ARCHIVES OF ELECTRICAL ENGINEERING VOL. 63(3), pp. 345-365 (2014)

DOI 10.2478/aee-2014-0026

An adaptive particle swarm optimization algorithm for robust trajectory tracking of a class of under actuated system

VINODH KUMAR E, JOVITHA JEROME

Department of Instrumentation and Control Systems Engineering, PSG College of Technology Coimbatore, India-641004

e-mail: vinothmepsg@gmail.com, jjovitha@yahoo.com

(Received: 23.09.2013, revised: 19.05.2014)

Abstract: This paper presents an adaptive particle swarm optimization (APSO) based LQR controller for optimal tuning of state feedback controller gains for a class of under actuated system (Inverted pendulum). Normally, the weights of LQR controller are chosen based on trial and error approach to obtain the optimum controller gains, but it is often cumbersome and tedious to tune the controller gains via trial and error method. To address this problem, an intelligent approach employing adaptive PSO (APSO) for optimum tuning of LQR is proposed. In this approach, an adaptive inertia weight factor (AIWF), which adjusts the inertia weight according to the success rate of the particles, is employed to not only speed up the search process but also to increase the accuracy of the algorithm towards obtaining the optimum controller gain. The performance of the proposed approach is tested on a bench mark inverted pendulum system, and the experimental results of APSO are compared with that of the conventional PSO and GA. Experimental results prove that the proposed algorithm remarkably improves the convergence speed and precision of PSO in obtaining the robust trajectory tracking of inverted pendulum.

Key words: inverted pendulum, LQR controller, particle swarm optimization, genetic algorithm, adaptive inertia weight factor, state feedback control

1. Introduction

Under actuated systems are mechanical control systems which have the number of actuators less than the degrees of freedom. Controller design for a class of under actuated systems is currently an active field of research due to their wide applications in robotics, aerospace and maritime vehicles. The inverted pendulum on a cart system is a class of under actuated mechanical system with one control input (motor voltage) and two outputs (cart position and pendulum angle), so it is a type of single input multiple output (SIMO) system. In this paper, we use pendulum driven cart pole system to investigate the tracking control of

under actuated mechanical systems. Inverted pendulum is a fourth order, unstable, nonlinear, and multivariable system which can be treated as a typical control problem to study various modern control theories. It is a well established benchmark system that provides many challenging problems to control design. According to control purposes of inverted pendulum, the control of inverted pendulum can be divided into three aspects. The first aspect that is widely researched is the swing-up control of inverted pendulum [1-2]. The second aspect is the stabilization of the inverted pendulum [3-4]. The third aspect is tracking control of the inverted pendulum [5-7]. In practice, stabilization and tracking control is more useful for plenty of real time applications. Once the pendulum is swung up to vertical position, the stabilization controller has to maintain the pendulum upright while the cart tracks the position reference trajectory. Two control schemes namely PV control and state feedback control based on LQR technique are implemented for swinging up the pendulum to unstable position and maintaining it in vertical position, respectively. The main focus of this paper is only on the stabilizing controller. The problem of state feedback control design is conventionally handled by pole assignment or Linear Quadratic Regulator (LQR) method via Algebraic Riccati Equation (ARE). However, these methods still suffer from the disadvantage of trial and error approach for parameter tuning. To be more specific, selecting the weighting matrices Q and R of LQR has to be done by trial and error approach [8-9]. So as to address this problem, in this work an intelligent based tuning of state feedback controller gains is implemented using Particle Swarm Optimization (PSO).

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart, is one of the modern heuristic algorithms which combine the social psychology principles in socio-cognition human agents and evolutionary computations. The PSO technique can generate a high quality solution within shorter calculation time than other stochastic methods. In addition, unlike other heuristic optimization methods, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. However, the conventional PSO has a drawback of getting trapped in local optima when solving multimodal tasks. Similarly, when reaching near optimal solution, the PSO algorithm stops optimizing, which makes the accuracy of the algorithm limited. These limitations have imposed constraints on the wider applications of the PSO to real world problems [10]. So to improve the accuracy and convergence speed of the algorithm, in this paper, an adaptive inertial weight factor (AIWF) is incorporated in the velocity update equation of PSO. The main objective of this paper is not only to show the use of evolutionary algorithms for state feedback controller design but also to demonstrate the ability of APSO to obtain optimal control gains with reduced number of iterations through the AIWF.

The rest of the paper is organized as follows. The nominal mathematical model of an inverted pendulum system obtained using Lagrange's equation is presented in Section 2. Control scheme of inverted pendulum is given in Section 3. Genetic algorithm based controller tuning is explained in Section 4. Particle swarm optimization based state feedback controller design is detailed in Section 5. An adaptive PSO which incorporates adaptive weighting strategy is given in Section 6. Experimental results and discussion are given in Section 7, and the paper ends with the concluding remarks in Section 8.

2. Mathematical modeling of inverted pendulum

The linear Self Erecting Single Inverted Pendulum (SESIP) consists of a pendulum system which is attached to a cart equipped with a motor that drives it along a horizontal track. The schematic diagram of inverted pendulum system is shown in Figure 1.

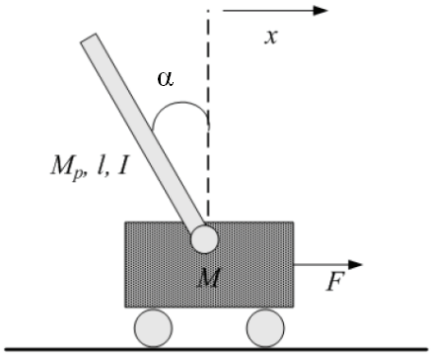


Fig. 1. Schematic diagram of cart-inverted pendulum

The position and velocity of the cart can be changed through the motor while the track restricts the cart movement in the horizontal direction. Encoders are attached to the cart and the pivot in order to measure the cart position and pendulum joint angle, respectively. The block diagram representation of the overall system is shown in Figure 2.

