitude H respectively. The method to calculate I_θ and I_H is illustrated as follows.

Considering the errors between the outputs and the desired values, integrated time and absolute error (ITAE), integrated time and square error (ITSE), and integrated absolute error (IAE) are compared. The mathematical expressions are presented as follows respectively.

$$\text{ITAE}=\int_0^\infty t\left|e(t)\right|\mathrm{d}t \tag{27}$$

$$\text{ITSE}=\int_0^\infty te^2(t)\mathrm{d}t \tag{28}$$

$$\text{IAE}=\int_0^\infty \left|e(t)\right|\mathrm{d}t \tag{29}$$

Figs. 7 and 8 show the pitch angle θ and the altitude H of three fitness functions during the hover to level flight transition mode. ITAE performs the best with a narrow margin.

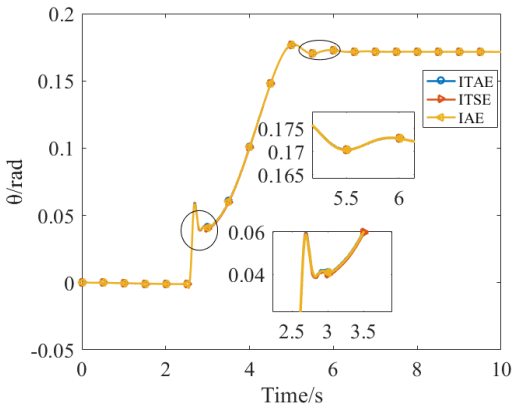


Fig. 7 Pitch angle comparison of 3 fitness functions

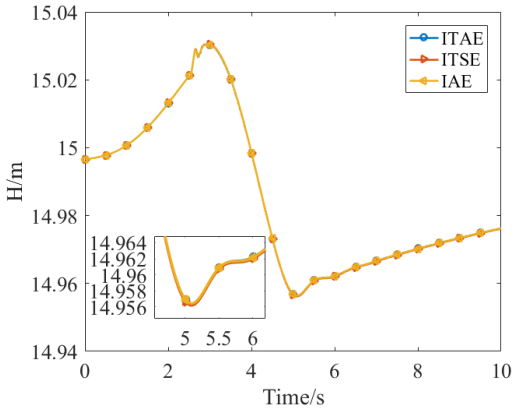


Fig. 8 Height comparison of 3 fitness functions

5. Simulation results and analysis

5.1. Simulation of benchmark functions

MSPIO are applied to eight benchmark functions including unimodal and multimodal functions to testify the fast convergence speed and better exploitation ability of the optimization algorithm. The benchmark functions are listed in Table 2. The particle swarm optimization (PSO), genetic algorithm (GA), PIO and an improved PIO named CMPIO, which is also proposed for parameter tuning, are compared. [28]

The results are listed in Table 3 and the convergence curves are provided in Fig. 9. The standard deviations (STDs) of MSPIO in the eight experiments are the smallest, demonstrating the stability of MSPIO. Moreover, the exploitation ability of MSPIO can be confirmed comparing the maximum (Max) and the minimum (Min) values. From Fig. 9, it can be concluded that MSPIO converges the fastest among the five optimization algorithms which also testifies the superiority of MSPIO.

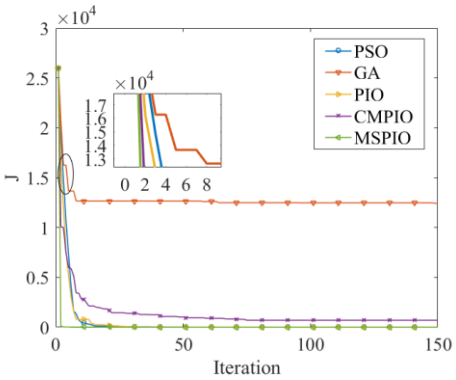
Table 2 Benchmark functions

Function	Expression	Dimension	Range
Sphere	$F_1(x)=\sum_{i=1}^{d}x_i^2$	16	[-100,100]
Schwefel_2.21	$F_2=\max\{\lvert x_i\rvert\}$	16	[-100,100]
Schwefel_2.22	$F_3=\sum_{i=1}^{d}\lvert x_i\rvert+\prod_{i=1}^{d}\lvert x_i\rvert$	16	[-10,10]
Step	$F_4=\sum_{i=1}^{d}\lvert x_i+0.5\rvert^2$	16	[-100,100]
Rastrigin	$F_5=10d+\sum_{i=1}^{d}\left(x_i^2-10\cos(2\pi x_i)\right)$	16	[-5,5]
Ackley	$F_6=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)+20+e$	16	[-32,32]
Griewangk	$F_7=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	16	[-600,600]
RosenBrock	$F_8=\sum_{i=1}^{d-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	16	[-30,30]

