Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Self-tuning speed and flow control of micro turbojet engines based on an improved evolutionary strategy

Q. Gao, Jiahao Li & Yuxin Zhang

To cite this article: Q. Gao, Jiahao Li & Yuxin Zhang (2024) Self-tuning speed and flow control of micro turbojet engines based on an improved evolutionary strategy, Automatika, 65:3, 1154-1162, DOI: 10.1080/00051144.2024.2349869

To link to this article: https://doi.org/10.1080/00051144.2024.2349869

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

Published online: 13 May 2024.

Submit your article to this journal

Article views: 251

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=taut20

AUTOMATIKA
2024, VOL. 65, NO. 3, 1154–1162
https://doi.org/10.1080/00051144.2024.2349869

OPEN ACCESS Check for updates

Self-tuning speed and flow control of micro turbojet engines based on an improved evolutionary strategy

Q. Gao, Jiahao Li and Yuxin Zhang

School of Energy and Power Engineering, Xihua University, Chengdu, People's Republic of China

ABSTRACT

At present, the controllable parameters of micro turbojet engines in engineering applications are mainly speed-fuel flow (hereinafter referred to as flow) control, in which closed-loop proportional–integral–derivative (PID) control is mostly used to achieve a stable control of engine speed under slow engine conditions. For the optimal adjustment of PID parameters, this paper designs an improved evolutionary strategy for the self-tuning of control parameters in the engine speed and flow control system and formulates an improved PI controller based on a neural network. The simulation experimental results show that the method can realistically achieve stable and fast control of the engine under above slow conditions.

ARTICLE HISTORY
Received 16 August 2023
Accepted 24 April 2024

KEYWORDS
Micro turbojet engine; improved PID controller; improved evolutionary strategy

1. Introduction

At present, micro turbojet engines have been widely used in small UAVs [1], small target aircraft and missiles and other types of aircraft [2], drones have a small size, flexible, low maintenance and manufacturing costs, a wide range of scenarios and easy to use, etc., and are currently used in a large number of military, weather detection and address survey and many other fields [3,4]. For micro turbojet engine control system in the slow and above state, the micro turbojet engine realizes the closed-loop control of engine speed and flow rate through proportional–integral–derivative (PID) link. However, the traditional engineering applications of PID controllers have problems such as poor robustness and difficult parameter tuning. With the improvement of hardware performance of electronic controllers, the use of intelligent PID in engine controllers has become possible.

Micro turbojet engines operate in a wide range of operating conditions at and above slow speed, and their dynamic characteristics change continuously with changes in flight conditions and operating conditions. The single-point controller designed by a small regional linearization model cannot meet the control requirements of the engine in the full speed range in the speed domain above the slow speed. To solve this problem, some scholars use the gain scheduling method to dynamically adjust the control parameters under different operating conditions according to the target speed range. However, this method still has some limitations, especially, if the speed adjustment is large. For example, there is still a possibility that the overshoot exceeds the maximum overshoot, which could cause the engine damage in the actual test run. To address this problem, this paper proposes an improved PI controller based on neural network to better adapt to the nonlinear system of the engine [5,6].

For the selection of PI parameters in each operating interval, most of the current engineering uses the trial-and-error method for selection, which has strong subjective randomness, poor portability and the selected parameters are generally not optimal control parameters in the global range. With the improvement of computer performance, Particle Swarm optimization, Genetic Algorithm and Evolutionary Strategies are widely used in the calculation of PID parameter adjustment [7–9]. Some scholars have used the genetic algorithm to adjust the PID controller parameters, but the genetic algorithm itself still suffers from the problem of being trapped in local optimum and the loss of good individuals due to random variation [10,11]. To address this problem, in the present work, an improved evolutionary strategy is proposed to optimize the pre- and post-iteration processes of the evolutionary strategy, respectively.

2. Micro turbojet engine system modeling

A micro turbojet engine studied in this paper is shown in Figure 1.

If the micro turbojet engine is working in the slow engine condition or above, some scholars have tried to use BP neural to establish the engine speed prediction model and achieved high prediction accuracy [12], but because the engine runs with certain nonlinearity, the generalization ability of BP neural network is poor

CONTACT Q. Gao hyqgxu@126.com School of Energy and Power Engineering, Xihua University, 610039 Chengdu, People's Republic of China

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

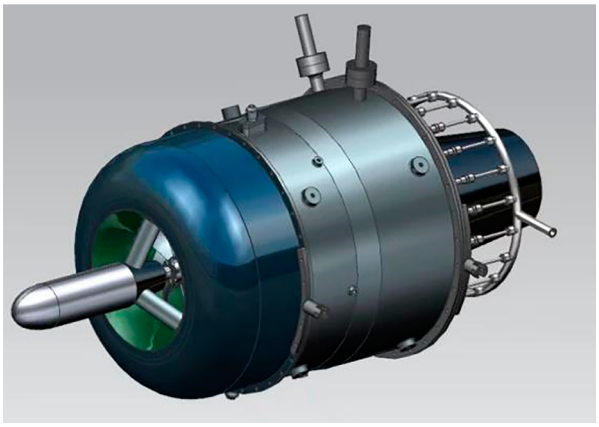

Figure 1. A certain type of micro turbojet engine.

