Proceedings of the 7th IFAC Symposium on Robust Control Design
The International Federation of Automatic Control
Aalborg, Denmark, June 20-22, 2012

A New Approach to Design a Simple Structure μ Synthesis Controller: Application to an Uncertain DC Motor

A. Lari. A. Khosravi.
R. Karamnezhad

Electrical Engineering Department, Noshirvani University of Technology, Babol, Iran
(e-mails: a.lari@stu.nit.ac.ir, akhosravi@nit.ac.ir, r.karamnezhad@stu.nit.ac.ir)

Abstract: This paper proposes an evolutionary approach to solve μ synthesis problem. The goal is to achieve simple structure μ synthesis controllers without any order reduction. In the proposed approach, an evolutionary algorithm is used to solve the problem as a constraint optimization problem in which robust stability and robust performance based on μ analysis are the constraint and the cost function respectively. The evolutionary algorithm tries to find the coefficients of a structure-specified controller with the minimum cost value. An improved particle swarm optimization (PSO) algorithm is employed (as an evolutionary algorithm) to solve the constraint optimization problem. Effectiveness of the proposed approach is evaluated by an uncertain DC motor system. Results of simulation demonstrate the advantages of the proposed controller in terms of simple structure and robustness against plant perturbations in comparison with the conventional solution to μ synthesis problem (D-K iteration method).

1. INTRODUCTION

Designing controllers using a structure-specified approach are obtaining increasing interest due to practical issues. In the implementation of controllers, high order controllers lead to high cost, practical difficulties and poor reliability. Robust controller design techniques usually yield higher-order controllers in comparison with other areas. Although, model order reduction methods are used to reduce order of the resulted controllers, but the reduced order controllers which have a close behaviour to the main controllers are usually not yet practical enough. Accordingly, robust control is one of the most considered areas, in which designing controllers using a structure-specified approach has been further investigated. In the last decade, designing structure-specified robust controllers using evolutionary algorithms has attracted considerable attention to solve the problem (Olranthichachatand and Kaitwanidvilai, 2008; Ho et al., 2005; Zamani et al., 2009; Chen and Cheng, 2005; Kaitwanidvilai and Parnichkun, 2008). In this approach, coefficients of a low order structure-specified controller are calculated by an evolutionary algorithm. These coefficients are determined such that the obtained controller satisfies the robust design objectives.

MU (μ) synthesis problem is one of the most powerful robust controller design techniques. D-K iteration is the most common method to solve this problem. One of the main drawbacks of D-K iteration is that it does not guarantee convergence to a global or even local minimum, which leads to non-optimality of the resulting controller (Stein and Doyle, 1991). On the other hand, the resulted controllers by this method are usually impractical, high order controllers. To overcome this problem, in this paper, designing structure-specified controllers using an evolutionary algorithm is proposed to solve the μ synthesis problem. In this approach, μ synthesis problem is solved as a constraint optimization problem in which the μ analysis is used to examine the robust stability and performance features of the designed controller; robust stability as the constraint and robust performance as the cost function which should be minimized. In this paper, an improved particle swarm optimization (PSO) algorithm is used as a simple evolutionary algorithm to solve the problem.

PSO algorithm is a population-based optimization technique, inspired by social behaviour of bird flocking and fish schooling. It has been widely applied in many areas to solve optimization problems because of the ease of implementation and fast convergence speed (Franken and Engelbrecht, 2005; Ho et al., 2008).

An uncertain DC motor introduced in (Balas et al., 1998) is considered to evaluate efficiency of the proposed controller in comparison with D-K iteration controller.

The rest of this paper is organized as follows: Section 2 discusses the basic μ synthesis problem. In section 3, particle swarm optimization is illustrated and in section 4 the proposed structure-specified μ synthesis controller is stated. Description of the uncertain DC motor and controller design is given in section 5. Simulation results are presented in section 6 and finally section 7 is assigned to conclusion.

