A DUNG BEETLE OPTIMIZED MPC ALGORITHM FOR MULTI-OBJECTIVE OPTIMIZATION OF UNMANNED AGRICULTURAL VEHICLE

/

一种蜣螂算法优化 MPC 的无人农机多目标优化方法

Zhenning CHEN[1)], Youtong ZHANG[1,2)], Wenqiang ZHAO[1)], Haishi DOU[2)], Hongqian WEI[1)]
[1)]School of Machinery and Vehicle, Beijing University of Technology, Beijing/P.R China;
[2)]Beijing Institute of Technology Yangtze River Delta (Jiaxing) Research Institute, Jiaxing/P.R China
Tel: +86-18901309693; E-mail: youtong@bit.edu.cn
Correspondent author: Youtong ZHANG
DOI: https://doi.org/10.35633/inmateh-78-35

Keywords: *Unmanned Agricultural Vehicle; Dung Beetle Optimization; Trajectory tracking; Model Predictive Control*

ABSTRACT

This paper presents a multi-objective optimization approach for unmanned agricultural vehicles operating in complex farmland environments. To overcome the limitations of traditional Model Predictive Control (MPC) and heuristic algorithms, a Dung Beetle Optimization-based MPC (D-MPC) multi-objective optimization method is proposed. Specifically, a kinematic model of the unmanned agricultural vehicle is established, incorporating the operational characteristics of complex farmland conditions. The Dung Beetle Optimization (DBO) algorithm is integrated into the MPC framework to enhance performance by leveraging the population-based search behavior of dung beetles. This integration improves both control accuracy and computational efficiency by dynamically adjusting control inputs based on real-time motion predictions, enabling more precise trajectory optimization. Experimental validation is conducted through a dual-verification approach, including both simulation and real-vehicle tests. The results indicate that, compared with conventional control methods, the proposed approach improves trajectory tracking accuracy by approximately 50% and 75% in two representative simulation scenarios, while increasing the battery State of Charge (SOC) by 0.1% and 0.12%, respectively. In real-vehicle experiments, trajectory tracking accuracy is improved by 70%, and SOC is increased by 0.015%.

摘要

本文研究了无人农机在复杂农田场景下的多目标优化问题。针对传统模型预测控制（MPC）和启发式算法容易陷入局部最优或收敛速度慢的问题，提出了一种蜣螂算法优化 MPC（D-MPC）的多目标优化算法。首先，针对复杂农田作业的特点，建立了无人农机的运动学模型。其次，采用蜣螂优化算法（DBO）对传统 MPC 算法进行优化。利用蜣螂种群的搜索特性，显著提高了控制器的控制精度和计算速度。该方法根据实时运动预测调整控制输入，实现对未来状态的预测和优化。最后，设计了一个包括仿真和实车试验的双重验证试验方案。实验结果表明，与传统的控制方法相比，在两种典型的仿真场景下，该方法的轨迹跟踪精度分别提高了 50%和 75%，而电池的荷电状态（SOC）分别提高了 0.1%和 0.12%。在实车试验中，轨迹跟踪精度提高了 70%，SOC 提高了 0.015%。

INTRODUCTION

As agricultural modernization accelerates, intelligent machinery has become indispensable for improving production efficiency and operational precision *(Luo et al., 2021)*. Among these technologies, unmanned agricultural vehicles (UAVs) represent a critical form of automation, featuring autonomous navigation and high-accuracy operation capabilities. It is progressively replacing traditional manual labor and becoming one of the key technologies in modern agricultural production *(Zhang et al., 2020)*. However, the inherently complex and unstructured nature of farmland environments continues to pose significant challenges for UAVs, particularly in maintaining reliable trajectory tracking, stable velocity regulation, and effective State of Charge (SOC) management *(Shi et al., 2023)*. Therefore, improving trajectory tracking accuracy and speed stability in complex farmland operations has become one of the critical issues to address in the field of intelligent agriculture *(Zhen et al., 2023)*. The development of advanced multi-objective optimization control algorithms with high precision and computational efficiency is therefore of substantial theoretical significance and offers broad application prospects for UAVs.

The multi-objective optimization problem addressed in this study for unmanned agricultural vehicles involves trajectory tracking, velocity regulation, and State of Charge (SOC) control. Conventional methods such as Proportional-Integral-Derivative (PID), Sliding Mode Control (SMC), and Linear Quadratic Regulator (LQR) often struggle with nonlinearities, limited precision, and chattering effects (*Chen et al., 2023; Yu et al., 2021*). In contrast, Model Predictive Control (MPC) predicts future system states and optimizes control actions in real time while accounting for model dynamics and system constraints (*Rokonuzzaman et al., 2023*). Owing to its adaptability and robustness, MPC achieves high accuracy in trajectory and speed tracking under nonlinear, multi-constraint conditions. Consequently, it has become one of the most advanced and widely adopted control frameworks for autonomous unmanned agricultural vehicles (*Kim et al., 2023*). However, MPC may still fall into local optimization, high computational complexity and high dependence on controlled models *(Gong et al., 2023; Bayat et al., 2023)*. To overcome these limitations, extensive research has focused on enhancing its robustness and real-time performance under dynamic conditions. Chu et al. (2022) developed a hybrid control strategy integrating PID feedback with MPC, where the PID controller refines MPC adjustments to achieve high-precision trajectory tracking. *Zhai et al. (2022)* proposed an adaptive MPC-based controller that dynamically modifies the prediction horizon according to vehicle speed, optimizing torque distribution among four hub motors and improving control responsiveness. *Chen et al. (2024)* introduced a proximal policy optimization framework to adaptively adjust the prediction horizon for performance maximization. Furthermore, *Zhang et al. (2021)* designed an adaptive learning MPC using least-squares parameter estimation to mitigate environmental uncertainties in unmanned agricultural vehicle operation.

Compared to the aforementioned MPC optimization methods, heuristic algorithms optimize MPC more effectively by navigating complex solution spaces and avoiding local optima, providing superior adaptability and robustness for MPC optimization in complex dynamic environments *(Seyyedabbasi et al., 2021)*. Common heuristic algorithms include Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Grey Wolf Optimization (GWO) *(Moazen et al., 2023)*. However, these widely used algorithms often face challenges such as poor global search capability or slow convergence rates *(Hou et al., 2022; Almufti et al., 2021)*. In contrast, the Dung Beetle Optimization (DBO) algorithm demonstrates faster convergence and a stronger ability to avoid local optima. DBO is capable of identifying the global optimum more quickly, offering higher stability and reliability *(*Wang *et al., 2024)*. When integrated with MPC, DBO helps to overcome the local optima issue commonly encountered in traditional MPC, thereby improving the system's control accuracy and computational speed *(He et al., 2024)*.

Due to the operation of agricultural vehicles in complex terrain and uncertain conditions, the dynamic system usually shows strong nonlinearity and model uncertainty. Therefore, when this method is applied to the motion control of agricultural vehicles, compared with the traditional gradient based MPC method, DBO, as a population intelligent optimization algorithm, has better global search ability and does not rely on gradient information, so it has potential advantages in dealing with nonconvex optimization problems. According to the above research, this study proposes a Dung Beetle Optimized Model Predictive Control (D-MPC) algorithm to address the multi-objective optimization problem of trajectory tracking, velocity regulation, and State of Charge (SOC) management for unmanned agricultural vehicles operating in complex farmland environments.

MATERIALS AND METHODS

Vehicle model

This study focuses on the motion control of vehicles and employs the vehicle kinematics model depicted in Figure 1. The model is designed to represent the vehicle's motion dynamics, incorporating key parameters such as velocity, steering angle, and wheel forces, which are crucial for accurate trajectory and speed control in complex environments.

