Autonomous maneuver decision for unmanned aerial vehicle via improved pigeon-inspired optimization

Haibin Duan, Senior Member, IEEE
Beihang University, Beijing, China

Yangqi Lei
Beihang University, Beijing, China

Jie Xia
Beihang University, Beijing, China

Yimin Deng
Beihang University, Beijing, China

Yuhui Shi, Fellow, IEEE
Southern University of Science and Technology, Shenzhen, China

Abstract-**As a crucial technology of air-to-air confrontation, autonomous maneuver decision has attracted wide attention in recent years. This paper proposes an improved pigeon-inspired optimization method to realize autonomous maneuver decision for unmanned aerial vehicles (UAVs) rapidly and accurately in an aerial combat engagement. The maneuver library is designed, including some advanced offensive and defensive maneuvers. A dependent set of trial maneuvers is generated to help UAVs make decisions in any tactical situation, and a future engagement state of the opponent UAV is predicted for each trial maneuver. The core of the decision-making process is that the objective function to be optimized is designed using the game mixed strategy, and the optimal mixed strategy is obtained by the improved pigeon-inspired optimization. A comparative analysis with other classical optimization algorithms highlights the advantage of the proposed algorithm. The simulation tests are conducted under four different initial conditions, namely, neutral, offensive, opposite, and defensive conditions. The simulation results verify the effectiveness of the proposed autonomous maneuver decision method.**

Index Terms-**Unmanned aerial vehicle (UAV), autonomous maneuver decision, air-to-air confrontation, maneuver library, improved pigeon-inspired optimization**
[1]

Manuscript received October 17, 2021; revised February 20, 2022, June 12, 2022, and August 13, 2022.

DOI. XXXXXX

Refereeing of this contribution was handled by XXXXX.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have attracted more and more attentions owing to their significant advantages in performing complex tasks [1], such as aerial surveillance, trajectory planning, and aerial refueling [2-6]. With the improvement in the performance of UAVs, they are likely to become a mainstay, able to perform autonomous aerial combat and gain air dominance [7]. Autonomous aerial combat is a complex mission including situation awareness, maneuver decision and motion control. Among them, autonomous maneuver decision is the core of air combat, and how to realize autonomous maneuver decision for UAVs is a crucial and challenging issue. Many algorithms have been proposed to investigate the autonomous maneuver decision problem, which can be divided into mathematical solution, data-driven, and intelligent optimization method.

This work was partially supported by Science and Technology Innovation 2030-Key Project of "New Generation Artificial Intelligence" under grant #2018AAA0102303, National Natural Science Foundation of China under grant #U20B2071, #91948204, #T2121003 and #U1913602.

Authors' addresses: Haibin Duan, Yangqi Lei, Yimin Deng, and Jie Xia are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China. Yuhui Shi is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, 518000, China. E-mail: (hbduan@buaa.edu.cn, leiyangqi@buaa.edu.cn, xiaj@buaa.edu.cn, ymdeng@buaa.edu.cn, and shiyh@sustc.edu.cn).

(*Corresponding author: Haibin Duan*)

The mathematical solution methods include pursuit-evasion game [8], differential game [9,10], game theory [11,12], influence graph [13], and nonlinear control [14]. For example, the differential game method describes the decision-making process as a differential equation and determines the optimal maneuver of a UAV against its target. Game theory has been successfully applied to the aerial combat maneuver decision problem. Nelson et al. [11] considered the air combat problem as a zero-sum solved by a min-max search algorithm. However, the elemental maneuver library is considered inaccurate and does not consider the dynamic characteristics. Li et al. [12] proposed a constraint strategy game model for time-sensitive information in an aerial combat environment. Although these mathematical solution methods enable the UAV to respond to changes in the tactical situation, the mathematical formulations of the equations are quite complex. Due to the limitation of the real-time performance, it is difficult to apply these approaches in a complex combat environment.

The data-driven methods include Bayesian theory [15], approximate dynamic programming [16], neural network [17], and reinforcement learning [18]. Aerial combat is regarded as a Markov process, and the Bayesian inference theory is utilized to ensure the superiority situation for UAV [15]. McGrew et al. [16] solved the aerial combat problem using the approximate dynamic programming approach. However, the point mass model was used, and it could not reflect the flight performance of the UAV. A maneuver decision method based on a deep neural network is proposed to improve the combat ability of UAVs [17]. However, the deep learning method needs many aerial combat samples for its learning process, which are not quite available. An autonomous maneuver decision method based on a deep Q network is proposed for the UAV in short-range aerial combat [18], and obtains an effective decision policy to defeat the opponent. However, a large number of training cases are required to attain a competitive level of performance, and it is difficult to verify the physical meaning of the novel maneuvers generated.

Intelligent optimization algorithms have been successfully utilized for the aerial combat maneuver decision, including genetic algorithm (GA) [19-20], particle swarm optimization (PSO) [21], and some novel algorithms imitating swarm intelligent behaviors. Smith et al. [19] proposed a decision model for maneuvering decisions based on a genetic learning system. The simulation results demonstrated the ability of the genetic algorithm-based learning system to discover novel tactics in a dynamic air combat environment. Ernest et al. [20] proposed a genetic algorithm based fuzzy inference system, namely ALPHA. ALPHA successfully defeated two jet fighters operated by retired fighter pilots in air combat simulation. A grey wolf optimizer method was proposed to solve the uncertain factors of UAVs in aerial combat [22], which could be used for real-time performance. Although the methods above achieved some success in aerial combat, the objective function designed for the problem is complex, and the efficiency and search ability of the algorithm are not satisfied. Motivated by these facts, another intelligent optimization method is proposed.

Pigeon-inspired optimization (PIO) [23] is a novel optimization algorithm proposed in recent years, which imitates the navigation behavior of the homing pigeons. The PIO algorithm was applied in many fields, such as UAV formation, parameter tunning, path planning [24-26], etc. However, the basic PIO still has some shortcomings such as fast prematurity and less search space. Therefore, several variants were put forward to overcome these weaknesses [27, 28].

In this paper, an improved PIO algorithm namely CLPPPIO algorithm is proposed to handle the maneuver decision problem. To begin with, the guidance law based on the maneuver library is established, and the UAV performs the recommended maneuver according to the guidance law. Then, when the tactical situation can not match the precondition of the guidance law, the maneuver decision problem is transformed into the optimization problem. The optimal maneuver is selected through the CLPPPIO algorithm. The main contributions of the proposed method are as follows:

1) Different from the point-mass model used in [16], a nonlinear six-degree-of (6-DOF) freedom UAV model is established for dogfight engagement, and the flight path commands are generated from the proposed method are fed into the 6-DOF model. Instead of using an elemental maneuver library, a novel maneuver library including some advanced defensive and offensive maneuvers is designed, by considering the dynamic flight characteristic.

2) To improve the autonomy of UAVs and allow them to make maneuver decisions in any tactical situation, a set of trial maneuvers is generated, and the situation and state of motion of the opponent UAV for each trial maneuver are predicted. The decision-making problem is transformed into the optimization problem. The objective function is based on the situation assessment and the game mixed strategy.

3) A novel intelligent optimization algorithm namely CLPPPIO is proposed, and the maneuver decision process via CLPPPIO is detailed. The proposed algorithm performs fast convergence speed and finds a better solution compared with other existing intelligent optimization algorithms. Specifically, the convergence and time complexity of CLPPPIO are analyzed.

The rest of the paper is organized as follows. Section II presents the problem statement. In section III, the autonomous maneuver decision method based on

CLPPPIO is designed. In section IV, various experiments are conducted, and results are given to prove the effectiveness of the proposed method for autonomous maneuver decision. Conclusions are presented in section V.

II. PROBLEM STATEMENT

A. UAV DYNAMICS

A six-degree of freedom UAV model is considered for autonomous maneuver decision [25], which is controlled by throttle, elevator, aileron, and rudder. The mathematical model of the model is described as

$$\begin{cases} \dot{x}_g = u\cos\theta\cos\psi \\ \qquad +v(\sin\theta\sin\phi\cos\psi-\cos\phi\sin\psi) \\ \qquad +w(\sin\phi\sin\psi+\cos\phi\sin\theta\cos\psi) \\ \dot{y}_g = u\cos\theta\sin\psi \\ \qquad +v(\sin\phi\sin\theta\sin\psi+\cos\phi\sin\psi) \\ \qquad +w(-\sin\phi\cos\psi+\cos\phi\sin\theta\sin\psi) \\ \dot{h}=u\sin\theta-v\sin\phi\cos\theta-w\cos\phi\cos\theta \\ \dot{V}=\dfrac{u\dot{u}+v\dot{v}+w\dot{w}}{V} \\ \dot{\alpha}=\dfrac{u\dot{w}-w\dot{u}}{u^2+w^2} \\ \dot{\beta}=\dfrac{\dot{v}V-v\dot{V}}{V^2\cos\beta} \\ \dot{\phi}=p+(r\cos\phi+q\sin\phi)\tan\theta \\ \dot{\theta}=q\cos\phi-r\sin\phi \\ \dot{\psi}=\dfrac{1}{\cos\theta}(r\cos\phi+q\sin\phi) \\ \dot{p}=\dfrac{1}{I_xI_z-I_{xz}^2} \\ \left[I_zL+I_{xz}N+(I_x-I_y+I_z)I_{xz}pq+(I_yI_z-I_z{}^2+I_{xz}^2)qr\right] \\ \dot{q}=\dfrac{1}{I_y}\left[M-I_{xz}(p^2-r^2)\right] \\ \dot{r}=\dfrac{1}{I_xI_y-I_{xz}^2} \end{cases} \tag{1}$$

where x_g, y_g, z_g denote position coordinates of UAV in the inertial reference frame, V,α,β are ground velocity, attack angle, and sideslip angle, respectively. ϕ,θ,ψ are roll angle, pitch angle, and yaw angle. p,q,r are angular rates in the body-fixed reference frame. I_x, I_y, I_z denote the coordinate components of inertia moment. L, M, N are moments along the axis in the body-fixed reference frame.

B. UAV CONTROLLER

The longitudinal and lateral control laws based on proportional-integral-derivative (PID) algorithm are designed.

The longitudinal channel refers to pitch angle, altitude, and velocity control through elevator. The short-period damping of the longitudinal channel is improved by giving pitch angle rate feedback. The longitudinal loop control law is expressed as:

$$\delta_e = k_\alpha(\alpha-\alpha_{com})+k_q(q-q_{ref}) \tag{2}$$

where α is the attack angle, q is the pitch angle rate, and PID control law is:

$$k_\alpha = K_{P\alpha}+K_I(\frac{1}{s})+K_Ds \tag{3}$$

The lateral movement refers to roll and yaw movements, which are achieved by aileron and rudder channels. For the aileron channel, the feedback of roll angle and roll angle rate are considered. For the rudder channel, the feedback of yaw angle rate is used to increase the damping of the Dutch roll mode, and the lateral overload feedback is beneficial to increase the Dutch roll frequency. The lateral loop control law is:

$$\begin{cases} \delta_a = K_\phi(\phi-\phi_{com})+K_pp \\ \delta_r = K_rr-K_\beta\beta \end{cases} \tag{4}$$

where ϕ_{com} is the given roll angle command, K_p, K_ϕ, K_r and K_β are the proportional coefficients.