Table 1. List of parameters

Symbol	Description	Value/unit
R	Motor armature resistance	2.6 Ω
L	Motor armature inductance	0.18 mH
K_t	Motor torque constant	0.00767 Nm/A
η_m	Motor efficiency	100%
K_m	Motor EMF constant	0.00767 Ns/rad
J_m	Rotor moment of inertia	3.9×10^{-7} kgm^2
K_g	Gearbox ratio	3.71
η_g	Gearbox efficiency	100%
r_{mp}	Motor pinion radius	6.35×10^{-3} m
r_p	Position pinion radius	1.48×10^{-2} m
B_{eq}	Equivalent viscous damping coefficient at motor	5.4 Nms/rad
B_p	Viscous damping coefficient at pendulum pivot	5.4 Nms/rad
l_p	Pendulum length from pivot to centre of mass	0.3302 m
I	Pendulum moment of inertia	7.88×10^{-3} kgm^2
M_p	Pendulum mass	0.23 kg
M	Cart mass	0.94 kg
V_m	Motor nominal input voltage	5 V

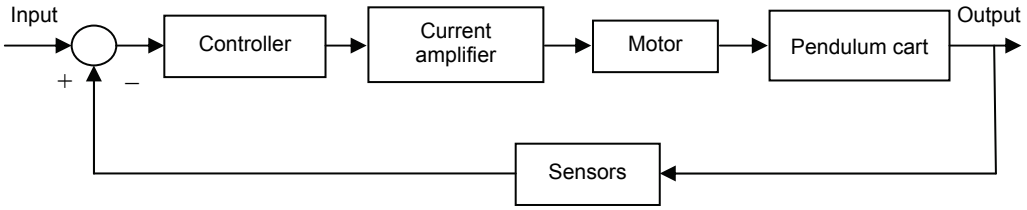


Fig. 2. Block diagram of pendulum cart system

2.1. Single inverted pendulum equation of motion

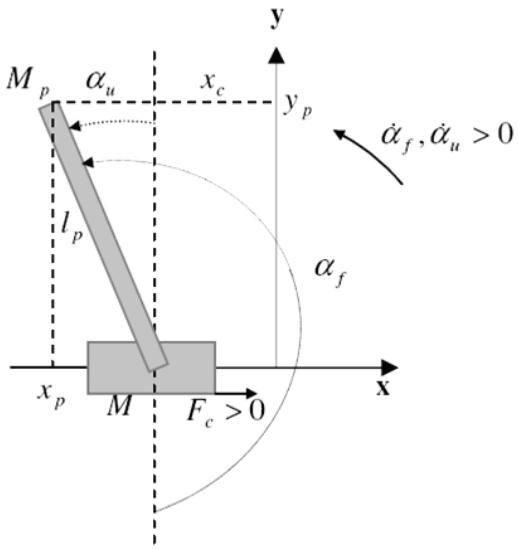


Fig. 3. SESIP schematic diagram

The schematic diagram and angle definitions of SESIP are shown in Figure 3, and the parameters of the system are given in Table 1. The single inverted pendulum (SIP) system is made of a motor cart on top of which pendulum is pivoted. The movement of the cart is constrained only in the horizontal x direction, whereas the pendulum can rotate in the x-y plane. The SIP system has two DOF and can be fully represented using two generalized coordinates such as horizontal displacement of the cart, x_c and rotational displacement of pendulum, α. The nominal plant model is obtained with the assumption that the Coulomb's friction applied to the linear cart and the force on the linear cart are negligible. Using Lagrange's equation, the nonlinear equation of motion can be obtained as

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x}_c}\right] - \frac{\partial L}{\partial x_c} = Q_{x_c} \tag{1}$$

and

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\alpha}_c}\right]-\frac{\partial L}{\partial \alpha}=Q_\alpha\,, \tag{2}$$

with $L = T_T - V_T$, where T_T is total kinetic energy, V_T is total potential energy, Q_{x_c} and Q_α are the generalized forces applied on the coordinate x_c and α, respectively. Both the generalized forces can be defined as follows:

$$Q_{x_c}(t)=F_c(t)-B_{eq}\dot{x}_c \tag{3}$$

and

$$Q_\alpha=B_p\dot{\alpha}(t)\,. \tag{4}$$

This energy is usually caused by its vertical movement from normality (gravitational potential energy) or by a spring sort of displacement. The cart linear motion is horizontal, so the total potential energy is fully represented by the pendulum's gravitational potential energy as characterized below.

$$V_T=M_p g l_p\cos(\alpha(t)). \tag{5}$$

The amount of energy in a system due to its motion is measured by the kinetic energy. Hence, the total kinetic energy can be depicted as follows:

$$T_T=T_c+T_p\,, \tag{6}$$

where T_c and T_p are the translational and rotational kinetic energies arising from both the cart and inverted pendulum, respectively. First, the translational kinetic energy of the motorized cart T_{ct}, is expressed as follows:

$$T_{ct}=\frac{1}{2}M\dot{x}_c^2. \tag{7}$$

Second, the rotational kinetic energy due to DC motor of the cart system, T_{cr}, can be represented by:

$$T_{cr}=\frac{1}{2}\frac{J_m K_g^2\dot{x}_c^2}{r_{mp}^2}. \tag{8}$$

Therefore, the total kinetic energy of the cart can be written as:

$$T_c=\frac{1}{2}M_c\dot{x}_c^2, \tag{9}$$

where

$$M_c=M+\left(J_m\frac{K_g^2}{r_{mp}^2}\right).$$

The total kinetic energy of the pendulum, T_p, is the sum of translational kinetic energy, T_{pt} and rotational kinetic energy, T_{pr}.

