The Aeronautical Journal (2026), **130**, pp. 1497–1517
doi:10.1017/aer.2026.10127

RESEARCH ARTICLE

UAV tracking one moving target in an oilfield environment with dynamic obstacles based on an improved salp swarm algorithm

Xuzhao Chai, Yiming Lu, Zhongyun Liu, Hui Yang, Ping Liu and Li Yan

Zhongyuan University of Technology, China
Corresponding author: Li Yan; Email: 17788125878@163.com

Received: 25 April 2025; **Revised:** 15 December 2025; **Accepted:** 23 December 2025

Keywords: distributed model predictive control; dynamic target tracking; oilfield inspection; salp swarm algorithm

Abstract

Unmanned aerial vehicles (UAVs) tracking one moving target in an oilfield environment with dynamic obstacles has been studied. Firstly, the UAV tracking one target is modeled mathematically, including the oilfield environment, the UAV kinematic model, the moving target model and so on. Considering the oilfield environment with dynamic obstacles, the chance-constraint method is incorporated into the distributed model predictive control (DMPC) to realise the target tracking and avoid the obstacles in the environment. Secondly, to obtain high-quality control-input sequences, the improved salp swarm algorithm (ISSA) is designed as the solver for the chance-constraint DMPC, integrating the sine-tent-cosine chaotic mapping in the initialisation, the differential mutation operation in the leader-position updating and the nonlinear adaptive inertia weights in the follower-position updating. Finally, the simulation results show that the proposed method has great performance in tracking the moving target in an oilfield environment with dynamic obstacles.

Nomenclature

UAVs	unmanned aerial vehicles
DMPC	distributed model predictive control
CCDMPC	chance-constrained DMPC
ISSA	improved salp swarm algorithm
FOV	field of view
LOS	line of sight
SI	swarm intelligence
PSO	particle swarm optimisation
GWO	grey wolf optimisation
WOA	whale optimisation algorithm
HHO	Harris hawk optimisation
DE	differential evolution
HJPSO	hybrid of jellyfish and PSO
CGWO	competitive leader GWO
AOA-HHO	arithmetic optimisation algorithm (AOA) with the HHO
W	population size
$T_{\max}$	maximum iteration
g	ratio factor
K	weight control factor
N	prediction horizon length

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

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

Greek symbol

φ	heading angle
θ	pitching angle
λ	weight coefficient

1.0 Introduction

Oilfield inspection is an important task for guaranteeing oil-producing equipment in operation, including fixed-target inspection and dynamic-target inspection. The fixed-target inspection monitors the operation status regularly, collecting parameters from the facilities while running. Dynamic-target inspection tracks the unknown objects invading the oilfield area. The traditional inspection methods are mainly used for manual inspecting and fixed-video monitoring. Manual inspecting is labor-intensive and inefficient due to the dispersion of the oil wells, and fixed-video monitoring offers poor flexibility, high cost and blindness in some zones. To overcome these shortcomings, unmanned aerial vehicles (UAVs) have been developed in recent years for target inspection in the oilfield area due to their low cost, high flexibility and low-security risk [1–3]. Shukla et al. have used the UAV inspection in oil or gas pipelines with an autonomous tracking and navigation controller [4]; Xu et al. have equipped the UAVs with an advanced embedded processor and binocular vision sensor for inspecting automatic transmission lines [5]; and Kochetkova et al. have used UAVs to identify pipeline leaks for reducing time and labor resources [6]. These reports mainly focus on inspecting the static targets in the environments.

UAVs are also used to track the dynamic targets. In fact, the real environment is rich with some obstacles, leading in the limitation of the field of view (FOV) and the blocking of the line of sight (LOS). Tracking dynamic targets is a challenge compared with tracking static targets. In order to track the dynamic targets exactly, the model predictive control (MPC) is widely applied to predict the states for a finite time in the future [7]. After that, a solver is selected to obtain optimal control-input sequences. Generally, selecting a solver plays a key role in tracking targets accurately. For example, mathematical optimisation methods, such as dynamic programming, are generally selected as the solvers for obtaining control-input sequences, but they have a combinatorial explosion with the dimension increasing [8]. Swarm intelligence (SI) algorithms, inspired by the behaviour of groups in nature, have also been applied as the solvers due to their strong search capability and parallel computing, including particle swarm Optimisation (PSO) algorithm [9], differential evolutionary (DE) algorithm [10], grey wolf optimisation (GWO) algorithm [11], whale optimisation algorithm (WOA) [12] Harris hawk optimisation (HHO) algorithm [13], et al. But they have inherent limitations. For example, GWO suffers from narrow exploration and limited diversity; PSO is sensitive to initialisation and step sizes, impairing global search; WOA is inclined to be trapped into premature convergence; and HHO relies heavily on fixed models, slowing convergence in high-dimensional spaces. Recently, some SI modified versions have been applied to address specific problems. Reference [14] has proposed the novel predictive model with the hybrid of jellyfish and PSO (HJPSO) algorithms, aiming to efficiently manage massive datasets. Reference [15] has proposed the competitive leader GWO (CGWO) to train multi-layer perceptron classifiers for obtaining optimal network parameters. Reference [16] has combined the arithmetic optimisation algorithm (AOA) with the HHO algorithm (AOA-HHO) to search the optimal solutions in the multilevel thresholding image segmentation problem. In addition, Yu et al. have combined the adaptive DE algorithm with Nash optimisation to solve multi-UAV target tracking [17]; in Ref. [18], the GWO has been improved with introducing the individual memory and the survival of the fittest strategy to optimise the control input sequences in the UAV tracking; Ref. [19] has reported that the modified grasshopper algorithm has been proposed to address the tracking targets, integrating the dynamic feedback, the natural selection and the democratic decision-making mechanism. In a word, SI and its modified versions are selected as the solvers for obtaining high-quality target tracking. But these reports focus on the flight environments with the static obstacles.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

The UAV tracking target has also been studied in an environment with dynamic obstacles. Yao et al. have proposed a hybrid approach based on the Lyapunov guidance vector field to solve the target tracking and obstacle avoiding in a three-dimensional dynamic environment [20]. In Ref. [21], the fusion algorithm has been proposed to track a moving target in a complex three-dimensional environment. Reference [22] has proposed the Gaussian process regression integrated into the MPC framework for avoiding the moving obstacles. The chance-constrained method has been proposed to solve dynamic collision avoidance in target tracking, considering the relative position of UAVs and dynamic obstacles [23–26]. In Ref. [27], the chance-constrained MPC method has also been used to avoid colliding with moving obstacles, considering the relative position and velocity. However, most of the reports have assumed that the UAVs are fixed at a flight altitude for simplicity. This assumption is not reasonable for UAVs to track targets in real-world environments.

In this work, the environment with the static and dynamic obstacles has been considered in tracking a moving target, as well as UAV height-change. The improved salp swarm algorithm (SSA) has been proposed as the solver. The contributions of this work are listed as follows:

(1) The oilfield environment has been modeled, considering the LOS occlusion, the UAV sensor coverage area and the static and dynamic obstacles.

(2) The chance-constrained distributed model predictive control (CCDMPC) method has been proposed to solve the target tracking and obstacle avoidance in three-dimensional dynamic environments, considering the relative position and velocity of the dynamic obstacles and the UAVs.

(3) The improved SSA has been proposed as the solver to obtain better control-input sequences for improving the target tracking performance compared to other comparative algorithms.

The rest of the paper is organised as follows: Section 2 models the target tracking problem in an oilfield environment. Section 3 introduces the CCDMPC method for UAVs' target tracking in an oilfield environment. In Section 4, the ISSA method is proposed to solve the CCDMPC formulation. Section 5 gives simulation experiments to verify the proposed method. The conclusion is given in Section 6.

2.0 Mathematical formulation

2.1 Oilfield environment modeling

The oilfield dynamic environment is modeled in Fig. 1. The UAVs need to track the moving target, avoiding the static and dynamic obstacles in the oilfield environment. In Fig. 1, the static obstacles contain oil tanks, water tanks, pumping wells and so on. The dynamic obstacles, such as flying birds, are assumed to move at a constant speed v_d in a certain direction. The position at k is defined as $p_d(k) = [x_d(k), y_d(k), z_d(k)]^T$. The dynamic obstacles are modeled kinematically in Equation (1). ΔT is the sampling time, and I_3 is the third-order identity matrix.

$$\begin{bmatrix} p_d(k+1) \\ v_d(k+1) \end{bmatrix} = \begin{bmatrix} I_3 & \Delta T I_3 \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} p_d(k) \\ v_d(k) \end{bmatrix} \tag{1}$$

2.2 UAV model

In this work, the N_u quad-rotor UAVs track the target moving on the ground corporately. The wind disturbance is not considered here. The UAVs must avoid all the obstacles in the oilfield environment in order to track safely. Figure 2 shows the geometry of the target tracking. The UAV$_i$ position vector is $p_i = [x_i, y_i, z_i]^T$ $(i = 1, \ldots, N_u)$, and the target position vector is $p_t = [x_t, y_t, z_t]^T$. Here z_t is zero due to the target moving on the ground.