compared with RBF neural network and RBF has global approximation ability, so this paper uses RBF neural network to establish its speed prediction model. The engine works in a relatively stable steady-state condition under slow running conditions with stable combustion efficiency, and generally does not show large fluctuations. Therefore, when the actual test data are available, the RBF neural network with good nonlinear fitting ability is used to build the engine speed prediction model to have a better prediction accuracy.

2.1. Introduction to RBF neural networks

RBF neural network was proposed by Moody and Darken in 1988, and has been receiving academic attention. It is a three-layer feed-forward network with a single hidden layer that can approximate any function with arbitrary accuracy and has some advantages in pattern classification. It consists of an input layer, a hidden layer, and an output layer. The signal is passed through the input layer to the hidden layer, and it is the hidden layer function (usually radial basis function) that determines the characteristics of the hidden layer. The signal output from the hidden layer is linearly operated by the output layer to obtain the output of the RBF neural network, which is the linearly weighted sum of the hidden layer outputs. The structure of the RBF neural network is illustrated in the Figure 2.

From the above figure, we can see that the RBF neural network mainly contains two parts, the hidden layer and the output layer, in addition to the input layer. The second layer is the hidden layer, and the hidden layer

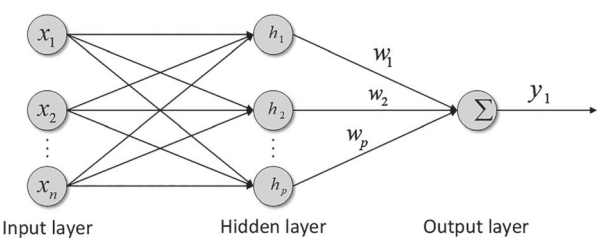

Figure 2. RBF neural network structure.

mainly consists of radial basis functions, which can realize the nonlinear mapping from the input layer to the output layer. At present, the radial basis function is most often used as the basis function is Gaussian kernel function, and its formula is as follows:

$$G(x) = \exp\left(-\frac{1}{2\sigma^2}||x_n - h_p||^2\right) \tag{1}$$

In the above equation, x_n is the nth input point, h_p is the pth centroid, and σ is the centroid width. The width of the centroid determines the sensitivity of the radial neuron, which is activated only when the distance of the given input point from the centroid is less than the width of the radial base neuron, thus indicating that the RBF neural network is a local approximation network. After passing the hidden layer output through a fully connected layer, the output of the RBF neural network can be finally obtained as:

$$y_j = \sum_{i=1}^{h} w_{ij} \exp\left(-\frac{1}{2\sigma^2}||x_n - h_p||^2\right); j = 1, 2, 3 \ldots n \tag{2}$$

Other important radial basis functions are:

(1) Inverse S-type function

$$\varphi(x) = \frac{1}{1 + e^{\frac{x^2}{\sigma^2}}} \tag{3}$$

(2) Proposed multi-quadratic function

$$\varphi(x) = \frac{1}{(x^2 + c^2)^{1/2}} \tag{4}$$

The selection of radial basis functions for RBF neural networks needs to be decided according to the actual problem, and the performance gap between different kernel functions is large. In this paper, we use Gaussian basis functions, which are simple in form and not too complicated even when there are more input variables.

2.2. RBF neural network prediction model building

The operation of a miniature turbojet engine has strong nonlinear characteristics, so the RBF neural network can be better applied to this nonlinear system than the multi-segment tacho partition transfer function model established by the system identification method. Like the system identification method, the RBF neural network is used to build the speed prediction model for the micro turbojet engine, which only requires a large amount of test data in the full speed domain of the engine for the training of the model. The accurate speed prediction model built by RBF neural network can be effectively used for simulation experiments and validation of speed control algorithms.

For a micro turbojet engine, the fuel supply and its corresponding speed at a certain time can be used as the