$$T_p = T_{pt} + T_{pr} = \frac{1}{2} M_p \dot{r}_p^2 + \frac{1}{2} I_p \dot{\alpha}^2(t), \tag{10}$$

where $\dot{r}_p^2 = \dot{x}_p^2 + \dot{y}_p^2$. From Figure 3, $\dot{x}_p$ and $\dot{y}_p$ can be expressed as:

$$\dot{x}_p = \dot{x}_c - l_p \cos(\alpha(t))\dot{\alpha}(t). \tag{11}$$

$$\dot{y}_p = -l_p \sin(\alpha(t))\dot{\alpha}(t). \tag{12}$$

Substituting (9), (10), (11) and (12) into (6), gives the total kinetic energy, T_T of the system as:

$$T_T = \frac{1}{2}(M_c + M_p)\dot{x}_c^2(t) - M_p l_p \cos(\alpha(t))\dot{\alpha}(t)\dot{x}_c(t) + \frac{1}{2}\left(I_p + M_p I_p^2\right)\dot{\alpha}^2(t). \tag{13}$$

The Lagrangian can be expressed using (5) and (13)

$$\begin{gathered} L = T_T - V_T = \\ = \frac{1}{2}(M_c + M_p)\dot{x}_c^2(t) - M_p l_p \cos(\alpha(t))\dot{\alpha}(t)\dot{x}_c(t) + \frac{1}{2}\left(I_p + M_p I_p^2\right)\dot{\alpha}_c^2(t) - M_p g l_p \cos(\alpha(t)). \end{gathered} \tag{14}$$

From (1) and (2), the pendulum's non linear equation of motion can be obtained as:

$$(M_c + M_p)\ddot{x}_c(t) - B_{eq}\dot{x}_c(t) - \left(M_p l_p \cos(\alpha(t))\right)\ddot{\alpha}(t) + M_p l_p \sin(\alpha(t))\dot{\alpha}^2(t) = F_c(t). \tag{15}$$

and

$$-M_p l_p \cos(\alpha(t))\ddot{x}_c(t) + \left(I_p + M_p I_p^2\right)\ddot{\alpha}(t) + B_p\dot{\alpha}(t) - M_p g l_p \sin(\alpha(t)) = 0. \tag{16}$$

The nonlinear model can be linearized around the equilibrium point (upright pendulum) such that $\sin(\alpha) \cong \alpha$, $\cos(\alpha) \cong 1$, and the higher order terms in the models are also neglected for simplicity. The linearized model is represented in state space in order to design the state feedback controller for upright pendulum stabilization.

$$\begin{gathered} \dot{X} = AX + BU, \\ Y = CX \end{gathered} \tag{17}$$

where $X = [x_c \ \alpha \ \dot{x}_c \ \dot{\alpha}]^T$, $U = V_m$ and $Y = [x_c \ \alpha]^T$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{(M_p l)^2}{q} & \dfrac{-B_{eq}(M_p l^2 + I)}{q} & \dfrac{M_p l B_p}{q} \\ 0 & \dfrac{(M + M_p)M_p g l}{q} & \dfrac{M_p l B_{eq}}{q} & \dfrac{(M + M_p)B_p}{q} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \dfrac{(M_p l^2 + I)}{q} \\ \dfrac{M_p l}{q} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The parameters presented in Table 1 are substituted into (17) to obtain the following state space model.

$$\begin{bmatrix} \dot{x}_c \\ \dot{\alpha} \\ \ddot{x}_c \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2.2643 & -15.8866 & -0.0073 \\ 0 & 27.8203 & -36.6044 & -0.0896 \end{bmatrix} \begin{bmatrix} x_c \\ \alpha \\ \dot{x}_c \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2.2772 \\ 5.2470 \end{bmatrix} u \,, \tag{18}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ \alpha \\ \dot{x}_c \\ \dot{\alpha} \end{bmatrix}. \tag{19}$$

The eigen values of the system matrix are –16.2577, –4.5611, 0, 4.8426. It can be clearly seen that the open loop system has one pole in the Right Half Plane (RHP) i.e., positive pole. Therefore the system is unstable in open loop. As a consequence, in order to maintain the pendulum balanced in the inverted position, a controller is to be designed such that all the resulting closed loop poles lie in the Left Half Plane (LHP).

3. Control scheme of inverted pendulum

The controller design for the inverted pendulum system is broken up into two components. The first part involves the design of a swing up controller that swings the pendulum up to the unstable equilibrium. The second part involves the design of an optimal state feedback controller for the linearized model that will stabilize the pendulum around the upright position. When the pendulum approaches the linearized point, the control will switch to the stabilizing controller which will balance the pendulum around the vertical position.

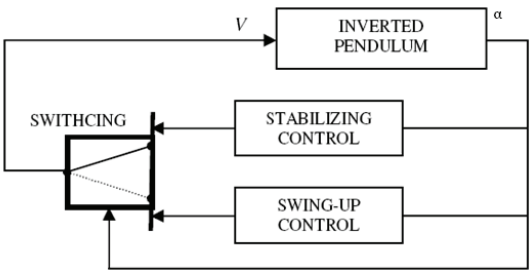


Fig. 4. Control scheme of inverted pendulum

The control scheme of SESIP consists of two main control loops and a decision making logic to switch between the two control schemes. The control scheme of the inverted pendulum is shown in Figure 4. One control loop is a PV controller on the cart position that follows a set point designed to swing up the pendulum from the suspended to the inverted posture. The other control loop is active when the pendulum is around the upright position and

consists of a Linear Quadratic Regulator for maintaining the inverted pendulum in vertical position.