2. PRELIMINARY

In this section, we describe a brief introduction to μ synthesis Problem.

2.1 μ Synthesis Problem

The definition of structure singular value, μ, depends on the underlying block structure of the perturbations which is defined as follows. Perturbation matrix $\mathbf{\Delta} \subset \mathbb{C}^{n \times n}$ is defined as (1).

$$\mathbf{\Delta} = \left\{ \begin{array}{l} diag\left[\delta_1 I_{r1}, ..., \delta_s I_{rs}, \Delta_1, ..., \Delta_F\right]: \\ \qquad\qquad \delta_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j \times m_j} \end{array} \right\} \tag{1}$$

The block diagonal matrix $\mathbf{\Delta}$, includes two types of blocks: repeated scalar block (δ_i) and full block (Δ_j). In (1) S represents the number of repeated scalar blocks and F represents the number of full blocks. For consistency among all the dimensions, (2) should be held.

$$\sum_{i=1}^{S} r_i + \sum_{j=1}^{F} m_j = n \tag{2}$$

Structure singular value, $\mu_\Delta(M)$, of a Matrix $M \in \mathbb{C}^{n \times n}$ with respect to a block structure $\Delta \in \mathbf{\Delta}$ is defined as (3).

$$\mu_\Delta(M) := \frac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \mathbf{\Delta}, \det(I - M\Delta) = 0\}} \tag{3}$$

unless no $\Delta \in \mathbf{\Delta}$ makes $I - M\Delta$ singular, in which case $\mu_\Delta(M) := 0$.

Unfortunately, (3) is not suitable for computing μ since the implied optimization problem may have multiple local minima (Zhou and Doyle, 1998). However, upper and lower bound for μ may be effectively computed as (4).

$$\max_{U \in \mathcal{U}} \rho(MU) \le \mu_\Delta(M) \le \inf_{D \in \mathbf{D}} \bar{\sigma}(DMD^{-1}) \tag{4}$$

where $\mathcal{U}$ and D are introduced in (5) and (6).

$$\mathcal{U} = \left\{U \in \mathbf{\Delta} : UU^* = I_n\right\} \tag{5}$$

$$\mathbf{D} = \left\{ \begin{array}{l} diag\left[D_1, ..., D_S, d_1 I_{m_1}, ..., d_{F-1} I_{m_{F-1}}, I_{m_F}\right]: \\ D_i \in \mathbb{C}^{r_i \times r_i}, D_i = D_i^* > 0, d_j \in \mathbb{R}, d_j > 0 \end{array} \right\} \tag{6}$$

2.2 Robust Stability and Robust Performance based on μ Analysis

In this subsection it is described how robust stability and performance are measured based on μ analysis (Zhou and Doyle, 1998). Any interconnected system can be rearranged to fit the general framework in Figure 1. In this figure, K is the controller, P is the nominal plant and Δ is the uncertainty set (perturbation matrix). Moreover, w denotes the exogenous input typically including command signals, disturbances, noises, etc.; z denotes the error output usually consisting of regulator output, tracking errors, filtered actuator signals, etc; v and d are the input and output signals of the dynamic uncertainties, respectively. Finally, u and v are the control input and the measurement signal, respectively.

The general framework in figure 1 can reduce to figure 2. In figure 2, $M(P,K) = F_l(P,K)$ and $F_l(P,K)$ is lower linear fractional transformation (lower LFT) of P and K.

