

# Robust Controller Tuning for Two-Mass System via Optimization using Differential Evolution

Mahmud Iwan Solihin<sup>1</sup>, Aravind CV<sup>2</sup>, Rini Akmeliawati<sup>1</sup>

1- Intelligent Mechatronics Systems Research Unit, IIUM, Kuala Lumpur, Malaysia.

2- School of Engineering, Taylor's University, Malaysia.

Email:aravindcv@ieee.org

Received: January 2012    Revised: March 2012    Accepted: July 2012

## ABSTRACT:

Design of a robust controller via single objective constrained optimization using differential evolution (DE) is presented in this paper. A set robust feedback controller gain is optimized based on plant's linear model having structured parametric uncertainty such that the closed-loop system would have the maximum stability radius. A wedge region is assigned as the optimization constraint to specify the desired closed-loop poles location which is directly related to the desired time-domain response. The proposed controller design is applied to a two-mass system which is known as the benchmark problem for robust controller design. The simulation results show that the robustness performance is achieved in the presence of parameter variations of the plant. The proposed controller performs than the conventional LQR (linear quadratic regulator) controller.

**KEYWORDS:** Differential evolution, optimization, robust control design, Two-mass system.

## 1. INTRODUCTION

Robustness has been an important issue in any control system design. A successfully designed control system should be always able to maintain stability and performance level in spite of uncertainties in system dynamics including, parameter variations of the plant.

In the robust control theory,  $H_\infty$  optimization approach and the  $\mu$ -synthesis/analysis method are well-developed and elegant [1]. They provide systematic design procedures of robust controllers for linear systems. However, the mathematics behind the theory is not trivial and is not straightforward to formulate a practical design problem into the  $H_\infty$  or  $\mu$  design framework. These conventional robust controller designs are also followed by a lengthy parameters tuning, i.e. weighting functions.

In this work, an alternative technique of the robust feedback control design via modern constrained optimization is proposed. Differential evolution (DE) as a modern optimization algorithm is employed, which is considerably fast and reasonably robust.

A number of researches have proposed modern optimizations, particularly GA (genetic algorithm), DE or PSO (particle swarm optimization), in order to overcome the difficulties in the conventional robust control design.

Alfaro-Cid et al. [2] uses GA to tune the weighting functions' parameters for a  $H_\infty$  controller. The controller is applied to propulsion and navigation control of a ship. A low-order controller with good

tracking capability and smooth actuator signal can be obtained. However, the computation time required for the optimization using GA is usually excessive.

In other work by Thanh and Parnichkun (3), PSO is proposed in balance control of Bicyrobo, which is an unstable system with a variety of uncertainties. Specifically, PSO is used to search for parameters of a structure-specified controller which satisfies mixed  $H_2/H_\infty$  performance index. The efficiency of the proposed algorithm is compared with GA.

Furthermore, some researches on design of a robust control using DE are found. For example, Neumann and Araujo [4] used hybrid DE (by combining DE and Evolutionary Gradient Search) to solve mixed  $H_2/H_\infty$  control problem combined with pole placement constraint in LMI regions. One of their motivations in using DE is the interesting global search properties of the DE algorithm.

In another work by Smirnov and Jastrzebski [5], DE is used to select weighting functions in  $H_\infty$  loop to shape controller for active magnetic bearing system. Objective functions and constraints were developed based on singular values of the system. The results show that the proposed algorithm can improve all the objectives compared with a system tuned by a human operator.

This work combines the advantages of modern optimization algorithm with robust control theory. To deal with the plant's parameter uncertainty the complex

stability radius as a tool of measuring system robustness is used. In addition, the desired response is automatically defined by assigning a regional closed loop poles placement. This region will be incorporated in the DE-based optimization as a constraint. In other word, the proposed controller design technique is search for a set of robust feedback controller gains such that the closed-loop system would have maximum complex stability radius.

At the end of the work, the simulation results of the proposed robust control design for two-mass system is presented. This two-mass system is commonly known as a benchmark problem for robust control design [6-10].