The UAV ardupilot is designed based on the above longitudinal and lateral control systems, and the inputs are attack angle command and roll angle command. The architecture block diagram of UAV control system is shown in Fig. 1.

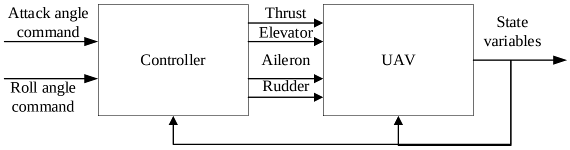


Fig. 1. The architecture block diagram of UAV control system.

C. CONSTRUCTION OF COMBAT GEOMETRY

Fig. 2 displays the relative geometric relationship between the attacker and the target. R is the distance vector between the attacker and the target, which is the line of sight. V_A and V_T are the velocity vectors of the attacker and the target, respectively. λ_A and λ_T are the deviation angles between the line of sight and the velocity vectors of the attacker and the target, which are expressed as

$$\begin{cases} \lambda_A = \cos^{-1}\left[\dfrac{V_A\cdot R}{|V_A||R|}\right] \\ \lambda_T = \cos^{-1}\left[\dfrac{V_T\cdot R}{|V_T||R|}\right] \end{cases} \tag{5}$$

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2022.3221691

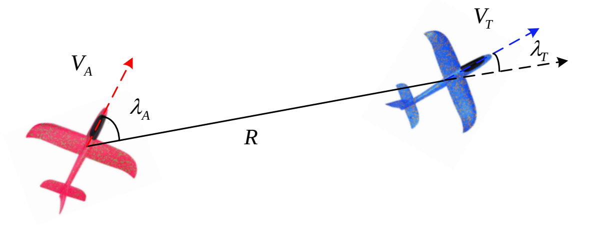


Fig. 2. Deviation angles of the attacker and the target.

As shown in Fig.2, the weapon engagement zone is defined as Z^A in which the attacker UAV can employ its weapon against the target UAV.

$$Z^A = \{(x_T, y_T, z_T) : \lambda_{TA} < \lambda_A^{\max}, R_{\min} \le R \le R_{\max}\} \quad (6)$$

where λ_{TA} is the radar angle of the attacker, $\lambda_A^{\max}$, $R_{\min}$, $R_{\max}$ indicate the maximum radar angle, minimum range, and maximum range at which weapons can be employed.

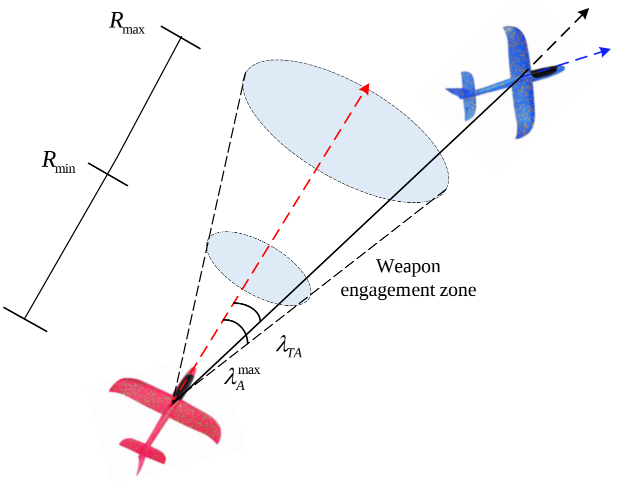


Fig. 3. Weapon engagement zone.

D. MANEUVER LIBRARY

Maneuver library is an optional set for maneuvering decision-making. Currently, there are two ways to design maneuver library, a typical tactical maneuver library and an elemental maneuver library. The elemental maneuver library is based on airborne maneuvers, the most famous being the seven elemental maneuver library designed by NASA scholars. Compared with the elemental maneuver library, the typical tactical maneuver library has the advantage of the flexibility. It contains some advanced offensive and defensive maneuvers which are more in line with the actual application requirements and have higher practicability. Therefore, a novel maneuver library is designed based on typical tactical maneuver library.

The maneuver library contains barrel-roll, chandelle, break-turn, high-yo-yo, low-yo-yo, vertical-turn-up, point-maneuver, break-hover, boom-zoom, split-s, and zoom-reverse maneuvers. The typical tactical maneuver can be designed as a sequence of actions. The calculation command in each step consists of three commands: normal load factor, roll angle, and thrust. Take high-yo-yo maneuver as an example. The high-yo-yo maneuver is a kind of offensive maneuver, that tends to increase the UAV's altitude using kinetic energy and then search for opportunities to hit the target. It is accomplished by pulling up the UAV and rolling back to the target altitude. The increasing height is determined by:

$$\delta H = \frac{C}{4}\frac{V^2}{2g} \quad (7)$$

where C is a constant indicating the intensity of maneuver, and $\frac{V^2}{2g}$ represents the operator by converting the energy into the altitude.

After attaining the desired height, the UAV dives back into the plane of the target through the roll angle command γ. Then, it aims at the target through the normal load factor and bank angle command. The normal load factors along the y-axis and z-axis are:

$$n_y = k_2(\frac{V_A}{180}) \times (\psi_{AT} - \psi_{yaw}) \quad (8)$$

$$n_z = k_1(\frac{V_A}{180}) \times (\theta_{AT} - \theta_{pitch}) + \cos(\theta_{pitch}) \quad (9)$$

where $k_1 = 0.06$, $k_2 = 0.08$, θ_{AT} and ψ_{AT} are the intersection angles between the intercept along z-axis or y-axis and relative distance, respectively, θ_{pitch} is the tracking pitch angle of the attacker, ψ_{yaw} is the tracking yaw angle. The bank angle γ and normal load factor n_f are calculated by:

$$\gamma = \tan^{-1}(\frac{n_y}{n_z}) \quad (10)$$

$$n_f = n_y \sin\gamma + n_z \cos\gamma \quad (11)$$

E. GUIDANCE LAW

The guidance law consists of the tactical situation and recommended maneuver. Six recommended offensive maneuvers and five recommended defensive maneuvers are presented in Table I and Table II, λ represents deviation angle, H is the altitude, V is the velocity, θ is the pitch angle, δH is the different height of two UAVs, D is the distance between two UAVs.

High-yo-yo, low-yo-yo, barrel-roll, vertical-turn-down, vertical-turn-up, and point-maneuver are offensive maneuvers adopted in advantageous tactical situations. Break-hover, boom-zoom, barrel-roll, zoom-reverse, and break-turn are defensive maneuvers performed to escape the attack from the target.

TABLE I **Guidance law of offensive engagement**

Precondition		Recommended
$\lambda_A < 60\deg$ $\lambda_T > 60\deg$	$abs(\delta H) < 100m$ $abs(\theta) < 10\deg$	*high-yo-yo*
$\lambda_A < 60\deg$ $60\deg < \lambda_T < 120\deg$ $abs(\theta) < 10\deg$	$abs(\delta H) < 100m$ $D > 1000m$	*low-yo-yo*
$\lambda_A < 30\deg$ $\lambda_T > 30\deg$ $abs(\theta) < 10\deg$	$abs(\delta H) < 100m$ $D \le 500m$	*barrel-roll*
$\lambda_A > 120\deg$ $1000 < \delta H < 2500m$	$V \le 250m/s$ $D > 1200m$	*vertical-turn-down*
$\lambda_A > 120\deg$ $-2500 < \delta H < -1000m$ $\theta > -30\deg$	$V \ge 200m/s$ $D > 1200m$	*vertical-turn-up*
WOZ		*point-maneuver*

TABLE II **Guidance law of defensive engagement**

Precondition		Recommended
$\lambda_A > 150\deg$ $\lambda_T > 150\deg$ $1000m < D < 2000m$	$H < 4000m$ $abs(\theta) < 10\deg$	*break-hover*
$\lambda_A > 150\deg$ $\lambda_T > 150\deg$ $abs(\theta) < 10\deg$	$H > 4000m$ $1500m < D < 2000m$	*boom-zoom*
$\lambda_A > 150\deg$ $\lambda_T > 150\deg$ $abs(\theta) < 10\deg$	$H > 5000m$ $D \le 800m$	*barrel-roll*
$\lambda_A > 150\deg$ $\lambda_T > 150\deg$ $abs(\theta) < 10\deg$	$V \le 250m/s$ $2000m < H < 5000m$	*zoom-reverse*
$\lambda_A > 150\deg$ $\lambda_T > 150\deg$ $abs(\theta) < 10\deg$	$H > 4000m$ $D \le 1500m$	*break-turn*

F. ARCHITECTURE OF AUTONOMOUS MANEUVER DECISION SYSTEM

The system models a dogfight engagement as a series of discrete decisions. The UAV chooses the optimal maneuver to follow at each decision-making point until the next decision-making time. According to the guidance law, the UAV can perform the recommended maneuver under a specific tactical situation. The block diagram of the autonomous maneuver decision system is shown in Fig.3. The system collects information about its state and the predicted state of its opponent. Thus, the situation assessment is formulated, and with the game mixed strategy, the maneuver decision objective function can be designed. The improved pigeon-inspired optimization algorithm is used to optimize the objective function to obtain the optimal maneuver.