3.1. Problem formulation

Consider a linear time invariant multivariable system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad t \geq 0, x(0) = x_0, \\ y(t) &= Cx(t) + Du(t), \quad t \geq 0, \end{aligned} \tag{20}$$

where $A \in R^{n \times n}$, $B \in R^{n \times m}$, $C \in R^{p \times n}$, $D \in R^{p \times \mathrm{m}}$, are system matrix, input matrix, output matrix and feedforward matrix, respectively. x is the state vector, u is the control input vector, and y is the output vector. The conventional LQR problem is to determine the control input u^* which minimizes the following cost function.

$$J(u) = \int_0^\infty \left[x^T(t) Q\, x(t) + u^T(t) R u(t) \right] dt, \tag{21}$$

where $Q = Q^T$ is a positive semidefinite matrix that penalizes the departure of system states from the equilibrium, and $R = R^T$ is a positive definite matrix that penalizes the control input [11]. The solution of the LQR problem, the optimal control gain K, can be obtained via the following Lagrange multiplier based optimization technique.

$$K = R^{-1} B^T P\,. \tag{22}$$

The optimal state feedback control gain matrix K of LQR can be determined by solving the following Algebraic Riccati Equation (ARE).

$$A^T P + PA + Q - PBR^{-1} B^T P = 0, \tag{23}$$

where $P \in R^{n \times m}$ is a solution of ARE. The weighting matrices Q and R are important components of an LQR optimization process. The compositions of Q and R elements have great influence on system performance. The number of elements of Q and R matrices is dependent on the number of state variable and the number of input variable, respectively. If the weighting matrices are selected as diagonal matrices, the quadratic performance index is simply a weighted integral of the squared error of the states and inputs. Commonly, a trial and error method has been used to construct the matrices Q and R elements [12-13]. This method is cumbersome, time consuming and does not result in optimum performance. To address this problem, in the following section, a bio-inspired algorithm based tuning methods namely GA, PSO and APSO, incorporated in the controller design to obtain the optimum state feedback controller gains, are explained.

4. Genetic algorithm

Genetic algorithm (GA), an evolution method based on natural selection and an evolutionary theory, has been employed to optimize parameters of control system that are difficult

to solve by conventional optimization techniques [14-15]. The working of the GA is based on the Darwinian's theory of survival of the fittest. GA starts with a set of solutions characterized by chromosomes, which are called population. Possible solutions from one population are identified and used to form a new population [16]. The new population is formulated with the motivation that the new one will be better than the previous one. Moreover, solutions are chosen according to their fitness to form new solutions, which are called offspring, and the process is repeated until the terminating condition is satisfied. Figure 5 shows the flow chart of GA for optimizing the controller gains.

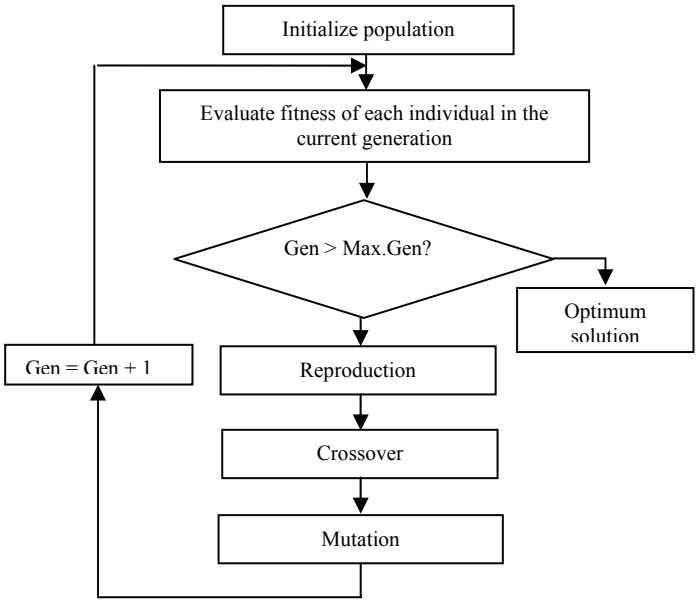


Fig. 5. Flowchart of GA

The first step in finding the optimum controller gains using GA is to initialize the range of diagonal elements of Q and R matrices. In this case, $Q = \text{diag}(q_{11}, q_{22}, 0, 0)$ and the ranges of elements are $q_{11} \in [10, 20]$, $q_{22} \in [10, 20]$. The ranges of q_{11} and q_{22} are chosen in such a way that the resultant cart movement lies within the track length, which is 0.3 m. After selecting the ranges of weighting matrices, the next step is to produce the initial population. Two components namely size of the population and selection of the initial populations are involved in this step. The size of the population is very important in the optimization process because it can affect the end result. The performance of the GA can be improved by increasing the size of the population. But the high value of population costs the high computation time in obtaining the optimized result. In this work, the size of the population is chosen as 100. The next step is to evaluate the fitness function of each chromosome. The individual with lower fitness is discarded and the one with higher fitness is chosen, and it becomes the new individual in the next generation. The arithmetic cross over is chosen for this example and the new individual is created by linear operations of the old individuals. Let the two previous individuals be q_1 and

q_2 and the new individual be q_1^{t+1} and q_2^{t+1}. The following operations are used to select the new individuals.

$$q_1^{t+1} = \delta q_2^t + (1-\delta) q_1^t,$$
$$q_2^{t+1} = \delta q_1^t + (1-\delta) q_2^t, \tag{24}$$

where δ is a constant, and it is set to 0.4 empirically. Finally, the aberrance operation is implemented to search the results randomly and maintain the diversity of the group. The main intentions of this operation are to add the diversity of swarm and to avoid the immature convergence. Once the aberrance operation is complete, a new population is formed, and the new population repeats the above process until the stopping condition, which is number of generations, is reached and the optimized gains are used to design the state feedback controller.

5. Standard particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation technique, developed by Russell Eberhart and James Kennedy [17-18] in 1995, and was inspired by the social behavior of bird flocking and fish schooling. PSO has its roots in artificial life and social psychology as well as in engineering and computer science. It utilizes a "swarm" of the particles that "fly" through the problem hyperspace with given velocities. The computational flow chart of PSO is shown in Figure 6. During each iteration, the velocities of the individual particles are stochastically adjusted according to the historical best position for the particle itself and the neighborhood best position. Both the particle best and the neighborhood best are derived according to the user defined fitness function [19]. The movement of each particle naturally evolves to an optimal or near-optimal solution. The word "swarm" comes from the irregular movements of the particles in the problem space, now more similar to a swarm of mosquitoes rather than a flock of birds or a school of fish [20-21]. PSO is a computational intelligence based technique that is not largely affected by the size and the nonlinearity of the problem, and it can converge to the optimal solution in many problems where most analytical methods fail to converge. Moreover, PSO has some advantages over other similar optimization techniques such as Genetic algorithm (GA), namely:

a) PSO is easier to implement and there are fewer parameters to adjust.
b) In PSO, every particle remembers its own previous best value as well as the neighborhood best; therefore, it has a more effective memory capability than the GA.
c) PSO is more efficient in maintaining the diversity of the swarm [22] (more similar to the ideal social interaction in a community), since all the particles use the information related to the most successful particle in order to improve themselves, whereas in GA, the probability of good solutions to take part in the new population is high compare to the worse ones; therefore, in GA the population evolves around a subset of the best individuals.