In the inertial coordinate system OXY, the vehicle kinematics equation is described as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \dfrac{\tan\delta}{l} \end{bmatrix} v \tag{1}$$

where x and y are the coordinate of the rear axle center of the vehicle, $\dot{\varphi}$ is the heading angular velocity, δ is the front wheel deflection angle, v is the rear axle speed of the vehicle, and l is the wheelbase *(Pei et al., 2022).*

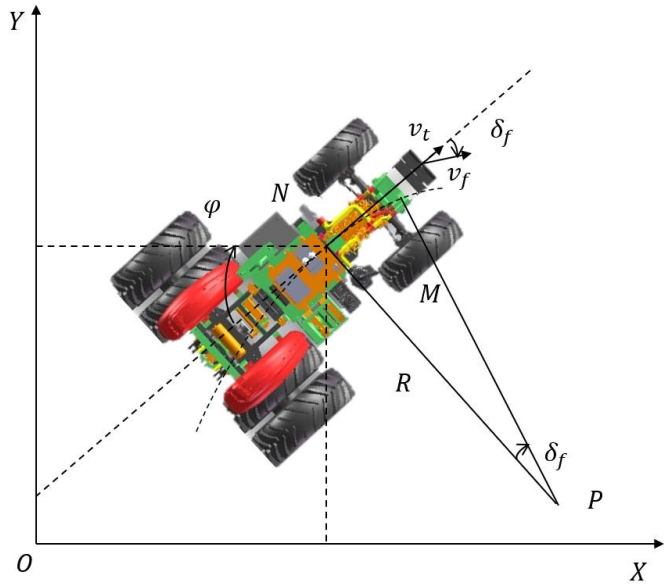


Fig. 1 - Vehicle kinematic model

R is the turning radius, P is the instantaneous steering center of the vehicle, N, M are the contact point of the front wheel and the vehicle centroid, OXY is the inertial coordinate system, v_f velocity component of the front wheel of the vehicle, v_t the velocity component of the vehicle at the centroid, δ_f front wheel angle, φ is the heading angle of the vehicle body.

At the same time, state of charge (SOC) is introduced as the fourth state quantity, which can be expressed as:

$$\dot{soc} = g(soc, v) = -\frac{P_{bat}(v)}{U_{ocv}(soc) \bullet Q_{nom}} \tag{2}$$

where $U_{ocv}(soc), Q_{nom}$ represent the relationship between battery SOC and voltage U_{bat} and the rated capacity of the battery, respectively, which are measured by the actual experiment of the modeled battery prototype. $P_{bat}(v)$ indicates the battery power at current v, which is:

$$P_{bat}(v) = \frac{F_{req}(v) \bullet v}{\eta_{drive}}$$
$$st. \quad F_{req}(v) = f_r mg + \frac{1}{2}\rho C_d A v^2 + mg \sin\theta_a \tag{3}$$

where f_r is rolling resistance coefficient, m is the vehicle mass, g is gravity acceleration, $\rho$is the air density, C_d is the air drag coefficient, A is the windward area, θ_a is the road slope, η_{drive} is the transmission efficiency.

At the same time, the controller proposed in this paper is mainly for the trajectory tracking layer under the condition of low-speed agricultural operation. The lateral sliding angle is relatively small, which can be regarded as bounded interference. In the actual agricultural system, the lower-level traction or speed control loop compensates the sliding effect through the feedback measured by RTK-GPS and IMU. Therefore, in a typical low-speed field operation, using the full tire soil dynamics model will significantly increase the computational burden, but will not improve the tracking performance accordingly. Therefore, the wheel soil interaction and slip mechanics are not modelled.

From the above equation, it can be seen that the system is regarded as an input with $u(v, \delta)$ as state variables and control system of $\chi(x, y, \varphi, soc)$. Thus, the system can be written as:

$$\dot{\chi} = f(\chi, u) \tag{4}$$

For a given reference trajectory, it can be described by the trajectory of the reference vehicle. Each point on it satisfies the above kinematic equation, represented by r as the reference quantity, such that:

$$\dot{\chi}_r = f(\chi_r, u_r) \tag{5}$$

with $\chi_r = [x_r, y_r, \varphi_r, soc_r], u_r = [v_r, \delta_r]$, soc_r is defined as 1 to represent the minimum power consumption of the controller.

Expand equation (5) using Taylor technique at the reference trajectory point and ignore higher-order terms, resulting in:

$$\dot{\chi} = f(\chi_r, u_r) + \frac{\partial f(\chi,u)}{\partial \chi}\Big|_{\substack{\chi=\chi_r \\ u=u_r}} (\chi - \chi_r) + \frac{\partial f(\chi,u)}{\partial u}\Big|_{\substack{\chi=\chi_r \\ u=u_r}} (u - u_r) \tag{6}$$

Subtracting equation (6) from equation (5) yields:

$$\dot{\chi} = \begin{bmatrix} \dot{x}-\dot{x}_r \\ \dot{y}-\dot{y}_r \\ \dot{\varphi}-\dot{\varphi}_r \\ \dot{soc}-\dot{soc}_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_r\sin\varphi_r & 0 \\ 0 & 0 & v_r\cos\varphi_r & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{P_{bat}(v_r)\bullet U'_{ocv}(soc)}{U^2_{ocv}(soc)\bullet Q_{nom}} \end{bmatrix} \begin{bmatrix} x-x_r \\ y-y_r \\ \varphi-\varphi_r \\ soc-soc_r \end{bmatrix} + \begin{bmatrix} \cos\varphi_r & 0 \\ \sin\varphi_r & 0 \\ \dfrac{\tan\varphi_r}{l} & \dfrac{v_r}{l\cos^2\delta_r} \\ \dfrac{f_r mg + \frac{3}{2}\rho C_d A v_r^2 + mg\sin\theta_a}{U_{ocv}(soc)\bullet Q_{nom}\bullet\eta_{drive}} & 0 \end{bmatrix} \tag{7}$$

Equation (7) with the linearized error model is discretized as:

$$\tilde{\chi}(k+1) = A_{k,t}\tilde{\chi}(k) + B_{k,t}\tilde{u}(k) \tag{8}$$

with $A_{k,t=}\begin{bmatrix} 1 & 0 & -v_r\sin\varphi_r T & 0 \\ 0 & 1 & v_r\cos\varphi_r T & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1+\dfrac{P_{bat}(v_r)\bullet U'_{ocv}(soc)}{U^2_{ocv}(soc)\bullet Q_{nom}}T \end{bmatrix}$, $B_{k,t=}\begin{bmatrix} \cos\varphi_r T & 0 \\ \sin\varphi_r T & 0 \\ \dfrac{\tan\varphi_r T}{l} & \dfrac{v_r T}{l\cos^2\delta_r} \\ \dfrac{f_r mg + \frac{3}{2}\rho C_d A v_r^2 + mg\sin\theta_a}{U_{ocv}(soc)\bullet Q_{nom}\bullet\eta_{drive}}T & 0 \end{bmatrix}$, where

T is the sampling time.

Controller design

The algorithm controller is composed of MPC algorithm and DBO algorithm. The MPC part is based on the kinematics model of the unmanned agricultural machine, which predicts the future state and generates the optimal control sequence to achieve trajectory tracking, speed control and SOC control. The DBO part is embedded in the MPC framework as an optimization solver to improve the efficiency and accuracy of the optimization process through its global search and local development capabilities, effectively avoiding falling into local optimization, so as to significantly improve the control performance while ensuring real-time performance. The controller architecture is shown in Figure 2.

