TRACKING CONTROL METHOD FOR AERIAL MAINTENANCE INSPECTION ROBOTS BASED ON ACO AND DYNAMIC ANALYSIS

Huihong Zheng
Department of Food, MinBei Vocational and Technical College, Nanping, 353000, China

Abstract - Path planning and tracking control are key technologies in the field of robotics, and they are widely applied in areas such as intelligent security. Traditional methods for path planning and tracking control have limitations, including low planning efficiency and poor adaptability to dynamic environments. To address these issues, this paper puts forward a tracking control model that integrates Differential Evolution-optimized Maximum Minimum Ant Colony Optimization with an Extended State Observer and Dynamic Analysis. The model performs path optimization through the improved ant colony algorithm and applies the observer to compensate for disturbances. The experimental results show that this algorithm is significantly superior to the benchmark method in terms of optimization accuracy and convergence speed. Furthermore, in the square wave interference estimation test, the deviation between the estimated value and the actual value of this model is the smallest, which is significantly superior to the performance of the particle swarm optimization - genetic algorithm - proportional-integral-derivative model. These results demonstrate that the proposed model achieves high-precision path planning and tracking control under complex scenarios. This study contributes to enhancing the robot's ability in path optimization and tracking regulation, supporting further innovations in the field of intelligent systems.

Keywords: Maximum minimum ant colony optimization, Differential evolution, Robot, Dynamic analysis, Extended state observer.

1. Introduction

With the rapid development of science and technology, robots are increasingly integrated into various production and daily life scenarios. On one hand, the types of robots are becoming more diverse, covering a wide range of functions and applications. On the other hand, their level of intelligence continues to improve [1]. Maintenance inspection robots, with their precise control, efficient operation, and quick response, play important roles in fields such as military applications, clinical treatment, and healthcare services [2]. Path planning and motion control are core components of autonomous robot control and have always been a major focus in robotics research [3]. Traditional methods such as Rapidly-exploring Random Tree and Artificial Potential Field work well in simple environments, but they often suffer from low planning efficiency, weak obstacle avoidance for dynamic objects, and poor adaptability to complex environments in real-world applications [4]. The Maximum Minimum Ant Colony Optimization (MMACO) algorithm improves search capabilities and reduces convergence time, allowing mobile robots to find optimal paths from the start point to the goal point [5]. The Differential Evolution (DE) algorithm is a population-based evolutionary search method that solves high-dimensional and complex optimization problems [6]. Therefore, this paper combines MMACO and DE to construct a path planning algorithm, named DE-MMACO. Then based on DE-MMACO and Dynamic Analysis (DA), it proposes a tracking control model for aerial maintenance inspection robots, named DME-DA. This model is expected to enhance the robot's autonomous navigation and decision-making capabilities, enabling intelligent applications across different task environments. The novelty of this work lies in three aspects: (1) the hybrid DE-MMACO algorithm, which leverages DE's global search capability to prevent MMACO from falling into local optima, a common issue in traditional ACO variants; (2) the tight integration of the optimized path planner with a dynamics-aware tracking controller that incorporates an Extended State Observer (ESO) for real-time disturbance estimation and compensation, which is often treated separately in existing literature; and (3) the holistic solution tailored for

https://doi.org/10.17683/ijomam/issue22.v2.25

aerial maintenance robots operating in complex 3D environments, addressing both planning and control challenges simultaneously. This model is expected to enhance the robot's autonomous navigation and decision-making capabilities, enabling intelligent applications across different task environments.

2. Related Works

To address the low efficiency of path planning algorithms in complex environments, scholars at home and abroad carried out extensive research on robot path planning and put forward a series of improved methods. Ihsan A et al. raised an improved Ant Colony Optimization (ACO) to solve navigation problems in complex environments. Their study showed that Path 2 identified the shortest route, with a total length of 213 cm, and the robot took 14 to 17 seconds to complete the path [7]. Jing Niu et al. aimed to solve misjudgment and delay in obstacle avoidance, and proposed a wheeled plant-protection robot path planning method based on ACO and Dynamic Window Approach. Experiments demonstrated that this method, by collecting orchard information through LiDAR, enhanced the robot's ability to pass through orchards and improved operational efficiency [8]. Wu P et al. focused on path conflicts in warehouse robots and proposed a path planning model that combined static and dynamic planning strategies. Simulation results indicated that the model outperformed comparative methods in terms of blocked node count and average travel distance [9]. Morin M's group worked on path optimization for search and rescue missions and introduced a moving target search path planning method based on ACO. Experimental results revealed that this method had higher target discovery probability and better computational efficiency compared to traditional greedy algorithms and mixed-integer programming solvers [10]. Zhang D et al. dealt with multi-objective path planning in nuclear accident scenarios and raised an ACO-A* algorithm. Their study demonstrated that the algorithm had significant advantages in stability and in minimizing the overall cost of multi-objective optimization [11].

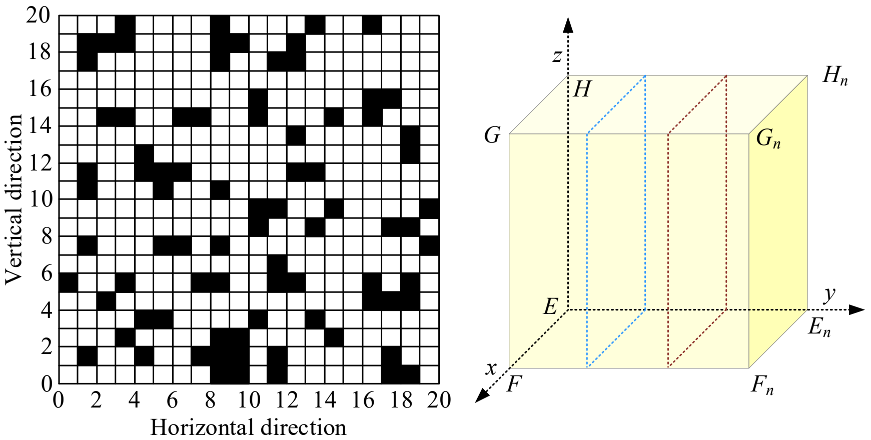
With the rapid growth of intelligent technologies, there has been a rising demand for mobile robot tracking control methods, especially for real-time response and trajectory tracking in complex environments. In response, many researchers carried out in-depth studies in this area. Yu X et al. addressed the challenge of flexible joint robots interacting with unknown environments and proposed a force tracking control scheme that compensated for uncertainties in dynamics by designing an outer force loop and an inner position loop. Simulation experiments confirmed the stability and effectiveness of the scheme [12]. Lu K's team studied issues in multi-robot systems such as unknown input disturbances and uncertain dynamics, and proposed a time-varying region tracking controller based on adaptive sliding mode control. Their results showed that the controller offered high tracking accuracy and strong disturbance rejection [13]. Ahmed S et al. aimed to control and stabilize robots with nonlinear uncertain disturbances and put forward an adaptive sliding mode controller. Their findings indicated that this controller performed better than finite-time sliding mode controllers [14]. Son J's team focused on robot instability caused by varying load environments and proposed a time-delay control strategy. Experimental results showed that, compared to disturbance observers, the time-delay control had better performance in handling complex control problems [15]. Van M and his team raised an adaptive fuzzy backstepping control strategy to address the dependency of fault-tolerant control on dynamic knowledge. Experiments on the PUMA560 robot showed that this strategy allowed tracking errors to converge within a certain time, even without complete dynamic knowledge [16].

