International Journal of the Physical Sciences Vol. 6(30), pp. 6958 - 6966, 23 November, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.726
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Linear quadratic optimal control system design using particle swarm optimization algorithm

S. Mobayen, A. Rabiei*, M. Moradi and B. Mohammady

Department of Engineering, Abhar Branch, Islamic Azad University, Abhar, Iran.

Accepted 18 August, 2011

Selecting appropriate weighting matrices for desired linear quadratic regulator (LQR) controller design using evolutionary algorithms is presented in this paper. Obviously, it is not easy to determine the appropriate weighting matrices for an optimal control system and a suitable systematic method is not presented for this goal. In other words, there is no direct relationship between weighting matrices and control system characteristics, and selecting these matrices is done by using trial and error based on designer's experience. In this paper, we use the particle swarm optimization (PSO) method which is inspired by the social behavior of fish and birds in finding food sources to determine these matrices. Stable convergence characteristics and high calculation speed are the advantages of the proposed method. Simulation results demonstrate that in comparison with genetic algorithms (GAs), the PSO method is very efficient and robust in designing of optimal LQR controller.

Key words: Linear quadratic regulator (LQR), weighting matrices, particle swarm optimization, genetic algorithm.

INTRODUCTION

In designing of many systems and solving their problems, we need to choose a solution between feasible solutions as an optimal solution. But, because of the wide range of solution, all of them cannot be tested, so the test should be performed stochastically. On the other hand, this stochastic procedure should lead to the best answer (Athens, 1966).

For the sake of simple implementation of engineering problems, special attention has been paid on linear quadratic optimal control theory. Linear quadratic optimal control is significant for modern control theory and it can be implemented easily for engineering applications and it is the basic theory of other control techniques. However, in a special case which the cost function is a linear quadratic function, the optimal answer converges to linear quadratic regulator (LQR). LQR has a simple process and can achieve the closed loop optimal control with linear state or output feedback. This method has a spread application in the aspects such as induction motors control, vehicular drive-shaft control and airplane system control (Kirk, 1937; Athens, 1966).

The selection of LQR weighting matrices is very significant and it affects the control input (Neto et al., 2010). Various methods have been proposed in order to select suitable weight matrices in LQR controller design. Kalman (1964) proposed a method to determine the weight matrices based on given poles for the first time and Wang (1992) developed this research. Recently, many attempts are performed to design the LQR controller using genetic algorithms (GAs) (Bottura and Neto, 1999; Bottura and Neto, 2000; Sung and Chen, 2006). GA is a stochastic search method that helps the designer to achieve optimal solution.

The controller design problem is defined as choosing the weighting matrices such that the desired performance of control system is satisfied in a minimum possible time. In this paper, we propose particle swarm optimization (PSO) method for determining weighting matrices and illustrating that the achieved results satisfy the control system requirements and desired system characteristics. Also, the superiorities of the aforementioned method are compared with GA.

*Corresponding author. E-mail: abbasrabiee@yahoo.com.

LINEAR QUADRATIC OPTIMAL CONTROL

The stare space representation of a linear time-invariant (LTI) system is as follows (Nguyen and Gajic, 2010):

$$\begin{aligned} \dot{x} &= Ax + Bu \\ u &= -Kx \end{aligned} \qquad (1)$$

where x and u are $n\times 1$ state vector and $m\times 1$ input vector, respectively. A and B are constant matrices, (A,B) is a stabilizable pair and K is state feedback matrix. The linear quadratic cost function is defined as:

$$J = \frac{1}{2}\int_{t_0}^{t_f} (x^T Qx + u^T Ru)dt \qquad (2)$$

where Q is $n\times n$ semi-positive definite matrix and R is $m\times m$ positive definite matrix. The conventional LQ optimal control problem is to find the optimal input u^* such that the cost function J is minimized. For $t_f = \infty$, the state feedback matrix $K = R^{-1}B^T G$ is obtained by solving the following algebraic Riccati equation:

$$Q - GBR^{-1}B^T G + A^T G + GA = 0 \qquad (3)$$

When the control objective in optimal control systems is assigned, the weighting matrices are chosen and the corresponding optimal state feedback matrix is then unique (Fisher and Bhattacharya, 2009; Yang, 2011). However, there are no suitable systematic techniques yet for selecting the weighting matrices. The selection of weighting matrices is based on the designer's experience and usually performed by trial and error. In this work, according to the importance of weighting matrices selection, we employ the PSO method to find the optimal weighting matrices in the desired LQR controller design and compare it with other evolutionary method, that is, GA.

