Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Missile fixed-structure μ controller design based on constrained PSO algorithm

Min Zhang, Zhenzhen Ma, Liang Chen & Xin Chen

To cite this article: Min Zhang, Zhenzhen Ma, Liang Chen & Xin Chen (2017) Missile fixed-structure μ controller design based on constrained PSO algorithm, Automatika, 58:3, 312-320, DOI: 10.1080/00051144.2018.1432332

To link to this article: https://doi.org/10.1080/00051144.2018.1432332

AUTOMATIKA, 2018
VOL. 58, NO. 3, 312–320
https://doi.org/10.1080/00051144.2018.1432332

Taylor & Francis
Taylor & Francis Group

REGULAR PAPER

OPEN ACCESS Check for updates

Missile fixed-structure μ controller design based on constrained PSO algorithm

Min Zhang, Zhenzhen Ma, Liang Chen and Xin Chen

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

ABSTRACT
This paper provides a method of fixed-structure μ controller design with a novel particle swarm optimization (PSO) technique. The structured synthesis problem based on optimization is formulated and solved by a kind of constrained PSO algorithm which is relatively simple and without any new parameters added in the objective function. The experimental results tested on a set of 12 benchmark functions show that the proposed algorithm can outperform others in most cases. What is more, an air-to-air missile longitudinal channel control structure based on the proposed algorithm is studied and the simulation results are compared with other algorithms. The final comparison results also show the superior performance improvement over other algorithms.

ARTICLE HISTORY
Received 21 September 2017
Accepted 23 January 2018

KEYWORDS
Fixed-structure controller; μ synthesis; constrained optimization; particle swarm optimization; air-to-air missile

1. Introduction

During the last decades, as one kind of robust approaches, the H_∞ control theory has been the most widely employed technique in controller design for system under uncertain and disturbed conditions [1,2]. However, H_∞ controller may lead to unnecessary conservatism since it usually ignores robust performance design requirement and emphasizes on stability too much. To solve this problem, μ synthesis theory is proposed, where structured uncertainty is considered so that the controlled system can analyse "the worst disturbance" without conservatism [3]. In comparison with H_∞ control theory, μ theory has distinct advantages, which directly introduces the structured uncertainty and simultaneously analyses robust stability and robust performance [4,5].

On the other hand, it is well known that the controllers obtained either by H_∞ or μ theory have problems when directly applied in practical control engineering framework due to computer resources and on-site controller tuning limitations [6,7]. Hence, controller order reduction needs to be adopted as a systematic posterior step, but it will also bring unwanted performance and robustness degradation [8].

In order to overcome the problems above, fixed-structure robust controller design method has been proposed, which satisfies not only stability but also robust specifications. In Hwang and Hsiao [9], a derivative gain should be chosen in advance by the designer. Then, the maximum allowable proportional and integral gains are obtained. What is more, the choice of good initial derivative gain is a crucial factor. Ho [10] presented a synthesis of H_∞-PID controllers based on the generalized Hermite Biehler theorem for complex polynomials. Similarly, a suitable proportional gain should be chosen in advance, which makes the reasonable selections of proportional gain to be considerably important. However, the works mentioned above only tune fixed-structure controller manually, which is a time-consuming task.

Contrary with the above deterministic approach, Kennedy and Eberhart [11] proposed a particle swarm optimization (PSO) algorithm. The PSO algorithm is a swarm intelligence technique, which is one of the evolutionary computation algorithms. It is suitable to solve the design problem which is inherently NP hard. However, it cannot handle the problem with rank constraints, which could be a serious barrier from practical use. Michalewicz, Dasgupta, Riche and Schoenauer [12] augmented the cost function with penalty functions to handle the optimization problems subject to a set of constraints. Soufiene Bouallègue, Joseph Haggège and Mohamed Benrejeb [13] applied the above method to fixed-structure H_∞ controller design. However, the values of the penalty parameters are usually difficult to choose. To overcome this, a great amount of time has been made. Cai Hairan [14] proposed a new adaptive penalty function method. This method adjusts the penalty factor dynamically according to constraint particle proportion at each iteration step, which has become a widely adopted penalty function method and solved the difficulty of selecting penalty factor. The deficiency of the algorithm is that its convergence performance has very strong dependence on the parameter α, which is usually chosen according to engineering experience. Sedlaczek and Eberhard [15] developed an augmented Lagrangian PSO (ALPSO) algorithm. Kim, Maruta and

CONTACT Min Zhang zhangmin@nuaa.edu.cn

Sugie [16] applied it to PID controller design. The ALPSO algorithm is based on the assumption that the objective function is differentiable and the algorithm becomes complex when the augmented Lagrangian parameters need to be determined. Maruta, Kim and Sugie [17] proposed a modified PSO to simplify the ALPSO, which is still complex to some extent.

In this paper, a new approach is proposed to design fixed-structure controllers using μ specifications, based on a novel constrained PSO algorithm. In Section 2, fixed-structure μ synthesis problem is described. In Section 3, we explain the basic framework of our proposed algorithm. In Section 4, we describe 12 benchmark problems and the simulation results are compared with other constrained optimization algorithms. In Section 5, we apply our proposed algorithm to the air-to-air missile control design progress. In Section 6, we conclude our proposed algorithm.