**2. PRELIMINARIES**

**2.1. Problem Statement**

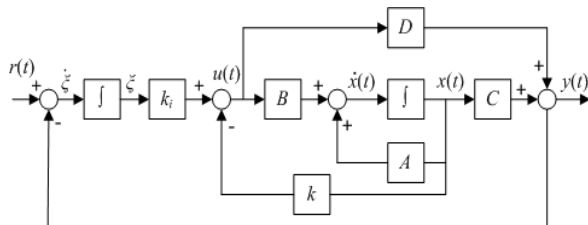
Consider a plant model of linear time-invariant continuous-time system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (1)$$

with  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{l \times m}$ ,  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$ , and  $y(t) \in \mathbb{R}^p$  are state matrix, input matrix, output matrix, state vector, control input and output vectors, respectively. It is assumed that the system given by equation (1) is completely state controllable and all state variables are available for feedback. One can use state feedback controller with feed-forward integral gain as shown in Fig. 1. The control signal ( $u$ ) is given by a linear control law:

$$u(t) = -kx(t) + k_i \xi(t) \quad (2)$$

where  $k = [k_1, k_2, k_3, \dots, k_n]$  is the state feedback gain,  $k_i$  is integral feedforward gain and  $\xi$  is output of the integrator. The controller gains for the system in Fig.1 consists of the feedback gain  $k$  and integral feedforward gain  $k_i$ , which can be computed using some conventional techniques such as pole placement or optimal control method which is well-known as linear quadratic regulator (LQR). However, these conventional techniques do not consider plant's parametric uncertainty explicitly.



**Fig. 1.** State feedback controller with feed-forward integral gain

In feedback controller design in with diagram in Fig.1, the main objective is to locate the closed-loop poles into a specific region such that the time-domain

performance is satisfactory. In addition, the obtained feedback system is also robust to parameter variation of the plant. Therefore, it is naturally a bi-objective problem.

To simplify the design process, the aforementioned problem is transformed into a single objective constrained optimization. In this work, a single objective constrained optimization using DE is employed to find a set of robust controller gains ( $K = [k \ k_i]$ ) such that closed-loop system would have maximum stability radius (explained in Section 2.2). Plant's parametric uncertainty is automatically handled with the use of stability radius. In addition, a wedge region for closed-loop poles is incorporated as optimization constraint to allow the designers to specify the desired time-domain control performance. For efficiency of the constrained optimization, a dynamic constraint handling technique (explained in Section 3.3) is adopted instead of common constraint handling technique such as penalty function approach.

**2.2. Stability radius**

In this section, a tool of measuring system robustness called stability radius [11] is presented. Stability radius is the maximum distance to instability. Equivalently, a system with larger stability radius implies that the system can tolerate more perturbations. In general, stability radius is classified as complex stability radius and real stability radius. Compared to real stability radius, complex stability radius can handle a wider class of perturbations including nonlinear, linear-time-varying, nonlinear-time-varying and nonlinear-time-varying-and-dynamics perturbations [8]. For this reason, complex stability radius is used as a measure of robustness for the feedback system.

The definition of complex stability radius is given here. Let  $\mathbb{C}$  denote the set of complex numbers.  $\mathbb{C}_- = \{z \in \mathbb{C} | \text{Real}(z) < 0\}$  and  $\mathbb{C}_+ = \mathbb{C} \setminus \mathbb{C}_-$ ; is the closed right half plane. Consider a nominal system in the form:

$$\dot{x}(t) = Ax(t) \quad (3)$$

$A(t)$  is assumed to be stable. The perturbed open-loop system is assumed as:

$$\dot{x}(t) = (A(t) + E\Delta(t)H)x(t) \quad (4)$$

where  $\Delta(t)$  is a bounded time-varying linear perturbation.  $E$  and  $H$  are scale matrices that define the structure of the perturbations. The perturbation matrix itself is unknown. The stability radius of (4) is defined as the smallest norm of  $\Delta$  for which there exists a  $\Delta$  that destabilizes (3) for the given perturbation structure  $(E, H)$ .