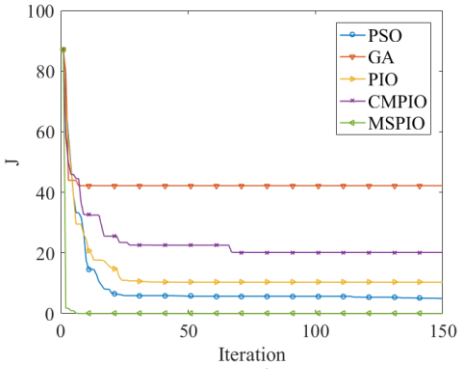
Table 3 Results of benchmark functions

Function	Algorithm	Min	Max	STD
Sphere	PSO	1.44E-01	1.24E+01	3.76E+00
	GA	7.03E+03	1.70E+04	2.69E+03
	PIO	1.17E+01	2.07E+02	5.20E+01
	CMPIO	1.01E+02	1.01E+03	2.38E+02
	MSPIO	1.43E-18	6.02E-03	1.12E-03
Schwefel_2.21	PSO	2.26E+00	1.33E+01	2.24E+00
	GA	2.92E+01	5.64E+01	6.53E+00
	PIO	4.04E+00	1.99E+01	3.51E+00
	CMPIO	1.04E+01	3.77E+01	6.28E+00
	MSPIO	2.68E-07	3.46E-02	7.19E-03
Schwefel_2.22	PSO	4.99E-01	4.00E+00	6.73E-01
	GA	2.00E+01	3.92E+01	4.20E+00
	PIO	1.73E+00	6.43E+00	1.29E+00
	CMPIO	4.17E+00	1.53E+01	2.63E+00
	MSPIO	8.25E-07	9.44E-03	2.45E-03
Step	PSO	2.91E-01	9.81E+00	2.28E+00
	GA	4.88E+03	1.62E+04	2.90E+03
	PIO	9.58E+00	2.95E+02	6.29E+01
	CMPIO	1.72E+02	1.06E+03	2.13E+02
	MSPIO	2.33E+00	3.85E+00	3.71E-01
Rastrigin	PSO	7.26E+00	2.48E+01	4.42E+00
	GA	6.23E+01	1.38E+02	1.88E+01
	PIO	1.27E+01	8.16E+01	1.42E+01
	CMPIO	5.34E+01	1.18E+02	1.59E+01
	MSPIO	0.00E+00	2.65E-03	4.85E-04
Ackley	PSO	1.39E+00	6.39E+00	1.13E+00
	GA	1.10E+01	1.80E+01	1.58E+00

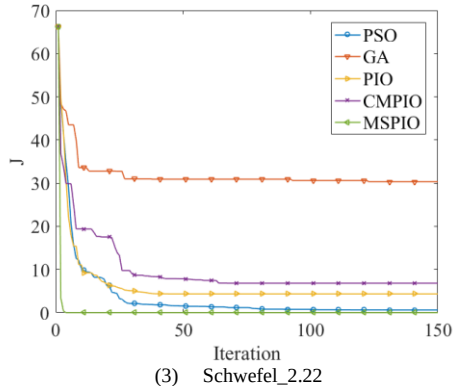
	PIO	3.25E+00	8.35E+00	1.08E+00
	CMPIO	3.54E-01	1.60E+01	3.51E+00
	MSPIO	7.49E-11	2.85E-02	6.68E-03
Griewangk	PSO	2.01E-01	9.36E-01	1.91E-01
	GA	5.34E+01	1.59E+02	2.38E+01
	PIO	1.34E+00	3.69E+00	5.93E-01
	CMPIO	2.47E+00	1.07E+01	2.23E+00
	MSPIO	1.37E-11	5.69E-03	1.36E-03
Rosenbrock	PSO	2.31E+01	4.61E+02	1.18E+02
	GA	1.16E+05	3.17E+05	6.59E+04
	PIO	8.89E+02	1.52E+04	2.96E+03
	CMPIO	2.33E+03	1.42E+04	2.65E+03
	MSPIO	1.50E+01	1.50E+01	1.23E-02