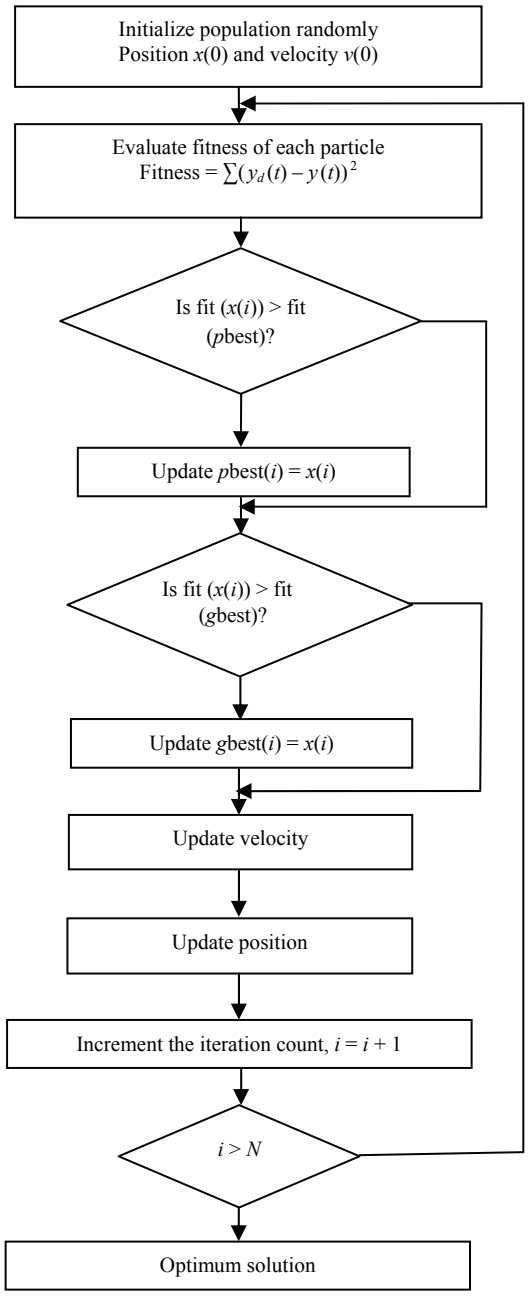


Fig. 6. Flow chart of PSO

The position vector x and the velocity vector v of n particles in the D dimensional search space can be represented respectively as

$$x = \left[x_1^d, x_2^d \ldots x_i^d \ldots x_n^d\right], i \in \{1, 2, \ldots n\},\ \ d \in \{1, 2, \ldots D\}, \tag{25}$$

$$v = \left[v_1^d, v_2^d \ldots v_i^d \ldots v_n^d\right], i \in \{1, 2, \ldots n\},\ \ d \in \{1, 2, \ldots D\}, \tag{26}$$

According to a user defined fitness function, the best position pbest of each particle and the best of the fittest particle gbest found so far can be represented respectively as

$$\text{pbest} = \text{pbest}_i^d, i \in \{1, 2, \ldots n\},\ \ d \in \{1, 2, \ldots D\}, \tag{27}$$

$$\text{gbest} = \text{gbest}_g^d, i \in \{1, 2, \ldots n\},\ \ d \in \{1, 2, \ldots D\}, \tag{28}$$

Then, the new velocities and the positions of the particles for the next fitness evaluation are calculated using the following two equations.

$$v_i^d(t+1) = w * v_i^d(t) + c_1 * rand_1\left(p\text{best}_i^d(t) - x_i^d(t)\right) + c_2 * rand_2 * \left(p\text{best}_g^d(t) - x_i^d(t)\right), \tag{29}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1), \tag{30}$$

where w is the inertia weight factor, c_1 and c_2 are constants known as acceleration coefficients, which are two separately generated and uniformly distributed random numbers. Motivated by the desire to better control the scope of the search, Shi and Eberhart [23] have observed that the optimal solution can be improved by varying the value of the inertia weight w from 0.9 at the beginning of the search to 0.4 at the end of the search for most problems. In literatures, mostly two stopping criteria are applied for single objective optimization. One is based on the error measure and the other one is based on the number of function evaluations (number of iterations). In the first method, the optimum value should be known prior to the execution, which is not possible in real world problems. The second method is highly dependent on the fitness function. Basically, no correlation can be visualized between an optimization problem and the required number of iterations, and it has to be usually computed by trial and error methods. However, improper choice of the number of iterations to stop the optimization can result in either premature convergence or high computational effort. The detailed review of several stopping criteria is given in [24]. In this work, we adopt distribution based stopping criterion which considers the diversity of the population. If the diversity is minimum, the individuals are close to each other. Thus, it can be assumed that the convergence has been obtained. In each dimension, standard deviation of the best positions is checked, and if it is less than the threshold for sufficiently large number of iterations, the optimization algorithm will be terminated.

5.1. Fitness function

The convergence of the optimization algorithms towards the global optimal solution is characterized by the fitness function. Some of the commonly used fitness functions to design controllers are integral of the absolute error (IAE), integral of the square error (ISE) and integral of the time weighted absolute error (ITAE). In the present study, the ISE as given in (31) is chosen as fitness function.

$$f = ISE = \sum [y_d(t) - y(t)]^2 = \sum e^2(t). \tag{31}$$

The problem of optimization is to find the state feedback controller gain matrix $K = [k_1\ k_2\ k_3\ k_4]$ such that the fitness function given in (31) is minimized. Moreover, the values of controller gain matrix are computed in such a way that the resultant cost function (J) is minimized to obtain an optimal performance between amount of control input and speed of response of the system.

