BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(1), 2026, Article number: e156797
DOI: 10.24425/bpasts.2025.156797

MECHANICAL AND AERONAUTICAL ENGINEERING, THERMODYNAMICS

Improved snake optimization algorithm for parameter identification based on the genetic algorithm

Baolu YANG*, Liangming WANG, and Jian FU

College of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Abstract. To address the issue that traditional snake optimization (SO) algorithms tend to become trapped in local optima when identifying aerodynamic parameters of high-spin projectiles – where complex flight dynamics and measurement noise further complicate the process – this paper proposes an enhanced snake optimization algorithm integrated with genetic algorithm (GA) mechanisms. Specifically, the improved algorithm incorporates GA-based selection and crossover operations into the SO framework, aiming to strengthen global search capability by simulating not only snakes' natural foraging and combat behaviors but also the evolutionary characteristics of genetic algorithms. For handling noisy trajectory data, Kalman filtering is applied to denoise measured information, laying a reliable foundation for subsequent parameter identification. The method utilizes segmented trajectory data of high-spin projectiles across different speed stages for analysis. Comparative experiments with the traditional SO algorithm and other optimized variants demonstrate that the proposed approach reduces identification errors by 49%, significantly outperforming conventional methods in accuracy. Further validation with full trajectory measured data shows that when the identified aerodynamic parameters are substituted into ballistic equations, the deviation between calculated and actual impact point coordinates is minimal, confirming their effectiveness. Notably, the improved algorithm does not rely on precise initial parameter settings, enhancing its adaptability in practical scenarios. In summary, it provides a robust solution for accurately identifying projectile aerodynamic parameters and holds promises for engineering applications.

Keywords: projectile aerodynamic parameter identification; improved snake optimization algorithm; Kalman filtering; genetic algorithm integration.

1. INTRODUCTION

In modern warfare, accurate identification of the aerodynamic parameters of projectiles is a crucial task, and the aerodynamic force of projectiles is one of the primary conditions that affect their flight trajectory and hitting accuracy [1–3]. The aerodynamic parameters of the projectile, including drag coefficient, lift coefficient, etc., determine the flight performance of the projectile, which not only affects the flight trajectory and stability, but also affects the optimization of the projectile design, the safety of the flight, and the maximization of the payload [4, 5].

Traditional methods for aerodynamic parameter identification rely on wind tunnel experiments, which are costly and have certain limitations, and theoretical computational methods, such as computational fluid dynamics (CFD) simulations, which require high computational resources and time costs [6, 7]. In [6], it is pointed out that wind tunnel experiments are limited by model size and experimental conditions, which makes it difficult to simulate the real flight environment fully. Therefore, Hou *et al.* applied the differential evolution (DE) algorithm to aerodynamic parameter identification, replacing the Newton iterative gradient optimization method in the traditional maximum likelihood method, and conducted full trajectory identification of the aerodynamic parameters of high-spin projectiles. While CFD simulation is mentioned in [7], the accuracy of its simulation of complex flow fields depends on the degree of mesh delineation fineness, and the amount of computational work increases sharply with the accuracy requirement.

In recent years, aerodynamic parameter identification has resulted in several methods, and the development is extremely rapid. More classical methods include the least squares method, the great likelihood method, the C-K method, and certain intelligent algorithms that have recently emerged [8–11]. Wang *et al.* [12] used Kalman filtering to estimate the aerodynamic parameters of a quadcopter UAV online, calculated the corresponding scale factor, compensated the rotational speed, and improved the UAV altitude and heading control performance. Li *et al.* [13] used the neural network-Newton method based on the great likelihood criterion to process the flight state data of an uncontrolled rotating projectile and extract its zero-lift drag coefficient. Hui *et al.* [14] proposed a method for identifying aerodynamic parameters that combines the gated recurrent unit (GRU) neural network model with the Gauss-Newton (GN) optimization algorithm. Gan *et al.* [15] combined a genetic algorithm with an optimized extreme learning machine to identify the aerodynamic parameters of a certain type of uncontrolled bomb. Puri *et al.* [16] proposed the DM-AEO hybrid metaheuristic algorithm by combining dwarf mongoose optimization (DMO) and artificial ecosystem optimization (AEO) for optimal EEG channel selection in Schizophrenia detection, using four decomposition methods (EMD, VMD, DWT, LCOFBs) and achieving 99.26% accuracy with only four channels on a public

*e-mail: yangbaolu1998@163.com

Manuscript submitted 2025-07-29, revised 2025-10-24, initially accepted for publication 2025-10-28, published in January 2026.

dataset. Gao *et al.* [17] proposed the IDBO-KELM algorithm, which optimizes the kernel parameters and regularization coefficients of the kernel extreme learning machine (KELM) using the improved dung beetle optimization (IDBO) algorithm (with four improvements, including Chebyshev chaotic mapping and golden sine strategy), and applied it to projectile aerodynamic parameter identification, achieving better accuracy than ELM and IDBO-ELM, especially in the transonic region. Zhong *et al.* [18] presented the self-adaptive competitive swarm optimizer (SACSO), which introduces a parameter sorting scheme and linear population reduction strategy to improve the competitive swarm optimizer (CSO), verified its performance on CEC2017, CEC2022, seven engineering problems, and human-powered aircraft design tasks, and confirmed its competitiveness.

In this context, Hashim *et al.* [19] proposed the snake optimization algorithm (SO), which is a heuristic algorithm inspired by hunting, fighting, mating, and reproduction behaviors of snakes in the natural environment. Theoretically, in continuous function optimization, the SO algorithm has some advantages by simulating the behavioral patterns of snakes and searching in the solution space. However, there are many limitations when applied to such specific fields as the identification of aerodynamic parameters of projectiles and arrows. Al-Shourbaji *et al.* [20] indicated that SO had a fast decay of population diversity in complex constrained optimization and needed to repeat searches many times to avoid local optima, resulting in a waste of computational resources, which is particularly prominent in the aerodynamic parameter identification of projectiles, where the solution space involves multiple local optima. Puri *et al.* [21] confirmed in their study of SO-based EEG channel selection is that SO alone is difficult to balance global exploration and local exploitation in high-dimensional spaces, and its robustness decreases by 20% when facing noisy data. According to the study of [19] on heuristic algorithms in complex constrained optimization problems, the solution space of aerodynamic parameter identification of projectiles and arrows is complex, and there are multiple local optimal solutions, so the SO algorithm is prone to falling into the local optimum, which leads to suboptimal solutions in the end and reduces the robustness of the algorithm. Due to the ease of slipping into local optimality, the SO algorithm needs to repeat the search many times to obtain a better solution, resulting in unnecessary consumption of computational resources. Moreover, the uncertainty and complexity of the model increase in the actual flight environment, which limits the application scope of the SO algorithm and may be infeasible in practical applications, thus seriously affecting the accuracy of the prediction of the projectile and arrow performance.

To tackle the aforementioned issues, this paper develops an enhanced snake optimization algorithm incorporating genetic algorithm (GA) mechanisms. This algorithm integrates the global exploration capability of GA with the local search precision of SO, aiming to boost optimization performance in identifying aerodynamic parameters for projectiles and arrows. Specifically, GA selection, crossover, and mutation operations are introduced to enhance population diversity – this directly addresses the tendency of the SO algorithm to fall into local optima in the complex solution space of aerodynamic parameter identification, eliminating the need for repeated searches and thus reducing unnecessary computational resource consumption. The study first elaborates on the working principles of the traditional snake optimization algorithm and its improved version, which incorporates GA-based operations (e.g., selection and crossover) and Kalman filtering for noise reduction. The Kalman filtering module is designed to mitigate the impact of uncertainty in actual flight environments, solving the problem of the limited applicability of the original SO algorithm in practical scenarios and ensuring the accuracy of projectile performance prediction. It then introduces the three-degree-of-freedom ballistic model for projectiles and arrows, validating the improved algorithm's parameter estimation accuracy using simulated trajectory data. Finally, field test data are employed to further verify the algorithm, confirming its applicability and effectiveness in practical scenarios. This research offers a novel optimization approach for identifying aerodynamic parameters of projectiles and arrows, while also providing insights for related studies in adjacent fields.

