Velocity Control of Near Space Vehicle Based on Enhanced Pigeon-Inspired Optimization

Meng Liu[1], Qiang Feng[1], Xingshuo Hai[1,2](✉), Yi Ren[1], and Dongming Fan[3]

[1] School of Reliability and System Engineering, Beihang University, Beijing 100191, China
haixingshuo@buaa.edu.cn

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

[3] School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

Abstract. The problem of near space vehicle (NSV) control has aroused widespread concern in recent years. However, optimal control parameters are not easy to obtain which still remains a pressing challenge. This paper addresses the issue of NSV control parameters optimization. First, the velocity control method based on an active disturbance rejection control (ADRC) technique for the longitudinal nonlinear model of the NSV is given. Then, an enhanced pigeon-inspired optimization algorithm with a golden-sine mutation mechanism and an opposition-based learning strategy (GOPIO) is presented to achieve control parameter optimization. A case study is conducted to prove the validity of the proposed method. Simulation results indicate that the local search ability is strengthened in GOPIO compared with traditional algorithms.

Keywords: Near space vehicle (NSV) · Parameters optimization · Active disturbance rejection control (ADRC) · Pigeon-inspired optimization (PIO) · Golden-sine mutation · Opposition-based learning

1 Introduction

Near space vehicle (NSV) has the advantages of low maintenance cost and multi-task processing capability, which make it widely applied in military and civilian fields. However, compared to existing atmosphere vehicles, the environment without air convection posed a great challenge to the flight control of NSV [1]. Moreover, NSV has highly nonlinear dynamics characteristics due to the strong coupling relationship between the body, propulsion system and structural dynamics, which also increase the difficulty of flight control. Velocity has a great impact on the attitude of the NSV and its control performances directly affect the flight quality [2]. Therefore, the improvement of the flight control of NSV has become a subject that needs to be further investigated.

In recent years, various control methods have emerged in order to achieve better performances of NSV flight control problems. Zhao *et al.* [3] designed an attitude control system under engine faults. Xia *et al.* [4] studied the strategy for the flight control of NSV based on a disturbance observer, which effectively reduces the impact of external

L. Yan et al. (Eds.): ICGNC 2022, LNEE 845, pp. 7070–7078, 2023.
https://doi.org/10.1007/978-981-19-6613-2_681

disturbances. Guo *et al.* [5] proposed an adaptive attitude control method that improved system tracking performances by avoiding computational scaling. Almost all of the methods discussed above cannot avoid the problem of parameters adjustment, which is a critical issue that greatly affects the performances of the controller [6].

In general, the adjustment of control parameters is regarded as an optimization problem [7]. Aiming to find a more suitable matching method for optimal control, Agamawi *et al.* [8] investigated four derivative and differential methods. Gui *et al.* [9] solved the reaction control system problem with an improved particle swarm optimization (PSO) algorithm. As a matter of fact, the same control method may lead to different results by applying different parameter adjustment approaches. As an efficient technique, heuristic algorithms play an important role in solving the problems of control parameters optimization. A proper method to optimize the parameters of the controller of NSV will improve the control performances and its applications.

By imitating the behavior of the pigeon swarm, pigeon-inspired optimization (PIO) algorithm has shown effectiveness in various scenarios [10]. Feng *et al.* [11] presented an optimization method of formation reconfiguration with an enhanced PIO algorithm for a multi-UAV system. Hai *et al.* [12] proposed a PIO with game theory for the main control parameters of mobile robot. Xu *et al.* [13] combined PIO with quantum rules to control the curved paths for underwater snake robot, which reduced the energy loss effectively. However, the solution accuracy of PIO is a concern for complex multi-dimensional optimization problems. In the vicinity of the local optimal solution, it is difficult to adjust the optimization direction appropriately, which is one of the shortcomings of the algorithm. In the domains of control and navigation of an aircraft, the property of fast convergence is necessary [14]. Thus, the advantage of extremely fast convergence makes it stand out, but also brings the risk of precocity [15]. In order to improve the performance of PIO and realize better velocity control of NSV, the improvement of the algorithm is ought to be designed.