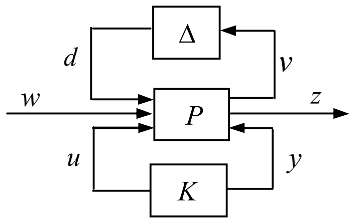

Fig. 1. General framework

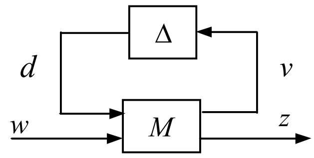

Fig. 2. Analysis framework

Let M be partitioned accordingly as (7). In this partitioning M_{11} is used for robust stability analysis and M is used for robust performance analysis.

$$\begin{bmatrix} v \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}}_{M} \begin{bmatrix} d \\ w \end{bmatrix} \tag{7}$$

The structure is used for robust stability analysis (based on μ) has been demonstrated in figure 3. When M_{11} is an interconnected transfer function $M_{11}(s)$ formed with respect to the uncertainty set Δ, the structure singular value of $M_{11}(s)$ indicates the robust stability of the perturbed system. Without loss of generality, assume that the uncertainties have been normalized. The standard configuration in figure 3 is robust stable if $M_{11}(s)$ is stable and $\mu_\Delta(M_{11}(s)) < 1$ (or $\|M_{11}\|_\mu < 1$).

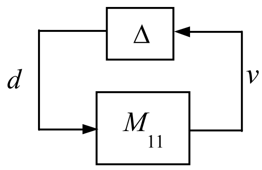

Fig. 3. Structure for robust stability analysis

Figure 4 is a standard configuration for robust performance analysis. System performance specification can usually be interpreted as reduction of z with respect to w. The robust performance problem based on figure 4, is equivalent to robust stability problem in figure 3, if the uncertainty set Δ replace with $\tilde{\Delta}$ in (8).

$$\begin{array}{c} \tilde{\Delta} \in \tilde{\mathbf{\Delta}} := \left\{diag\left\{\Delta, \Delta_P\right\} : \Delta \in B\mathbf{\Delta}, \left\|\Delta_P\right\|_\infty \le 1\right\} \\ B\mathbf{\Delta} = \left\{\Delta \in \mathbf{\Delta}: \bar{\sigma}(\Delta) \le 1\right\} \end{array} \tag{8}$$

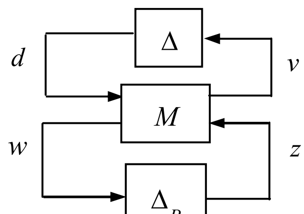

Fig. 4. Structure for robust performance analysis

With normalized uncertainties, structure of figure 4 is robustly stable against uncertainty set $\tilde{\Delta}$ (robust performance against uncertainty set Δ) if $\mu_{\tilde{\Delta}}(M(s))<1$ (or $\|M\|_{\mu}<1$).

Designing controllers based on the mentioned robust stability and performance measurements is called μ synthesis problem. D-K iteration method is the conventional solution to μ synthesis problem.

3. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (Kennedy et al., 1995) is based on the social behavior of collection of animal such as birds flocking. In PSO algorithm each individual of the swarm, be called particle, remembers the best solution found by itself and by the whole swarm along the search trajectory. The particles move along the search space and exchange information with other particle according to the following.

$$V_{id}=wV_{id}+c_1r_1\left(P_{id}-X_{id}\right)+c_2r_2\left(P_{gd}-X_{id}\right) \tag{9}$$

$$X_{id}=X_{id}+V_{id},\ \ d=1,2,\ldots,N\ i=1,2,\ldots,S \tag{10}$$

where X_{id} and V_{id} represent the current position and velocity of the particles, respectively, P_{id} is the best individual particle position, P_{gd} denotes the best swarm position, the parameters c_1 and c_2 are cognitive and social parameters, respectively; the parameters r_1 and r_2 are random numbers between 0 and 1. Finally, w is the inertia weight to balance the global and local search abilities. A large inertia weight facilitates global search while a small inertia weight facilitates local search. In an empirical study on PSO (Shi and Eberhart, 1998) it was claimed that a linearly decreasing inertia weight could improve local search towards the end of a run, rather than using a constant value throughout. A decreasing function for the dynamic inertia weight can be devised as (11).

$$w=\left(iter_{\max}-iter_{cur}\right)\cdot\left(\frac{w_{initial}-w_{final}}{iter_{\max}}\right)+w_{final} \tag{11}$$

where $w_{initial}$ and w_{final} represent the initial and final inertia weights at the start of a given run, respectively; $iter_{max}$ is the maximum number of iterations in a offered run, and $iter_{cur}$ denotes the current iteration number at the present time step.

