

Optimal Tuning of Linear Quadratic Regulators Using Quantum Particle Swarm Optimization

Kaveh Hassani, Won-Sook Lee

University of Ottawa, School of Electrical Engineering and Computer Science
161 Louis Pasteur, Ottawa, Ontario, Canada K1N 6N5
kaveh.hassani@uottawa.ca; wslee@uottawa.ca

Abstract - Linear Quadratic Regulator (LQR) is an optimal multivariable feedback control approach that minimizes the excursion in state trajectories of a system while requiring minimum controller effort. The behaviour of a LQR controller is determined by two parameters: state and control weighting matrices. These two matrices are main design parameters to be selected by designer and greatly influence the success of the LQR controller synthesis. However, it is not a trivial task to decide these two matrices. The classic approaches such as trial-and-error, Bryson's method, and pole placement are labour-intensive, time consuming and do not guarantee the expected performance. Furthermore, these techniques only aim to minimize the quadratic performance index and do not consider other control objectives such as minimizing the overshoot, rise time, settling time, and steady state error. In this paper, for the first time, we apply quantum particle swarm optimization (QPSO) algorithm to automatically and optimally adjust weighting matrices. QPSO is an extension of conventional PSO algorithm, in which particles obey the quantum mechanics rather than Newtonian mechanics. We applied the proposed approach to stabilize an inverted pendulum system. The results suggest that QPSO-based LQR outperforms LQR tuned by trial-and-error, genetic algorithm and conventional PSO methods in terms of rising time, settling time and quadratic performance index. Also, it is competitive with mentioned approaches in terms of maximum overshoot percentage and steady-state error.

Keywords: Linear quadratic regulator, quantum particle swarm optimization.

1. Introduction

Optimal control theory refers to controller design paradigms that force the target process to satisfy the physical constraints and optimize predetermined performance criteria, concurrently. The evolution of optimal control theory has led to the emergence of linear quadratic regulator (LQR) which is an optimal multivariable feedback control approach that minimizes the excursion in state trajectories of a system while requiring minimum controller effort, and improves stability performance. In other words, applying LQR approach to a controllable linear time-invariant (LTI) system results in a set of optimal feedback gains that minimize a quadratic criterion and stabilize the system (Lewis, 1986). LQR has been widely exploited in various applications such as but not limited to missile guidance, flight control, control of autonomous ground vehicles, active car suspension, ABS break system, power converters, and optimal tuning of PID controllers.

LQR minimizes a quadratic cost function (i.e. performance index) that contains two penalty matrices: state weighting matrix (Q) and control weighting matrix (R). These two parameters are main design parameters to be selected by designer, and greatly influence the behaviour of the LQR controller. However, it is not a trivial task to decide these two matrices. The problem of selecting weighting matrices has been investigated by various methods. In general, weighting matrices are determined by a trial-and-error method in which an expert adjusts the weighting matrices intuitively, and then refines them iteratively to obtain a satisfying performance. The trial-and-error method is not feasible for high dimensional systems and even for simpler systems is a labour-intensive and time consuming approach. Bryson's method (Johnson and Grimble, 1987) is another iterative method. In this technique, initially, state and feedback variables are normalized with respect to their largest permissible, and utilized to initialize the weighting matrices. Then, similar to trial-and-error method, the weighting matrices are

gradually refined to approach the minimum index value. Pole placement is another popular technique for determining the weighting matrices (Saif, 1989). However, in pole placement technique, the weighting matrices are based on the given poles, and thus cannot guarantee the performance and constraints of the system. Although other approaches such as utilizing asymptotic modal properties (Harvey and Stein, 2003), and expressing the system as an explicit function of the weighting matrix elements (Tyler and Tuteur, 1962) are proposed, they suffer from the similar deficiencies.