Although the existing hybrid algorithms (such as ACO-NN and PSO-GA-PID) have improved path planning to a certain extent, they often have problems of slow convergence speed and poor adaptability in dynamic environments. In contrast, the proposed DE-MACO algorithm introduces a differential evolution mechanism to enhance the global search ability of MMACO, effectively avoiding local optimal solutions and accelerating convergence. In addition, the integration of ESO and dynamic analysis enables real-time interference compensation, which is rarely involved in traditional tracking control models. The model is expected to improve the efficiency of path planning and the precision of tracking control in complex environments, promoting refined development of robots in fields such as industrial manufacturing and intelligent services.

3. Design of Robot Tracking Control Model based on DE-MMACO

3.1 Hybrid Algorithm Design based on DE and MMACO

As the complexity of robot application scenarios continues to increase, path planning algorithms face challenges in both real-time performance and accuracy control [17]. Traditional algorithms lack the ability to respond dynamically to changes in the environment and fail to find optimal paths in complex scenarios. Therefore, this study raises a path planning algorithm based on DE-MMACO to help robots plan motion trajectories under complicated conditions. Before using MMACO to plan paths, it is necessary to build a map model. Among common map modeling methods, the grid method fits well with various path planning algorithms and is commonly used for aerial maintenance inspection robots [18]. The 2D and 3D grid maps are shown in Figure 1.



(a) 2D planar grid

(b) 3D spatial grid method

Figure 1: Schematic diagram of 2D plane and 3D space grid map

As shown in Figure 1(a), the white grids represent free areas where movement is allowed, while the black grids indicate obstacles. Each grid is the minimum movement unit of the aerial maintenance inspection robot. As shown in Figure 1(b), the spatial coordinate system uses E as the origin, EF as the x-axis, EEn as the y-axis, and EH as the z-axis. The space EFGH-EnFnGnHn models the robot's working environment. Based on the dimensions of the robot and obstacles, the study divides EEn into equal segments. Then, it draws parallel planes through these division points to construct a 3D map. The coordinates (x,y,z) represent points in the 3D space. After map modeling, the MMACO algorithm is used for path planning. Suppose there are M ants. Ant k moves from grid point i to j. The transition probability depends on the pheromone level along the path, as shown in Equation (1).

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum\limits_{s\subset d_k}\left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]^{\beta}}, j\in d_k \\ 0, otherwise\end{cases} \tag{1}$$

In Equation (1), the pheromone factor α reflects the importance of the pheromone on edge (i,j). The heuristic factor β evaluates the attractiveness of grid point j to i. Let k be an element in ant d_k's candidate grid set s. At time t, the pheromone intensity on edge (i,j) is denoted by $\tau_{ij}(t)$, and $\eta_{ij}(t)$ represents the heuristic function of the robot at time t. The Euclidean distance and heuristic function are calculated as shown in Equation (2).

$$\begin{cases}d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \\ \eta_{ij}=\dfrac{1}{d_{ij}}\end{cases} \tag{2}$$

In Equation (2), $i(X_i,X_j)$ represents the Euclidean distance between grid points $i(x_i,y_i)$ and $j(x_j,y_j)$. After all ants finish their route search, the algorithm updates the pheromone levels. The updated amount is given in Equation (3).

$$\begin{cases}\tau_{ij}(t+1)=(1+\delta)\tau_{ij}(t)+\Delta\tau_{ij} \\ \Delta\tau_{ij}=\sum\limits_{k=1}^{M}\Delta\tau_{ij}^{k}\end{cases} \tag{3}$$

In Equation (3), $\Delta\tau_{ij}$ and $\delta\in(0,1)$ represent the pheromone increment and evaporation coefficient, respectively. A larger δ value results in faster evaporation. τ_{ij}^{k} is the pheromone released by ant k on edge (i,j). This study uses the ant-cycle model to update pheromone levels, as shown in Equation (4).

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, tour(i,j)\in tour_k \\ 0, otherwise\end{cases} \tag{4}$$

In Equation (4), k is the total path length of ant L_k. $tour_k$ is the set of grid points searched in this round. $tour(i,j)$ represents the path from grid point i to j. The pheromone enhancement coefficient is denoted by Q. A higher Q leads to a more significant influence on the pheromone.

MMACO selects the ant with the highest fitness to update the path. It tracks the maximum and minimum pheromone trajectories and updates the final pheromone trails. The number of pheromone trails is calculated as shown in Equation (5).

$$\rho_{i,j}(t)=\begin{cases}\rho_{\max} & ,\rho_{i,j}(t)\geq\rho_{\min}\\ \rho_{i,j}(t) & ,\rho_{\max}\geq\rho_{i,j}(t)\geq\rho_{\min}\\ \rho_{\min} & ,\rho_{\min}>\rho_{i,j}(t)\end{cases}\qquad(5)$$

In Equation (5), $\rho_{i,j}(t)$, $\rho_{\min}$, and $\rho_{\min}$ represent the number of final, minimum, and maximum pheromone trails, respectively. MMACO performs well in path optimization, but its performance is limited in continuous optimization scenarios.

Therefore, this study applies the DE algorithm to handle continuous optimization. The process of DE is shown in Figure 2.