However, due to the utilization of a linearly decreasing inertia weight, the global search ability at the end of the run may be inadequate. The PSO may fail to find the required optimal in cases when the problem is too complicated. But to some extent, this can be overcome by employing a self-adapting strategy for adjusting the acceleration coefficients. An optimizing method (Suganthan, 1999) applied that make the two factors decrease linearly with the increase of iteration numbers, but the results were not as good as the fixed value 2 of c_1 and c_2. An investigation (Ratnaweera et al., 2004) proved that the convergence of particles to the global optima improve based on the way that make c_1 decrease and c_2 increase linearly with the increase of iteration numbers, the parameters c_1 and c_2 are given in (12).

$$\begin{aligned}c_1&=c_{1s}+iter_{cur}(c_{1e}-c_{1s})/iter_{\max}\\ c_2&=c_{2s}+iter_{cur}(c_{2e}-c_{2s})/iter_{\max}\end{aligned} \tag{12}$$

where c_{1s} and c_{2s} represent the initial values of c_1 and c_2, respectively; and c_{1e} and c_{2e} show the final values of c_1 and c_2, respectively.

4. STRUCTURE-SPECIFIED μ SYNTHESIS CONTROLLER

In this section, it is described how the new approach solves μ synthesis problem. In the core of proposed approach, the PSO algorithm described in section 3 is employed to solve a constraint optimization problem as a different approach to the μ synthesis problem. Based on the structure introduced in section 2, the optimization problem has been given in (13).

$$\min_{K(s)}\ \|M(s)\|_{\mu}\ ;\quad such\ that:\ \|M_{11}(s)\|_{\mu}<1 \tag{13}$$

where M and M_{11} are defined in (7), K(s) is the controller, the constraint is the robust stability based on μ analysis whereas the cost function is the robust performance based on μ analysis. Similar to the D-K iteration method, the upper limit of μ is considered instead of real value of μ. In the proposed method, first a structure-specified controller (normally in form of transfer function) should be determined. Then, coefficients of a structure-specified controller are specified using the PSO algorithm such that the cost function given in (13) is minimized. Generally, the procedure of proposed approach is summarized as follows:

Step 1 Determine a proper structure for the controller.

Step 2 Set the parameters of PSO algorithm.

Step 3 Initialize a group of particles in an M-dimensional space as random points, where M denotes the number of coefficients of controller. All of initial particles must verify the robust stability constraint.

Step 4 Calculate P_{id} , P_{gd} , and the cost function for each particle.

Step 5 Update c_1, c_2 and w according to the iteration number given in (11) and (12).

Step 6 Update the velocities and positions of each particle according to (9) and (10), respectively.

Step 7 If each new position verifies the stability constraint, set it as the new position. Else, set the last position as the new position.

Step 8 If the number of iterations is lower than a predefined value, go to step 4. Else, go to step 9.

Step 9 Consider P_{gd} as the coefficients of desired controller. If the obtained controller is not efficient enough, order of the structure-specified controller can be incremented.

In addition to give a robust stable controller, using the evolutionary approach to solve μ synthesis problem makes it possible to apply the desired objectives in time domain for designing robust controllers (especially about nominal system). It can be done by defining a weighted cost function in (13) based on desired objectives in time and frequency domains. These are the main advantages of the proposed approach in comparison with the D-K iteration method.

