Complex & Intelligent Systems (2021) 7:1797–1803
https://doi.org/10.1007/s40747-020-00190-z

ORIGINAL ARTICLE

# Adsorption control of a pipeline robot based on improved PSO algorithm

Yilin Yu[1] · Yanli Xu[1] · Fusheng Wang[1] · Wensheng Li[2] · Xiaoming Mai[2] · Hao Wu[2]

Received: 2 April 2020 / Accepted: 17 August 2020 / Published online: 31 August 2020

## Abstract

Particle swarm optimization (PSO) is a widely used method that can provide good parameters for the motion controller of mobile robots. In this paper, an improved PSO algorithm that optimize the control PID parameters of a specific robot have been proposed. This paper first presents a brief review of recently proposed PSO methods, and then presents a detailed analysis of the PID optimization algorithm, which uses H∞ theory to reduce the search space and fuses the information entropy to ensure the diversity of particles. Simulations in Matlab show that the algorithm can improve the convergence speed and get a better global optimization ability than the standard PSO algorithm. Experimental results present a sound effects for the control of the negative pressure adsorption motor in the power grid pipeline robot during its adsorption along the circular movements, which verifies the effectiveness of the proposed method.

**Keywords** Particle swarm optimization · Pipeline robot · Parameter optimization · PID control · Surface adsorption · H∞ theory

## Introduction

When adsorption robots carry out circular movement along the inner wall of the electric power pipeline [1], it is necessary to adjust the parameters of the adsorption motor in real time, so as to control the appropriate adsorption force, thus the robot can stably adsorb on each position of the inner wall of the pipeline under the composite action of gravity and suction [2]. In actual motion, the robot often appears adsorption instability or motion jam, due to the time delay of control parameters [3]. This problem of the control system is mainly due to the delay between the feedback and the system output. For the classical PID control algorithm used for the adsorption motor, the main problem is the optimization of key parameters, whose quality directly affects the performance of the system, and even leads to the divergence and oscillation of the control system [4]. Therefore, the parameter optimization of the controller plays an important role for this kind of time-delay system [5], and meanwhile the high-order and nonlinearity of the control make it very difficult to adjust the parameters [6].

The traditional PID parameter optimization methods include stable boundary method, attenuation curve method, dynamic characteristic method and Ziegler Nichols empirical method, etc. [7–9]. In recent years, there are many PID parameter tuning methods based on artificial intelligence technology [10], such as expert system method. The PSO PID control algorithm proposed in reference [11] has a good control effect on the process control with large time delay.

In reference [12], a chaos PSO algorithm based on chaos optimization is proposed, which can directly identify the closed-loop transfer function of the controlled object with time delay without transforming it into state equation. In reference [13], a simple method based on PSO is proposed to determine the stability of time-delay systems, where PSO is used to search the characteristic roots of quasi characteristic polynomials. In reference [14], a method of parameter estimation based on PSO is proposed to solve the problem of step response identification for low order delay systems. In reference [15], taking the linear motor servo system as the research object, a PSO PID controller parameter optimization algorithm is proposed, which optimizes the number of particles and decreases the inertia weight by sections. In ref-

✉ Yanli Xu
xuyanlilynn@hotmail.com

[1] Northeast Forestry University, Harbin, China

[2] Electric Power Research Institute of Guangdong Power Grid Co., Ltd, Guangzhou, China

erence [16], the fuzzy PID control method based on PSO is applied to the design of motor controller.

In this paper, aiming at the problem that the pipeline robot moves along the pipeline due to poor control of the negative pressure adsorption motor, a PID control algorithm based on an improved particle swarm algorithm (PSO) is proposed to solve the problem of robot movement along the pipe wall. The rest of this paper is organized as follows. In section "Related work", we discuss the related work. In section "The basic idea of improved particle swarm optimization algorithm", we introduce the basic idea of improved PSO algorithm. Section "Parameter optimization of the PID based on the improved PSO algorithm" describes parameter optimization of the PID based on the improved PSO. We validate the performance of the proposed method by simulations and experiments in section "Simulations and experiments" and we conclude in section "Conclusions".