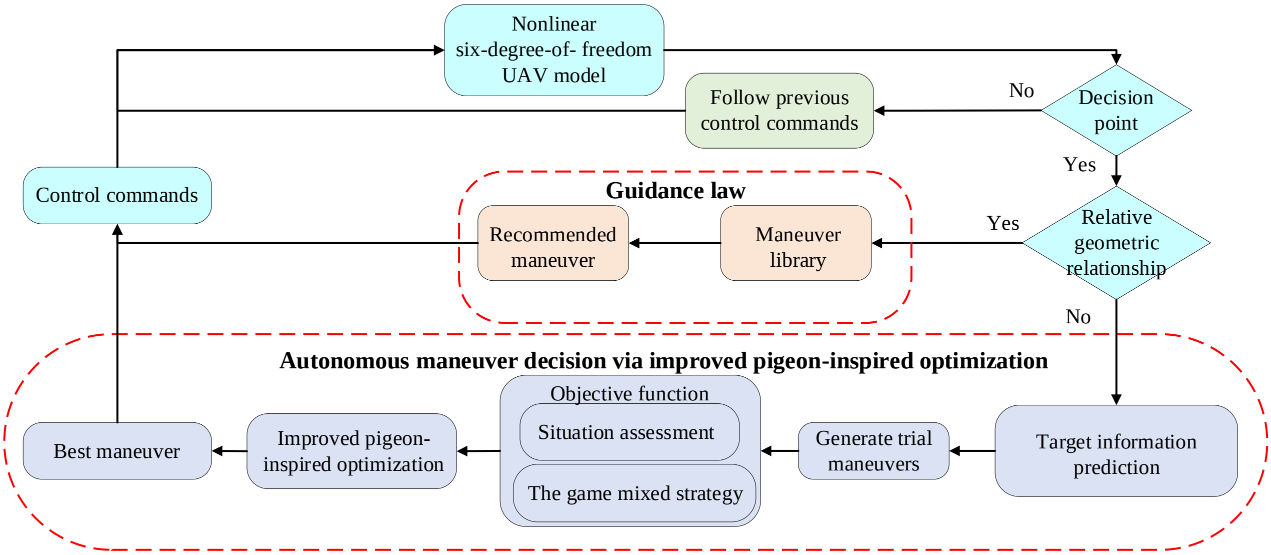

Nonlinear six-degree-of- freedom UAV model
Follow previous control commands
No
Decision point
Yes
Control commands
Guidance law
Recommended maneuver
Maneuver library
Yes
Relative geometric relationship
No
Autonomous maneuver decision via improved pigeon-inspired optimization
Objective function
Situation assessment
The game mixed strategy
Best maneuver
Improved pigeon-inspired optimization
Generate trial maneuvers
Target information prediction

Fig. 4. Architecture of autonomous maneuver decision system.

III. AUTONOMOUS MANEUVER DECISION VIA IMPROVED PIGEON-INSPIRED OPTIMIZATION

A. TARGET INFORMATION PREDICTION

To begin the decision cycle, the system obtains certain information about the attacker and the target UAVs. However, information about the target UAV is less available during combat. Thus, a quadratic curve fit is made for the flight trajectory based on the most recent position of the target. It is assumed that the target UAV maintains the trial maneuver during the predicted time. To simplify this process, the target is considered a three-degree of freedom UAV model, and its motion can be described as a second-order polynomial equation concerning time.

The parameter t is the decision moment, δt the time of prediction. The prediction position of the target is $[X_p(t+\delta t), Y_p(t+\delta t), h_p(t+\delta t)]$, $V_p(t+\delta t)$ indicates the prediction of the velocity, $\theta_p(t+\delta t)$ and $\psi_p(t+\delta t)$ are the prediction of flight path angles. The estimated target position is calculated as follows:

$$\begin{bmatrix} X_p(t) \\ Y_p(t) \\ h_p(t) \end{bmatrix} = C_{3\times3}\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} c_1^x & c_2^x & c_3^x \\ c_1^y & c_2^y & c_3^y \\ c_1^h & c_2^h & c_3^h \end{bmatrix}\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix} \tag{12}$$

where $C_{3\times3}$ is the parameter matrix and can be obtained based on the position information using the polynomial curve fitting method. For example, c_1^x, c_2^x and c_3^x can be calculated as:

$$\begin{cases} c_1^x = \dfrac{1}{2\delta t^2}(X(t) - 2X(t-\delta t) + X(t-2\delta t)) \\ c_2^x = \dfrac{1}{2\delta t^2}(3X(t) - 4X(t-\delta t) + X(t-2\delta t)) \\ c_3^x = X(t) \end{cases} \tag{13}$$

Based on these parameters, the target's position, velocity, and attitude angles can be calculated by:

$$\begin{cases} X_p(t+\delta t) = c_1^x \delta t^2 + c_2^x \delta t + c_3^x \\ Y_p(t+\delta t) = c_1^y \delta t^2 + c_2^y \delta t + c_3^y \\ h_p(t+\delta t) = c_1^h \delta t^2 + c_2^h \delta t + c_3^h \\ V_{px}(t+\delta t) = \dfrac{X_p(t+\delta t) - X_p(t)}{\delta t} \\ V_{py}(t+\delta t) = \dfrac{Y_p(t+\delta t) - Y_p(t)}{\delta t} \\ V_{ph}(t+\delta t) = \dfrac{h_p(t+\delta t) - h_p(t)}{\delta t} \\ V_p(t+\delta t) = \sqrt{V_{px}^2 + V_{py}^2 + V_{ph}^2} \\ \theta_p(t+\delta t) = \sin^{-1}(\dfrac{V_{ph}}{V_p}) \\ \psi_p(t+\delta t) = \tan^{-1}(\dfrac{V_{py}}{V_{px}}) \end{cases} \tag{14}$$

B. TRIAL MANEUVERS

To select an appropriate maneuver, a set of trial maneuvers different from the maneuver library of the guidance law are designed based on the pure-pursuit maneuver and flight agility of the UAV. The pure-pursuit maneuver is a tracing maneuver used mainly in a neutral combat engagement. It uses the bank angle command γ_c to keep the target in the attacker's flight plane of the velocity system. The position vectors of the attacker and the target are defined as $[X_A, Y_A, h_A]$ and $[X_T, Y_T, h_T]$, respectively. The relative position vector from the target's position to the attacker's position in the ground axis system is:

$$AT_i = [X_T - X_A, Y_T - Y_A, h_T - h_A] \tag{15}$$

The projection in the flight path frame is:

$$AT_k = T_{ik} AT_i = [x_k, y_k, z_k] \tag{16}$$

where T_{ik} is the transform matrix. The bank angle command γ_c can be generated by:

$$\begin{cases} y_k \cos\gamma_c + z_k \sin\gamma_c = 0 \\ \gamma_c = \tan^{-1}(-\dfrac{y_k}{z_k}) \end{cases} \tag{17}$$

The current load factor is adopted as an assumption in the tracking pursuit maneuver. The rolling angle around the speed axis must be corrected based on the lateral acceleration to make the resultant external force on the flight in the longitudinal plane. The normal load factor is:

$$n_f = \sqrt{(\cos\theta_A \sin\gamma_c)^2 + (\cos\gamma_c \cos\theta_A + \frac{V_A^2}{g\|AT_i\|_2})^2} \tag{18}$$

The corrected bank angle command is:

$$\sigma = \sin^{-1}(\frac{\cos\theta_A \sin\gamma_c}{n_f}) \tag{19}$$

Finally, the bank angle command is generated as follows:

$$\gamma_c' = \gamma_c - \sigma \tag{20}$$

The maneuver library based on pure-pursuit maneuver is designed as Table III, consisting of 21 trial tactical maneuvers:

TABLE III **Trial maneuver library**

No	Normal Load Factor	Roll Angle	No	Normal Load Factor	Roll Angle
1	$1+\Delta n_z$	0	12	$1+\Delta n_z$ -0.3	ϕ
2	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	0	13	$1+\Delta n_z$ $+0.8\Delta n_{c1}$	90°
3	1	0	14	$1+\Delta n_z$ $+0.8\Delta n_{c2}$	90°
4	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	ϕ	15	$1+\Delta n_z$ $+0.8\Delta n_{c1}$	-90°
5	$1+\Delta n_z$ $-0.8\Delta n_{c2}$	ϕ	16	$1+\Delta n_z$ $+0.8\Delta n_{c2}$	-90°
6	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	$\phi+10^\circ$	17	$1+\Delta n_z$ $+0.8\Delta n_{c1}$	γ_c+180°
7	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	$\phi-10^\circ$	18	$1+\Delta n_z$ $+0.8\Delta n_{c2}$	γ_c+180°
8	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	γ_c	19	$1+\Delta n_z$ $+0.8\Delta n_{c1}$	γ_c-180°
9	1	γ_c	20	$1+\Delta n_z$ $+0.8\Delta n_{c2}$	γ_c-180°
10	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	γ_c+90°	21	$1+\Delta n_z$ $+\dot{n}_z$	0
11	$1+\Delta n_z$ $-0.8\Delta n_{c1}$	γ_c-90°			

where Δn_{c1} and Δn_{c2} are the mean changing rate of normal load factor.

C. OBJECTIVE FUNCTION

1) SITUATION ASSESSMENT

Evaluating the tactical situation between the UAV and its target in instantaneous time and constructing the air combat advantage function can make the decision-making system choose the appropriate maneuver. The advantage function includes orientation scoring function, range scoring function, and energy scoring function. The comprehensive air combat situation assessment value is calculated by weighting these components.

According to Fig. 1, a smaller value of λ_A implies that the attacker is aimed more accurately at the target. And a smaller value of λ_T indicates a lower probability of being shot by the target. The attacker's orientation is aligned with the target when the deviation angles are zero. The orientation scoring function can be calculated by the following equation:

$$S_A = \frac{1}{2}(1+\cos(\frac{1}{2}(\lambda_A+\lambda_T))) \tag{21}$$

The range of missile attacks is relative to the distance between two UAVs. The range scoring function considers the effect of the orientation function and relative distance. The parameter R_0 is defined as the effective missile attack range, and the parameter d indicates the relative distance. The expression of the range scoring function is:

$$S_R = 0.5+(S_A-0.5)\times\max(0,\frac{10R_0-d}{9R}) \tag{22}$$

The energy proportion of the attacker and the target energy is calculated by:

$$E_A = h_A+\frac{V_A^2}{g},\;\; E_T = h_T+\frac{V_T^2}{g}, k=\frac{E_A}{E_T} \tag{23}$$

The energy scoring function is evaluated according to the value of parameter *k*:

$$S_e = \begin{cases} 1 & k>2 \\ 0.5 & 0.5\le k\le 2 \\ 0 & k<0.5 \end{cases} \tag{24}$$

The final scoring function is the sum of the orientation, range, and energy scoring functions:

$$S = S_A+S_R+S_e \tag{25}$$

2) OBJECTIVE FUNCTION BASED ON THE GAME MIXED STRATEGY

The game mixed strategy is introduced to construct the objective function suitable for the dynamic combat environment. Define mixed weight vector *P* containing 21 pure strategies. Define mixed weight vector *P* containing 21 pure strategies. *P* is described as a tuple, $P=(p_1,p_2,\cdots,p_{21})^T$ of probabilities that add to 1. Thus, the objective function is obtained by using the score function vector *S* and probability vector *P* as follows:

$$J = P\cdot S = p_1s_1+p_2s_2+\cdots+p_{21}s_{21} \tag{26}$$

The improved pigeon-inspired optimization optimizes the objective function, and define the optimal solution as P^*, the pure strategy with the highest probability in the mixed strategy will be selected as the optimal tactical maneuver.