2. ALGORITHMS

2.1. Snake optimization algorithm (SO)

The snake optimization algorithm is designed based on the mating habits of snakes, with particular emphasis on how they choose mates when food is abundant and temperatures are favorable. The algorithm key procedures can be outlined as follows:

1. Generate an initial population with random distribution N, then split it evenly into two subgroups: male and female individuals.
2. Compute the initial fitness value for each individual in both subgroups.
3. Establish parameters for temperature$Temp$and food$F$

$$\text{Temp} = \exp\left(\frac{-t_c}{T_m}\right), \tag{1}$$

where t_c denotes the current iteration count and T_m stands for the maximum number of iterations.

$$F = P_1 * \exp\left(\frac{t_c - T_m}{T_m}\right), \tag{2}$$

where P_1 is a constant with a value of 0.5.

4. Exploration stage (food absent). When the condition $F <$ Threshold is satisfied, a global search is initiated, where the snake conducts food-seeking by selecting arbitrary positions and adjusting its location in relation to them.

$$X_{i,m}(t+1) = X_{\text{rand},m}(t) \pm P_2 * A_m * \big((X_{\max} - X_{\min}) * \text{rand} + X_{\min}\big), \tag{3}$$

where $X_{i,m}$ represents the position of the i-th male individual, $X_{\text{rand},m}$ stands for a randomly selected male position, $rand$is a random value in the range [0, 1], $X_{\min}$and $X_{\max}$

denote the lower and upper limits of the search range, respectively, and P_2 is a constant set to 0.5. A_m indicates the food-searching capability of male individuals, with its mathematical expression given as follows

$$A_m = \exp\left(\frac{-\text{fitness}_{\text{rand},m}}{\text{fitness}_{i,m}}\right), \tag{4}$$

where $\text{fitness}_{\text{rand},m}$ denotes the fitness of a randomly selected male individual, and $\text{fitness}_{i,m}$ represents the fitness of the current i-th male individual.

$$X_{i,f}(t+1) = X_{\text{rand},f}(t) \pm P_2 * A_f * \left((X_{\max} - X_{\min}) * \text{rand} + X_{\min}\right), \tag{5}$$

where $X_{i,f}$ refers to the i-th female position and $X_{\text{rand},f}$ refers to the random female position. A_f denotes the female's ability to find food and is expressed as

$$A_f = \exp\left(\frac{-\text{fitness}_{\text{rand},f}}{\text{fitness}_{i,f}}\right), \tag{6}$$

where $\text{fitness}_{\text{rand},f}$ indicates the fitness of a randomly chosen female individual, and $\text{fitness}_{i,f}$ stands for the fitness of the current i-th female individual.

5. Development stage. At this point $F >$ Threshold, the temperature condition is considered.
   (a) Foraging mode: If Temp > Threshold2, that is, when the temperature gets excessively high, it starts to make a move to the location of the food, to the globally optimal location.

$$X_{i,j}(t+1) = X_{\text{food}}(t) \pm P_3 * \text{Temp} * \text{rand} * \left(X_{\text{food}} - X_{i,j}(t)\right), \tag{7}$$

   where $X_{i,j}$ represents the position of an individual (either male or female), X_{food} denotes the position of the optimal individuals, and P_3 is a constant with a value of 2.

   (b) Combat mode: When the condition Temp < Threshold2 is met (indicating suitable temperature), the snake enters combat mode if rand > 0.6.

$$X_{i,m}(t+1) = X_{i,m}(t) \pm P_4 * FM * \text{rand} * \left(F * X_{\text{best},f} - X_{i,m}(t)\right), \tag{8}$$

   where $X_{\text{best},f}$ denotes the position of the optimal individual in the female group, FM represents the combat capability of the male agent, and P_4 is a constant with a value of 0.01.

$$X_{i,f}(t+1) = X_{i,f}(t) \pm P_4 * FF * \text{rand} * \left(F * X_{\text{best},m} - X_{i,f}(t)\right), \tag{9}$$

   where $X_{\text{best},m}$ refers to the position of the best individual in the male group, FF is the fighting ability of the female agent.

$$FM = \exp\left(\frac{-f_{\text{best},f}}{f_i}\right), \tag{10}$$

$$FF = \exp\left(\frac{-f_{\text{best},m}}{f_i}\right), \tag{11}$$

   where $f_{\text{best},f}$ indicates the fitness of the optimal agent in the female group, $f_{\text{best},m}$ is the fitness of the best agent of the male group, f_i is the agent fitness.

   (c) Mating mode: If Temp < Threshold2 and rand < 0.6, the snake will be in the mating mode.

$$X_{i,m}(t+1) = X_{i,m}(t) \pm P_4 * MM * \text{rand} * \left(F * X_{i,f} - X_{i,m}(t)\right), \tag{12}$$

   where MM refers to the mating ability of the male.

$$X_{i,f}(t+1) = X_{i,m}(t) \pm P_4 * MF * \text{rand} * \left(F * X_{i,m} - X_{i,m}(t)\right), \tag{13}$$

   where MF refers to the mating ability of the female.

$$MM = \exp\left(\frac{-\text{fitness}_{i,f}}{\text{fitness}_{i,m}}\right), \tag{14}$$

$$MF = \exp\left(\frac{-\text{fitness}_{i,m}}{\text{fitness}_{i,f}}\right). \tag{15}$$

   In cases where mating succeeds, the algorithm produces a new population member to take the place of the currently least fit individual.

$$X_{\text{worst},m} = X_{\min} + \text{rand} * (X_{\max} - X_{\min}), \tag{16}$$

$$X_{\text{worst},f} = X_{\min} + \text{rand} * (X_{\max} - X_{\min}), \tag{17}$$

   where $X_{\text{worst},m}$ stands for the poorest-performing individual in the male group, and $X_{\text{worst},f}$ represents the poorest-performing individual in the female group.

6. Termination step. Assess and refresh the optimal positions and individuals within the population. The algorithm terminates once a predefined iteration count or other stopping criteria are met.

2.2. Introduction to genetic algorithms

Genetic algorithm is a bionic algorithm inspired by Darwin's theory of natural selection [22]. The fundamental concept of this algorithm lies in initializing the population via coding, usually in binary or real numbers, to map the problem solution to chromosomes. Each member of this population is then evaluated using a fitness function, which is based on the problem objective and is used to measure individual merit. Based on the results of these evaluations and a specific genetic manipulation strategy, a new generation of individuals is generated and undergoes the fitness evaluation process again, and the cycle continues until a predefined termination condition is met [23, 24]. A detailed understanding of the various terms and concepts related to genetic algorithms can be gained by consulting Table 1.

Table 1

Fundamental explanations and their associated connotations of genetic algorithm terms

Terminology	Means
Individual	Denoting a possible solution
Genes	Constituent elements or characteristics forming the solution
Chromosomes	Coding format selected for the solution
Population	A set consisting of numerous solutions
Fitness	Measures of individual performance
Selection	Selection of superior individual procedures
Replication	Transmission of hereditary data
Crossover	Combining genes of two solutions for new offspring
Mutation	Genetic change for new traits in individuals
Coding	Translating solutions to chromosome codes
Decoding	Translating chromosomes to original solutions

2.3. Introduction to the Kalman filter algorithm

In data analysis, the presence of noise often leads to information distortion, affecting the accuracy of the final results. To address this issue, the Kalman filter provides an effective solution and is widely applied in signal processing, navigation, and control systems [25, 26].

The core process of the Kalman filter consists of two stages: prediction and update. Below is the basic mathematical framework of the Kalman filter:

1. Prediction step

$$\hat{x}_k = A_k \hat{x}_{k-1}, \tag{18}$$

$$P_k = A_k P_{k-1} A_k^T + Q_k, \tag{19}$$

where $\hat{x}_k$ denotes the state estimate at time step k, A_k represents the state transition matrix, P_k stands for the covariance matrix of the estimation error, and Q_k is the covariance matrix of the process noise.