OVERVIEW ON GA AND PSO METHODS

In the following, a brief review of GA and PSO algorithms are presented.

Genetic algorithm (GA)

GA is a search method based on the principles of natural genetics and natural selection, and is widely recognized as an effective optimization paradigm in various areas (Davis, 1991). This algorithm was first described by John Holland (1975) over the course of the 1960s and 1970s and was popularized by David Goldberg who was able to solve a hard problem such as the controlling of gas-pipeline transmission for his thesis (Davis, 1991; Cabral and Melo, 2011). The biological origin of this algorithm is Darwinian natural selection. Considering Darwin's original ideas, life in all its diverse forms is evolved by natural selection and adaptation processes controlled by the survivability of the fittest species. GA is an evolutionary optimizer that takes a sample of possible individuals and employs selection, crossover and mutation as the primary operators for optimization. Binary genetic algorithm introduces variables as an encoded binary string, and works with the binary strings to arrive at the global best solution and maximize the fitness (that is, minimize the cost function).

The optimization process is performed in cycles entitled generations and in each generation, a set of the chromosomes is created using the crossover, inversion, and mutation stages and only the best chromosomes are allowed to survive to the next cycle of reproduction.

Particle swarm optimization (PSO)

Considering the social performance of the swarm of fishes, birds, bees and other animals, the concept of the PSO method is developed. The PSO is a robust stochastic evolutionary computation method based on the movement of swarms looking for the most fertile feeding location (Eberhart and Kennedy, 1995a, 1995b).

From the earlier statements, it is clear that the theoretical basis of the two optimization methods rest upon two completely different structures. The GA is based on genetic encoding and natural selection, and PSO method is based on social swarm behavior. PSO is based on the principle that all solutions can be represented as particles in a swarm. Each particle has a position and velocity vector and each position coordinate, represents a parameter value. Similar to GA, PSO method requires a fitness evaluation function that takes the particle's position and assigns a fitness value to it. X_{PB} and X_{GB} are the personal best position and global best position of the i^{th} particle. Each particle is initialized with a random position and velocity. The velocity of each particle is accelerated toward the global best and its own personal best according to the following equation:

$$\begin{aligned} V_i(new) = {} & w \times V_i(old) + c_1 \times rand() \times (X_{PB} - X_i) \\ & + c_2 \times Rand() \times (X_{GB} - X_i) \end{aligned} \qquad (4)$$

Here, $rand()$ and $Rand()$ are the random numbers in the range between 0 and 1, c_1 and c_2 are the acceleration constants and w is the inertia weight factor. The parameter w helps the particles converge to global best, rather than oscillating around it. Suitable selection of w provides a balance between global and local searches. In general, w is set according to the following equation (Eberhart and Kennedy, 1995a):

$$w = 0.5(1 + rand(0,1)) \qquad (5)$$

The positions are updated based on particles movement over discrete time interval (Δt) as follows:

$$X_i = X_i + V_i \times \Delta t \qquad (6)$$

Therefore, the fitness at each position is re-evaluated. If any fitness is greater than the global best value, then the new position becomes global best and the particles are accelerated toward this new point. If the particle's fitness value is greater than personal

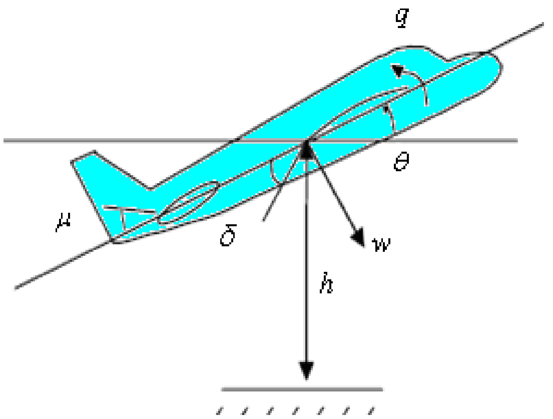

Figure 1. Aircraft landing system.

best value, then the personal best is replaced by the current position.

