

Robust Event-Triggered MPC with Parallel Optimization and Disturbance Rejection for Mobile Robots

Hejia Gao¹², Yanling Guo¹², Puhai Yang¹², Changyin Sun^{* 123}

ABSTRACT

This paper proposes an event-triggered parallel optimization model predictive control (MPC) strategy integrated with an extended state observer (ESO) for mobile robots operating under unknown disturbances. In the proposed framework, a conventional MPC formulation is employed to minimize the tracking error and control increments, thereby generating real-time control outputs. To improve control performance and computational efficiency, a parallel optimization strategy is introduced to optimize MPC parameters by jointly considering the accumulated tracking error, state convergence time, and computational cost. An improved Cuckoo Catfish Optimizer (ICCO) is adopted to perform the parallel parameter optimization in real time, enabling adaptive tuning of the MPC parameters. Furthermore, to address unknown external disturbances and model uncertainties, a nonlinear ESO is designed to estimate and compensate disturbances, thereby enhancing the robustness and tracking accuracy of the system. The effectiveness of the proposed control strategy is validated through experimental studies conducted on the MR500 mobile robot platform. The comparative experimental results demonstrate that the proposed method achieves an approximately 40% reduction in tracking error compared with the conventional MPC, while also exhibiting faster response speed, higher tracking accuracy, and stronger robustness against both internal and external disturbances.

KEYWORDS

Mobile Robots, Model Predictive Control, Cuckoo Catfish Optimizer, Extended State Observer, Robust Control

1 Introduction

Trajectory tracking control of mobile robots has attracted significant attention over the past decades due to its fundamental role in a wide range of applications, including autonomous navigation, logistics automation, service robotics, and intelligent transportation systems. The objective of trajectory tracking is to ensure that a mobile robot accurately follows a predefined reference trajectory while satisfying system constraints and maintaining robustness against uncertainties and disturbances. However, the inherent nonlinear dynamics, nonholonomic constraints, and susceptibility to external disturbances make high-precision trajectory tracking a challenging control problem [1, 2, 3, 4, 5].

Model Predictive Control (MPC) has emerged as a powerful control framework for mobile robot trajectory tracking owing to its ability to explicitly handle system constraints and optimize control performance over a finite prediction horizon [6, 7, 8, 9]. Sabitha et al. proposed a ROS-based trajectory tracking method for autonomous tracked vehicles via optimized backstepping and sliding mode control [10]. A robust MPC strategy with a finite time horizon is proposed in [11] for coordinated

path tracking and direct yaw moment control in autonomous independent-drive vehicles, effectively addressing parametric uncertainties and time-varying conditions to enhance stability and accuracy. Article [12] presents a combined MPC and Active Disturbance Rejection Control (ADRC) methodology incorporating radial basis function neural network-based parameter optimization for emergency obstacle avoidance in commercial vehicles, achieving superior tracking precision and effective disturbance rejection [13]. A robust optimal parallel tracking control framework has been proposed for continuous-time nonlinear systems with uncertainties, combining virtual control design and ADP-based neural solutions of HJB equations [14]. Xiong et al. proposed an ESO-based ADRC strategy for autonomous vehicle speed tracking, incorporating a mean value engine model (MVEM) to enhance performance [15]. Composite inner?outer-loop control combining tube-based MPC and ADRC has been proposed for trajectory tracking of mobile vehicles under disturbances and system constraints [16]. Zhu et al. presented a composite control strategy integrating adaptive MPC and improved ADRC, optimizing parameters via an immune genetic algorithm for robust path tracking of intelligent vehicles [17]. To address random packet

1 School of Artificial Intelligence and Anhui Provincial Key Laboratory of Security Artificial Intelligence, Anhui University, Anhui 230601, China

2 Engineering Research Center of Autonomous Unmanned System Technology, Ministry of Education, Hefei, Anhui 230601, China

3 School of Automation, Southeast University, Nanjing 210096, China

* Corresponding author. Email: cysun@seu.edu.cn

dropouts in sensor-controller and controller-actuator channels, robust MPC schemes integrating switched-system observers and tightened constraints have been proposed [18]. To overcome limitations of traditional controllers, MPC schemes augmented with PINN models have been developed for complex trajectory tracking tasks in noisy unmanned driving scenarios [19]. A learning-based ET-MPC framework incorporating Gaussian Process (GP) models was proposed to capture unknown system dynamics and uncertainties. By embedding probabilistic learning into the prediction model and introducing terminal constraints, this method improves robustness and guarantees stability under model uncertainty. However, such approaches typically suffer from high computational complexity due to GP inference and kernel matrix operations, which may limit their real-time applicability in embedded platforms [20]. Mohsen Farbood reviews the applications of MPC in large-scale wind power integration. It highlights how MPC improves power, frequency, and voltage control by handling uncertainties and system constraints. The study discusses multi-level control strategies, optimization methods, and future developments combining MPC with artificial intelligence and energy storage technologies [21]. Peng Lu proposes a fuzzy-based MPC strategy for hydraulic wind turbine drivetrains. By combining T-S fuzzy modeling, quadratic programming, and LMI-based terminal constraints, the method reduces computational complexity and improves real-time feasibility. Simulation results show better speed regulation, lower control effort, and faster response compared with conventional controllers [22].

Recent advancements in metaheuristic optimization have led to the development of nature-inspired algorithms, such as the Artificial Protozoa Optimizer (APO), which emulates the foraging and reproductive strategies of protozoa [23, 24]. Similarly, the Beluga Whale Optimization (BWO) draws from the social and predatory behaviors of beluga whales, demonstrating efficiency in handling multimodal functions [25]. These algorithms have been validated through extensive benchmarks and practical engineering applications, highlighting their robustness and versatility. Dung beetles' diverse behaviors motivated Xue and Shen to create Dung beetle optimizer (DBO), a competitive meta-heuristic that excels in balancing exploration and exploitation for global optimization [26]. Han et al. introduced Walrus optimizer (WO), derived from walruses' social actions, which performs robustly in solving real-world optimization problems [27]. Çimen and Yalçın proposed a hybrid firefly-whale optimization algorithm (FA-WOA) to optimize MPC parameters [28]. Al-Betar et al. introduced the elk herd optimizer (EHO) as a novel nature-inspired metaheuristic [29], and the parrot optimizer (PO) has been successfully applied to address various medical-related optimization problems [30]. Yu et al. proposed a genetic algorithm (GA)-based real-time weight optimization method for nonlinear model predictive control (NMPC) to enhance ship trajectory tracking [31].

Motivated by the above observations, this paper proposes a parallel optimization MPC (MPC-PO) framework that combines an event-triggered parallel optimization strategy with an ESO-based disturbance compensation mechanism. A conventional MPC is used to generate control inputs, while a parallel objective function enables adaptive parameter tuning under real-time constraints. The resulting high-dimensional optimization problem is efficiently solved using an Improved

Cuckoo Catfish Optimizer (ICCO). Furthermore, a nonlinear ESO is incorporated to estimate and compensate disturbances and modeling uncertainties. Experimental results on the MR500 mobile robot platform demonstrate enhanced transient response and improved disturbance rejection performance. Fig. 1 shows the control strategy overall structure proposed in this paper. The main contributions of this paper can be summarized as follows:

1. Unlike traditional MPC tuning that relies on offline or time-triggered updates, this study proposes a parallel objective function designed specifically for real-time deployment. By jointly optimizing the accumulated tracking error, convergence time, and computational cost, it effectively resolves the inherent conflict between maintaining high control performance and satisfying strict real-time feasibility constraints during parameter adaptation.
2. To overcome the slow convergence and limited accuracy of conventional meta-heuristic algorithms in high-dimensional parameter spaces, an ICCO algorithm is proposed. Uniquely tailored for real-time applications, ICCO guarantees the generation of usable intermediate solutions and supports continuation across t_{max} , ensuring that optimal parameters are reliably delivered within strict computational time limits without interrupting control execution.
3. While ESOs have been widely studied for disturbance rejection, their collaborative combination with real-time MPC optimization regulation strategies remains insufficiently explored. This framework incorporates a nonlinear ESO to actively estimate and compensate for multi-directional external disturbances and modeling uncertainties. Crucially, the ESO enhances the robustness of the optimization process itself, effectively resolving tracking performance degradation caused by dynamic environmental changes during the tuning phase.

The remainder of this paper is organized as follows. Section II presents the kinematic modeling of the mobile robot. Section III introduces the design of the MPC controller and analyzes its stability. Section IV describes the ICCO and the proposed event-triggered parallel optimization MPC framework. Section V presents the design of the nonlinear ESO and provides its stability analysis. Section VI reports the experimental setup and comparative results obtained on the mobile robot platform. Finally, Section VII concludes this paper.

2 Kinematic Model of Mobile Robot

The four-wheel differential-drive mobile robot model is shown in the Fig. 2. XOY is the global coordinate system, X_rCY_r is the local coordinate system, The center of mass of the mobile robot coincides with the center of the two wheel axes at point C , and the pose of the mobile robot under the global coordinate system is represented as $q = [x, y, \theta]^T$, where x and y are the abscissa and ordinate of the robot's center of mass respectively, and θ is the heading angle of the robot [32].

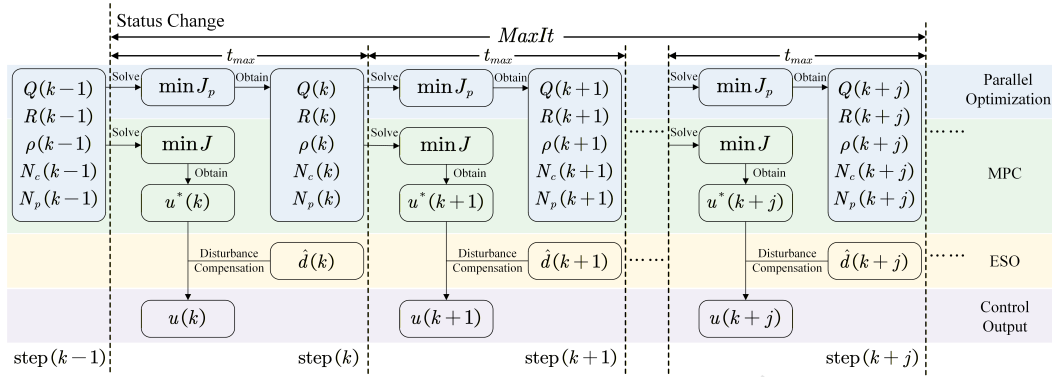


Fig. 1 Control strategy overall structure.