6. Adaptive particle swarm optimization

In the last decade, several variants of PSO have been proposed to improve the performance of PSO. All the proposed modifications are mainly to improve the exploration, exploitation or even both exploration and exploitation ability of the PSO [25]. One of the modifications introduced in the PSO is the use of inertia weight parameter in the velocity update equation [26]. The contribution rate of a particle's previous velocity to its current velocity is determined by the inertia weight. The crucial aspect in PSO to find the optimal solution is the proper control of global exploration and local exploitation [27]. Equation (29), indicates that the velocity of the current particle is influenced by the previous velocity which gives the necessary momentum for particles to travel across the search space. The impact of the previous velocity depends on the inertia weight factor w, which dictates the balance between global exploration and local search exploitation in PSO. Therefore, proper control of the inertia weight is significant in finding the optimum solution. The higher value of inertia weight focuses more on global exploration, while the smaller inertia weight focuses highly on fine tuning the current search area.

To implement an adaptive inertia weight strategy, it is important to evaluate the situation of the swarm at each iteration. For this purpose, the percentage of success of the particles is used to update the inertia weight adaptively. A low percentage success indicates that the particles are moving around the optimum without much improvement, whereas a high percentage success implies that the particles have converged to a point which is far from the optimum value and the whole swarm is slowly moving towards the optimum point. The structure of the modified PSO which incorporates the adaptive inertia weight to obtain the optimum solution is shown in the form of flow chart in Figure 7. The success of particle i at iteration t is given as:

$$S(i,t) = \begin{cases} 1 & \text{if } p\text{best}_i^t > p\text{best}_i^{t-1} \\ 0 & \text{if } p\text{best}_i^t \geq \text{if } p\text{best}_i^{t-1} \end{cases} \tag{32}$$

where $p\text{best}_i^t$ is the best position found by particle i till iteration t. The success percentage of the swarm is calculated using the success values of the particles as given below.

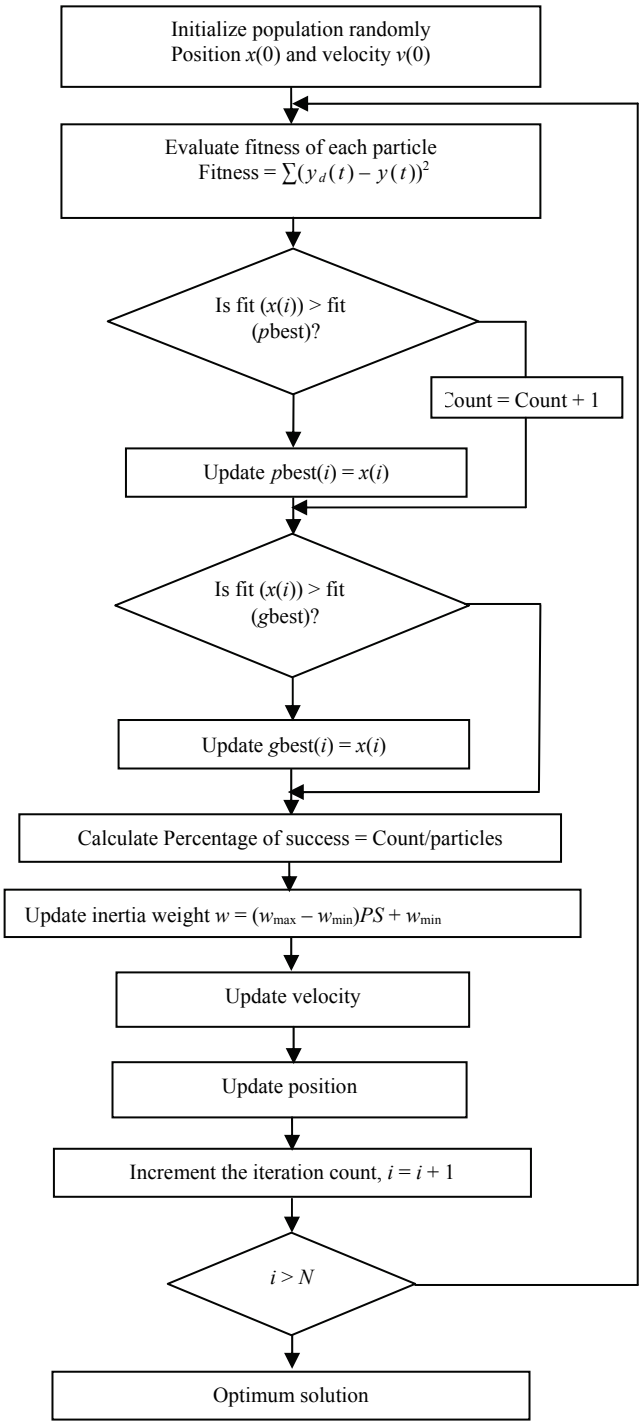


Fig. 7. Flow chart of APSO

$$K = [k_1 \ k_2 \ k_3 \ k_4] \, PS(t) = \frac{\sum_{i=1}^{n} S(i,t)}{n}, \tag{33}$$

where n represents the number of particles and $PS \in [0, 1]$ indicates the percentage of the particles whose fitness value had improved during the previous iteration. In order to achieve the balance between exploration and exploitation, the reasonable choice of w proportional to PS is to be made. The following linear function which is used to map the values of PS to the possible range of inertia weights is used to initialize the inertia weight adaptively.