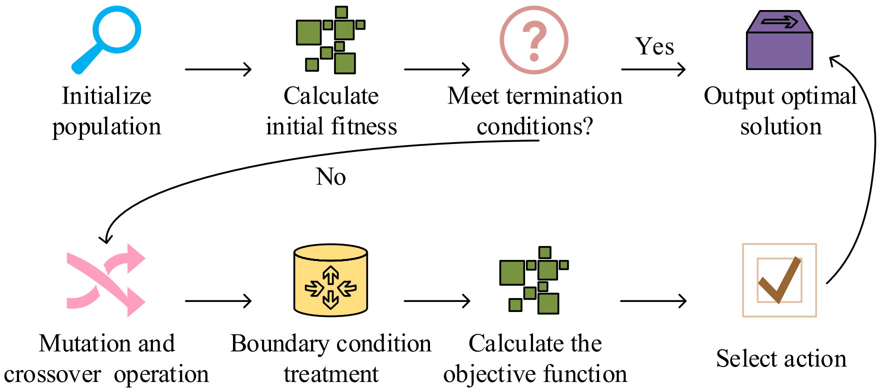

Initialize population
Calculate initial fitness
Meet termination conditions?
Yes
Output optimal solution
No
Mutation and crossover operation
Boundary condition treatment
Calculate the objective function
Select action

Figure 2: Schematic diagram of DE workflow

As shown in Figure 2, DE first initializes the ant colony, then checks whether the crossover probability meets the termination condition. If it does, the algorithm ends and outputs the result. If not, it performs mutation by disturbing individuals in the colony to introduce new directions and diversity. Then, it conducts crossover to replace the original individuals with fitter ones. Next, it processes boundary conditions to ensure individuals stay within the feasible domain and calculates the objective function to determine the next generation. These steps repeat until the algorithm meets the termination condition. The optimization process of DE is shown in Equation (6).

$$\rho_{i,j}(t)=\begin{cases}\rho^{t+1}_{i,j} & ,t\leq P_{cr}\\ \rho^{t}_{i,j} & ,t>P_{cr}\end{cases}\qquad(6)$$

In Equation (6), P_{cr} represents the crossover probability, reflecting the likelihood of crossover in the algorithm. $\rho^{t+1}_{i,j}$ represents the number of pheromone trails between grid points i and j. DE-MMACO combines the advantages of both algorithms to perform path search. The process of the DE-MMACO-based path planning algorithm is shown in Figure 3.

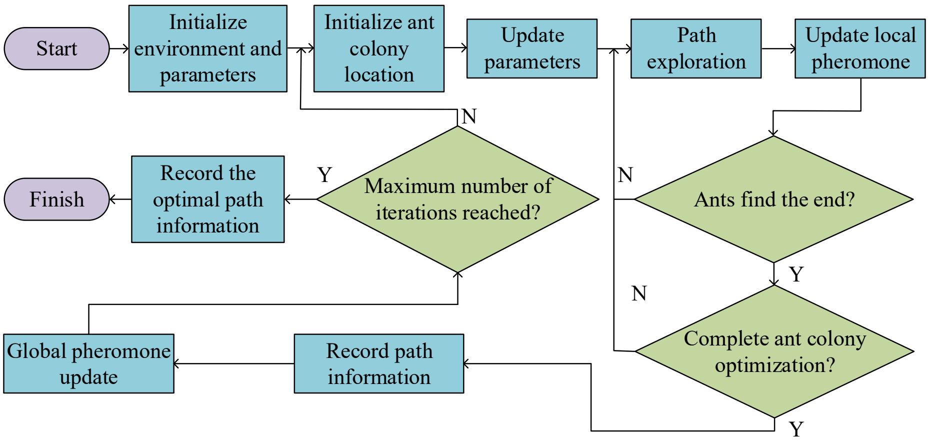

Start
Initialize environment and parameters
Initialize ant colony location
Update parameters
Path exploration
Update local pheromone
N
Finish
Record the optimal path information
Y
Maximum number of iterations reached?
N
Ants find the end?
Y
N
Complete ant colony optimization?
Global pheromone update
Record path information
Y

Figure 3: Path planning algorithm operation flow chart based on DE-MMACO

As shown in Figure 3, the algorithm first builds the grid map model of the working environment, sets pheromone values, and defines the maximum number of iterations, initializing the iteration count to zero. Then, it updates relevant factor coefficients based on the iteration count. Ants move based on the transition probability and update local pheromone after reaching the next node. Then, the algorithm checks whether each ant reaches the endpoint and whether the colony finishes the path search. If complete, it records the path to find the optimal one. If not, the next ant continues the search. After all ants reach the endpoint, it updates the global pheromone. If the maximum number of iterations is not reached, it reinitializes the ant colony and repeats the process. If the maximum iteration is reached, it uses the fitness function to select the optimal path. In summary, the DE-MMACO-based path planning algorithm effectively avoids local optima and finds the optimal path for aerial maintenance inspection robots with high precision and efficiency, thus improving task execution performance.

2.2 Construction of Robot Tracking Control Model Integrating DE-MMACO and DA

DE-MMACO overcomes the issue of traditional algorithms easily falling into local optima by making efficient use of pheromones and adopting a unique search strategy. It achieves good performance in the field of robot tracking control [19]. However, during actual operations, external environmental disturbances often lead to trajectory deviation and positioning errors. The Extended State Observer (ESO) not only reconstructs the model states but also accurately estimates nonlinear disturbances affecting the observation model, which improves the disturbance rejection and control performance of the robot [20]. The dynamic model describes the relationship among the spatial position, instantaneous velocity, and instantaneous acceleration of the robot, and it is used to calculate the control input force or torque at different speeds. Therefore, a mobile robot tracking control model based on ESO-DA is put forward. The structure is shown in Figure 4.