Velocity is $v = [v, \omega]^T$, where v is the forward velocity and ω is the angular velocity. To facilitate the modeling of mobile robots, the following assumptions are made:

Assumption 1. The mobile robot wheels roll on the ground without sliding.

Assumption 2. The mobile robot does not move laterally.

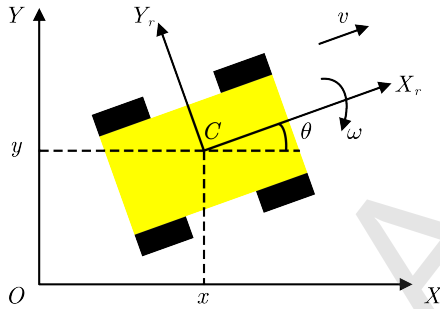


Fig. 2 Four-wheel differential-drive mobile robot model.

When the motion of the mobile robot satisfies the above assumptions, the incomplete constraint of the mobile robot considering that the center of mass and the geometric center do not coincide can be expressed as:

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (1)$$

written in matrix form:

$$A(q)\dot{q} = 0 \quad (2)$$

where $A(q) = [\sin \theta \quad \cos \theta \quad 0]$. Take a set of spatial basis vectors of $A(q)$ as $g_i(q)$, $i = 1, 2$, which represents the change of the mobile robot's motion posture and satisfies $A(q)g_i(q) = 0$, $i = 1, 2$. $S(q) = [g_1 \quad g_2]^T$ is a full rank matrix, which satisfying $S^T(q)A^T(q) = 0$. Therefore, the kinematics model of the mobile robot is:

$$\dot{q} = S(q)u \quad (3)$$

where $S(q) = \begin{bmatrix} \sin \theta & 0 \\ \cos \theta & 0 \\ 0 & 1 \end{bmatrix}$ is the Jacobian matrix, which defines the relative relationship between the attitude and speed of the

mobile robot. $u = [v \quad \omega]^T$ is the control input of the system [33].

when considering the presence of disturbance, the kinematic model is:

$$\dot{q} = S(q)u + d \quad (4)$$

where $d = [d_1 \quad d_2 \quad d_3]^T$ represents unknown disturbance, d_1 , d_2 and d_3 are the disturbances on the three state components. Let $f(q, u) = S(q)u$, and expand equation (4) with the Taylor series near the reference point, ignoring the higher-order terms, to get:

$$\begin{aligned} \dot{q} = & f(q_r, u_r) + \frac{\partial f(q, u)}{\partial x} \bigg|_{q=q_r, u=u_r} (q - q_r) \\ & + \frac{\partial f(q, u)}{\partial u} \bigg|_{q=q_r, u=u_r} (u - u_r) + d \end{aligned} \quad (5)$$

equation (5) can be simplified to:

$$\dot{q} = f(q_r, u_r) + f_{q,r}(x - x_r) + f_{u,r}(u - u_r) + d \quad (6)$$

where, $f_{q,r}$ and $f_{u,r}$ are the Jacobian matrices about q and u respectively. In summary, the kinematic error model of the mobile robot is:

$$\dot{\tilde{q}} = f_{q,r}\tilde{x} + f_{u,r}\tilde{u} + d \quad (7)$$

where $\tilde{q} = q - q_r$ is the difference between the actual pose and the reference pose of the mobile robot, and $\tilde{u} = u - u_r$ is the difference between the actual input and the reference input. In order to design the controller, the error model can be discretized:

$$\tilde{q}(k+1) = A(k)\tilde{q}(k) + B(k)\tilde{u}(k) + D(k)d(k) \quad (8)$$

where $A(k) = \begin{bmatrix} 1 & 0 & -Tv_r(k)\sin\theta_r(k) \\ 0 & 1 & Tv_r(k)\cos\theta_r(k) \\ 0 & 0 & 1 \end{bmatrix}$, $B(k) = \begin{bmatrix} T\cos\theta_r(k) & 0 \\ T\sin\theta_r(k) & 0 \\ 0 & 1 \end{bmatrix}$, $D(k) = \begin{bmatrix} T \\ T \\ 0 \end{bmatrix}$. T is the sampling period and k is the sampling time.

3 Design of Model Predictive Controller

3.1 Design of Controller

The MPC adopts rolling optimization in a limited time domain instead of the global optimization in the optimal control, and the mobile robot will be limited by various constraints in the actual environment, so the control input bundle constraints and control incremental constraints are considered when designing the controller, so the objective function can be designed as:

$$J(k) = \sum_{j=1}^{N_p} [(q(k+j|k) - q_r(k+j|k))^T Q (q(k+j|k) - q_r(k+j|k))] + \sum_{j=0}^{N_c-1} [\Delta \tilde{u}^T(k+j|k) R \cdot \Delta \tilde{u}(k+j|k)] + \rho \epsilon^2 \quad (9)$$

where $q(k+j|k)$ and $q_r(k+j|k)$ represent the output state and the reference output state in the prediction horizon, respectively, and $\Delta \tilde{u}(k+j|k)$ represents the feedback control increment in the control time domain. N_p is the prediction horizon, N_c is the control time domain, and $1 \leq N_c \leq N_p$ is satisfied. ρ is the weight coefficient and ϵ is the relaxation factor. Q and R are weight matrices that satisfy $Q \geq 0$ and $R \geq 0$. The purpose of the relaxation factor is to prevent the objective function from being unsolvable^[34].

Since the control input of equation (9) is the forward speed and angular velocity of the mobile robot, the forward speed and angular velocity increment cannot be accurately controlled, that is, the sudden change of the increment cannot be ensured. Therefore, define a new state variable as follows:

$$\chi(k|k) = \begin{bmatrix} \tilde{q}(k|k) \\ \tilde{u}(k-1|k) \end{bmatrix} \quad (10)$$

equation (10) can then be converted to a state space expression as follows:

$$\begin{cases} \chi(k+1|k) = \tilde{A}(k)\chi(k|k) + \tilde{B}(k)\Delta \tilde{u}(k|k) + \tilde{D}(k)d(k) \\ \eta(k|k) = \tilde{C}\chi(k|k) \end{cases} \quad (11)$$

where $\tilde{A} = \begin{bmatrix} A(k) & B(k) \\ 0_{2 \times 3} & I_2 \end{bmatrix}$, $\tilde{B} = \begin{bmatrix} B(k) \\ I_2 \end{bmatrix}$, $\tilde{D} = \begin{bmatrix} D(k) \\ 0_{2 \times 1} \end{bmatrix}$, $\tilde{C} = [I_3 \quad 0_{3 \times 2}]$, I is the unit vector, $\Delta \tilde{u}(k|k) = \tilde{u}(k|k) - \tilde{u}(k-1|k)$. Ignoring the effects of interference, the error system can be defined as:

$$\begin{cases} \hat{\chi}(k+1|k) = \tilde{A}(k)\hat{\chi}(k|k) + \tilde{B}(k)\Delta \hat{u}(k|k) \\ \hat{\eta}(k|k) = \tilde{C}\hat{\chi}(k|k) \end{cases} \quad (12)$$

When solving the objective function optimization problem, it is necessary to calculate the system output of the system in the prediction time domain. Therefore, the next state quantity can

be obtained iteratively in the N_p :

$$\begin{aligned} \hat{\chi}(k+1|k) &= \tilde{A}(k)\hat{\chi}(k|k) + \tilde{B}(k)\Delta \hat{u}(k|k) \\ \hat{\chi}(k+2|k) &= \tilde{A}(k+1)\hat{\chi}(k+1|k) + \tilde{B}(k+1)\Delta \hat{u}(k+1|k) \\ &= \tilde{A}(k+1)\tilde{A}(k)\hat{\chi}(k|k) + \tilde{A}(k+1)\tilde{B}(k)\Delta \hat{u}(k|k) \\ &\quad + \tilde{B}(k+1)\Delta \hat{u}(k+1|k) \\ &\vdots \\ \hat{\chi}(k+N_p|k) &= \varpi(N_p-1, 0)\hat{\chi}(k|k) \\ &\quad + \sum_{j=1}^{N_p-1} \varpi(N_p-1, j)\tilde{B}(k+j-1)\Delta \hat{u}(k+j-1|k) \\ &\quad + \tilde{B}(k+N_p-1)\Delta \hat{u}(k+N_p-1|k) \end{aligned} \quad (13)$$

where $\varpi(\varrho, i) = \prod_{j=i}^{\varrho} \tilde{A}(k+j)$. Combining equations (12) and (13), the predicted output expression can be obtained:

$$\Psi(k) = \Theta(k)\hat{\chi}(k|k) + \Gamma(k)\Delta U(k) \quad (14)$$

$$\begin{aligned} \text{where } \Psi(k) &= [\hat{\chi}(k+1|k) \quad \hat{\chi}(k+2|k) \cdots \hat{\chi}(k+N_p|k)]^T, \\ \Delta U(k) &= [\Delta \hat{u}(k|k) \quad \Delta \hat{u}(k+1|k) \cdots \Delta \hat{u}(k+N_p-1|k)]^T, \\ \Theta(k) &= [\tilde{C}\tilde{A}(k) \quad \tilde{C}\tilde{A}(k+1)\tilde{A}(k) \cdots \tilde{C}\varpi(N_p-1, 0)]^T, \\ \Gamma(k) &= \begin{bmatrix} \tilde{C}\tilde{B}(k) & 0 & 0 \\ \tilde{C}\tilde{A}(k+1)\tilde{B}(k) & \tilde{C}\tilde{B}(k+1) & 0 \\ \vdots & \ddots & \vdots \\ \tilde{C}\varpi(N_p-1, 1)\tilde{B}(k) & \cdots & \tilde{C}\varpi(N_p-1, N_c) \\ & & \tilde{B}(k+N_c-1) \end{bmatrix}. \end{aligned}$$

For actual systems, the control inputs are often subject to some physical limitations, so these should be taken into account when designing the controller. For the constraints that control the input, the following constraints are designed as: $\tilde{u}_{\min}(k) \leq \tilde{u}(k) \leq \tilde{u}_{\max}(k)$. To avoid abrupt variations in the control output, the following control increment constraints are imposed: $\Delta \tilde{u}_{\min}(k) \leq \Delta \tilde{u}(k) \leq \Delta \tilde{u}_{\max}(k)$. Where $\tilde{u}_{\min}(k)$ and $\tilde{u}_{\max}(k)$ are the lower and upper bounds of the control quantity. $\Delta \tilde{u}_{\min}(k)$ and $\Delta \tilde{u}_{\max}(k)$ are the lower and upper bounds of the feedback control increment. Therefore, the controller can be obtained by minimizing the objective function with constraints.