2. Fixed-structure μ synthesis problem

2.1. Preliminaries

Structured singular value μ is an effective tool for the robust analysis of a system. It can analyse the robust stability and robust performance of the system simultaneously. For any multi-input linear feedback systems with uncertainty, they can always be presented by Figure 1,

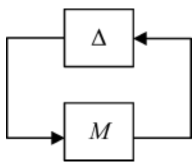

Figure 1. Standard feedback control structure.

where M denotes system transfer function matrix which is made up of controller and generalized controlled object. where $\Delta \in B\Delta^{-}$ denotes the model uncertainty, indicating the deviation degree between actual controlled object and the nominal of the mathematical model:

$$\bar{\Delta} = \left\{ \begin{array}{l} \mathrm{diag}\left(\overbrace{\Delta_1, \ldots, \Delta_1}^{m_1} \overbrace{\Delta_2, \ldots, \Delta_2}^{m_2} \Delta_3 \ldots, \Delta_{n-1}, \overbrace{\Delta_n, \ldots, \Delta_n}^{m_3} \right) \\ \Delta_j \in C^{k_j \times k_j}, \bar{\sigma}(\Delta_j) \le \delta, \delta \in R, \delta > 0 (j = 1, 2, \ldots, n) \end{array} \right\} \tag{1}$$

$$B\bar{\Delta} = \left\{ \Delta \in \bar{\Delta} \,|\, \bar{\sigma}(\Delta) \le 1 \right\} \tag{2}$$

$$n_1 = \sum_{j-1}^{n} m_j \times r_j, n_2 = \sum_{j-1}^{n} m_j \times k_j, \Delta \in C^{n_2 \times n_1}$$

and $\bar{\sigma}(\Delta)$ means the max singular value of Δ.

Suppose that the system uncertainty Δ conforms to formula (2), and then define the structured singular value of the system transfer function matrix M for formula (3):

$$\mu(M) := \begin{cases} 0, \forall \Delta \in \bar{\Delta}, \det(I - M\Delta) \ne 0 \\ \left\{ \min\left(\bar{\sigma}(\Delta) \,|\, \det_{\Delta \in \bar{\Delta}} (I - M\Delta) = 0 \right) \right\}^{-1}, \text{else} \end{cases} \tag{3}$$

While solving μ value by formula (3) directly is very difficult and any precise mathematical analytic methods is not available now. The value of μ is usually obtained by solving its upper and lower bounds, as long as the difference between which is small enough. This approximate method is feasible and effective, which is called "D-K" iterative algorithm and was put forward by Doyle [18].

2.2. Problem formulation

The general scheme of robust control problem is shown in Figure 2:

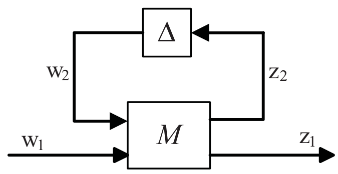

Figure 2. Scheme for robust control.

where $M(s)$ denotes the obtained generalized plant, which represents the dynamic interactions between the following signals: w_1 and z_1 denote the exogenous inputs and the controlled output vector, w_2 and z_2 denote the outputs and inputs of the model uncertainty, respectively.

The feedback system of Figure 2 is denoted as $\sum(s, x)$, and can be described as

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{4}$$

And the optimization-based controller synthesis problem considered in this paper can be stated as follows: given the objective function

$$f(x) = \inf_{K(jw,x)} \sup_{\omega \in R} \mu(\Sigma(j\omega)) \tag{5}$$

subject to

$$\lambda_{\max}\left(\sum(s, x) \right) < 0 \tag{6}$$

Let $\lambda_j[\Sigma(s, x)]$ denote the jth pole of the feedback system $\Sigma(s, x)$ and $\lambda_{\max}(\Sigma(s, x))$ is the pole with maximum real part:

$$\lambda_{\max}\left(\sum(s, x) \right) = \max\left\{ \mathrm{Re}\left(\lambda_j \left[\sum(s, x) \right] \right), \forall j \right\} \tag{7}$$

Then, a feedback system can be called robust stable as long as $\lambda_{\max}(\Sigma(s, x))$ is less than 0.

3. The PSO algorithm

In this section, the basic PSO algorithm proposed by Kennedy and Eberhart and our modifications based on PSO are described.

3.1. Basic PSO algorithm

Generally, an optimization problem is given by an objective function $f(x)$ to be optimized with respect to the design variable $x \in R^m$, as follows:

$$\min_{X\in F} f(x),\ F = \{x \in R^m\} \tag{8}$$

where the objective function $f : R^m \to R$, and the initial search space $D \subset R^m$, which is supposed to contain the desired design parameters $x_i (i = 1, 2, ..., m)$ given in advance.

The basic PSO algorithm uses a swarm consisting of n_p particles (i.e. $x^1, x^2, ..., x^{np}$), randomly distributed in the considered initial search space D, to obtain an optimal solution $x^* \in R^m$.

At every iteration k, the position of the ith particle and its velocity are denoted, respectively, as $\mathbf{x}_i^k := (x_{i,1}^k, x_{i,2}^k, ..., x_{i,m}^k)^T \in R^m$ and $\mathbf{v}_\mathbf{i}^k := (v_{i,1}^k, v_{i,2}^k, ..., v_{i,m}^k)^T \in R^m$ where $(i, k) \in [1, n_p] \times [1, k_{\max}]$. The values of $X_i^k$and V_i^k are modified based on the following update rules:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{9}$$

$$v_i^{k+1} = wv_i^k + c_1 r_{1,i}^k \left(x_i^{\text{best},k} - x_i^k\right) + c_2 r_{2,i}^k (x_{\text{swarm}}^{\text{best},k} - x_i^k) \tag{10}$$

where the inertia factor w represents the particle motion inertia, can be described as

$$w = w_{\max} - (w_{\max} - w_{\min}) \frac{k}{k_{\max}} \tag{11}$$

where $w_{\max}$ and $w_{\min}$ represent the maximum and minimum inertia factor values, respectively, $k_{\max}$ represents the maximum iteration.