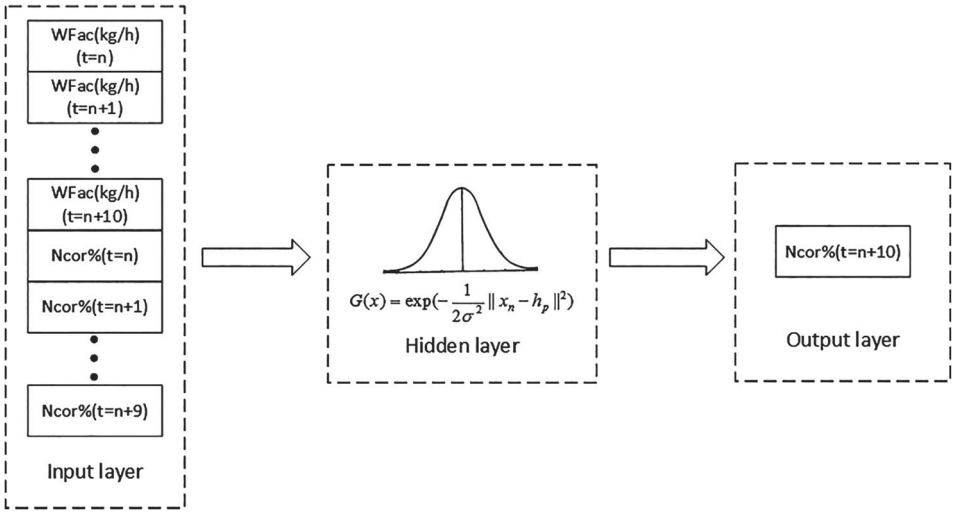

Figure 3. RBF neural network input/output structure.

input value of the neural network, and the speed at the next time can be used as the output value of the neural network to build a speed prediction model for the micro turbojet engine. Based on the above principle, the sampling interval of 0.25s is taken as the input feature value of the RBF neural network, and the fuel supply at a certain sampling moment ($t = n$) and the fuel supply for the next 10 sampling periods ($t = n + 10$), as well as the actual RPM value for the next 9 sampling periods ($t = n + 9$) are selected as the input feature values of the RBF neural network, and the RPM value for the 10th sampling time after that moment ($t = n + 10$) after that moment as the output values of this RBF neural network. The structure is shown in the Figure 3.

In this paper, we use matlab for programming and simulation experiments. The newrbe function that comes with matlab software can build a radial basis function more easily and quickly, and the newrbe function automatically increases the number of neurons according to the input vector of the training set when building a radial basis neural network, so that it can minimize the error.

2.3. RBF neural network speed prediction model simulation validation

In this paper, we use the open-loop control test data of a certain engine under slow-motion conditions and above as the model sample, and divide it into two parts: the training set data are firstly brought into the model for training, and then the test set data are used for testing to verify the accuracy of the model. The test data and the prediction results of the test set obtained by the model are shown in the Figure 4.

From the above figure, it can be seen that when using the test set data for speed simulation, the overall error is small, the maximum error does not exceed 3%, and the average error is 1.35%, and its prediction accuracy meets the control performance for the subsequent speed control algorithm tests.

3. Improving the design of evolutionary strategies

3.1. Traditional evolutionary strategy design

The basic methods of traditional evolutionary strategies are:

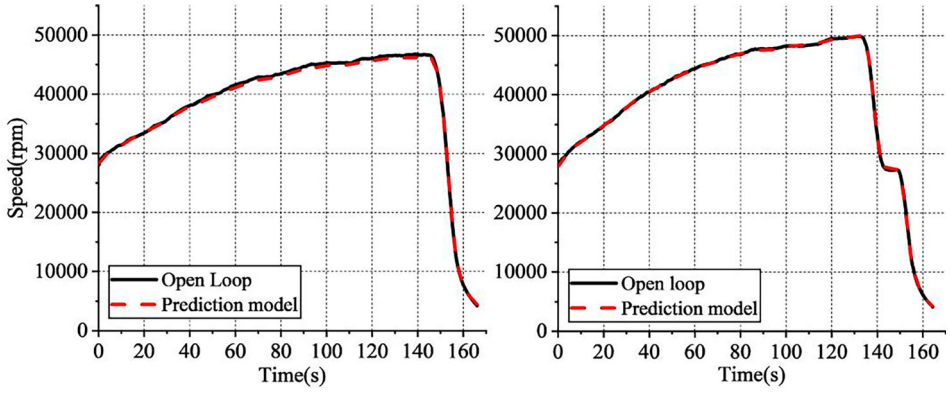

Figure 4. Model test set prediction results.

(1) Determining the variable space: the parameters K_p, T_t, and T_d in the PID controller are used as the parameters to be optimized for the evolutionary strategy and as the population composition of the evolutionary strategy.

(2) Confirmation of the target adaptation function: In the PID control link, the system transition time can be used as the performance indicator of the system, the transition time refers to the period from the occurrence of disturbance to the controlled quantity and the establishment of a new equilibrium state. It is an important indicator of the rapidity of the transition process, so it can be used as the target adaptation function, namely: Fitness $= \frac{1}{t_s}$; Furthermore, the primary goal of the system is to ensure that the control process can converge, so the need to join the penalty function for whether the oscillation, when the system oscillation will be adapted to the minimum 0; at the same time, the overshoot of the system should not be too large, in the actual turbojet engine engineering control, if the overshoot is too large may cause the engine to be rich in oil for a short period of time, may cause the engine to overheat and damage the engine, so also need to join the penalty function for the overshoot, based on actual engineering experience overshoot should not exceed 110% of the target value that is:

$$\text{Fitness} = 0\,,\ \sigma_p > 1.1\text{sv} \tag{5}$$

(3) Confirm the basic parameters of the evolutionary strategy: population ecological niche size POP_size = 30, number of iterations N_generation = 30, and generation of offspring of size K_size = 30.