D. CLPPPIO ALGORITHM

1) BASIC PIO ALGORITHM

The PIO algorithm consists of the map, compass operator and landmark operator. Assume that there are M pigeons in the *D*-dimensional space. Each pigeon's position is defined as $X_i=[x_{i1},x_{i2},\cdots,x_{iD}]$ and the velocity is defined as $V_i=[v_{i1},v_{i2},\cdots,v_{iD}]$. In the map and compass operator, the pigeon's position and velocity are updated as:

$$V_i(t)=V_i(t-1)\cdot e^{-Rt}+rand\cdot(X_{gbest}-X_i(t-1)) \tag{27}$$

$$X_i(t)=X_i(t-1)+V_i(t) \tag{28}$$

where X_{gbest} represents the optimal global position, and *rand* is a random number uniformly distributed between 0 and 1, R is the map and compass factor, which represents the influence of last iteration velocity on the current velocity.

In the landmark operator, the homing pigeons decrease by half at each iteration. The remaining pigeons update their positions on the basis of the landmarks X_{center} :

$$M(t)=\frac{M(t-1)}{2} \tag{29}$$

$$X_{center}(t)=\frac{\sum X_i(t)\cdot F(X_i(t))}{\sum F(X_i(t))} \tag{30}$$

$$X_i(t)=X_i(t-1)+rand\cdot(X_{center}(t-1)-X_i(t-1)) \tag{31}$$

where M is the current number of pigeons, and *F*() is the cost function of each pigeon.

2) CLPPPIO ALGORITHM

In the basic PIO algorithm, each pigeon in the population updates its position according to the global best position. PIO easily traps into a local optimum when searching the multi-dimensional space. The predator-prey mechanism is adopted to improve the basic PIO algorithm to overcome the drawbacks of trapping into a local optimum solution,. The predator-prey mechanism is a common phenomenon in the natural environment. The predator hunts the prey, and the prey tries its best to escape from the predator. By imitating the survival ability of the predator-prey mechanism, the updated predator is expressed as:

$$J_{predator} = J_{worst} + \rho(1-\frac{t}{t_{\max}}) \tag{32}$$

where $J_{predator}$ represents the solution of the predator, J_{worst} is the worst solution, t is the current iteration, $t_{\max}$ represents the total iteration, and the parameter ρ represents the hunting rate.

Furthermore, to improve the search ability of the optimization algorithm, a level learning strategy is introduced based on the group learning strategy. Pigeons are divided into different levels according to their fitness values. The pigeon in a lower level can improve its exploration range in space by learning from superior pigeons. Thus, the pigeon's position and velocity can be updated as follows:

$$V_{i,j}(t) = V_{i,j}(t-1)\cdot e^{-Rt} + r_1(X_{l1,k1} - X_{i,j}) + r_2(X_{l2,k2} - X_{i,j}) \tag{33}$$

$$X_{i,j}(t) = X_{i,j}(t-1) + V_{i,j}(t) \tag{34}$$

where $X_{i,j}$ is the position of *jth* pigeon in *ith* level, $X_{l1,k1}$ and $X_{l2,k2}$ are pigeons in two different higher levels. The indexes $l1$ *and* $l2$ are within the range $[1, i-1]$ and $l1 < l2$. Parameters $k1$ and $k2$ are randomly selected within the size of each level. The weights $r1$ and $r2$ are random numbers within [0,1].

To further improve the exploitation ability during the landmark operator stage, the Cauchy mutation mechanism is introduced at this stage. This mechanism helps to prevent the too fast convergence speed and can enhance the searching space for a better solution. The Cauchy mutation factor *c* is calculated as follows:

$$c = \tan(\pi(rand - \frac{1}{2})) \tag{35}$$

Thus, the pigeon's position is updated as:

$$X_i(t) = X_i(t-1) + c(X_{center} - X_i(t-1)) \tag{36}$$

The detailed optimization method is outlined in Algorithm1.

3) THEORETICAL AND COMPLEXITY ANALYSIS OF CLPPPIO ALGORITHM

The predator-prey mechanism used in the CLPPPIO algorithm increases the algorithm's ability to jump out of the local optimum. Therefore, the convergence of the predator-prey mechanism based PIO is consistent with that of basic PIO [26], which is as follows.

Definition 1 (Markov Chain): If a sequence of discrete random variables $\{\zeta(k), k \geq 0\}$ satisfy the Markov property:

$$\begin{aligned} &P(\zeta(t+1) = i(t+1) \mid \zeta(t) = i(t), \cdots, \zeta(0) = i(0)) \\ &= P(\zeta(t+1) = i(t+1) \mid \zeta(t) = i(t)) \end{aligned} \tag{37}$$

then, $\{\zeta(k), k \geq 0\}$ is Markov Chain. The possible values of random variables form a set *S* called the state space. If *S* is finite, $\{\zeta(k), k \geq 0\}$ is a finite Markov Chain.

Lemma 1: The population sequence $X(t)$ of CLPPPIO is a finite Markov Chain.

Proof: The population sequence $X(t)$ of CLPPPIO is a collection of discrete random variables. $X(t)$ is bounded and the population size of CLPPPIO is finite and determined, thus, the state space for the pigeon swarm is finite. Moreover, the value of $X(t+1)$ depends only on $X(t)$. Therefore, the population sequence $X(t)$ of CLPPPIO is a finite homogeneous Markov Chain.

Definition 2: The global optimal solution set is described as $\Gamma = \{X_{gbest}, \forall s \in S, s.t. F(X_{gbest}) \geq F(s)\}$.

Definition 3: In CLPPPIO, for any initial positions of pigeons $X(0) = S_0 \in S$, $\lim\limits_{t\to\infty} P(X(t) \in \Gamma \mid X(0) = S_0) = 1$ means that the algorithm strongly converges in probability to the global optimal solution set Γ , while $\lim\limits_{t\to\infty} P(X(t) \cap \Gamma \neq 0 \mid X(0) = S_0) = 1$ indicates a weak convergence to the global optimal solution set.

Lemma 2: The population of CLPPPIO evolves monotonically, $F(X(t+1)) \geq F(X(t))$.

Proof: It is obvious that pigeons always evolve in the direction of bigger fitness. Thus, the optimal fitness of population is monotonically non-decreasing.

Theorem 1: The population sequence $X(t)$ of CLPPPIO converges with probability 1 to the global optimal solution set Γ , $\lim\limits_{t\to\infty} P(X(t) \in \Gamma) = 1$.

Proof: According to Lemma 1 and Lemma 2, it is concluded that if $X(t)$ belongs to the global optimal solution set, $X(t+1)$ also belongs to global optimal solution set:

$$P(X(t+1) \in \Gamma \mid X(t) \in \Gamma) = 1 \tag{38}$$

Then

$$\begin{aligned} &P(X(t+1) \in \Gamma) = P(X(t) \in \Gamma) \cdot P(X(t+1) \in \Gamma \mid X(t) \in \Gamma) \\ &+ (1 - P(X(t) \in \Gamma)) \cdot P(X(t+1) \in \Gamma \mid X(t) \notin \Gamma) \\ &= P(X(t) \in \Gamma) + (1 - P(X(t-1) \in \Gamma)) \cdot P(X(t+1) \in \Gamma \mid X(t) \notin \Gamma) \end{aligned} \tag{39}$$

Suppose

$$P(X(t+1) \in \Gamma \mid X(t) \notin \Gamma) \geq p(t) \geq 0 \tag{40}$$

Then

$$\lim_{t\to\infty} \prod_{i=1}^{t} (1 - p(i)) = 0 \tag{41}$$

Then

$$\begin{aligned}
&1-P(X(t+1)\in\Gamma)=1-P(X(t)\in\Gamma)\\
&+P(X(t)\in\Gamma)\cdot P(X(t+1)\in\Gamma\mid X(t)\notin\Gamma)\\
&-P(X(t+1)\in\Gamma\mid X(t)\notin\Gamma)\\
&=(1-P(X(t)\in\Gamma))(1-P(X(t+1)\in\Gamma\mid X(t)\notin\Gamma))\\
&\le(1-P(X(t)\in\Gamma))(1-p(t))\\
&\Rightarrow 1-P(X(t+1)\in\Gamma)\le(1-P(X(0)\in\Gamma))\prod_{i=1}^{t}(1-p(i))\\
&\Rightarrow \lim_{t\to\infty}P(X(t+1)\in\Gamma)\ge 1-(1-P(X(0)\in\Gamma))\lim_{t\to\infty}\prod_{i=1}^{t}(1-p(i))\\
&\Rightarrow \lim_{t\to\infty}P(X(t+1)\in\Gamma)\ge 1
\end{aligned} \tag{42}$$

However, $P(X(t)\in\Gamma)\le 1$,thus $\lim_{t\to\infty}P(X(t)\in\Gamma)=1$.

The level learning strategy promotes the diversity of the population, which can help the algorithm escape from the local optimum and find a global optimum. Without generality, considering the search space in one dimension, the equation (33) can be rewritten as follows:

$$V_{i,j}(t)=V_{i,j}\cdot e^{-Rt}+\alpha_1(\beta_1-X_{i,j}) \tag{43}$$

$$\alpha_1=r_1+r_2 \tag{44}$$

$$\beta_1=\frac{r_1}{r_1+r_2}X_{l1,k1}+\frac{r_2}{r_1+r_2}X_{l2,k2} \tag{45}$$

Compared with X_{gbest} in basic PIO, $X_{l1,k1}$ provides chances for each pigeon with lower fitness to learn from various pigeons with better fitness. Therefore, the diversity of the exemplars used to guide the learning of pigeons is high, which indicates the great exploration ability of the algorithm.

The time complexity of CLPPPIO algorithm could be obtained by the mathematical expressions. Define the population size of pigeons as M, the dimension of the parameter vectors as D, the objective function J as L_f. The complexity of the map and compass operator in basic PIO can be described as $O(M(D+L_f))$, and the complexity of the landmark operator can be obtained as $O(M\log M+D\log M+L_f\log M)$. Define the total iteration as $Nc\max$, the computation cost of basic PIO is obtained as $O(Nc\max(M\log M+D\log M+L_f M))$. As for the proposed CLPPPIO algorithm, define the computational cost of the predator-prey mechanism as L_{PP}, the number of levels as NL. The computational cost increased by the predator-prey mechanism is $O(M(D+L_f+L_{PP}))$, and the level learning strategy takes extra $O(M\log M+M)$ in each iteration. Therefore, the total computational cost of CLPPPIO algorithm is $O(N_c(M\log M+DM+L_fM+L_{PP}M+M))$.