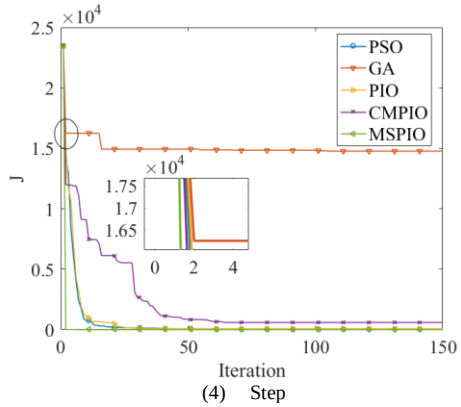
(1) Sphere



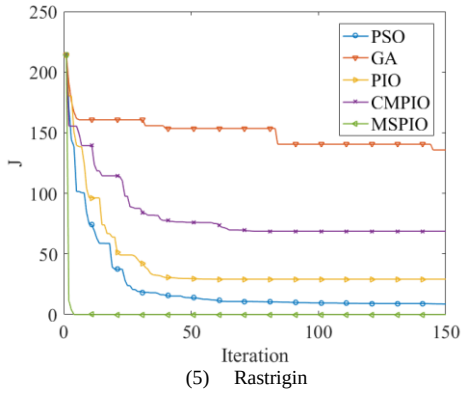
(2) Schwefel_2.21



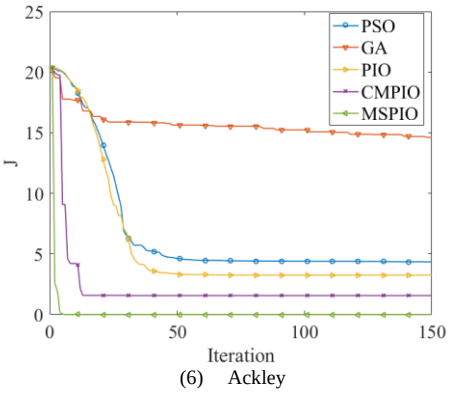
(3) Schwefel_2.22



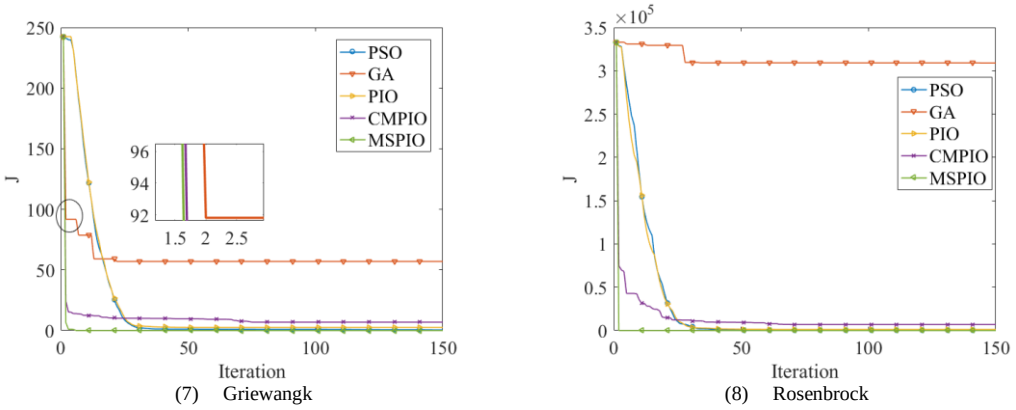
(4) Step



(5) Rastrigin



(6) Ackley



(7) Griewangk (8) Rosenbrock

Fig. 9 Convergence curves of the optimization algorithms

5.2. Parameter optimization

The parameters in the controllers to be optimized are the r_0, h in TD of the ADRC controller in the quadrotor mode; β_1, β_2, b_0 in the NLSEF; β_{11}, β_{12}, β_{13} in the ESO of both ADRC controllers and P, I in the altitude control of the fixed-wing mode controller.

Fig. 10 reveals the search ability of the five algorithms. When dealing with a multi-dimensional optimization problem, MSPIO converges the fastest in the first 2 iterations and owns the lowest fitness value. During the iteration, immaturity results in two platforms. Then the fitness value decreases since the stagnations invoke the random opposite learning strategy. Table 4 shows the optimized parameters of the controllers according to MSPIO.