In the first step, a random initial population is created with its corresponding variation intensity, after which the second step starts to generate new populations from the parents, the third step the population performs mutation and crossover to achieve population evolution, the fourth step calculates the fitness of each individual, the fifth step iterates the population according to the fitness and repeats the second to fifth steps until the calculation is completed [13].

In the design of the above traditional evolutionary strategy, it is easy to fall into the situation that the PID parameters are locally optimal when applied to the self-integration of the PID parameters, and there is a convergence boundary problem when the PID parameters are integrated. This will lead to the elimination of some individuals who may be in the optimal convergence region but have not reached the optimal point because of their low fitness, and the top ranked individuals will form a genetic monopoly and lead to the problem that the final result is a local optimal solution instead of a global optimal solution. In addition, when the number of iterations enters the middle and late stages, the variation cross evolution of the population becomes completely random, which may lead to the deterioration of the best individuals and their elimination, as well as the slow convergence of the population.

3.2. Improving evolutionary strategies

To address some of the problems in the design of the traditional evolutionary strategy in Section 3.1 above, this paper designs an improved evolutionary strategy to optimize the traditional evolutionary strategy.

The improved evolutionary strategy divides the computational process into two major parts, the pre-iteration process and the post-iteration process, whose specific functions and design steps are as follows.

3.2.1. Improving the pre-evolutionary strategy iteration process

The pre-iterative process will optimize two major aspects of individual variation rate and population iteration selection.

(1) Individual variation rate improvement: variation rate is determined by introducing a probability function together with the current fitness of the individual and the median fitness of the whole population, so that individuals with fitness lower than the median fitness of the current population have a larger variation rate; while individuals with fitness higher than the median fitness of the current population slow down their variation rate to a certain extent according to their fitness, where the variation probability function is shown below [14]:

$$\text{mut\_}P_m = \begin{cases} \text{mut\_}P_{m1}, & Fitness < Fitness_M \\ \text{mut\_}P_{m2}, & Fitness \geq Fitness_M \\ mut\_P_{m3}, & Fitness = 0 \end{cases} \tag{6}$$

$$\begin{cases} \text{mut\_}P_{m1} = 2 + C_1 \\ \dfrac{Fitness_M - Fitness}{Fitness_M}, & 0 < C_1 < 2 \\ \text{mut\_}P_{m2} = 2 - C_2 \\ \dfrac{Fitness - Fitness_M}{Fitness}, & 0 < C_2 < 1 \\ mut\_P_{m3} = 3 \end{cases} \tag{7}$$

where C_1 and C_2 are penalty coefficients, adjusting the values of C_1 and C_2 can change the rate of population variation, the larger the C_1 is the greater the rate of variation of individuals with backward adaptation, and the larger the C_2 is the slower the rate of variation of individuals with forward adaptation. In this paper, we use $C_1 = 1, C_2 = 1$.

(2) Population iteration selection improvement: The population iterations are selected by a population probability selection function. The population probability selection function makes its overall semi-normal distribution as follows:

$$p_m = F(\mathrm{m}), \mathrm{m} = 1, 2, 3 \ldots \mathrm{POP\_size} \quad (8)$$

$$F(m) = C_4 e^{-C_3 m^2} \quad (9)$$

C_3 is a distribution parameter, and adjusting the value of C_3 can change the weight of population selection. Increasing C_3 can make the population selection more uniform and individuals with lower fitness have a higher probability of being selected, and decreasing the value of C_3 can make the population iterative selection more biased towards individuals with higher fitness, while C_4 needs to be adjusted so that it satisfies:

$$\int_0^{POP\_size+K\_size} F(m)dm \approx POP\_size \quad (10)$$

In the present work, $C_3 = -0.0008$, $C_4 = 1$.

3.2.2. Improving the evolutionary strategy late iteration process

The later iterative process follows the earlier iterative process, and the later iterative process mainly addresses the problem of losing good genes and slowing down the overall population convergence due to the completely random evolution of individuals in the traditional evolutionary strategy [15], so this paper introduces a directional evolutionary link at the beginning of the later iterative process:

In the directed evolutionary link, the variation of the individual is introduced into the gradient calculation, and after the gradient calculation is introduced, the evolutionary direction of the individual is selected by calculating the gradient to evolve in the optimal direction for the current evolutionary link; the gradient is calculated as follows:

$$\mathrm{gradN} = \frac{Fitness_{N+1} - Fitness_N}{\mathrm{distance}} \quad (11)$$

Among them: distance $= \mathrm{mut}\_P_m$

The gradient calculation can be used to obtain the fastest convergence direction under this iteration calculation. The calculation steps of the improved evolutionary strategy after the introduction of the improvement link are shown in Table 1.