Algorithm 1: CLPPPIO algorithm

```
Initialize: population size $M$ , position $X$ and velocity $V$, number of levels $NL$, level size $LS$;
For $n=1$; $n<Nc1\max$; $n++$ do
Update the new position and velocity using (24) and (25);
  Select predator according to the worst solution using (29);
  Sort pigeons in ascending order of fitness and divide them into NL levels;
    For $i=\{NL,\cdots,3\}$ do
      For $j=\{1,\cdots,LS\}$ do
        Select two levels from the $(i-1)$ levels: $l_1$, $l_2$;
        Randomly select two pigeons from $l_1,l_2$ : $X_{l1,k1},X_{l2,k2}$;
        Update pigeon $X_{i,j}$ according to (30) and (31);
      End For
    End For
    For $j=\{1,\cdots,LS\}$ do
      Select two pigeons from the first level: $X_{1,k1},X_{1,k2}$;
      Update pigeon $X_{2,j}$ according to (30) and (31);
    End For
End For
For $n=Nc1\max+1$; $n<Nc\max$; $n++$ do
  Update population size $M$ using (26);
  Calculate the landmark center $X_{center}$ using (27);
  Update new position using (33);
End For
Return $X_{gbest}$
```

IV. SIMULATION RESULTS AND ANALYSIS

A. COMPARISION ANALYSIS OF OPTIMIZATION ALGORITHM

To select the algorithm-related parameters, the effect of the parameter variations on the CLPPPIO algorithm is shown in Fig. 5. The parameters include population size M, the map and compass factor R, hunting rate ρ, and the number of levels NL. Their values are tabulated in Table IV. The maximum iteration number of the map and compass operator in CLPPPIO is defined as $Nc1\max$ and the maximum iteration number of the total algorithm $Nc\max$.

TABLE IV **Specification of Different parameters**

	Case 1	Case 2	Case 3	Case 4
M	$50\sim500$	300	300	300
R	0.2	$0.05\sim0.5$	0.2	0.2
ρ	0.01	0.01	$0.005\sim0.04$	0.01
NL	3	3	3	$3\sim6$

From Fig. 5, the performance of the algorithm is affected by all these four parameters to some extent. In Fig.5 (a), the increase in population size enriches the solution space, to improve the performance. However, the increasing population size does not help find a better solution when the population size exceeds a limit. Fig. 5 (b) describes the effect of varying map and compass factor R. The smaller the map and compass factor is, the higher the search accuracy is. But it will increase the risk of falling into the local optima. As shown in Fig.5 (b), the most suitable value is 0.2. In Fig.5 (c)-(d), it can be concluded that the algorithm performance is better only when the parameter value is moderate. From Fig.5 (c), the most suitable value of the hunting rate ρ is 0.01. The algorithm performance is better when the population is divided into three levels.

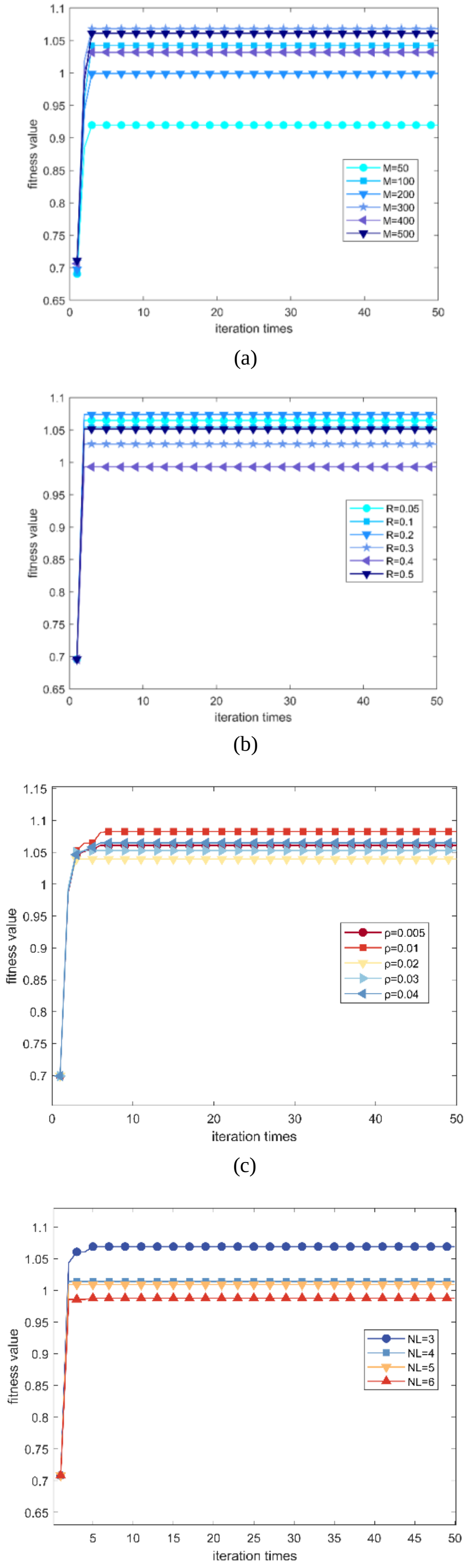
To prove the effectiveness and superiority of the CLPPPIO algorithm, comparative studies are performed between the proposed algorithm and other existing methods, including basic PIO, PSO, predator-prey pigeon-inspired optimization (PPPIO) [25], Cauchy mutation pigeon-inspired optimization (CMPIO) [27], and adaptive learning pigeon-inspired optimization (ALPIO) [28]. The initial parameters settings of these algorithms are shown in Table V, where $c1$ and $c2$ are two acceleration coefficients, and w is termed the inertia weight.

TABLE V **Specification of different parameters**

PSO Parameters			
$Nc\max$	50	M	300
$c1$	1.3	$c2$	1.7
w	0.8		
PIO Parameters			
$Nc1\max$	30	$Nc\max$	50
M	300	R	0.2
PPPIO Parameters			
$Nc1\max$	30	$Nc\max$	50
M	300	R	0.2
ρ	0.01		
CLPPPIO Parameters			
$Nc1\max$	30	$Nc\max$	50
M	300	R	0.2
ρ	0.01	NL	3
ALPIO Parameters			
$Nc1\max$	30	$Nc\max$	50
$c1$	1.3	$c2$	1.7
w	0.8		
CMPIO Parameters			
$Nc1\max$	30	$Nc\max$	50
M	300	R	0.2

TABLE VI **Comparison analysis of different algorithms**

Algorithm	Time Cost	Convergence iterations
PIO	0.027032	3
PSO	0.054766	3
PPPIO	0.068369	5
CLPPPIO	0.041626	2
ALPIO	0.056128	3
CMPIO	0.158004	5



(a)

(b)

(c)

(d)

Fig. 5. Effect of parameters on CLPPPIO algorithm (a) Case 1: Population size. (b) Case 2: The map and compass factor. (c) Case 3: Hunting rate. (d) Case 4: Number of levels.

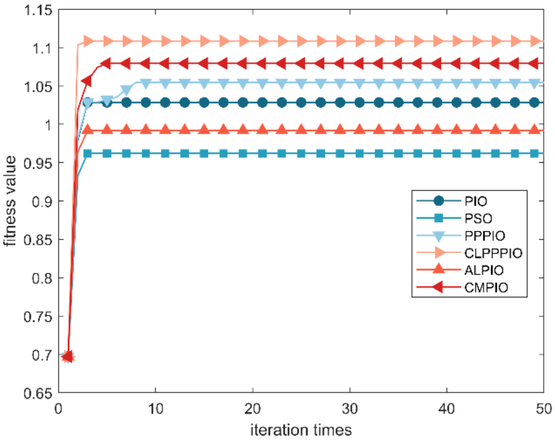
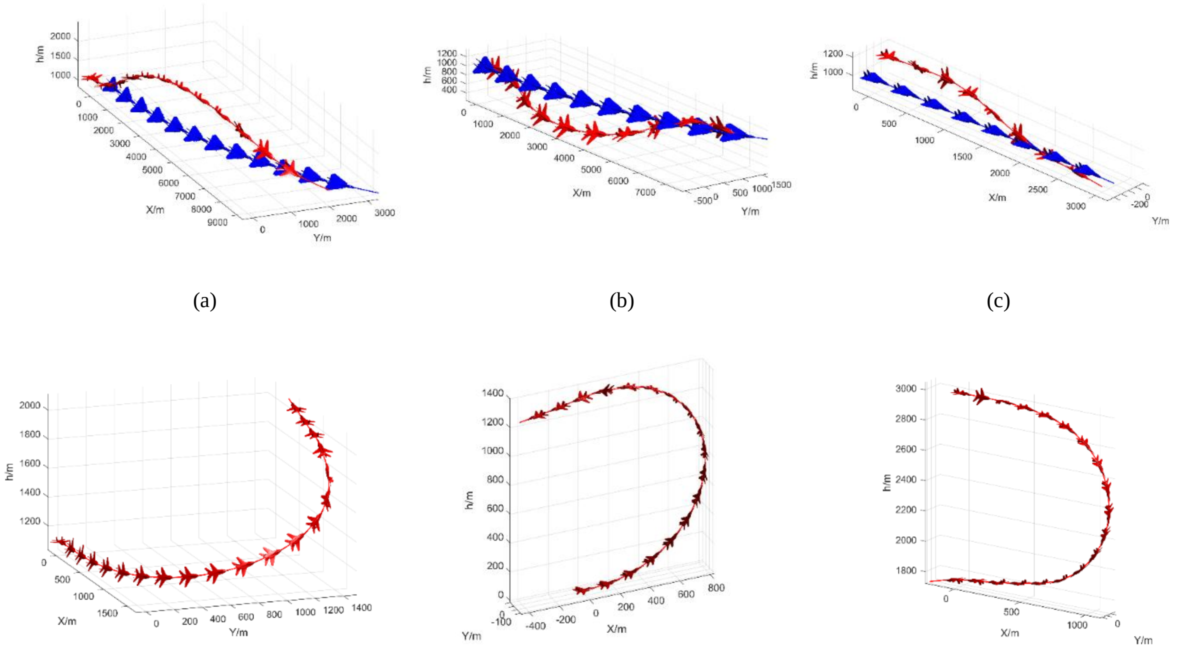


Fig. 6. Comparison evolution curves.

The contrast experimental results are depicted in Fig. 6, and Table 6 shows the time cost of the objective function. From Fig. 6, the evolutionary curve of the CLPPPIO algorithm obtains the maximum, proving the effectiveness and superiority of the proposed algorithm. However, the ALPIO, PPPIO, and CMPIO algorithms do not perform well compared with the CLPPPIO algorithm. The conclusion could be drawn that the superior performance is mainly brought by utilizing the predator-prey mechanism, level-based learning mechanism, and Cauchy mutation mechanism. The predator-prey mechanism increases the diversity of the pigeon and overcomes the problem of local optimum traps. The level-based learning mechanism helps balance global exploration and local exploitation. It could be concluded that the basic PIO and PSO are unsuitable for addressing the multi-dimensional search problem proposed in this paper. Define the convergence iterations as the criterion to evaluate the algorithm performance. Convergence iterations refer to the iterations finding the optimal global solution. For time cost, although the CLPPPIO algorithm does not spend less time than basic PIO, its convergence iterations is improved. And the level of the CLPPPIO algorithm is approximately 10^{-2} s, which is much less than the total aerial combat simulation time.