5. SYSTEM DESCRIPTION AND CONTROLLER DESIGHN

5.1 Uncertain DC motor

The nominal model of the DC motor is defined by the resistance R, the inductance L, the emf constant Kb, armature constant Km, the linear approximation of viscous friction Kf and the inertial load J. Each of these components varies within a specific range of values (Balas et al., 1998). The resistance and inductance constants range within +/- 40% of their nominal values 2 and 0.5, respectively. The values of Kf and Kb are the same and are correlated. They have a nominal value of 0.015 and range between 0.012 and 0.019. Viscous friction, Kf, has a nominal value of 0.2 with a 50% variation in its value. The inertia J is also uncertain. It is modeled as a constant 0.02, which is a simplistic model that neglects dynamics present in the real system. We assume that the maximum I/O gain of these dynamics does not exceed 10% of J's nominal value 0.02. These unmodeled dynamics are included in the problem. The uncertain state-space matrices A, B and known matrices C, D are constructed from the motor parameters as (14):

$$A=\begin{bmatrix} -\frac{R}{L} & \frac{kb}{L} \\ \frac{km}{J} & -\frac{kf}{J} \end{bmatrix},\ B=\begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix},\ C=\begin{bmatrix} 0 & 1 \end{bmatrix},\ D=0 \tag{14}$$

5.2 Design requirements

Figure 5 demonstrates the control scheme which is employed to design a robust controller using the proposed approach. Coefficient weights of W_p and W_u are responsible for tracking of reference signal and preserving the control law in a proper bound, respectively and are shown in (15). Also, in this figure P and C are nominal plant and controller, respectively.

$$W_P = 0.25\frac{s^2+7s+15}{s^2+8s+.01} \quad ; \quad W_u = 0.01 \tag{15}$$

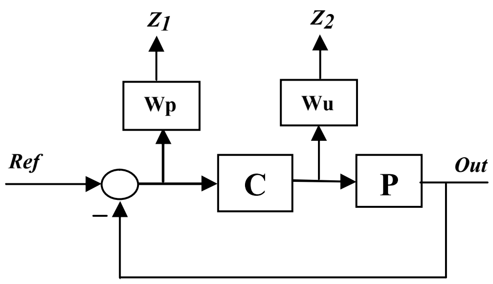

Fig. 5. Control scheme

For designing the proposed structure-specified controller, a structure as PID with first-order derivative filters is chosen. This structure is expressed in (16). K_p, K_i , K_d and τ_d are parameters to be evaluated. This controller has proposed (Kaitwanidvilai and Parnichkun, 2008) for designing a robust controller.

$$K(p)=(K_p+\frac{K_i}{s}+\frac{K_d s}{\tau_d s+1}) \tag{16}$$

Table 1 contains the values of PSO parameters. In addition, in the proposed approach, coefficients of (16) are considered in a range from 0 to 10 to find the desired values.

Table 1. Parameters of PSO algorithm

Parameters	Value	Parameters	Value
c_{1e}	0.5	Number of Particles	25
c_{1s}	1.5	Number of Iterations	100
c_{2e}	1.5	$w_{initial}$	1.2
c_{2s}	0.5	w_{final}	0.4

6. SIMULATION RESULTS

In order to evaluate the proposed approach, the uncertain DC motor, is considered to simulation. Using the coefficient weights the D-K iteration method leads to a controller of order 12. This controller has been reduced using Hankle norm approximation. Among reduced order controllers, the 3rd order controller which is shown in (17), is smallest controller in term of order which can keep robust stability and performance measurements with respect to the main D-K iteration controller (Peak value of μ performance plot for the 2nd order controller is about 1.73 which is not desired). These controllers are too similar to see any clear difference, as it is shown in figures 6 and 7. In spite of being a low order controller, the computed 3rd order controller is relatively complex.

$$C_{D-K}=\frac{7.933e004s^2+7.333e005s+1.342e006}{s^3+870.5s^2+2.619e004s+32.79} \tag{17}$$

By the proposed approach, a controller as (18) is obtained. It is clear that, this low order controller is more practical in comparison with the reduced order controller shown in (17).