## Related work

Particle swarm optimization algorithm which is proposed by Eberhart and Kennedy, is a collective optimization algorithm and a research algorithm inspired by the foraging behavior of the bird flock [17]. In PSO algorithm, every positioning particle is a potential solution to a question. Every particle has its own position and speed and produces random solution at the phase of initialization [18–20]. Speed stands for the rapidness and slowness of the motion and position indicates the direction of the motion. The optimal solution searched independently by each particle is called individual extremum and the optimal individual extremum in the particle swarm is taken as the global optimal solution. With the continual iteration, the speed and position are improved and the optimal solution is obtained to satisfy the terminal conditions.

The formula representing the speed and position of PSO algorithm is as follows:

$$v_{ij}^{k+1} = \omega v_{ij}^{k} + c_1 \left( pbest_{ij}^{k} - x_{ij}^{k} \right) rand_1 (0, 1) + c_2 \left( gbest_{j}^{k} - x_{ij}^{k} \right) rand_2 (0, 1) \tag{1}$$

In which, *Vid* represents the velocity of the particle i in the d dimension, $\omega$ is the inertia weight, $c1$ and $c2$ are parameters for regulating the relative importance of $P_{id}$ and $P_{gd}$, $rand()$ is a random number between 0 and 1. Thus, the next position of the particle can be obtained:

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \tag{2}$$

It can be seen from Eqs. (1) and (2) that the moving speed of particles is determined by three parts, its original speed *Vid*, the best distance to itself ($P_{id}-X_{id}$) and the best distance to the group ($P_{gd}-X_{id}$). And the relative importance is determined by weight coefficient $\omega$, $c_1$ and $c_2$ respectively. The standard PSO algorithm is described as follows:

Step 1 Initialize all particles, set the population size as S, and the dimension of search space is D. Give each particle an initial position Pi and initial velocity at random, and the best value in Pi is recorded as Pg;

Step 2 Regenerate the speed and position of particles according to Eqs. (1) and (2). Calculate the velocity of particles to make sure the value is between $[- V_{\max}, V_{\max}]$; calculate the position of particles to make sure the value is between $[- X_{\max}, X_{\max}]$;

Step 3 Calculate the adaptation value of each particle according to the objective function;

Step 4 For each particle, compare the adaptation value calculated in step 3 with the best adaptation value Pi experienced by the individual. If it is better than Pi, make it the best place to be Pi; also compare the adaptation value with the group best adaptation value Pg. If it is better than Pg, make it the best place to be Pg;

Step 5 Judge whether the termination conditions of algorithm is reached. If termination condition is achieved, output relevant results, if not, turn to step 2.

PSO algorithm reveals good effect for parameter optimization in robot control. Reference [21] presents an efficient and fast method for tuning the controller parameters of robot manipulators in constrained motion. The stability of the robotic system is proved using a Lyapunov-based impedance approach whereas the optimal design of the controller parameters are tuned, in offline, by a PSO algorithm [22, 23]. Also PSO algorithm is used to design a structure-specified mixed H2/H infinity controller for control balancing of Bicyrobo, which is an unstable system associated with many sources of uncertainties due to un-model dynamics, parameter variations, and external disturbances [24]. Reference [25] considers a robot path planning problem originated from a robot factory inspection scenario for a dynamic uncertain environment, in which an improved PSO algorithm with a random positive feedback factor in velocity updating is used to optimize the path. A method based on PSO is proposed to search the unknown parameters in cartesian path planning of free-floating space robot without affected by the initial parameters [26]. A PSO-based fixed structure H infinity controller [27] is proposed to control the developed arm exoskeleton, which can avoid permanent damages to the device and user.

## The basic idea of improved particle swarm optimization algorithm

PSO algorithm reveals good adaptability in controlling parameter optimization and is widely applied in the scientific researches. However, the parameters in PSO algorithm must be carefully selected, such as the situation of swarm number. If the swarm is relatively small, it will cause the problem of partial optimization. And if the swarm is too large the convergence speed will slow down. Therefore, it is important to choose the system parameters according to different systems.

In view of the problems in the standard PSO algorithm, it is improved and optimized to apply into the PID parameter design. Firstly, H∞ theory is introduced to determine the initial searching range of PSO algorithm along with the information entropy to evaluate and adjust the initialized particle swarm. In this way, the initialized swarm becomes more dispersed, the convergence speed of PSO algorithm is improved, and the global searching ability is achieved. When applied in delay PID parameter design, the delay problem can be better improved.