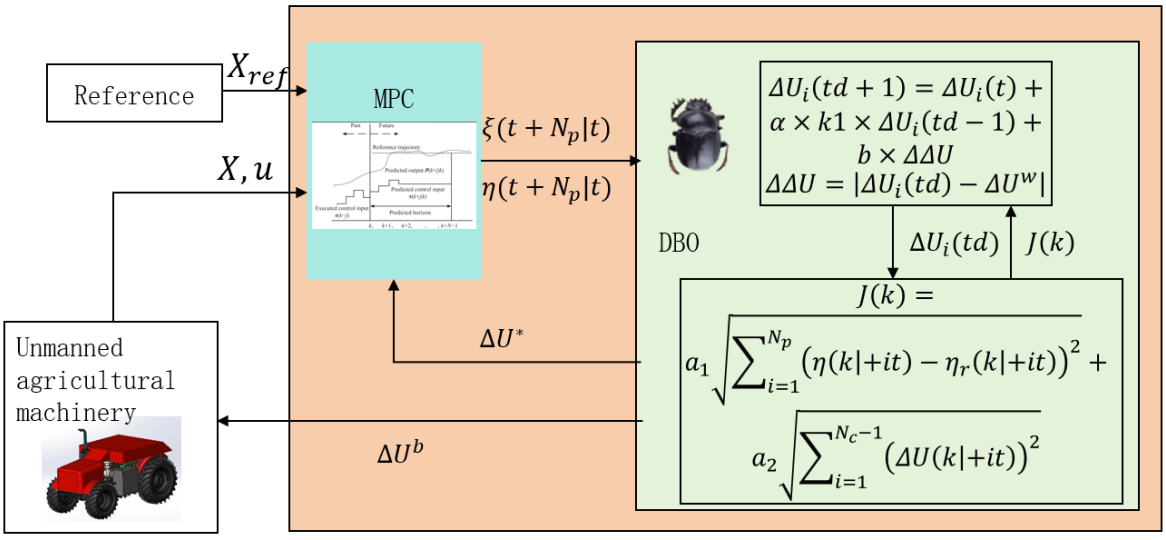


Fig. 2 - Controller architecture

Basic theoretical framework of MPC

Model predictive control mainly includes three components: prediction, optimization, and feedback. In this paper, linearized model predictive control is used as the basic controller *(Ammour et al., 2023)*. Consider the following discrete-time linearized model:

$$x(k+1) = A_{k,t}x(k) + B_{k,t}u(k) \tag{9}$$

Then, the system can be expressed as:

$$\xi(k|t)=\begin{bmatrix} x(k|t) \\ u(k-1|t) \end{bmatrix} \tag{10}$$

Accordingly, a new state-space representation can be formulated as:

$$\begin{cases} \xi(k+1|t)=\tilde{A}_{k,t}\xi(k|t)+\tilde{B}_{k,t}\Delta u(k|t) \\ \eta(k|t)=\tilde{C}_{k,t}\xi(k|t) \end{cases} \tag{11}$$

The definitions of matrices in the equation are as follows:

$$\tilde{A}_{k,t}=\begin{bmatrix} A_{k,t} & B_{k,t} \\ 0_{m\times n} & I_m \end{bmatrix},\tilde{B}_{k,t}=\begin{bmatrix} B_{k,t} \\ I_m \end{bmatrix}\tilde{C}_{k,t}=\begin{bmatrix} C_{k,t} & 0 \end{bmatrix} \tag{12}$$

Assume the predicted time domain of the system is N_p, and the control time domain is N_c, then the state variables and system output variables within the predicted time domain can be calculated using the following equation:

$$\begin{cases} \xi(t+N_p|t)=\tilde{A}_t^{N_p}\xi(t|t)+\tilde{A}_t^{N_p-1}\tilde{B}_t\Delta u(t|t)+\cdots\tilde{A}_t^{N_p-N_c-1}\tilde{B}_t\Delta u(t+N_c|t) \\ \eta(t+N_p|t)=\tilde{C}_{t,t}\tilde{A}_{t,t}^{N_p}\xi(t|t)+\tilde{C}_t\tilde{A}_t^{N_p-1}\tilde{B}_t\Delta u(t|t)+\cdots\tilde{C}_t\tilde{A}_t^{N_p-N_c-1}\tilde{B}_t\Delta u(t+N_c|t) \end{cases} \tag{13}$$

To make the entire relationship clearer, the output of the system at future moments is expressed as *(Liu et al., 2022)*:

$$Y(t)=\psi_t\xi(t|t)+\Theta_t\Delta U(t) \tag{14}$$

With

$$Y(t)=\begin{bmatrix} \eta(t+1|t) \\ \eta(t+2|t) \\ \cdots \\ \eta(t+N_c|t) \\ \cdots \\ \eta(t+N_p|t) \end{bmatrix},\ \psi_t=\begin{bmatrix} \tilde{C}_t\tilde{A}_t \\ \tilde{C}_t\tilde{A}_t^2 \\ \cdots \\ \tilde{C}_t\tilde{A}_t^{N_c} \\ \cdots \\ \tilde{C}_t\tilde{A}_t^{N_p} \end{bmatrix},\ \Delta U(t)=\begin{bmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \cdots \\ \Delta u(t+N_c|t) \end{bmatrix}\Theta_t=\begin{bmatrix} \tilde{C}_t\tilde{B}_t & 0 & 0 & 0 \\ \tilde{C}_t\tilde{A}_t\tilde{B}_t & \tilde{C}_t\tilde{B}_t & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots \\ \tilde{C}_t\tilde{A}_t^{N_c-1}\tilde{B}_t & \tilde{C}_t\tilde{A}_t^{N_c-2}\tilde{B}_t & \cdots & \tilde{C}_t\tilde{B}_t \\ \tilde{C}_t\tilde{A}_t^{N_c}\tilde{B}_t & \tilde{C}_t\tilde{A}_t^{N_c-1}\tilde{B}_t & \cdots & \tilde{C}_t\tilde{A}_t\tilde{B}_t \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_t\tilde{A}_t^{N_p-1}\tilde{B}_t & \tilde{C}_t\tilde{A}_t^{N_p-2}\tilde{B}_t & \cdots & \tilde{C}_t\tilde{A}_t^{N_p-N_c-1}\tilde{B}_t \end{bmatrix}$$

Based on the above formulation, the prediction function is established.

After each optimization cycle, a sequence of control increments over the control horizon can be obtained:

$$\Delta U_t^*=\left[\Delta u_t^*,\Delta u_{t+1}^*,\cdots\Delta u_{t+N_c-1}^*\right]^T \tag{15}$$

The first element of the optimized control sequence is then selected as the control increment applied to the system, that is:

$$u(t)=u(t-1)+\Delta u_t^* \tag{16}$$

By imposing constraints on both the control variables and the control increments, the following constrained form is obtained:

$$\begin{cases} U_{\min}(k+i)\le U(k+i)\le U_{\max}(k+i) \\ \Delta U_{\min}(k+i)\le\Delta U(k+i)\le\Delta U_{\max}(k+i) \end{cases} \tag{17}$$

This procedure is repeated at each sampling instant to realize the receding-horizon feedback control of the system.

This paper aims to solve a multi-objective optimization problem, where the objective function must ensure both rapid and smooth trajectory tracking by the autonomous vehicle and minimize the reduction in battery SOC. To achieve these goals, the following objective function is described as:

$$J(k)=a_1\sqrt{\sum_{i=1}^{N_p}(\eta(k+i|t)-\eta_r(k+i|t))^2}+a_2\sqrt{\sum_{i=1}^{N_c-1}(\Delta U(k+i|t))^2}+a_3SOC_e \tag{18}$$

where SOC_e is the minimum SOC, and a_1, a_2, a_3 are the weight coefficients.

Multi objective optimization method based on D-MPC

The Dung Beetle Optimizer (DBO) is an iterative optimization algorithm that mimics the rolling behavior of dung beetles in nature *(Xue et al., 2023)*. By using multiple different beetle populations to cover the entire search range, it can effectively solve the problem of local optima in MPC. The schematic diagram of the DBO algorithm is shown in Fig. 3.