The classic approaches are labour-intensive, time consuming and do not guarantee the expected performance. Furthermore, these techniques only aim to minimize the quadratic performance index and do not consider other control objectives such as minimizing the overshoot, rise time, settling time, and steady state error. In order to enhance the accuracy, design process and performance of the system, and mitigate the drawbacks of classic techniques, computational intelligence paradigms have been exploited. Among them, as far as the authors' knowledge is concerned, genetic algorithm (GA) which mimics the biological natural selection process, is the most frequently applied algorithm. Although GA is well-founded and indicates high exploration capability, it suffers from two pitfalls: low exploitation and convergence speed. On the other hand, particle swarm optimization (PSO) algorithm, a swarm intelligence based technique inspired by social interactions of flocking birds, is able to reach the globally optimal solution within a few iterations. It has experimentally shown that the PSO is scalable and its processing time grows at a linear rate with respect to the size of the problem (Akay, 2013).

In this paper, for the first time, we apply quantum PSO (QPSO) meta-heuristic technique proposed by Sun et al. (2004) to optimally adjust weighting matrices. QPSO is an extension of conventional PSO algorithm, in which particles obey the quantum mechanics rather than Newtonian mechanics. We applied the proposed algorithm to stabilize an inverted pendulum (IP) system. The results suggest that QPSO-based LQR outperforms LQR tuned by trial-and-error, GA and classic PSO methods in terms of rising time, settling time and quadratic performance index. Also, in terms of maximum overshoot percentage and steady-state error, it is competitive with mentioned approaches.

The paper is organized as follows: in section 2 an overview of related works is presented. In section 3, we describe the characteristics of a general LQR. In section 4, we investigate the QPSO algorithm and then present our proposed technique in section 5. In section 6, we discuss the evaluations and experimental results. Finally, section 7 concludes the paper.

2. Related Works

During the last two decades, some computational intelligence techniques such as artificial immune systems (AIS) (Ramaswamy, 2007), artificial bee colony (ABC) (Changhao and Duan, 2013), ant colony optimization (ACO) (Ali et al. 2008), genetic algorithm (GA) (Wongsathan and Sirima, 2008), probabilistic differential evolution (PDE) (Liouane et al. 2012), memetic algorithm (MA) (Zhang et al. 2011), imperialist competitive algorithm (ICA) (Rakhshani, 2012), artificial neural networks (ANN) (Wiklendt et al. 2009), and fuzzy logic (Taoa et al. 2010) have been employed to enhance the performance of LQR controllers. As far as the authors' knowledge is concerned, GA is the most frequent biologically inspired algorithm applied for adjusting LQR weighting matrices.

Superiority of PSO over GA in finding optimal weighting matrices of LQR controller has been shown in some studies (Ghoreishi and Nekoui, 2012; Chen and Sung, 2012). Hamidi (2012) applied PSO, GA and error-and-trial methods to adjust the LQR weighting matrices which, in turn, is applied to control an aircraft landing flare system. He concluded that PSO-based LQR is more efficient and robust compared to other methods. Vardhana et al. (2009) compared ordinary LQR, LQR with prescribed degree of stability (LQRPDS) and PSO-based LQR for controlling distribution static compensator, and showed that PSO-based LQR gives the best performance under different operating conditions over the other two controller design techniques. Solihin et al. (2010) proposed a method for determining weighting matrices by applying PSO with pole region constraint for controlling a flexible-link manipulator. Selamat and Bilong (2013) suggested that PSO-based LQR controller produces better result compared to trial-and-error approach for the active suspension system. Rakhshani (2012) applied conventional LQR, PSO-based LQR, adaptive weighted PSO (AWPSO)-based LQR and ICA-based LQR for optimal load frequency

control, and concluded that AWPSO-based LQR outperforms other approaches in terms of settling time and maximum overshoot. Guoping et al. (2010) proposed an optimal real-time LQR controller based on PSO for the control of an IP system. Amini et al. (2013) devised wavelet PSO-based LQR technique and applied it for optimal structural control. HaiBin and ChangHao (2013) introduced a PSO-based LQR controller for stabilizing the pendulum-like oscillation and showed that it enhances the micro-aerial vehicles' surveillance performance. Karanki et al. (2010) concluded that PSO-based state feedback controller, unlike the trial-and-error approach, does not have sub-optimal performance in the case of partial state feedback. More studies on PSO-based LQR can be found in (Khan et al. 2011; Solihin and Akmeliawati, 2009; Tsaia et al. 2013; Xiong and Wan, 2010; Yu and Huang, 2008; Yuan et al. 2009). It is noteworthy that as far as the authors' knowledge is concerned, it is for the first time that quantum PSO technique is being applied for tuning LQR controller.