DETERMINATION OF WEIGHTING MATRICES

PSO algorithm stages for searching proper weighting matrices are as follows:

First, specify the lower and upper bounds of the parameters and initialize the particles of the population (weighting matrices elements) randomly. The controller gains are then calculated using LQR command in Matlab® software. It is important to note that two conditions $\det(R) \geq 0$ and $\det(Q) > 0$ must be satisfied. After that, the control energy matrix (u) and state variable (x) are calculated. Then, the linear quadratic cost function (J) is evaluated for each particle. If the cost for local best solution is less than cost of the current global best solution, the global solution is replaced with the local solution. In each stage, the program saves the cost value and minimum error value and according to the following equation, the velocity of each particle K is modified:

$$k_i^{(t+1)} = k_i^{(t)} + v_i^{(t+1)} \tag{7}$$

where $K_i^{(t)}$ is the position of i^{th} particle in time t (Marinaki et al., 2011; Xiong and Wan, 2010). At the end of each iteration, the algorithm checks the stopping criterion. If the number of iterations reaches the maximum designated by the user, the latest global best solution is recorded and the algorithm comes to the end. In order to investigate the performance of PSO algorithm for determination of proper weighting matrices in LQR controller design, the simulations are run on landing flare system and compared with the results obtained by GA.

SIMULATION RESULTS

An aircraft landing system illustrated in Figure 1 is used for simulation purpose. This system is a high dimensional system and has 6 state variables. The state-space modeling of this system is presented in Equation 8.

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} -0.058 & 0.065 & 0 & -0.171 & 0 & 1 \\ -0.303 & -0.685 & 1.109 & 0 & 0 & 0 \\ 0.072 & -0.685 & -0.947 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1.133 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.571 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \\ e \end{bmatrix} + \begin{bmatrix} 0 & 0 & -0.119 \\ -0.054 & 0 & 0.074 \\ -1.117 & 0 & 0.115 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.571 & 0 \end{bmatrix} \begin{bmatrix} \mu \\ \gamma \\ \delta \end{bmatrix} \tag{8}$$

where u is the aircraft forward speed (m/s), w is the velocity downwards at right angles (m/s), q is the angular velocity of pitch with respect to the ground (rad/s), θ is the pitch with respect to the ground ($\deg$), h is the height with respect to the ground (m), e is the forward acceleration (m/s^2), μ is the elevator angle ($\deg$), γ is the throttle value (m/s^2) and δ is the spoiler angle ($\deg$).

The input $(\mu \ \gamma \ \delta)^T$ is designed such that the aircraft comes into land in the following exponential path:

$$\dot{h} + 0.2h = 0 \tag{9}$$

The control criterion is to minimize the integral of absolute error (IAE), that is, $IAE = \int_0^{t_f} |e| dt$, (for t_f = 30 s in this paper). The system states initial values are as follows:

$$\begin{aligned} x(0) &= [u(0) \ \ w(0) \ \ q(0) \ \ \theta(0) \ \ h(0) \ \ e(0)]^T \\ &= [\,5 \quad -2.5 \quad -1 \quad -3 \quad 15 \quad 0.5]^T \end{aligned}$$

From Equation 8, we have $\dot{h} = -w + 1.133\theta$. Substituting this equation into Equation 9, we can obtain:

$$\int_0^{30} |-w + 1.133\theta + 0.2h| dt = \int_0^{30} |-x_2 + 1.133x_4 + 0.2x_5| dt \tag{10}$$

where $w = x_2$, $\theta = x_4$ and $h = x_5$.

The design method first is to select the matrices Q and R. In second step, Equations 1 and 3 are solved using computer computations, and the simulation results reveal whether any of the system constraints exceed or not. If the constraints are not satisfied, weighting matrices are reselected and this procedure is repeated. According to