The cognitive scaling factor c_1 and the social scaling factor c_2 given in advance, represent the influence of individual behaviour and group behaviour on the particle, respectively. $r_{1,i}^k$ and $r_{2,i}^k$ are randomly uniformly distributed in [0, 1] and represent the stochastic behaviours. In formula (10), $x_i^{\text{best},k}$ denotes the best previously obtained position of the ith particle and $x_{\text{swarm}}^{\text{best},k}$ denotes the best one in the entire swarm at the current iteration k.

Then, the PSO algorithm consists of the following steps:

Step [1]: Define all PSO algorithm parameters such as swarm size n_p, inertia factor w, cognitive c_1 and social scaling factors c_2, etc.

Step [2]: Randomly initialize each particle's position within the search space D and calculate the objective function . Set $k = 0$.

Step [3]: Determine $x_i^{\text{best},k}$and $x_{\text{swarm}}^{\text{best},k}$. If the termination criterion is satisfied, the algorithm terminates with the solution $x^* := x_{\text{swarm}}^{\text{best},k}$, otherwise, go to Step [4].

Step [4]: Apply formulas (9) and (10) to all particles and evaluate the corresponding objective functional values at each position. Increase the iteration number k and then go to Step [3].

3.2. Proposed PSO algorithm

It is crucial to handle the given constraint conditions in optimization-based controller design problems. Although there are a variety of effective approaches to process the above constraints, the following two are relatively representative. One augments the cost function with adaptive penalty functions, which is proposed by Cai Hairan [14]. Another augmented Lagrangian PSO (ALPSO) algorithm, which is developed by Sedlaczek and Eberhard. However, as mentioned in Section 1, these two methods have some shortcomings. To overcome these shortcomings, in this subsection, a novel way based on the PSO algorithm is introduced.

The optimization problem subject to multiple constraints can be formulated as

$$\min_{x\in F} f(x) \tag{12}$$

where $F = \{x \in R^m \mid h(x) < 0\},\ h(x) := [h_1(x), h_2(x), ..., h_n(x)]$

The function $h(x), (R^m \to R^n)$ denotes the constraints and F represents the allowing region. It is assumed that $\min_{x\in R^m} f_m(x)$ is not empty. In order to simplify multiple constraints, we change the restrictions by taking the minimum one, which requires no additional new parameters, and the new constraint can be described as

$$\min_{x\in R^m} f_m(x) \tag{13}$$

with

$$f_m(x) := \begin{cases} (\arctan(f(x)) + \dfrac{\pi}{2} + 1)^{h_{\max}(x)} & \text{if } h_{\max}(x) \geq 0 \\ \arctan(f(x)) - \dfrac{\pi}{2} & \text{otherwise} \end{cases} \tag{14}$$

where $f(x)$ is the original objective function, $h_{\max}(x) := \max[h_1(x), h_2(x), ..., h_n(x)]$ denotes the maximum violation. It overcomes the disadvantage of the PSO with adaptive penalty functions, which considers all constrains all time, and only part of constraints may not meet at some iterations. We can see that the PSO algorithm with adaptive penalty functions needs more time to calculate, but in fact it is less unnecessary. What is more, it is obviously that all the constraint conditions will be satisfied if the worst constraint has been satisfied, so it is convenient and reasonable to take maximum violation instead of all the constraints.

$\arctan(f(x))$ is adopted to ensure that $f_m(x)$ and $f(x)$ have the consistent characteristics of increase or decrease. What is more, when the particles are out of the feasible region (i.e. $h_{\max}(x) \geq 0$), we select an index form to integrate the objective function and constraint conditions, which enhances the ability of particle's global search ability, and improves the ability of finding optimal solutions when time tends to end. The aim of adding $\frac{\pi}{2}+1$ to $\arctan(f(x))$ is to guarantee that all particles can be put in the feasible region, which means if $h_{\max}(x_i{}^{k+1}) < h_{\max}(x_i{}^k)$, we can get $f_m(x_i{}^{k+1}) < f_m(x_i{}^k)$. When all of the particles are in the feasible region, namely $h_{\max}(x) < 0$, as all constrains meet the requirements, we only take the original objective function $f(x)$ into considers. $\frac{\pi}{2}$ is subtracted to give the new method the guarantee that each particle tries to return to the feasible region due to the following important feature: once x_i^kmoves into the feasible region, then $f_m(x_i^k) < 0$; hence, if x_i^k evolves according to (9) and (10) and leads $h_{\max}(x_i^{k+1}) > 0$, namely $f_m(x_i^{k+1}) < 0$. Because $f_m(x_i^k) < 0 < 1 \leq f_m(x_i^{k+1})$, $x_i^{\text{best},k+1}$and $x_{\text{swarm}}^{\text{best},k+1}$ remain within the feasible region, as a result, each particle also tends to return to the feasible region.

It is obvious that we can obtain a new solution of the constrained optimization problem by replacing the original $f(x)$ to $f_m(x)$. What is more, it overcomes the shortage of ALPSO. In the next section, we will test the performance of our proposed algorithm.