3. Optimal LQR Controller

For a controllable LTI system with a space state model shown in Eq. (1):

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

Where $x(t)$, $y(t)$ and $u(t)$ are n , r , and m dimensional state, output, and control vectors, respectively, LQR approach constructs a linear state feedback law (i.e. control law) as Eq. (2):

$$u(t) = -Kx(t) \quad (2)$$

The control law indicated in Eq. (2) minimizes the quadratic performance index shown in Eq. (3). The performance index consists of state and control energies.

$$J = \int_0^{\infty} [x^T Qx + u^T Ru] dt \quad (3)$$

Where $Q \in \mathcal{R}^{n \times n}$ is the symmetric, positive semi-definite state weighting (i.e. state penalty) matrix, and $R \in \mathcal{R}^{m \times m}$ is the symmetric, positive definite control weighting (i.e. control penalty) matrix. In Eq. (2), K is the control gain matrix given by Eq. (4).

$$K = R^{-1}B^T P \quad (4)$$

Where P is the unique symmetric, positive semi-definite solution to the algebraic Riccati equation as depicted in Eq. (5).

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (5)$$

Using the control law depicted in Eq. (2) guarantees to maintain the output as close as possible to the desired output with minimum control energy.

4. Quantum Particle Swarm Optimization

PSO algorithm introduced by Eberhart and Kennedy (1995) is a swarm intelligence based meta-heuristic approach inspired by the individual and social behaviour of the flocking birds. PSO algorithm consists of population of candidate solutions called particles. Each particle is characterized by its position and velocity vectors, and follows its trajectory towards the global optimum based on Newtonian

mechanics. In a D-dimensional space, position and velocity vectors of the i th particle are depicted in Eq. (6). Also, the rules governing the particle's trajectory are depicted Eq. (7) and (8), respectively.

$$\vec{x}_i(t) = [x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t)]^T, \quad \vec{v}_i(t) = [v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t)]^T \quad (6)$$

$$\vec{v}_i(t+1) = w \cdot \vec{v}_i(t) + \varphi_1^T (\vec{x}_{pbi} - \vec{x}_i(t)) + \varphi_2^T (\vec{x}_g - \vec{x}_i(t)) \quad (7)$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (8)$$

Where x and v are position and velocity vectors, respectively. w is inertia utilized to enhance the convergence speed and balance between exploitation and exploration. φ_1 and φ_2 are two positive parameters known as personal and social cognitive factors, respectively. x_{pbi} is the best personal position that i th particle have seen so far, and x_g is the position of the best particle seen so far in the swarm history. It has been shown that if the upper limits of cognitive parameters are selected properly, the position of the i th particle converges to position p computed as Eq. (9). Position p can be interpreted as the centre of gravity towards which particles are careen while their kinetic energy declines.

$$p_d = (\varphi_{1d} \times x_{pbd} + \varphi_{2d} \times x_{gd}) / (\varphi_{1d} + \varphi_{2d}) \quad (9)$$

Sun et al. (2004) proposed a novel version of PSO algorithm in which particles obey the quantum mechanics rather than Newtonian mechanics. In this approach, the quantum state of a particle is expressed by a wave function based on Schrodinger's uncertainty principle. Also, the trajectory of a particle is modelled using quantum Delta potential well model. In this model, it is assumed that a particle moves in a Delta potential well in search space, of which the centre is point p calculated by Eq. (9). In order to compute the fitness of an individual particle, its exact position is needed. However, in quantum model, only the probability density function of the position is available. To address this issue, Monte Carlo method is applied to simulate the measurement process from wave function. Finally, particles' estimated position vector is governed by Eq. (10).

$$x_{id} = p_d \pm \frac{|x_{id} - p_d| \ln(1/u)}{g} \quad (10)$$

u is a random number between 0 and 1, and g is a control parameter greater than $\ln\sqrt{2}$.