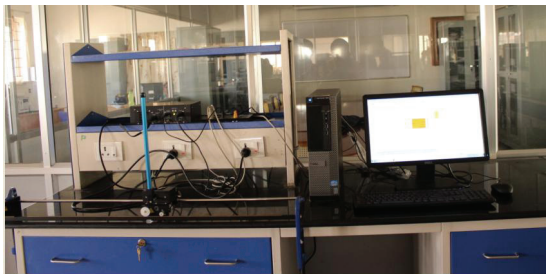
$$w = (w_{\max} - w_{\min}) PS + w_{\min}, \tag{34}$$

where $w_{\max}$ and $w_{\min}$ represent the maximum and minimum of w respectively. The range of the inertia weight is chosen to be [0, 1]. Since the value of PS is in the range of [0, 1], the value of w can be in any suitable range. Thus, the inclusion of adaptive inertia weight protects the particles whose objective values are low and disrupts the particles whose objective values are greater than the average value.

7. Experimental results and discussion

The implementation of optimization algorithms, namely GA, PSO and APSO are facilitated by MATLAB 2009. In order to show the effectiveness of the proposed scheme, experiments are conducted using Quanser IP-02 inverted pendulum system. The snapshot of the experimental setup is shown in Figure 8. Real time experimental setup consists of a computer with MATLAB, Simulink, Q8 data acquisition board and Quanser IP02 Linear inverted pendulum module. Some hardware limitations are considered in the controller design for the pendulum system. The Digital-to-Analog voltage converter in data acquisition board is limited between –10 V and 10 V. The safety watchdog is turned on when the cart reaches the maximum displacement of 0.35 m from the centre of the track, and the control process is aborted when the pendulum or cart touches the limit switch.

Fig. 8. Snapshot of experimental setup



7.1. Performance assessment of optimization methods

A number of parameters are to be specified while applying PSO and GA for solving optimization problems. The proper choice of parameter values influences the convergence

speed and accuracy of the algorithms. Table 2 shows the values used for GA, PSO and APSO optimization methods. In both PSO and APSO, the population size is taken as 100, the cognitive acceleration constant and social acceleration constant are set as $c_1 = c_2 = 1.3$, and v_{max} is fixed to be 20% of the search space. The standard PSO uses a linearly varying inertia weight over the generations, which varies from 1.1 to 0.4, whereas the APSO makes use of the AIWF specified in (34) with $w_{min} = 0.3$ and $w_{max} = 1.0$. The parameters used in GA are population size = 100 chromosomes, cross over rate = 0.96, mutation rate = 0.1, search interval = [0, 20] and generation number = 400. The search interval is chosen in accordance with the range of Q which will result in trade off between the control effort and improved tracking response. The corresponding solution of Riccati equation, state feedback controller gains, and the closed loop eigen values obtained by three optimization methods are given in Table 3.

Table 2. Parameters used for GA, PSO and APSO

GA Parameters	**PSO Parameters**	**APSO Parameters**
Population size: 100	Swarm size: 100	Swarm size: 100
Number of generations: 400	Number of generations: 400	Number of generations: 400
Type of selection: normal geometric	Cognitive acceleration constant: $c_1 = 1.3$	Cognitive acceleration: constant $c_1 = 1.3$
Type of cross over: arithmetic	Social acceleration constant: $c_2 = 1.3$	Social acceleration constant: $c_2 = 1.3$
Type of mutation: non uniform	Inertia weight $w_{min} = 0.4$, $w_{max} = 1.1$	Inertia weight $w_{min} = 0.3$, $w_{max} = 1.0$

The optimization process is repeated for 15 times, and the results are recorded to perform statistical analysis. Table 4 shows the corresponding minimum, maximum, average and standard deviation of the fitness function for all the three tuning methods. The minimum value of the fitness function obtained via APSO is the least among all three optimization techniques, which proves that the precision of standard PSO has been improved with the introduction of AIWF into the algorithm. Furthermore, the standard deviation of the fitness function obtained via APSO is the least of all three methods, which suggests that the APSO can provide consistent global optimal value. From Figure 9, convergence curve of the optimization methods, it can be observed that the fitness function of the APSO descend faster than the other two methods. The final value of the fitness function minimized via APSO algorithm is lesser than that of the PSO, which proves that the performance of APSO is superior to that of PSO towards obtaining the optimum controller gains in reduced number of iterations.

Figure 10 shows the closed loop eigen value plot of the optimization methods. The closed loop eigen values of the system tuned via APSO are located comparatively far from the imaginary axis, which suggest that the stability of the system is improved. Dynamic performance indices such as rise time, settling time and overshoot are chosen to evaluate the performance of cart position response. Figures 11, 12, and 13 show the response of tuning methods for control input, pendulum angle and cart position, respectively. Based on the performance indices given in Table 5 and Table 6, it can be noted that the LQR controller tuned using

Table 3. State Feedback Controller gains and closed loop eigen values obtained via GA, PSO and APSO

Optimi-zation method	ARE Solution (P)	Feedback gain matrix (K)	Closed loop eigen values
GA	$\begin{bmatrix} 5.758 & -2.905 & 1.769 & -0462 \\ -2.905 & 3.166 & -1.364 & 0.434 \\ 1.769 & -1.364 & 0.803 & -0.226 \\ -0462 & 0.434 & -0.226 & 0.068 \end{bmatrix}$	[–63.245 159.130 –49.322 18.545]	–8.801 + 4.69*i* –8.801 – 4.69*i* –2.278 + 1.434*i* –2.278 – 1.434*i*
PSO	$\begin{bmatrix} 1.389 & -0.641 & 0.475 & -0.117 \\ -0.641 & 0.757 & -0.320 & 0.094 \\ 0.475 & -0.320 & 0.261 & -0.066 \\ -0.117 & 0.094 & -0.066 & 0.017 \end{bmatrix}$	[–42.426 93.588 –32.544 12.680]	–10.197 + 6.983*i* –10.197 – 6.983*i* –2.652 + 1.916*i* –2.652 – 1.916*i*
APSO	$\begin{bmatrix} 2.857 & -1.823 & 1.020 & -0.283 \\ -1.823 & 3.211 & -1.185 & 0.440 \\ 1.020 & -1.185 & 0.261 & -0.187 \\ -0.283 & 0.440 & -0.187 & 0.066 \end{bmatrix}$	[–18.708 65.298 –20.674 9.039]	–14.828 + 12.743*i* –14.828 + 12.743*i* –1.932 + 1.625*i* –1.932 + 1.625*i*