4. Application to test functions

Since the nonlinear optimization algorithms cannot be proved theoretically now, we can only compare the experimental results with other algorithms. In order to verify the performance of the proposed constrained PSO algorithm, we chose 12 single objective optimization problems with inequality constraints used in [19], which are widely employed in the constraint optimization problems. In addition to the two algorithms mentioned above, we also use a representative genetic algorithm called rough penalty genetic algorithm (RPGA) in [20] as a comparison.

All the four algorithms are implemented in Matlab7.8 and the simulations are performed on a computer with an Intel Core I5-4590 @ 3.3 GHz CPU and 4 GB of RAM in Windows 7 environment.

Each task runs 30 times, and the termination criterion is based on the maximum number of iteration which is chosen as $K_{\max} = 1000$. The maximum and the minimum inertia factor values are chosen as $W_{\max} = 0.9$, $W_{\min} = 0.4$, respectively, swarm size is fixed as 30. And the RPGA employs a stochastic universal selection and its crossover probability and mutation probability are 0.95 and 0.05, respectively.

Test results are shown in Table 1. We can see that the simple constrained PSO algorithm proposed in this paper has superior performance than other algorithms. What is more, the success ratio is significantly higher than the standard penalty function PSO algorithm, ALPSO algorithm and RPGA algorithm. Both the final optimization results and distribution characteristics are better than the other three algorithms.

The results of the execution are also shown in Figures 3 and 4. Because the Adaptive PSO is not applicable for G02 and ALPSO is the same for G09, functions G02 and G09 have three compared convergence curves, respectively. It is observed that the simple constrained PSO algorithm proposed in this paper performs better than other three algorithms in both G02 and G09. What is more, it was also noted that our proposed algorithm has a better converge speed. The computation time of the proposed algorithm is also presented, shown in Table 2.

5. Application to an air-to-air missile control

In this section, a sample missile longitudinal Raytheon control structure is introduced, we present the optimization results of three PSO versions and a RPGA version, comparisons between the four algorithms are provided. The final result also shows the superiority of our proposed algorithm.

A sample missile's longitudinal channel state space model is chosen, which works under the following states: 15,000 m altitude, 2.8 Mach and a 40 degree angle of attack. The state equation can be described as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0.091 & 0.042 & 9.760 & 12.109 \\ 0.203 & 0.588 & 92.06 & 0 \\ 0.041 & 0.746 & 98.72 & 0 \\ 0.034 & 0.112 & 1.314 & 0 \\ 0 & 0 & 0.121 & 2.513 \end{bmatrix} \cdot \begin{bmatrix} u \\ w \\ q \\ \alpha \end{bmatrix} + \begin{bmatrix} 0.299 \\ 2.431 \\ 4.201 \\ 0 \end{bmatrix} \cdot \delta_e$$

The output equation can be formulated as

$$\begin{bmatrix} A_z \\ q \end{bmatrix} = \begin{bmatrix} 0.384 & -1.276 & 1.033 & 0 \\ 0 & 0 & 57.295 & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ w \\ q \\ \alpha \end{bmatrix} + \begin{bmatrix} -2.487 \\ 0 \end{bmatrix} \cdot \delta_e$$

where u 、 w are the velocity components along the body axes x_b, z_b, respectively, α is angle of attack, q is pitch rate, A_z is normal overload, δ_e is equivalent elevator angle.

Table 1. Test results of four kinds of constrained optimization algorithms.

No.	Dimension	Known extremum	Test algorithm	Success ratio (%)	Optimal value	Worst value	Mean value	Variance
G01	13	−15.00	Adaptive PSO	100	−15	−15	−15	0
			ALPSO	16	−14.933	−14.878	−14.89552	0.133243
			RPGA	100	−15	−15	−15	0
			Novel PSO	100	−15	−15	−15	0
G02	20	−0.804	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	96	−0.75214	−0.71495	−0.7397	0.000991
			RPGA	100	−0.77384	−0.71895	−0.75185	0.003045
			Novel PSO	100	−0.7851	−0.71678	−0.78327	0.001975
G04	5	−30,666	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	0	N/A	N/A	N/A	N/A
			Novel PSO	18	−28,337	−26,545	−27,510	0.016639
G06	2	−6962	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	0	N/A	N/A	N/A	N/A
			Novel PSO	63	−5734.17	−1966.06	−3500.69	1109.2
G07	10	24.31	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	11	24.2226	1713.201	1012.656	72.4
			RPGA	15	24.3335	1921.731	1129.835	70.8
			Novel PSO	62	24.4495	1517.915	817.776	38.6
G08	2	−0.096	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	0	N/A	N/A	N/A	N/A
			Novel PSO	19	−0.09571	−0.0785	−0.0898	0.35
G09	7	680.6	Adaptive PSO	100	680.6	680.6	680.6	0
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	100	680.6	680.6	680.6	0
			Novel PSO	100	680.6	680.6	680.6	0
G10	8	7049	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	0	N/A	N/A	N/A	N/A
			Novel PSO	72	8408.19	26738.14	21824.9	71045
G12	3	−1.000	Adaptive PSO	100	−1	−1	−1	0
			ALPSO	100	−1	−1	−1	0
			RPGA	100	−1	−1	−1	0
			Novel PSO	100	−1	−1	−1	0
G18	9	−0.866	Adaptive PSO	70	0	0	0	0
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	50	0	0	0	0
			Novel PSO	50	−0.838	0	−0.2	0.0278
G19	15	32.66	Adaptive PSO	0	N/A	N/A	N/A	N/A
			ALPSO	0	N/A	N/A	N/A	N/A
			RPGA	0	N/A	N/A	N/A	N/A
			Novel PSO	12	32.68	137.83	38.95	2.2744
G24	2	−5.508	Adaptive PSO	100	−5. 508	−5. 508	−5. 508	0
			ALPSO	100	−5. 508	−5. 508	−5. 508	0
			RPGA	100	−5.508	−5. 508	−5. 508	0
			Novel PSO	100	−5. 508	−5. 508	−5. 508	0

Table 2. Average computation time (G01–G24).