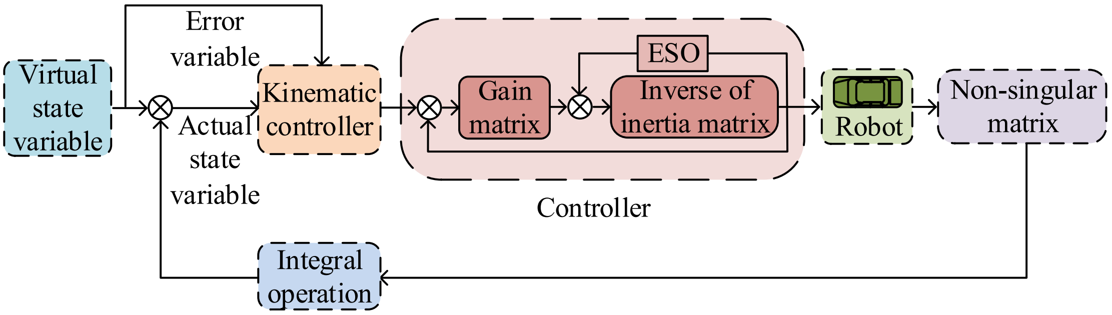


Figure 4: Framework diagram of robot control model based on ESO-DA

As shown in Figure 4, the system compares the expected trajectory with the actual trajectory of the mobile robot to produce an error. After processing, the error is fed into the kinematic controller. The output and feedback form a new error, which is adjusted by gain and applied to the dynamic model of the robot. ESO estimates the state variables and feeds them back. The model runs based on the dynamic model to control the motion states of the mobile robot and completes trajectory tracking. The dynamic equation of the mobile robot is shown in Equation (7).

$$\begin{cases} M(q)\ddot{q}+V_m(q,\dot{q})\dot{q}+F(\dot{q})+G(q)+\varepsilon_d=B(q)\varepsilon-A^T(q)\lambda \\ \dot{q}=S(q)\cdot V(t) \end{cases} \quad (7)$$

In Equation (7), $V(t)$ is the velocity vector and $S(q)$ is a full-rank matrix. q and $\dot{q}$ represent the generalized coordinate vector and velocity vector, respectively. $F(\dot{q})\in R^{n\times 1}$, $G(q)\in R^{n\times 1}$, and ε_d denote the contact surface friction effect, generalized gravity vector, and external disturbance term. $B(q)\in R^{n\times r}$, $M(q)\in R^{n\times n}$, and $A(q)\in R^{m\times n}$ represent the control input matrix, inertia tensor matrix, and constraint Jacobian matrix, while $V_m(q,\dot{q})\in R^{n\times n}$ denotes the centrifugal and Coriolis force matrix. Based on matrix differentiation and the derivation formula of the generalized velocity vector, Equation (7) is optimized by setting d to 0. The dynamic model of the mobile robot is shown in Equation (8).

$$\bar{M}(q)\dot{V}(t)+\bar{V}_m(q,\dot{q})V(t)+\bar{F}(v)+\bar{\varepsilon}_d=\bar{\varepsilon} \quad (8)$$

As shown in Equation (8), v represents a component of the velocity vector and $\dot{V}(t)$ is the derivative of the velocity vector. $\bar{\varepsilon}_d$ and $\bar{F}(v)$ denote the optimized external disturbance and contact surface friction. $\bar{V}_m(q,\dot{q})$ represents the optimized centrifugal and Coriolis force matrix, and $\bar{M}(q)$ and $\bar{\varepsilon}$ represent the optimized inertia tensor and control input matrix. ESO updates the model state variables and accurately estimates external disturbance terms. Therefore, this study introduces ESO with the dynamic model to achieve accurate tracking control of the robot. Let the state variable be $x_0=v$, and the ESO is shown in Equation (9).

$$\dot{x}_0=-\bar{M}(q)\left[\bar{F}(v)+\bar{\varepsilon}_d\right]+M^{-1}(q)\bar{\varepsilon} \quad (9)$$

In Equation (9), $\dot{x}_0$ is the first derivative of variable x_0. By defining $u = M^{-1}(q)\bar{\varepsilon}$, intermediate variable $x_t = -\bar{M}(q)\left[\bar{F}(v)+\bar{\varepsilon}_d\right]$, and $h = \dot{x}_t$, the updated ESO is obtained, as shown in Equation (10).

$$\begin{cases} \dot{x}_0 = x_t + u \\ \dot{x}_t = h \end{cases} \tag{10}$$

In Equation (10), u and h are intermediate variables, $\dot{x}_t$ is the first derivative of intermediate variable x_t. The ESO principle is shown in Figure 5.

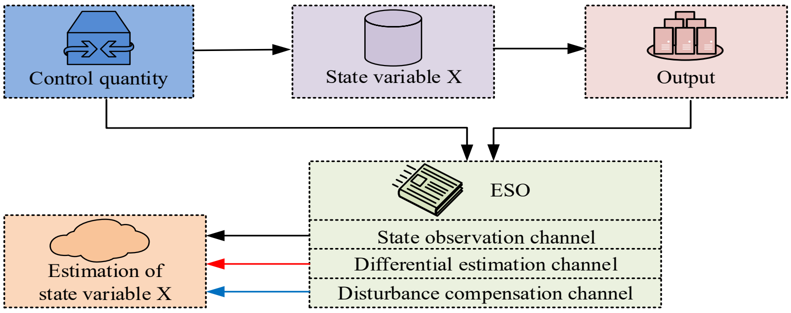

Control quantity
State variable X
Output
ESO
State observation channel
Differential estimation channel
Disturbance compensation channel
Estimation of state variable X

Figure 5: Schematic diagram of ESO principle

As shown in Figure 5, ESO analyzes the control input and output to estimate the actual state variable X, providing more accurate state estimation. The specific calculation of ESO using the error variable is shown in Equation (11).

$$\begin{cases} \dot{\hat{x}}_0 = \hat{x}_t - b_0 r_0 + u \\ \dot{\hat{x}}_t = -b_t fal(r_0, \phi, \gamma) \end{cases} \tag{11}$$

In Equation (11), $\dot{\hat{x}}_0$ is the first derivative of the estimated state variable $\hat{x}_0$. b_0 and b_t are tunable parameters. r_0 represents the error variable related to x_0, and ϕ and γ are parameters of the nonlinear function fal. Based on the kinematic model, the expected values of linear and angular velocities are derived, as shown in Equation (12).

$$\begin{bmatrix} v_{qr} \\ w_{qr} \end{bmatrix} = \begin{bmatrix} v_r \cos e_{th} + k_0 e_0 \\ w_r + k_t v_r e_t - k_{th} v_r \sin e_{th} \end{bmatrix} \tag{12}$$

In Equation (12), v_{qr} and w_{qr} represent the expected values of linear velocity v_r and angular velocity w_r, respectively. k_t, k_0, and k_{th} are positive parameters used to adjust the expected values of linear and angular velocities. e_0, e_{th}, and e_t represent the related error variables. The tracking controller is shown in Equation (13).

$$\varphi = \hat{x}_t + K_f e_c \tag{13}$$

In Equation (13), φ is the output of the tracking controller and e_c is the control error. The controller gain is a tunable parameter represented by K_f. The DME-DA tracking control model for high-altitude inspection robots is shown in Figure 6.