Substituting equation (10) into (4) can write the objective function as quadratic:

$$\begin{aligned} \min_{\Delta U(k)} J(\hat{\chi}(k), \Delta U(k)) \\ = \frac{1}{2} \hat{\chi}^T(k) G(k) \hat{\chi}(k) + \frac{1}{2} \Delta U^T(k) H(k) \Delta U(k) + F^T(k) \Delta U(k) \end{aligned} \quad (15)$$

which satisfies the following constraints:

$$\begin{aligned} \hat{u}_{\min} \leq \hat{u}(k+j|k) \leq \hat{u}_{\max}, \forall j = 0, \dots, N-1 \\ \Delta \hat{u}_{\min} \leq \Delta \hat{u}(k+j|k) \leq \Delta \hat{u}_{\max}, \forall j = 0, \dots, N-1 \end{aligned} \quad (16)$$

where $G(k) = 2(Q + \Theta^T \tilde{Q} \Theta)$, $H(k) = 2(\Gamma^T Q \Gamma + \tilde{R})$, $F(k) = 2\Gamma^T \tilde{Q} e_x(k|k)$, $\tilde{Q} = \tilde{Q} = \sum_{i=1}^{N_p-1} Q(i)$, $\tilde{R} = \sum_{i=1}^{N_c-1} R(i)$. The variable $\hat{\chi}^T(k) G(k) \hat{\chi}(k)$ is not related to $\Delta U(k)$, so it can be ignored.

Since the variable in the optimization problem equation (15) is the control increment $\Delta U(k)$, the restriction problem in

equation (16) can be converted into an expression related to the control increment $\Delta U(k)$ according to the formula $\tilde{u}(k|k) = \tilde{u}(k-1|k) + \Delta \tilde{u}(k|k)$:

$$\begin{bmatrix} \Lambda \\ -\Lambda \\ \Gamma \\ -\Gamma \end{bmatrix} \Delta U(k) \leq \begin{bmatrix} U_{\max} - U_{k-1} \\ -U_{\min} + U_{k-1} \\ \Delta U_{\max} \\ -\Delta U_{\min} \end{bmatrix} \quad (17)$$

$$\text{where } \Gamma = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{N \times N} \otimes I_{2 \times 2}, U_{k-1} = 1_N \otimes$$

$\tilde{u}(k-1)$, $\otimes$ is the Kronecker product, $I_{2 \times 2}$ is the unit matrix of 2×2 , 1_N represents a column vector where each row is 1 and the number of rows is N , and $U_{\min}$ and $U_{\max}$ represent the set of minimum and maximum values of the control quantity, respectively. $\Delta U_{\min}$ and $\Delta U_{\max}$ represent the set of minimum and maximum values of the control increment, respectively. Let $\hbar = [\Lambda \quad -\Lambda \quad \Gamma \quad -\Gamma]^T$, $\iota = [U_{\max} - U_{k-1} \quad -U_{\min} + U_{k-1} \quad \Delta U_{\max} \quad -\Delta U_{\min}]^T$. The objective function with soft constraints in equation (15) is transformed into the following quadratic planning problem:

$$\begin{aligned} \min_{\Delta U(k)} J(\hat{\chi}(k), \Delta U(k)) \\ = \frac{1}{2} \Delta U^T(k) H(k) \Delta U(k) + F^T(k) \Delta U(k) \end{aligned} \quad (18)$$

which satisfies the constraints: $\hbar \Delta U(k) \leq \iota$. By solving the optimization problem in equation (18), all control increments in the predicted time domain can be obtained:

$$\Delta U^T(k) = [\Delta \hat{u}(k|k), \Delta \hat{u}(k+1|k), \dots, \Delta \hat{u}(k+N_c-1|k)]^T \quad (19)$$

Therefore, the final output $u^*(k)$ of the model predictive controller is:

$$u^*(k) = \Delta \hat{u}(k|k) + \hat{u}(k-1) + u_r(k) \quad (20)$$

3.2 Stability Analysis of Controller

Theorem 1. For the mobile robot model in equation (4) and the objective function in equation (9), if the appropriate positive matrix R , semi-definite matrix Q , the prediction horizon $N_p \geq 1$ and the control horizon $1 \leq N_c \leq N_p$ are given, the objective function has the optimal solution $[\Delta U^*(k) \quad \varepsilon^*(k)]^T$, assume that the optimal solution of the objective function at time k is $J^*(k)$, so let $J^*(k)$ is the Lyapunov function $V^*(k)$, and if the conditions $V^*(k+1) \leq V^*(k)$ are satisfied at time $k+1$, the system can be asymptotically stable.

The system controller consists of a feedforward controller and a feedback controller. The feedforward part is obtained from the desired trajectory. Therefore, the stability of the system mainly depends on the feedback controller component.

The Lyapunov function set according to the optimal solution has the following relationship:

$$\begin{aligned} V^*(k) = \min J(k) = \min \left\{ \sum_{j=1}^{N_p} \|q(k) - q_r(k)\|_Q^2 \right. \\ \left. + \sum_{j=0}^{N_c-1} (\|\Delta u(k)\|_R^2 + \rho \varepsilon^2(k)) \right\} \end{aligned} \quad (21)$$

In the same way, the control input at time $k+1$ is:

$$\begin{aligned} \Delta \tilde{u}(k+1+j|k+1) \\ = [\Delta \tilde{u}(k+1|k+1), \Delta \tilde{u}(k+2|k+1), \dots, \\ \Delta \tilde{u}(k+N_c|k+1)] \\ = [\Delta \tilde{u}^*(k+1|k), \Delta \tilde{u}^*(k+2|k), \dots, \Delta \tilde{u}^*(k+N_c|k)] \end{aligned} \quad (22)$$

$$\varepsilon(k+1) = \varepsilon^*(k+1|k) \quad (23)$$

According to equations (22) and (23), the optimal solution $[\Delta U^*(k+1) \quad \varepsilon^*(k+1)]^T$ at time $k+1$ is a set of feasible solutions to the optimal problem of the objective function, and the control output obtained by the solution is within the range of constraints.

The relationship between $J^*(k+1)$ and $V^*(k)$ is as follows:

$$\begin{aligned} J(k+1) \\ = \sum_{j=1}^{N_p} \|q(k+1+j|k+1) - q_r(k+1+j|k+1)\|_Q^2 + \\ \sum_{j=0}^{N_c-1} (\|\Delta u(k+1+j|k+1)\|_R^2 + \|\varepsilon(k+1+j|k+1)\|_\rho^2) \\ = \sum_{j=1}^{N_p} \|q^*(k+i|k) - q_r(k+i|k)\|_Q^2 \\ + \sum_{j=0}^{N_c-1} (\|\Delta u^*(k+j|k)\|_R^2 + \|\varepsilon^*(k+j|k)\|_\rho^2) \\ - \|q(k|k) - q_r(k|k)\|_Q^2 - \|\Delta u(k|k)\|_R^2 - \|\varepsilon(k|k)\|_\rho^2 \\ = V^*(k) - \|q(k|k) - q_r(k|k)\|_Q^2 - \|\Delta u(k|k)\|_R^2 - \rho \varepsilon^2(k|k) \end{aligned} \quad (24)$$

The optimal objective function value at time $k+1$ is $J(k+1)$, so that $J(k+1)$ is not less than $V^*(k)$, then:

$$\begin{aligned} V^*(k+1) &\leq J(k+1) \\ &\leq V^*(k) - \|q(k|k) - q_r(k|k)\|_Q^2 - \|\Delta u(k|k)\|_R^2 - \rho \varepsilon^2(k|k) \end{aligned} \quad (25)$$

In summary, $V^*(k+1) \leq V^*(k)$, so the system in equations (4) and (9) is stable.

4 Parallel Optimization Model Predictive Control

4.1 Improved Cuckoo Catfish Optimizer

The Cuckoo Catfish Optimizer (CCO) is a meta-heuristic optimization algorithm inspired by the brood parasitism behavior of cuckoo catfish toward mouth-brooding cichlids in Lake Tanganyika. The CCO algorithm abstracts these biological interactions and formulates them as search operators for solving optimization problems [35]. The ICCO is designed based on CCO algorithm to better accommodate the MPC-PO framework, and its procedure is described as follows.