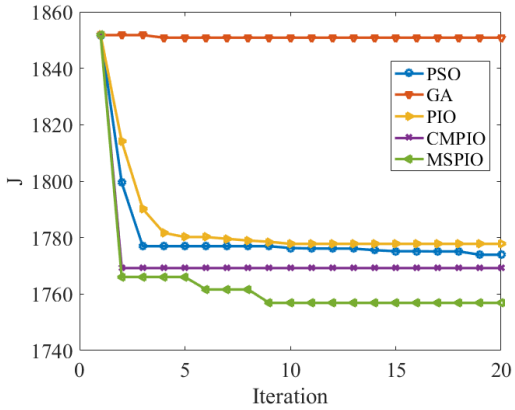


Fig. 10 Convergence curves of the optimization algorithms

Fig. 11 indicates the velocity change during the transition. The comparisons of the states θ and H are shown in Figs. 12 and 13. The curve of the pitch angle reveals two significant changes during the weight conversion process. The first change results from the rise of θ as it goes up rapidly to a high point then drops back, indicating the weight transform between the quadrotor mode controller and the fixed-wing mode controller. According to Eq. (13), the controlling weight is calculated by the velocity and therefore when executing the weight transform, the controller of the fixed-wing mode soon takes over. The control ability of the quadrotor mode decreases, leading to a sudden increase in θ. The second change happens when the states are controlled by the fixed-wing mode controller, θ keeps rising with a chattering phenomenon before being steady.

Table 4 Optimized parameters of the controllers

Parameter	Description	Value
β_{11_quad}	Parameter in the ESO of the quadrotor mode	500
β_{12_quad}	Parameter in the ESO of the quadrotor mode	8000
β_{13_quad}	Parameter in the ESO of the quadrotor mode	8000
b_{0_quad}	Parameter in the NLSEF of the quadrotor mode	39.6465
β_{1_quad}	Parameter in the NLSEF of the quadrotor mode	10
β_{2_quad}	Parameter in the NLSEF of the quadrotor mode	0.02
r_0	Parameter in the TD of the quadrotor mode	10

h	Parameter in the TD of the quadrotor mode	0.05
β_{11_fix}	Parameter in the ESO of the fixed-wing mode	50
β_{12_fix}	Parameter in the ESO of the fixed-wing mode	2000
β_{13_fix}	Parameter in the ESO of the fixed-wing mode	4000
b_{0_fix}	Parameter in the NLSEF of the fixed-wing mode	12
β_{1_fix}	Parameter in the NLSEF of the fixed-wing mode	8
β_{2_fix}	Parameter in the NLSEF of the fixed-wing mode	0.1992
K_p	Parameter in the altitude controller of the fixed-wing mode	1.8
K_i	Parameter in the altitude controller of the fixed-wing mode	0.2