5. Proposed Method

The schematic of our proposed method is shown in Figure 1. We apply QPSO algorithm to automatically and optimally adjust the LQR weighting matrices. Similar to practical applications, we define the $Q = \text{diag}[Q_1, Q_2, \dots, Q_n]$ and $R = \text{diag}[R_1, R_2, \dots, R_m]$ as diagonal matrices to alleviate the curse of dimensionality problem (n and m are dimensions of state and control vectors, respectively). We consider the concatenation of Q and R matrices as our individual particle representation as depicted in Eq. (11).

$$\vec{x}_i(t) = [x_{i1}(t), \dots, x_{i(n+m)}(t)]^T = [Q_{i1}(t), Q_{i2}(t), \dots, Q_{in}(t), R_{i1}(t), R_{i2}(t), \dots, R_{im}(t)]^T \quad (11)$$

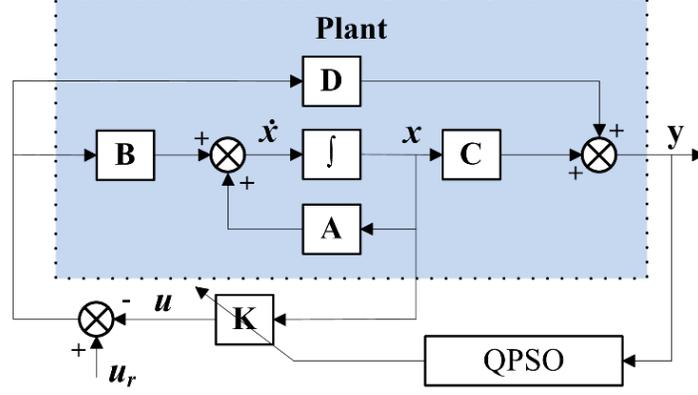


Fig. 1. The schematic of proposed method for tuning LQR using QPSO algorithm.

In order to evaluate a particle's fitness value, we employ a hybrid time domain criterion. In addition to minimizing the quadratic performance index depicted in Eq. (3), an optimal controller should minimize overshoot, rise time, settling time and steady-state error as well. Furthermore, according to the definition, Q and R matrices must be positive. Thus, those particles with negative values in their position vector are not feasible solutions. To address this issue, we add a penalty value to those individuals to weaken their fitness. Finally, we integrate mentioned elements in a weighted cost function to get the overall cost of a specific configuration of Q and R matrices as depicted in Eq. (12).

$$f_i = \lambda_1 \log_{10}(J) + \lambda_2 kp + \lambda_3(OS + E_{ss}) + \lambda_4(T_s - T_r), \quad \sum \lambda_i = 1 \quad (12)$$

Where f_i is the cost function of particle i , $\log_{10}(J)$ is the normalized quadratic performance index depicted in Eq. (3), k is number of negative values in i th particle, p is penalty value for negative values, OS is overshoot percentage, E_{ss} is steady-state error, T_s is settling time, T_r is rise time, and $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are corresponding weights. We propose our QPSO-based LQR tuning algorithm as Eq. (13).