The relative position vector between UAV$_i$ and the target is described in Equation (2).

$$p_{i,t} = p_i - p_t = [x_i - x_t, y_i - y_t, z_i - z_t]^T \tag{2}$$

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

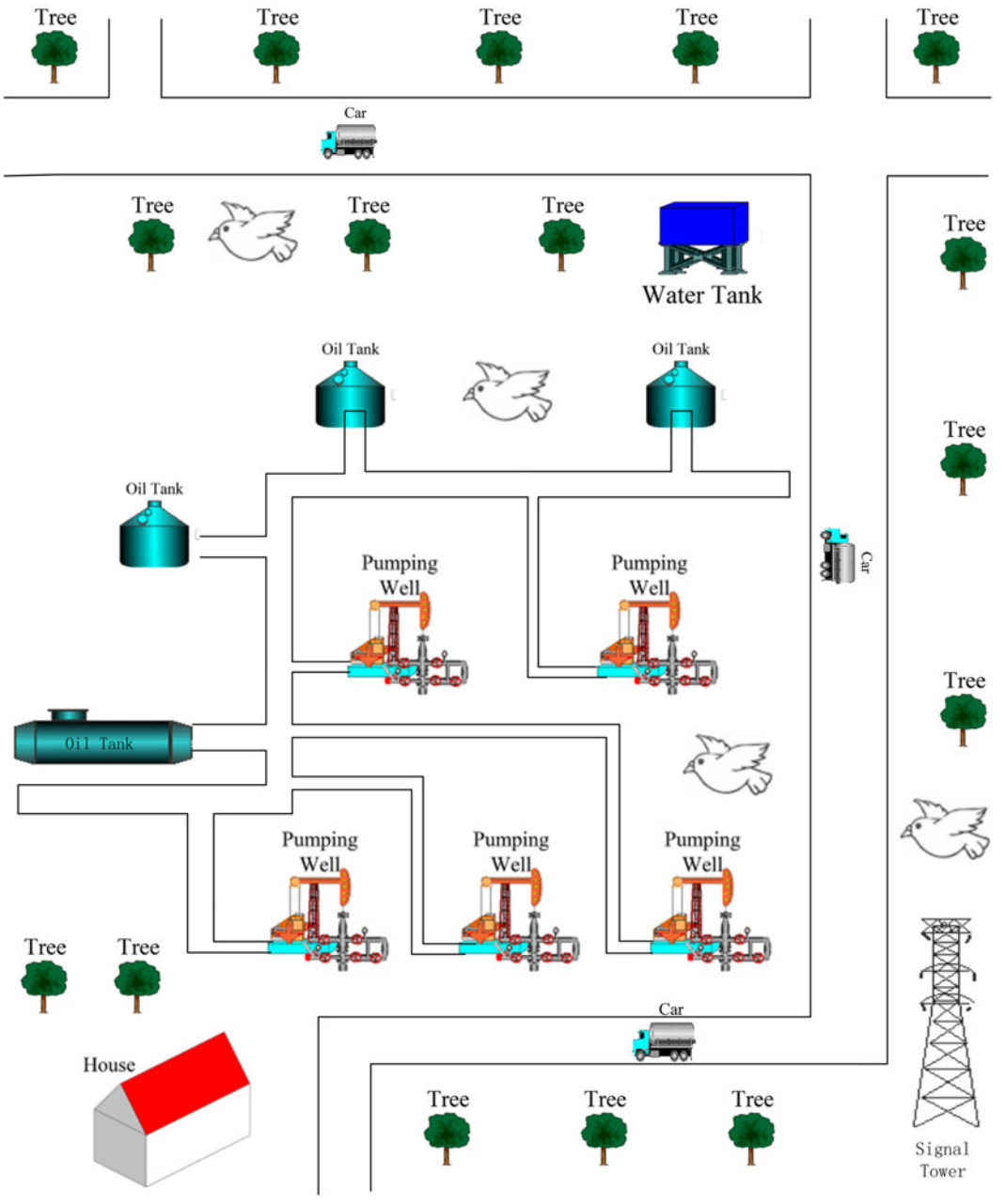


Figure 1. *The oilfield environment model.*

The distance between the UAV$_i$ and the target is expressed in Equation (3), and its projection on the *X-Y* plane is written in Equation (4).

$$d_{i,t} = |p_{i,t}| = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2 + (z_i - z_t)^2} \tag{3}$$

$$r_{i,t} = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2} \tag{4}$$

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

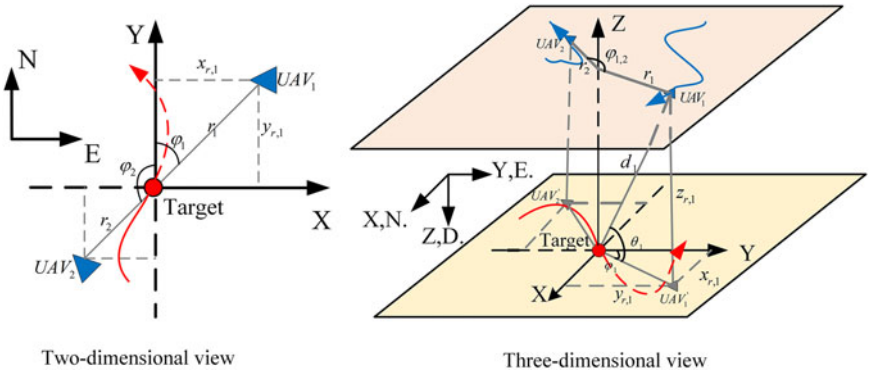


Figure 2. *The geometry of the target tracking.*

Assuming that the quad-rotor UAV is a mass point in the tracking process, the UAV dynamic model is shown in Equation (5) [28] :

$$\begin{cases} x_i = V_i \cos\theta_i \cos\varphi_i \\ y_i = V_i \cos\theta_i \sin\varphi_i \\ z_i = V_i \sin\theta_i \\ \varphi_i = u_i^{\varphi} \\ \theta_i = u_i^{\theta} \\ V_i = u_i^{V} \end{cases} \tag{5}$$

The heading angle φ_i and pitching angle θ_i for the UAV_i are written in Equation (6).

$$\begin{cases} \sin\varphi_i = \frac{x_{r,i}}{r_i} \\ \cos\varphi_i = \frac{y_{r,i}}{r_i} \\ \sin\theta_i = \frac{z_{r,i}}{d_i} \\ \cos\theta_i = \frac{r_i}{d_i} \end{cases} \tag{6}$$

where V_i is the speed for the UAV_i. The minimal speed $V_{\min}$ is 0. u_i^{φ}, u_i^{θ}, and u_i^{V} are the control inputs for φ_i, θ_i and V_i. The state vector of the UAV_i is denoted as $\mathbf{s}_i = [x_i, y_i, z_i, \varphi_i, \theta_i]^T$, and the control input vector of UAV_i is $u_i = [u_i^{\varphi}, u_i^{\theta}, u_i^{V}]^T$. The UAV kinematic constraints are shown as follows:

$$\begin{cases} \left|u_i^{\varphi}\right| \le u_{\max}^{\varphi} \\ \left|u_i^{\theta}\right| \le u_{\max}^{\theta} \\ \left|u_i^{V}\right| \le u_{\max}^{V} \\ 0 \le V_i \le V_{\max} \\ H_{\min} \le z_i \le H_{\max} \end{cases} \tag{7}$$

where $H_{\min}$ and $H_{\max}$ represent the minimum and maximum flight altitude of the UAVs, respectively. $u_{\max}^{\varphi}$, $u_{\max}^{\theta}$ and $u_{\max}^{V}$ are the maximum value for the control input u_i^{φ}, u_i^{θ} and u_i^{V}. $V_{\min}$ is the maximum speed.

2.3 Target model

In this work, the target moving on the ground is modeled as a mass point. The next position of the target can be obtained as follows [10]:

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

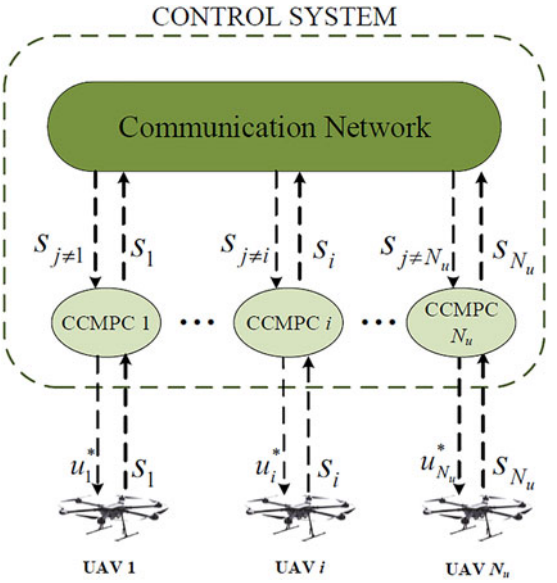


Figure 3. *The DMPC framework.*

$$x_{t,k+1} = v_{t,k}\Delta T \sin\varphi_{t,k} + x_{t,k} \tag{8}$$

$$y_{t,k+1} = v_{t,k}\Delta T \cos\varphi_{t,k} + y_{t,k} \tag{9}$$

where $(x_{t,k+1}, y_{t,k+1})$ and $(x_{t,k}, y_{t,k})$ are the position of the target at time $k+1$ and k, respectively. $v_{t,k}$ and $\varphi_{t,k}$ are the speed and heading angle of the target at time k, respectively. ΔT is the sampling time.

3.0 DMPC for UAVS target tracking in the oilfield environment

3.1 Description of DMPC

DMPC, as a typical control strategy, is used to predict future states and determine the sequence of the control inputs. The DMPC framework is shown in Fig. 3.

The DMPC process is described as follows. At time k, $u_i[k:k+N-1\,]$=[$u_i(k), \ldots, u_i(k+N-1)$] is the control-input sequences in future for the UAV$_i$, and N is the predicted steps. The control-input sequence is initialised independently and synchronously. $\mathbf{s}_i[k:k+N-1\,]$=[$s_i(k), \ldots, s_i(k+N-1)$] is the future state sequences, and the state information is shared between the UAVs through the communication network. $J_i(s[k:+N-1], u_i[k:k+N-1])$ is the cost function and need to be minimised for obtaining the optimal control-input sequence (Equation (10)). The step k in the optimised control-input sequence is only executed for the accuracy, and the subsequent steps ($u_i[k+1:k+N-1]$) are served as the initial control-inputs for the next prediction. The above procedure is repeated until the termination condition is satisfied.

$$u_i^*\,[k:k+N-1] = \arg\min J_i\,(s[k:k+N-1], u_i[k:k+N-1]) \tag{10}$$

3.2 Cost function

To obtain the optimal control-input sequences, the cost function J_i is established in Equation (11) and optimised.

$$J_i(s[k:k+N-1], u_i[k:k+N-1]) = J_{\text{Obs}} + J_{\text{Dis}} + J_{\text{FOV}} + J_{\text{LOS}} + J_{\text{safe}} + J_{\text{com}} + J_u \tag{11}$$

where the cost function includes the obstacle avoidance cost J_{Obs}, the distance cost J_{Dis}, the FOV cost J_{FOV}, the LOS occlusion cost J_{LOS}, the safety cost J_{safe}, the communication cost J_{com} and the control cost J_u.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

3.2.1 Obstacle avoidance cost $J_{\mathbf{Obs}}$

The UAVs must avoid obstacles in the oilfield environment in the tracking for safety. The obstacle avoidance cost is written in Equation (12). $F_i(j)$ represents the avoidance of the static obstacles, which is expressed in Equation (13) [29]; $B_i(j)$ represents the avoidance of the dynamic obstacles based on the chance constraint.

$$J_{\text{Obs}} = \lambda_{\text{Obs}} \sum_{j=k}^{k+N-1} \sum_{i=1}^{N_u} [F_i(j) + B_i(j)] \tag{12}$$

where

$$F_i = \sum_{o=1}^{M} P_1 \, (h_o - z_i)$$

$$P_1 = \begin{cases} 1 \; (x_i - x_o)^2 + (y_i - y_o)^2 - r_o^2 \leq 0 \\ 0 \; (x_i - x_o)^2 + (y_i - y_o)^2 - r_o^2 > 0 \end{cases} \tag{13}$$

where (x_o, y_o) and r_o are the position and the radius of the projection centre of the obstacle on the *X-Y* plane, respectively. N is the predicted steps, and N_u is the number of the UAVs. λ_{Obs} is the weight of the obstacle avoidance cost. h_o is the height of the static obstacles, and M is the number of the static obstacles in the oilfield. P_1 is a sub-function for determining the position relationship of the UAVs with static obstacles. The UAVs should be located outside the static obstacle for avoiding the collision, i.e. the distance between the UAVs and the static obstacle is larger than the radius of the static obstacle. The UAVs collide with the static obstacles when the distance is less than the radius and the UAV flight height is lower than the height of the static obstacles.