The basic flow chart of parameter optimization using the improved PSO method is shown below:

## Parameter optimization of the PID based on the improved PSO algorithm

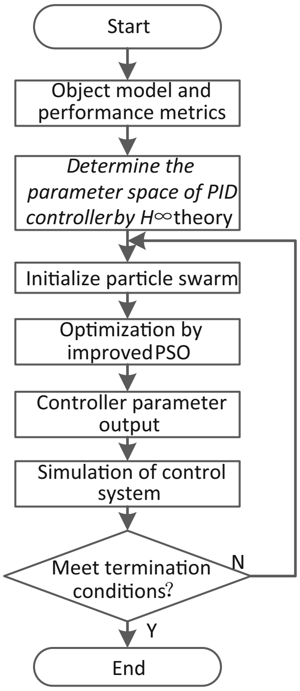
### Initializing methods of information entropy based on H∞ theory

In 1981, Zames proposed H∞ theory and widely applied in the field of controlling. While applied in the PID parameter design, H∞ theory can reduce the initial searching radius, but its random sorting does not suit the situation of the diversity of particles. Shann put forward a concept of information entropy, which well solved the problem of information measurement and maintained the uncertainty of the system. The feature of uncertainty can be utilized to deal with the problem of low degree of disorder at the stage of swarm initialization. Information entropy represents the degree of disorder to optimize the initial swarm of PSO algorithm, and improve the searching speed, convergence and the global optimization ability of searching space.

The concrete implementation procedures of PSO algorithm with initialization function are shown as follows:

*Step 1* Modify the initialization parameters of PSO algorithm: swarm number $N$, searching space dimension $D$, maximum iteration number *iter_max*, lower limit and higher limit of searching space $[L_\mathrm{d}, H_\mathrm{d}]$, minimum and maximum of flight velocity $[v_{\min}, v_{\max}]$, learning factor $c_1$,$c_2$, weight coefficient $w^{iter}$, entropy threshold of inter-particle similarity $E_{ps}$ and entropy threshold of internal parameter $E_{kp}$, $E_{ki,} E_{kd}$, etc..

**Fig. 1** Basic flow chart of parameter optimization using the improved PSO method


Start
Object model and performance metrics
Determine the parameter space of PID controllerby H∞theory
Initialize particle swarm
Optimization by improved PSO
Controller parameter output
Simulation of control system
Meet termination conditions?
N
Y
End

*Step 2* Identify the searching range of particle swarm based on H∞ theory, and initialize the particles randomly with information entropy theory $x_i$,$1 \leq i \leq$ N.

*Step 3* Calculate the similarity entropy between particles $S_{ps}$, internal parameter entropy $S_{kp}$, $S_{ki}$ and $S_{kd}$. Judge whether they are within the threshold range. If within the range, enter step 4. If not step 2 is resumed.

*Step 4* Calculate the adaptation value of each particle according to the selected adaptative function and compare the individual extremum *pbest* and global extremum *gbest* of particle swarm.

*Step 5* Regenerate the speed and position of particles according to Eqs. (1) and (2). Calculate the renewal adaptation value of particles.

*Step 6* Judge whether the termination conditions of algorithm is reached. If maximum iteration number is achieved, calculation accuracy is less than ε or the maximum stopping pace of optimal solution is $\Delta t$, turn to step7.

*Step 7* Discriminate optimization again. If necessary, redefine the particle searching space range with the aid of the present optimization results and return step 1. If not, stop the optimization (Fig. 1).

### Delay PID parameter design based on the improved PSO algorithm

Aiming at solving the problems of great searching extent, poor searching abilities and failure in convergence in the process of designing big delay PID control parameter by PSO algorithm, PID design based on the improved particle swarm algorithm is put forward. It is applied to adsorb the robot's main motor control system and the parameters $K_\mathrm{P}$,$K_\mathrm{I}$, $K_\mathrm{D}$

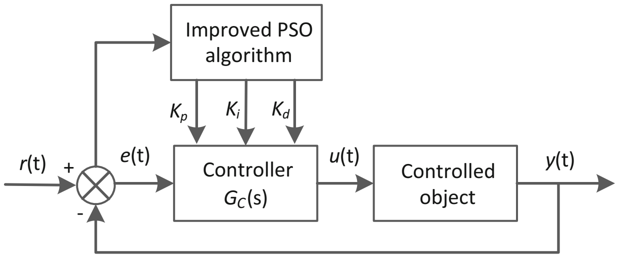


**Fig. 2** Diagram of PID control system based on the improved PSO algorithm

are set and adjusted. The diagram of control system of PID parameter design based on the improved PSO algorithm is shown in Fig. 2.