Initialize population: random p_i, q_i (13)

For j = 1 to iteration N

For i = 1 to population size M

Calculate P from Riccati equation as Eq. (5)

Calculate the feedback gain K using Eq. (4)

Simulate the closed – loop system with new controller using Eq. (2)

Calculate the fitness of particle using Eq. (12)

If $f(x_i) < f(p_i)$ then $p_i = x_i$

$p_g = \text{Min}(p_i)$

For d = 1 to dimension D

$\varphi_1 = \text{rand}(0,1), \varphi_2 = \text{rand}(0,1)$

$p = (\varphi_1 \times p_{id} + \varphi_2 \times p_{gd}) / (\varphi_1 + \varphi_2)$

$u = \text{rand}(0,1)$

$L = (1/g) \times |x_{id} - p|$

If $\text{rand}(0,1) > 0.5$ then $x_{id} = p - L \times \ln(1/u)$

else $x_{id} = p + L \times \ln(1/u)$

End

End

End

return p_g

6. Experimental Results

In order to evaluate our proposed method, we employed it to design a LQR controller which in turn, is applied to stabilize a 2-DOF IP system. A 2-DOF IP system consists of a pendulum mounted on a movable cart which is restricted to linear motion. The control objective is to maintain the unstable equilibrium position by moving the cart along a horizontal track. The pendulum initially starts in an upright position. A schematic of a 2-DOF IP system is shown in Figure 2.

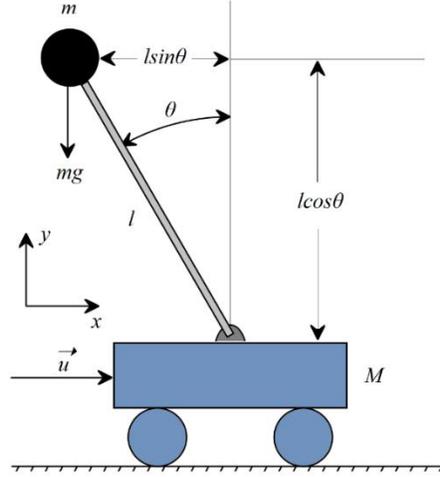


Fig. 2. Inverted pendulum-cart system.

The state space representation of the IP system shown in Figure 2 derived by Lagrange's equation and linearized using Taylor series expansion is depicted in Eq. (14).

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(m+M)}{lM} & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{lM} \end{bmatrix} u \quad (14)$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

In order to evaluate the performance of our proposed method and compare its results with existing approaches, we stabilized the IP system by using our approach, trial-and-error method, GA, and ordinary PSO algorithm. We set the mass of cart to 0.5kg , mass of pendulum to 0.2kg , length of pendulum to 0.6m , and gravitational acceleration to 9.81ms^{-2} . For GA, PSO and QDPSO techniques, we set the population size to 20 and the iteration numbers to 100. Also, we set all the weights of cost function ($\lambda_1, \lambda_2, \lambda_3$ and λ_4) to 0.25 (i.e. equal weights for all elements). The penalty value is set to 5. For GA algorithm, the crossover and mutation probabilities are set to 0.7 and 0.1, respectively, and tournament method is utilized for selection process. For PSO algorithm, the social, individual and inertia weights are set to 2, 2, and 0.5, respectively. Finally, for QPSO algorithm the parameter g is set to $1.5ln\sqrt{2}$. The simulation results are shown in Table 1.

Table 1. Results of stabilizing a 2-DOF IP system by trial-and-error, GA, PSO and QPSO methods.

Applied method	Elements of control objective				
	E_{ss}	OS	T_r	T_s	J
Trial-and-error	0	23 %	0.46 s	2.09 s	26000
GA	0	85 %	0.54 s	1.19 s	22000
PSO	0	89 %	0.48 s	1.11 s	21600
QPSO	0	89 %	0.43 s	1.02 s	20500

Results indicate that none of the four methods have steady-state error. Also, it is shown that trial-and-error method has the least overshoot percentage, whereas PSO and QPSO have the largest overshoot percentage. Although high overshoot percentage is not satisfactory, it indicates that PSO and QPSO respond faster than other two methods. Furthermore, QPSO has the smallest rising and settling times in comparison with other methods. In other words, QPSO-based LQR responds to the input and stabilizes the system faster than other three methods. Finally, the results show that QPSO leads to smallest performance index value and control effort. All in all, the results suggest that our proposed method outperforms other techniques in terms of rising time, settling time and quadratic performance index. Also, it is competitive in terms of maximum overshoot percentage and steady-state error.