Table 1. Steps to improve evolutionary strategy.

Algorithm 2: Improved evolutionary strategies

Require: POP_size //Population size
Require: N_generation //Number of iterations
Require: K_size //Offspring size
Require: random pop //Initial random population
Repeat:
if (Enter the later iteration)
kids = get_dicMut(pop,K_size)
else
kids = make_kids(pop, K_size)
pop_mut(pop)
endif
Fitness = get_fitness(pop)
mut_strength of $x_{i,N} \leftarrow \begin{cases} \mathrm{mut}\_P_{m1}, & Fitness < Fitness_M \\ \mathrm{mut}\_P_{m2}, & Fitness \geq Fitness_M \\ mut\_P_{m3}, & Fitness = 0 \end{cases}, i =$ 0, 1, 2 . . . *pop*
Select samples by Probability selection function $p_m = C_4 e^{-C_3 m^2}$
Untile: stop iteration
Return: pop

3.3. Improved PI speed controller design based on neural network

3.3.1. Traditional PID controller design

PID control is obtained by giving the difference between a target input value and the actual output value of the current system, and then correcting the output value by three links: proportional, integral, and differential, respectively, so that it gradually approaches the target input value. One of the traditional PID formulas is as follows:

$$y(t) = K_p \left[e(t) + \frac{1}{T_t} \int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt} \right] \quad (12)$$

From the above equation, it can be seen that the key parameters of the PID control link is in the selection of the rectification of the K_p, T_t and T_d parameters. The main role of each of these links is as follows:

(1) Proportional link: the proportional link reflects the basic deviation of the system, and the larger the proportional coefficient, the faster the adjustment speed, but when the proportional coefficient is too large, it will make the system less stable and even cause the system to have unstable oscillations;
(2) Differential link: The differential link responds to the rate of change of the deviation signal of the system and has the foresight to predict the trend of deviation change, thus producing an override control effect, and before the deviation is formed, it has been eliminated by the differential adjustment effect, so it can improve the dynamic performance of the system. However, the differential has an amplifying effect on noise disturbance, and strengthening the differential is unfavorable to the system anti-disturbance [16].

The basic principle of PID control is shown in the Figure 5.

3.3.2. Improved PI speed controller design based on neural network

Due to the strong nonlinear characteristics of the engine in different speed domains, it is difficult to achieve optimal control in the full speed domain using a single PI control parameter, and there is even a risk

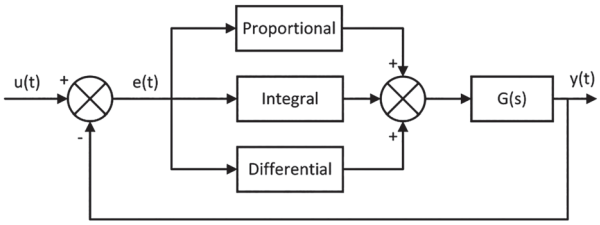

Figure 5. The basic principle of PID control.

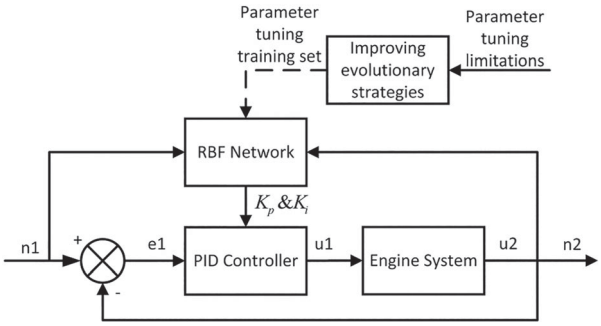

Figure 6. Improved PI closed-loop speed controller structure.

of engine damage due to uncontrolled speed control or excessive overshoot due to the change of engine characteristics. The miniature turbojet engine can be considered as a benign nonlinear system in the above-slow-speed domain, and the engine is divided into multiple speed domains to calculate the optimal PI control parameters under different input conditions by the improved evolutionary strategy designed in the previous subsection and based on the RBF neural network speed prediction model designed in Section 4.2. Finally, the improved neural network-based variable parameter PI speed controller can match the appropriate PI parameters under different input conditions.

The structure of the closed-loop speed controller after the introduction of the improved neural network-based PI speed controller is shown in the Figure 6. After the target speed is input, the target speed n1 and the actual speed n2 are input to the neural network gain scheduler in a single control cycle, and then the PI parameters of the current control cycle are obtained and substituted into the PI controller.