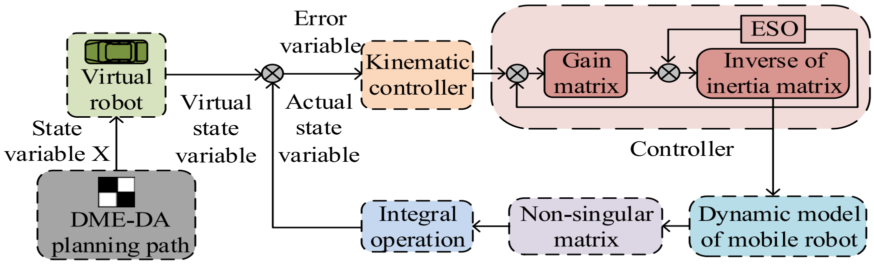


Figure 6: DME-DA high-altitude maintenance robot tracking control model framework

As shown in Figure 6, DE-MMACO searches for the optimal path from the starting point to the target point and determines the robot's motion route. The tracking controller combines DA and ESO, using the DE-MMACO-planned path as the reference. It compares the robot's actual trajectory with the expected one to obtain the error, processes it using the kinematic controller, and combines it with the values estimated by ESO. Then, it controls the robot based on the dynamic model. The integration of the path planning algorithm and dynamic model achieves accurate tracking control, ensuring the robot operates efficiently and stably in complex environments.

4. Performance Analysis of the DE-MMACO-based Tracking Control

Model for Aerial Maintenance Robots

4.1 Validation of the Effectiveness of DE-MMACO

To verify the optimization capability of DE-MMACO, it was compared with the Improved Ant Colony Optimization (IACO) algorithm and the Particle Swarm Optimization-Rapidly-exploring Random Tree (PSO-RRT) algorithm. The simulation experiments were conducted on the MATLAB R2018b platform under a 64-bit Windows 10 operating system, using an Intel Core i5-10300H processor.

Two unimodal test functions, the Schwefel 1.2 function and the Quartic function, were selected for the experiments. The optimal value of the test functions was defined as 0, and the optimization dimension was set to 30. The search ranges were set to [-100, 100] for the Schwefel 1.2 function and [-1.28, 1.28] for the Quartic function.

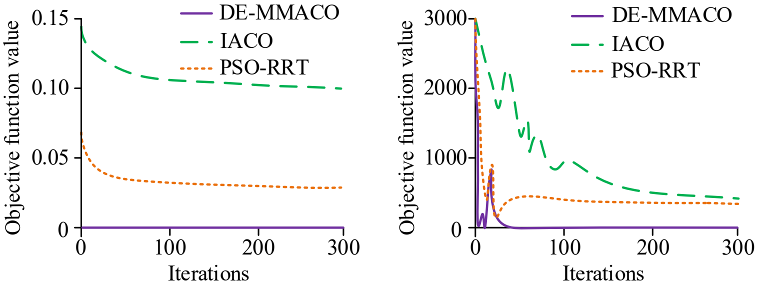
The maximum number of iterations was set to 500, with 50 ants used in the ant colony algorithms, and the values of other parameters with similar meanings were kept consistent. Each of the three algorithms—DE-MMACO, IACO, and PSO-RRT—was run 20 times on the two test functions.

The experimental results are shown in Table 1.

Table. 1 Experimental results on two test functions

Function	Algorithm	Optimal value	Average value	Intermediate value	Worst value
Schwefel 1.2 function	DE-MMACO	3.15E-54	1.33E-49	7.87E-51	8.21E-49
	IACO	3.95E-13	7.75E+00	7.92E-01	6.32E+01
	PSO-RRT	5.96E-03	5.96E-03	5.96E-03	5.96E-03
Quartic function i.e. noise	DE-MMACO	1.12E-01	3.00E-01	3.19E-01	5.41E-01
	IACO	1.64E-01	5.61E-01	5.91E-01	9.98E-01
	PSO-RRT	2.91E-01	1.12E+00	2.93E-01	8.85E+00

As shown in Table 1, when tested on the Schwefel 1.2 function, the best result of DE-MMACO was 3.15E-54, while IACO and PSO-RRT achieved best results of 3.95E-13 and 5.96E-03, respectively. When tested on the Quartic function, which involved noise, DE-MMACO achieved the best value of 1.12E-01, outperforming IACO and PSO-RRT, which reached 1.64E-01 and 2.91E-01, respectively. In terms of average, median, and worst values, DE-MMACO also performed better in most cases. In summary, DE-MMACO showed superior optimization performance compared to IACO and PSO-RRT on both test functions. To better illustrate the convergence of DE-MMACO, the Bent Cigar function and Discus function were used for convergence testing. The convergence curves are shown in Figure 7.

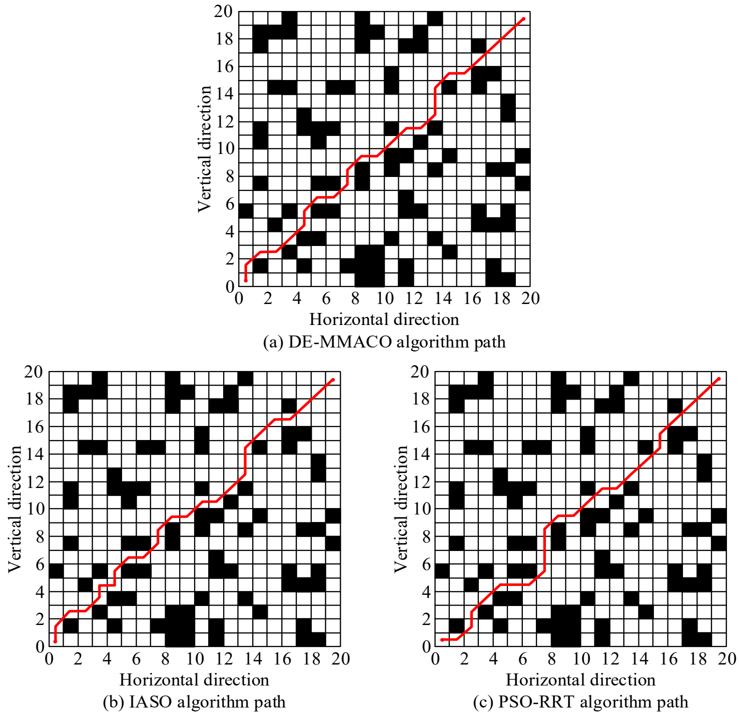


(a) Bent Cigar function convergence curve (b) Discus function convergence curve

Figure 7: Convergence test results of two test functions

As shown in Figure 7(a), when using the Bent Cigar function, the DE-MMACO curve rapidly converged in the early search phase, and the objective function value stabilized at 0. At iteration 50, the objective values of PSO-RRT and IACO dropped to 0.04 and 0.12, respectively. Figure 7(b) shows that, in the Discus function test, DE-MMACO's objective function value approached 0 at iteration 40, while PSO-RRT remained at 400. IACO exhibited significant fluctuations between iterations 10 and 100 and stabilized only after 200 iterations. These results indicated that DE-MMACO achieved faster convergence and higher accuracy under different test functions.