Population Initialization

When the state of the mobile robot changes, certain MPC parameters obtained in the previous step can provide valuable prior information to accelerate the optimization process. To exploit this information while preserving global exploration capability, a hybrid population initialization strategy incorporating prior knowledge is proposed. The parameter value $N_p(k-1)$, $N_c(k-1)$, $Q(k-1)$, $R(k-1)$ and $\rho(k-1)$ of step $k-1$ before the state change of step k is recorded as X_{k-1}^d .

During population initialization, 30% of the individuals are generated locally within a $\pm 20\%$ range around X_{k-1}^d , while the remaining 70% are initialized randomly over the entire search space. The globally initialized individuals $X_{(\text{global})i}^d$ and locally initialized individuals $X_{(\text{local})i}^d$ are generated as follows:

$$\begin{aligned} x_{(\text{global})i}^d &= \text{rand} \times (Ub^d - Lb^d) + Lb^d, \begin{cases} i = 1, 2, \dots, 0.7N \\ d = 1, 2, \dots, D \end{cases} \\ x_{(\text{local})i}^d &= (\text{rand} \times 0.4 + 0.8) X_{k-1}^d, \begin{cases} i = 1, 2, \dots, 0.3N \\ d = 1, 2, \dots, D \end{cases} \end{aligned} \quad (26)$$

where D denotes the dimension of the optimization problem, x_i^d represents the d -th component of the i -th individual, and Ub^d and Lb^d are the upper and lower bounds of the d -th decision variable, respectively. rand is a random number uniformly distributed within $[0, 1]$, and R_i denotes a D -dimensional random vector with each element in $[0, 1]$. Their fitness values are evaluated using the objective function $F(\cdot)$, and the best and worst individuals in the initial population are recorded as X_{best} and X_{worst} , respectively.

Logarithmic Transformation

Since the parameters P , Q , and ρ are strictly positive and span multiple orders of magnitude, directly performing the search in the original parameter space may result in scale inconsistency, thereby degrading the search efficiency of the CCO algorithm. To address this issue, logarithmic mapping is applied to P , Q , and ρ to normalize their scales and enhance optimization efficiency. The transformed variables are defined as: $\tilde{Q} = \lg(Q)$, $\tilde{P} = \lg(P)$, $\tilde{\rho} = \lg(\rho)$. Accordingly, the search space in the logarithmic domain

$$\text{becomes: } \begin{cases} \tilde{Q} \in [\lg(Q_{\min}), \lg(Q_{\max})] \\ \tilde{R} \in [\lg(R_{\min}), \lg(R_{\max})] \\ \tilde{\rho} \in [\lg(\rho_{\min}), \lg(\rho_{\max})] \end{cases} \quad \text{Before each fitness}$$

evaluation, the inverse mapping is performed to recover the original parameters: $Q = \exp(\tilde{Q})$, $R = \exp(\tilde{R})$, $\rho = \exp(\tilde{\rho})$, which are then substituted into the objective function J_p .

Compressed Space Strategy

To facilitate prey capture, the Cuckoo Catfish population collaborates to compress the search area, gradually narrowing the escape or hiding space of the prey. In the early stages of iteration, the i -th individual X_i compresses the prey's movement space by cooperating with multiple random individuals in the population, making full use of spatial information within the encirclement through enhanced intra-population connections,

as described as follow:

$$\begin{aligned} X_i^{\text{new}} &= X_i + Z_1 \times |rd| \times \left(\frac{X_{\text{best}} + X_{r_1}}{2} - X_{r_2} \right) \\ &\quad + \frac{r_3}{2} \times (X_{r_3} - X_{r_4}) \end{aligned} \quad (27)$$

$$X_i^{\text{new}} = X_{r_1} + |rd| \times (X_{\text{best}} - X_i + X_{r_2} - X_{r_3}) \quad (28)$$

where X_i^{new} is the updated position of the i -th individual, X_r denotes a random individual in the population; Z_1 is a random number selected from $\{0, 1\}$; the inter-population cooperation intensity is characterized by $|rd|$, and $rd \sim \mathcal{N}(0, 1)$; r_3 is a random number in $[0, 1]$.

After a period of searching, the population accumulates certain spatial information, enabling more precise compression by reducing the number of random individuals X_r and shrinking the encirclement, as shown as follow:

$$X_i^{\text{new}} = Z_2 \times (X_{i_1} + |rd| \times (X_{i_1} - X_{i_2})) + (1 - Z_2) \times X_i \quad (29)$$

where Z_2 is a $1 \times D$ vector with each element randomly selected from $\{0, 1\}$.

To ensure rational utilization of spatial resources, a counter t_c (initialized to 0) is used to record the number of consecutive iterations where X_i fails to find a better position. When $t_c > 0.8N$ (more than 80% of the population fails to improve) or $\text{rand} > C$, the individual and population enter an alert state, considering the current area may be resource-exhausted. To avoid being trapped in poor resource areas, for each dimension, if a different rand is less than $p = 0.2C + C$, the candidate new position $X_{i,j}^{\text{new}}$ is retained; otherwise, $X_{i,j}^{\text{new}}$ is reset to the original position $X_{i,j}$. Where $C = 1 - \frac{It}{MaxIt}$ is a decreasing factor regulating the balance between exploration and exploitation. This strategy balances the retention of new search information and fallback to reliable historical positions.

Surround Search Strategy

This strategy simulates the spiral and spherical encircling behavior of Cuckoo Catfish around prey, ensuring full exploration and exploitation of spatial information. Cuckoo Catfish adjust their positions by sensing and sharing spatial information with each other.

Spiral Encircling Strategy The i -th individual X_i randomly selects another individual X_r in the population for position exchange, with two scenarios: 1. If X_r has a better position than X_i , X_i approaches X_r in a spiral manner, with the attraction vector $\text{step} = X_r - E \times X_i$; 2. If X_i has a better or equal position to X_r , X_r tends towards X_i , with the attraction vector $\text{step}_1 = X_i - E \times X_r$.

The position update formula for the spiral encircling strategy is:

$$X_i^{\text{new}} = \begin{cases} X_i + F \times R_1 \times \frac{\text{step}}{2} + sT^n \times (1 - R_1) \times |\text{step}| \\ \quad + V \times \frac{J_i}{It}, & \text{mod}(i, 2) = 0 \\ X_r + F \times R_1 \times \frac{\text{step}_1}{2} + cT^n \times (1 - R_1) \times |\text{step}_1| \\ \quad + V \times \frac{J_i}{It}, & \text{otherwise} \end{cases} \quad (30)$$

where F is a direction factor randomly selected from $\{-1, 1\}$; $T = (1 - \sin(\frac{\pi}{2} \times \frac{It}{MaxIt}))^{\frac{It}{MaxIt}}$ is a shrinkage factor; $c = a \times e^{\frac{b\theta_c}{2}} \times \cos \theta_c$ and $s = a \times e^{\frac{b\theta_c}{2}} \times \sin \theta_c$ are logarithmic spiral coefficients. Where $a = 1.34$, $b = 0.3$, $\theta_c = (1 - 10 \times \frac{i}{N}) \times \pi$; E is a kinetic energy factor: $E = T + r_1$, where r_1 is a 1-dimensional random perturbation; $J_{in} = \left| \frac{1}{N} \sum_{i=1}^N (\frac{1}{D} \sum_{j=1}^D \frac{X_i - X_{best}}{X_{worst} - X_{best}}) \right|$ is the initial average aggregation degree of the population. And $J_i = \frac{1}{D} \sum_{j=1}^D \frac{X_i - X_{best}}{X_{worst} - X_{best}}$ is the aggregation degree of the i -th individual. And V is an adaptive velocity factor:

$$V = \begin{cases} 2(r_1(X_{mean} - X_i) + r_2(X_{best} - X_i)), & J_i < J_{in} \\ 2(r_1(X_{rand2} - X_{rand3}) + r_2(X_{rand1} - X_i)), & \text{otherwise} \end{cases} \quad (31)$$

where X_{mean} is the average position of the population, and r_1, r_2 are random numbers in $[0, 1]$.

Spherical Wrap-around Strategy The pre-selected center positions for encirclement are composed of the top three optimal individuals and the population average position: $RotX = [X_{best}, X_{best2}, X_{best3}, mean(X)]$. X_i randomly selects a center position from $RotX$ and a parameter X_q (with $P(X_q = e_q) = \frac{1}{3}$ for $e_q \in \{1, 2, 3\}$) to perform a spherical search:

$$X_i^{new} = \begin{cases} RotX_r + 2wF \times \cos(Rt_1) \times \sin(Rt_2) \times (RotX_r - X_i), & X_q = 1 \\ RotX_r + 2wF \times \sin(Rt_1) \times \cos(Rt_2) \times (RotX_r - X_i), & X_q = 2 \\ RotX_r + 2wF \times \cos(Rt_2) \times (RotX_r - X_i), & X_q = 3 \end{cases} \quad (32)$$

where $w = 1 - \frac{e^{\frac{It}{MaxIt}} - 1}{e - 1}$ is an adaptive encirclement radius. And Rt is a D -dimensional rotation angle within $[\theta_c, 2\pi]$. When the fitness of X_i^{new} is better than the previous position, the worst individuals in the population have a 25% chance to be guided to the position of X_i .

Adjustment and Transition Strategies To enhance the spiral and spherical encircling strategies, adjustment and transition mechanisms are introduced:

First, For the spiral encircling strategy, after multiple executions:

$$X_i^{new} = round(lesp) + F \times \frac{r_1 \times R_1}{It^4} \times X_i^{new} \quad (33)$$

where $lesp = r_1 X_{r1} + (1 - r_1) X_{r2}$, and $round$ is the rounding function.