## Evaluation function selection

The three parameters $K_\mathrm{p}$, $K_\mathrm{i}$, $K_\mathrm{d}$ are optimized by the real coding of PSO, and the particle coding strings [$K_\mathrm{p}$, $K_\mathrm{i}$, $K_\mathrm{d}$] of PSO algorithm are formed by these parameters. The optimization of the control parameters aims at leading the controlling deviation of the step response to zero, enabling rapid response speed and less and even no overshooting. Performance index of error absolute value time integration is adopted as the minimum target function of parameter selection and the quadratic term of controlling input is added to the target function to limit the excess controlling energy. The following formula is used as the optimal index of parameter selection [11].

$$J = \int_0^\infty \left( w_1|e(t)| + w_2u^2(t)\mathrm{d}t + w_3 * t_u \right) \quad (3)$$

In the formula, $e\,(t)$ indicates system deviation, $u(t)$ is the controller output, $t_u$ is the rising time, and $w_1$, $w_2$, $w_3$ are correlation weights. The emphasis points of performance are adjusted with the setting of the weights.

Punitive function is used to avoid overshooting. That is, if overshooting occurs, its value is taken as one of the optimal index. And the optimal index at the moment is:

$$J = \int_0^\infty \left( w_1|e(t)| + w_2u^2(t) + w_4|ey(t)| \right)\mathrm{d}t + w_3 * t_u \quad (4)$$

In this formula, $w_4$ is set much larger than $w_1$, and $ey\,(t) = y_{out}(t) - y_{out}(t-1)$, $y_{out}(t)$ as the controlled object output. The minimum value of target function derived by the algorithm corresponds to the optimization results of PID parameters. The design objective of PID is to minimize the index function $J$ of system performance. And the conditions of minimum phase and gain margin must be satisfied to keep the robustness of the system.

**Table 1** Results of parameter optimization by different methods

| Algorithm | $K_\mathrm{P}$ | $K_\mathrm{I}$ | $K_\mathrm{D}$ | $J$ |
|---|---|---|---|---|
| PSO | 40.561 | 0.021 | 0.421 | 22.13 |
| Improved PSO | 59.524 | 0.032 | 0.499 | 22.01 |

## Algorithm realization

The calculation procedures of PID controller parameter design are shown as follows:

*Step 1* According to $K_\mathrm{p}$, $K_\mathrm{i}$, $K_\mathrm{d}$ parameter design, modify the initialization parameters of swarm algorithm: swarm number is set as N = 30, searching space dimension is $D = 3$, maximum iteration number is *iter_max* = 100, lower limit and higher limit of searching space are $L_\mathrm{d}$, and $H_\mathrm{d}$ respectively, among which $L_\mathrm{d} = \left[K_\mathrm{p}^{\min}, K_\mathrm{i}^{\min}, K_\mathrm{d}^{\min}\right]$, $H_\mathrm{d} = \left[K_\mathrm{p}^{\max}, K_\mathrm{i}^{\max}, k_\mathrm{d}^{\max}\right]$, flight velocity $v_i = [v_{i1}, v_{i2}, v_{i3}]$ ranges $[v_{\min}, v_{\max}]$, learning factor is $c_1 = c_2 = 2$, weight coefficient decrease linearly, i. e. $w^{iter} = 0.9{-}0.4t/iter\_max$, entropy threshold of inter-particle similarity $E_{ps}$ and entropy threshold of internal parameter $E_{kp}$, $E_{ki,}E_{kd}$, etc.

*Step 2* Identify the searching range of particle swarm based on H∞ theory, and initialize the particle swarm objectively with information entropy theory $x_j = \left[K_p^j, K_i^j, K_d^j\right], 1 \leq j \leq N.$

*Step 3* Calculate the similarity entropy between particles $S_{ps}$, internal parameter entropy $S_{kp}$, $S_{ki}$ and $S_{kd}$. Judge whether they are within the threshold range. If within the range, enter step 4. If not step 2 is resumed.

*Step 4* Calculate the adaptation value of each particle according to the selected adaptative function and compare the individual extremum *pbest* and global extremum *gbest* of particle swarm.

*Step 5* Regenerate the speed and position of particles according to Eqs. (1) and (2). Calculate the renewal adaptation value of particles.

*Step 6* Judge whether the termination conditions of algorithm is reached. If maximum iteration number is achieved, calculation accuracy is less than ε or the maximum stopping pace of optimal solution is $\Delta t$, turn to step7.