Function	G01	G02	G04	G06	G07	G08	G09	G10	G12	G18	G19	G24
Times (ms)	**3.80**	**4.18**	**2.14**	**1.16**	**2.44**	**1.62**	**2.43**	**2.80**	**1.40**	**2.59**	**2.80**	**1.14**

The bold numbers indicate the time for each iteration.

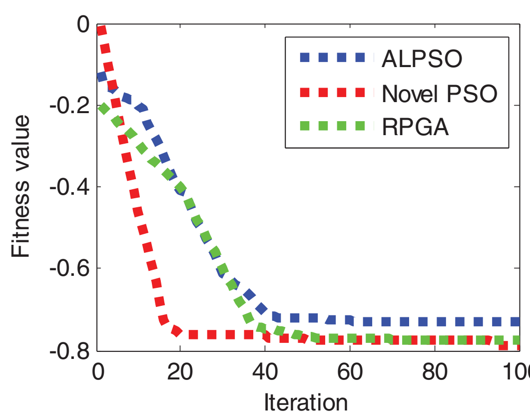

Figure 3. Convergence curves of ALPSO and Novel PSO and RPGA for function G02.

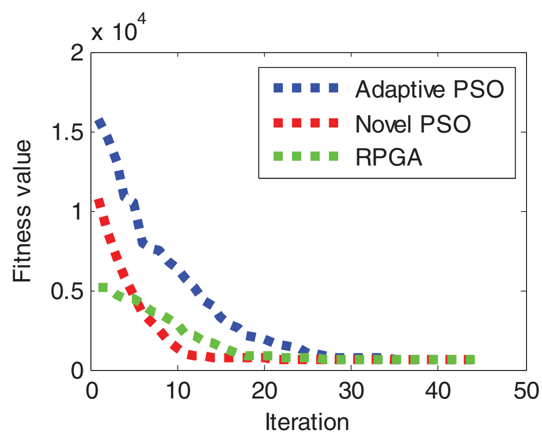

Figure 4. Convergence curves of Adaptive PSO and Novel PSO and RPGA for function G09.

Table 3. Longitudinal channel step response performance index.

Performance	tr	σ%	tp	ts	ess
Index	0.3s	15%	0.7s	1s	5%

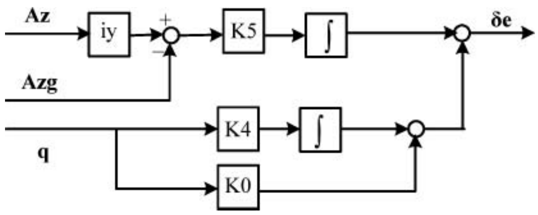

Figure 5. Sample missile longitudinal Raytheon control structure.

In order to ensure the generality, this paper selects the classic Raytheon controller as the sample missile control structure, and its longitudinal control structure is shown in Figure 5.

The time domain performance index of the sample missile longitudinal channel is shown in Table 3.

We can see that the missile fixed-structure μ controller design problem is much more complex than the test functions. However, with our proposed algorithm, the parameters can be simply taken as

- Swarm size: 20
- Swarm dimension: 4
- Maximum swarm speed: $V_{\max,d} = x_{\max,d}/2 \quad i = 1...4$,
- Learning factors: $c_1 = 2, c_2 = 2$
- The maximum and minimum of inertia coefficient: $W_{\max} = 0.9, W_{\min} = 0.4$

Also for RPGA, the parameters can be taken as

- Popsize: 20
- Individual dimension: 4
- Crossover probability: 0.9
- Mutation probability: 0.05

Each task runs 50 times and the constraints are listed in table (2). What is more, we introduce the parameter "μ" as a robust performance constraint. The optimization results based on four different algorithms are shown in Tables 4–6.

As shown in Tables 4–7, the μ value obtained by PSO algorithm with adaptive penalty function is the biggest, and the minimum μ value is greater than 1, which shows that the controller has poorest robustness. The ALPSO algorithm's overall performance takes second place. The RPGA is about the same with ALPSO. Simple constraint PSO algorithm presented in this paper has the best performance and the traditional amplitude and phase margin index. The convergence curves of objective function based on the three PSO algorithms are shown in Figures 6–9.

From Figures 6–9, we can see that simple constraint PSO algorithm proposed in this paper has the best statistical convergence characteristics. And that we obtain the best optimum result: iy = 0.97, K5 = 0.2, K4 = 2.44,

Table 4. Statistical analysis of optimization results: PSO with adaptive penalty functions-design case.

	K5	K4	K0	iy	Pm	Gm	μ
Maximum	0.426139	5.446269	0.804308	0.974	52.14646	20.37846	3.376226
Minimum	0.197782	2.659609	0.401655	0.974	41.91909	14.36969	1.079223
Average	0.307782	4.017192	0.618247	0.974	49.52932	16.83758	1.59549
Variance	0.003391	0.524464	0.012105	0	4.990968	2.622232	0.143544

Table 5. Statistical analysis of optimization results: ALPSO-design case.