7. Conclusion

In this paper, for the first time, we applied quantum particle swarm optimization for automatic and optimal tuning of weighting matrices of a general linear quadratic regulator. We applied the proposed method to stabilize a 2-DOF inverted pendulum system and compared the results with trial-and-error, genetic algorithm and particle swarm optimization methods. The results suggest that our proposed method outperforms other techniques in terms of rising time, settling time and quadratic performance index. Also, it is competitive in terms of maximum overshoot percentage and steady-state error. As future works, we are planning to extend the proposed approach in two direction. First, we will enhance the utilized weighted cost function to a multi-objective cost function to acquire a set of optimal solutions rather than one. Second, we will enhance the online learning by embedding parallel processing techniques.

References

- Akay B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Appl. Soft. Comput.* 13, 3066–3091.
- Ali D., Hend L., Hassani M. (2008). Optimised Eigenstructure assignment by ant system and LQR approaches. *Int. J. Comput. Sci. Appl.*, 5, 45-56.
- Amini F., Hazaveh N.K., Rad A.A. (2013). Wavelet PSO-based LQR algorithm for optimal structural control using active tuned mass dampers. *Computer-Aided Civil and Infrastru. Eng.*, 28, 542–557.
- Changhao S., Duan H. (2013). Artificial bee colony optimized controller for unmanned rotorcraft pendulum. *Aircraft Eng. Aerosp. Technol.*, 85, 104-114.
- Chen R., Sung W. (2012). A PSO-based LQR controller for accelerator PWM power supply. *Adv. Mat. Res.*, 490, 71-75.
- Eberhart R., Kennedy J. (1995). A New Optimizer Using Particle Swarm Theory, “Proc. 6th Symp. Micro Mach. Man. Sci.”, Nagoya, Japan, Oct. 4-6, pp. 29–43.
- Ghoreishi S.A, Nekoui M.A. (2012). Optimal weighting matrices design for LQR controller based on genetic algorithm and PSO. *Adv. Mat. Res.*, 443, 7546-7553.
- Guoping L., Genfu X., Xiaohui Y. (2010). The LQR Real-time Control for the Inverted Pendulum Based on PSO, "Int. Conf. Elect. And Contr. Eng. ", Wuhan, China, June 25-27, pp. 2363-2366.
- HaiBin H., ChangHao S. (2013). Pendulum-like oscillation controller for micro aerial vehicle with ducted fan based on LQR and PSO. *Sci. CHINA Tech. Sci.*, 56, 423–429.