*Step 7* Discriminate optimization again. If necessary, redefine the particle searching space range with the aid of the present optimization results and return step 1. If not, stop the optimization.

## Simulations and experiments

In order to verify the effectiveness of the PID parameter optimization method based on the improved PSO algorithm described above, this paper selects a typical second-order

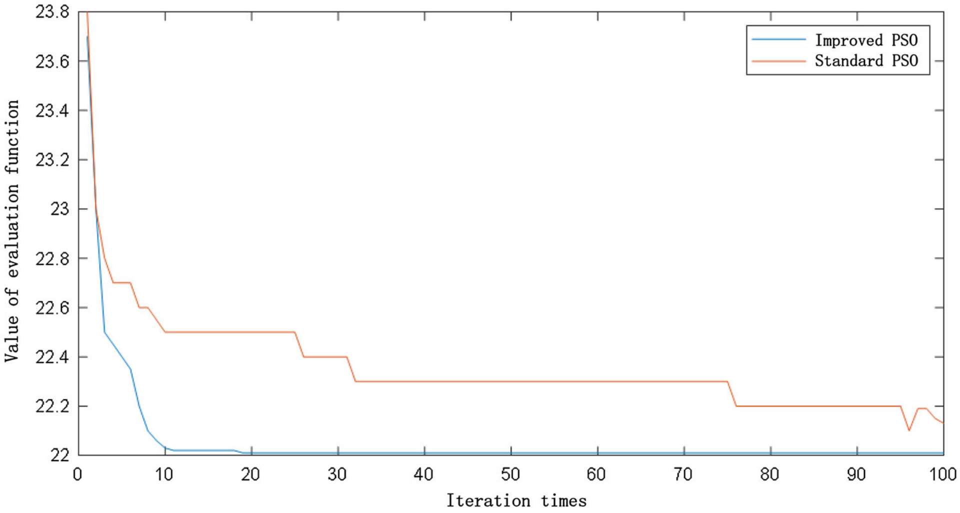


**Fig. 3** Comparison of evaluation function by different methods

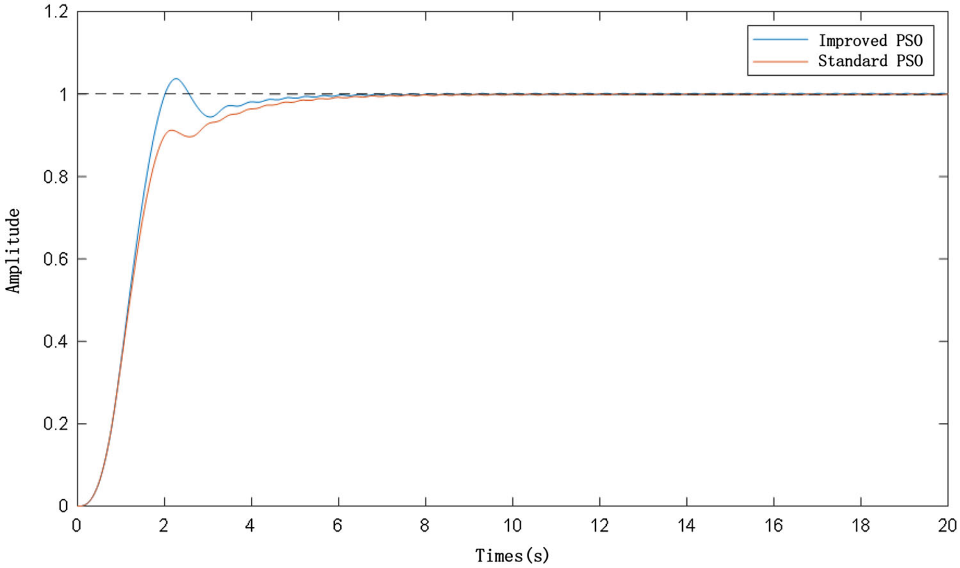


**Fig. 4** Comparison of step response by different methods

inertial time-delay controlled object of the adsorption robot main motor, and performs optimization and calculation of PID parameters. The experiment uses Intel Core i7-6700 2.6 GHz CPU, 8GRAM PC, Windows 10 64-bit operating system, and Matlab software to implement the algorithm in this paper.

$$G(s) = \frac{400}{s^2 + 50s} \tag{5}$$

In order to facilitate the simulation operation, the input signal of the model is set as a unit step signal, and the sampling period is 2 ms. We use the improved PSO algorithm and the standard PSO algorithm of this paper to tune the PID controller parameters of the controlled object, and the respective step response control effects are compared.