For the dynamic obstacles, the chance constraints have been used in this work including the position chance constraint and the velocity chance constraint, which is reported in Ref. [27]. In Fig. 4, the position space C_{pd} and the velocity space C_{vd} are defined, respectively. In order to explain conveniently, the point P is introduced as below. A vertical line is drawn from the centre O of the obstacle and intersects with the extension line in the direction of the relative velocity $(v_i - v_d)$. The cross point is defined as P. In order to avoid colliding with the dynamic obstacles, the probability of the UAV entering the position space is limited below a set threshold and suppresses the probability of the point P entering the velocity space. The collision probability for the position chance constraints is described as follows:

$$\mathrm{g}_{pid} = a_{pid}^T(\hat{p}_i - \hat{p}_d) - b_{id} - erf^{-1}(1 - 2\sigma_{pid})\sqrt{2a_{pid}^T \, (\Sigma_i + \Sigma_d) \, a_{pid}} \tag{14}$$

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{15}$$

$$a_{pid} = \frac{(\hat{p}_i - \hat{p}_d)}{||\, \hat{p}_i - \hat{p}_d||} \tag{16}$$

$$B_p = \frac{w_p}{\mathrm{g}_{pid}^2} \tag{17}$$

where r_i and r_d are the radius of the UAV_i and the dynamic obstacles $(b_{id} = r_i + r_d)$, respectively. σ_{pid} is the collision probability threshold $(0 < \sigma_{pid} < 0.5)$. Σ_i and Σ_d are the error covariance matrices of the UAV_i and the dynamic obstacles, respectively. $erf(.)$ is the standard error function. w_p is the weight coefficient.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

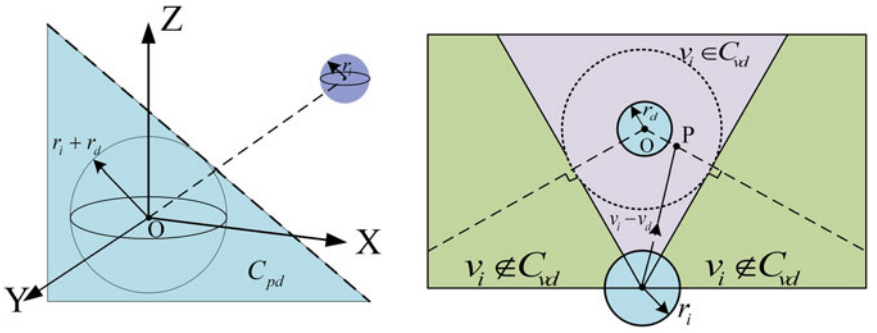


Figure 4. *Position space C_{pd} and speed space C_{vd}.*

The collision probability for the velocity chance constraints is written as below:

$$\mathrm{g}_{vid} = a_{vid}^{T}\hat{r}_{vid} - b_{id} - erf^{-1}(1 - 2\sigma_{\mathrm{vi}})\sqrt{2a_{vid}^{T}(\Sigma_i + \Sigma_d)a_{vid}} \tag{18}$$

$$B_v = \begin{cases} w_v \cdot \mathrm{g}_{vid} & \text{if } \mathrm{g}_{vid} \geq 0 \\ 0 & \text{else} \end{cases} \tag{19}$$

$$a_{vid} = \frac{\hat{r}_{vid}}{||\hat{r}_{vid}||} \tag{20}$$

where w_v is the weight coefficient. r_{vid} is the vector from the intersection P to O. σ_{vid} is the same as σ_{pid}.

The chance-constrained cost function is written in Equation (21).

$$B = B_p + B_v \tag{21}$$

3.2.2 Distance cost $J_{\mathbf{Dis}}$

The distance cost is the distance between the UAV and the target on the horizontal plane. The smaller the distance, the better the tracking performance. The distance cost is expressed as:

$$J_{\mathrm{Dis}} = \lambda_{\mathrm{d}} \sum_{j=k}^{k+N-1} \sum_{i=1}^{N_u} \sqrt{(x_i(j) - x_t(j))^2 + (y_i(j) - y_t(j))^2} \tag{22}$$

where (x_i, y_i) and (x_t, y_t) are the coordinates of the UAV and the target on the 2D plane, respectively. λ_d is the weight of the distance cost.

3.2.3 FOV cost $J_{\mathbf{FOV}}$

The FOV cost is used to keep the target in the FOV of the UAV sensor. The target is within the coverage range of the UAV sensors when the detection distance and the field angle constraints are satisfied simultaneously. The FOV cost is defined as [19]:

$$J_{\mathrm{FOV}} = \begin{cases} \lambda_{m1} \sum_{j=k}^{k+N-1} \sum_{i=1}^{N_u} \left(1 - \exp\left[-c_1 \left(\frac{|p_i(j) - p_t(j)|}{R_{\max}}\right)^2\right]\right), \\ |p_i(j) - p_t(j)|^2 \leq R_{\max}^2 \cap \left(\cos^{-1}\left(\frac{Z_i}{|p_i(j) - p_t(j)|}\right) \leq \gamma_{\max}\right) \\ \sum_{j=k}^{k+N-1} \sum_{i=1}^{N_u} \lambda_{m1} + \lambda_{m2} |p_i(j) - p_t(j)|, \\ |p_i(j) - p_t(j)|^2 > R_{\max}^2 \cup \left(\cos^{-1}\left(\frac{Z_i}{|p_i(j) - p_t(j)|}\right) > \gamma_{\max}\right) \end{cases} \tag{23}$$

where λ_{m1} and λ_{m2} are the weight coefficients. c_1 is the parameter determining the shape of J_{FOV}. $R_{\max}$ and $\gamma_{\max}$ represent the maximum detection distance and the field angle of the UAV sensor, respectively.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

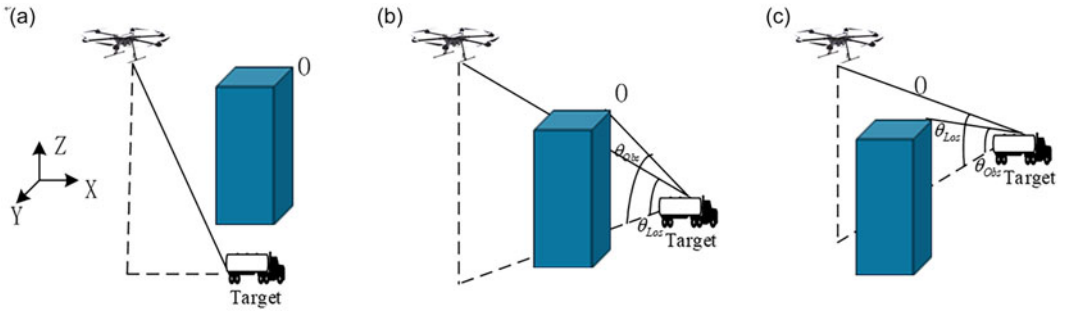


Figure 5. *LOS obstruction conditions.*
Note: (a) LOS is not obstructed. (b) LOS is obstructed. The LOS's projection on the X-Y plane intersects with the obstacle and $\theta_{\text{Obs}} > \theta_{\text{LOS}}$. *(c) LOS is not obstructed. The LOS's projection on the X-Y plane intersects with the obstacle and* $\theta_{\text{Obs}} < \theta_{\text{LOS}}$.

The target is within the coverage range of UAV sensors when the detection distance and field angle constraints are satisfied simultaneously.

3.2.4 LOS occlusion cost J_{LOS}

The obstacles in the oilfield environment, including the static and dynamic obstacles, can obscure the UAV's LOS to the target. Figure 5 shows the LOS obstruction conditions. To determine the relationship between the LOS's projection on the *X-Y* plane and the obstacle, a sub-function P_1 is introduced according to Ref. [29].

$$P_1 = \begin{cases} 1 & \dfrac{|mx_a - y_a + y_t - mx_t|}{\sqrt{m^2 + (-1)^2}} - r_a \le 0 \\ 0 & \dfrac{|mx_a - y_a + y_t - mx_t|}{\sqrt{m^2 + (-1)^2}} - r_a > 0 \end{cases} \tag{24}$$

where (x_a, y_a) and r_a are the position and the radius of the projection centre of the obstacle on the *X-Y* plane, respectively. m is the slope of the LOS's projection on the *X-Y* plane. The LOS is not obstructed at $P_1 = 0$, but LOS is obstructed by the obstacles at $P_1 = 1$ and $\theta_{\text{Obs}} > \theta_{\text{LOS}}$.

$$\begin{aligned} J_{\text{LOS}} &= \lambda_{\text{LOS}} \sum_{j=k}^{k+N-1} \sum_{i=1}^{Nu} P_1 \max\left(0, \tan(\theta_{\text{Obs}}) - \tan\left(\theta^i_{\text{LOS}}(j)\right)\right) \\ \tan(\theta_{\text{Obs}}) &= \sum_{a=1}^{A} \frac{h_a}{\sqrt{(x_t - x_a)^2 + (y_t - y_a)^2} - r_a} \\ \tan(\theta_{\text{LOS}}) &= \frac{Z_i}{\sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}} \end{aligned} \tag{25}$$

where λ_{LOS} is the weight coefficient. A is the number of obstacles in the oilfield, including the static and dynamic obstacles.