2. Update steps

$$K_k = P_k H_k^T \left(H_k P_k H_k^T + R_k \right)^{-1}, \tag{20}$$

$$\hat{x}_k = \hat{x}_k + K_k \left(z_k - H_k \hat{x}_k \right), \tag{21}$$

$$P_k = \left(\mathrm{I} - K_k H_k \right) P_k, \tag{22}$$

where K_k denotes the Kalman gain, z_k represents the observation, H_k stands for the observation matrix, and R_k is the covariance matrix of the observation noise. The updated $\hat{x}_k$ and P_k represent the covariance matrices of the updated state estimates and estimation errors, respectively [27, 28].

2.4. Genetic-snake optimization algorithm (ESO-GO algorithm)

The essence of the ESO-GO algorithm is the integration of the local search efficiency of the snake optimization algorithm and the global search superiority of the genetic algorithm. Based on their behavioral patterns of foraging, fighting, and mating in nature, snakes search for a better solution in the vicinity, and combine with the selection, crossover, and mutation operations of the genetic algorithm to perform a global search in the solution space to find the approximate region of the better solution to further excavate the potentially more optimal solution. In this way, repeated iterations are performed to continuously optimize the estimates of the bullet and arrow aerodynamic parameters so that the algorithm can find the bullet and arrow aerodynamic parameters that are as close as possible to the optimal solution in the global range. The general steps of the snake optimization algorithm improved by the genetic algorithm are as follows:

1. Initialize a randomly generated population and split it evenly into two subgroups: male and female individuals. In the optimization algorithm, the population initialization is often generated randomly, and whether its distribution in the search space is broad and uniform significantly impacts the convergence speed of the whole search process and the algorithmic efficiency. This paper uses a chaotic optimization strategy to initialize the population. Logistic mapping ensures the wide distribution of the population in the solution space and improves the search efficiency of the algorithm due to its high uniformity in distribution and its ability to generate sequences that maintain chaotic characteristics for a long time.

The expression for this mapping is shown below

$$\begin{cases} x_{n+1} = 4x_n(1 - x_n), \\ 0 < x_1 < 1. \end{cases} \tag{23}$$

The reasons for choosing this mapping are as follows: The Logistic map has a simple form, involving only quadratic term operations, with a short single-iteration time, which can avoid increasing the computational burden in the initialization phase. Moreover, it has strong stability – when $\mu = 4$, it can stably remain in a chaotic state without complex parameter adjustment, whereas mappings like the Henon map need to control multiple parameters and are prone to degenerate into periodic sequences due to improper settings.

2. Compute the initial fitness values for each subgroup, adopt the snake optimization algorithm to pinpoint the optimal individual of each interval, and the initial ballistic parameters c_{x0}, v_x, v_y, v_z, x, y, z. Next, input these initial ballistic parameters into the three-degree-of-freedom ballistic equations to derive the calculated ballistic value for each step within the corresponding interval, and compute the residuals (eps) using the calculated value and measured data. The formula for residuals is as follows:

$$\mathrm{eps} = \sum_{i=1}^{N} \sum_{m=1}^{N_1} \left[y_{\mathrm{mea},i} - y_{\mathrm{cal},m}(x_i) \right]^2, \tag{24}$$

where N refers to the number of measurements in a small interval, N_1 refers to the state quantity of the ballistic equation, which is equal to 6, $y_{\mathrm{mea},i}$ refers to the measured value, and $y_{\mathrm{cal},m}(x_i)$ refers to the calculated value of the ballistic equation. Find the individuals Xbest_m and Xbest_f with the smallest fitness values in the two groups, respectively, and take their smallest values as the position of the Xfood.

3. Perform mutation operations. To boost the algorithmic global search performance and avoid premature convergence to a local optimum, this study incorporates a mutation strategy that aids the algorithm in escaping local optima.

(1) Implementation of periodic mutation and dynamic mutation rate: Employing a periodic mutation strategy and dynamically modifying the mutation rate based on the fitness of the current optimal solution contributes to preserving population diversity while preventing excessive disruption to the algorithm's convergence process. The period of periodic mutation is set as a fixed iteration interval, i.e., mutation is triggered every seven iterations. The reason for choosing a fixed period is as follows: In the early exploration stage of the algorithm, a stable mutation frequency (once every seven iterations) can maintain population diversity and avoid premature convergence. Moreover, combined with the dynamic mutation rate (when the global optimal value is less than 100, the mutation rate decreases from 0.1 to 0.05), it can not only maintain the stability of periodic mutation but also enhance the local search ability by adaptively adjusting the mutation intensity.

(2) Local optimality identification mechanism: To further keep the snake optimization algorithm from becoming trapped in local optimality, a minor random mutation operator is established. When the algorithmic optimal position remains unchanged through 10 consecutive searches and the eps exceed 0.0001, a minor random mutation operator is appended to the historical optimal position of the group, enhancing the algorithmic capability to break free from local optimality. Once the local optimal identification mechanism is activated, the optimal position will undergo mutation.

$$X\text{food} = X\text{food}(1+0.05\omega), \tag{25}$$

where ω represents Gaussian white noise.

4. Execute the selection and crossover operation. During the mating stage of the snake optimization algorithm, the selection and crossover operation from the genetic algorithm is employed to mimic the natural mating behavior of snakes. This mechanism aims to produce individuals with new traits through exchange of information, thereby boosting the population diversity and overall adaptability.

5. Termination. Compute the fitness of each individual in the population and update the Xfood that is optimal for each iteration and the Xbest_m and Xbest_f in the male and female individuals. If the stopping condition is satisfied, the algorithm stops; otherwise, the iteration continues.

The algorithm flowchart is presented in Fig. 1.