Then, for the spherical encircling strategy, after multiple executions:

$$X_i^{new} = X_{best} \times (1 - (1 - \frac{1}{X_r}) \times R_1) \quad (34)$$

Finally, a transition strategy balances exploration and exploitation by dividing the population equally into two parts:

one moving towards X_{best} (exploitation, $step_2 = X_{best} - E \times X_i$) and the other moving in the opposite direction (exploration, $step_2 = X_i - E \times X_{best}$):

$$X_i^{new} = \begin{cases} \frac{C}{It} \times (r_1 X_{best} - r_3 X_i) + T^2 \times lev(D) \times |step_2|, & mod(i, 2) = 0 \\ \frac{X_{best} + X_r}{2} + De \times (2R_1 step_2 - \frac{R_2}{2} (De \times R_3 - 1)), & \text{otherwise} \end{cases} \quad (35)$$

where $De = C \times F$ limits the movement distance, and $lev(D) = 0.05 \times \frac{u}{|v|^{-\beta}}$ is a D -dimensional Levy distribution, where u and v following normal distribution, $\beta = 1.5$.

Chaotic Predation Strategy

During the invasion of cichlids' nests, Cuckoo Catfish cause chaotic reactions in cichlids. The mathematical model of this predation behavior considers the movement states and mutual interactions between Cuckoo Catfish and cichlids:

$$X_i^{new} = \begin{cases} X_{best} + F \times S \times (X_{best} - X_i), & J_i > J_{in} \\ X_{best} \times (1 + T^5 \times Cy \times E) + F \times S \times (X_{best} - X_i), & J_{in} \geq J_i > J_{in} \times Lx \\ X_{best} \times (1 + T^5 \times Gs) + F \times S \times (X_{best} - X_i), & \text{otherwise} \end{cases} \quad (36)$$

where $Lx = |rd| \times r_1$ is a warning factor; $Cy = \frac{1}{\pi \times (1 + C^2)}$ is a chaotic coefficient simulating small-scale movements of cichlids; Gs is a chaotic coefficient following a Gaussian distribution with mean 0 and variance C^2 , simulating large-scale movements of cichlids; S is a swimming vector.

Death and Parasitic Strategy

A linearly decreasing death probability die is designed to simulate the survival risk of Cuckoo Catfish. In the early iterations, individuals far from prey approach to swallow eggs and lay their own, with the mathematical model:

$$best = X_{best} \times (lev(1) \times A + |rd_1| \times (1 - A)) \quad (37)$$

where A is randomly selected from $\{0, 1\}$; $best$ represents the eggs distributed near X_{best} .

The rebirth method is switched using:

$$X_i^{new} = \begin{cases} r_1 \times (Up_c - Low_c) + Low_c, & rand > C \\ r_1 \times (Up - Low) + Low, & \text{otherwise} \end{cases} \quad (38)$$

where $Up_c = \max(best)$, $Low_c = \min(best)$.

The final position update rule for CCO is:

$$X_i = \begin{cases} X_i^{new}, & F_i^{new} < F_i \\ X_i, & \text{otherwise} \end{cases}, \quad t = \begin{cases} 0, & F_i^{new} < F_i \\ t + 1, & \text{otherwise} \end{cases} \quad (39)$$

where F_i^{new} and F_i are the fitness values of X_i^{new} and X_i , respectively.

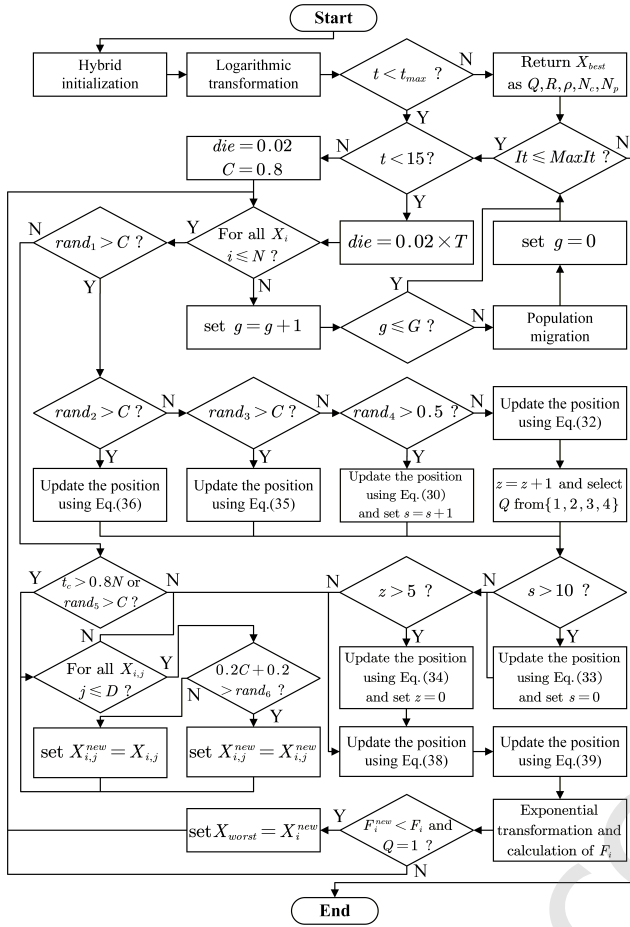


Fig. 3 Flowchart of ICCO.

Multi-Population Cooperative Evolution

To further enhance population diversity, mitigate premature convergence, and improve global search capability, a multi-population cooperative evolution mechanism is employed. The original population is divided into multiple sub-populations, each of which independently executes the ICCO algorithm, while high-quality individuals are periodically exchanged among sub-populations.

Let m denote the number of sub-populations, each consisting of N individuals. The overall population P size M is given by:

$$P = \bigcup_{j=1}^m X^{(j)}, M = m \times N \quad (40)$$

where $X^{(j)}$ represents the j -th sub-population.

Every G generations, a migration operation is performed. Specifically, the top 10% elite individuals in each sub-population are used to replace the bottom 10% worst individuals in neighboring sub-populations following a cyclic migration scheme:

The top 10% individuals in $P^{(2)}$ replace the bottom 10% individuals in $P^{(1)}$;

The top 10% individuals in $P^{(3)}$ replace the bottom 10% individuals in $P^{(2)}$;

...

The top 10% individuals in $P^{(1)}$ replace the bottom 10% individuals in $P^{(j)}$.

This cooperative migration strategy enables effective information sharing among sub-populations while maintaining diversity, which is particularly beneficial for optimizing the multimodal objective function J_p .

In summary, the flowchart of ICCO is shown in Fig. 3.

4.2 Design of MPC-PO

In a conventional MPC framework, control inputs are obtained by solving an optimization problem that minimizes the cost function in (9), which incorporates tracking errors and control increments under system dynamics and constraints. While this approach effectively addresses multivariable and predictive control, its performance strongly depends on the tuning of weighting matrices as well as the prediction and control horizons. Inappropriate parameter selection can degrade tracking accuracy, increase control effort, or impose excessive computational load, posing challenges for real-time mobile robot applications.

To address this issue, a parallel optimization MPC framework, referred to as MPC-PO, is proposed. In this framework, the standard MPC optimization is retained to compute real-time control commands, while a parallel parameter optimization strategy is executed concurrently to adaptively tune MPC parameters. The parameter optimization process evaluates the overall control performance and computational efficiency without interrupting the real-time MPC execution, thereby ensuring real-time feasibility.

Specifically, the MPC-PO framework optimizes the weighting matrices Q and R , the relaxation factor weight ρ , the prediction horizon N_p , and the control horizon N_c . The performance of a given parameter set is evaluated using a parallel optimization objective function J_p defined as:

$$J_p = w_e \sum_{j=0}^t e^2(j) + \sum_{i=1}^3 t_i + t_{solve}, \quad (41)$$

where w_e is a weighting coefficient introduced to balance the contributions of different terms by normalizing their physical units. It should be noted that an inappropriate selection of w_e may cause one term in the objective function J_p to dominate the optimization process, thereby degrading trajectory tracking accuracy. However, the influence of w_e on the computational time required to solve J_p and J is relatively minor, $\sum_{j=0}^t e^2(j)$ denotes the sum of squared tracking errors (SSE), and $t_x = t_1$, $t_y = t_2$, $t_\theta = t_3$ respectively represent the time when the mobile robot is not in the reference state under the three state components of x , y and θ in the predicted trajectory. The term t_{solve} denotes the average computational time required to solve the MPC optimization problem, which is primarily determined by the prediction horizon N_c , control horizon N_c , control frequency f_c and the onboard computational capability of the mobile robot.

To further improve computational efficiency, the parallel parameter optimization is executed in an event-triggered manner rather than at every control step. A parameter optimization event is triggered when significant changes in the robot state occur, including abrupt variations in linear or angular velocity, transitions into or out of the reference

trajectory, or situations in which the MPC predicted trajectory deviates from the reference path.

Let f_c denote the control frequency of the mobile robot. The maximum allowable time for parameter optimization is therefore constrained by $t_{max} = \frac{1}{f_c}$. The proposed MPC-PO framework is capable of handling cases where the computation time exceeds t_{max} . When external conditions change, a triggering event occurs at time step k , activating the parallel parameter optimization process. At the beginning of the first t_{max} interval, the MPC controller computes the control inputs $u(k)$ (and $w(k)$, if applicable) using the current parameter set $N_p(k)$, $N_c(k)$, $Q(k)$, $R(k)$ and $\rho(k)$. Simultaneously, the ICCO algorithm is initialized with these parameter values, while the auxiliary optimization variables t_c , die , s and z are reset to their initial values, and the optimization process begins. Within the first t_{max} interval, the ICCO algorithm searches for the optimal solution X_{best} by minimizing the parallel objective function J_p . At the end of this interval, if the ICCO has converged or the maximum number of optimization iterations $It > MaxIt$ is reached, the obtained optimal solution $X_{best}(k)$ is decomposed into the corresponding control parameters $N_p(k+1)$, $N_c(k+1)$, $Q(k+1)$, $R(k+1)$ and $\rho(k+1)$. These optimized parameters are then applied by the MPC controller in the next t_{max} interval to compute the control inputs $u(k+1)$ (and $w(k+1)$). If the ICCO has neither converged nor reached $MaxIt$ within the first t_{max} interval, the MPC instead employs a temporary parameter set corresponding to the minimum value of J_p obtained during that interval to compute the control inputs in the subsequent step. Meanwhile, the ICCO continues the optimization process over successive t_{max} intervals, progressively refining the parameters. This procedure repeats across multiple t_{max} intervals until the ICCO converges or the maximum number of iterations $It > MaxIt$ is satisfied, yielding the final optimized parameter set. Fig. 1 depicts the proposed architecture of MPC-PO.