For the controlled perturbed system in the form (3), let:

$$G(s) = H(sI - A)^{-1}E \quad (5)$$

be the "transfer matrix" associated with  $(A, E, H)$ , then the complex stability radius is defined by the

following definition.

*Definition 1:* [11] The complex stability radius,  $r_c$ :

$$r_c(A, E, H, \mathbb{C}_+) = \left[ \max_{s \in \partial \mathbb{C}_+} \|G(s)\| \right]^{-1} \quad (6)$$

where  $\partial \mathbb{C}_+$  is the boundary of  $\mathbb{C}_+$ . In other words, a maximum  $r_c$  can be achieved by minimizing the  $H_\infty$  norm of the “transfer matrix”  $G$  [12].

*Proposition 1:* Using Definition 1, the complex stability radius of the feedback system as shown in Fig.1 is given as:

$$r_c(\tilde{A}, \tilde{E}, \tilde{H}, \mathbb{C}_+) = \left[ \max_{s \in \partial \mathbb{C}_+} \left\| \tilde{H}(sI - \tilde{A})^{-1} \tilde{E} \right\| \right]^{-1} \quad (7)$$

where  $\tilde{A} = \hat{A} - \hat{B}K$ ,  $\hat{A}$  and  $\hat{B}$  are given by the following equations:

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

For the structure of perturbation given by  $\hat{E}$  and  $\hat{H}$ , a robust control system can be obtained by maximizing  $r_c$  described by equation (7). Therefore, a suitable controller gain  $K$  can be optimized by min-max optimization algorithms.

### 3. DE-BASED CONTROL DESIGN

#### 3.1. Brief Overview of DE

A DE algorithm is a stochastic search optimization method which is fast and reasonably robust in handling non-differentiable, non-linear, and multimodal objective functions. DE is one of the most promising novel evolutionary algorithms for solving global optimization problems [13]. It was proposed by Storn and Price not long ago in 1995 [14].

The structure of DE is similar to other evolutionary algorithms. The first generation is initialized randomly and further generations evolve by applying the evolutionary operators: mutation, recombination and selection to every population member until a stopping criterion is satisfied.

There are some variants (strategies) of DE. The DE variant called DE/rand/1/bin [15] is used here. DE stands for differential evolution algorithm, *rand* means that the vector (solution candidate) to be perturbed is chosen randomly, *1* is the number of difference vectors considered for perturbation and *bin* is the binomial type of crossover being used (other type is exponential). This DE variant (DE/rand/1/bin) is considerably simple and the most competitive variant in solving various applications as reported in [16].

Furthermore, there are only few parameters defined by user in DE. Similar to other evolutionary algorithms, users have to select number of population,  $NP$ . The other control parameters are  $F$  (mutation scaling factor) and  $CR$  (crossover rate factor) which are valued between  $[0,1]$ . Outlining an absolute value for  $CR$  is difficult. However a few guidelines have been laid down [17]. It is largely problem dependent. When using binomial scheme, intermediate values of  $CR$  produce

good results. In addition, the general description of  $F$  is that it should be at least above 0.5, in order to provide sufficient scaling of the produced value [18].

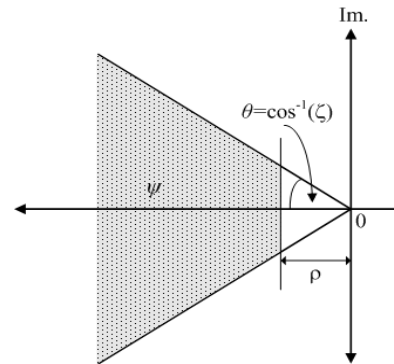
#### 3.2. Constrained Optimization

The objective of the optimization is to maximize the complex stability radius ( $r_c$ ), however in this work the  $r_c$  is converted into minimization mode by putting negative sign. Based on our approach, the searching procedure of the robust controller gains using constrained optimization can be formulated as follows (Table 1).