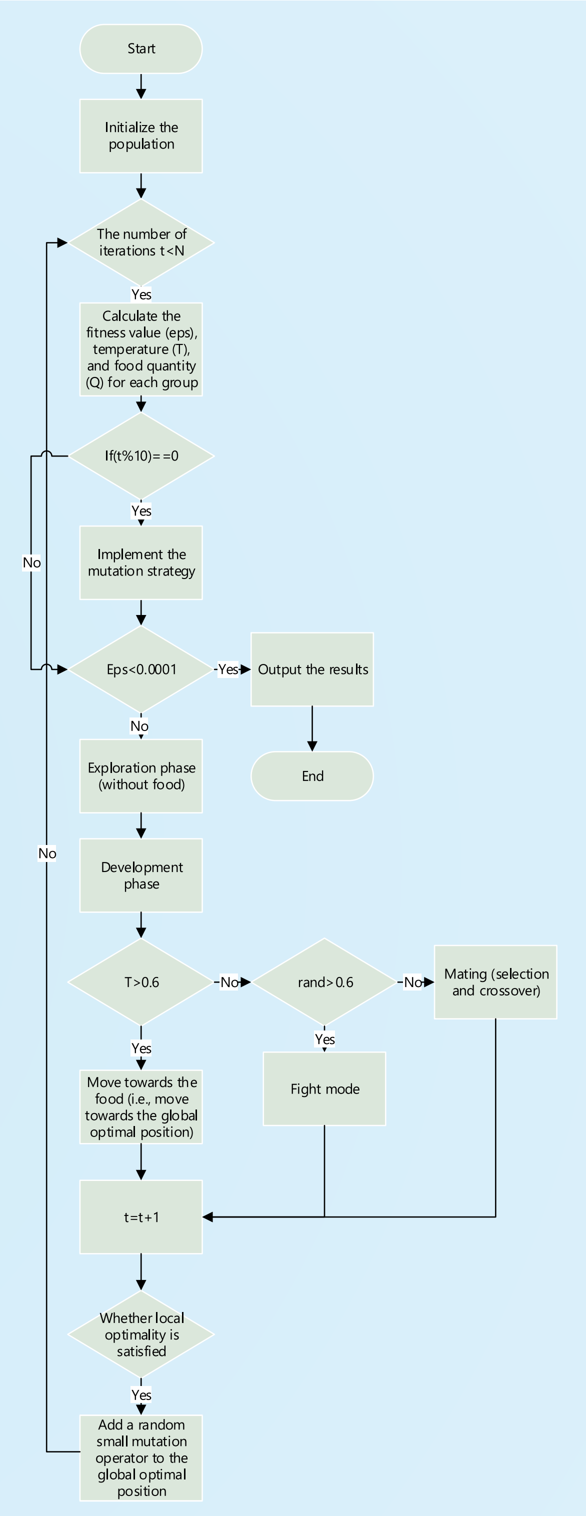

Start
Initialize the population
The number of iterations t<N
Yes
Calculate the fitness value (eps), temperature (T), and food quantity (Q) for each group
If(t%10)==0
Yes
No
Implement the mutation strategy
Eps<0.0001
Yes
Output the results
No
End
Exploration phase (without food)
No
Development phase
T>0.6
No
rand>0.6
No
Mating (selection and crossover)
Yes
Yes
Move towards the food (i.e., move towards the global optimal position)
Fight mode
t=t+1
Whether local optimality is satisfied
Yes
Add a random small mutation operator to the global optimal position

Fig. 1. Flowchart of the improved snake optimization algorithm

2.5. Particle swarm-snake optimization algorithm (SO-PSO algorithm)

The standard PSO algorithm is widely used in function optimization, parameter identification, control system design, and other fields [29]. Its advantages include being easy to implement, simple operation, etc., but there are also shortcomings, such as being easy to quickly fall into local optimization, etc. Given these problems, this paper combines the PSO algorithm with the SO algorithm to obtain the SO-PSO algorithm. The flow of the SO-PSO algorithm is as follows:

1. The PSO algorithm parameters are set based on the SO algorithm in Section 2.1. The range of inertia weights is $\{w_{\min} = 0.4,\ w_{\max} = 0.9\}$, c_{1p} refers to the individual cognitive coefficient, and c_{2p}refers to the social cognitive coefficient.

2. Initialization particles. Generate chaotic initialization particles according to the method in Section 2.4.

3. Adaptive inertia weights

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right)\frac{t}{T}\,. \tag{26}$$

4. In Section 2.1, the PSO algorithm is employed to take the place of the exploration phase of the SO algorithm. The expressions for velocity and position updates are shown in (27), (28), (29), and (30).

$$\begin{aligned} v_{i,m}(t+1) &= w * v_{i,m}(t) + c_{1p} * \text{rand} * \left(X\text{best}_{i,m} - X_{i,m}(t)\right) \\ &\quad + c_{2p} * \text{rand} * \left(X_{\text{best},m} - X_{i,m}(t)\right), \end{aligned} \tag{27}$$

$$X_{i,m}(t+1) = X_{i,m}(t) + v_{i,m}(t+1), \tag{28}$$

$$\begin{aligned} v_{i,f}(t+1) &= w * v_{i,f}(t) + c_{1p} * \text{rand} * \left(X\text{best}_{i,f} - X_{i,f}(t)\right) \\ &\quad + c_{2p} * \text{rand} * \left(X_{\text{best},f} - X_{i,f}(t)\right), \end{aligned} \tag{29}$$

$$X_{i,f}(t+1) = X_{i,f}(t) + v_{i,f}(t+1), \tag{30}$$

where $v_{i,m}$ and $v_{i,f}$ denote the velocity of the i-th particle in the male and female populations, respectively, $X\text{best}_{i,m}$ and $X\text{best}_{i,f}$ represent the optimal position of the i-th particle in the male and female populations, respectively.

5. Adoption of periodic perturbations. Introducing stochastic perturbations during the iteration process enhances the algorithmic capability to break free from local optimality.

3. MODEL

The structure of the projectile is shown in Fig. 2, and its key components include a well-designed radome and projectile body. The radome, located at the front end of the projectile, is made of white high-strength composite material, which not only provides the necessary protection for the GPS sensor inside but also optimizes the signal reception performance. The front tip of the radome is specially designed to enhance its aerodynamic properties and signal reception efficiency. The body part is made of lightweight alloy material to ensure structural stability and overall lightness, which is crucial for the projectile flight performance. GPS sensors inside the radome are responsible for capturing real-time position and velocity data, which are then transmitted to the ground radar system via wireless signals from a transmitter in the body. Upon receiving these signals, the ground radar can track the flight trajectory of the projectile in real time and collect critical flight data. By analyzing these, the aerodynamic parameters of the projectile during flight can be identified, thus providing vital information for the assessment of the flight performance and aerodynamic characterization of the projectile.

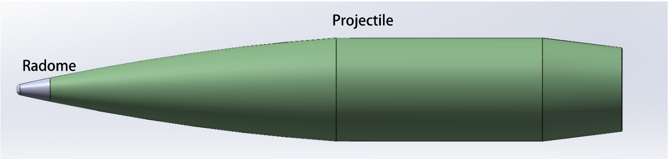


Fig. 2. Projectile model

In projectile dynamics analysis, the three-degree-of-freedom model emphasizes the translational motion of the projectile, characterized by the variations in velocity and position along the coordinate axes. This model streamlines the dynamical equations by disregarding the rotational movements of the projectile about its center of mass, including roll, pitch, and yaw. While the three-degree-of-freedom model has constraints in depicting the full trajectory of a projectile, it offers notable benefits in computational efficiency and in examining the basic features of the trajectory. The three-degree-of-freedom model for the motion of the projectile center of mass is employed.

$$\begin{cases} \dfrac{dv_x}{dt} = -\dfrac{\rho v^2}{2m} S c_{x0} \cos\theta_a \cos\psi_2\,, \\ \dfrac{dv_y}{dt} = -\dfrac{\rho v^2}{2m} S c_{x0} \sin\theta_a \cos\psi_2 - g, \\ \dfrac{dv_z}{dt} = -\dfrac{\rho v^2}{2m} S c_{x0} \sin\psi_2\,, \\ \dfrac{dx}{dt} = v_x\,, \\ \dfrac{dy}{dt} = v_y\,, \\ \dfrac{dz}{dt} = v_z\,, \end{cases} \tag{31}$$

$$\begin{cases} v = \sqrt{v_x^2 + v_y^2 + v_z^2}\,, \\ S = \pi * D * D/4, \\ \theta_a = \arctan\left(\dfrac{v_y}{v_x}\right), \\ \psi_2 = \arcsin\left(\dfrac{v_z}{v}\right), \end{cases} \tag{32}$$

where v_x, v_y, v_z, x, y, z denote the velocity components and position components on the ground three-axis coordinate system, respectively, v stands for the projectile resultant velocity, θ_a represents the ballistic inclination, ψ_2 indicates the ballistic declination, ρ refers to density,m denotes the projectile mass, S signifies the projectile cross-sectional area, D represents the projectile diameter, c_{x0} stands for the resistance coefficient, and g indicates gravitational acceleration.

4. SIMULATION ANALYSIS

With a high-speed spinning projectile as the context, the three-degree-of-freedom ballistic equations are employed to produce simulated flight trajectory data. And then the simulated ballistic data are used for parameter identification, and the values of the ballistic and aerodynamic parameters obtained from the simulation are compared to verify the effectiveness of the algorithms in solving the problem of identifying the aerodynamic parameters of the projectile. The flight test data for simulation are $v_{x,\text{mea}}$, $v_{y,\text{mea}}$, $v_{z,\text{mea}}$, x_{mea}, y_{mea}, z_{mea}, and the parameters to be recognized are c_{x0}, v_x, v_y, v_z, x, y, z. For offline identification, a set of ballistic data is known, and the Kalman filter algorithm is first used to filter this set of ballistic data to reduce the interference of noise, and then the interval constant method is used to divide the trajectory into$N$intervals, and the time interval of the ballistic data in each interval is 0.05 s. Assuming that the value of the aerodynamic parameter in each interval is a fixed constant, the ESO-GO algorithm is used to divide the ballistic parameters and drag force into N intervals, and then the value of the aerodynamic parameter is 0.05 s in each interval. The starting ballistic parameters and drag coefficients are recognized.