Through this event-triggered parallel optimization strategy, the proposed MPC-PO framework achieves adaptive parameter tuning while preserving the real-time execution of MPC, thereby enhancing tracking performance, computational efficiency, and robustness in mobile robot applications.

5 Design of ESO and Stability Analysis

5.1 Design of ESO

According to modeling analysis and kinematics model equation (4) in Section II, the mobile robot error system can be written as follows:

$$\begin{cases} \dot{x} = v \cos \theta + d_1 \\ \dot{y} = v \sin \theta + d_2 \\ \dot{\theta} = \omega + d_3 \end{cases} \quad (42)$$

Make the following assumptions [36]:

Assumption 3. The disturbances suffered by mobile robots when driving are bounded and differentiable.

For the equation of state in the x direction in equation (42), let $x = x_1$, $d_1 = x_2$, $h_0 = \cos \theta$, then the equation of

state can be rewritten as:

$$\begin{cases} \dot{x}_1 = h_0 v + x_2 \\ \dot{x}_2 = h \end{cases} \quad (43)$$

Let $e_1 = z_1 - x_1$, $e_2 = z_2 - x_2$, $e_{s1} = z_2 - x_2$, $e_{s1} = [e_1 \ e_2]^T$ is the observation error [37]. Therefore, the ESO of system in equation (43) is designed to:

$$\begin{cases} \dot{z}_1 = z_2 - \eta_1 e_1 + b_0 v \\ \dot{z}_2 = -\eta_2 \text{fal}(e_1, \rho, \lambda) \end{cases} \quad (44)$$

where z_1 and z_2 are the observed values of x_1 and x_2 , η_1 and η_2 are positive constants [38]. Let the disturbance estimate $\hat{d}_1 =$

z_2 . $\text{fal}(e_1, \rho, \lambda) = \begin{cases} \frac{e_1}{\lambda(1-\rho)}, & |e_1| \leq \lambda \\ |e_1|^\rho \text{sign}(e_1), & |e_1| > \lambda \end{cases}$ is a nonlinear function, where ρ and λ are two constants, and $0 < \rho < 1$, $\lambda > 0$. Substituting the equation (43) into (44), the error system is:

$$\begin{cases} \dot{e}_1 = e_2 - \eta_1 e_1 \\ \dot{e}_2 = -\eta_2 \text{fal}(e_1, \rho, \lambda) - h \end{cases} \quad (45)$$

In the same way, the observation errors in the other two directions are $e_{s2} = [e_3 \ e_4]^T$ and $e_{s3} = [e_5 \ e_6]^T$, the disturbance estimates are $\hat{d}_2$ and $\hat{d}_3$, and the parameters of the observer are $\eta_{s1} = \text{diag}\{\eta_1, \eta_3, \eta_5\}$ and $\eta_{s2} = \text{diag}\{\eta_2, \eta_4, \eta_6\}$, respectively. Therefore, the estimated control output is $\hat{u} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{d}$, where $\hat{d} = [\hat{d}_1 \ \hat{d}_2 \ \hat{d}_3]^T$.

The estimated control output of interference obtained by the expansion state observer $\hat{u}(k)$ is compensated into the mobile robot model prediction controller output $u^*(k)$ in equation (20), and the final control output $u(k)$ of the system is obtained:

$$u(k) = u^*(k) - \frac{\hat{u}(k)}{b_0} \quad (46)$$

5.2 Stability Analysis of ESO

Theorem 2. Consider the system dynamics in (9) and (10), and the proposed ESO in (44). Assume the derivative of the lumped disturbance is bounded, i.e., there exists a constant $H > 0$ such that $|h(t)| \leq H$ for all $t \geq 0$. Then there exist positive observer gains η_1, η_2 and positive coefficients U, O, K satisfying

$$\begin{aligned} U &> \frac{4}{3}\eta_2 K, \quad \eta_2 > \frac{3}{4}\eta_1^2 \\ 4\left(\frac{3}{2}U\eta_1 - O\eta_2\right)O &> \left(\frac{3}{2}U - 2K\eta_2 + O\eta_1\right)^2 \end{aligned} \quad (47)$$

such that the ESO estimation errors e_1 and e_2 are uniformly ultimately bounded, with the residual set size determined by the disturbance bound H .

Define estimation errors $e_1 = x_1 - \hat{x}_1$ and $e_2 = x_2 - \hat{x}_2$. Construct the Lyapunov function:

$$V = U|e_1|^{\frac{3}{2}} - O|e_1|^{\frac{1}{2}}\text{sign}(e_1)e_2 + Ke_2^2 \quad (48)$$

where $U, O, K > 0$ satisfy (47).

Positive definiteness and radial unboundedness. Let $w_1 = |e_1|^{\frac{1}{2}}$ and $w_2 = e_2$. Then V can be rewritten as $V = w_1 \cdot w^T P w$, where $w = [w_1, w_2]^T$ and $P = \begin{bmatrix} U & -\frac{O}{2} \\ -\frac{O}{2} & K \end{bmatrix}$. From

(47), $UK > \frac{O^2}{4}$, so $P \succ 0$. Thus V is positive definite and radially unbounded with respect to e_1, e_2 .

For $e_1 \neq 0$, taking the time derivative of V along the error trajectories yields

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial e_1} \dot{e}_1 + \frac{\partial V}{\partial e_2} \dot{e}_2 \\ &= \left(\frac{3}{2} U |e_1|^{\frac{1}{2}} \text{sign}(e_1) - \frac{O}{2} |e_1|^{-\frac{1}{2}} e_2 \right) (e_2 - \eta_1 |e_1|^{\frac{1}{2}} \text{sign}(e_1)) \\ &\quad + \left(-O |e_1|^{\frac{1}{2}} \text{sign}(e_1) + 2K e_2 \right) (-h - \eta_2 \text{sign}(e_1)) \end{aligned} \quad (49)$$

Expanding and collecting terms, we obtain

$$\dot{V} = -|e_1|^{-\frac{1}{2}} \left(p_1 |e_1| - p_2 |e_1|^{\frac{1}{2}} \text{sign}(e_1) e_2 + p_3 e_2^2 \right) + \Upsilon \quad (50)$$

where $p_1 = \frac{3}{2} U \eta_1 - O \eta_2$, $p_2 = \frac{3}{2} U - 2K \eta_2 + O \eta_1$, $p_3 = O$, $\Upsilon = O h |e_1|^{\frac{1}{2}} \text{sign}(e_1) - 2K h e_2$

The term in parentheses is a quadratic form in w_1, w_2 . By (47), $p_1 > 0$, $p_3 > 0$ and $4p_1 p_3 > p_2^2$, so the quadratic form is positive definite. Thus $\exists \lambda > 0$ such that

$$p_1 |e_1| - p_2 |e_1|^{\frac{1}{2}} \text{sign}(e_1) e_2 + p_3 e_2^2 \geq \lambda (|e_1| + e_2^2) \quad (51)$$

Substituting into (50) gives

$$\dot{V} \leq -\lambda (|e_1|^{\frac{1}{2}} + |e_1|^{-\frac{1}{2}} e_2^2) + |\Upsilon| \quad (52)$$

Under $|h(t)| \leq H$, the disturbance term satisfies $|\Upsilon| \leq O H |e_1|^{\frac{1}{2}} + 2K H |e_2|$. By Young's inequality, $|e_2| \leq \frac{1}{2} |e_1|^{\frac{1}{2}} + \frac{1}{2} |e_1|^{-\frac{1}{2}} e_2^2$. Thus

$$|\Upsilon| \leq H(O + K) |e_1|^{\frac{1}{2}} + H K |e_1|^{-\frac{1}{2}} e_2^2 \quad (53)$$

Substituting into (52):

$$\begin{aligned} \dot{V} &\leq -\alpha (|e_1|^{\frac{1}{2}} + |e_1|^{-\frac{1}{2}} e_2^2) \\ \alpha &= \min \{ \lambda - H(O + K), \lambda - H K \} \end{aligned} \quad (54)$$

By selecting sufficiently large η_1, η_2 , we ensure $\alpha > 0$. Since V is homogeneous of degree $3/2$ with weights $(2, 1)$, there exists $c > 0$ such that $|e_1|^{\frac{1}{2}} + |e_1|^{-\frac{1}{2}} e_2^2 \geq c V^{\frac{1}{3}}$. Hence the errors converge to a residual set determined by H in finite time, and are uniformly ultimately bounded.

6 Experimental Results and Analysis

This section presents the experimental results of three different control strategies (MPC, MPC-PO and MPC-POESO) designed for the same four-wheel differential-drive mobile robot system. The mobile robot used in the experiments is shown in Fig. 4, and its specific parameters are listed in Table 1. Owing to the hardware characteristics of the mobile robot, the t_{solve} satisfies: $t_{solve} = -0.1033 + 0.00325 N_p + 0.00122 N_c$. To evaluate the performance of the three control strategies, three representative reference trajectories were considered, namely, a straight line, a curved path, and a circular path. All experiments were conducted under the same desired velocity

to ensure a fair comparison. The root mean square error (RMSE) and mean absolute error (MAE) were adopted as quantitative performance indices to assess tracking accuracy of the control strategies, as these metrics are widely recognized as comprehensive and general tools for control performance evaluation. Furthermore, to verify the effectiveness of ESO employed in this study in estimating and compensating for disturbances, random noise disturbances were deliberately injected into the sensors of the mobile robot during the experiments.