**Table 1.** Constrained optimization

<b>Minimize:</b>	$f(X) = -r_c(X)$	
<b>Subject to constraint:</b>	$\lambda_n(X) \in \psi$	<b>for</b>
		<b><math>n=1,2,\dots</math></b>
<b>and boundary constraint:</b>	$X \in [l_b, u_b]$	

where  $X=K=(k_1, k_2, \dots, k_n, k_i)$  is the vector solutions such that  $X \in S \subseteq R^{n+1}$ .  $S$  is the search space, and  $F \subseteq S$  is the feasible region or the region of  $S$  for which the constraint is satisfied. The constraint here is the closed loop poles region; in the feasible region, the controller gains are found such that the closed loop poles ( $\lambda$ ) lie within a wedge region ( $\psi$ ) of a complex plane as given in Fig. 2. The wedge region can be specified by two parameters  $\theta$  and  $\rho$  which are related to desired transient response characteristics i.e., damping ratio ( $\zeta$ ) and settling time ( $t_s$ ).



**Fig. 2.** A wedge region in complex plane for closed loop poles placement

#### 3.3. Constraint Handling

An efficient and adequate constraint-handling technique is a key element in the design of stochastic algorithms to solve complex optimization problems. Although the use of penalty functions is the most common technique for constraint-handling, there are a lot of different approaches for dealing with constraints [18].

Instead of using penalty approach like in [19] where the optimizer seemed to be inefficient (high iterations), a Dynamic Objective Constraint Handling Method (DOCHM) [20] is adopted here in order to improve the efficiency. By defining the distance function  $F(X)$ , DOCHM converts the original problem into bi-objective optimization problem  $\min(F(X), f(X))$ , where  $F(X)$  is treated as the first objective function and  $f(x)$  is the second (main) one.

The auxiliary distance function  $F(X)$  will be merely used to determine whether or not an individual (candidate of solution) is within the feasible region and how close a particle is to the feasible region. If an individual lies outside the feasible region (at least an eigenvalue lies outside the wedge region), the individual will take  $F(X)$  as its optimization objective. Otherwise, the individual will instead optimize the real objective function  $f(X)$ . During the optimization process if an individual leaves the feasible region, it will once again optimize  $F(X)$ . Therefore, the optimizer has the ability to dynamically drive the individuals into the feasible region.

The procedure of the DOCHM applied to the eigenvalue assignment in the wedge region is illustrated in the following pseudo-code (Table 2). Referring to Fig. 3 let  $d_n$  is an outer distance of an eigenvalue ( $\lambda_n$ ) to the wedge region. It is noted that if an eigenvalue lies within the wedge region,  $d_n=0$ .  $F(X)$  is defined by:

$$F(X) = \sum_{i=1}^{n+1} \max(0, d_n(\lambda_n(X))) \quad (8)$$

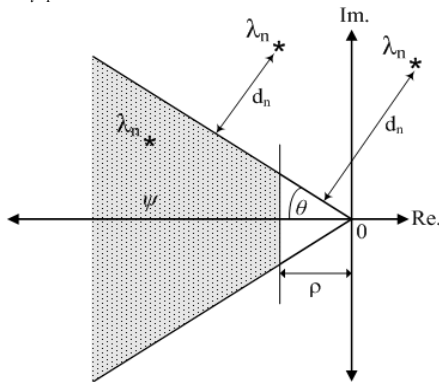


Fig. 3. Eigenvalue distance to the wedge region in complex plane

Table 2. Pseudo-code for constraint handling

<b>If</b>	$F(X) = 0$
	$f(X) = -r_c(X)$
<b>Else</b>	
	$f(X) = F(X)$
<b>End</b>	

### 3.4. Stopping criterion

In literatures, mostly two stopping criteria are applied in single-objective optimization: either an error

measure if the optimum value is known is used or the number of function evaluations (number of iterations). There are some drawbacks for both. The knowledge of the optimum has to be known in the first method however the second method is highly dependent on the objective function. Improper selection of the number of iterations to terminate the optimization can lead to either premature convergence or expensive optimization runs (excessive computational effort). As a result, it would be better to use stopping criterion that consider knowledge from the state of the optimization run. The time of termination would be determined adaptively, so the optimization run would be efficient. Several stopping criteria are reviewed in [21]. Although the authors did not conclude which one is the best for all problems, it is believed that performance improvement can be obtained with adaptive stopping criterion.