To further demonstrate the path optimization performance of DE-MMACO, the algorithm was tested along with IACO and PSO-RRT in a grid map environment. The optimal paths obtained by the three algorithms through simulation are shown in Figure 8.



(a) DE-MMACO algorithm path

(b) IASO algorithm path

(c) PSO-RRT algorithm path

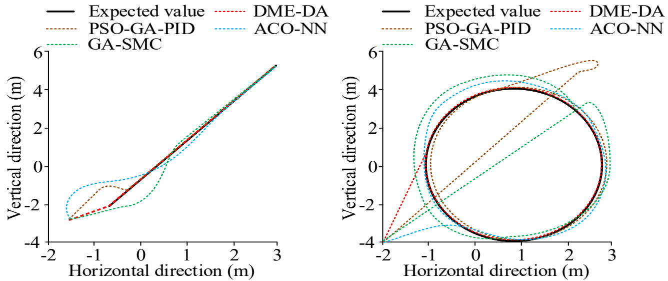
Figure 8: Grid environment simulation test results

As shown in Figure 8(a), DE-MMACO found the global shortest path, which followed a trend close to the straight line between the starting and target points without redundant segments. Figure 8(b) shows that although IACO found a path to the target, it was not globally optimal, and the path length increased by 15% compared to DE-MMACO. Figure 8(c) shows that the path found by PSO-RRT was shorter than that of IACO, but it was still not the shortest. The path length increased by 13% compared to DE-MMACO. In conclusion, DE-MMACO exhibited significant advantages in path planning, precisely identifying the global shortest path and enhancing both the efficiency and quality of the planning process.

4.2 Evaluation and Analysis of the DE-MMACO Robot Tracking Control Model

After validating the performance of the DE-MMACO path planning algorithm, the tracking control model based on DME-DA was evaluated by comparing it with three other models: the Particle Swarm Optimization-Genetic Algorithm-Proportional-Integral-Derivative (PSO-GA-PID), the Ant Colony Optimization-Neural Network (ACO-NN), and the Genetic Algorithm-Sliding Mode Control (GA-SMC).

The simulation experiments were conducted in the MATLAB environment. The main parameters were set as follows: the vehicle mass was 20 kg, the effective radius of the driving wheel was 0.04 m, the lateral wheelbase was 0.25 m, and the rotational inertia of the system was 0.8 kg·m^2. The values of the external disturbance and contact friction were denoted as $\left[2.1\ 1.4\ 2.0\right]^T$ and $\left[5.0\ 3.3\ 4.5\right]^T$, respectively. The observer parameters were set as $b_0 = diag\left\{-160,-160\right\}$ and $b_t = diag\left\{-110,-110\right\}$, while the controller parameters were set as $k_t = 50.0$, $k_0 = 2.8$, $k_{th} = 6.5$, and $K_f = \left\{100,100\right\}$. The four models were tested on both straight and circular trajectories. The results are shown in Figure 9.



(a) Linear trajectory tracking results

(b) Circular trajectory tracking results

Figure 9: Trajectory tracking simulation test results

As shown in Figure 9(a), in the straight-line tracking test, DME-DA quickly approached the reference trajectory at the initial stage. When the horizontal distance reached 0.5m, the vertical deviation between the actual and reference trajectories was 0.1m, indicating high overlap. The PSO-GA-PID and ACO-NN models showed large deviations in the early stage. When the horizontal distance reached 1.5m, their vertical deviation remained at 0.35m. Figure 9(b) shows that in the circular trajectory tracking test, the DME-DA model moved around the circular reference trajectory, maintaining a distance deviation within ±0.2m throughout the process. The GA-SMC model deviated from the circular trajectory with a large fluctuation, and the maximum positional deviation reached 0.72m, which was inferior to the proposed model. In summary, the DME-DA tracking control model outperformed the comparison models in tracking accuracy and stability, reducing energy loss and time waste due to trajectory deviation and improving the efficiency of aerial maintenance tasks. To further evaluate the disturbance estimation capability of the proposed model, it was tested against square wave disturbances along with PSO-GA-PID, ACO-NN, and GA-SMC. The results are shown in Figure 10.

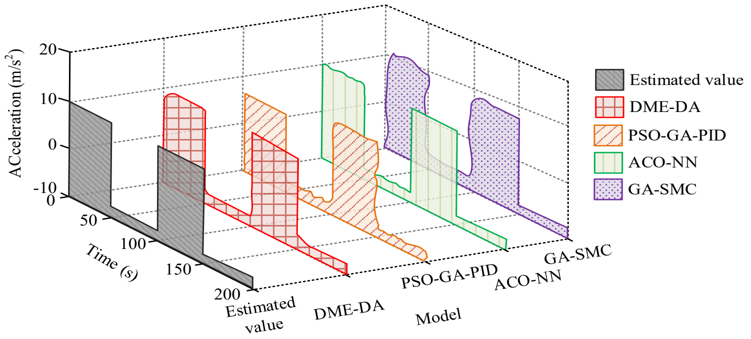
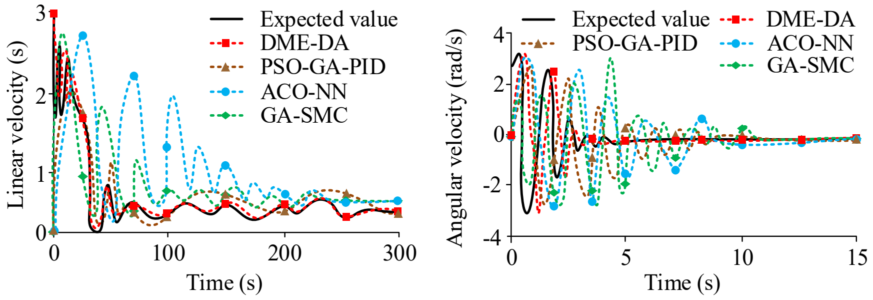