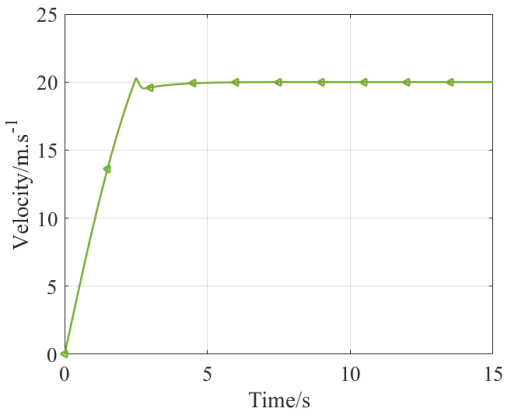


Fig. 11 Velocity change in the transition mode

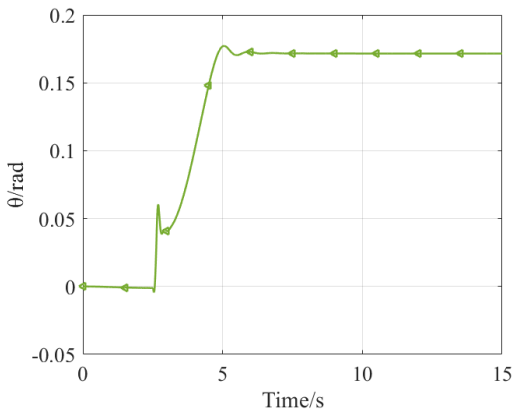


Fig. 12 Pitch angle of the selected method

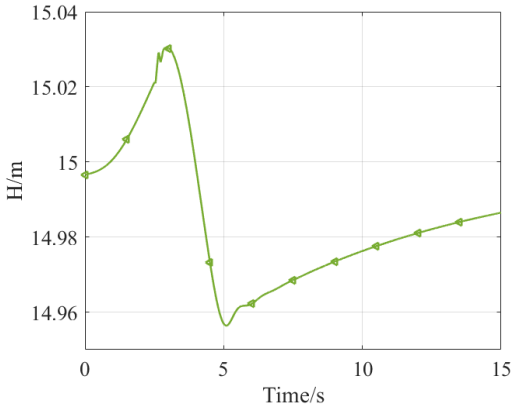


Fig. 13 Height change of the selected method

5.3. Comparison with PID control method and initial ADRC method

The presented transition approach is compared with a pure PID control method and the initial non-optimized ADRC method to verify the efficacy. The longitudinal motion is still the focus. In the figures below, 0-5s is the period of forward transition. The only difference between the two control methods lies in the pitch loop and therefore in Fig. 14, the velocity changes containing the transition process are identical.

Fig. 15 is the pitch angle of the hybrid UAV. During the first stage, the blue line representing the PID control method goes down as the UAV is mainly controlled by the quadrotor mode controller and the fixed-wing mode controller only controls the forward airspeed. Then the elevon command is zero. The coefficient $C_{m_\alpha} < 0$ indicates the aircraft is longitudinal static stable and the UAV intends to pitch down to maintain zero pitching moment, resulting in a sudden rise of 5°. However, the change of the pitch angle controlled by the optimized ADRC controller is smaller than that in the pure PID method and that controlled by the initial ADRC controller. Compared with the initial ADRC controller, the optimized one has a shorter convergence time and smaller overshooting after 5s. The pure PID method

has the longest convergence time.

In Fig. 16, the overshooting of the height in the pure PID method is 0.14m while that in the ADRC controller is 0.04m. The proposed controller aims to minish the altitude change during the transition. It can be concluded that both two ADRC controllers have achieved the goal of minishing the altitude fluctuation and the performance is better than the pure PID. The highest altitude of the two ADRC controllers is similar but the lowest altitude of the initial ADRC is 14.89m, while that of the optimized ADRC controller is 14.96 m, demonstrating the effectiveness of the improved algorithm.