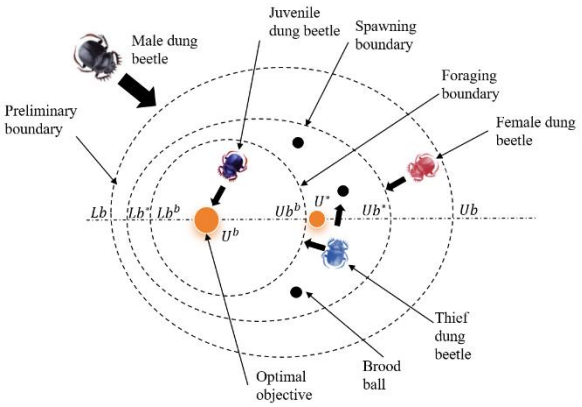


Fig. 3 - Algorithm flowchart of DBO

To simulate rolling behavior, the beetle needs to move along a given direction throughout the entire search space. The beetle uses sunlight to navigate, and it is assumed that the intensity of the light source also affects the beetle's path. During the rolling process, the beetle's position is updated, represented as *(Zhong et al., 2024)*:

$$U_i(t_d+1) = U_i(t) + \alpha \times k1 \times U_i(td-1) + b \times \Delta U$$
$$\Delta U = |U_i(td) - U^w| \tag{19}$$

where t_d represents the current iteration number, U_i denotes the position information of the i beetle at the t_d iteration, $k_1 \epsilon (0,0.2]$ is a constant represented as a deflection coefficient, b represents a constant belonging to (0,1), α is a natural coefficient assigned to 1 or -1, U^w is the global worst position, and ΔU simulates changes in light intensity.

Upon encountering obstacles that block its forward movement, the dung beetle repositions itself by performing a dance and then selects a new route, as represented by a tangent function:

$$U_i(t_d+1) = U_i(t_d) + \tan(\theta)|U_i(t_d) - U_i(t_d-1)| \tag{20}$$

where θ is the deflection angle belonging to $[0,\pi]$, and when θ is equal to 0, $\pi/2$, or π.

The male dung beetle searches for food, and the female, based on its natural behavior, selects a suitable spawning site. The spawning site boundary is determined using a boundary selection strategy, as described by the following formula:

$$\begin{cases} Lb^* = \max(U^* \times (1-R), Lb) \\ Ub^* = \min(U^* \times (1+R), Ub) \end{cases} \tag{21}$$

where U^* is the local optimal position, Lb^* and Ub^* are the lower and upper bounds of the spawning area respectively, $R = 1 - t_d/T_{max}$, T_{max} is the maximum number of iterations, Lb and Ub correspond to the lower and upper bounds of the optimization problem, respectively.

Male beetles search for food, while female beetles select spawning areas for egg-laying. As shown in equation (19), the boundary of the spawning area dynamically changes, primarily determined by the R value. The position of the hatching ball is also dynamic and is defined as follows:

$$B_i(t_d+1) = U^* + b_1 \times (B_i(t_d) - Lb^*) + b_2 \times (B_i(t_d) - Ub^*) \tag{22}$$

where $B_i(t_d)$ represents the position information of the i brood ball at the t_d iteration, b_1 and b_2 represent two independent random vectors of size $1 \times D$, D represents the dimension of the optimization problem.

After growing up, the small dung beetle will emerge from the ground to forage, and the boundary of the foraging area is defined as:

$$\begin{cases} Lb^b = \max(U^b \times (1-R), Lb) \\ Ub^b = \min(U^b \times (1+R), Ub) \end{cases} \tag{23}$$

where U^b is the global optimal position, Lb^b and Ub^b represent the lower and upper bounds of the optimal foraging area, respectively.

Therefore, the position of the small beetle has been updated as follows:

$$U_i(t_d+1) = U_i(t_d) + C_1 \times (U_i(t_d) - Lb^b) + C_2 \times (U_i(t_d) - Ub^b) \quad (24)$$

where $U_i(t_d)$ represents the position information of the i small beetle at the t_d iteration, C_1 represents a random number that follows a normal distribution, and C_2 represents a random vector belonging to $(0,1)$.

On the other hand, there may be a thief beetle stealing the ball from other beetles. It is assumed that the global optimal position U^b is the best food competition point, and the thief's position is updated as follows:

$$U_i(t_d+1) = U^b + S \times g \times (\left|U_i(t_d) - U^*\right| + \left|U_i(t_d) - U^b\right|) \quad (25)$$

where $U_i(t_d)$ represents the position information of the i thief at the t_d iteration, g is a random vector of size $1 \times D$, following a normal distribution, and S represents a constant.

Based on equations (7) to (23), the proposed algorithm combines the Dung Beetle Optimizer (DBO) and Model Predictive Control (MPC) algorithms. The DBO initializes the control input matrix, which is then fed into the MPC controller. The MPC controller computes the cost function matrix, which is subsequently input into the DBO. The hybrid algorithm approaches the global optimal solution from four distinct directions, progressively narrowing the boundary of the search region to accelerate the convergence rate. Finally, through multiple iterations, the current optimal control input is obtained by combining the value function.
The calculation process of fusion algorithm is shown in Table 1 *(Xue et al., 2023)*.

Table 1

D-MPC algorithm flow

Input: The maximum iterations T_{max}, the size of the particle's population N.
Output: Current suboptimal control quantity U^* and optimal value f^*; Optimal control quantity U^b and its fitness value f_b.

1. Initialize the particle's population $U_0(i)$ $i(1,2,\dots,N)$ and define its relevant parameters.
2. While $(t \le T_{max})$ do
3. For $i(1,2,\dots,N)$ do
4. $f(i) = J(i)$ as equation (18)
5. Select the optimal U^* through the optimal f^*
6. Calculate the control quantity according to (19) and (20)
7. Update $f(i) = J(i)$ as equation (18)
8. Obtain the optimal f^* and update the optimal U^*
9. Update the optimization boundary according to (21)
10. Repeat steps 6-9 according to (22) - (25)
11. If the newly control quantity is better than before then
12. Update it
13. End if
14. $t = t + 1$
15. end while
16. return U^b and its fitness value f_b

SIMULATION AND EXPERIMENT

Simulation design

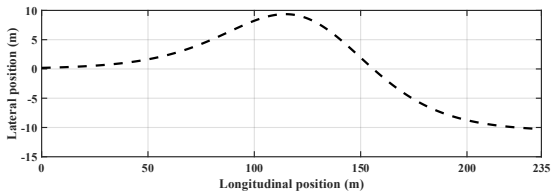
In this part, the performance of the designed control algorithm is verified through two simulation conditions, in order to comprehensively evaluate the target ability of the controller to the unmanned agricultural machinery in different scenarios. The double lane shifting mode shown in Fig.4 (a) is used to evaluate the working performance of the unmanned agricultural machine under road driving and transfer conditions, while the U-turn mode shown in Fig. 4 (b) is used to simulate the working performance of the unmanned agricultural machine under field operation scenarios. Fig. 4 (c) shows the CarSim Simulink co-simulation scenario. The software versions used are MATLAB 2022a and CarSim 2019.1. The controller parameters for simulation are shown in Table 2.