2 Velocity Control with GOPIO

2.1 Velocity Control Model for NSV

At present, numerous controllers have shown effectiveness on velocity active disturbance rejection control for NSV. In [16], when the height command remains unchanged and the speed command is a step command, the overshoot f_O, rise time f_R, settling time f_{Set}, steady accuracy f_S and elevator command integration f_I are used as the evaluation indicators of the step response. The objective function can be calculated by:

$$fitness = w_1 f_O + w_2 f_R + w_3 f_{Set} + w_4 f_S + w_5 f_I \tag{1}$$

$$f_O = \max_{t>0} \left| \frac{V(t)}{V_c(t)} \right| \tag{2}$$

$$f_R = t_2|_{V(t_2)=0.9} - t_1|_{V(t_1)=0.1} \tag{3}$$

$$f_{Set} = \max\{t|0.05V_c \leq |V(t) - V_c|\} \tag{4}$$

$$f_S = \int_{t > f_{Set}} |V(t) - V_c| \tag{5}$$

$$f_I = \int \beta \tag{6}$$

where, W_i, $i = 1$~5 is used to measure the importance of each objective. Particularly, the weight coefficient w_4 should be larger to guarantee the stability of the response.

2.2 Overview of PIO

By imitating the behavior of pigeons, two main operators make up the traditional PIO. One is the map and compass operator which conducted in the first phase that uses the information of best position in the population. As a result, the process of finding an optimal solution is guided by the best position. The other is the landmark operator which is executed in the later of optimization. In this operator, the number of pigeons decreases sharply and moves closer to the center of the swarm.

In map and compass operator, the velocity V and the position X of a pigeon i at t iteration are computed as follows.

$$V_i(t) = V_i(t-1) * e^{-Rt} + rand(X_g - X_i(t-1)) \tag{7}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{8}$$

where, $X_i = [x_{i1}, x_{i2}, ..., x_{iD}]$, $V_i = [v_{i1}, v_{i2}, ..., v_{iD}]$, *rand* is a random number between 0 and 1, X_g denotes the current bestposition in the population, R is the map and compass factor, and D is the dimension of the problem respectively.

In landmark operator, the number of pigeons $N_{num}(t)$ decreases rapidly. As iterations t increases, they converge to the center of the population where the position and velocity are obtained by

$$X_{center}(t-1) = \frac{\sum_{i=1}^{N_{num}(t-1)} X_i(t-1) * fitness(X_i(t-1))}{N_{num}(t-1) * \sum_{i=1}^{N_{num}(t-1)} fitness(X_i(t-1))} \tag{9}$$

$$N_{num}(t) = \frac{N_{num}(t-1)}{2} \tag{10}$$

$$X_i(t) = X_i(t-1) + rand(X_{center}(t-1) - X_i(t-1)) \tag{11}$$

where, *fitness* (X_i) is the fitness value for pigeon i. X_{center} represents the center position of the swarm.

Clearly, the information of all pigeons is not fully utilized in the early stage. In fact, any pigeon may possess the best position during the iterative procedure. However, the mechanism in traditional PIO limits the capacity of the algorithm to search for a better position of other pigeons. In addition, the rapid decline in quantity of the pigeons will also results in a local optimum easily.

2.3 The Enhanced PIO with Golden-Sine Mutation and Opposition-Based Learning

To address the issue of PIO, a golden-sine mutation mechanism and the opposition-based learning strategy are applied to improve the performance of the algorithm.

Golden-Sine Mutation. Golden sine mutation can be used as a local search method to make pigeons search in a small area around their current location. In the two core operators of PIO, the specific implementation of mutation is shown in the following formula.

$$X_i^{mu}(t) = X_i(t) * |\sin R_1| + R_2 * \sin R_1 * \left| x_1 * X_g - x_2 * X_i(t) \right| \tag{12}$$

where, $R_1 \in [0, 2\pi]$, $R_2 \in [0, \pi]$. Both R_1 and R_2 are random number. x_1 and x_2 are the coefficient obtained by introducing the golden ratio.

$$\begin{aligned} x_1 &= -\pi + (1 - \tau) * 2\pi \\ x_2 &= -\pi + \tau * 2\pi \\ \tau &= \sqrt{5} - 1/2 \end{aligned} \tag{13}$$

Obviously, τ is the golden ratio. The mutation process takes place after each normal iteration of the pigeon. A greedy rule is introduced to determine whether to accept a mutated solution. The specific rules are shown in the following formula.

$$\begin{cases} X_i(t) = X_i(t),\ \textit{if fitness}(X_i^{mu}(t)) < X_i(t) \\ X_i(t) = X_i^{mu}(t), \textit{if fitness}(X_i^{mu}(t)) \geq X_i(t) \end{cases} \tag{14}$$

The introduction of golden-sine mutation has many advantages. On the one hand, the chance of finding a better position each pigeon is increased. On the other hand, it will not deviate from the optimal direction of the pigeon group due to the excessive moving range.