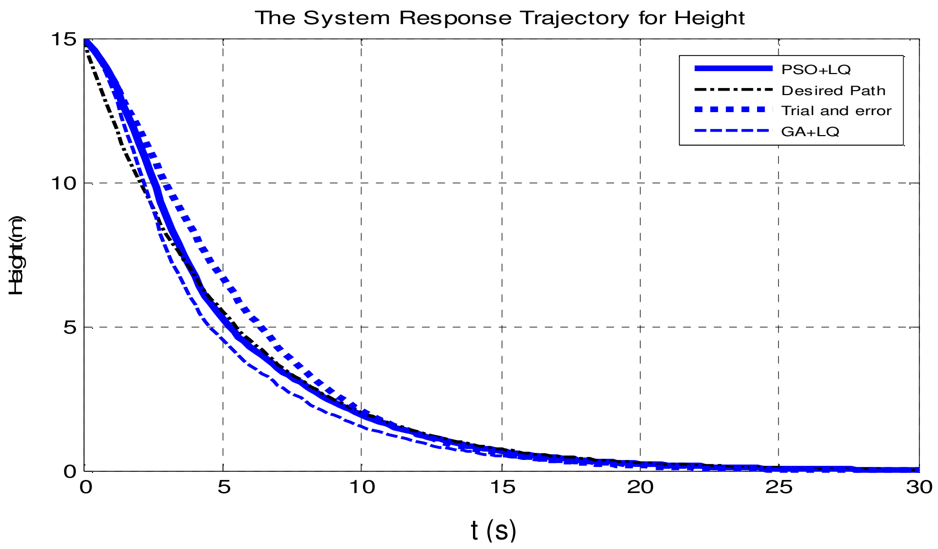

Figure 2. Landing system time response.

this algorithm, the weighting matrices selection is difficult and longer time for simulation is needed using trial and error. One of the obtained results using trial and error method is:

$$Q=\begin{bmatrix}0.618 & 0 & 0 & 0 & 0 & 0\\ 0 & 0.073 & 0 & -0.129 & -0.8 & 0\\ 0 & 0 & 0.484 & 0 & 0 & 0\\ 0 & -0.129 & 0 & 0.055 & 0.667 & 0\\ 0 & -0.8 & 0 & 0.667 & 0.054 & 0\\ 0 & 0 & 0 & 0 & 0.798 & 0\end{bmatrix},\ R=\begin{bmatrix}0.621 & 0 & 0\\ 0 & 0.421 & 0\\ 0 & 0 & 0.143\end{bmatrix},$$

and the state feedback matrix is calculated as follows:

$$K=\begin{bmatrix}-0.3477 & 0.805 & -0.8403 & -1.6266 & -0.1734 & -0.1620\\ 1.0778 & -0.2552 & 0.1345 & 0.4379 & 0.1669 & 1.5443\\ -1.0942 & 0.4669 & -0.0395 & -0.8032 & -0.4046 & -0.679\end{bmatrix}.$$

Simulation results of airplane landing trajectory is shown in Figure 2. The IAE value using trial and error method is 32.448.

PSO and GA methods simulation results

The optimal weighting matrices Q and R and the state feedback matrix K obtained by GA method are as follow:

$$Q=\begin{bmatrix}0.8576 & 0 & 0 & 0 & 0 & 0\\ 0 & 0.2632 & 0 & -0.5673 & -0.9422 & 0\\ 0 & 0 & 0.3183 & 0 & 0 & 0\\ 0 & -0.5673 & 0 & 0.1033 & 0.248 & 0\\ 0 & -0.9422 & 0 & 0.248 & 0.3585 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.9617\end{bmatrix}$$

$$R=\begin{bmatrix}0.1749 & 0 & 0\\ 0 & 0.3625 & 0\\ 0 & 0 & 0.1477\end{bmatrix}$$

$$K=\begin{bmatrix}-0.7012 & 1.5938 & -0.9416 & -2.8291 & -0.366 & -0.3625\\ 1.3722 & -0.5417 & 0.2795 & 1.014 & 0.4719 & 1.4732\\ -0.6538 & 0.5989 & -0.1836 & -1.1315 & -0.5803 & -0.407\end{bmatrix}$$

where the necessary parameters used in GA are population size = 50 chromosomes, crossover rate = 0.96, mutation rate = 0.1, search interval = [0, 20] and generation number = 60.

Also, the weighting matrices and state feedback matrix obtained by PSO method are as follows:

$$Q=\begin{bmatrix}0.7568 & 0 & 0 & 0 & 0 & 0\\ 0 & 0.4415 & 0 & -0.7114 & -0.9582 & 0\\ 0 & 0 & 0.2555 & 0 & 0 & 0\\ 0 & -0.7114 & 0 & 0.1109 & 0.3899 & 0\\ 0 & -0.9582 & 0 & 0.3899 & 0.2494 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.2334\end{bmatrix}$$

$$R=\begin{bmatrix}0.5213 & 0 & 0\\ 0 & 0.3814 & 0\\ 0 & 0 & 0.281\end{bmatrix}$$

$$K=\begin{bmatrix}-0.7012 & 1.5938 & -0.9416 & -2.8291 & -0.366 & -0.3625\\ 1.3722 & -0.5417 & 0.2795 & 1.014 & 0.4719 & 1.4732\\ -0.6538 & 0.5989 & -0.1836 & -1.1315 & -0.5803 & -0.407\end{bmatrix}$$

where the parameters used in PSO algorithm are population size = 50 particles, search interval = [0, 20], generation number = 60 and acceleration constants: $c_1=1.5$, $c_2=1.5$.