In this work, the stopping criterion which is distribution-based criterion which considers the diversity in the population is adopted. If the diversity is low, the individuals are close to each other, so it is assumed that convergence has been obtained [21]. Standard deviation ( $\sigma$ ) of the best individuals in each dimension during iterations is checked. If it is below a threshold  $\epsilon$  (small number) for sufficiently large number of iterations  $\eta$ , the optimization will be terminated. It can be formulated as in Table 3; where  $x_{best,d}^j$  represents the best individual in  $j$ -th generation (iteration) for  $d$  dimension.

Table 3. Stopping criterion

<b>If</b>	$\sigma_d = \sqrt{\frac{1}{\eta} \sum_{j=1}^{\eta} (x_{best,d}^j - \bar{x}_{best,d})^2} < \epsilon (\max(x_{best,d}) - \min(x_{best,d}))$ (for $d=1, 2, \dots, D$ )
	<b>stop iterations.</b>
<b>End</b>	

## 4. ROBUST CONTROL DESIGN FOR TWO MASS SYSTEM

In this section, an illustrative example of the proposed method to two-mass system is presented. This system has been used as benchmark problem for robust control design [5]. Consider the two-mass system shown in the Fig. 4. A control force ( $u$ ) acts on body 1 and the position of body 2 is measured. Both masses are equal to one unit ( $m_1=m_2=1$ ) and the spring constant is assumed to be in the range  $0.5 \leq k \leq 2$ . The system can be represented in state-space form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k & k & 0 & 0 \\ k & -k & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \quad (9)$$

where:  $x_1$ : position of mass-1  
 $x_2$ : position of mass-2  
 $x_3$ : velocity of mass-1  
 $x_4$ : velocity of mass-2

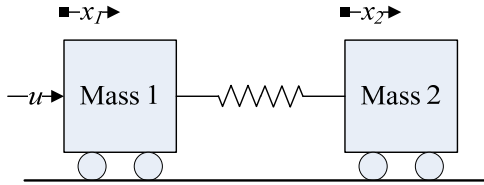


Fig. 4. Two-mass and spring system

The plant uncertainty is due to variations of the spring constant where the nominal value is selected for the worst case of  $k=0.5$ . Therefore uncertainties appear in the rows 3-4 and the columns 1-2 of the state matrix. The scale matrices as the perturbation structure for the closed loop system are  $\hat{E}$  and  $\hat{H}$  whose diagonal elements in rows 3-4 of  $\hat{E}$  and in columns 2-3 of  $\hat{H}$  are respectively equal to 1.

$$\hat{E} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \hat{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The next is to choose the parameters of the wedge region (Fig. 2) to locate the closed loop poles. The damping ratio is usually set to  $\zeta=0.7$  to produce sufficient overshoot damping in the response. The transient margin ( $\rho$ ) is specified according to the desired speed of the response. This is problem-dependent parameter and the value of  $as\rho=1$ . In addition, the main DE-based optimization parameters are listed in Table 4.

Table 4. DE-based optimization parameters

Dimension of the problem	$D$	5
Population size	$NP$	100
Mutation scaling constant	$F$	vary
Crossover rate constant	$CR$	vary
Upper and lower bounds of solution	$[l_b, u_b]$	$\pm 50$
Maximum iteration	$j_{max}$	1000
Number of iteration for which stopping criterion applies	$\eta$	100
Standard deviation threshold for which stopping criterion applies	$\varepsilon$	1%

In the first experiment, the effect of  $CR$  and  $F$  for DE-based optimization is crucial. Therefore, it will be investigated to obtain the best performance for the optimization. The results will be presented in the following section.

5. RESULTS

The optimization run has been performed in MATLAB 2006. Since DE is a stochastic optimization, a number of optimization runs need to be executed with different initial random seeds. To get an optimal solution and to evaluate the quality of the solution (robustness, convergence, repeatability), 15 runs have been executed here. For different value of  $CR$  and  $F$ , the mean value, the standard deviation of the fitness value ( $f(X)=-r_c$ ) and other results are recorded in Table 5 - Table 8.