To generate the simulation ballistic data, the main initial data required are muzzle velocity 930 m/s and angle of fire 38. Therefore, the simulation initial conditions are shown below:

$$v_x = 732.81\ \text{m/s},\quad v_y = 572.61\ \text{m/s},\quad v_z = 10\ \text{m/s},$$
$$x = 0\ \text{m},\quad y = 0\ \text{m},\quad z = 0\ \text{m},\quad m = 33.4\ \text{kg},\quad D = 0.13\ \text{m}.$$

4.1. Comparison of parameter identification results before and after Kalman filtering

Kalman filtering has shown remarkable results in dealing with datasets containing noise. By applying Kalman filtering to the ballistic measurement data, the noise interference can be effectively reduced, thus improving the data quality. Building on this, parameter identification is conducted using the enhanced snake optimization algorithm, with the results presented in Fig. 3. Compared with the identification results of directly applying the ESO-GO algorithm without filtering, the filtered data show a smoother trend in parameter identification, and the identification results are closer to the simulated ballistic data. Among them, the ‘standard value’ curve in Fig. 3 is the value obtained based on the simulated ballistic data, and the ‘standard value’ in the subsequent graphs has the same meaning, so we will not repeat it.

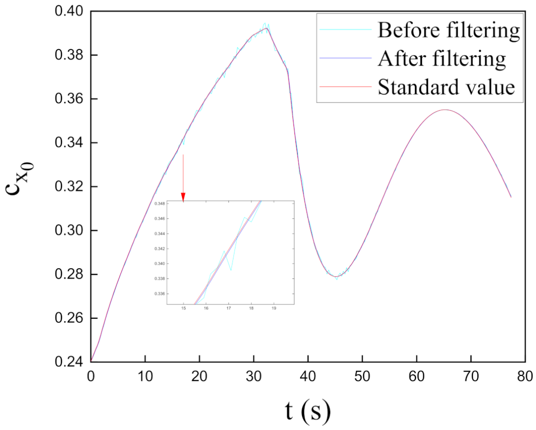


Fig. 3. Results of drag coefficient identification using the Kalman filter

4.2. Parameter identification utilizing the enhanced snake optimization algorithm

Kalman filter is firstly used to filter this set of simulated ballistic data, and then SO method, ESO-GO method and SO-PSO method are used to identify the parameters, respectively, and the identification results are shown in Fig. 4. In the process of drag coefficient identification, when the search space is limited to the range of 0.1 to 0.5, the SO algorithm and the SO-PSO algorithm tend to converge to the local optimal solution, resulting in a significant deviation between the identification results and the simulated ballistic data. When the identified parameters lie at the front end of this small interval, the algorithms adjust the drag coefficients to a level much higher than their true values to minimize the objective function eps. Conversely, if the identified results are located at the back end of the interval, the algorithm adjusts the resistance coefficient to a negative value to avoid an increase in the objective function eps. These reveal the limitations of the SO algorithm and the SO-PSO algorithm in dealing with locally optimal solutions, especially in the application scenario of ballistic data parameter identification.

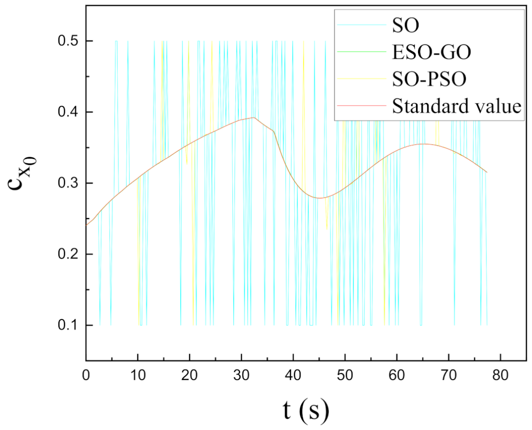


Fig. 4. Results of drag coefficient identification

As can be seen from Table 2, according to the results of the sum of squares of errors (SSR) of the identified parameters and the simulated ballistic data, the drag coefficients and ballistic parameters identified by the ESO method, the ESO-GO method and the SO-PSO method are closer to the values of the ballistic parameters and drag coefficients obtained from the simulation in the identification of the parameters from the simulation ballistic data. The expression of the SSR is

$$\text{SSR} = \sum_{i=1}^{N_a} \sum_{m=1}^{N_b} \left[y_{i,m} - y_{\text{mea},i,m}\right]^2, \tag{33}$$

Table 2
Sum of squared residuals (SSR) of different algorithms

Error	SO	SO-PSO	ESO-GO
c_{x0}	3.6450	0.4461	2.56×10^{-5}
x	2.6009	0.1228	1.917×10^{-6}
y	1.1774	0.0641	4.875×10^{-6}
z	0.04843	0.0389	4.649×10^{-6}
v_x	199.9368	9.1974	1.6×10^{-3}
v_y	91.2114	2.7694	9.904×10^{-4}
v_z	0.0372	0.0035	1.142×10^{-6}

where N_a is the length of the simulated ballistic data, N_b is the number of parameters to be recognized, $y_{i,m}$ is the recognition result of different algorithms on this set of simulated ballistic data, and $y_{\mathrm{mea},i,m}$ is the simulated ballistic data.

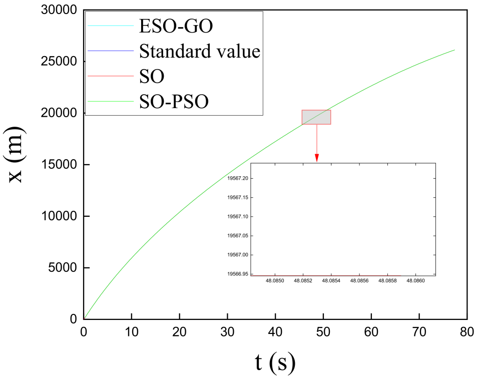
The complexity calculation results of the SO algorithm, SO-PSO algorithm, and ESO-GO algorithm are shown in Table 3. The measurement conditions for "running time" in the table are as follows: the time required for parameter identification on the same interval data under the same number of iterations, population size, and problem dimension. All time data were measured by running on the MATLAB 2022 platform.

Table 3
Comparison of complexity and running time among different algorithms

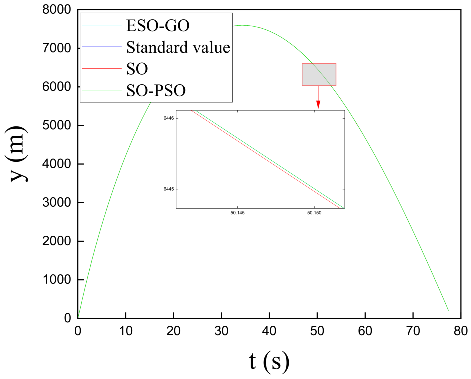
Algorithms	Iteration complexity	Running time
SO	$O(T_m \times N \times D_m)$	3.41 s
SO-PSO	$O(T_m \times N \times D_m)$	3.82 s
ESO-GO	$O(T_m \times N \times D_m)$	4.21 s

In the table, T_m represents the number of iterations with a value of 500, N represents the population size with a value of 100, and D_m represents the dimension with a value of 7.