Figure 10: Test results for square wave disturbance estimation

As shown in Figure 10, during the square wave disturbance test, the estimation error of the DME-DA model did not exceed ±0.5 m/s^2. The ACO-NN model fluctuated significantly from 0 to 100s and reached the actual value of 10 m/s^2 after 100s. The PSO-GA-PID model estimated a value of 8 m/s^2 from 0 to 50s, showing a clear deviation from the actual value. The GA-SMC model showed delayed estimation and gradually aligned with the actual value after 150s. In summary, the DME-DA model demonstrated strong anti-disturbance capability and higher stability than the comparison models. To illustrate the convergence performance of the DME-DA model, a comparison of linear and angular velocity tracking was conducted with PSO-GA-PID, ACO-NN, and GA-SMC. The results are shown in Figure 11.



(a) Linear velocity tracking effect (b) Angular velocity tracking effect

Figure 11: Comparison of linear velocity and angular velocity tracking effects

As shown in Figure 11(a), for linear velocity tracking, the DME-DA model rapidly decreased and approached the desired value within 10s. The ACO-NN model consistently deviated from the desired trend and failed to reach the target value. The PSO-GA-PID model exhibited large fluctuations in the early stage and only stabilized at 280s. Figure 11(b) shows that for angular velocity tracking, the DME-DA model converged to the desired value within 4s. The GA-SMC model fluctuated heavily from 1s to 10s and deviated significantly from the desired value. The PSO-GA-PID model had a bumpy convergence process and stabilized at the target value at 10s. Overall, the DME-DA model showed high stability and fast convergence, enabling both linear and angular velocities to quickly stabilize near the desired values. This ensured steady robot operation in complex dynamic environments and reduced the risk of collisions caused by sudden speed changes or external disturbances. To comprehensively evaluate the performance and effectiveness of research methods in practical applications, the study evaluates them from the aspects of computational complexity, stability, and scalability. At the same time, the study repeated the experiment 20 times and conducted statistical tests, and the results are shown in Table 2.

Table 2. The performance effects of different methods in practical applications

Category	Evaluation metric	DME-DA	PSO-GA-PID	ACO-NN	GA-SMC
Anti-interference capability	Square wave disturbance error (m/s ^{2})	0.50±0.012	3.00±0.072	2.20±0.053*	2.20±0.05*
	Error under 20% speed noise (m)	0.08±0.002	0.35±0.008	0.25±0.006*	0.25±0.0004*
	100ms delay overshoot (%)	4.2±0.10	25.3±0.61*	15.2±0.36*	15.2±0.36*
	Success rate of dynamic obstacle replanning (%)	96.0±2.30	82.0±1.97*	87.0±2.09*	87.0±2.09*
Scalability	20x20 grid success rate (%)	100.0±0.0	92.0±2.21*	96.0±2.30*	96.0±2.30*
	50x50 grid success rate (%)	98.5±2.36	85.3±2.05*	89.7±2.15*	89.7±2.15*
	Path length growth rate (%)	5.2±0.12	16.8±0.40*	14.9±0.36*	14.9±0.36*
Computational complexity	Single planning time (ms)	125±3.0	156±3.7*	138±3.3*	138±3.3*
	Memory usage (MB)	45±1.1	52±1.2*	48±1.3*	46±1.1

Note: "*" indicates p<0.05 compared to DME-DA algorithm.

According to Table 2, the DME-DA model demonstrates comprehensive advantages in practical applications. The tracking accuracy test verified the superiority of DME-DA. In the linear tracking task, the maximum position error is controlled within 0.10 ± 0.002m, and the circular trajectory tracking error does not exceed 0.20 ± 0.005m. The accuracy is improved by about 50-70% compared to the comparative model, reflecting the stability and reliability of the control system. The anti-interference ability test results show that DME-DA maintains an estimation error within the range of 0.50 ± 0.012m/s2 when facing square wave disturbances, which is much lower than other comparative methods. In the 20% speed noise test simulating actual working conditions, its tracking error (0.08 ± 0.002m) showed good noise suppression ability. Especially under the condition of 100ms control delay, the overshoot of DME-DA is only 4.2 ± 0.10%, while the PSO-GA-PID model reaches 25.3 ± 0.61%, highlighting the effectiveness of the former in time delay compensation. Scalability evaluation shows that as the complexity of the environment increases, DME-DA still maintains excellent performance. In the 50 × 50 large-scale grid test, the planning success rate reached 98.5 ± 2.36%, and the path length only increased by 5.2 ± 0.12%, significantly better than the compared algorithms. This proves that the powerful global search capability of the DE-MMACO algorithm will not significantly degrade due to the expansion of the problem scale, and has good scalability, laying the foundation for its application in practical large-scale and complex scenarios. In terms of computational efficiency, DME-DA has a single planning time of 125 ± 3.0ms and a memory usage of 45 ± 1.1MB, maintaining reasonable computational overhead while ensuring performance, making it suitable for real-time application scenarios. Comprehensive analysis shows that the DME-DA model has achieved significant improvements in tracking stability, anti-interference ability, computational efficiency, and scalability, providing an effective technical solution for the reliable operation of high-altitude maintenance robots in complex environments.

5. Conclusions

To address the limitations of traditional robot path planning algorithms, such as low planning efficiency, poor adaptability to complex environments, and suboptimal path optimization, this study put forward the DE-MMACO algorithm. Based on this algorithm, a tracking control model for aerial maintenance inspection robots, called DME-DA, was developed. During the model construction, DE-MMACO was used to iteratively optimize path planning by exploring better combinations of path nodes, which improved the accuracy and efficiency of the planning process. At the same time, the DA method was applied to analyze the force characteristics and motion patterns of the robot under complex aerial conditions. It calculated the dynamic parameters of each component and provided accurate control references for the controller.

Experimental results showed that, when tested using the Bent Cigar function, the objective function value of DE-MMACO rapidly dropped to 0 and remained stable. In comparison, the objective values of IACO and PSO-RRT slowly decreased to 0.12 and 0.04 after 50 iterations, respectively, both of which were inferior to DE-MMACO. In addition, in the angular velocity tracking test, the GA-SMC model showed significant deviation from the expected trend between 0 and 10 seconds, while the proposed model converged rapidly within 4 seconds, outperforming the comparison models. Overall, the DME-DA tracking control model for aerial maintenance inspection robots successfully achieved trajectory tracking in aerial operating environments. However, this study did not test the model's stability under different levels of electromagnetic interference, nor did it explore path planning in multi-robot collaboration scenarios. Future work can focus on building electromagnetic interference simulation environments and simulation platforms for multi-robot cooperation.