According to the actual engine design requirements, the engine transition state control speed overshoot should be no more than 2% in the speed range of 27,500–40,000 rpm, no more than 1.5% in the speed range of 40,000–50,000 rpm, and no more than 1% in the speed range of 50,000–55,000 rpm. The optimal PI parameters for each speed are obtained by using the improved evolutionary strategy according to this constraint, and are substituted into the RBF neural network as the training set for training. Some of the PI parameters obtained by the improved evolutionary strategy are shown in Tables 2 and 3.

The improved PI speed controller is obtained by bringing the superscript PI parameters into the neural network for training. The improved PI controller can realize variable parameter PI control under different input conditions, so that the engine can realize stable and fast speed transition control under different input conditions in each speed domain under slow running and above operating conditions.

4. Experimental results and analysis

4.1. Improving the pre-evolutionary strategy iteration process

Firstly, the simulation experiments are conducted without adding the post-improvement iterative process, in which the pre-improvement iterative process mainly addresses the problem that the traditional evolutionary strategy is easy to fall into local optimum, and the genetic diversity of the whole population can be reflected by analyzing the Euclidean distance variance of the population in the iterative process. Its Euclidean distance variance is calculated as follows:

(1) First, the $K_{p,m}$, $T_{i,m}$, and $T_{d,m}$ in DNA are normalized by the following normalization formula:

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, x = K_{p,m}, T_{i,m}, T_{d,m}, \quad m = 1, 2, 3 \ldots \text{POP\_size} \tag{13}$$

(2) The Euclidean distance variance is calculated from the normalized $K_{p,m}$, $T_{i,m}$, and $T_{d,m}$ values with the following equation:

$$\text{distance\_x} = \sqrt{\begin{matrix}\mathrm{F_p(K_{p,norm,m}) + F_i(T_{i,norm,m})} \\ \mathrm{+F_d(T_{d,norm,m})}\end{matrix}} \tag{14}$$

$$\begin{cases} \mathrm{F_p(K_{p,norm,m})} = \left[\dfrac{\mathrm{(K_{p,norm,m} - K_{p,norm,avg})^2}}{\mathrm{POP\_size}}\right]^2 \\ \mathrm{F_i(T_{i,norm,m})} = \left[\dfrac{\mathrm{(T_{i,norm,m} - T_{i,norm,avg})^2}}{\mathrm{POP\_size}}\right]^2 \\ \mathrm{F_d(T_{d,norm,m})} = \left[\dfrac{\mathrm{(T_{d,norm,m} - T_{d,norm,avg})^2}}{\mathrm{POP\_size}}\right]^2 \end{cases}, \quad m = 1, 2, 3 \ldots \text{POP\_size} \tag{15}$$

The main reason for normalizing K_p, T_i, and T_d separately is that in the actual calculation process generally the value of K_p is larger than the value of T_i or T_d. Therefore, in order to equally reflect the degree of difference between the three values and avoid the difference of a certain value being ignored, they should be normalized. The simulation results of the average variation of its Euclidean distance variance with the number

Table 2. Partial Kp parameters for each speed domain.

n1 (rpm)	n2 27,500rpm	n2 35,000rpm	n2 40,000rpm	n2 45,000rpm	n2 50,000rpm	n2 55,000rpm
27,500	–	0.0619	0.0697	0.0695	0.0611	0.0532
35,000	0.0543	–	0.0685	0.0652	0.0463	0.0429
40,000	0.0487	0.0451	–	0.0395	0.0379	0.0366
45,000	0.0436	0.0403	0.0241	–	0.0231	0.0286
50,000	0.0386	0.0212	0.0196	0.0208	–	0.0116
55,000	0.0274	0.0186	0.0152	0.0149	0.0131	–

Table 3. Partial Ki parameters for each speed domain.

n1 (rpm)	n2 27,500rpm	n2 35,000rpm	n2 40,000rpm	n2 45,000rpm	n2 50,000rpm	n2 55,000rpm
27,500	–	0.0112	0.0107	0.0104	0.0110	0.0125
35,000	0.0112	–	0.0111	0.0131	0.0136	0.0162
40,000	0.0121	0.0140	–	0.0166	0.0176	0.0191
45,000	0.0128	0.0149	0.0171	–	0.0240	0.0256
50,000	0.0168	0.0169	0.0211	0.0313	–	0.0407
55,000	0.0217	0.0248	0.0284	0.0368	0.0535	–

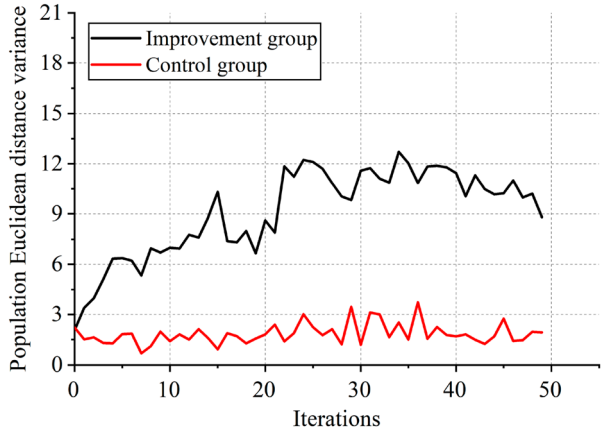

Figure 7. Variation of population Euclidean distance variance.