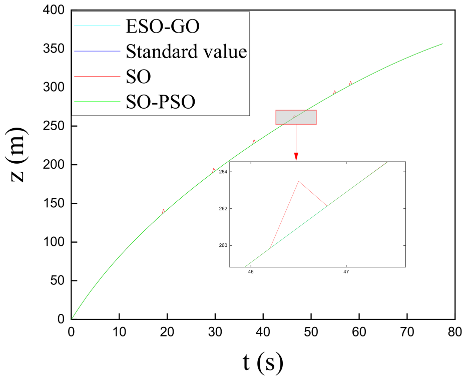
Figures 5 and 6 show the recognition results of the position parameter x, y, z and the velocity parameters v_x, v_y, and v_z, respectively. The results show that the identification results of the ESO-GO algorithm are closer to the simulated ballistic data



(a) Identification results of range x

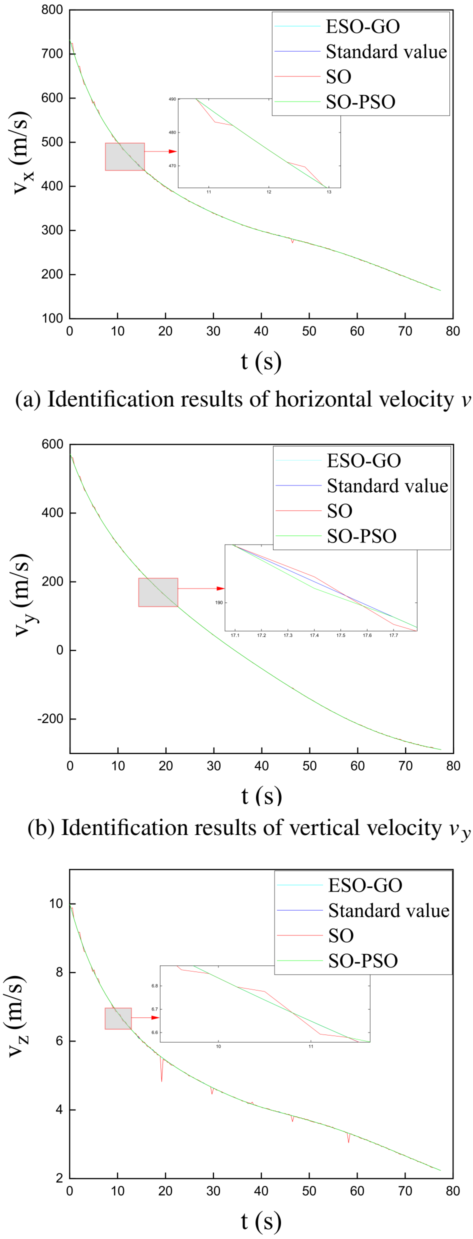


(b) Identification results of altitude y



(c) Identification results of lateral deviation z

Fig. 5. Identification results of position parameters



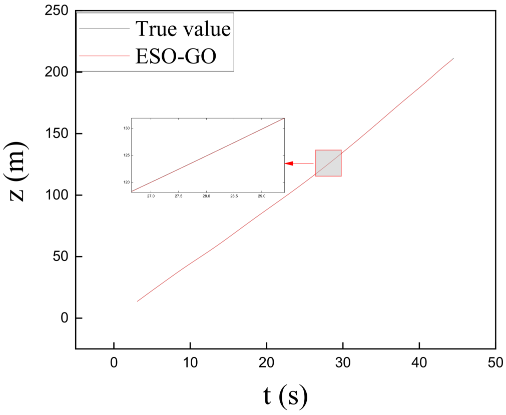
(a) Identification results of horizontal velocity v_x

(b) Identification results of vertical velocity v_y

(c) Identification results of axial velocity v_z

Fig. 6. Identification results of velocity parameters

and overlap completely, while the identification results of the SO algorithm show a large deviation from the simulated ballistic data, and the identification results of the SO-PSO algorithm also show a small deviation from the simulated ballistic data. It can be seen that the SO algorithm has limitations in finding the global optimal solution and is prone to falling into the local optimum; while the PSO algorithm is added to the SO algorithm for parameter identification, the algorithm may still encounter the dilemma of the local optimal solution, even though this combination significantly improves the global search capability. This indicates that the ESO-GO algorithm is more effective in avoiding falling into local optimums and improving the accuracy and reliability of the identification when dealing with complex optimization problems.

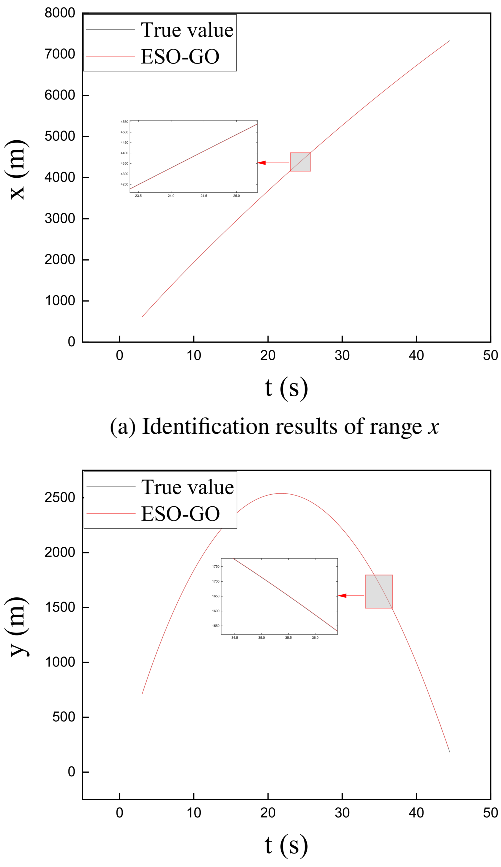


(c) Identification results of lateral deviation z

Fig. 7. Identification results of position parameters

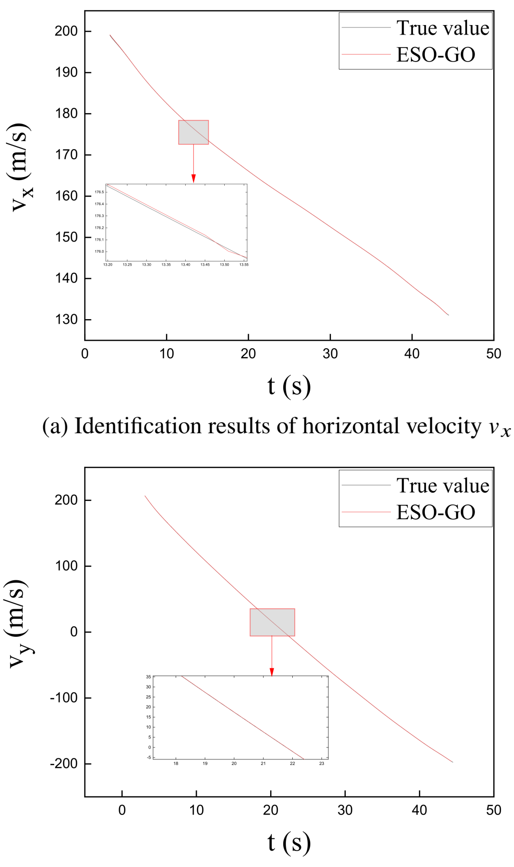
5. EXPERIMENTAL VALIDATION

Following the parameter identification of simulated ballistic data, the ESO-GO algorithm has higher accuracy in the field of projectile and arrow parameter identification. To further explore the parameter identification process of real experimental data, the ESO-GO algorithm is applied to the actual flight data of a certain type of grenade, which is the radar-measured full trajectory data, to verify the performance and applicability of the algorithm in real conditions. After filtering the experimental data of a certain grenade for parameter identification, the identification results are shown in Figs. 7 and 8, the position parameter x, y, z and the velocity parameters v_x, v_y, v_z identified by the ESO-GO algorithm show a high degree of consistency with the actual flight data of the projectile, and the phenomenon confirms that the ESO-GO algorithm can accurately and efficiently extract the accurate position and velocity information from the experimental data, and reconstruct the flight path of the projectile. In addition, this high degree of matching also reflects algorithmic strong adaptability and robustness in the face of complex flight environments and uncertainties. Among them, the "true value" curves in Figs. 7 and 8 are the actual projectile flight data of a certain type of grenade.