$$K(p)=(23.5+\frac{81.4}{s}+\frac{75.7s}{51.4s+1}) \tag{18}$$

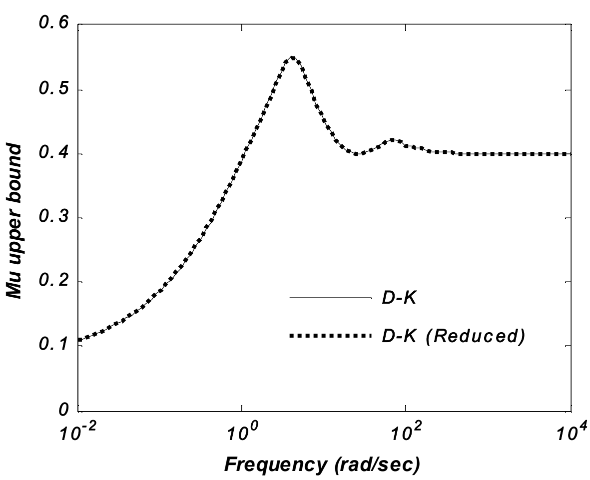

Fig. 6. Comparison of robust stability

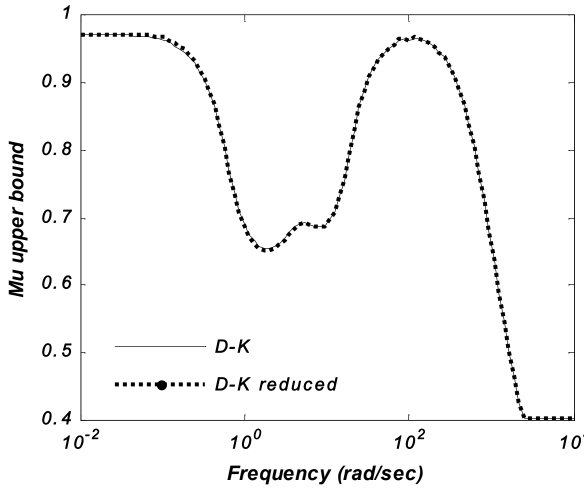

Fig. 7. Comparison of robust performance

In the next, robust stability and performance of the proposed structure-specified controller in (18) is compared with that of D-K iteration controller in figures 8 and 9. In figure 8, peak value of μ stability plot for D-K iteration controller is 0.550 and for the structure-specified controller is 0.702. This means that both of controllers have a proper stability margins, although D-K iteration controller is more conservative. According to figure 9, peak value of μ performance plot for D-K iteration controller is 0.97 and for the proposed controller is 0.94 and in most of frequency ranges structure-specified controller performs better and it can be equal to better time responses which have been investigated by figures 10 and 11. Figure 10 shows that the structure-specified controller performs much better than D-K iteration controller in both transient and steady state responses when a step signal applied as a command signal. Figure 11, shows step response of the DC motor for different values of uncertain parameters (20 states). Based on this figure, the structure-specified controller is completely robust against different uncertainties in parameters. It should be noted that in the simulations calculating destabilizing frequency by Matlab commands (The MathWorks, Inc) can be done instead of sampling the frequency axis to find peak value of μ plots.