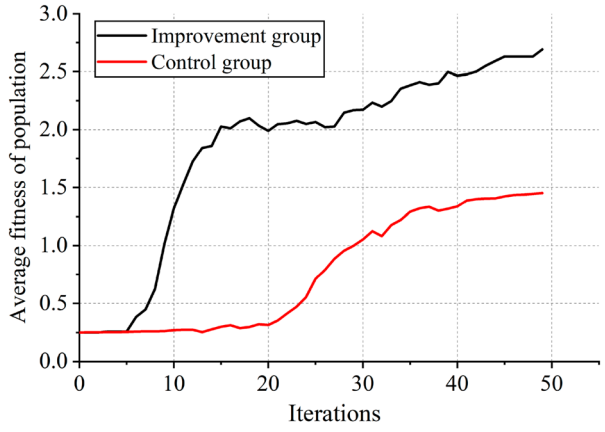

Figure 8. Plot of population adaptation.

of iterations are shown in Figure 7, and the average simulation results of the variation of the fitness with the number of iterations are shown in Figure 8.

From the above figure, it can be seen that after the introduction of the improved pre-iteration process, the population genetic diversity of the improved group is better than that of the control group, and the population Euclidean distance variance of the control group is always lower and its genetic diversity is poorer. In addition, it can be seen that the population variance peaks between about the 23rd and 26th iterations, when the population genetic diversity is at its maximum and the population individuals are widely distributed among multiple convergence domains, and the directional variation link in the later iterative process can be introduced in the subinterval to accelerate the population convergence and prevent the loss of excellent genes due to random variation.

The graph above shows the change of population fitness, and it can be seen that the average speed of population fitness convergence is better than that of the traditional evolutionary strategy after introducing only the pre-improvement iteration process.

4.2. Improving the evolutionary strategy late iteration process

According to the simulation results, due to the strong stochasticity of the iterative calculations process itself, the later iterative improvement link can be introduced at the 24th iteration. The simulation results of the change in the average fitness of the population after its introduction are shown in Figure 9.

From the simulation results of the fitness change in the above graph, we can see that the fitness increases

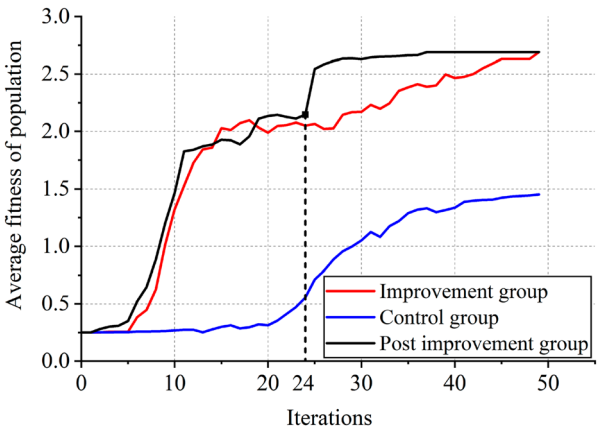

Figure 9. Plot of change in population fitness after the introduction of the post-improvement link.

faster after the introduction of the post-improvement link in the 24th iteration than the improved evolutionary strategy without the post-improvement link and the traditional control evolutionary strategy; and the fitness increases smoothly and rapidly after the introduction of the post-improvement link with almost no fluctuation.

4.3. Improved PI speed controller simulation

The closed-loop control strategy designed in Section 3.3 is simulated numerically by programming in matlab. The simulation results are obtained by comparing with the single parameter PI controller as shown Figure 10.

As can be seen from the simulation results above, when using only a single PI parameter, the overshoot is too large in the low speed region, exceeding the maximum limit overshoot, while its response time is too long in the high speed domain. After adding the improved gain scheduler, the engine has a relatively stable control effect in different speeds. Afterwards, the simulation results are compared with the conventional gain scheduler PI controller as Figure 11.

From the simulation results, we can see that the control effect of the traditional gain scheduling PI controller and the improved gain scheduling PI controller is similar when the difference between the target speed

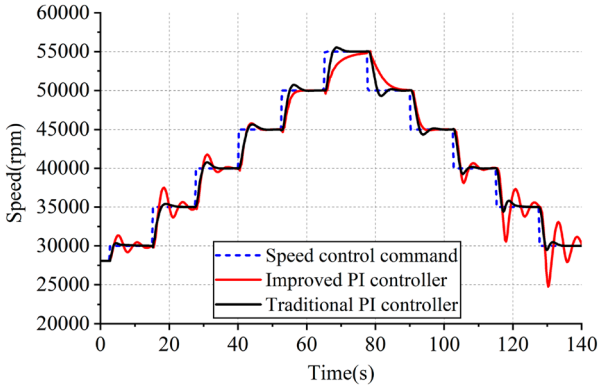

Figure 10. Comparison of simulation results for a single PI parameter controller.