In the process of optimizing PID parameters, the search spaces of $K_\mathrm{P}$, $K_\mathrm{I}$, and $K_\mathrm{D}$ are [0, 60], [0, 1], and [0, 1], respectively. The number of individuals (particles) in the initial population is $N = 30$, and the number of optimization iterations is $G = 100$. The parameters of the evaluation function are $w_1 = 0.999$,$w_2 = 0.001$, $w_4 = 100$, and $w_3 = 2.0$. After 100 iterations of the algorithm, the optimal solution is shown in Table 1. Figure 3 shows the convergence trend of the evaluation function values of two different algorithms to optimize the PID parameters, and Fig. 4 is a comparison chart of the unit step response of the system after PID parameter tuning.

From the convergence trend curve of the evaluation function in Fig. 3, it can be concluded that the convergence speed and search ability of the tuning PID controller parameters improved by the improved PSO algorithm are superior to the traditional PSO algorithm. The iteration time can be quickly reduces for 30 to 10. From the comparison of the results in Fig. 4 and Table 1, it can be seen that the improved PSO

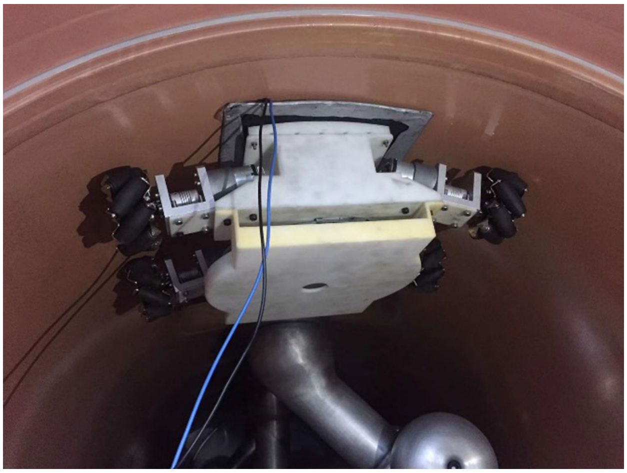


**Fig. 5** Adsorption and walking movements of the robot

algorithm PID controller is superior to the traditional particle swarm algorithm PID controller in terms of system response adjustment time and rise time indicators. As a whole, the step response control of the improved particle swarm controller is better than the basic PSO algorithm controller, and it has a good control effect.

The algorithm is transplanted to a controller based on the ARM architecture, and the robot is controlled to walk along the pipeline from 0° to 360°. Experiments show that the algorithm can meet the speed control of the brushless DC motor during the adsorption process of the robot from stationary to upside down. The effect of experimental adsorption is shown in Fig. 5.

## Conclusion

Aiming at the problem that the pipeline robot moves along the pipeline due to poor control of the negative pressure adsorption motor, a PID control algorithm based on an improved particle swarm algorithm (PSO) is proposed to solve the problem of robot movement along the pipe wall. The improved algorithm uses H$\infty$ theory to reduce the search space and fuses the information entropy to ensure the diversity of particles. A PSO algorithm with initial optimization is proposed. The parameters obtained are used to control the speed of the motor in the PID parameter tuning. After simulation in Matlab, experiments show that the algorithm can improve the convergence speed and have a better global optimization ability than the standard PSO algorithm. At the same time, the algorithm can be used to effectively solve the problem of robot motion freeze when it is applied to a power grid pipeline detection robot. Also, this parameter optimization method by Improved PSO Algorithm is believed to be effective in the similar dynamic motion control.

**Acknowledgements** Research presented in this paper was funded by the Science and Technology Project (GDKJXM20173040) of Guangdong Power Grid Co. Ltd.