Table 1. Integral of absolute error (IAE) for airplane trajectory using different methods.

Trajectory following	Trial and error	GA + LQ	PSO + LQ
IAE	32.448	22.214	10.406

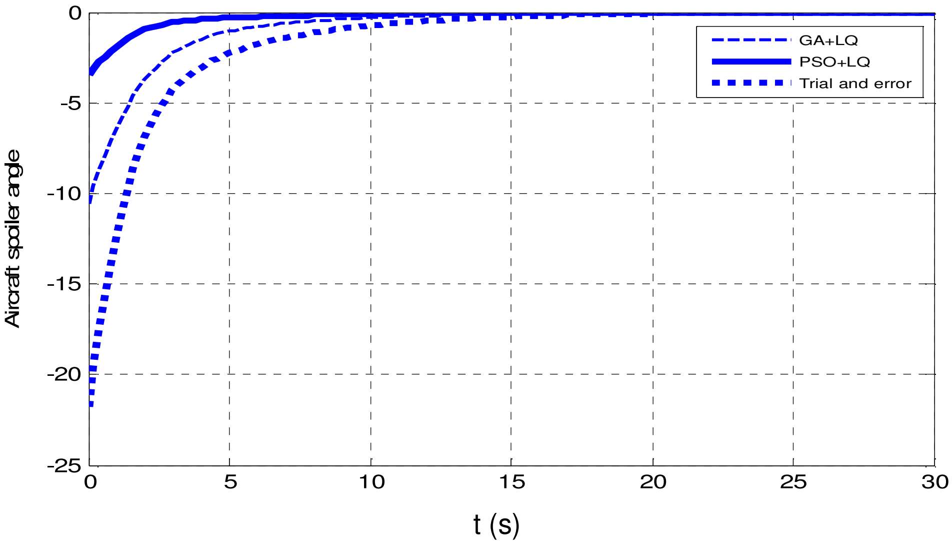

Figure 3. Landing system elevator angle.

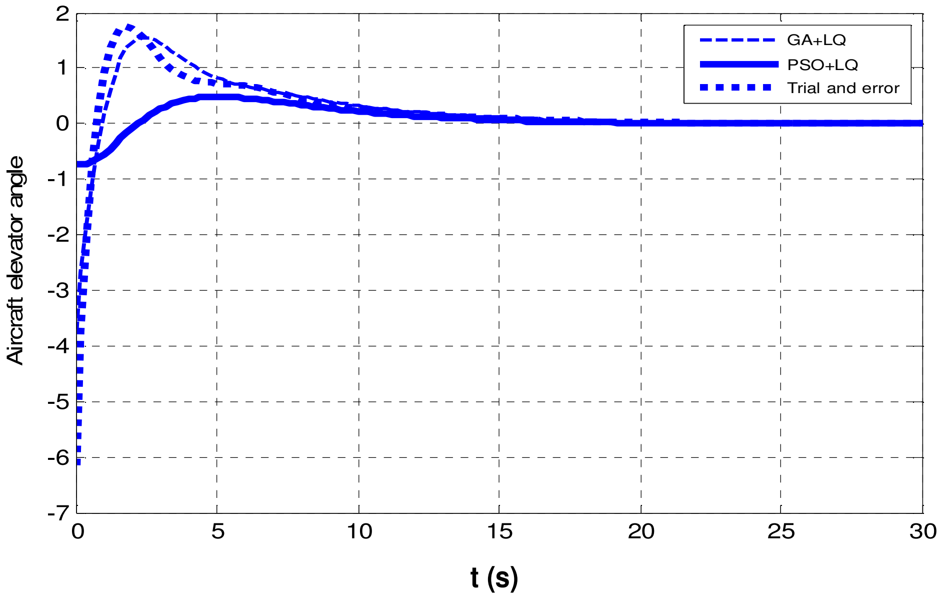

Figure 4. Landing system spoiler angle.

The Integral of absolute error (IAE) values using the proposed methods are indicated in Table 1. Figure 2 shows that the aircraft comes into land in a specified trajectory by determining weighting matrices obtained by PSO algorithm. The system input vector for elevator and spoiler angle are shown in Figures 3 and 4, respectively. These figures show that the proposed PSO method works better in improving the control system performance when compared with GA algorithm.