	K5	K4	K0	iy	Pm	Gm	μ
Maximum	0.45335	5.14782	0.957044	0.974	55.71681	20.39364	2.637763
Minimum	0.200236	2.402963	0.406267	0.974	45.92925	12.81485	0.988533
Average	0.2816	3.280196	0.587651	0.974	51.61155	17.45787	1.194117
Variance	0.005556	0.649556	0.027371	0	6.555494	5.296641	0.118109

Table 6. Statistical analysis of optimization results: RPGA-design case.

	K5	K4	K0	iy	Pm	Gm	μ
Maximum	0.432745	5.305936	0.924359	0.974	55.79524	20.42399	2.898844
Minimum	0.216643	2.372155	0.408335	0.974	42.09922	13.08642	0.951545
Average	0.273813	3.643545	0.617537	0.974	50.89723	17.65566	1.236918
Variance	0.003996	0.452355	0.021436	0	6.192434	5.036654	0.107466

Table 7. Statistical analysis of optimization results: the proposed PSO-design case.

	K5	K4	K0	iy	Pm	Gm	μ
Maximum	0.410577	4.667844	0.879264	0.974	55.88111	20.47188	2.517808
Minimum	0.203696	2.433613	0.41349	0.974	46.29897	14.58738	0.890642
Average	0.266061	3.104618	0.549285	0.974	51.98557	17.93639	1.065869
Variance	0.002964	0.337044	0.013827	0	6.107922	3.040519	0.070271

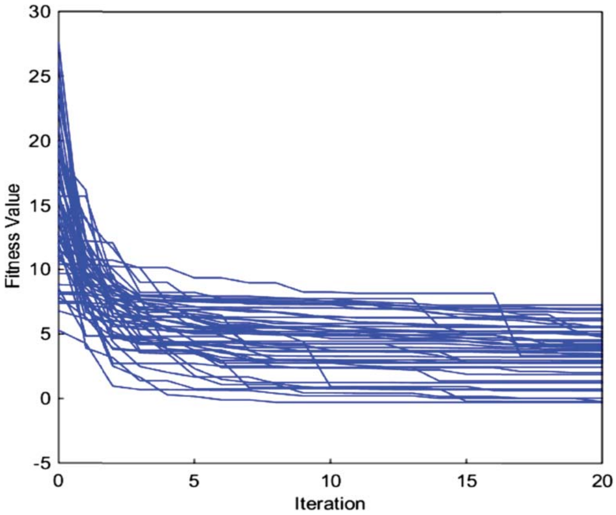

Figure 6. Objective function convergence curve – PSO with adaptive penalty function.

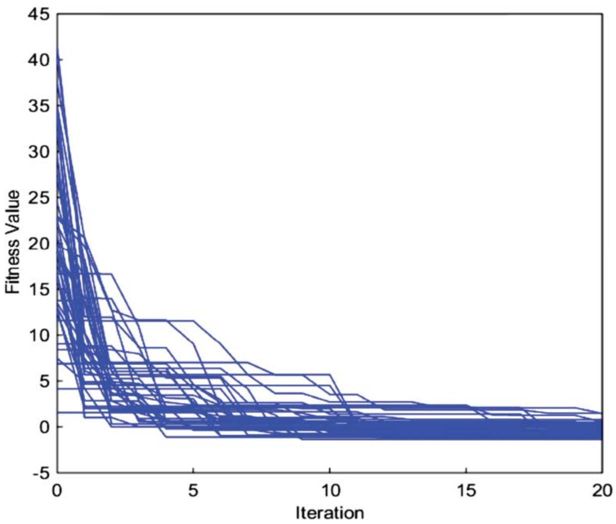

Figure 7. Objective function convergence curve – ALPSO.

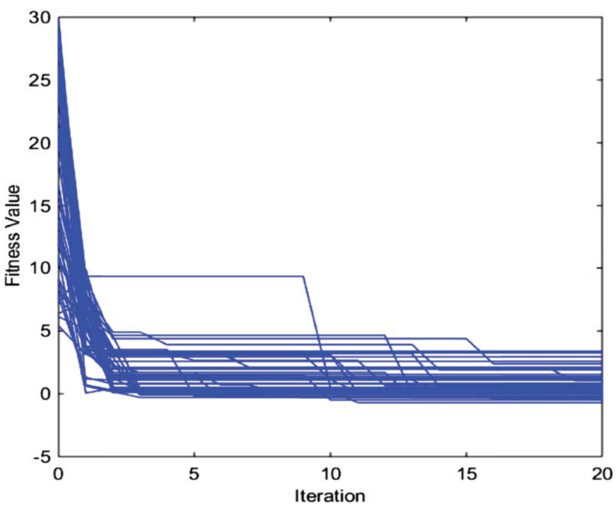

Figure 8. Objective function convergence curve – RPGA.

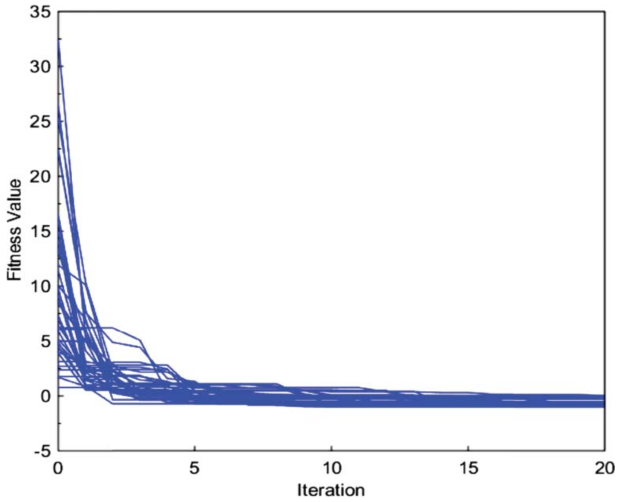

Figure 9. Objective function convergence curve – PSO presented in this paper.