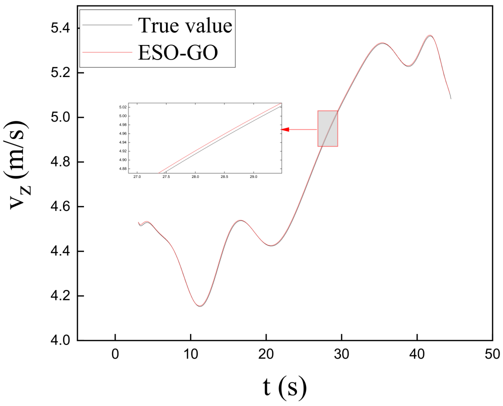
(a) Identification results of range x

(b) Identification results of altitude y



(a) Identification results of horizontal velocity v_x

(b) Identification results of vertical velocity v_y



(c) Identification results of axial velocity v_z

Fig. 8. Identification results of velocity parameters

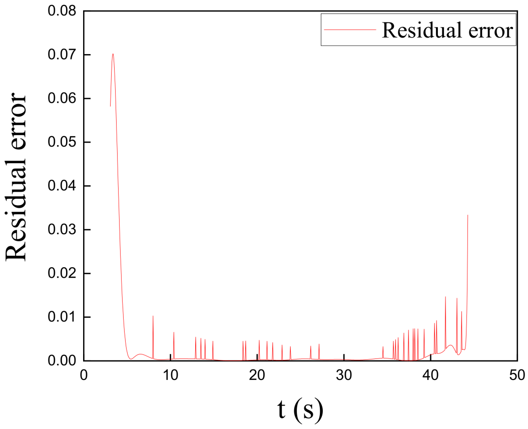


Fig. 10. Residual eps

In Fig. 9, the results of the drag coefficients identified using the ESO-GO algorithm are shown. These results provide an insight into the aerodynamic drag force on the projectile during flight. To assess the algorithmic performance, we conducted a further analysis of the residual eps throughout the parameter identification process, with the findings illustrated in Fig. 10. The maximum value of the residual eps is 0.07, while the residual eps stay below 0.01 in most cases, a result that strongly demonstrates the power of the ESO-GO algorithm for global search.

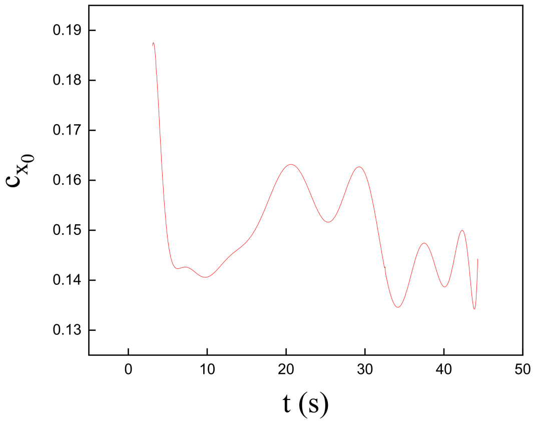


Fig. 9. Identification results of drag coefficient

In order to further verify the effectiveness of the ESO-GO algorithm in practical applications, the resistance coefficients obtained from the identification in Fig. 9 were substituted into (31) and (32), and the following initial conditions were used to calculate the values of each observed physical quantity:

$$v_x = 199.0607\text{ m/s},\quad v_y = 207.0640\text{ m/s},$$
$$v_z = 4.5319\text{ m/s},\quad x = 616.7365\text{ m},\quad y = 715.1286\text{ m},$$
$$z = 13.6305\text{ m},\quad m = 16.5\text{ kg},\quad D = 0.12\text{ m}.$$

The ballistic trajectory obtained from the simulation is shown in Fig. 11, in which the coordinates of the predicted impact point are (7419.8 m, 207.3956 m). In contrast, the coordinates of the actual impact point are (7435 m, 218 m). Here, the first value represents the projection distance of the predicted impact point on the firing direction of the muzzle (i.e., the straight-line distance from the launch point along the firing direction), and the second value denotes the horizontal distance of the predicted impact point deviating from the firing direction (lateral deviation perpendicular to the firing direction). By comparing the simulation results with the actual observed data, it is found that there is a small deviation between the two, and this small deviation may be due to a variety of factors, such as the variation of the ambient wind speed, the nonuniformity of the air density, or other dynamical factors that are not considered in the model. This indicates that the aerodynamic parameters obtained by the ESO-GO algorithm identification have high applicability in real ballistic simulations. This result not only confirms the accuracy of the improved algorithm in parameter identification but also demonstrates its value in practical engineering applications.

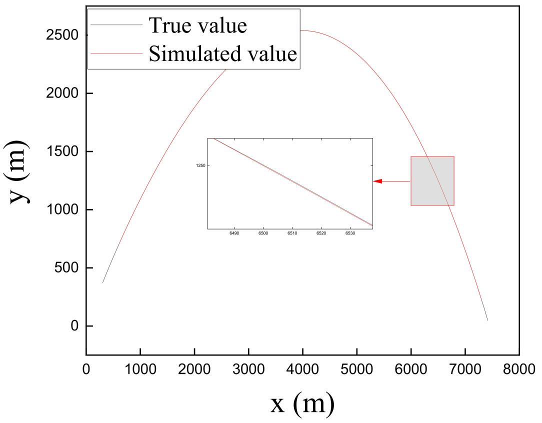


Fig. 11. Comparison between simulation data and actual data

6. CONCLUSIONS

In this study, an improved snake optimization algorithm (ESO-GO) incorporating a genetic algorithm is proposed, aiming to enhance the accuracy and global optimization capability of ballistic parameter identification. The research results show:

1. Algorithm performance advantage: The ESO-GO algorithm has significant advantages in improving the accuracy of ballistic parameter identification and avoiding falling into local optimization compared with the traditional SO method and the SO-PSO method.
2. Effectiveness of practical application: The ESO-GO algorithm is combined with Kalman filtering technology to conduct parameter identification for the actual ballistic data of a certain type of grenade. After comparative analysis, the predicted point of impact based on the algorithm has less deviation from the actual point of impact, which fully confirms that the ESO-GO algorithm is practicable and effective in the actual ballistic parameter identification task.
3. Actual data processing capability: The ESO-GO algorithm shows high accuracy and reliability in the process of processing actual flight data and can effectively deal with the data challenges brought by the complex and changeable actual flight environment.
4. Value of research results: Research findings provide a new optimization strategy for ballistic analysis and guidance system design. In the theoretical dimension, it enriches the algorithmic research system in the related fields; in the practical dimension, it builds a solid foundation for the deepening development and wide application of the subsequent related technologies.

Despite these achievements, the ESO-GO algorithm still has limitations that require further improvement. On the one hand, the algorithm assumes that aerodynamic parameters remain constant within small time intervals, which may introduce errors in transonic flight (Mach number 1.2–1.8) or high-angle-of-attack scenarios (angle of attack $> 10°$) due to unsteady aerodynamic effects. On the other hand, the algorithmic performance is sensitive to hyperparameters such as population size and maximum iterations, and it lacks an adaptive adjustment mechanism for unknown projectile types. Future research will focus on three directions: 1) Optimize the ballistic model by integrating a six-degree-of-freedom model to consider the coupling effects of projectile rotation on aerodynamic parameters, improving the algorithmic adaptability to complex flight states; 2) Introduce a reinforcement learning module to realize adaptive adjustment of hyperparameters, reducing the algorithmic dependence on empirical settings; 3) Expand the algorithm to multi-parameter identification (e.g., lift coefficient, pitching moment coefficient) and verify its performance in hypersonic projectile scenarios, further promoting the engineering application of the algorithm in the field of precision guidance.

REFERENCES

[1] Z.P. Gao and W.J. Yi, “Aerodynamic parameter identification of projectiles optimized by improved sparrow search algorithm based kernel extreme learning machine,” *J. Electron. Meas. Instrum.*, vol. 32, no. 1, pp. 28–33, 2025, doi: 10.13382/j.jemi.B2407780.

[2] L.X. Wang, R. Zhao, and Y. Zhang, “Angular acceleration estimation and aerodynamic parameter identification based on angular velocity equivalent model,” *Proc. Inst. Mech. Eng. G J. Aerosp. Eng.*, vol. 380, pp. 959–973, 2024, doi: 10.1177/09544100241249350.