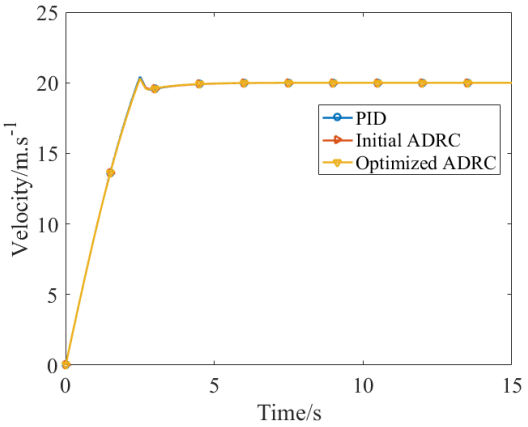


Fig. 14 Velocity change comparison of PID and ADRC

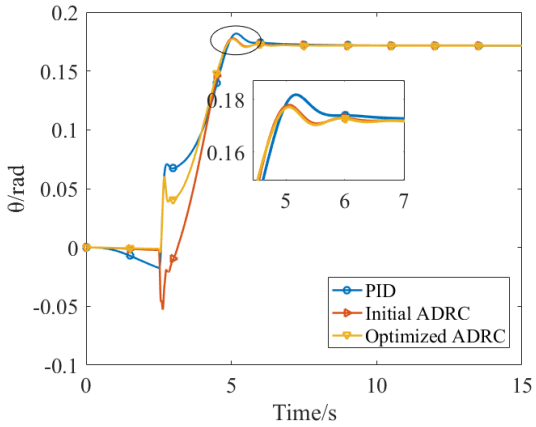


Fig. 15 Pitch angle comparison of PID and ADRC

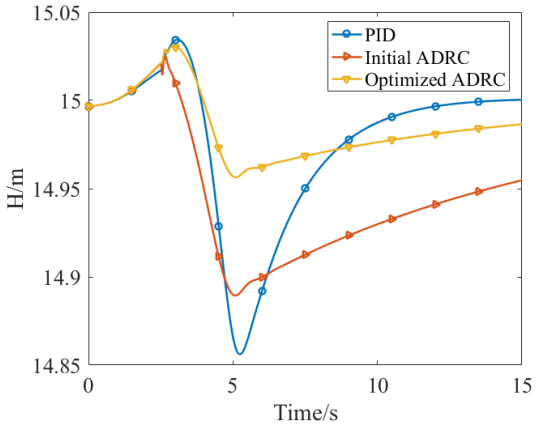


Fig. 16 Altitude comparison of PID and ADRC

6. Conclusion

(1) In this paper, a novel transition approach is proposed employing ADRC to address the problem of the height fluctuation. The approach consists of two parts. First, two ADRC controllers are developed respectively for the quadrotor mode and the fixed-wing mode. Second, a strategy is designed for the transition from the quadrotor mode to the fixed-wing mode. Furthermore, to enhance the efficiency of the controllers, parameter optimization is realized by MSPIO. The simulation result is compared with that of the PID method and that of the initial non-optimized ADRC controller. The ADRC method with optimal parameters owns the least convergence time and the smallest chattering of the pitch angle after the transition from hover to level flight. In addition, the altitude fluctuation is the smallest, demonstrating the effectiveness of the designed control method.

(2) The basic PIO is improved for better efficiency and search ability by utilizing multiple strategies, including Dynamic Inheritance, Hovering and Approaching, and Random Opposite Learning. The most significant advantage of the proposed algorithm is the ability to address multi-dimensional search problems with fast convergence speed

and the efficacy of the proposed algorithm is manifested by being compared with PSO, GA, PIO, and CMPIO. The stability analysis of the transition mode and theoretical research on the proposed optimization algorithm are expected to be conducted in the future.

Acknowledgements

This work was partially supported by Science and Technology Innovation 2030-Key Project of "New Generation Artificial Intelligence" under grant #2018AAA0100803, National Natural Science Foundation of China under grant #U20B2071, #91948204, and #U1913602, and Aeronautical Foundation of China under grant #20185851022