K0 = 0.41, which is shown in Table 6. Moreover, we present a closed-loop system robust simulation verification structure, which is shown in Figure 10. The model uncertainty function and the noise interference are taken as: win: $\mathbf{2}\cdot(s+3.2)/(s+160)$, acceleration meter noise: 1%; angular rate gyro noise: 0.1%, respectively, which can be seen in reference [15].

The frequency domain and time domain performance of the closed-loop control system are analysed as follows:

(1) Performance analysis in frequency domain
(a) Robust stability analysis

As shown in Figure 10, if we only consider the response of Z1, we can see that the upper bound of the structure singular value of the transfer function from D1 to Z1 is always below 0.3, which is shown in Figure 11, and shows its good robust stability.

(b) Robust performance analysis

The performance of the system under the action of the external input and the multiplicative uncertainties is investigated. As shown in Figure 11, the structure singular value is also below 1, which indicates that the robust performance of the system also meets the robust performance requirements.

(2) Performance analysis in time domain
(a) Unit step response of nominal system

Figure 12 shows the step response of the nominal system. We can see that the simulation results meet the time domain performance requirements shown in Table 3.

(b) Unit step response of perturbed system

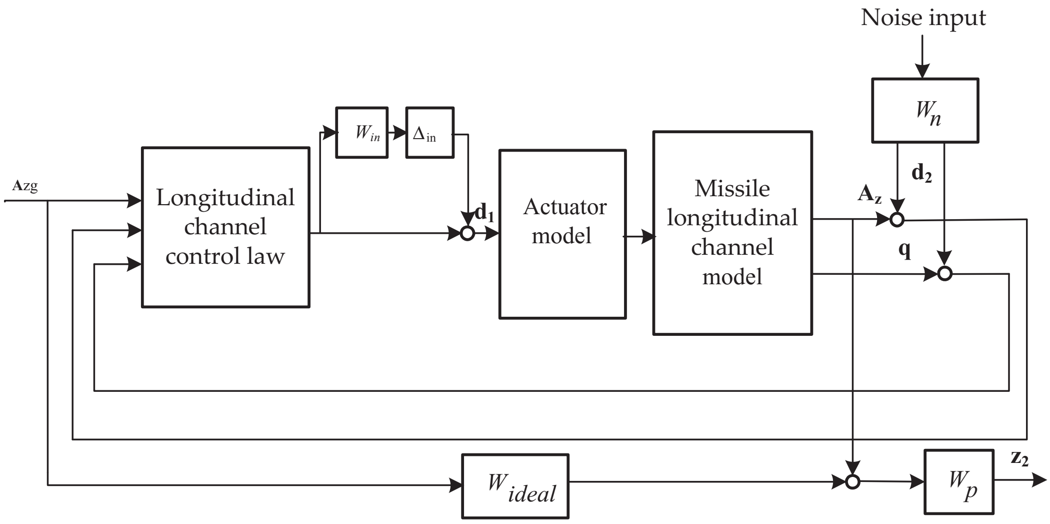

Figure 10. Closed-loop system robust simulation verification structure.

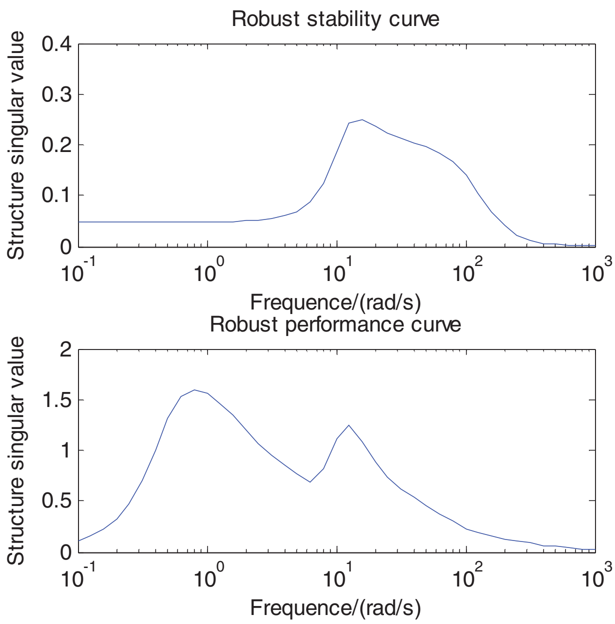

Figure 11. Robust curve of the system.

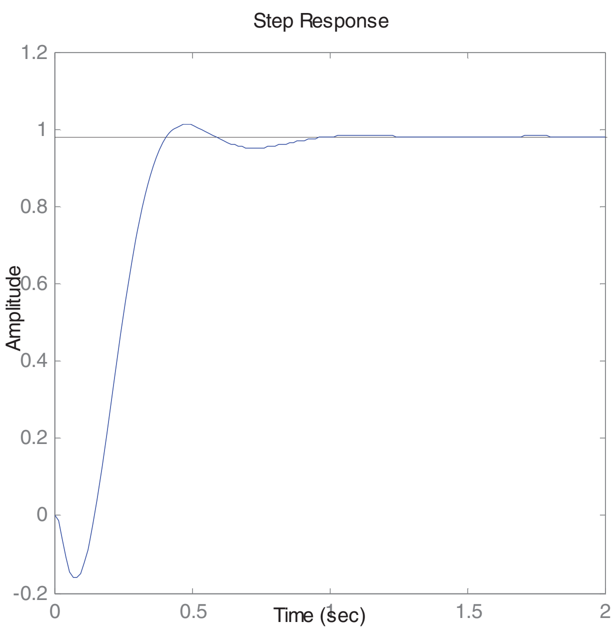

Figure 12. The nominal performance response curve.