Also, Table 2 illustrates that the IAE values are strongly

Table 2. Integral of absolute error (IAE) for control inputs using various methods.

IAE	Elevator angle	Throttle value	Spoiler angle
Trial and error	28.305	71.235	124.598
GA + LQ	27.2	62.236	64.189
PSO + LQ	11.852	39.824	22.621

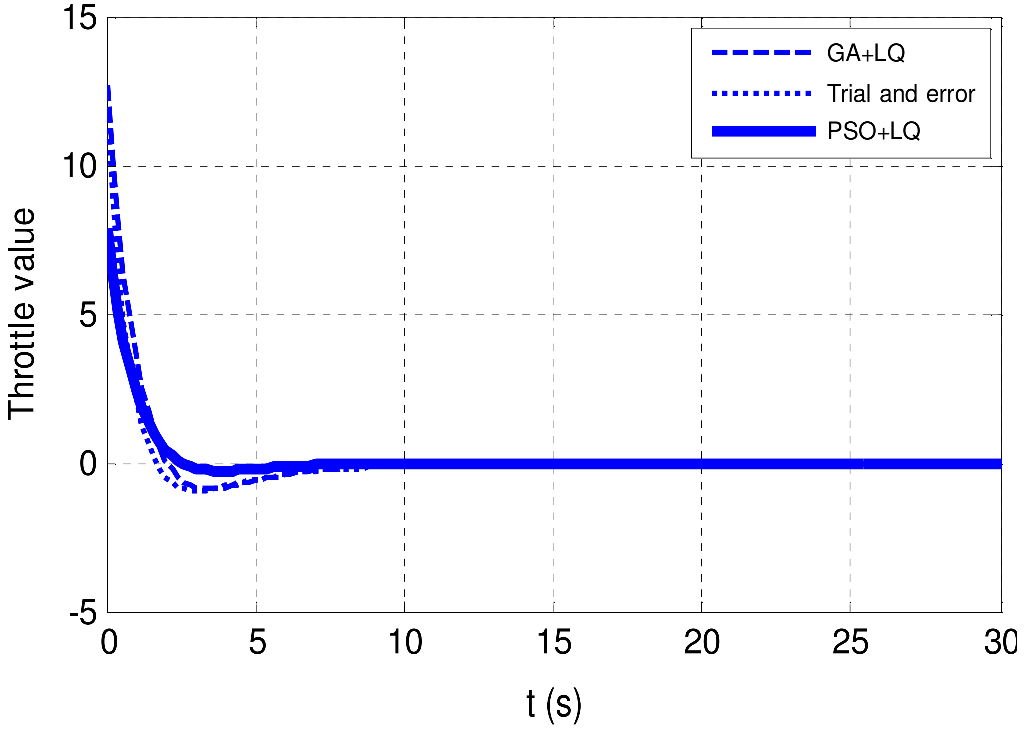

Figure 5. Throttle value control.

decreased using the PSO algorithm. Figure 5 shows the throttle value control.

Robustness analysis

The absolute value of matrices A and B element are increased by 10% in order to analyze the control system robustness against parameters variations.

The performances of the designed controllers are studied. The new A and B matrices are as follows:

$$A = \begin{bmatrix} -0.0638 & 0.0715 & 0 & -0.188 & 0 & 1.1 \\ -0.334 & -0.72 & 1.22 & 0 & 0 & 0 \\ 0.079 & -0.753 & -1.0417 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0 & 0 & 0 \\ 0 & -1.1 & 0 & 1.24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.62 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & -0.13 \\ -0.061 & 0 & 0.081 \\ -1.0053 & 0 & 0.126 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.61 & 0 \end{bmatrix}.$$

The simulation results of aircraft landing trajectory, elevator angle, spoiler angle and throttle value using new matrices are shown in Figures 6 to 9, respectively. The IAE values for aircraft landing trajectory and control inputs are compared in Tables 3 and 4. According to the simulation results, we can observe that the designed system is very robust versus parameters variations.