B. SIMULATION RESULTS OF TACTICAL MANEUVERS

After optimization, the optimal maneuver is selected based on the mixed strategy with maximum probability. The simulation results of tactical maneuvers are shown in Fig.7. The complete process of the high-yo-yo maneuver is presented in Fig. 7 (a). The results of low-yo-yo, barrel-roll, chandelle, zoom-reverse, and split-s maneuvers are shown in Fig. 7 (b)-(f).



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 7. Simulation of tactical maneuvers. (a) High-yo-yo maneuver. (b) Low-yo-yo maneuver. (c) Barrel-roll maneuver. (d) Chandelle maneuver. (e) Split-s maneuver. (f) Zoom-reverse maneuver.

C. AERIAL COMBAT SIMULATION

To verify the effectiveness of the autonomous maneuver decision via CLPPPIO algorithm, four different initial scenes are set for aerial combat simulation. According to the initial position of two UAVs, it can be divided into neutral, offensive, opposite, and defensive scene.

Red and blue UAVs are in dogfight engagement, the red one is the attacker and the blue one is the target. The maximal air speed $V = 340m/s$, the attack radius $R_a = 900m$, the decision-making time is $T_D = 1s$, the trial maneuver time $T_M = 2s$, and the sample time of simulation $T_s = 0.01s$. The CLPPPIO parameters settings are as follows: the population size $M = 300$, the iteration number for map and compass operator $Nc1\max = 30$, the maximum iteration number of the total algorithm $Nc\max = 50$, the map and compass factor $R = 0.2$, the number of levels $NL = 3$.

Four cases are set to conduct the comparative experiments. Case 1 is designed to testify the effect of the advantage scenario in maneuver decision-making. Case 2 utilizes CLPPPIO algorithm and PIO algorithm to verify the effectiveness of the improved algorithm. Case 3 and Case 4 are designed as comparative experiments, min-max search algorithm [12] and stochastic search algorithm are compared with CLPPPIO algorithm. The detailed parameters settings of four initial conditions are listed in Table 7.

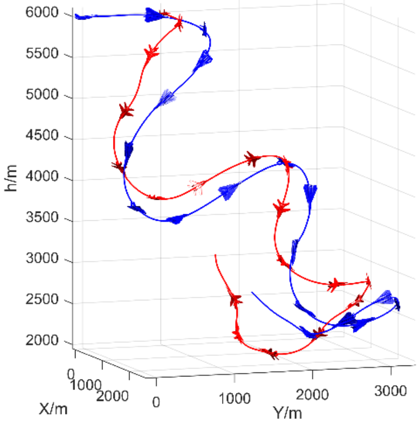
TABLE Ⅶ **Initial conditions**

Scene	UAV	*X/m*	*Y/m*	*H/m*	*V/[m/s]*	ψ / deg
Case 1	Red	0	1000	6000	220	0
	Blue	0	0	6000	220	90
Case 2	Red	0	0	6000	220	0
	Blue	1800	1000	6000	220	0
Case 3	Red	0	0	6000	220	0
	Blue	1800	1000	6000	220	180
Case 4	Red	0	0	3000	220	0
	Blue	0	1000	2000	220	0

CASE 1. Neutral scenario

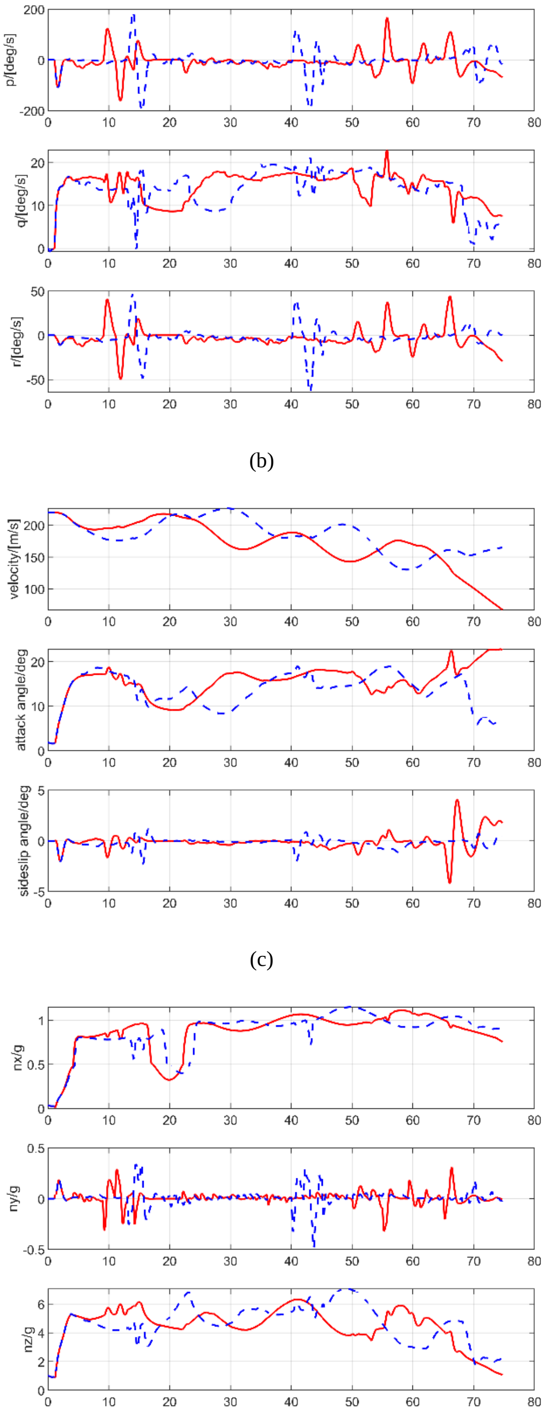
The heading angle of the red and blue UAV are 0° and 90°, respectively. The red and the blue are in orthogonal position, and the initial altitude and flight velocity are the same. The simulation results are shown in Fig. 8. The trajectories of the UAVs are depicted in the Fig 8. (a). At the beginning, the blue and the red select the optimal maneuver through the CLPPPIO algorithm. The blue rolls to pursue the red, and the red turns right to evade the attack from the blue. The pitch angle of the red is too large in the second 10, it increases the normal load factor to pull itself up. In the second 15, the dive angle of the red is too large, it decreases the normal load factor until second 22. The blue is in advantageous situation for attacking the red, therefore, it adopts the point-maneuver based on guidance law to pursue the red until second 33. After 75 s, the blue arrives at the firing position and the simulation is terminated.

The body-axis roll, pitch, and yaw angular rate variation curves are shown in Fig. 8 (b). The value of the roll angular rate is obviously larger than that of pitch and yaw. The velocity, attack angle, and sideslip angle curves are shown in Fig. 8 (c). The decrease of the flight velocity indicates the loss of energy during the aerial combat. The value of the sideslip angle is small and sometimes zero in engagement. Fig. 8 (d) illustrates the variation of load factor. The load factor is the key parameter to determine and hence the maneuver performance. The results indicates that high load factor occurs often, and zero/negative load factors are rare in dogfight engagement. Therefore, the CLPPPIO algorithm is effective for maneuver decision, and the blue wins the combat because of its advantage in orientation initially.



(a)

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2022.3221691



(b)

(c)

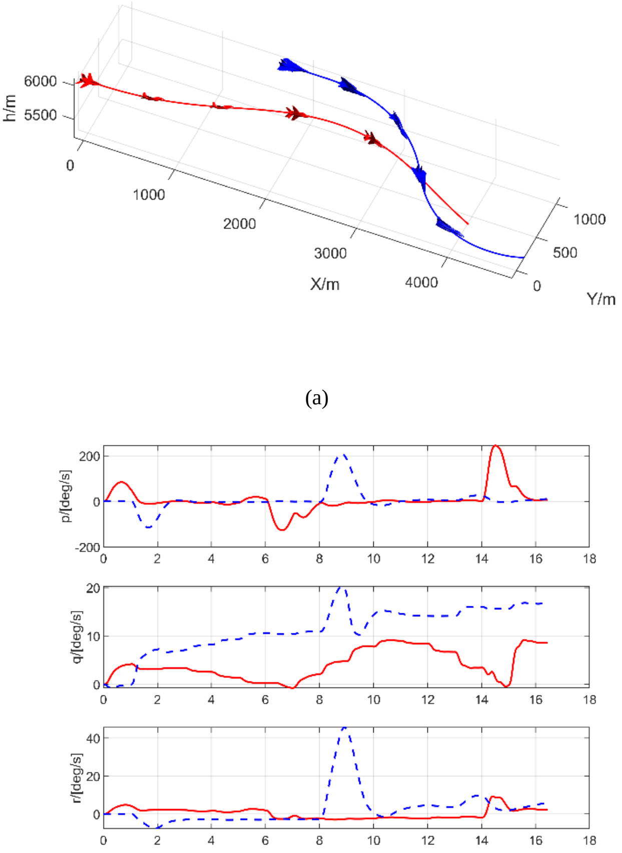
(d)

Fig. 8. Simulation results of the neutral intial position. (a) 3D positions of red and blue UAVs. (b) Body-axis roll, pitch, and yaw angular rates. (c) Velocity, attack angle, and sideslip angle curves. (d) Load factor in x, y, z-axis direction

CASE 2. Offensive scenario

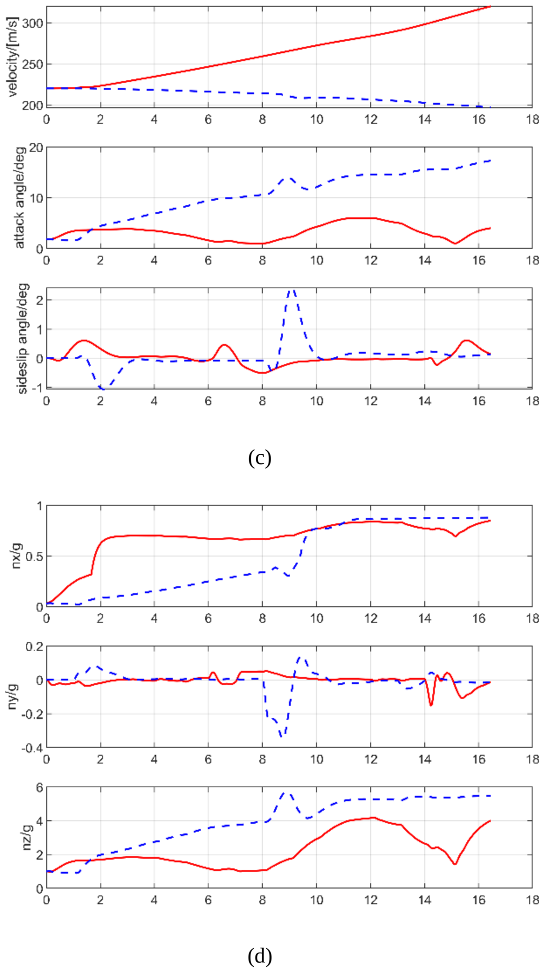
The heading angles of the red and blue UAV are both zero initially. It means that both sides are flying in similar direction, and the red UAV is in offensive position. The initial altitude and velocity of both sides are the same. The red UAV utilizes the CLPPPIO algorithm for decision-making, and the blue utilizes the PIO algorithm. The simulation results are shown in Fig. 9. Fig. 9 (a) presents the flight trajectories. At the initial moment, the red UAV is on the tail of the blue UAV. Benefiting its offensive situation, the red UAV shortens the distance and maintains the tailing pursuit. Then, the blue adopts break turn maneuver with large turning angle to escape the attack from the red. After 17s, red UAV arrives at the firing position and the engagement is terminated.