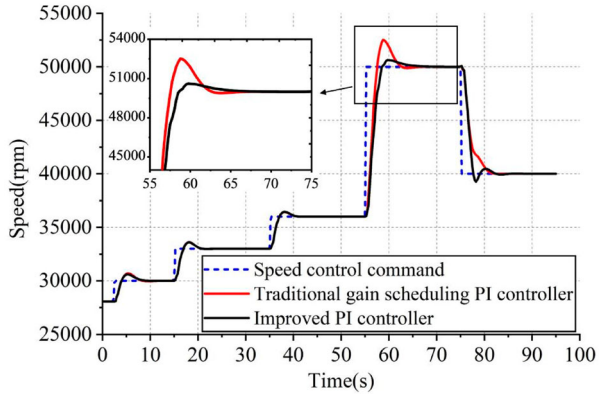

Figure 11. Comparison of simulation results of conventional gain scheduling PI controller.

control value and the current actual speed is small. However, when the speed adjustment is larger, the overshoot of the traditional gain scheduling PI controller is about 5% which exceeds the maximum limit overshoot of the engine, while the improved gain scheduling PI controller based on neural network can obtain a smoother and faster control effect under different input conditions.

5. Conclusions

In the closed-loop control process of slow train, optimization is carried out to address some shortcomings of traditional PID closed-loop control. Firstly, an improved evolutionary strategy was designed. Compared to traditional evolutionary strategies and genetic algorithms, the improved evolutionary strategy has stronger optimal search ability and can converge faster in the later iterative calculation process; Afterwards, the algorithm is used to calculate the optimal PI control parameters of the engine under different input conditions in various speed domains, and an improved PI speed controller based on neural networks is established. Finally, simulation experiments show that the closed-loop control algorithm can achieve optimal control of the engine in various speed domains, without the problem of a single parameter PI control with excessive overshoot in the low speed domain and a long adjustment time in the high speed domain. At the same time, it can avoid the problem of overshoot exceeding the maximum limit overshoot of the engine when the single speed adjustment is large, and has great practical value in engineering applications.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Wang CL, Dai J, Wang TS, et al. Current development state and investigation of application technique for micro turbojet engine. Album of the 8th China aeronautical society youth Science and Technology Forum. China Aviation Publishing & Media Co.; 2018; pp. 5–9.
[2] General Aviation Manufacturers Association. (2018). 2018 Annual Report.
[3] Li FL, Shen QT, Xu LS. Nonlinear model predictive pressure for grouting system. J Syst Simul. 2008;20(23):6535–6506.
[4] Lyon DH. A military perspective on small unmanned aerial vehicles. IEEE Instrum Meas Mag. 2004;7(3): 27–31. doi:10.1109/MIM.2004.1337910
[5] Vojtech V, Adrian IK. Gain-scheduled PID controller design. J Process Control. 2013;23(8):1141–1148.
[6] Pan ZR. Design and verification of electronic control unit for micro turbojet engine. Dalian University of Technology; 2022.
[7] Zhang W. Comparison and research of auto-tuning PID control algorithm. Shenyang University of Technology; 2021.

[8] Cnfnas N. Judaism as a group evolution strategy: A critical analysis of Kevin MacDonald's theory. Human Nat. 2018;29(2):1–23.
[9] Wang W, Yang N, Automation SO. Research on PID parameter tuning based on improved genetic algorithm. Comput Digit Eng. 2018;46(12).
[10] Zhao GY, Wang C. Optomization of PID controller parameters based on improved seeker optimization algorithm. Comput Simul. 2020;37(08):302–305+310.
[11] Fu Z, Zhuan X. PID parameter self-tuning algorithm based on neural network and genetic algorithm. Eng J Wuhan Univ. 2023;56(03):379–386.
[12] Wang T. Intelligent control of variable cycle engine based on dynamic neural network. Dalian University of Technology; 2020.
[13] Wang Z, Cao L, Si H. An improved genetic algorithm for determining the optimal operation strategy of thermal energy storage tank in combined heat and power units. J Energy Storage. 2021;43:103313, doi:10.1016/j.est.2021.103313
[14] Chen W, Xu XL, Zhong XW, et al. Parameter tuning of PID controller for annular inverted pendulum system based on improved genetic algorithm. Comput Simul. 2021;38(03):165–169.
[15] Junchi F, Lei Y. Genetic algorithm improvement in test data generation. J Comput-Aided Design Comput Graph. 2015;27(10):2008–2014.
[16] Li ZZ, Zhu HM. Design of fuzzy PID controller. Telecom Power Technol. 2019;36(11):48–50.