3.2.5 Safety cost J_{safe} *and communication cost* J_{com}

The safety distance cost J_{safe} and the communication distance cost J_{com} are introduced to keep the safety and communication for the UAVs, which is normalised between 0 and 1 by the hyperbolic tangent functions [18] :

$$f_{\text{Sdi}}(d, d_{\min}, \Delta d) = \frac{1}{2}\left[1 - \tanh\left(\frac{d - d_{\min} - \frac{\Delta d}{2}}{\frac{\Delta d}{8}}\right)\right] \tag{26}$$

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

$$f_{\mathrm{Cdi}}(d, d_{\max}, \Delta d) = \frac{1}{2}\left[1 + \tanh\left(\frac{d - d_{\max} + \frac{\Delta d}{2}}{\frac{\Delta d}{8}}\right)\right] \tag{27}$$

where $d_{\min}$ and $d_{\max}$ represent the minimum safe distance and maximum communication distance, respectively. d is the distance between the UAVs, and Δd is the buffer distance ($0 < \Delta d << d_{\min}$).

$$J_{\mathrm{safe}} = \lambda_{\mathrm{safe}} \sum_{j=k}^{k+N-1} \sum_{i \neq j}^{N_u} f_{\mathrm{Sdi}}(d_{\mathrm{ij}}, d_{\min}, \Delta d) \tag{28}$$

$$J_{\mathrm{com}} = \lambda_{\mathrm{com}} \sum_{j=k}^{k+N-1} \sum_{i \neq j}^{N_u} f_{\mathrm{Cdi}}(d_{\mathrm{ij}}, d_{\max}, \Delta d) \tag{29}$$

$$d_{\mathrm{ij}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{30}$$

where d_{ij} is the distance between UAV_i and UAV_j ($i \neq j$), λ_{safe} and λ_{com} are the weights of the safe and communication costs, respectively.

3.2.6 Control cost $J_{\mathbf{u}}$

UAVs have limited energy in the tracking target. Balancing control inputs helps reduce energy consumption. The balance of the control inputs is introduced as the control cost, which is given by:

$$J_u = \lambda_u \sum_{j=k}^{k+N-1} \sum_{i=1}^{N_u} ||u_i(j) - u_i(j-1)|| \tag{31}$$

where λ_u is a parameter.

4.0 ISSA for solving CCDMPC formulation

To obtain the optimal control sequences by minimising the cost function, the solver has been proposed based on the salp swarm algorithm, integrating the sine-tent-cosine mapping, differential mutation operations and adaptive inertia weights. The sine-tent-cosine mapping strategy is introduced into the initial stage to improve the uniform distribution for the population. The differential variance operation is introduced into the leader position update to improve the convergence speed and computational accuracy. The adaptive inertia weights are introduced into the follower position update to balance the global and local search capability. The improved SSA and strategies are described as follows.

4.1 SSA method

SSA is inspired by the swarming behaviour of the salps with the great performance including superior exploration-exploitation balance, practical simplicity and robust performance [30]. The population is divided into the leader and follower groups. The leaders are individuals at the front of the chain, leading the entire population to search for food sources. The other individuals as the followers connect one by one in the chain, following the leaders. The food source is identified, and the information is transferred from one individual to another. The mathematical process is described below.

First, the population is initialised randomly as:

$$x_{W \times D} = rand(W, D) \times (ub - lb) + lb \tag{32}$$

where W is the population size and D is the dimension.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

In SSA, the position of the leader is updated as:

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)\, c_2 + lb_j), & c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)\, c_2 + lb_j), & \text{else} \end{cases} \tag{33}$$

where x_j^1 is the location of the leader in the jth dimension. F_j is the location of the food source in the jth dimension. ub_j and lb_j are the upper and lower bounds in the search space. c_2 and c_3 are the random numbers distributed uniformly in the interval [0, 1]. c_1 decreases adaptively with the iteration and is denoted as Equation (34).

$$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^2} \tag{34}$$

where t is the current iteration, and $T_{\max}$ is the maximum iteration.

In the salp chain, the position of the followers is updated as Equation (35).

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{35}$$

where x_j^{i-1} and x_j^i are the positions of the $(i-1)$ th following individual and the ith following individual in the jth dimension, respectively ($i \geq 2$,).

4.2 ISSA method

4.2.1 Sine-tent-cosine chaotic mapping

The initialisation stage is very important to improve the convergence speed and optimisation accuracy for SSA. The population of the traditional SSA is initialised randomly. In Ref. [31], the initial population is distributed uniformly in the search space and is helpful for improving the performance.

In this work, sine-tent-cosine chaotic mapping is used to initialise the salp population. In the sine-tent-cosine chaotic mapping, a continuous chaotic sequence is first generated using the sine function, and then the sequence is segmented with the tent function, and finally, the sequence is further transformed using the cosine function. The mathematical expression is defined as Equation (36).

$$x\,(i+1) = \begin{cases} \cos\left(\pi\left(r\sin(\pi x(i)) + 2(1-r)x(i) - 0.5\right)\right), & x(i) < 0.5 \\ \cos\left(\pi\left(r\sin(\pi x(i)) + 2(1-r)(1-x(i)) - 0.5\right)\right), & \text{else} \end{cases} \tag{36}$$

where $x(i)$ denotes the input value in the current iteration, $x(i+1)$ is the output value in the next iteration and r is a parameter in the range [0, 1].

4.2.2 Leader position updating based on differential variance operation

As can be seen from Equation (33), the leaders update their position according to the food source. However, the individuals do not share information in the updating stage. Inspired by the differential variance, the mutation strategy of DE/best/2 is used to improve the convergence speed and computational accuracy. The updating formula is written as Equation (37).

$$x_j^1 = F_j + \mathrm{g}\left(x_j^a - x_j^b\right) + \mathrm{g}\left(x_j^c - x_j^d\right) \tag{37}$$

where F_j is the location of the food source in the jth dimension, and g is the ratio factor. x_j^a, x_j^b, x_j^c and x_j^d are the different individuals selected from the population.

4.2.3 Follower position updating based on nonlinear adaptive inertia weights

To trade off the exploratory and exploitative capabilities in searching, the nonlinear adaptive inertia weights are introduced to update the position of the followers. The follower position is updated as Equations (38) and (39).

$$x_j^i = \frac{1}{2}\left(x_j^i + \omega(t) x_j^{i-1}\right) \tag{38}$$

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

Algorithm 1 The pseudo-code of ISSA

Setting the population size $W = 30$, the maximum iteration $T_{\max} = 100$, the parameter $r = 0.6$ in the sine-tent-cosine chaotic mapping, the ratio factor g=0.8 and the weight control factor $K = 8$.
Initialising the population with the **Sine-Tent-Cosine chaotic mapping strategy** (Equation (36)), calculating the fitness value for all the individuals and selecting the optimal individual as the food source.
while $t \leq T_{\max}$ **do**
  for $i = 1{:}W$ **do**
    if $i < \frac{W}{2}$ **then**
      Updating the leader position with **the differential variance operation strategy** (Equation (37)).
    else
      Updating the follower position with **the adaptive inertia weights strategy** (Equation (38)).
    end if
  end for
  Calculating the fitness values for all the individuals and updating the food source location.
  $t = t+1$
end while
Output the best solution in the search space.

$$\omega(t) = \frac{0.55}{\sqrt{1 + \left(\frac{Kt}{T_{\max}}\right)^{10}}} + 0.4 \tag{39}$$

where K is the weight control factor, and t is the current iteration. $T_{\max}$ is the maximum iteration. The weights are initialised as the maximum value of 0.95. As the iteration increases, the weights gradually converge to the minimum value of 0.4.

The pseudo-code of ISSA is shown as Algorithm 1. The flowchart of the trajectory optimisation process is shown in Fig. 6.

5.0 Experimental simulation and result discussion

To verify the performance of the proposed algorithm, Fig. 7 shows the specific oilfield environment with the size of 10 km × 10 km. Several objects in the oilfield environment are modeled. Specifically, the yellow cube represents the house, and the green cylinder represents the trees. The red rectangle represents the pumping well. The purple rectangle and cylinder are oil storage tanks. The yellow cylinder on the road is the oil truck, and the blue rectangle is the signal tower. The cyan rectangle is the water tank, and the pink ellipsoid is the flying bird flock. The dashed grey areas around the pumping wells and storage tanks represent the pipelines.

The simulation experiments are conducted in MATLAB R2022b software with the following configurations: CPU 2.90 GHz, RAM 16.0 GB, and Windows 10. The total running time is 1650 s, and the sampling time ΔT is 6 s. The prediction horizon length N is 5. Two UAVs are used to track targets invading the oilfield environment. The UAV initial positions are (560, 3080, 80) m and (510, 3000, 80) m. The initial flight speed V_0 is 20 m/s. The initial heading angle φ_0 and the initial pitch angle θ_0 are 90° and 0°, respectively. The sensor field angle $\gamma_{\max}$ is 60°. The maximum detection distance $R_{\max}$ is 400 m, the minimum safe distance $d_{\min}$ between the UAVs is 50 m. The maximum communication distance $d_{\max}$ is 1500 m. The safe distance d_{US} between UAVs and the bird flock is 200 m. The minimum and maximum UAV flight height are $H_{\min} = 40$ m and $H_{\max} = 100$ m. The minimum and maximum UAV flight velocity are $V_{\min} = 17$ m/s and $V_{\max} = 30$ m/s. The collision probability threshold is $\sigma_{pid} = \sigma_{vid} = 0.02$. The weight coefficient $w_p = 0.5$ and $w_v = 5 \times 10^6$.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