## Compliance with ethical standards

**Conflict of interest** The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## References

1. Deng J, Chen J, Wang D (2019) Mechanism design and mechanical analysis of multi-suction sliding cleaning robot used in glass curtain wall. Comput Syst Sci Eng 34(4):201–206
2. Su Y, Hou X, Li L (2020) Study on impact energy absorption and adhesion of biomimetic buffer system for space robots. Adv Space Res 65(5):1353–1366
3. Wang L, Lam H-K (2018) A new approach to stability and stabilization analysis for continuous-time Takagi-Sugeno fuzzy systems with time delay. IEEE Trans Fuzzy Syst 26(4):2460–2465
4. Dian S, Liang W, Zhao T (2018) Finite-time stability and stabilization of interval type-2 fuzzy systems with time delay. J Intell Fuzzy Syst 35(6):6537–6549
5. Yang Z, Chen Y (2016) Improved particle swarm optimization algorithm and its application in PID tuning. Control Eng 23(02):161–166
6. Li X, Li G, Mao L, Xue J, Meng H (2013) Model free control of nonlinear large time delay systems using improved particle swarm optimization algorithm. J Intell Syst 8(03):254–260
7. Tan J, Dian S, Zhao T (2018) Stability and stabilization of T-S fuzzy systems with time delay via Wirtinger-based double integral inequality. Neurocomputing 275(1):1063–1071
8. Druzhinina OV, Sedova NO (2017) Analysis of stability and stabilization of cascade systems with time delay in terms of linear matrix inequalities. J Comput Syst Sci Int 56(1):19–32
9. Sipahi R, Niculescu S-I, Abdallah CT (2011) Stability and stabilization of systems with time delay limitations and opportunities. IEEE Control Syst Mag 31(1):38–65
10. Dai B, Zhou Z, Chen Y, Li Y (2019) Three degree of freedom helicopter control based on variable Universe Fuzzy PID. Control Eng 26(11):2061–2066
11. Crichigno FJM (2017) Applying extended Oxley's machining theory and particle swarm optimization to model machining forces. Int J Adv Manuf Technol 89(1):1127–1136
12. Wang Q, Sun J (2015) Closed loop time-delay system identification based on improved particle swarm optimization algorithm. J East China Univ Sci Technol 41(02):159–165

13. Mao P, Xiao Y (2012) Application of particle swarm optimization in the design of stable time-delay systems. Comput Appl Res 29(01):214–216
14. Li M, Bai M, Junjun LV (2019) Identification of low order delay systems based on particle swarm optimization. Pattern Recogn Artif Intell 32(6):524–530
15. Huang Z, Luo C (2020) PID setting of PSO linear motor based on optimizing particle number and segment weight. Modular Mach Tool Autom Process Technol 02:96–100
16. Yu Z, Zhou F (2019) Research on electric load simulator based on particle swarm optimization fuzzy PID. Electron Measure Technol 42(10):16–21
17. Omidinasab F, Goodarzimehr V (2020) A Hybrid particle swarm optimization and genetic algorithm for truss structures with discrete variables. J Appl Comput Mech 6(3):593–604
18. Cui Z, Zhang J, Wu D (2020) Hybrid many-objective particle swarm optimization algorithm for green coal production problem. Inf Sci 518(5):256–271
19. Dereli S, Koker R (2020) A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm. Artif Intell Rev 53(2):949–964
20. Kaitwanidvilai S, Chanarungruengkij V, Konghuayrob P (2020) Remote sensing to minimize energy consumption of six-axis robot arm using particle swarm optimization and artificial neural network to control changes in real time. Sens Mater 32(2):499–510
21. Mehdi H, Boubaker O (2011) Impedance controller tuned by particle swarm optimization for robotic arms. Int J Adv Rob Syst 8(5):93–103
22. Bakhale M, Hemalatha V, Dhanalakshmi S (2020) A dynamic inertial weight strategy in micro PSO for Swarm robots. Wirel Pers Commun 110(2):573–592
23. Balochian S, Baloochian H (2019) Social mimic optimization algorithm and engineering applications. Expert Syst Appl 134(11):178–191
24. Thanh BT, Parnichkun M (2008) Balancing control of Bicyrobo by particle swarm optimization-based structure-specified mixed H2/H infinity control. Int J Adv Rob Syst 5(4):395–402
25. Xu W, Li C, Liang B (2008) The Cartesian path planning of free-floating space robot using particle swarm optimization. Int J Adv Rob Syst 5(3):301–310
26. Li W, Tan M, Wang L (2019) A cubic spline method combing improved particle swarm optimization for robot path planning in dynamic uncertain environment. Int J Adv Rob Syst 16(6):1729881419891661
27. Silawatchananai C, Parnichkun M (2019) Haptics control of an arm exoskeleton for virtual reality using PSO-based fixed structure H infinity control. Int J Adv Rob Syst 16(3):1729881419849198

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.