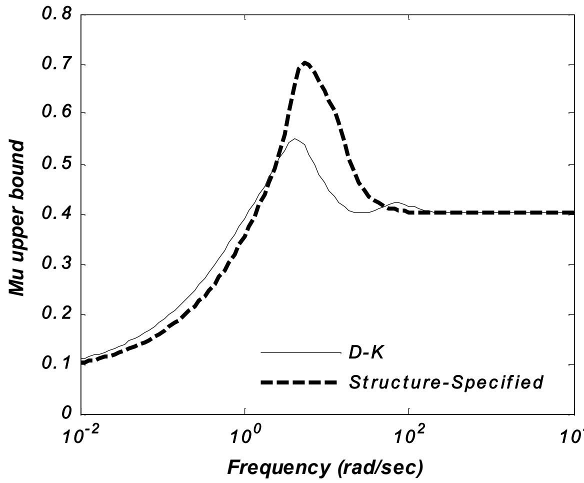

Fig. 8. Comparison of robust stability

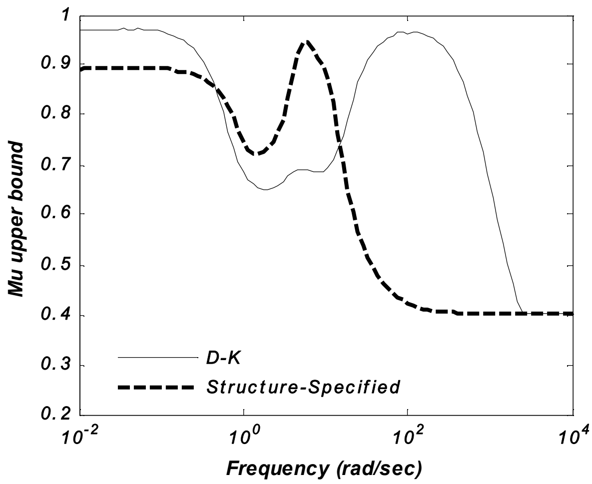

Fig. 9. Comparison of robust performance

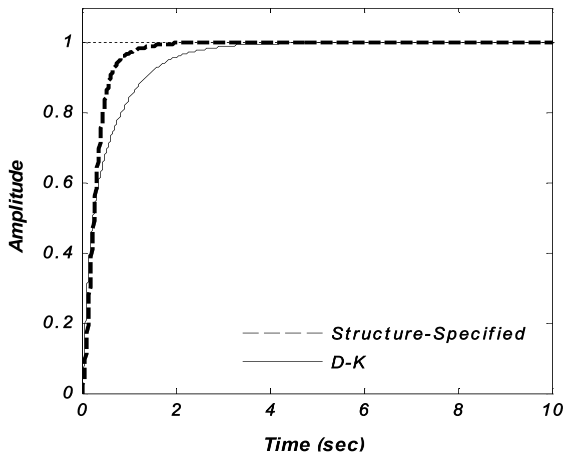

Fig. 10. Comparison of step response

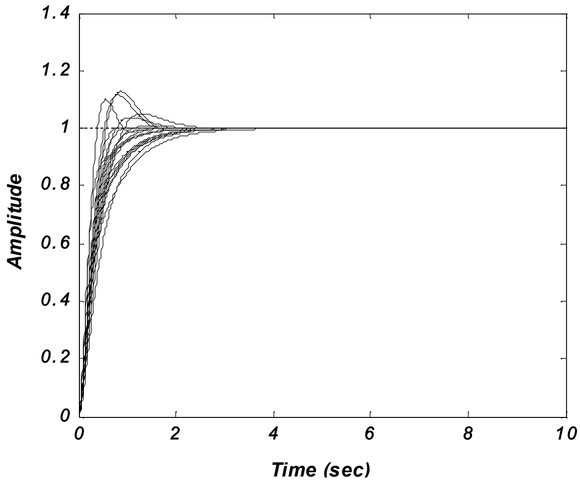

Fig. 11. Step response of structure-specified controller for different uncertainties

Finally, cost values of PSO algorithm versus iteration number is depicted in figure 12. The average computational time was about 400 seconds while the simulations were conducted on a Quad2Core CPU, 2.66 GHz, 1GB RAM computer and using Matlab version 7.11.