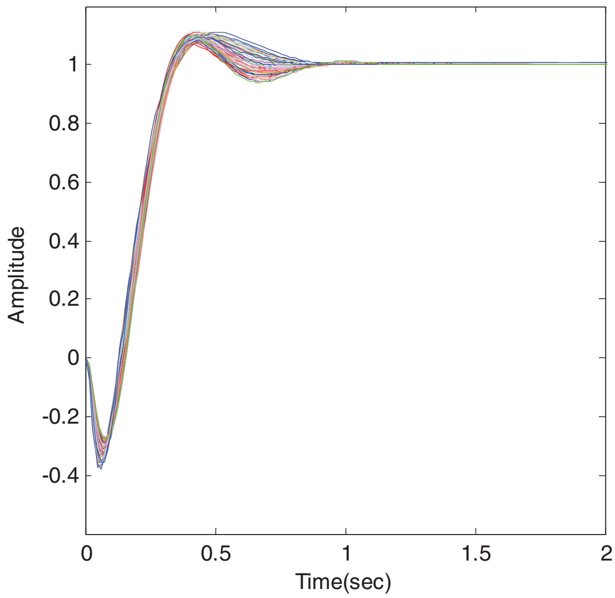

Figure 13. The perturbation system performance curve.

The step response under the external input and the uncertainty is shown in Figure 13. It shows that the perturbation system also has a good time domain performance.

6. Conclusion

In this paper, we have proposed a new algorithm based on PSO (denoted as simple constraint PSO) and apply it to fixed-structure μ controller design. A new constraint strategy is introduced and it solves the shortcomings of traditional methods. What is more, we test our proposed algorithm on a set of 12 benchmark functions widely used and a comparison is made with other algorithms. The simulation results show that our proposed algorithm has an absolute advantage. Second, we apply our algorithm to the missile fixed-structure μ controller design progress, and also get the best results than other algorithms. At last, we provide an analysis of the robustness of the system, and show its good robust performance.

Disclosure statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding

Aeronautical Science Foundation of China [grant number 20160152001].

References

[1] Zhou K, Doyle JC, Glover K. Robust and optimal control. London: Springer; 2014.

[2] Skogestad S, Postlethwaite I. Multivariable feedback control: Analysis and design. New York (NY): Wiley; 1996.

[3] Doyle JC. Analysis of feedback systems with structured uncertainties. IEEE Proc D. 1982;129(11):242–250.

[4] Doyle JC. A review of μ for case studies in robust control. IFAC Proc. 1987;20(5):365–372.

[5] Tuanjie L, Yujuan M. Robust vibration control flexible tensegrity structure via μ synthesis. Struct Control Health Monit. 2013;20(2):173–186.

[6] Bennis N, Ramzi M, Haloua M. LMI solutions for $H2H_\infty$ decentralized controllers applied to an aerothermic process. J Control Theory Appl. 2013;11(2):247–254.

[7] Roushandel A, Khosravi A, Alfi A. Bilateral control of teleoperation systems via robust PID controllers based on LMI. In: International Conference on Control, Instrumentation, and Automation IEEE; 2014. p. 16–21.

[8] Gonhua W, Dishan Q. Superior solution guided particle swarm optimization com bined with local search techniques. Expert Syst Appl. 2014;41:7536–7548.

[9] Hwang C, Hsiao CY. Solution of a non-convex optimization arising in PI/PID control design. Automatica. 2002;38(11):1895–1904.

[10] Ho M. Synthesis of H_∞-PID controllers: a parametric approach. Automatica. 2003;39(6):1069–1075.

[11] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks; 1995; Piscataway, NJ; 2002. p. 1942–1948.

[12] Michalewicz Z, Dasgupta D, Riche R, et al. Evolutionary algorithms for constrained engineering problems. Comput Ind Eng J. 1996;30:851–870.

[13] Bouallègue S, Haggège J, Benrejeb M. Particle swarm optimization-based fixed-structure H Control design. Int J Control Autom Syst. 2011;9(2):258–266.

[14] Hailuan C. A new adaptive penalty function in the application of genetic algorithm. J East China Normal Univ (Nat Sci). 2015;6(6):36–52.

[15] Sedlaczek K, Eberhard P. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. Struct Multidiscip Optim. 2006;32(4):27–286.

[16] Kim TH, Maruta I, Sugie T. Robust PID controller tuning based on the constrained particle swarm optimization. Automatica. 2008;44(4):1104–1110.

[17] Maruta I, Kim TH, Sugie T. Fixed-structure H controller synthesis: a meta-heuristic approach using simple constrained particle swarm optimization. Automatica. 2009;45(3):553–559.

[18] Doyle JC, Chu CC. Matrix interpolation and $H\infty$ Performance bounds. In: Proceedings of the American Control Conference; Jun; Boston, MA; 1985. p. 129–134.

[19] Liang J, Runarsson T, Mezura-Montes E, et al. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter. 2006.

[20] Lin C-H. A rough penalty genetic algorithm for constrained optimization. Inform Sci. 2013;241:119–137.