CONCLUSION

In this paper, a new method for determining weighting matrices R and Q for optimal control system design

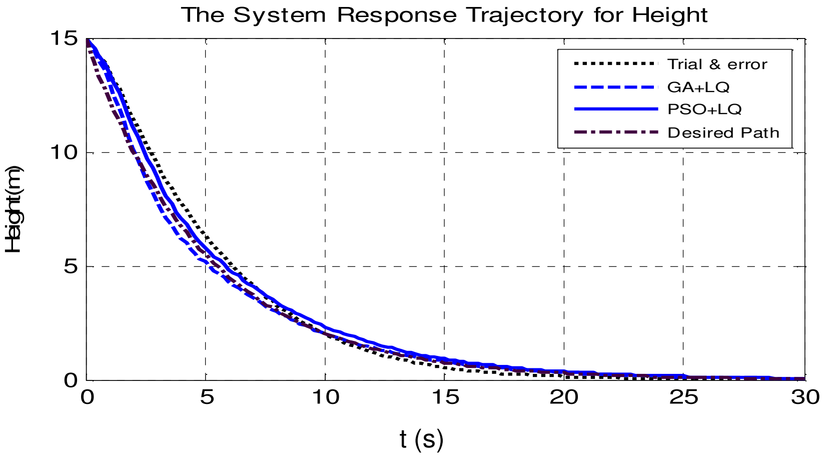

Figure 6. Landing system time response using new matrices (robustness analysis).

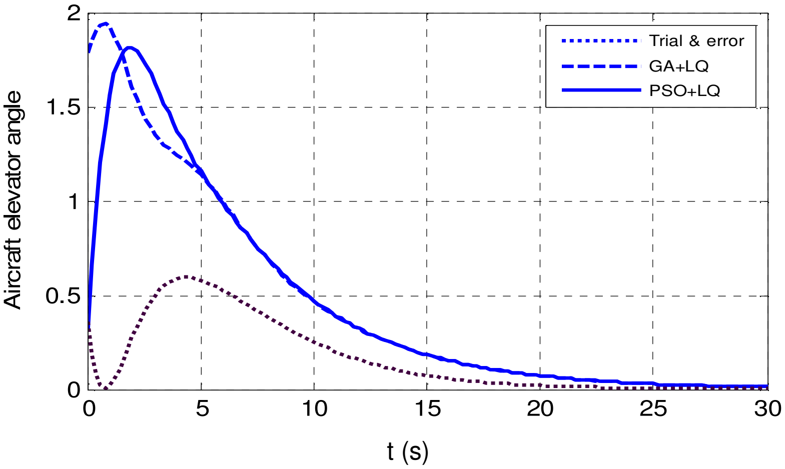

Figure 7. Landing system elevator angle using new matrices (robustness analysis).

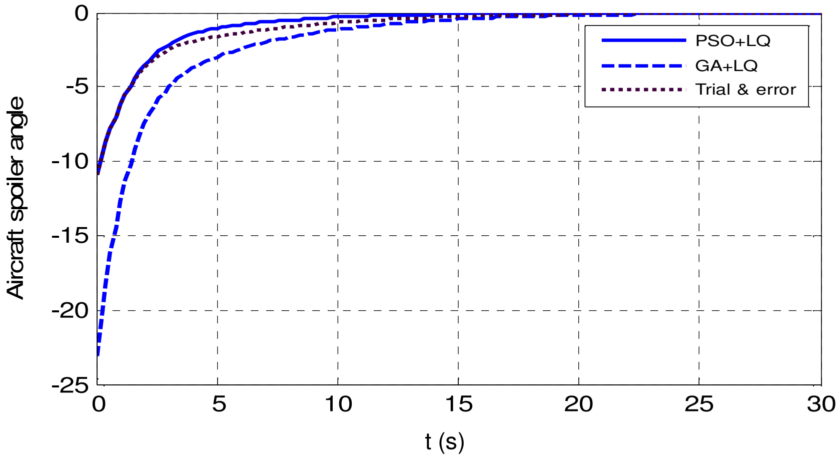

Figure 8. Spoiler angle using new matrices (robustness analysis).