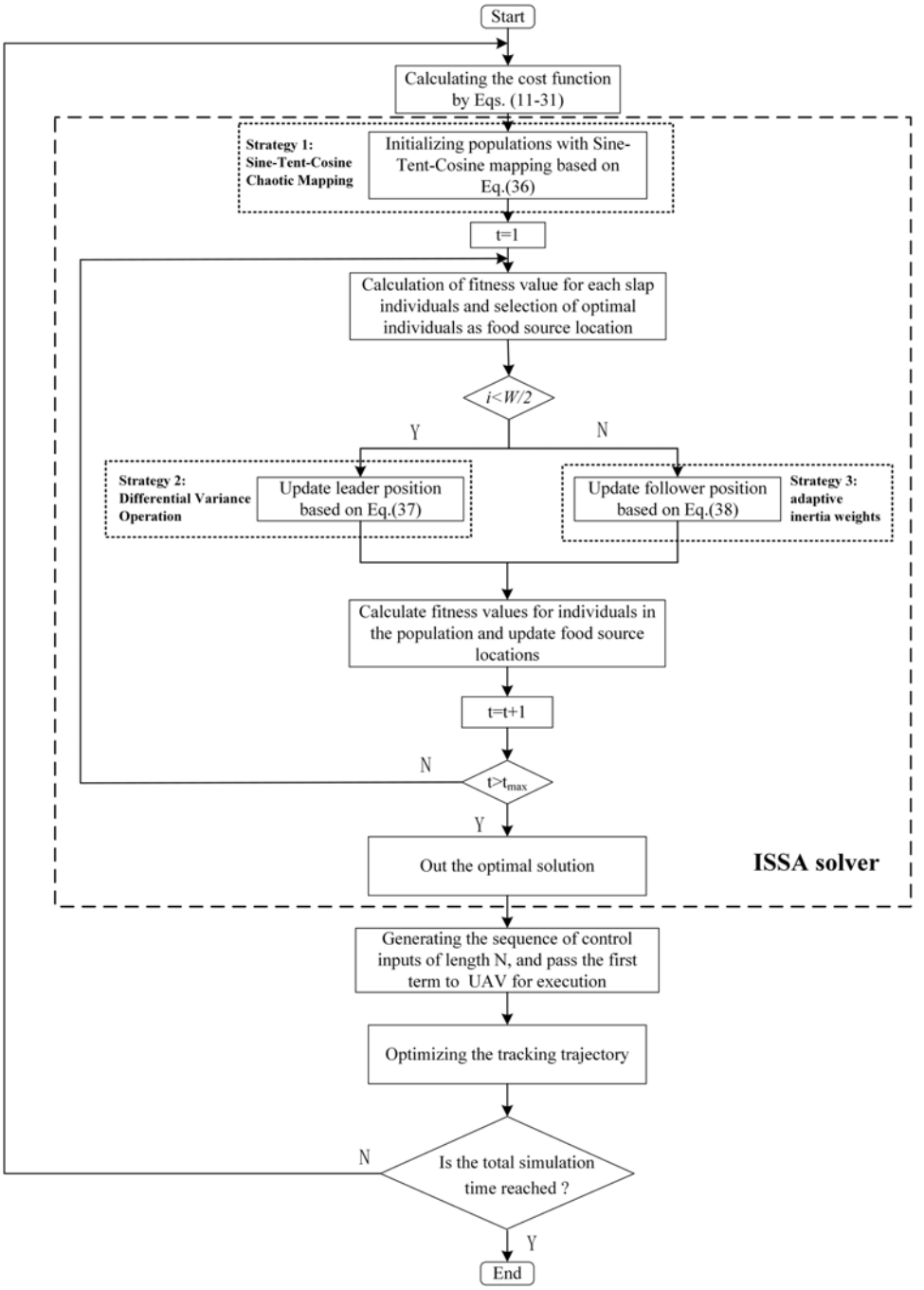


Figure 6. *The flowchart of the trajectory optimisation process.*

5.1 Simulation experiments and results

In this section, the ISSA is proposed as the solver for CCDMPC and compared with the seven typical algorithms. The specific parameter settings are shown in Table 1. The population size is $W = 30$, and the maximum iteration $T_{\max} = 100$.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

Table 1. *Parameter settings for the eight algorithms*

Algorithms	Specific parameters
ISSA	$r = 0.6$, $g = 0.8$, $K = 8$
WOA	$b = 1$
GWO	Decreases linearly from 2 to 0, and r_1 and r_2 are random numbers in [0, 1]
PSO	$c_1 = c_2 = 0.5$
HHO	q, r_1, r_2, r_3, r_4, r_5 are random numbers in [0, 1]
CGWO	Decreases linearly from 2 to 0, and r_1 and r_2 are random numbers in [0, 1]
HJPSO	$c_1 = c_2 = 0.5$
AOA-HHO	q, r_1, r_2, r_3, r_4, r_5 are random numbers in [0, 1]

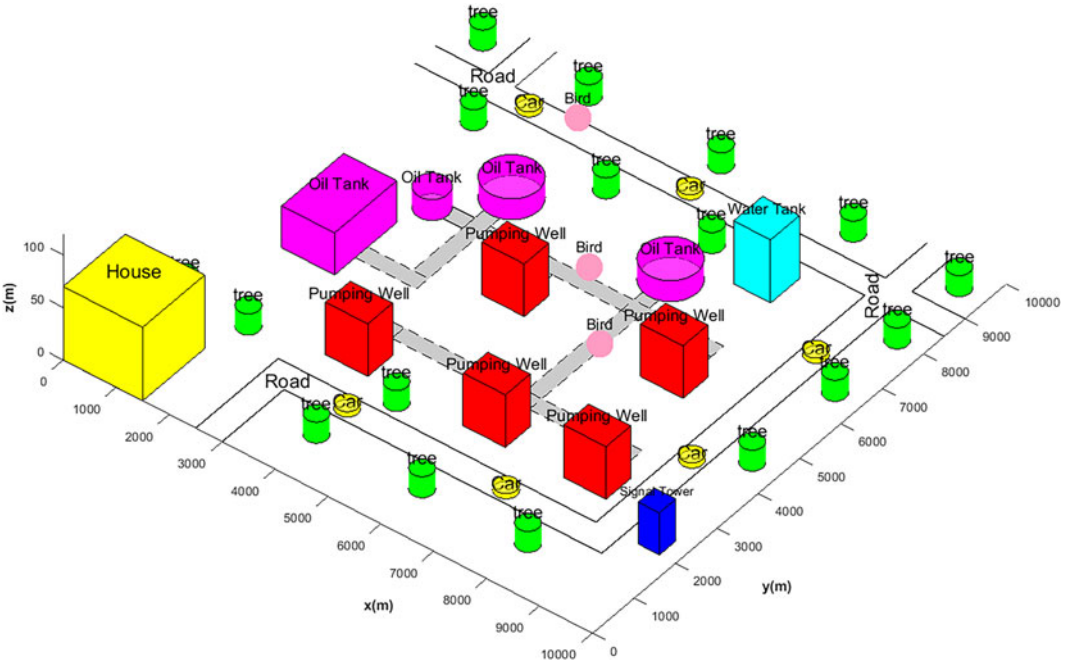


Figure 7. *Oilfield environment model.*

The target tracking trajectories planned by the eight algorithms are shown in Figs 8–11. All the planned trajectories can avoid static obstacles. However, the tracking trajectory planned by the ISSA method fits the motion trajectory of the moving obstacle more closely, and thus the tracking performance is better, which can be seen in Fig. 11.

To confirm the dynamic obstacle avoidance, the real-time distance between the UAVs and the bird flock is exhibited in Figs 12–15. Assume that the radius of the dynamic obstacles is 200 m and the UAV is regarded as a particle. d_{US} is defined as the Euclidean distance between the UAVs and dynamic obstacles. To prevent collision between the UAVs and dynamic obstacles, it should satisfy $d_{US} \geq 200$ m. The distance between the UAVs and the bird flock falls below 200 m during 1128–1140s for GWO, and the distance is less than 200 m during 1116–1128 s for CGWO. This indicates that the UAVs collided with the bird flock in this interval. This fact also appears for WOA and PSO. However, the proposed ISSA method has planned the trajectories with the safe distance at the tracking time, as well as HHO, HJPSO and AOA-HHO.

Figures 16–19 show the total tracking time for moving targets. The undetected flag is set to 0 when one UAV can track the target; otherwise, the undetected flag is set to 1. The two UAVs cannot track

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

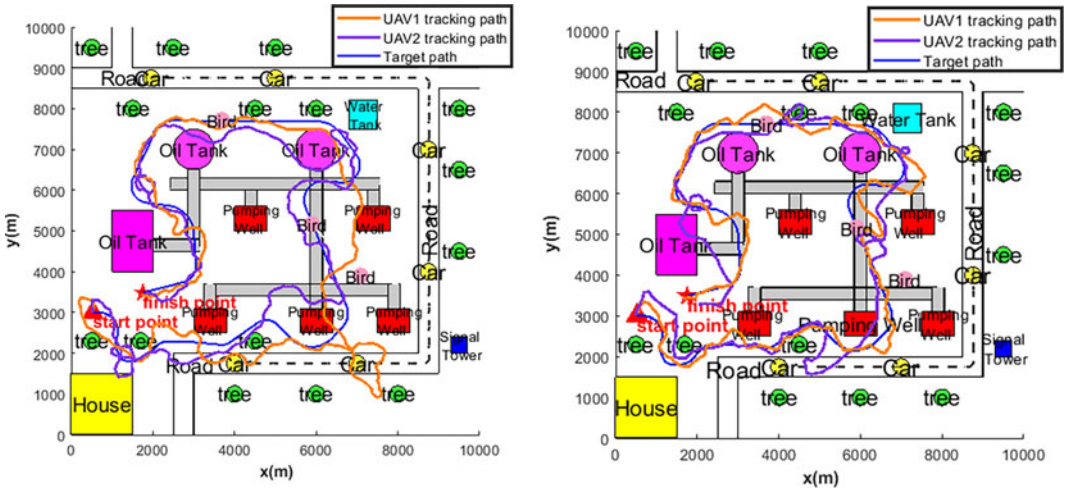


Figure 8. *Tracking trajectories planned by the WOA and GWO, respectively, in top-view.*

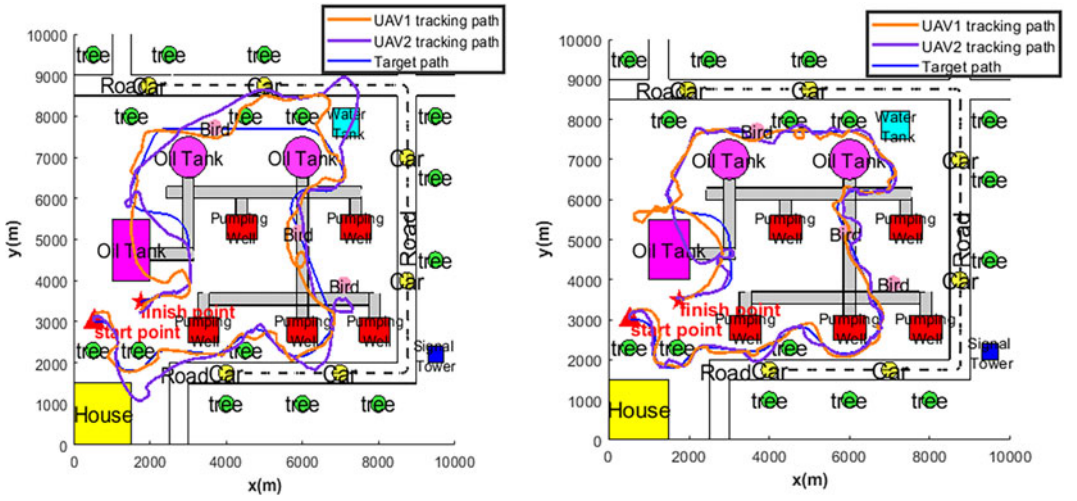


Figure 9. *Tracking trajectories planned by the PSO and HHO, respectively, in top-view.*

the moving target at the first 400 and 200 s for WOA and CGWO, respectively, due to the high density of the obstacle distribution. This indicates that the WOA and CGWO methods are difficult in solving the tracking problem with the high density of the obstacles. This fact also appears for the AOA-HHO method and even becomes more serious in the whole tracking process for the PSO method. The GWO, HHO and HJPSO methods show better performance in tracking time. The proposed method, ISSA, can provide the best tracking trajectories for the UAVs in the whole tracking process. The tracking time reaches 1620 s and accounts for 98.2% of the tracking time.