Table 5. Optimization results with F=0.5 & CR=0.5

Average $f(X)$	-0.32
Median $f(X)$	-0.32
Standard deviation $f(X)$	0.007
Range of $f(X)$	-0.31 to -0.34
Average number of iteration	422
Average computation time	1.84minutes

Table 6. Optimization results with F=0.9 & CR=0.5

Average $f(X)$	-0.30
Median $f(X)$	-0.31
Standard deviation $f(X)$	0.011
Range of $f(X)$	-0.28 to -0.32
Average number of iteration	542
Average computation time	0.45 minutes

Table 7. Optimization results with F=0.5 & CR=0.9

Average $f(X)$	-0.34
Median $f(X)$	-0.34
Standard deviation $f(X)$	0.005
Range of $f(X)$	-0.33 to -0.35
Average number of iteration	515
Average computation time	21.3 minutes

Table 8. Optimization results with F=0.9 & CR=0.9

Average $f(X)$	-0.34
Median $f(X)$	-0.35
Standard deviation $f(X)$	0.012
Range of $f(X)$	-0.32 to -0.35
Average number of iteration	465
Average computation time	23.0 minutes

Form Table 5 – Table 8, it can be seen that DE produces the best solution when F=0.5 and CR=0.9 (Table 7), i.e. in terms of the obtained  $f(X)=-r_c$ , although the computation time is considerably long. This is due to the time required to compute the solution in feasible region. In general, a robust solution with a

small standard deviation (good repeatability) can be achieved.

The distribution of eigenvalues for those 15 runs (F=0.5 and CR=0.9) can be seen in Fig.5. All eigenvalues lie within the specified wedge region. Furthermore, to see the controller performance, a set of controller gains solution is picked from the median data of those 15 runs and it is shown in Table 9.

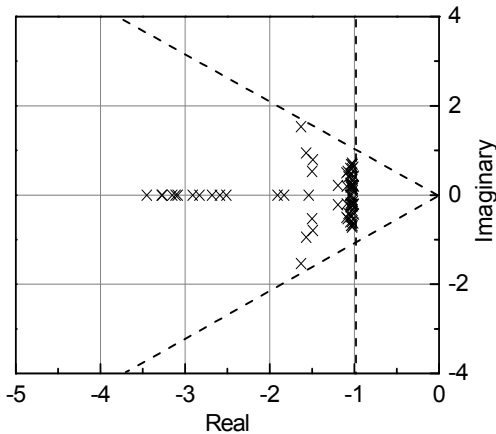


Fig. 5. Distribution of eigenvalues within the wedge region for 15 runs

Table 9. Controller gains for two-mass system

Gains	$k_1$	$k_2$	$k_3$	$k_4$	$k_i$
DEFC	18.49	19.04	47.27	7.30	-10.54
LQR1	1.57	5.76	3.23	3.54	-3.16
LQR2	8.96	20.89	5.95	6.54	-31.62

Fig. 6- Fig. 8 show 20 random samples of step response (position of the mass-2) for values of the spring constant  $0.5 \leq k \leq 2$ . The robustness performance of the proposed DE-based feedback controller (DEFC) under parameter variations of the plant is observed. For comparison, a conventional control design using LQR (linear quadratic regulator) is made. Two different LQR-based controllers (LQR1 and LQR2) are designed where the controller gains are also listed in Table 6. These two are obtained based on the following Q and R matrices:

$$Q1 = \text{diag}(10, 1, 1, 1, 10) \text{ and } R=1; \quad \text{for LQR1}$$

$$Q2 = \text{diag}(100, 1, 1, 1, 1000) \text{ and } R=1; \quad \text{for LQR2.}$$