Table 2

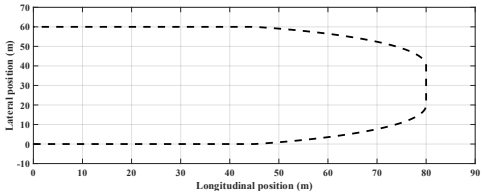
Controller parameters

Parameter category	Parameter name	Parameter value
Controller parameters	Prediction step	20
	Control step size	10
	Dung beetle population	12
	Iterations	20
	Solving dimensions	2

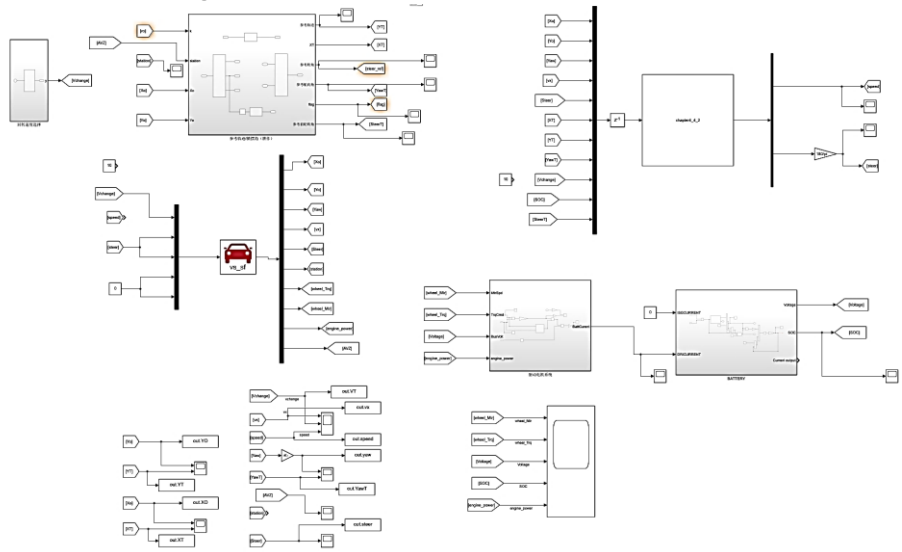
To show the superiority of the strategy developed in this paper, Grey Wolf Optimizer-based Model Predictive Control (G-MPC) *(Elsisi et al., 2024)*, and the traditional MPC controller are introduced as the control group. The differences of tracking accuracy, iterative convergence speed and calculation speed are compared.



(a) - Double lane-change



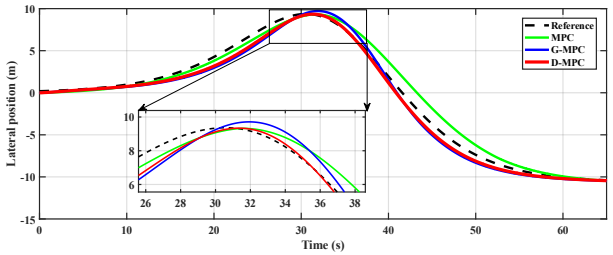
(b) - U-turn path simulation



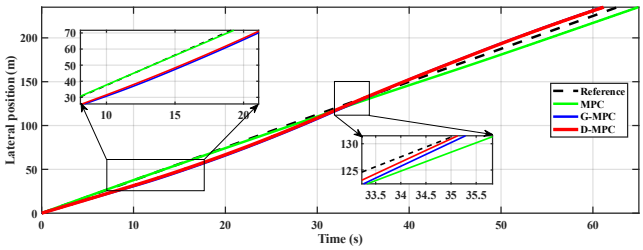
(c) - Simulation process

Fig. 4 - Reference curve chart and simulation scenario

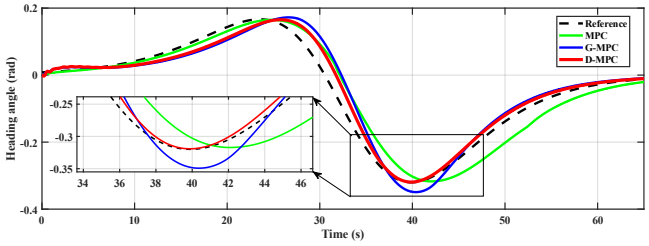
Fig. 5 illustrates the tracking performance in $y(t), x(t)$, and heading angle $\varphi(t)$. In the lateral position (a), MPC (blue) exhibits the largest deviation, with a peak error of 1.1 m. G-MPC (green) reduces the peak error to about 0.6 m, while D-MPC (red) achieves the best performance with a peak error in the range of 0.3–0.4 m. For the longitudinal position (b), the maximum tracking errors of the three methods are similar; however, D-MPC and G-MPC demonstrate improved real-time adjustment capability, resulting in smoother tracking behavior. In terms of heading angle (c), MPC yields a maximum error of approximately 0.06–0.07 rad, G-MPC reduces it to ~0.03 rad, and D-MPC further limits the error to 0.015–0.02 rad. Overall, D-MPC exhibits the lowest peak errors and smallest deviations, demonstrating its superior trajectory tracking performance under double lane-change conditions.



(a) - Lateral position comparison



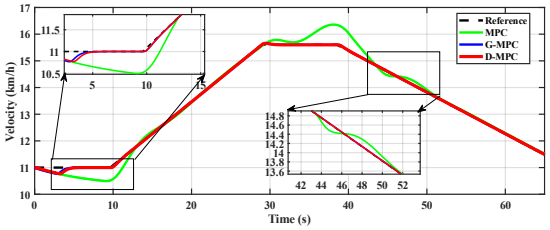
(b) - Longitudinal position comparison



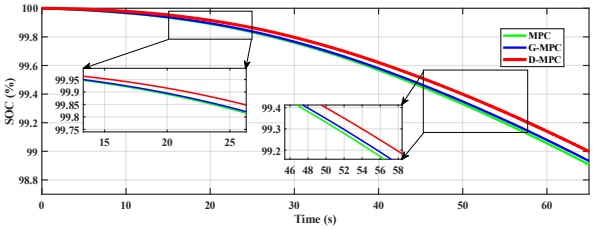
(c) - Heading angle comparison

Fig. 5 - Double lane-change trajectory tracking comparison

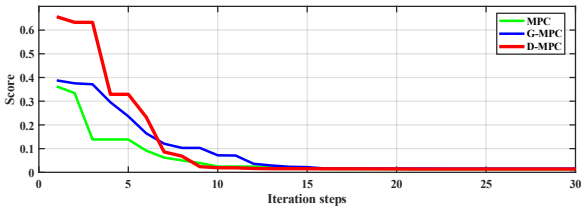
Under double lane-change conditions, the performance in terms of velocity tracking (Fig. 6a), SOC evolution (Fig. 6b), algorithm convergence (Fig. 6c), and computational time (Fig. 6d) was analyzed. As shown in Fig. 6a, the reference velocity increases from 11 km/h to 15.7 km/h at approximately 30 s, followed by a deceleration phase. MPC (blue) exhibits the largest overshoot, reaching approximately 0.8 km/h at the peak. G-MPC (green) significantly reduces the overshoot to about 0.05 km/h, while D-MPC (red) demonstrates a similar but more stable response. Figure 6b presents the SOC evolution curves. After 60 s, the SOC decreases to 98.89% for MPC, 98.95% for G-MPC, and 98.99% for D-MPC, indicating that D-MPC reduces energy consumption by approximately 0.1% compared to MPC. As illustrated in Fig. 6c, D-MPC reaches the target cost value within 9 iterations, whereas G-MPC and MPC require 12 and 15 iterations, respectively. Moreover, D-MPC maintains the lowest cost per iteration beyond the ninth iteration, indicating faster and more stable convergence. Figure 6d shows the computational time of each method. G-MPC has the longest solving time, approximately 0.03 s, while D-MPC requires about 0.018 s, which is 0.008 s longer than MPC. Overall, D-MPC achieves superior velocity tracking performance, slower SOC degradation, and faster convergence, with only a slight increase in computational time compared to MPC. These results demonstrate its advantages in control accuracy and energy efficiency.



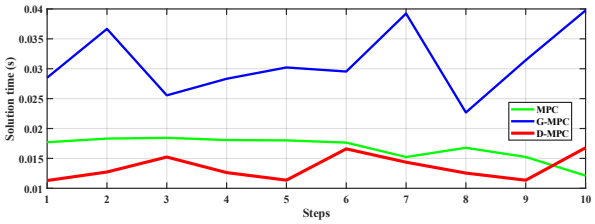
(a) - Velocity comparison



(b) - SOC comparison



(c) - Algorithm convergence comparison



(d) - Solution time

Fig. 6 - Comparison of other objectives of double lane-change

The root mean square error (RMSE) is an effective metric for evaluating overall tracking accuracy, with lower values indicating better performance. Table 3 presents a comparison of six performance indicators under double lane-change conditions. The results show that, compared with MPC, the D-MPC algorithm reduces the RMSE by 0.54 and 1.9 in lateral and longitudinal tracking errors, respectively. Compared with G-MPC, the reductions are 0.16 and 0.7, respectively. In terms of heading angle, D-MPC achieves further improvements, with RMSE reductions of 0.019 and 0.005 compared with MPC and G-MPC, respectively. Regarding speed stability, D-MPC exhibits slightly higher RMSE values, by 0.28 and 0.12 compared with MPC and G-MPC, respectively, indicating a minor trade-off in velocity smoothness. Overall, these results further demonstrate the advantages of the D-MPC approach in trajectory tracking accuracy, convergence performance, and control effectiveness.