Opposition-Based Learning. Golden-sine mutation is more helpful for local search, but it has less improvement in population diversity. Therefore, when PIO is trapped in a local optimal solution, an effective method should be given to adjust the direction of optimization by enhancing the swarm diversity. It has been demonstrated that opposition-based learning can effectively improve the diversity and quality of populations [17]. As one of the approaches, stochastic opposition-based learning is suitable for a variety of heuristic algorithms.

In map and compass operator, pigeons with poor fitness values are still need to search for a better position. It is unnecessary to execute opposition-based learning by all pigeons in this stage. On the contrary, a pigeon in a better position has a greater risk of falling into a local optimum, and a greater probability of a better reverse solution. Thus, the elite opposition-based learning strategy can be a feasible method at this stage. For a population consists of N^{num} pigeons, individuals with $N^{num}/4$ fitness value rankings are defined as elites in which opposition-based learning is carried out in each iteration as follows:

$$X_i^{Obl}(t) = Lb + Ub - rand * X_i(t), \forall\, elite\, pigeon\, i \tag{15}$$

where, $X_i^{Obl}(t)$ is the opposite solution for i elite pigeon in t generation. Lb and Ub denote the lower and upper bounds of the variables, respectively.

In landmark operator, all pigeons optimize their position according to the following formula:

$$X_i^{Obl}(t) = Lb + Ub - rand * X_i(t), \forall i = 1, 2, ..., N_{num}(t) \tag{16}$$

Whether the learned solution is accepted or not is determined by the greedy rule.

$$\begin{cases} X_i(t) = X_i^{Obl}(t), if\, fitness(X_i^{Obl}(t) < X_i(t) \\ X_i(t) = X_i(t), if\, fitness(X_i^{mu}(t)) \geq X_i(t) \end{cases}. \tag{17}$$

By utilizing the random opposition-based learning, the search space of a pigeon is expanded, especially in the later optimization process to find better solution, it is helpful to adjust the optimization direction.

Based on golden-sine mutation and opposition-based learning, an enhanced PIO named GOPIO is proposed. The specific implementation process of the algorithm is shown as follows.

Step 1: The GOPIO algorithm starts.
Step 2: Initialization of population and parameters.
Step 3: Execute the map and compass operator.
Step 4: Execute golden-sine mutation.
Step 5: Execute opposition-based learning for elites.
Step 6: Determine whether the number of iterations meets the requirement. If yes, go to Step 7. Otherwise, turn to Step 3.
Step 7: Execute the landmark operator.
Step 8: Execute opposition-based learning for all pigeons.
Step 9: Determine whether the number of iterations meets the requirement. If yes, go to Step 10. Otherwise, turn to Step 7.
Step 10: Output the optimal solution.
Step 11: The GOPIO algorithm ends.

Among them, the maximum iteration requirements in Step 6 and Step 9 are the same as those in the two phases of traditional PIO. Figure 1 shows the specific process of GOPIO algorithm.