Fig. 9 (b) shows the angular rate variation of the two UAVs. The angular rate tends to change in the approximately same trend, because that two UAVs are in dogfight engagement. The velocity, attack angle, and sideslip angle variation curves are depicted in Fig. 9 (c). The increase of the red's velocity indicates the lead pursuit to approach the blue UAV. The sideslip angle of the blue fluctuates throughout with a small fluctuation range. Fig. 9 (d) presents the variation curves of the load factor. It indicates that both sides use large load factor at most of the time, because they perform offensive maneuvers and break turn maneuver with large turning angle. It can be concluded that the CLPPPIO algorithm performs high accuracy making reasonable maneuver in offensive scenario.



(a)

(b)



(c)

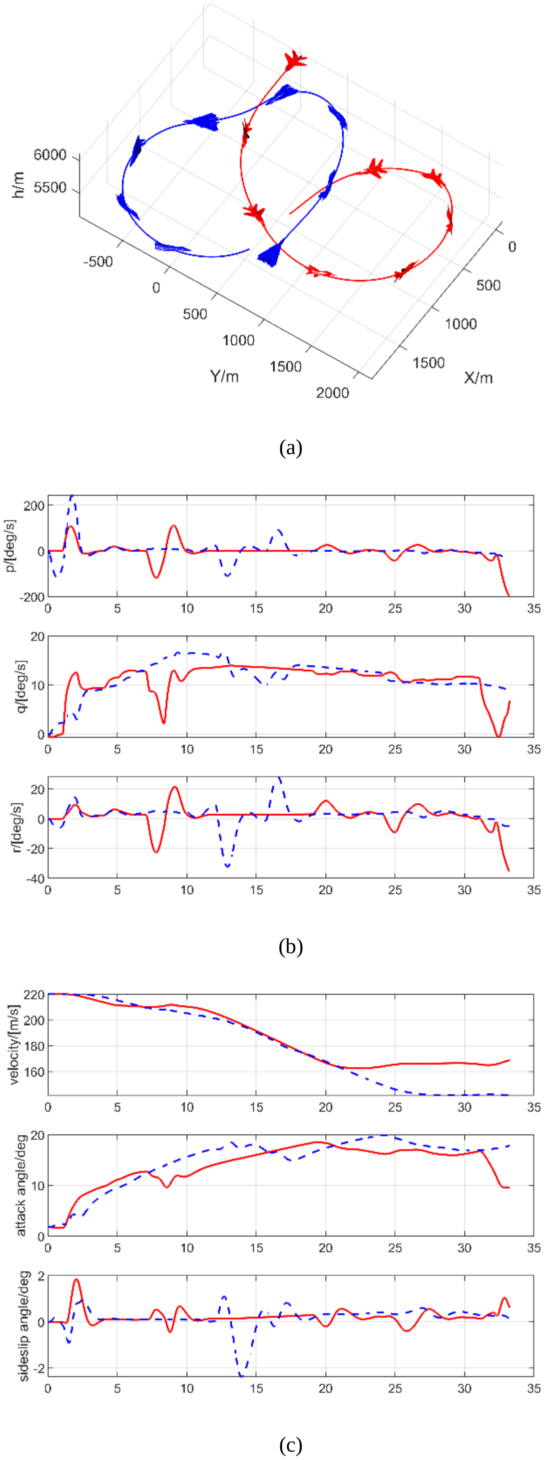
(d)

Fig. 9. Simulation results of the offensive intial position. (a) 3D positions of red and blue UAVs. (b) Body-axis roll, pitch, and yaw angular rates. (c) Velocity, attack angle, and sideslip angle curves. (d) Load factor in x, y, z-axis direction

CASE 3. Opposite scenario

At the initial moment, the heading angle of the red UAV is 0°, and the blue's heading angle is 180°. The red and blue UAVs are in opposite position, and their initial altitude and flight velocity are the same. The red UAV utilizes the CLPPPIO algorithm for decision-making, and the blue uses the min-max search algorithm. The simulation results are represented in Fig. 10. The flight trajectories are depicted in Fig. 10 (a). The two UAVs point at each other at the beginning. The strategy is to turn to the tail of the opponent. In the second 2, the red and blue UAVs perform forward turning. In the second 9, they point at each other again. And the red UAV wins the aerial combat in the second 35.

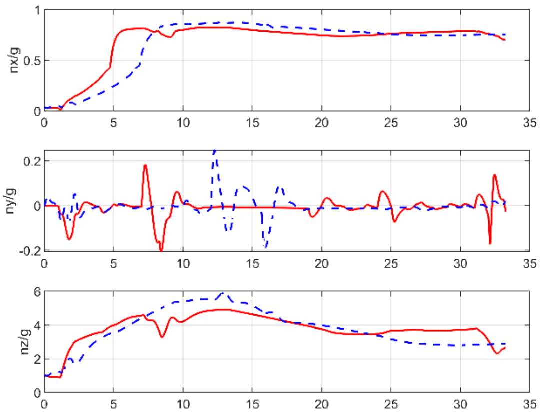
Fig. 10 (b) shows the angular rate variation curves. The values of body-axis roll and yaw angular rate fluctuate around zero. The velocity, attack angle, and sideslip angle variation curves are shown in Fig. 10 (c). The velocity of both sides fluctuates a little, and their trend of increase and decrease is similar. From Fig. 10 (d), the normal load factor determines the turn performance, and the forward turning during the combat process makes the normal load factor greater than 3g at most of the simulation time. This Case study indicates the superior performance of the proposed CLPPPIO algorithm. Compared with the min-max search algorithm, the proposed method is more flexible, and the prediction of the target's information is effective.



(a)

(b)

(c)



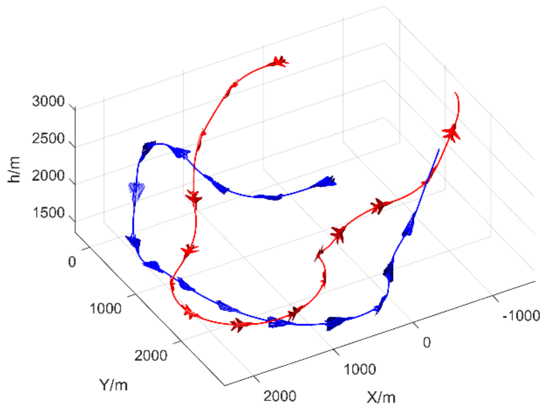
(d)

Fig. 10. Simulation results of the opposite intial position. (a) 3D positions of red and blue UAVs. (b) Body-axis roll, pitch, and yaw angular rates. (c) Velocity, attack angle, and sideslip angle curves. (d) Load factor in x, y, z-axis direction

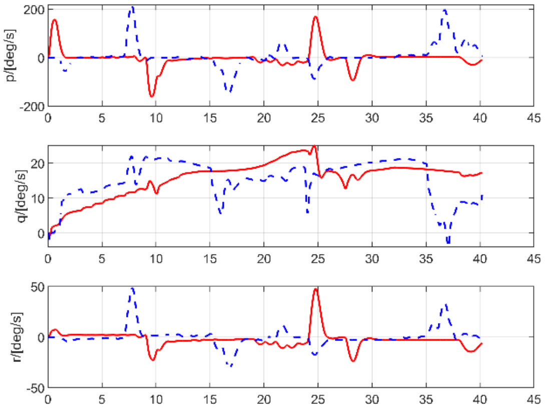
CASE 4. Defensive scenario

At the start, the heading angles of the red and the blue UAV are both 0°. It means that both sides are flying in similar direction, and the red has a position advantage. The blue UAV utilizes the CLPPPIO algorithm, and the red uses the stochastic search algorithm to select the trial maneuver in the engagement. The simulation results are presented in Fig. 11. From Fig. 11 (a), the red UAV decreases its altitude to attack the blue, both sides are aggressive, and the trajectories appear to be a “scissors maneuver”. In the second 10, the red performs a significant angle turn forcing the blue to overshoot. The blue performs a low yo-yo to enter the maneuvering plane of the red in the second 15. In the second 31, the red starts to climb, and the blue climbs and hovers to the tail of the red. Finally, the blue UAV wins the combat in the second 41.

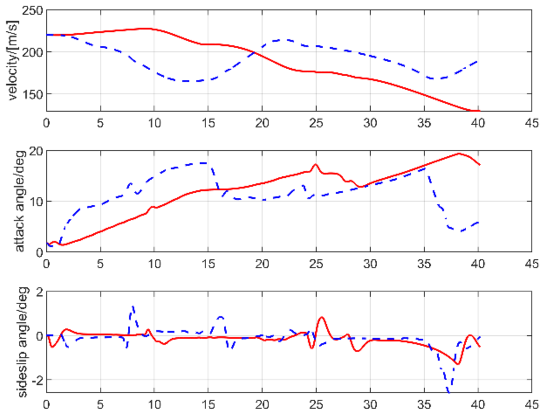
Fig. 11 (b) shows the angular rate variation curves of the two UAVs. The body-axis roll rate is larger than pitch and yaw angular rate. The variation curves of velocity, attack angle, and sideslip angle are shown in Fig. 11 (c). The dogfight is fierce, and the red’s velocity decreases gradually. As can be seen in Fig. 11 (d), the sharp turns sometimes make the normal load factor greater than 6*g*. The combat results explicitly show the proposed maneuver decision method is more effective than the stochastic search algorithm. Although the stochastic search algorithm increases the diversity of selecting the maneuver, the CLPPPIO algorithm has the advantage in making good maneuver and raises the possibility of winning the engagement.