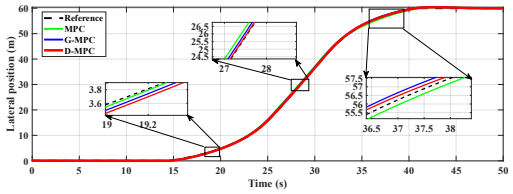
Table 3

Algorithm comparison table of double lane-change

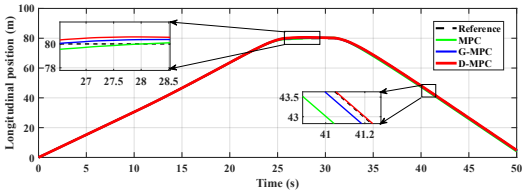
	MPC	G-MPC	D-MPC
Lateral RMSE (m)	0.92	0.54	0.38
Longitudinal RMSE (m)	3.7	1.8	1.1
Heading RMSE (rad)	0.031	0.017	0.012
Velocity RMSE (km/h)	0.37	0.21	0.09
Final SOC (%)	98.89	98.95	98.99
Iterations to Converge	15	12	9
Average solution time (s)	0.018	0.033	0.014

U-turn path simulation

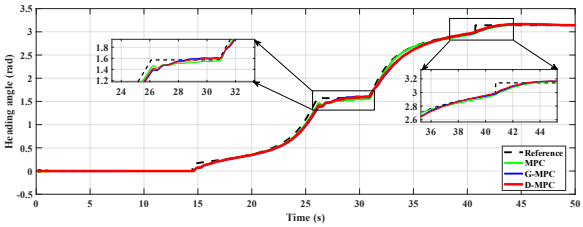
According to the results of the tractor working path, Fig. 7a–c shows the lateral position, longitudinal position, and heading angle tracking results. In Fig. 7a, D-MPC (red) best matches the reference lateral, with a maximum error below 0.3 m, while G-MPC and MPC show errors of 0.4 m and 0.45 m, respectively. In Fig.7b, D-MPC maintains the smallest deviation for longitudinal tracking, within 0.02 m, compared to 0.15 m for G-MPC and 0.8 m for MPC. In Fig.7c, D-MPC reaches the target heading angle faster, with a maximum error around 0.2 rad, demonstrating better response speed and stability.



(a) - Lateral position comparison



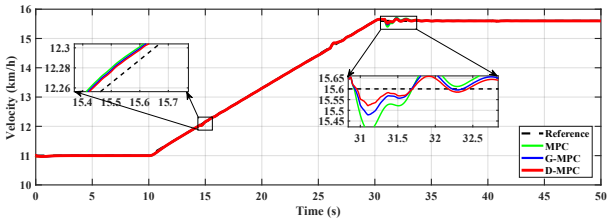
(b) - Longitudinal position comparison



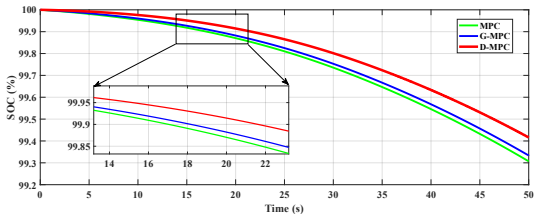
(c) - Heading angle comparison

Fig. 7 - U-turn path simulation trajectory tracking comparison

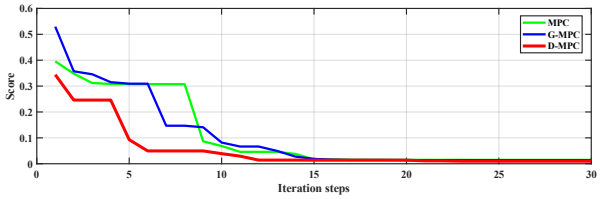
Further analysis in Fig. 8a–c presents speed tracking, SOC change, and algorithm performance. In Fig. 8a, D-MPC (red) limits speed tracking error to ±0.075 km/h, achieving faster stable speed and smaller amplitude compared to G-MPC (0.12 km/h) and MPC (0.2 km/h). In Fig. 8b, D-MPC maintains the highest SOC at 99.41%, reducing energy loss compared to MPC (99.3%) and G-MPC (99.33%). Fig. 8c shows D-MPC reaches convergence in 12 iterations, faster than G-MPC (14) and MPC (16), with the lowest output iteration cost. In Fig. 8d, under U-turn conditions, D-MPC has the fastest solving speed at 0.018s, compared to G-MPC (0.035s) and MPC (0.02s). Overall, D-MPC demonstrates the best control performance under the U-turn path simulation.



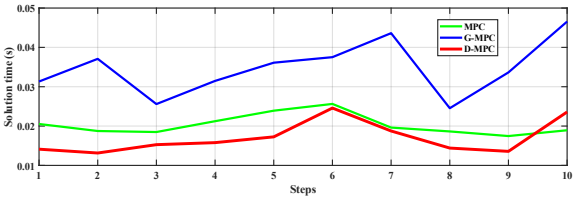
(a) - Velocity comparison



(b) - SOC comparison



(c) - Algorithm convergence comparison



(d) - Solution time

Fig. 8 - Comparison of other objectives of U-turn path simulation

Table 4 shows the comparison of the above six parameters under the U-turn path. In terms of horizontal and longitudinal trajectory tracking error, D-MPC increased by 0.6 compared with MPC, 0.3 compared with G-MPC, 0.035 and 0.015 in heading angle, and 0.28 and 0.12 in speed stability. It can be seen that the controller proposed in this paper can achieve better control effect, control accuracy and stability under both curves.

Table 4

Algorithm comparison table of U-turn path simulation

	MPC	G-MPC	D-MPC
Lateral RMSE (m)	0.9	0.6	0.3
Longitudinal RMSE (m)	0.8	0.5	0.2
Heading RMSE (rad)	0.05	0.03	0.015
Velocity RMSE (km/h)	0.38	0.22	0.1
Final SOC (%)	99.3	99.33	99.41
Iterations to Converge	16	14	12
Average solution time (s)	0.02	0.035	0.015

Experimental design

The experimental validation was conducted using a four-wheel-drive unmanned vehicle that emulates a pure electric tractor, equipped with a battery pack, drive-by-wire control, and a rear suspension system carrying a simulated plough. The positioning equipment brand used in this experiment is FDI Systems, the model is sigma-d4g, and RTK dual antenna inertial navigation integrated navigation is used. In RTK mode, the positioning accuracy is 0.01M, the speed accuracy is 0.01 M/s, the maximum data refresh frequency is 500 Hz, and it supports a variety of communication modes such as can, RS232, RS422, USB, etc. The experimental computer used in this paper is Alienware x15 R2, which contains rtx3070ti and i9-12900h. The experiment was conducted on a flat dirt road, with a hydraulic plow mounted at the rear of the vehicle.

The chosen test scenario was the double lane-change path (Fig. 9), designed to impose rapid lateral transitions and varying speed demands. This setup allows simultaneous evaluation of trajectory tracking, velocity control, and SOC behavior under realistic load conditions, in order to provide a fair basis for comparing the three algorithms in the actual operation, in order to distinguish from the simulation results, the names of the three algorithms in the experiment were changed to MPC, GWO-MPC and DBO-MPC. The experimental results better reflect the effectiveness of the algorithm proposed in this paper in the actual working process compared to simulation. The experimental equipment is shown in Fig.10.