Figure 20 is the graph of the distance between the UAVs and the schematic of the flight altitude during target tracking using the CCDMPC+ISSA method, respectively. The distance graph shows UAVs can avoid colliding and keep communicating with each other. In addition, the UAV moves up and down within a limited altitude range, which prevents the UAV from colliding with obstacles in the oilfield environment or obscuring the LOS between the UAV and the target.

Table 2 gives statistical results for all the algorithms. The algorithms run independently 20 times. Note that the algorithm with the addition of sine-tent-cosine mapping is referred to as the sine-tent-cosine

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

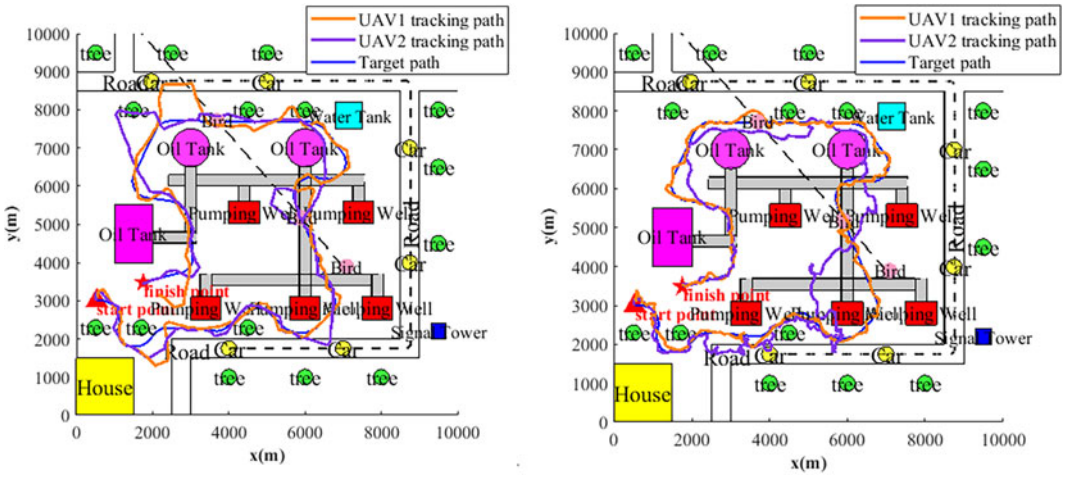


Figure 10. *Tracking trajectories planned by the CGWO and HJPSO, respectively, in top-view.*

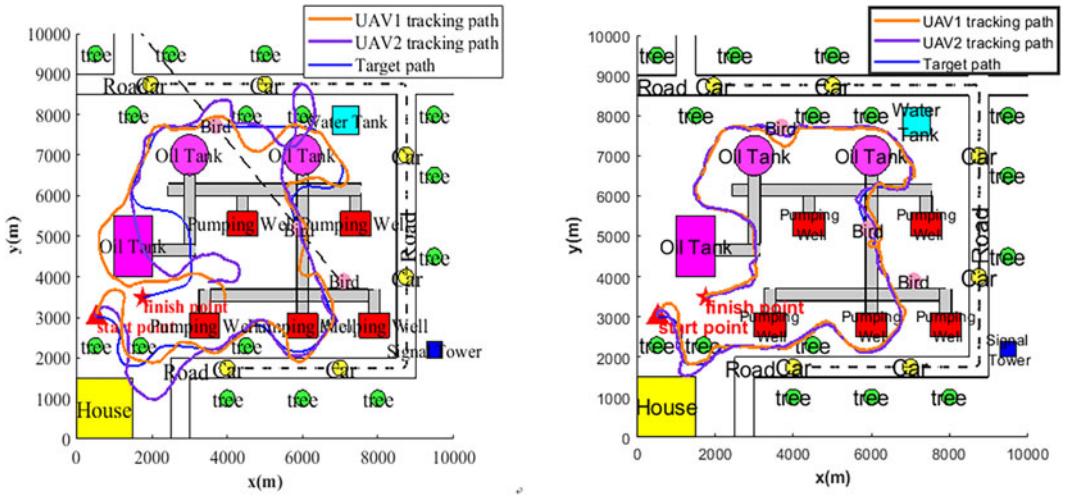


Figure 11. *Tracking trajectories planned by the AOA-HHO and ISSA, respectively, in top-view.*

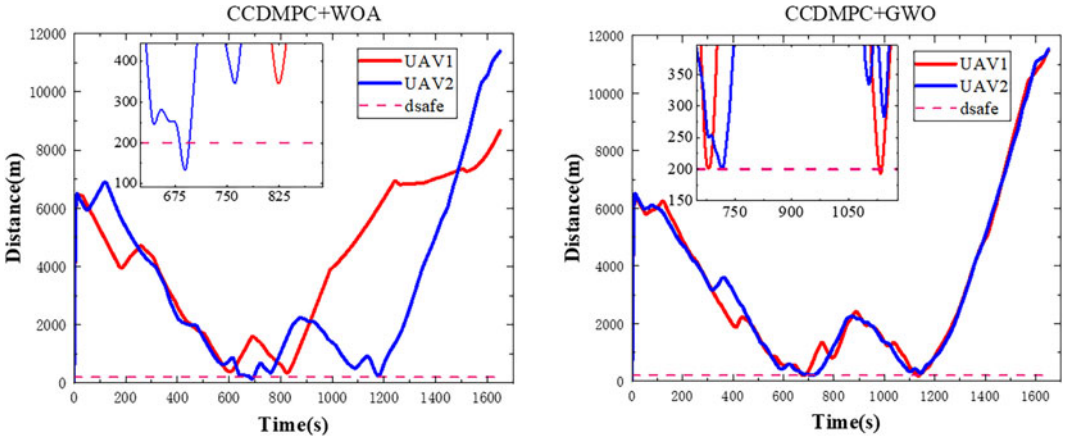


Figure 12. *Real-time distance between the UAVs and the bird flock using CCDMPC+WOA and CCDMPC+GWO.*

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

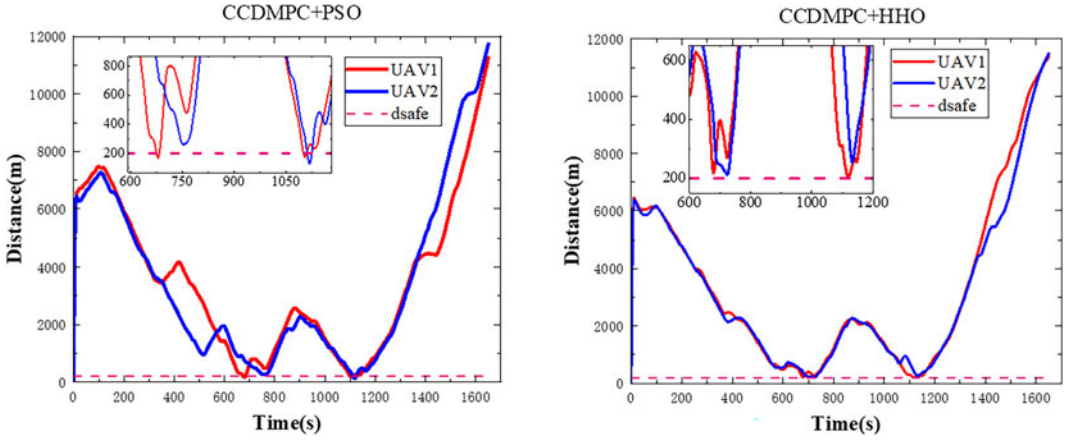


Figure 13. *Real-time distance between the UAVs and the bird flock using CCDMPC+PSO and CCDMPC+HHO.*

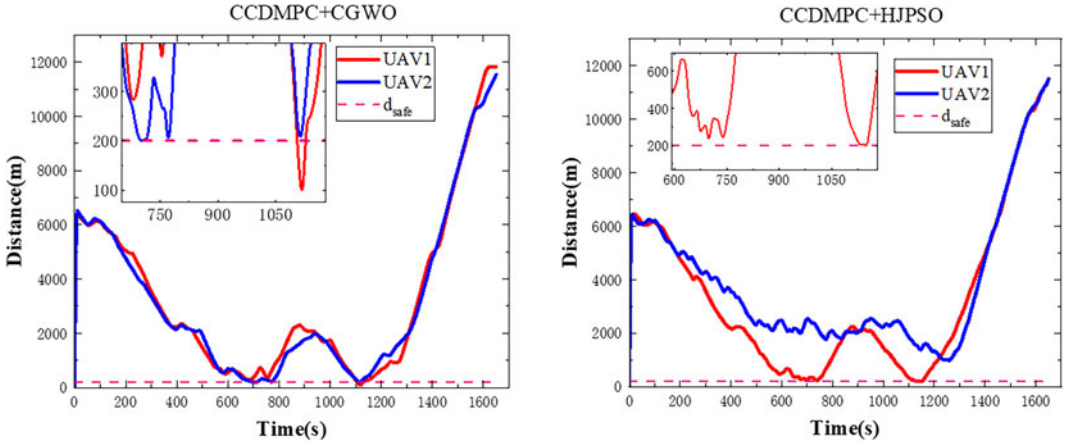


Figure 14. *Real-time distance between the UAVs and the bird flock using CCDMPC+CGWO and CCDMPC+HJPSO.*