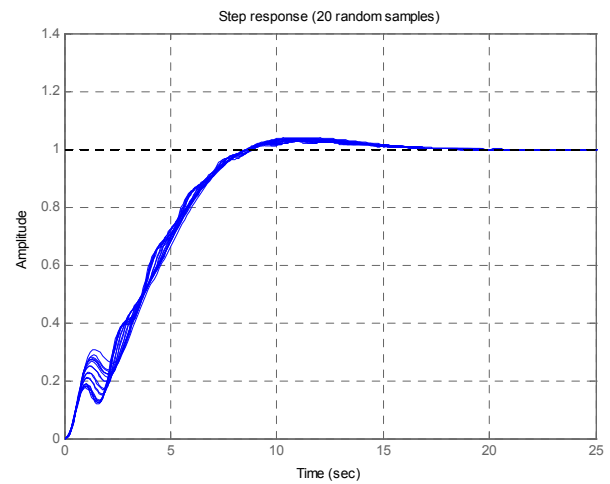


Fig. 6. 20 random step response of mass-2 displacement for DEFC

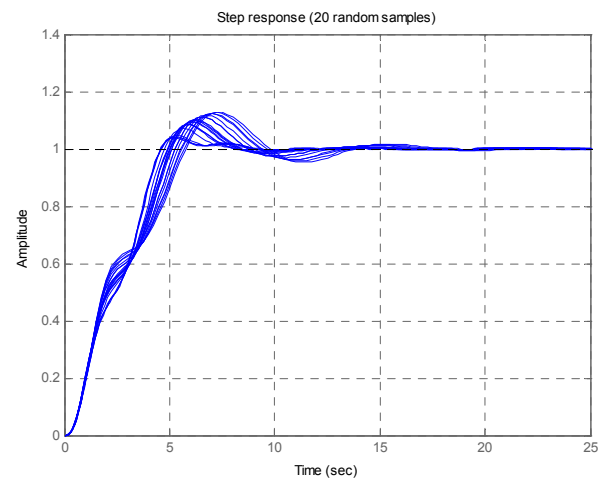


Fig. 7. 20 random step response of mass-2 displacement for LQR1

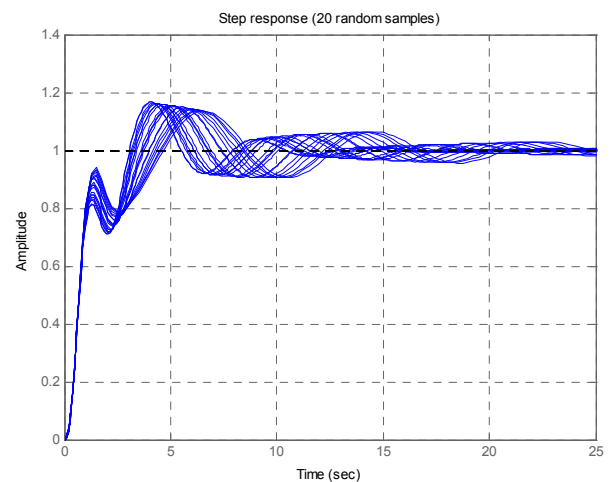


Fig. 8. 20 random step response of mass-2 displacement for LQR2

The time-domain performance comparison for the feedback system with DEFC, LQR1 and LQR2 is shown in Table 10. It is presented in terms of settling time ( $t_s$ ) and percentage of overshoot (PO);(both are averaged from those 20 random step responses). It is clear that the performance of the system with DEFC is better as compared to that with LQR1 and LQR2.

**Table 10.** Performance comparison

Controller	$t_s(sec)$	PO(%)
DEFC	13	2
LQR1	12	10
LQR2	22	17

## 6. CONCLUSIONS

A robust state feedback control design via single objective constrained optimization using DE to maximize stability radius has been proposed. The designed controller has shown the robust performance in the presence of parameter variations of the plant.

In addition, the DE-based constrained optimization effectively locates closed loop poles within a prescribed wedge region and able to maximize the stability radius with a good repeatability performance (especially when  $F=0.5$  and  $CR=0.9$  are set). The results have also shown the effectiveness of DE algorithm with the dynamic-objective constraint handling method adopted in this work.

However, it is necessary to further improve the performance of the DE-based optimization. Specifically, it is important to improve the computation time of DE-based optimization. In addition, it is also necessary to compare the performance of DE with other modern optimization techniques, such as particle swarm optimization, firefly algorithm, harmony search, etc., for this specific application.