- Hamidi J. (2012). Control system design using particle swarm optimization (PSO). *Int. J. Soft Comput. Eng.*, 1, 116-119.
- Harvey C., Stein G. (2003). Quadratic weights for asymptotic regulator properties. *IEEE Trans. Automat. Contr.*, 23, 378-387.
- Johnson M.A., Grimble M.J. (1987). Recent trends in linear optimal quadratic multivariable control system design. *IEE Proc. Contr. Theory and Appl.*, 134, 53-71.
- Karanki S.B., Mishra M.K., Kumar B.K. (2010). Particle swarm optimization-based feedback controller for unified power-quality conditioner. *IEEE Trans. Power Delivery*, 25, 2814-2824.
- Khan A.H., Weiguo Z., Jingping S., Khan Z.H. (2011). Optimized reconfigurable modular flight control design using swarm intelligence. *Procedia Eng.*, 24, 621-628.
- Lewis F.L. (1986). "Optimal Control" John-Wiley.
- Liouane H., Chiha I., Douik A., Messaoud H. (2012). Probabilistic Differential Evolution for Optimal Design of LQR Weighting Matrices, "IEEE Int. Conf. Comput. Intell. Meas. Syst. and Appl.", Tianjin, China, July 2-4, pp. 18-23.
- Rakhshani E. (2012). Intelligent linear-quadratic optimal output feedback regulator for a deregulated automatic generation control system. *Elect. Power Compon. Syst.*, 40, 513-533.
- Ramaswamy S.A.P., Venayagamoorthy G.K., Balakrishnan S.N. (2007). Optimal Control of Class of Non-linear Plants Using Artificial Immune Systems Application of the Clonal Selection Algorithm, "IEEE 22nd Int. Symp. Intell. Contr.", Singapore, Oct. 1-3, pp. 249-254.
- Saif M. (1989). Optimal linear regulator pole-placement by weight selection. *Int. J. Contr.*, 50, 399-414.
- Selamat H., Bilong S.D.A. (2013). Optimal Controller Design for a Railway Vehicle Suspension System Using Particle Swarm Optimization, "9th Asian Contr. Conf.", Istanbul, Turkey, June 23-36, pp. 1-5.
- Solihin M., Akmeliawati R. (2009). PSO Optimization of State Feedback Tracking Controller for a Flexible Link Manipulator, "Int. Conf. Soft Comput. Pattern Recog.", Malaysia, Dec. 4-7, pp. 72-76.
- Solihin M.I., Legowo A., Akmeliawati R. (2010). Comparison of LQR and PSO-based State Feedback Controller for Tracking Control of a Flexible Link Manipulator, "The 2nd IEEE Int. Conf. on Info. Manag. and Eng.", Chengdu, China, Apr. 16-18, pp. 354-358.
- Sun J., Feng B., Xu W. (2004). Particle Swam Optimization with Particles Having Quantum Behavior, "Proc. of Cong. on Evol. Comput." Portland, US, June. 19-23, pp. 325-331.
- Taoa C.W., Taurb J.S., Chena Y.C. (2010). Design of a parallel distributed fuzzy LQR controller for the twin rotor multi-input multi-output system. *Fuzzy Sets and Syst.*, 161, 2081-2103.
- Tsaia S., Huob C., Yangc Y., Sun T. (2013). Variable feedback gain control design based on particle swarm optimizer for automatic fighter tracking problems. *Appl. Soft. Comput.*, 13, 58-75.
- Tyler J., Tuteur F.B. (1962). The use of a quadratic performance index to design multivariable control systems. *IEEE Trans. Automat. Contr.*, 11, 84-92.
- Vardhana P.H., Kumar B.K., Kumar M. (2009). A Robust Controller for DSTATCOM, "Int. Conf. on Power Eng. Energy and Electr. Drives", Lisbon, Portugal, Mar. 18-20, pp. 546-551.
- Wiklendt L., Chalup S., Middleton R. (2009). A small spiking neural network with LQR control applied to the acrobot. *Neural Comput. & Appl.*, 18, 369-375.
- Wongsathan C., Sirima C. (2008). Application of GA to Design LQR Controller for an Inverted Pendulum System, "IEEE Int. Conf. Robot. Bio., Bangkok, Thailand, Feb. 22-25, pp. 951-954.
- Xiong X., Wan Z. (2010). The Simulation of Double Inverted Pendulum Control Based on Particle Swarm Optimization LQR algorithm, "IEEE Int. Conf. Softw. Eng. Service Sci.", Beijing, China, July 16-18, pp. 253-256.
- Yu K., Huang Z. (2008). LQ Regulator Design Based on Particle Swarm Optimization, "IEEE Int. Conf. on Syst., Man, Cybern." Taipei, China, Oct. 8-11, pp. 4142-4145.
- Yuan s., Wang D., Li X. (2009). Research on Control Problem of PenduBot Based on PSO Algorithm, "Int. Conf. on Comput. Intell. And Natural Comput." Wuhan, China, June. 6-7, pp. 346-349.
- Zhang J., Zhang L., Xie J. (2011). Application of Memetic Algorithm in Control of Linear Inverted Pendulum, "IEEE Int. Conf. Cloud Comput. Intell. Syst.", Beijing, China, Sept. 15-17, pp. 103-107.