[3] J.Z. Kang, S.F. Zhang, and C. Hu, “Application of aerodynamic parameter online identification in rocket ascent guidance,” *J. Harbin Eng. Univ.*, vol. 41, no. 7, pp. 1052–1058, 2020, doi: 10.11990/jheu.201901089.

[4] H.A. Saleh and R. Chelouah, “The design of the global navigation satellite system surveying networks using genetic algorithms,” *Eng. Appl. Artif. Intell.*, vol. 17, no. 1, pp. 111–122, 2004, doi: 10.1016/j.engappai.2003.11.001.

[5] Q.Z. Kang and K.J. Wang, “Research on aerodynamic parameter identification technology of controlled projectile,” *J. Ordnance Equip. Eng.*, vol. 45, no. 5, pp. 209–214, 2024, doi: 10.11809/bqzbgcxb2024.05.030.

[6] X.Q. Hou and L.M. Wang, “Application of Differential Evolution Intelligent Algorithm in High-vibration Aerodynamic Identification,” *J. Projectiles Rockets Missiles Guidance*, vol. 40, no. 3, pp. 103–107, 2020, doi: 10.15892/j.cnki.djzdxb.2020.03.024.

[7] Y.J. Mu, “Research on optimization algorithm for aerodynamic parameter identification of rocket projectiles,” PhD Thesis, Shanxi: North University of China, 2023, doi: 10.27470/d.cnki.ghbgc.2023.000308.

[8] S.X. Huang, L. Ni, and Z.P. Wang, “A novel data-driven method for online parameter identification of an electrochemical model based on cuckoo search and particle swarm optimization algorithm,” *J. Power Sources*, vol. 601, p. 234261, 2024, doi: 10.1016/j.jpowsour.2024.234261.

[9] T. Shang, L.X. Wang, Y.L. Wang, Ch. Bu, and T. Yue, “Flight dynamics modeling and aerodynamic parameter identification of four-degree-of-freedom virtual flight test,” *AIAA J.*, vol. 61, no. 6, pp. 2652–2665, 2023, doi: 10.2514/1.J062188.

[10] T.Y. Jiang, J. Li, and K.W. Huang, “Longitudinal parameter identification of a small unmanned aerial vehicle based on modified particle swarm optimization,” *Chin. J. Aeronaut.*, vol. 28, no. 3, pp. 865–873, 2015, doi: 10.1016/j.cja.2015.04.005.

[11] Y. Liu, C.N. Li, and Y. Fang, “A combined aerodynamic parameter identification method for missing test data,” *J. Northwestern Polytech. Univ.*, vol. 41, no. 2, pp. 282–292, 2023, doi: 10.1051/jnwpu/20234120282.

[12] Z.X. Wang and Y. Chen, “Quadrotor UAV control with disturbance based on aerodynamic parameter estimation,” *Inf. Control*, vol. 47, no. 6, pp. 663–670, 2018, doi: 10.13976/j.cnki.xk.2018.7490.

[13] J.S. Li, S.J. Chang, and S.F. Chen, “Identification of projectile drag coefficient based on neural network algorithm,” *J. Ballistics*, vol. 30, no. 4, pp. 38–43, 2018, doi: 10.12115/j.issn.1004-499X(2018)04-007.

[14] Z. Hui and D.Y. Du, “Research on aerodynamic parameter identification of non-stall and near-stall flight data using recurrent neural networks,” *Acta Aeronaut. Astronaut. Sin.*, vol. 46, no. 18, p. 131483, 2025, doi: 10.7527/S1000-6893.2025.31483.

[15] Y.P. Gan, Y.L. Wu, and K.J. Wang, “Optimization of boom aerodynamic parameter identification based on genetic algorithm for extreme learning machine,” *J. Ordnance Equip. Eng.*, vol. 43, no. 9, pp. 250–256, 2022, doi: 10.11809/bqzbgcxb2022.09.035.

[16] D.V. Puri, P.H. Kachare, S.B. Sangle, I. Al-Shourbaji, A. Jabbari, and R. Kirner, “A hybrid dwarf mongoose and artificial ecosystem-based EEG channel optimization for Schizophrenia detection,” *Biomed. Signal Process. Control*, vol. 112, Part A, p. 108513, 2026, doi: 10.1016/j.bspc.2025.108513.

[17] Z.P. Gao and W.J. Yi, "Optimizing projectile aerodynamic parameter identification of kernel extreme learning machine based on improved Dung Beetle Optimizer algorithm," *Measurement*, vol. 239, p. 115473, 2025, doi: 10.1016/j.measurement.2024.115473.

[18] R. Zhong and Z.M. Wang, "Self-adaptive competitive swarm optimizer: a memetic approach for global optimization and human-powered aircraft design," *Memetic Comput.*, vol. 17, p. 32, 2025, doi: 10.1007/s12293-025-00465-3.

[19] F.A. Hashim and A.G. Hussien, "Snake optimizer: a novel meta-heuristic optimization algorithm," *Knowl.-Based Syst.*, vol. 242, p. 108320, 2022, doi: 10.1016/j.knosys.2022.108320.

[20] Al-Shourbaji *et al.*, "An Efficient Parallel Reptile Search Algorithm and Snake Optimizer Approach for Feature Selection," *Mathematics*, vol. 10, no. 13, p. 2351, 2022, doi: 10.3390/math10132351.

[21] D.V. Puri, *et al.*, "Hybrid Reptile-Snake Optimizer Based Channel Selection for Enhancing Alzheimer's Disease Detection," *J. Bionic Eng.*, vol. 22, pp. 884–900, 2025, doi: 10.1007/s42235-024-00636-x.

[22] G.D. Qi, J.Q. Yu, and F.Z. Chen, "Aerodynamic parameter identification for UAV based on flight test," *J. Missiles Guidance*, vol. 39, no. 2, pp. 295–297, 2019, doi: 10.15892/j.cnki.djzdxb.2019.02.033.

[23] T. Zhou, H. Cheng, and J.Z. Zhang, "Research on identification of pneumatic servo system based on GA-BP neural network," *Mach. Tool Hydraul.*, vol. 49, no. 6, pp. 42–46, 2021, doi: 10.3969/j.issn.1001-3881.2021.06.009.

[24] D.H. Dou and J.F. Fan, "Aerodynamic identification of miniature ammunition based on AGA-BPNN," *Tactical Missile Technology*, vol. 2022, no. 3, pp. 1–8, 2022, doi: 10.16358/j.issn.1009-1300.20210079.

[25] N.G. Cui, B.G. Lu, and Y. Fu, "Aerodynamic parameter identification of a reentry vehicle based on Kalman filter method," *J. Chin. Inertial Technol.*, vol. 22, no. 6, pp. 755–758, 2014, doi: 10.13695/j.cnki.12-1222/o3.2014.06.010.

[26] Z.L. Wang, J.H. Cheng, and B. Qi, "An adaptive Kalman filtering algorithm based on maximum likelihood estimation," *Meas. Sci. Technol.*, vol. 34, pp. 115114, 2023, doi: 10.1088/1361-6501/ace9ef.

[27] H.P. Fu and Y.M. Cheng, "An enhanced adaptive Kalman filtering for linear systems with inaccurate noise statistics," *J. Control Asia*, vol. 25, no. 4, pp. 3269–3281, 2023, doi: 10.1002/asjc.2964.

[28] X.W. Wan, J. Wang, and H. Yang, "A random error compensation method of MEMS gyroscope based on BP neural network combined with PSO-Optimized Kalman filter," *J. Ordnance Sci. Technol.*, vol. 44, no. 2, pp. 556–565, 2023, doi: 10.12382/bgxb.2022.0110.

[29] O.D. Boundjou and L. Jacobsen, "Adaptive particle swarm optimization applied to aircraft control," *AIAA* 20th International Space Planes and Hypersonic Systems and Technologies Conf., Glasgow, U.K., 2015, doi: 10.2514/6.2015-3561.