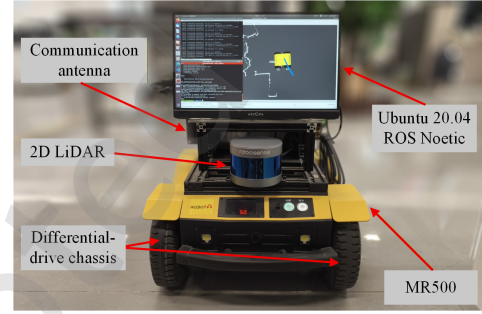


Fig. 4 MR500 mobile robot.

Table 1 Specification of the MR500 mobile robot

Parameters	Value	Parameters	Value
Length	595mm	Width	450mm
Height	545mm	Weight	43kg
Wheelbase	300mm	Tire Diameter	203mm
Max Speed	1.6m/s	Control Frequency	60Hz

6.1 Scenario 1: Line

The mobile robot is initialized with a lateral deviation of 0.2m from the straight reference path and tracks a linear trajectory with a length of 4m, the actual trajectory of the three control methods is shown in Fig. 5, the RMSE and MAE of the three control methods are shown in Table 2, the error plot of the three control strategies is shown in Fig. 6, and the ESO disturbance estimation of MPC-POESO is shown in Fig. 7. It can be observed that, due to the presence of random noise in the sensors, both MPC and MPC-PO suffer from trajectory oscillations caused by localization inaccuracies, leading to tracking errors. In contrast, the ESO incorporated in MPC-POESO is capable of effectively estimating and compensating for disturbances, thereby subjecting the mobile robot to less interference and resulting in a smoother trajectory with smaller tracking errors. Furthermore, in the absence of ESO, the MPC-PO algorithm enables the robot to converge to the reference trajectory more rapidly from an initial offset outside the trajectory, while achieving smaller errors compared with the conventional MPC.

Table 2 RMSE and MAE of Scenario 1

Control method	MPC	MPC-PO	MPC-POESO
RMSE lateral (m)	0.077303	0.057955	0.056391
RMSE yaw (deg)	7.034888	4.381998	3.777403
MAE lateral (m)	0.046773	0.027955	0.027296
MAE yaw (deg)	4.310027	2.131364	2.050812

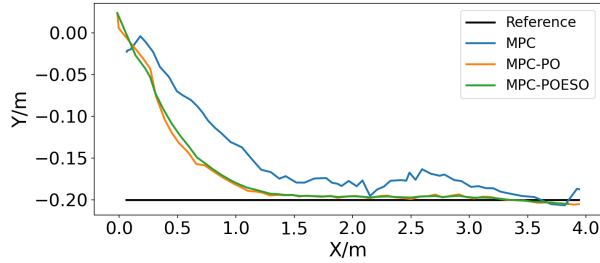


Fig. 5 Trajectory tracking of mobile robot.

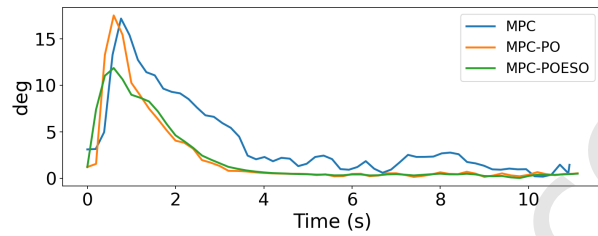
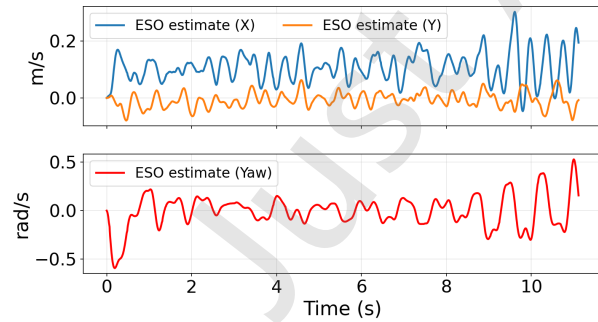


Fig. 6 Yaw error of three control methods.

Fig. 7 ESO estimations of x , y and yaw .

6.2 Scenario 2: Circle

The mobile robot tracks a circular trajectory with a diameter of 4m, the actual trajectory of the three control methods is shown in Fig. 8, the RMSE and MAE of the three control strategies are shown in Table 3, the lateral and yaw error plots of the three control methods is shown in Fig. 9, and the ESO disturbance estimation of MPC-POESO is shown in Fig. 10. Consistent with the observations in Scenario 1, the proposed MPC-POESO framework continues to demonstrate superior tracking performance under

the conditions of Scenario 2. Specifically, the combined effects of parallel optimization and ESO-based disturbance compensation further enhance trajectory smoothness and tracking accuracy, while maintaining rapid convergence to the reference path. Compared with both MPC and MPC-PO, the proposed method exhibits reduced oscillations and smaller tracking errors, confirming the robustness and effectiveness of the proposed optimization strategy under more challenging operating conditions.

Table 3 Max Error, RMSE and MAE of Scenario 2

Control method	MPC	MPC-PO	MPC-POESO
Max lateral error (m)	0.202133	0.109527	0.066345
Max yaw error (m)	11.937677	5.188657	7.453979
RMSE lateral (m)	0.088680	0.036303	0.023617
RMSE yaw (deg)	4.838885	2.959797	2.215103
MAE lateral (m)	0.076780	0.028464	0.016947
MAE yaw (deg)	4.183085	1.939714	1.688540

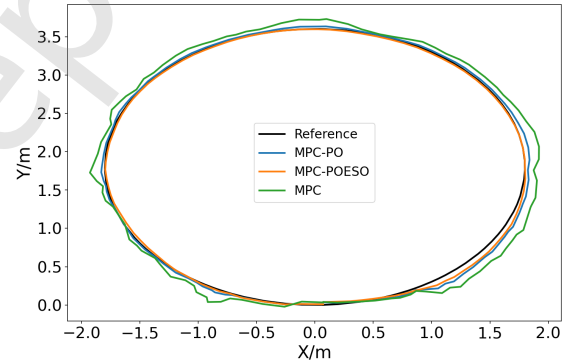


Fig. 8 Trajectory tracking of mobile robot.

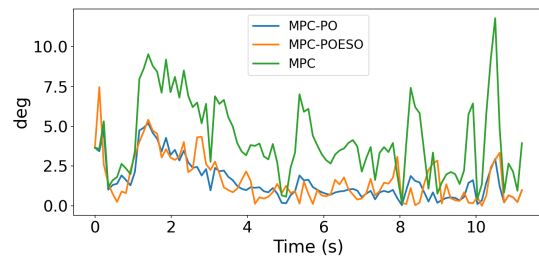
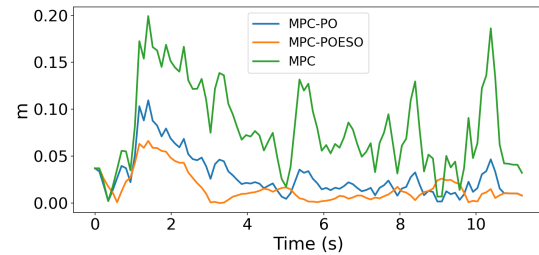


Fig. 9 Lateral and yaw error of three control methods.

7 Conclusions

This paper proposes a parallel optimization model predictive control framework that integrates an event-triggered parallel optimization strategy with an ESO-based disturbance compensation mechanism. A conventional MPC generates real-time control inputs, while a parallel objective function adaptively tunes key MPC parameters under real-time constraints. The resulting high-dimensional optimization problem is efficiently solved using an Improved Cuckoo Catfish Optimizer, and a nonlinear ESO is employed to estimate and compensate for disturbances and modeling uncertainties. Experimental results obtained on the MR500 mobile robot platform verify the effectiveness of the proposed approach. Overall, the proposed framework offers a practical and effective solution for robust trajectory tracking of mobile robots under real-time constraints.

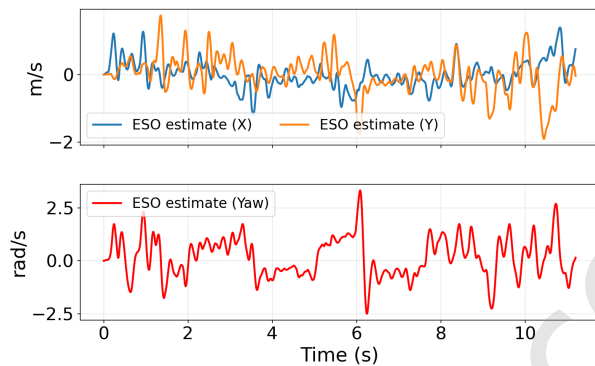


Fig. 10 ESO estimations of x , y and yaw .

Acknowledgements

This work was supported in part by the Basic Science Center Program of the National Natural Science Foundation of China (62388101), and in part by National Natural Science Foundation of China under Grant 62303010 and 62573003, in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2023-039.

References

- [1] Bai Li, Xiaoming Chen, Tankut Acarman, Xiaohui Li, and Youmin Zhang. Recent advances in motion planning and control of autonomous vehicles. *Electronics*, 12(23):4881, 2023.
- [2] Bai Li, Yazhou Wang, Siji Ma, Xuepeng Bian, Hu Li, Tantan Zhang, Xiaohui Li, and Youmin Zhang. Adaptive pure pursuit: A real-time path planner using tracking controllers to plan safe and kinematically feasible paths. *IEEE Transactions on Intelligent Vehicles*, 8(9):4155–4168, 2023.
- [3] Yiwen Qi, Simeng Zhang, and Yang Shi. Optimized active disturbance rejection control for switched systems:

A triggered-learning approach. *IEEE Transactions on Automatic Control*, 2025.