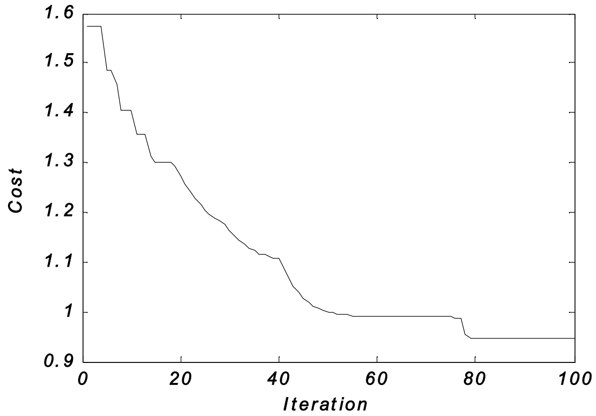

Fig. 12. Cost function of PSO algorithm

7. CONCLUSIONS

In this paper a structure-specified μ synthesis controller is designed by an evolutionary approach. In this approach the μ synthesis problem is solved as an optimization problem. Particle swarm optimization algorithm has been used as an evolutionary algorithm to solve the problem. Unlike the D-K iteration method, the proposed approach significantly improves in practical control viewpoints by simplifying the controller structure, reducing the controller order and retaining the robust performance. Simulated results demonstrate the proposed controller has better robust performance and time responses in comparison with the D-K iteration controller (while its robust stability margin is still proper).

REFERENCES

Balas, G.J., J.C. Doyle., K. Glover., A. Packard and R. Smith (1998). μ-Analysis and Synthesis Toolbox for Use with MATLAB, *The MathWork, Inc*.

Chen, B.S. and Y.M. Cheng (2005). A structure-specified H_∞ optimal control design for practical applications: a generic approach. *IEEE Trans. Contr. Syst. Technol,* **13(6)**, 1119-1124.

Franken, N. and A.P. Engelbrecht (2005). Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma. *IEEETrans. Evol. Comput.*, **9(6)**, 562–579.

Ho, S.J., S.Y. Ho., M.H. Hung., L.S. Shu and H.L. Huang (2005). Designing structure-specified Mixed H_2/H_∞ optimal controllers using an intelligent genetic algorithm IGA. *IEEE Trans. Contr. Syst. Technol*, **13(6)**, 1119-1124.

Ho, S.Y., H.S. Lin, W.H. Liauh, and S.J. Ho (2008). OPSO: Orthogonal particle swarm optimization and its application to task assignment problems. *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, **38(2),** 288–298.

Kaitwanidvilai, S. and M. Parnichkun (2008). Design of structured controller satisfying H infinite loop shaping using evolutionary optimization: application to a pneumatic robot arm. *Engineering Letters*, **16(2)**, 193-201.

Kennedy, J. and Eberhart, R. Particle swarm optimization (1995). *In Proceedings of the IEEE International Conference on Neural Networks*, 1942-1948.

Olranthichachatand , P. and S. Kaitwanidvilai (2008). GA based fixed structure H_∞ loop shaping controller for a buck-boost converter. *Engineering Letters*. **16(3)**, 346-352.

Ratnaweera, A., S.K. Halgamuge, and H.C. Watson (2004). Self_organizing hierarchical particle swarm optimizer with time_varying acceleration Coefficients. *IEEE Transactions on Evolutionary Computation,* **8** (3), 240–255.

Shi, Y. and R.C. Eberhart (1998). Parameter selection in particle swarm optimization. *In:Evolutionary Programming VII. Proc. EP98*, 591–600, Springer, New York.

Stein, G. and J. Doyle (1991). Beyond singular values and loop-shapes. *AIAA Journal of Guidance and Control*, **14(1)**, 5-16.

Suganthan, P.N. (1999). Particle Swarm Optimizer with Neighborhood Operator. In *Proceedings of IEEE International Conference on Evolutionary Computation. Washinington D.C.: IEEE Press* , 1958–1962.

Zamani, M., N. Sadati and M. Karimi Ghartemai (2009). Design of an H_∞ PID Controller Using Particle Swarm Optimization. *International Journal of Control, Automation, and Systems*, **7(2)**, 273-280.

Zhou, K. and J. Doyle (1998). *Essentials of Robust Contr*ol. Ch. 10, Prentice Hall.