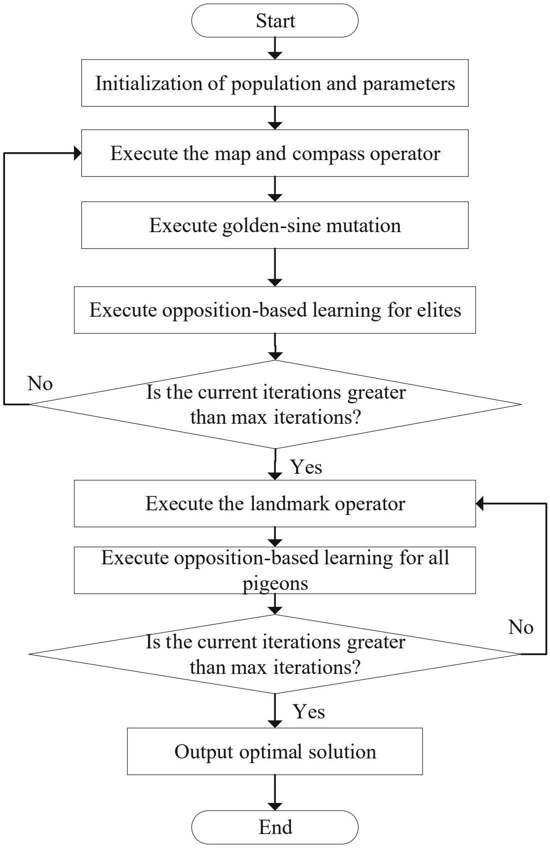

Fig. 1. The flow diagram of the GOPIO

3 Example Analysis

The six main parameters of the controller designed in reference [16] are optimized. For a certain NSV that needs velocity control, the initial state is conditions is given by Table 1.

Table 1. Initial conditions

Height	Angle of attack	Angle of pitch	Velocity
33528 m	0.0378 rad	0.0378 rad	4590 m/s

Meanwhile, some other optimization algorithms, such as traditional PIO, differential evolution algorithm (DE), and particle swarm optimization (PSO), are selected to compare with GOPIO. The parameters of various algorithms are set as shown in Table 2.

The comparative evolutionary curves are shown in **Fig. 2** to prove the superiority of GOPIO.

Table 2. Algorithm parameter description

Algorithm	Parameter setting	Value
PIO	Max iterations for map and compass operator	10
	Max iterations for landmark operator	5
	Map and compass factor	0.3
GOPIO	Proportion of elites	0.25
PSO	Inertia weight	0.5
	Self-best factor	2
	Global best factor	2
DE	Scaling factor	0.6
	Cross constant	0.5
In common	The maximum number of iterations	15
	Population size	20

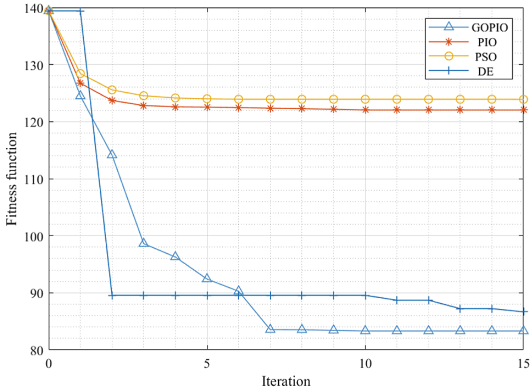

Fig. 2. Comparative evolutionary curves

As can be seen from Fig. 2, DE and GOPIO perform well in the parameter optimization of velocity loop and find satisfactory solutions, while PSO and PIO fall into local optimums. In particular, GOPIO shows the best convergence performance, but it also has the problem of convergence speed lag. Thus, GOPIO effectively improves the solution accuracy by utilizing the golden-sine mutation mechanism and an opposition-based learning strategy.

Based on the optimization results of our proposed GOPIO algorithm, the state response of velocity can be further shown in Fig. 3. Note that the input command is set to $V_c = 4650$ m/s.

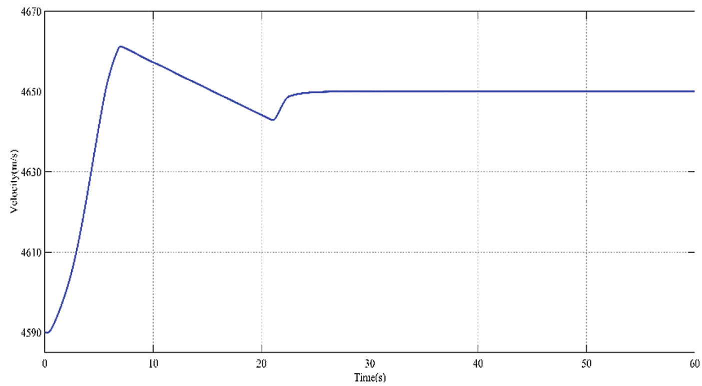

Fig. 3. The state response of velocity.

The comparative results obtained through 20 runs are listed in Table 3.

Table 3. Solution result comparison

Algorithm	Best fitness	Worst fitness	Average fitness
PSO	120.38	136.24	123.89
PIO	118.75	127.33	122.04
DE	85.53	94.36	86.65
GOPIO	**83.04**	**87.53**	**84.19**

As shown in Table 3, the best, worst, and average fitness values are given which further demonstrate the feasibility and effectiveness of GOPIO. It has great advantages for the optimization of NSV velocity control parameters.