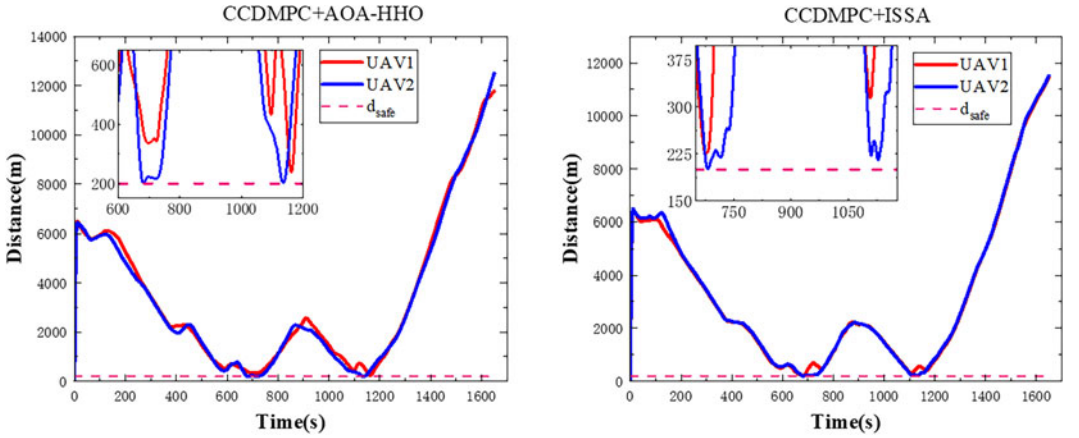


Figure 15. *Real-time distance between the UAVs and the bird flock using CCDMPC+AOA-HHO and CCDMPC+ISSA.*

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

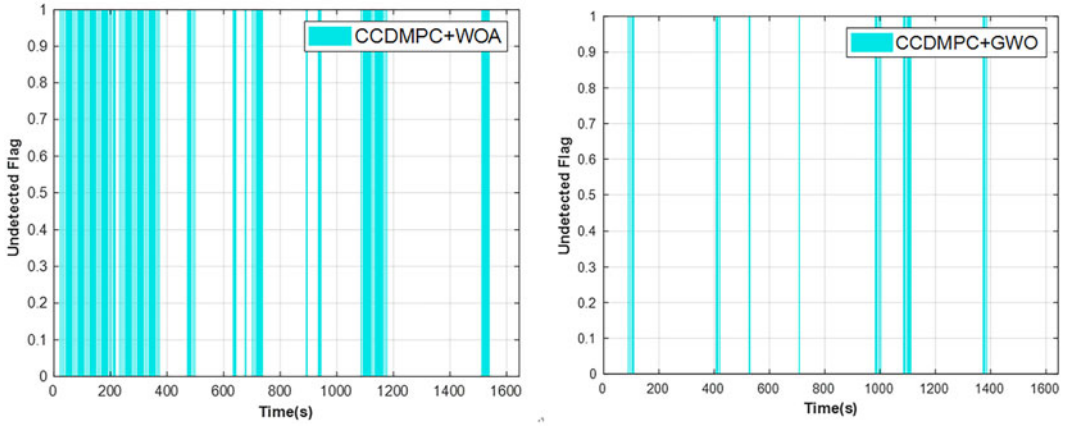


Figure 16. *The total tracking time of the moving target using CCDMPC+WOA and CCDMPC+GWO.*

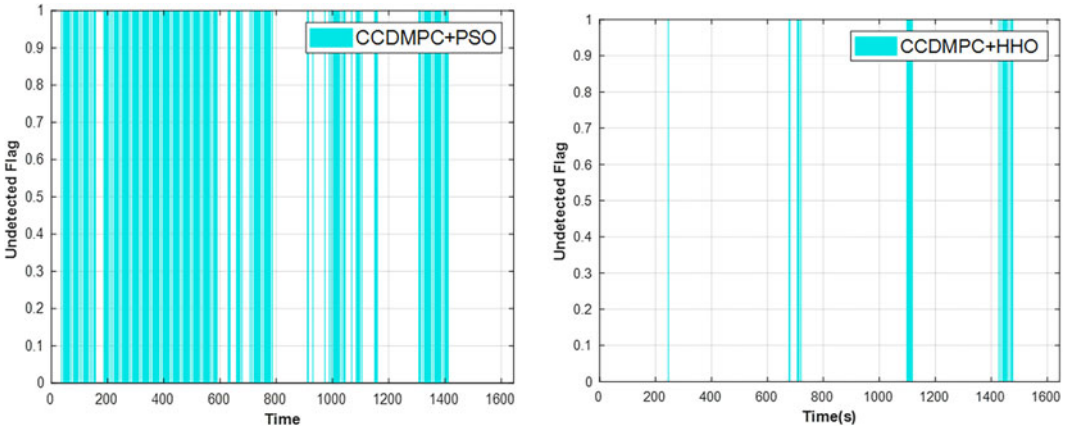


Figure 17. *The total tracking time of the moving target using CCDMPC+PSO and CCDMPC+HHO.*

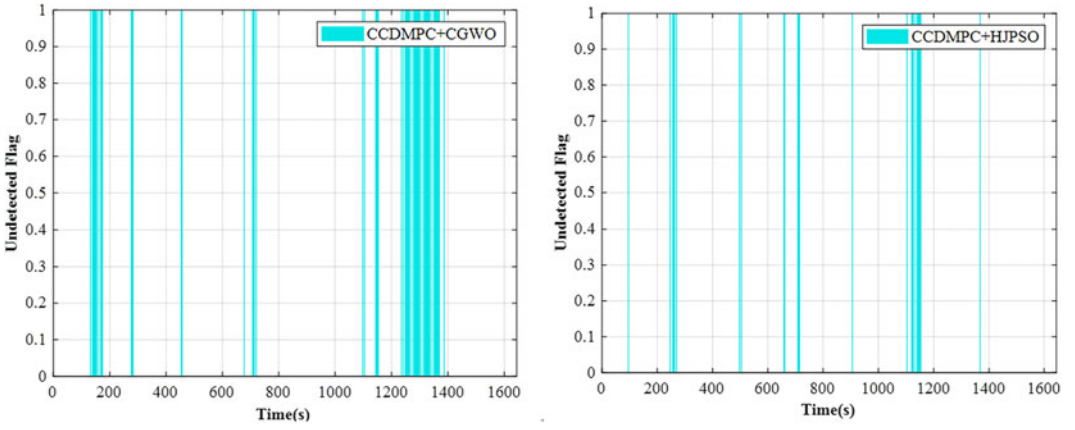


Figure 18. *The total tracking time of the moving target using CCDMPC+CGWO and CCDMPC+HJPSO.*

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

Table 2. *Statistical results of the total tracking time for all the algorithms*

Performances	SSA	TSSA	TDSSA	ISSA	WOA	GWO	PSO	HHO	CGWO	HJPSO	AOA-HHO
Max(s)	1308.0	1392.0	1548.0	**1632.0**	1074.0	1524.0	768.0	1542.0	1554.0	1614.0	1536.0
Ave(s)	1251.3	1340.1	1510.8	**1623.0**	1024.8	1434.0	675.6	1443.9	1394.8	1538.8	1295.81
Min(s)	1170.0	1272.0	1470.0	**1608.0**	990.0	1362.0	576.0	1338.0	1086.0	1284.0	1002.0
Std(s)	6.168	6.643	3.982	**1.192**	4.149	8.085	9.906	9.609	16.846	13.655	23.345
ASO(%)	100	100	100	**100**	100	100	100	100	100	100	100
ADO(%)	80	80	95	**100**	80	90	75	100	70	85	100

ASO: Avoiding static obstacles, ADO: Avoiding dynamic obstacles.

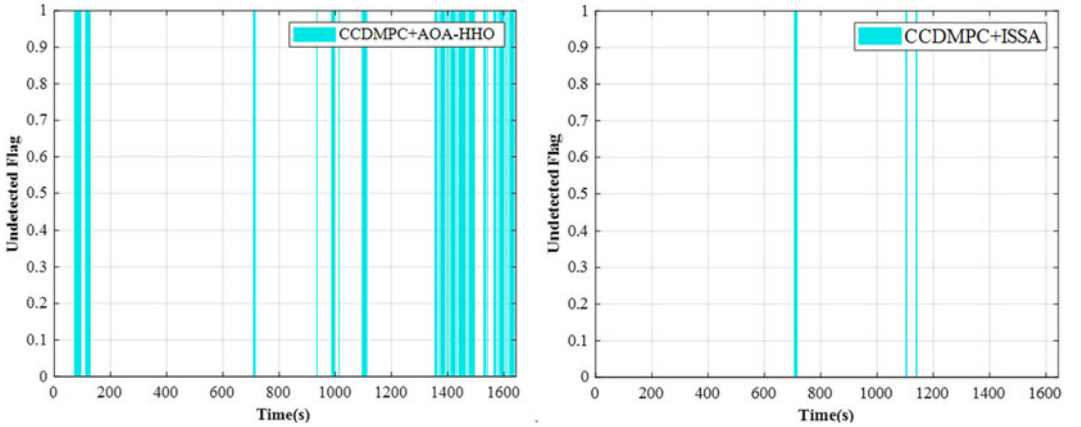


Figure 19. *The total tracking time of the moving target using CCDMPC+AOA-HHO and CCDMPC+ISSA.*