- [4] Jiliang Song, Dawei Shi, Shu-Xia Tang, Hao Yu, and Yang Shi. Event-triggered learning-based control for output tracking with unknown cost functions. *Automatica*, 176:112235, 2025.
- [5] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, and Long Chen. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023.
- [6] Jun Liu, Jinhao Tan, Hongbo Li, and Badong Chen. Active disturbance rejection consensus control of multi-agent systems based on a novel neso. *IEEE/ASME Transactions on Mechatronics*, 30(1):634–644, 2024.
- [7] Kaihui Wang, Wei Zou, Ruichen Ma, Yu Wang, and Hu Su. Model predictive trajectory tracking control of an underactuated bionic underwater vehicle. *IEEE/ASME Transactions on Mechatronics*, 29(3):1690–1701, 2023.
- [8] Liwen Xu, Bailing Tian, Cong Wang, Junjie Lu, Dandan Wang, Zhiyu Li, and Qun Zong. Fixed-time disturbance observer-based mpc robust trajectory tracking control of quadrotor. *IEEE/ASME Transactions on Mechatronics*, 2024.
- [9] Yifan Liu, Tao Hu, Xiaoqing Guan, Yixu Wang, Bixuan Zhang, You Wang, and Guang Li. Adaptive mpc-based multi-terrain trajectory tracking framework for mobile spherical robots. *IEEE/ASME Transactions on Mechatronics*, 2025.
- [10] Ahmed D Sabiha, Mohamed A Kamel, Ehab Said, and Wessam M Hussein. Ros-based trajectory tracking control for autonomous tracked vehicle using optimized backstepping and sliding mode control. *Robotics and Autonomous Systems*, 152:104058, 2022.
- [11] Haonan Peng, Weida Wang, Quan An, Changle Xiang, and Liang Li. Path tracking and direct yaw moment coordinated control based on robust mpc with the finite time horizon for autonomous independent-drive vehicles. *IEEE Transactions on Vehicular Technology*, 69(6):6053–6066, 2020.
- [12] Yang Tian, He Ma, Lei Ma, Shuqiang Li, Zhenfeng Wang, Xiangyu Wang, and Liang Li. Path tracking control of commercial vehicle emergency obstacle avoidance based on mpc and active disturbance rejection control. *IEEE Transactions on Transportation Electrification*, 2024.
- [13] Suhail Ahmad Suhail, Mohammad Abid Bazaz, and Shoeb Hussain. Mpc based active disturbance rejection control for automated steering control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(12):3199–3206, 2021.
- [14] Qinglai Wei, Shanshan Jiao, Fei-Yue Wang, and Qi Dong. Robust optimal parallel tracking control based on adaptive dynamic programming. *IEEE Transactions on Cybernetics*, 54(7):4308–4321, 2024.

- [15] Shuo Xiong, Hui Xie, Kang Song, and Guohui Zhang. A speed tracking method for autonomous driving via adrc with extended state observer. *Applied Sciences*, 9(16):3339, 2019.
- [16] Jinhui Zhang, Sihang Li, Yuanqing Wu, Ran Huang, and Huan Meng. Adrc-based tube mpc approach for trajectory tracking of four-mecanum-wheel mobile vehicle. *IEEE/ASME Transactions on Mechatronics*, 2025.
- [17] Zhendong Zhu, Qiangqiang Yao, and Yiheng Shi. Intelligent vehicle adaptive robust path tracking control strategy based on mpc and adrc. *International Journal of Automotive Technology*, pages 1–20, 2025.
- [18] Yue Song, Tianxiang Lu, and Yang Shi. Robust output feedback mpc for networked control systems with two-channel random packet dropouts. *Unmanned Systems*, 13(05):1449–1462, 2025.
- [19] Long Jin, Longqi Liu, Xingxia Wang, Mingsheng Shang, and Fei-Yue Wang. Physical-informed neural network for mpc-based trajectory tracking of vehicles with noise considered. *IEEE Transactions on Intelligent Vehicles*, 9(3):4493–4503, 2024.
- [20] Kazumune Hashimoto, Yuga Onoue, Akifumi Wachi, and Xun Shen. Learning-based event-triggered mpc with gaussian processes under terminal constraints. *IEEE Transactions on Cybernetics*, 55(4):1512–1525, 2025.
- [21] Peng Lu, Ning Zhang, Lin Ye, Ershun Du, and Chongqing Kang. Advances in model predictive control for large-scale wind power integration in power systems. *Advances in Applied Energy*, 14:100177, 2024.
- [22] Mohsen Farbood, Mokhtar Shasadeghi, Afshin Izadian, and Taher Niknam. Fuzzy model predictive mppt control of interconnected wind turbines drivetrain. *Asian Journal of Control*, 24(5):2714–2728, 2022.
- [23] Xiaopeng Wang, Václav Snášel, Seyedali Mirjalili, Jeng-Shyang Pan, Lingping Kong, and Hisham A Shehadeh. Artificial protozoa optimizer (apo): A novel bio-inspired metaheuristic algorithm for engineering optimization. *Knowledge-based Systems*, 295:111737, 2024.
- [24] Youfa Fu, Dan Liu, Jiadui Chen, and Ling He. Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems. *Artificial Intelligence Review*, 57(5):123, 2024.
- [25] Changting Zhong, Gang Li, and Zeng Meng. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-based systems*, 251:109215, 2022.
- [26] Jiankai Xue and Bo Shen. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing*, 79(7):7305–7336, 2023.
- [27] Muxuan Li, Zunfeng Du, Kum Fai Yuen, Haitao Zhu, Yancang Li, and Qiuyu Yuan. Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Systems with Applications*, 239:122413, 2024.
- [28] Murat Erhan Çimen and Yaprak Yalçın. A novel hybrid firefly-whale optimization algorithm and its application to optimization of mpc parameters. *Soft Computing*, 26(4):1845–1872, 2022.
- [29] Mohammed Azmi Al-Betar, Mohammed A Awadallah, Malik Shehadeh Braik, Sharif Makhadmeh, and Iyad Abu Doush. Elk herd optimizer: A novel nature-inspired metaheuristic algorithm. *Artificial Intelligence Review*, 57(3):48, 2024.
- [30] Junbo Lian, Guohua Hui, Ling Ma, Ting Zhu, Xincan Wu, Ali Asghar Heidari, Yi Chen, and Huiling Chen. Parrot optimizer: Algorithm and applications to medical problems. *Computers in Biology and Medicine*, 172:108064, 2024.
- [31] Dunjing Yu, Fang Deng, Hongyan Wang, Xiuhui Hou, Hualin Yang, and Tikun Shan. Real-time weight optimization of a nonlinear model predictive controller using a genetic algorithm for ship trajectory tracking. *Journal of Marine Science and Engineering*, 10(8):1110, 2022.
- [32] Li Li, Weiqun Cao, Hongjiu Yang, and Qing Geng. Trajectory tracking control for a wheel mobile robot on rough and uneven ground. *Mechatronics*, 83:102741, 2022.
- [33] Amine Abadi, Amani Ayeb, Moussa Labbadi, David Fofi, Toufik Bakir, and Hassen Mekki. Robust tracking control of wheeled mobile robot based on differential flatness and sliding active disturbance rejection control: Simulations and experiments. *Sensors*, 24(9):2849, 2024.
- [34] Achmad Imam Sudianto, Muhammad Aziz Muslim, and Moch Rusli. Sensor fusion using model predictive control for differential dual wheeled robot. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pages 461–472, 2023.
- [35] Tianlei Wang, Shaowei Gu, Renju Liu, Leqing Chen, Zhu Wang, and Zhiqiang Zeng. Cuckoo catfish optimizer: A new meta-heuristic optimization algorithm. *Artificial Intelligence Review*, 58(10):326, 2025.
- [36] Zhen Zhang, Yinan Guo, Dunwei Gong, and Jianxing Liu. Global integral sliding-mode control with improved nonlinear extended state observer for rotary tracking of a hydraulic roofbolter. *IEEE/ASME Transactions on Mechatronics*, 28(1):483–494, 2022.
- [37] Hongjiu Yang, Ran Dang, and Peng Li. Extended state observer-based trajectory tracking control of a wheeled mobile robot with one unpowered trailer. *Asian Journal of Control*, 27(2):1008–1019, 2025.
- [38] Bo Qin, Huaicheng Yan, Hao Zhang, Yueying Wang, and Simon X Yang. Enhanced reduced-order extended state observer for motion control of differential driven mobile robot. *IEEE Transactions on Cybernetics*, 53(2):1299–1310, 2021.



Hejia Gao (Member, IEEE) received the B.Eng. degree in intelligence science and technology from the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China, in 2016, and the Ph.D. degree in control science and engineering from the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China, in 2021. She has been working as a visiting researcher for fifteen months at the Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Canada, in 2019. Currently, she is working as an associate professor in the School of Artificial Intelligence, Anhui University, Hefei, China. Her research interests include neural networks, RL, flexible systems, and vibration control.



Puhai Yang received the bachelor's and master's degrees from North China Electric Power University, Beijing, China, in 2014 and 2019, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Beijing Institute of Technology, Beijing. His research interests include natural language processing and dialogue system.



Changyin Sun (Senior Member, IEEE) received the B.Eng. degree in applied mathematics from the College of Mathematics, Sichuan University, Chengdu, China, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2001 and 2004, respectively. He is currently a Professor with the School of Automation, Southeast University, Nanjing, China. He has co-authored four books and published over 160 international journal papers. Prof. Sun is a Chinese Association of Automation Fellow. He is serving as an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, Neural Processing Letters, and IEEE/CAA Journal of Automatica Sinica. His research interests include intelligent control, flight control, pattern recognition, and optimal theory.



Yanling Guo received the B.Eng degree in Jiangsu University, Zhenjiang, China, in 2023. where he is currently working toward the master's degree in control engineering in the School of Artificial Intelligence, Anhui University, Hefei, China. His research interests include the path planning and trajectory tracking.