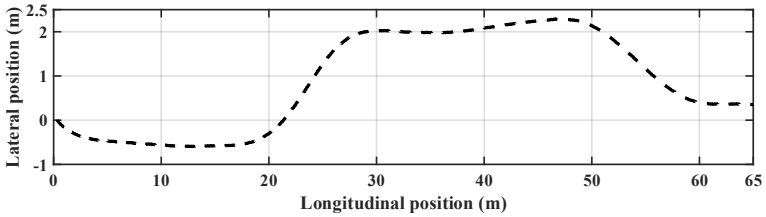


Fig. 9 - Experimental path

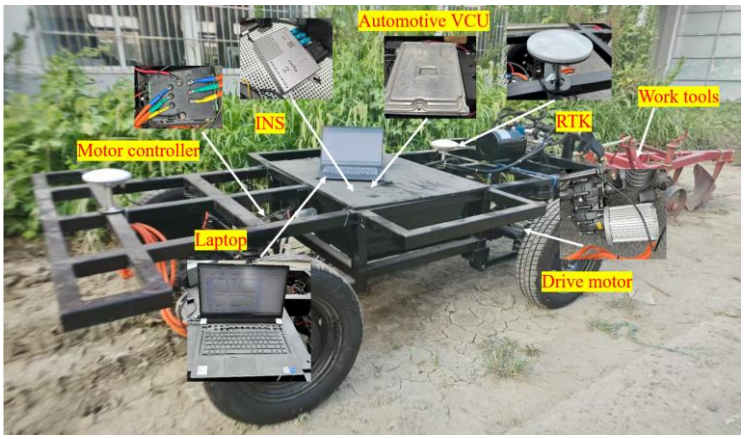
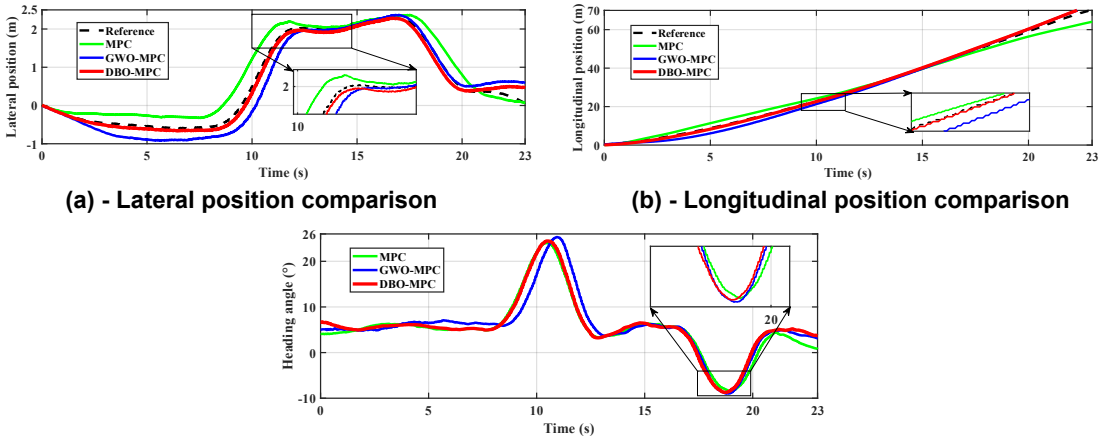


Fig. 10 - Experimental equipment

Fig.11a–c compare lateral, longitudinal, and heading tracking. In Fig. 11a, the reference lateral offset peaks at 2.3 m. MPC (green) shows the largest error (>0.25 m), GWO-MPC (blue) reduces it to ~0.18 m, while DBO-MPC (red) achieves the best tracking with deviations under 0.10 m. Ignoring the possibility of the tail of the curve exiting and ending the automatic control state in Fig. 11b, longitudinal errors reach 1.2 m with MPC and 0.8 m with GWO-MPC, whereas DBO-MPC consistently maintains errors less than 0.3 m.

Fig. 11c shows heading responses, based on the tracking accuracy and heading angle change curve mentioned earlier, it can be seen that DBO-MPC has better trajectory tracking accuracy and heading angle change. MPC and DBO-MPC exhibit similar heading angle curves but are ineffective in controlling trajectory tracking errors. While GWO-MPC responds later in heading angle control, it achieves better trajectory tracking accuracy than MPC.

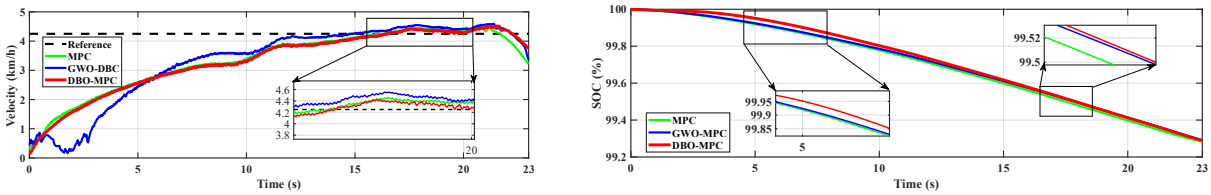


(a) - Lateral position comparison **(b) - Longitudinal position comparison**

(c) - Heading angle comparison

Fig.11 - Experimental path trajectory tracking comparison

As shown in Fig.12a, it can be seen that GWO-MPC reaches the desired speed faster, but with a larger overshoot (~0.35 km/h). Although DBO-MPC reaches the desired speed the latest, its acceleration trajectory is smooth and has a minimum overshoot of less than 0.15 km/h, indicating the advantage of DBO-MPC in control stability. The vehicle based on DBO-MPC control shown in Fig.12b has a smoother SOC decrease. Although GWO-MPC exhibits significant variations in vehicle speed, its superior tracking accuracy results in minimal differences in SOC decline. However, it can be seen that the difference between the SOC results of each algorithm is small, because the duration of this experiment is about 20 seconds, which belongs to a short-term discharge process relative to the battery capacity, so the overall small change range of SOC is expected. Although the absolute value of SOC difference under different working conditions is limited, the SOC curve shows a stable and consistent change trend in the whole test process, and is consistent with the change law of current and power. SOC is essentially the integral result of energy consumption, so even in a short time scale, its change can still reflect the relative difference of energy consumption under different operating conditions. In addition, SOC changes persist at multiple sampling points, indicating that the difference is recognizable, rather than random measurement noise. Therefore, the SOC change trend can be used as an auxiliary basis for analyzing the energy consumption characteristics of the system.



(a) - Velocity comparison **(b) SOC comparison**

Fig. 12 - Comparison of other objectives of experimental path

The above experiments were repeated three times in total. Table 5 shows the mean and standard deviation of the parameters of the experimental results.

Table 5

Algorithm comparison table of experimental path

	MPC	G-MPC	D-MPC
Lateral RMSE (m)	0.25±0.03	0.18±0.02	0.1±0.02
Longitudinal RMSE (m)	0.8±0.05	0.6±0.06	0.3±0.04
Velocity RMSE (km/h)	0.19±0.01	0.11±0.01	0.05±0.02
Final SOC (%)	99.3	99.312	99.315

CONCLUSIONS

This paper proposes a multi-objective optimization algorithm based on Dung Beetle Optimization–based Model Predictive Control (D-MPC) to address the multi-objective control problem of unmanned agricultural vehicles operating in complex farmland environments. The experimental results demonstrate that, compared with conventional control methods, D-MPC improves trajectory tracking accuracy by approximately 50% and 75% in two representative simulation scenarios, while reducing the battery State of Charge (SOC) drop by 0.1% and 0.12%, respectively. In real-vehicle experiments, trajectory tracking accuracy is improved by 70%, and the SOC drop is reduced by 0.015%. These results indicate that the proposed method effectively enhances operational accuracy while reducing energy consumption.