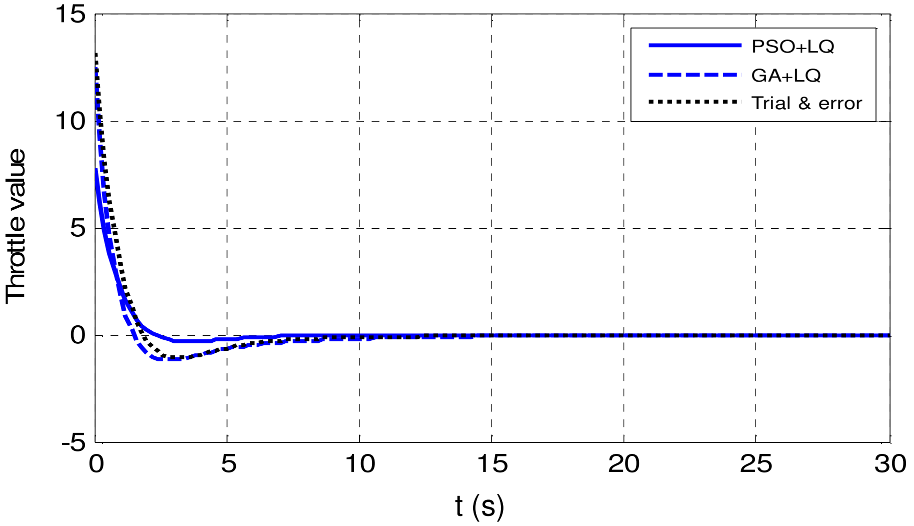

Figure 9. Throttle value control using new matrices (robustness analysis).

Table 3. Integral of absolute error (IAE) for airplane trajectory following (in the new situation).

Trajectory following	Trial and error	GA + LQ	PSO + LQ
IAE	29.463	12.776	14.544

Table 4. Integral of absolute error (IAE) for control inputs using various methods (in the new situation).

IAE	Elevator angle	Throttle value	Spoiler angle
Trial and error	35.243	74.067	65.71
GA + LQ	34.052	69.324	147.326
PSO + LQ	35.129	37.694	60.792

using PSO algorithm is proposed. High promising results demonstrate that the proposed method is very flexible, efficient and robust against changes in parameters, and can obtain higher quality solution with better computational efficiency and fast convergence. The simulation results are very satisfactory in comparison with previous experiments, that is, GA.

REFERENCES

Athens M (1966). The status of optimal control theory and applications for deterministic systems. IEEE Trans. Automatic Control, 11(3): 580-596.

Bottura CP, Neto F (1999). Parallel eigenstructure assignment via LQR design and genetic algorithms. Proc. of the American Control Conference, San Diego, pp. 2295-2299.

Bottura CP, Neto F (2000). Rule based decision making unit for eigenstructure assignment via parallel genetic algorithm and LQR design. Proc. of the American Control Conference. Chicago, Illinois, 1(6): 467-471.

Cabral HA, Melo MTD (2011). Using genetic algorithms for device modeling. IEEE Trans. Magnetics. 47(5): 1322-1325.

Eberhart RC, Kennedy J (1995a). A new Optimizer Using Particle Swarm Theory. Proc. of 6th Symposium on Micro Machine and Human Science. Nagoya, Japan, pp. 39-43.

Davis E (1991). Handbook of genetic algorithms. New York, Van Nostrand Reinhold.

Fisher J, Bhattacharya R (2009). Linear quadratic regulation of systems with stochastic parameter uncertainties. Automatica. 45(12): 2831-2841.

Kalman RE (1964). When is a linear control system optimal? J. Basci Eng. Trans. ASME., 86: 51-56.

Kennedy J, Eberhart RC (1995b). Particle swarm optimization. In Proc. of IEEE International Conference on Neural Networks. Perth, Australia, 4: 1942-1948.

Kirk DE (1937). Optimal control theory: an introduction. California, Carmel, pp. 70-94.

Marinaki M, Marinakis Y, Stavroulakis E (2011). Vibration control of beams with piezoelectric sensors and actuators using particle swarm optimization. Expert Systems with Applications. 38(6): 6872- 6883.

Neto JV, Abreu IS, Silva FN (2010). Neural–genetic synthesis for state-space controllers based on linear quadratic regulator design for eigenstructure Assignment. IEEE Trans. Systems, Man, and Cybernetics, 40(2): 266-285.

Nguyen T, Gajic Z (2010). Solving the matrix differential Riccati equation: A Lyapunov equation approach. IEEE Trans. Automatic Control, 55(1): 191-194.

Sung GC, Chen G (2006). Optimal control systems design associated with genetic algorithm. Proc. of 2006 CACS Automatic Control Conference St. John's University. Tamsui, Taiwan, pp. 10-11.

Wang Y (1992). The determination of weighting matrices in LQ optimal control system. ACTA Automatic Sinica, 18(2): 213-217.

Xiong X, Wan Z (2010). The simulation of double inverted pendulum control based on particle swarm optimization LQR algorithm. IEEE International Conference on Software Engineering and Service Sciences (ICSESS), pp. 253-256.

Yang Y (2011). Analytic LQR design for spacecraft control system based on quaternion model. J. Aerospace Eng., 1(105): 1-19.