## REFERENCES

- [1] Gu, D.W., Petkov, P.H., Konstantinov, M.M. (2005). "Robust Control Design with MATLAB". Springer-Verlag, London.
- [2] Alfaro-Cid, E., McGookin, E.W., & Murray-Smith, D.J. (2008). "Optimisation of the weighting functions of an  $H_\infty$  controller using genetic algorithms and structured genetic algorithms". *International Journal of Systems Science*, 39(4), 335–347.
- [3] Thanh, B.T., & Parnichkun, M. (2008). "Balancing control of bicyrobo by PSO-based structure-specified mixed  $H_2/H_\infty$  control". *International Journal of Advanced Robotic Systems*, 5(4), 395-402.
- [4] Nuemann, D. and Araujo, H.X. (2005). "Hybrid differential evolution method for the mixed  $H_2/H_\infty$  robust control problem under pole assignment". *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*. 1319-1324.
- [5] Smirnov, A., & Jastrzebski, R.P. (2009). "Differential evolution approach for tuning an  $H_\infty$  controller in AMB systems". *Proceedings of IEEE Annual Conference of Industrial Electronics*, 1514 – 1518.
- [6] Ly, U.L. (1990). "Robust control design using nonlinear constrained optimization. Benchmark problems for robust control design". *Proceedings of American Control Conference*, pp. 968-969.
- [7] Rotstein, H and Sideris, A. (1991). "Constrained  $H_\infty$  optimization: A design example". *Proceedings of the 30th Conference on Decision and Control*.
- [8] Byrns, E.V and Calise, A.J. (1992). "A loop transfer recovery approach to  $H_\infty$  design for the coupled mass benchmark problem". *Journal of Guidance, Control, and Dynamics*. vol. 15, no. 5, pp. 1118-1124
- [9] Wie, B and Berstein, D.S. (1990). "Benchmark problems for robust control design". *Proceedings of American Control Conference*, pp. 961-962.
- [10] Petersen, I.R., Ugrinovskii, V.A and Savkin, A.V. (2000). "Robust control design using  $H_\infty$  method". Springer-Verlag, London. pp.151-155.
- [11] Hinrichsen, D and Pritchard, A.J. (1986). "Stability radii of linear systems". *Systems & Control Letters*, 7:1–10.
- [12] Akmeliawati, R., and Tan, C. P. (2005). "Feedback controller and observer design to maximize stability radius". *Proceedings of the International Conference on Industrial Technology*, pp. 660-664.
- [13] Yousefi, H., Handroos, H., Soleymani, A. (2008). "Application of differential evolution in system identification of a servo-hydraulic system with a flexible load". *Mechatronics 18* : 513–528
- [14] R. Storn and K. Price. (1995). "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report", *International Computer Science Institute*, Berkley.
- [15] K. V. Price.D. Corne, M. Dorigo, and F. Glover, Eds. (1999). "An introduction to differential evolution, in New Ideas in Optimization", London: McGraw-Hill, pp. 79–108.
- [16] Mezura-Montes, E., Reyes, J.V., and Coello-Coello, C.A. (2006). "A comparative study of differential evolution variants for global optimization". *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 485-492
- [17] Davendra, D. and Onwubolu, G. (2009). "Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization". *Studies in Computational Intelligence*, Volume 175. Springer-Verlag Heidelberg.
- [18] Coello Coello, C.A. (2002). "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art". *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287.
- [19] Solihin, M.I, Wahyudi, Legowo, A and Akmeliawati, R. (2009). "Self-erecting inverted pendulum employing PSO for stabilizing and tracking controller". *Proceedings of 5th International Colloquium on Signal Processing & Its Applications*, pp.63-68.

- [20] Lu, H. and Chen, W. (2006). **“Dynamic-objective particle swarm optimization for constrained optimization problems”**. *Journal of Global Optimization*, 12:409-419.
- [21] Zielinski, K. and Laur, R. (2007). **“Stopping criteria for a constrained single-objective particle swarm optimization”**. *Informatika 31*: 51–59.