References

[1] Wang, J., & Ueda, T. (2023). A review study on unmanned aerial vehicle and mobile robot technologies on damage inspection of reinforced concrete structures. *Structural Concrete*, 24(1), 536–562. https://doi.org/10.1002/suco.202200846

[2] Ekren, N., Karagöz, Z., & Şahin, M. (2024). A review of line suspended inspection robots for power transmission lines. *Journal of Electrical Engineering & Technology*, 19(4), 2549–2583. https://doi.org/10.1007/s42835-023-01713-7

[3] Megantoro, P., Abror, A., Syahbani, M. A., Anugrah, A. W., Perkasa, S. D., Setiadi, H., & Vigneshwaran, P. (2023). Autonomous and smart cleaning mobile robot system to improve the maintenance efficiency of solar photovoltaic array. *Bulletin of Electrical Engineering and Informatics*, 12(6), 3288–3297. https://doi.org/10.11591/eei.v12i6.5950

[4] Li, J., Shi, X., Liang, P., Li, Y., Lv, Y., Zhong, M., & Han, Z. (2024). Research on Gait Trajectory Planning of Wall-Climbing Robot Based on Improved PSO Algorithm. *Journal of Bionic Engineering*, 21(4), 1747–1760. https://doi.org/10.1007/s42235-024-00538-y

[5] Chen, Y., Wang, Y., Tang, X., Wu, K., Wu, S., Guo, R., & Dong, E. (2023). Intelligent power distribution live-line operation robot systems based on stereo camera. *High Voltage*, 8(6), 1306–1318. https://doi.org/10.1049/hve2.12349

[6] Jiang, J., Bai, Y. S., Wu, D. H., Yu, Y. X., Ma, X. F., & Lin, C. (2025). Key structure and technology of bridge cable maintenance robot-a review. *Robotic Intelligence and Automation*, 45(1), 121–143. https://doi.org/10.1108/RIA-07-2024-0147

[7] Ihsan, A., Adlie, T. A., & Harliansyah, S. (2024). Optimalisasi Pencarian Jalur Terpendek Mobile Robot dengan Menggunakan Metode Ant Colony Optimization (ACO). *Techné: Jurnal Ilmiah Elektroteknika*, 23(1), 39–54. https://doi.org/10.31358/techne.v23i1.389

[8] Niu, J., Shen, C., Zhang, L., Li, Q., & Liu, S. (2024). Research on obstacle avoidance path of wheeled plant protection robot based on improved ACO-DWA algorithm. *Journal of Electronic Measurement and Instrumentation*, 38(5), 188–200.

[9] Wu, P., Zhong, L., Xiong, J., Zeng, Y., & Pei, M. (2023). Two-level vehicle path planning model for multi-warehouse robots with conflict solution strategies and improved ACO. *Journal of Intelligent and Connected Vehicles*, 6(2), 102–112. https://doi.org/10.26599/JICV.2023.9210011

[10] Morin, M., Abi-Zeid, I., & Quimper, C. G. (2023). Ant colony optimization for path planning in search and rescue operations. *European Journal of Operational Research*, 305(1), 53–63. https://doi.org/10.1016/j.ejor.2022.06.019

[11] Zhang, D., Luo, R., Yin, Y. B., & Zou, S. L. (2023). Multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colony optimization with modified A∗. *Nuclear Engineering and Technology*, 55(5), 1838–1854. https://doi.org/10.1016/j.net.2023.02.005

[12] Yu, X., Liu, S., Zhang, S., He, W., & Huang, H. (2023). Adaptive neural network force tracking control of flexible joint robot with an uncertain environment. *IEEE Transactions on Industrial Electronics*, 71(6), 5941–5949. https://doi.org/10.1109/TIE.2023.3290252

[13] Lu, K., Han, S., Yang, J., & Yu, H. (2023). Inverse optimal adaptive tracking control of robotic manipulators driven by compliant actuators. *IEEE Transactions on Industrial Electronics*, 71(6), 6139–6149. https://doi.org/10.1109/TIE.2023.3296831

[14] Ahmed, S., & Azar, A. T. (2024). Adaptive fractional tracking control of robotic manipulator using fixed-time method. *Complex & Intelligent Systems*, 10(1), 369–382. https://doi.org/10.1007/s40747-023-01164-7

[15] Son, J., Kang, H., & Kang, S. H. (2023). A review on robust control of robot manipulators for future manufacturing. *International Journal of Precision Engineering and Manufacturing*, 24(6), 1083–1102. https://doi.org/10.1007/s12541-023-00812-9

[16] Van, M., Sun, Y., Mcllvanna, S., Nguyen, M. N., Khyam, M. O., & Ceglarek, D. (2023). Adaptive fuzzy fault tolerant control for robot manipulators with fixed-time convergence. *IEEE Transactions on Fuzzy Systems*, 31(9), 3210–3219. https://doi.org/10.1109/TFUZZ.2023.3247693

[17] Daniyan, I., Mpofu, K., & Nwankwo, S. (2023). Design of a robot for inspection and diagnostic operations of rail track facilities. *International Journal of Quality & Reliability Management*, 40(3), 653–673. https://doi.org/10.1108/IJQRM-03-2020-0083

[18] Hou, F., Liu, C., Jiang, H., Tang, Z., Fang, P., & Wang, S. (2024). Structural design and control strategy of a cable-driven robot under high-altitude facade conditions with large span and multiple constraints. *Industrial Robot: the international journal of robotics research and application*, 51(6), 947–959. https://doi.org/10.1108/IR-03-2024-0097

[19] Zhang, S., Zhai, D. H., Lin, J., Xiong, Y., Xia, Y., & Wei, M. (2024). ESO-based safety-critical control for robotic systems with unmeasured velocity and input delay. *IEEE Transactions on Industrial Electronics*, 71(10), 13053–13063. https://doi.org/10.1109/TIE.2024.3349592

[20] Massou, S. (2023). Optimized robust control based ACO technique for two links robot. *Journal of Smart Cities and Society*, 2(3), 141–149. https://doi.org/10.3233/SCS-230011