4 Conclusion

In the interest of the control parameters optimization of NSV, an enhanced PIO algorithm based on golden-sine mutation and opposition-based learning (GOPIO) is presented. On the basis of the ADRC technique, a better control performance for the longitudinal nonlinear model is obtained. Besides, in the process of parameter optimization, GOPIO is compared with other well-known algorithms under the same condition. Simulation results including the evolutionary curves and the time-domain response verify the feasibility and effectiveness of our proposed method.

Acknowledgement. This study is supported by the research and demonstration of key technologies for the air-ground collaborative and smart operation of general aviation (No. 2022C01055).

References

1. Xia, R., Chen, M., Wu, Q.: Neural network based optimal adaptive attitude control of near-space vehicle with system uncertainties and disturbances. Proc. Inst. Mech. Eng. Part G: J Aerospace Eng. **233**, 641–656 (2018)
2. Wu, W., Wang, G.H., Sun, J.P.: Polynomial radon-polynomial Fourier transform for near space hypersonic maneuvering target detection. IEEE Trans. Aerosp. Electron. Syst. **54**, 1306–1322 (2018)
3. Zhao, J., Jiang, B., Xie, F., Gao, Z.F., Xu, Y.F.: Adaptive sliding mode backstepping control for near space vehicles considering engine faults. J. Syst. Eng. Electron. **29**, 343–351 (2018)
4. Xia, R.S., Wu, Q.X., Shao, S.Y.: Disturbance observer-based optimal flight control of near space vehicle with external disturbance. Trans. Inst. Meas. Control. **42**, 272–284 (2020)
5. Guo, J., Zhang, T., Cheng, C., Zhou, J.: Model reference adaptive attitude control for near space hypersonic vehicle with mismatched uncertainties. Trans. Inst. Meas. Control. **41**, 1301–1312 (2018)
6. Chen, M., Wu, Q., Jiang, C., Jiang, B.: Guaranteed transient performance based control with input saturation for near space vehicles. Science China Inf. Sci. **57**(5), 1–12 (2014). https://doi.org/10.1007/s11432-013-4883-9
7. Bai, C.C., Guo, J.F., Zheng, H.X.: Optimal guidance for planetary landing in hazardous terrains. IEEE Trans. Aerosp. Electron. Syst. **56**, 2896–2909 (2020)
8. Agamawi, Y.M., Rao, A.V.: Comparison of derivative estimation methods in optimal control using direct collocation. AIAA J. **58**, 341–354 (2020)
9. Gui, H., Sun, R.S., Chen, W., Zhu, B.: Reaction control system optimization for maneuverable reentry vehicles based on particle swarm optimization. Discret. Dyn. Nat. Soc. **2020**, 1–11 (2020)
10. Qiu, H., Duan, H.: Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design. Science China Technol. Sci. **58**(11), 1915–1923 (2015). https://doi.org/10.1007/s11431-015-5860-x
11. Qiang, F., et al.: Resilience optimization for multi-UAV formation reconfiguration via enhanced pigeon-inspired optimization. Chinese J. Aeronaut. **35**(1), 110–123 (2021)
12. Hai, X., et al.: Mobile robot ADRC with an automatic parameter tuning mechanism via modified pigeon-inspired optimization. IEEE/ASME Trans. Mechatron. **24**, 2616–2626 (2019)
13. Xu, B., Jiao, M.Y., Zhang, X.K., Zhang, D.L.: Path Tracking of an Underwater Snake Robot and Locomotion Efficiency Optimization Based on Improved Pigeon-Inspired Algorithm. J. Marine Sci. Eng. **10** (2022)
14. Alazzam, H., Sharieh, A., Sabri, K.E.: A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer. Expert Syst. Appl. **148** (2020)
15. Duan, H.B., Huo, M.Z., Shi, Y.H.: Limit-cycle-based mutant multiobjective pigeon-inspired optimization. IEEE Trans. Evol. Comput. **24**, 948–959 (2020)
16. Yang, Z.Y., Duan, H.B., Fan, Y.M., Deng, Y.M.: Automatic carrier landing system multi-layer parameter design based on Cauchy mutation pigeon-inspired optimization. Aerosp. Sci. Technol. **79**, 518–530 (2018)
17. Mahdavi, S., Rahnamayan, S., Deb, K.: Opposition based learning: a literature review. Swarm Evol. Comput. **39**, 1–23 (2018)