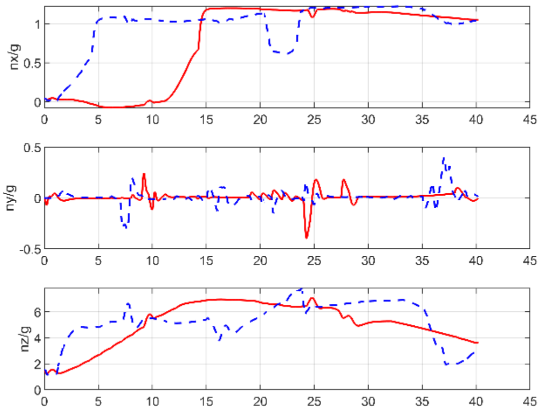
(a)



(b)



(c)



(d)

Fig. 11. Simulation results of the defensive intial position. (a) 3D positions of red and blue UAVs. (b) Body-axis roll, pitch, and yaw angular rates. (c) Velocity, attack angle, and sideslip angle curves. (d) Load factor in x, y, z-axis direction

V. CONCLUSION

This paper proposes an autonomous maneuver decision method based on improved pigeon-inspired optimization for UAV. For the six-degree-of-freedom UAV dogfight engagement, the maneuver decision problem is transformed into an optimization problem. The CLPPPIO algorithm performs great search ability and high accuracy in making reasonable maneuver decision. The simulation results indicate that the proposed method can help UAV win the combat engagement in neutral, offensive, opposite and defensive situations. Moreover, compared with the classical min-max search algorithm and stochastic search algorithm, the CLPPPIO algorithm increases the probability of winning the combat engagement and is more suitable for a dynamic aerial combat environment.

Our future work will focus on designing an outfield experiment platform to verify the autonomous maneuver decision method and develop the algorithm between the unmanned aerial vehicle and the manned aerial vehicle in air-to-air confrontation.

REFERENCES

[1] H. Shin, J. Lee, H. Kim, and D. H. Shim, “An autonomous aerial combat framework for two-on-two engagements based on basic fighter maneuvers,” Aerosp. Sci. Technol., vol. 72, pp. 305-315, Jan. 2018.

[2] P. Li and H. B. Duan, “A potential game approach to multiple UAV cooperative search and surveillance,” Aerosp. Sci. Technol., vol. 68, pp. 403-415, Sep. 2017.

[3] R. Chai, A. Tsourdos, A. Savvaris, et al., “Fast generation of chance-constrained flight trajectory for unmanned vehicles,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 2, pp. 1028-1045, Apr. 2021.

[4] R. Chai, A. Tsourdos, A. Savvaris, et al., “Solving constrained trajectory planning problems using biased particle swarm optimization,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 3, pp. 1685-1701, Jun. 2021.

[5] Yu, B. Zhu, and Z. Y. Zuo, “Three-dimensional optimal guidance with Lyapunov redesign for UAV interception,” Guidance, Navigation and Control, vol. 1, no. 4, pp. 2140004: 1-19, Dec. 2021, doi: 10.1142/S2737480721400045

[6] S. Sun, Y. Yin, X. Wang, and D. Xu, " Robust visual detection and tracking strategies for autonomous aerial refueling of UAVs," IEEE Trans. Instrum. Meas., vol. 68, no. 12, pp. 4640-4652, Dec. 2019.

[7] Y. F. Li, J. P. Shi, W. Jiang, W. G. Zhang, and Y. X. Lyu, “Autonomous maneuver decision-making for a UCAV in short-range aerial combat based on an MS-DDQN algorithm,” Def. Technol., pp. 1-18, Sep. 2021, doi: 10.1016/j.dt.2021.09.014

[8] Y. F. Li, N. M. Qi, and Z. W. Tang, “Linear quadratic differential game strategies with two-pursuit versus single-evader,” Chin, J. Aeronaut., vol. 25, pp. 896-905, Dec. 2012.

[9] H. Park, B. Y. Lee, M. J. Tahk, and D. W. Yoo, “Differential game based air combat maneuver generation using coring function matrix,” Int. J. Aeronaut. Space Sci., vol. 17, no. 2, pp. 204-213, Jun. 2016.

[10] M. L. Wang, L. X. Wang, T. Yue, and H. L. Liu, “Influence of unmanned combat aerial vehicle agility on short-range aerial combat effectiveness,” Aerosp. Sci. Technol., vol. 96, pp. 1-12, Nov. 2020.

[11] R. L. Nelson and Z. Rafal, “Effectiveness of autonomous decision making for unmanned combat aerial vehicles in dogfight engagements,” J. Guid., Control, Dyn., vol. 41, no. 4, pp. 1015-1021, Apr. 2018.

[12] S. Y. Li, M. Chen, Y. H. Wang, and Q. X, Wu, “Air combat decision-making of multiple UCAVs based on constraint strategy games,” Def. Technol., pp. 1-16, Feb. 2021, doi: 10.1016/j.dt.2021.01.005

[13] K. Virtanen, T. Raivio, and R. P. Hamalainen, “Modeling pilot’s sequential maneuvering decision by a multistage influence diagram,” J. Guid., Control, Dyn., vol. 27, no. 4, pp. 665-676, Jul.-Aug. 2004.

[14] J.-M. Eklund, J. Sprinkle, and S.-S Sastry, “Switched and symmetric pursuit-evasion games using online model predictive control with application to autonomous aircraft,” IEEE Trans. Aerosp. Electron. Syst., vol. 20, no. 3, pp. 604-620, May 2012.

[15] C. Q. Huang, K. S. Dong, H. Q. Huang, S. Q. Tang, and Z. R. Zhang, “Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization,” J. Syst. Eng. Electron., vol. 29, no. 1, pp. 86-97, Feb. 2018.

[16] J. S. McGrew, P. J. How, B. Williams, and N. Roy, “Air-combat strategy using approximate dynamic programming,” J. Guid., Control, Dyn., vol. 33, no. 5, pp. 1641-1653, Sep.-Oct. 2010.

[17] H. Zhang and C. Huang, “Maneuver decision-making of deep learning for UCAV through azimuth angles,” IEEE Access, vol. 8, pp. 12976-12987, Jan. 2020.

[18] Q. Yang, J. Zhang, G. Shi, J. Hu, and Y. Wu, "Maneuver decision of UAV in short-range air combat based on deep reinforcement learning," IEEE Access, vol. 8, pp. 363-378, Jan. 2020.

[19] R. E. Smith, B. A. Dike, R. K. Mehra, B. Ravivhandran, and A. El-Fallah, "Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft," Comput. Methods, Appl. Mech. Engrg. Vol. 186, pp. 421-437, Mar. 2000.

[20] N. Ernest, D. Carroll, M. Clark, K. Cohen, G. Lee, "Genetic fuzzy based artificial intelligence for unmanned combat aerial control in simulated air combat missions," J. Def. Manag., vol. 6, no. 1, pp. 1-7, Mar. 2016.

[21] H. B. Duan, P. Li, and Y. X. Yu, "A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory," IEEE CAA J. Autom. Sin., vol. 2, no. 1, pp. 11-18, Jan. 2015.

[22] T. Han, X. F. Wang, Y. J. Liang, and S. Ku, "Study on UCAV robust maneuvering decision in automatic air combat based on target accessible domain," J. Phys. Conf. Ser., pp. 1213: 1-11, 2019.

[23] H. B. Duan and P. X. Qiao, "Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning," Int. J. Intell. Comput. Cybern., vol. 7, no. 1, pp. 24-37, Mar. 2014.

[24] D. F. Zhang, H. B. Duan, "Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning," Neurocomputing, vol. 313, no. 3, pp. 229-246, Jun. 2018.

[25] H. B. Duan, M. Z. Huo, Z. Y. Yang, Y. H. Shi, and Q. N. Luo, "Predator-prey pigeon-inspired optimization for UAV ALS longitudinal parameters tuning," IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 5, pp. 2347-2358, Oct. 2019.

[26] B. Zhang and H. B. Duan, "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 14, no. 1, pp. 97-107, Jan./Feb. 2017.

[27] Z. Y. Yang, H. B. Duan, Y. M. Fan, and Y. M. Deng, "Automatic carrier landing system multilayer parameter design based on Cauchy mutation pigeon-inspired optimization," Aerosp. Sci. Technol., vol. 79, pp. 518-530, Aug. 2018.

[28] X. S. Hai, Z. L. Wang, F. Qiang, Y. Ren, B. Sun, and D. Z. Yang, "A novel adaptive pigeon-inspired optimization algorithm based on evolutionary game theory," Sci. China Inf. Sci., vol. 64, no. 3, pp. 139203:1-139203:2, Apr. 2021.

HAIBIN DUAN (M'07-SM'08) is a Full Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, he received his Ph.D. degree in control theory and engineering from Nanjing University of Aeronautics and Astronautics (NUAA) in 2005. He is the Vice Director of the China's State Key Laboratory of Virtual Reality Technology and Systems, and the Head of the Bio-Inspired Autonomous Flight Systems (BAFS) Research Group, Beihang University, Beijing, China. He received the National Science Fund for Distinguished Young Scholars of China in 2014. He is also enrolled in the Chang Jiang Scholars Program of China, Scientific and Technological Innovation Leading Talent of "Ten Thousand Plan"-National High Level Talents Special Support Plan, and Top-Notch Young Talents Program of China, Program for New Century Excellent Talents in University of China, and Beijing NOVA Program. He has authored or coauthored more than 70 publications. He is the Editor-in-Chief of *Guidance, Navigation and Control*, Associate Editor of the *IEEE Transactions on Cybernetics*, and *IEEE Transactions on Circuits and Systems II: Express Briefs*. His current research interests are bio-inspired intelligence, biological computer vision and multi-UAV swarm autonomous control.

YANGQI LEI received her B.S. degree in Computer Science from Beihang University in 2020. She is currently pursuing the master's degree with the BAFS Research Group, China's State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University. Her current research interests are autonomous flight control and bio-inspired computation.

JIE XIA received her B.Eng. degree in systems engineering and the M. Eng. degree in Weapon system engineering from the School of the Nanjing University of Science and Technology, China, in 1984 and 1987 respectively, and the Ph.D. degree from the School of

Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2003. She is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University. Her current research interests include autonomous flight control, unmanned aerial vehicle cooperative control and mission planning.

YIMIN DENG received the B.S. and Ph.D. degrees from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2011 and 2017, respectively. He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University. Dr. Deng is enrolled in the Young Elite Scientists Sponsorship Program by Chinese Association for Science and Technology and Young Top Talent Support Program by Beihang University. His current research interests include biological computer vision and autonomous flight control.

YUHUI SHI (Fellow, IEEE) is a Chair Professor in the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen, China. Before joining SUSTech, he was with the Department of Electrical and Electronic Engineering at the Xi’an Jiaotong-Liverpool University (XJTLU), Suzhou, China, from January 2008 to August 2017, was with the Electronic Data Systems Corporation (EDS), Indiana, USA, from October 1998 to December 2007, and was with Purdue School of Engineering and Technology, Indiana, USA, from October 1995 to September 1998. He is an IEEE Fellow, the Editor-in-Chief of the International Journal of Swarm Intelligence Research, and an Associate Editor of the IEEE Transactions on Evolutionary Computation. His research interests are swarm intelligence and applications.