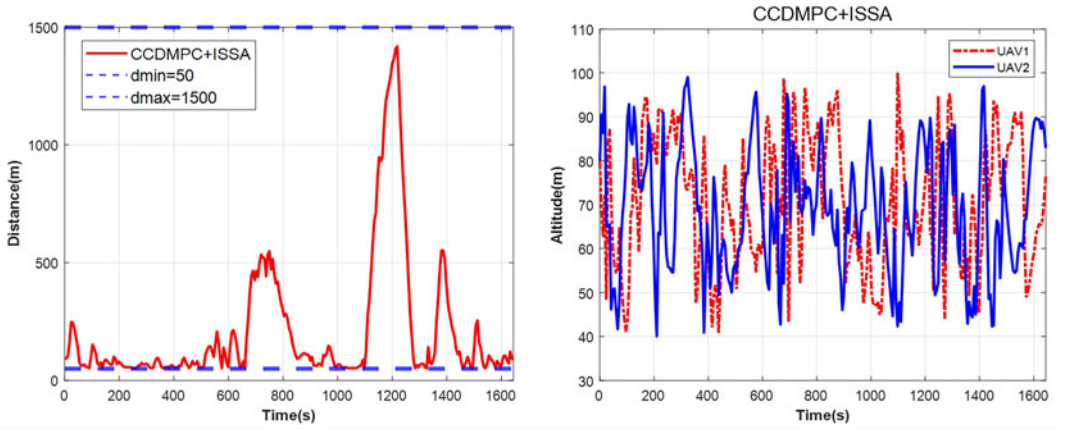


Figure 20. *Distance graph between UAVs and flight altitude graph of UAVs.*

salp swarm algorithm (TSSA), and the algorithm incorporating both the sine-tent-cosine mapping and differential variance operation is referred to as the sine-tent-cosine differential salp swarm algorithm (TDSSA). The indicators, such as maximum duration, average duration and standard deviation, show the superiority of the algorithms proposed in this paper. This also confirms that the strategies introduced have enhanced the capability of SSA in solving the UAV target tracking problem. In 20 runs, the proposed algorithm guarantees that tracking can avoid dynamic and static obstacles in the environment. The trajectories planned by the HHO and AOA-HHO algorithms are also capable of avoiding obstacles but become weaker than the proposed algorithm in other indicators.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

6.0 Conclusion

In this work, the UAV tracking the moving target in the dynamic oilfield environment has been studied. The oilfield environment is modeled as well as the UAV kinematic and target models. The CCDMPC method is used as the framework for the trajectory optimisation of UAVs. For the solver of CCDMPC, the three strategies have been introduced into the typical SSA to improve the optimisation search capability, including the sine-tent-cosine chaotic mapping in the initialisation phase, the differential mutation operation in the leader position updating and the adaptive inertia weights in the follower position updating. The simulation results show that the CCDMPC method based on the ISSA can improve the performance of target tracking in oilfield environments with dynamic obstacles.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant Nos. 62373389, 62576372), the leading Talents of Science and Technology in the Central Plain of China (Grant No. 254000510055), the Science and Technology Innovation Talents of Colleges and Universities in Henan Province (Grant No. 24HASTIT037), the Key Research and Development Program of Henan (Grant Nos. 241111210100), the Natural Science Foundation of Zhongyuan University of Technology (Grant No. K2025ZD008), the Frontier Exploration Projects of Longmen Laboratory (LMQYTSKT031), the Postgraduate Education Reform and Quality Improvement Project of Henan Province (YJS2026YBGZZ20, 202622) and the Foundation Research Project of Henan Provincial Key Scientific Research Program in Higher Education Institutions (26ZX023).

Data availability statement. The data that support the findings of this study are available from the authors upon reasonable request and with permission of corresponding authors of the dataset.

Author contributions. Conceptualisation, methodology, X.C. and L.Y.; validation, X.C. and Y.L.; formal analysis, Z.L. and H.Y; investigation, X.C.; resources, L.Y.; data curation, X.C.; writing – original draft preparation, X.C. and Y.L.; writing – review and editing, X.C., Z.L. and L.Y.; supervision, P.L.; project administration, L.Y.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

Competing interests. The authors have no competing interests.

Ethics approval. Not applicable.

Code availability. Codes are not publicly available duo to private restrictions.

References

[1] Hung, H.-A., Hsu, H.-H. and Cheng, T.-H. Optimal sensing for tracking task by heterogeneous multi-UAV systems, *IEEE Trans. Control Syst. Technol.*, 2023, **32**, (1), pp 282–289.
[2] Xiong, X., He, M., Li, T., Zheng, G., Xu, W., Fan, X. and Zhang, Y. Adaptive feature fusion and improved attention mechanism based small object detection for UAV target tracking, *IEEE Internet Things J.*, 2024, **11**, (12), pp 21239–21249.
[3] Sun, N., Zhao, J., Shi, Q., Liu, C. and Liu, P. Moving target tracking by unmanned aerial vehicle: A survey and taxonomy, *IEEE Trans. Ind. Inf.*, 2024, **20**, (5), pp 7056–7068.
[4] Shukla, A., Xiaoqian, H. and Karki, H. Autonomous tracking and navigation controller for an unmanned aerial vehicle based on visual data for inspection of oil and gas pipelines, 2016 16th International Conference on Control, Automation and Systems (ICCAS), IEEE, 2016, pp 194–200.
[5] Xu, C., Li, Q., Zhou, Q., Zhang, S., Yu, D. and Ma, Y. Power line-guided automatic electric transmission line inspection system, *IEEE Trans. Instrum. Meas.*, 2022, **71**, pp 1–18.
[6] Kochetkova, L. Pipeline monitoring with unmanned aerial vehicles, *J. Phys.: Conf. Ser.*, 2018, **1015**, p 042021. IOP Publishing.
[7] Altan, A. and Hacoglu, R. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances, *Mech. Syst. Signal Process.*, 2020, **138**, p 106548.
[8] Bertsekas, D.P. Dynamic programming and suboptimal control: A survey from ADP to MPC, *Eur. J. Control*, 2005, **11**, (4-5), pp 310–334.
[9] Marini, F. and Walczak, B. Particle swarm optimization (PSO). A tutorial, *Chemom. Intell. Lab. Syst.*, 2015, **149**, pp 153–165.
[10] Song, Y., Cai, X., Zhou, X., Zhang, B., Chen, H., Li, Y., Deng, W. and Deng, W. Dynamic hybrid mechanism-based differential evolution algorithm and its application, *Expert Syst. Appl.*, 2023, **213**, p 118834.
[11] Mirjalili, S., Mirjalili, S.M. and Lewis, A. Grey wolf optimizer, *Adv. Eng. Software*, 2014, **69**, pp 46–61.
[12] Mirjalili, S. and Lewis, A. The whale optimization algorithm, *Adv. Eng. Software*, 2016, **95**, pp 51–67.
[13] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, 2019, **97**, pp 849–872.

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press

[14] Kuo, R.J. and Chiu, T.-H. Hybrid of jellyfish and particle swarm optimization algorithm-based support vector machine for stock market trend prediction, *Appl. Soft Comput.*, 2024, **154**, p 111394.
[15] Yang, Z. Competing leaders grey wolf optimizer and its application for training multi-layer perceptron classifier, *Expert Syst. Appl.*, 2024, **239**, p 122349.
[16] Qiao, L., Liu, K., Xue, Y., Tang, W. and Salehnia, T. A multi-level thresholding image segmentation method using hybrid arithmetic optimization and Harris hawks optimizer algorithms, *Expert Syst. Appl.*, 2024, **241**, p 122316.
[17] Yu, Y., Wang, H., Liu, S., Guo, L., Yeoh, P.L., Vucetic, B. and Li, Y. Distributed multi-agent target tracking: A Nash-combined adaptive differential evolution method for UAV systems, *IEEE Trans. Veh. Technol.*, 2021, **70**, (8), pp 8122–8133.
[18] Yao, P., Wang, H. and Ji, H. Multi-UAVs tracking target in urban environment by model predictive control and Improved Grey Wolf Optimizer, *Aerosp. Sci. Technol.*, 2016, **55**, pp 131–143.
[19] Wu, J., Wang, H., Li, N., Yao, P., Huang, Y., Su, Z. and Yu, Y. Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm, *Aerosp. Sci. Technol.*, 2017, **70**, pp 497–510.
[20] Yao, P., Wang, H. and Su, Z. Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs, *Aerosp. Sci. Technol.*, 2016, **54**, pp 10–22.
[21] Li, B., Song, C., Bai, S., Huang, J., Ma, R., Wan, K. and Neretin, E. Multi-UAV trajectory planning during cooperative tracking based on a Fusion algorithm integrating MPC and standoff, *Drones*, 2023, **7**, (3), p 196.
[22] Olcay, E., Meeß, H. and Elger, G. Dynamic obstacle avoidance for UAVs using MPC and GP-based motion forecast, 2024 European Control Conference (ECC), IEEE, 2024, pp 1024–1031.
[23] Schwarm, A.T. and Nikolaou, M. Chance-constrained model predictive control, *AIChE J.*, 1999, **45**, (8), pp 1743–1752.
[24] Wakabayashi, T., Nunoya, Y. and Suzuki, S. Dynamic obstacle avoidance of multi-rotor UAV using chance constrained MPC, 2021 21st International Conference on Control, Automation and Systems (ICCAS), IEEE, 2021, pp. 412–417.
[25] Zhang, X., Ma, J., Cheng, Z., Huang, S., Ge, S.S. and Lee, T.H. Trajectory generation by chance-constrained nonlinear MPC with probabilistic prediction, *IEEE Trans. Cybern.*, 2020, **51**, (7), pp 3616–3629.
[26] Zhu, H. and Alonso-Mora, J. Chance-constrained collision avoidance for MAVs in dynamic environments, *IEEE Rob. Autom. Lett.*, 2019, **4**, (2), pp 776–783.
[27] Wakabayashi, T., Suzuki, Y. and Suzuki, S. Dynamic obstacle avoidance for multi-rotor UAV using chance-constraints based on obstacle velocity, *Rob. Auton. Syst.*, 2023, **160**, p 104320.
[28] Wang, Y., Li, K., Han, Y. and Yan, X. Distributed multi-UAV cooperation for dynamic target tracking optimized by an SAQPSO algorithm, *ISA Trans.*, 2022, **129**, pp 230–242.
[29] Tyagi, P., Kumar, Y. and Sujit, P.B. NMPC-based UAV 3D target tracking in the presence of obstacles and visibility constraints, 2021 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2021, pp. 858–867.
[30] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Software*, 2017, **114**, pp 163–191.
[31] Haupt, R. and Haupt, S. *Practical Genetic Algorithms*, John Wiley and Sons, 2004, New York, pp 38–39.

Cite this article: Chai X., Lu Y., Liu Z., Yang H., Liu P. and Yan L. (2026). UAV tracking one moving target in an oilfield environment with dynamic obstacles based on an improved salp swarm algorithm. *The Aeronautical Journal*, **130**, 1497–1517. https://doi.org/10.1017/aer.2026.10127

https://doi.org/10.1017/aer.2026.10127 Published online by Cambridge University Press