However, it should be noted that under extreme soil conditions (e.g., loose or muddy terrain), wheel slip may degrade tracking performance. Therefore, future work will focus on incorporating slip rate estimation or a tire–soil interaction model into the control framework. In addition, the proposed method will be further validated under more complex agricultural operating conditions through real-vehicle experiments, and compared with more advanced optimization strategies to comprehensively evaluate its practical applicability.

ACKNOWLEDGEMENT

This research was supported by the National Natural Science Foundation of China (52202461) and the Zhejiang provincial postdoctoral research program (ZJ2025143).

REFERENCES

[1] Almufti S.M., Ahmad H.B., Marqas R.B., Asaad R.R. (2021). Grey wolf optimizer: Overview, modifications and applications [J]. *International Research Journal of Science, Technology, Education, and Management*, Vol. 1, pp. 44-56, Iraq.

[2] Ammour M., Orjuela R., Basset M. (2022). A MPC combined decision making and trajectory planning for autonomous vehicle collision avoidance[J]. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, pp. 24805-24817, France.

[3] Bayat S., Allison J.T. (2023). LGR-MPC: A user-friendly software based on Legendre-Gauss-Radau pseudo spectral method for solving Model Predictive Control problems [J]. *arXiv preprint*, Vol. 10, USA.

[4] Chen L. (2023). Path planning of fruit and vegetable picking robots based on improved a* algorithm and particle swarm optimization algorithm (基于改进 A*算法与粒子群算法的果蔬采摘机器人路径规划) [J]. *INMATEH-Agricultural Engineering*, Vol. 71, pp. 470-482.

[5] Chen Z., Lai J., Li P., Omar I., Zhu Y. (2024). Prediction horizon-varying model predictive control (MPC) for autonomous vehicle control (基于预测时域变模型预测控制的自主车辆控制) [J]. Electronics, Vol. 8, pp. 1442-1450.

[6] Chu D., Li H., Zhao C., Zhou T. (2022). Trajectory tracking of autonomous vehicle based on model predictive control with PID feedback (基于 PID 反馈模型预测控制的自主车辆轨迹跟踪) [J]. IEEE *Transactions on Intelligent Transportation Systems*, Vol. 2, pp. 2239-2250, Wuhan/China.

[7] Elsisi M. (2024). Optimal design of adaptive model predictive control based on improved GWO for autonomous vehicle considering system vision uncertainty[J]. Applied Soft Computing, Vol. 158, Egypt.

[8] Gong L, Yu M, Kollias S. (2023). Optimizing crop yield and reducing energy consumption in greenhouse control using PSO-MPC algorithm [J]. *Algorithms*, Vol. 16, pp. 243-254, UK.

[9] Wang X., Wei Y., Guo Z., Wang J., Hu B. (2024). A sinh–cosh-enhanced DBO Algorithm applied to global optimization problems (一种应用于全局优化问题的 sinh-cosh 增强 DBO 算法) [J]. *Biomimetics*, Vol. 9, pp. 271-301, Yunnan/China.

[10] He J., Guo W., Wang S., Chen H., Li S. (2024). Application of multi-strategy based improved DBO algorithm in optimal scheduling of reservoir groups (基于多策略的改进 DBO 算法在水库群优化调度中的应用) [J]. *Water Resources Management*, Vol. 6, pp. 1883-1901, Zhengzhou/China.

[11] Hou W., Xiong Z., Wang C., Chen H. (2022). Enhanced ant colony algorithm with communication mechanism for mobile robot path planning (基于通信机制的改进蚁群算法在移动机器人路径规划中的应用) [J]. *Robotics and Autonomous Systems*, Vol. 148, Dalian/China.

[12] Kim S., Lee J., Han K., Choi S.B. (2023). Vehicle path tracking control using pure pursuit with MPC-based look-ahead distance optimization [J]. *IEEE Transactions on Vehicular Technology*, Vol. 73, pp. 53-66, South Korea.

[13] Liu W.J., Ding H.F., Ge M.F., Yao X.Y. (2022). Cooperative control for platoon generation of vehicle-to-vehicle networks: a hierarchical nonlinear MPC algorithm (车-车网络排生成的协同控制：一种分层非线性 MPC 算法) [J]. *Nonlinear dynamics*, Vol. 108, pp. 3561-3578, Wuhan/China.

[14] Luo X., Liao J., Hu L., Zhou Z., Zhang Z., Zang Y., Wang P., He J. (2021). Research progress of intelligent agricultural machinery and practice of unmanned farm in China (中国智能农机研究进展与无人农场实践) [J]. *CABI Digital Library*. Vol. 2, pp. 8-17, China.

[15] Moazen H., Molaei S., Farzinvash L., Sabaei M. (2023). PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator[J]. *Information Sciences*, Vol. 628, pp. 70-91, Canada.

[16] Oliveira T., Aguiar A.P., Encarnacao P. (2016). Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems[J]. *IEEE Transactions on Robotics*, Vol. 32, pp. 1062-1078, Portugal.

[17] Pei-cheng S., Li L., Ni X., Yang A. (2022). Intelligent vehicle path tracking control based on improved MPC and hybrid PID (基于改进 MPC 和混合 PID 的智能车辆路径跟踪控制) [J]. *IEEE Access*, Vol. 10, pp, 94133-94144. Wuhu/China.

[18] Rokonuzzaman M., Mohajer N., Nahavandi S. (2023). Effective adoption of vehicle models for autonomous vehicle path tracking: a switched MPC approach[J]. *Vehicle system dynamics*, Vol. 61, pp. 1236-1259, Australia.

[19] Seyyedabbasi A., Kiani F. (2021). I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems [J]. *Engineering with Computers*, Vol. 37, pp. 509-532, Turkey.

[20] Shi M., Feng X., Pan S., Song X., Jiang L. (2023). A collaborative path planning method for intelligent agricultural machinery based on unmanned aerial vehicles (基于无人机的智能农机协同路径规划方法) [J]. *Electronics*, Vol. 12, pp. 3232-3250, Zhenjiang/China.

[21] Xue J., Shen B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization (粪甲虫优化器：一种新的全局优化元启发式算法) [J]. *The Journal of Supercomputing*, Vol. 7, pp, 7305-7336. Shanghai/China.

[22] Yu S., Hirche M., Huang Y., Chen H., Allgower F. (2021). Model predictive control for autonomous ground vehicles: a review (地面自主车辆模型预测控制研究进展) [J]. *Autonomous Intelligent Systems*, Vol. 1, pp. 1-17, Jilin/China.

[23] Zhai L., Wang C., Hou Y., Liu C. (2022). MPC-based integrated control of trajectory tracking and handling stability for intelligent driving vehicle driven by four hub motor (基于 MPC 的四轮毂电机驱动智能驾驶车辆轨迹跟踪与操纵稳定性集成控制) [J]. *IEEE transactions on vehicular technology*, Vol. 3, pp. 2668-2680, Beijing/China.

[24] Zhang K., Sun Q., Shi Y. (2021). Trajectory tracking control of autonomous ground vehicles using adaptive learning MPC[J]. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, pp. 5554-5564, Canada.

[25] Zhang Z. (2020). Optimal analysis of farm agricultural machinery equipment based on mathematical modelling (基于数学建模的农机设备优化分析) [J]. *INMATEH-Agricultural Engineering*, Vol. 62, pp. 333-340.

[26] Zhen Q., Chengqian Jin., Zheng Liu, Teng-xiang Y. (2023). Development status and trends of intelligent control technology in unmanned farms (无人农场智能控制技术的发展现状与趋势) [J]. *Journal of Intelligent Agriculture Mechanization*, Vol. 4, China.

[27] Zhong K., Li Y., Huan W., Weng X., Wu B., Chen Z., Liang H., Feng H. (2024). A novel near infrared spectroscopy analytical strategy for soil nutrients detection based on the DBO-SVR method (基于 DBO-SVR 的土壤养分近红外光谱检测新策略) [J]. Spectrochimica Acta Part A: *Molecular and Biomolecular Spectroscopy*, Vol. 315, pp. 124-135, Zhejiang/China.