References

1. He W, Meng T, Zhang S, et al. Trajectory tracking control for the flexible wings of a micro aerial vehicle. *IEEE Trans Syst Man Cybern* 2018;48(12):2431-2441.
2. Zaludin Z and Gires E. Automatic flight control requirements for transition flight phases when converting long endurance fixed wing UAV to VTOL aircraft. 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS); 2019 Jun 29-29; Selangor, Malaysia. Piscataway: IEEE; 2019. p. 273-278.
3. Hudati I, Rusdhianto Effendie AK and Jazidie A. Transition control on hybrid unmanned aerial vehicles (UAV) using altitude change. 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA); 2019 Aug 28-29; Surabaya, Indonesia. Piscataway: IEEE; 2019. p. 276-281.
4. Zhong, J, Wang, C. Transition characteristics for a small tail-sitter unmanned aerial vehicle. *Chinese J Aeronaut* [Internet]. Available from: https://doi.org/10.1016/j.cja.2020.12.021
5. Manzoor T, Xia Y, Zhai D, Ma D. Trajectory tracking control of a VTOL unmanned aerial vehicle using offset-free tracking MPC. *Chinese J Aeronaut* 2020;33(7):2024-2042.
6. Gu H, Lyu X, Li Z, Shen S and Zhang F. Development and experimental verification of a hybrid vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV). 2017 International Conference on Unmanned Aircraft Systems (ICUAS); 2017 Jun 13-16; Miami, FL, USA. Piscataway: IEEE; 2017. p. 160-169.
7. Malang Y. Design and control of a vertical takeoff and landing fixed-wing unmanned aerial vehicle [Dissertation]. Toronto: University of Toronto; 2016.
8. Liu Z, Theilliol D, Yang L, et al. Observer-based linear parameter varying control design with unmeasurable varying parameters under sensor faults for quad-tilt rotor unmanned aerial vehicle. *Aerosp Sci Technol* 2019; 92:696-713.
9. Hochstenbach M, Notteboom C, Theys B, et al. Design and control of an unmanned aerial vehicle for autonomous parcel delivery with transition from vertical take-off to forward flight−vertical, a quadcopter tailsitter. *Int J Micro Air Veh* 2015;7(4):395-405.
10. Wang Y, Zhou Y, C. Lin. Modeling and control for the mode transition of a novel tilt-wing UAV. *Aerosp Sci Technol* 2019;91:593-606.
11. Yayla M, Kutay A, Senipek M, et al. An adaptive flight controller design for a tilt-prop fixed wing UAV for all flight modes. AIAA Scitech 2020 Forum; 2020 Jan 6-10; Orlando, FL, USA. New York: AIAA; 2020.
12. Liu N, Cai Z, Zhao j, et al. Predictor-based model reference adaptive roll and yaw control of a quad-tiltrotor UAV. *Chinese J Aeronaut* 2020;33(1):282-295.
13. Han J. From PID technique to active disturbances rejection control technique. *Control Engineering of China* 2002;9(3):13-18 [Chinese].
14. Gao Z. Active disturbance rejection control: a paradigm shift in feedback control system design. 2006 American Control Conference; 2006 Jun 14-16; Minneapolis, MN, USA. Piscataway: IEEE; 2006. p. 2399-2405.
15. Aydemir M, Arıkan K B. Evaluation of the disturbance rejection performance of an aerial manipulator. *J Intell Robot Syst* 2020;97:451-469.
16. Najm A A, Ibraheem I K. Altitude and attitude stabilization of UAV quadrotor system using improved active disturbance rejection control. *Arab J Sci Eng* 2020;45(3):1985-1999.
17. Chen Y, Liang J, Wang C, et al. Combined of Lyapunov-stable and active disturbance rejection control for the path following of a small unmanned aerial vehicle. *Int J Adv Robot Syst* 2017;14(2):1-10.
18. Kori D K, Kolhe J P, Talole S E. Extended state observer based robust control of wing rock motion. *Aerosp Sci Technol* 2014;33:107-117.
19. Duan H, Qiao P. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *Int J Intell Comput Cybern* 2014;7(1):24-37.
20. Duan H, and Qiu H. Advancements in pigeon-inspired optimization and its variants. *Sci China Inform Sci* 2019;62:70201.
21. Duan H, Wang X. Echo state networks with orthogonal pigeon-inspired optimization for image restoration. *IEEE*

Trans Neur Net Lear 2016;27(11):2413-2425.
22. Zhang B, Duan H. Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment. *IEEE-ACM Trans Comput Bi* 2017;14(1):97-107.
23. Hua B, Huang Y, Wu Y, et al. Spacecraft formation reconfiguration trajectory planning with avoidance constraints using adaptive pigeon-inspired optimization. *Sci China Inform Sci* 2019;62:70209.
24. Huo M, Duan H, Fan Y. Pigeon-inspired circular formation control for multi-UAV system with limited target information. *Guidance, Navigation and Control (GNC)* 2021;1(1):2150004 (23pages).
25. Xu X, Deng Y. UAV power component-DC brushless motor design with merging adjacent-disturbances and integrated-dispatching pigeon-inspired optimization. *IEEE Trans Magn* 2018;54(8):1-7.
26. Hai X, Wang Z, Feng Q, et al. Mobile robot ADRC with an automatic parameter tuning mechanism via modified pigeon-Inspired optimization. *IEEE-ASME Trans Mech* 2019;24(6):2616-2626.
27. Sun Y, Duan H, Xian N. Fractional-order controllers optimized via heterogeneous comprehensive learning pigeon-inspired optimization for autonomous aerial refueling hose-drogue system. *Aerosp Sci Technol* 2018;81:1-13.
28. Yang Z, Duan H, Fan Y, et al. Automatic carrier landing system multilayer parameter design based on Cauchy mutation pigeon-inspired optimization. *Aerosp Sci Technol* 2018;79:518-530.
29. Chen L, Duan H, Fan Y, et al. Multi-objective clustering analysis via combinatorial pigeon inspired optimization. *Sci China Inform Sci* 2020;63:1302-1313.
30. Beard R W, Mclain T W. *Small Unmanned Aircraft: Theory and Practice*. 1st ed. New Jersey: Princeton University Press; 2012. p. 56-62.
31. Mirjalili S, and Lewis A. The whale optimization algorithm. *Adv Eng Softw* 2016;95:51-67.