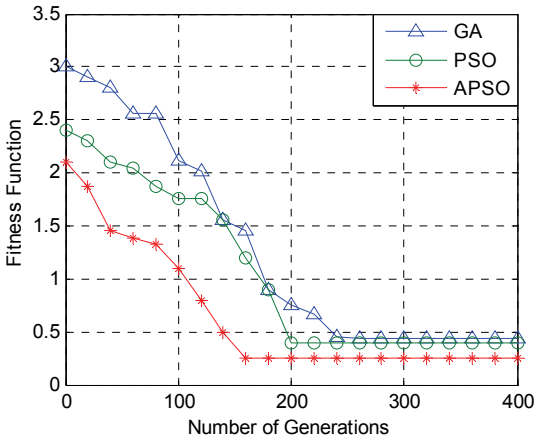


Fig. 9. Fitness function of optimization methods

Table 4. Statistical performance of optimization methods

Optimization method	Minimum *f*	Mean *f*	Maximum *f*	Standard deviation of *f*
GA	0.435	0.568	3.127	0.156
PSO	0.398	0.456	2.412	0.068
APSO	0.258	0.356	2.133	0.024

APSO has less rise time and reaches the set point quickly compare to that of conventional PSO and GA. The response of APSO based LQR is also characterized by a reduced overshoot

and short delay time. In summary, for dynamic response, the inverted pendulum controlled by APSO-LQR controller not only balances faster because of the shorter settling time but also has better robustness due to reduced maximum overshoot. The above points substantiate for the fact that the LQR controller tuned via APSO can guarantee the inverted pendulum system a better dynamic performance than the other two conventional evolutionary tuning methods PSO and GA.

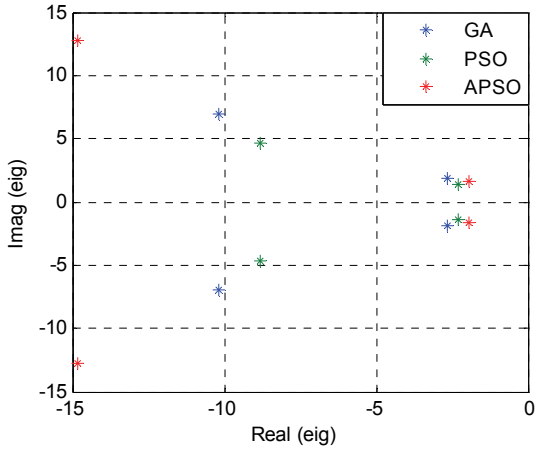


Fig. 10. Eigen values distribution

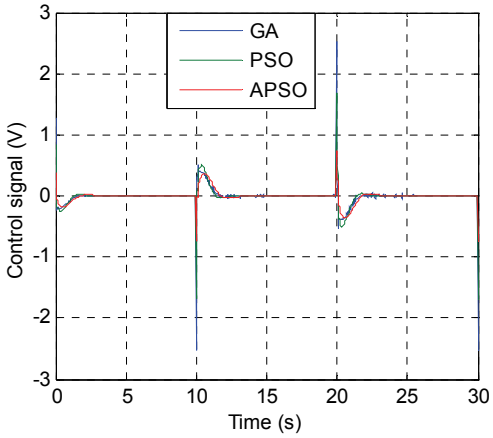


Fig. 11. Control Signal applied to cart

Table 5. Dynamic performance of pendulum angle and control input

Optimization method	Pendulum angle oscillation amplitude (deg)	Control signal oscillation amplitude (volts)
GA	1.8	0.65
PSO	0.9	0.53
APSO	0.4	0.43

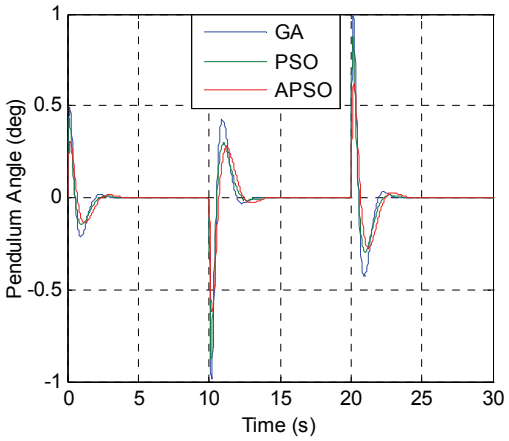


Fig. 12. Pendulum angle

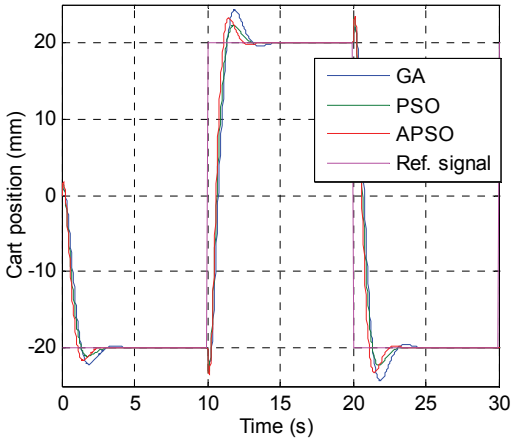


Fig. 13. Cart position response

Table 6. Comparison of cart position response

Optimization method	**Time domain parameters**			**Performance index**
	t_r	t_s	$\%M_p$	**ISE**
GA	2.6	5.1	15	0.435
PSO	2.0	2.9	11	0.398
APSO	1.5	2.7	8	0.258

8. Conclusions

In this paper, a modified PSO, which uses an adaptive inertia weight in the velocity update equation to accelerate the searching technique, has been employed to obtain the optimum state feedback controller gains of LQR which is used to stabilize the inverted pendulum in upright

position. In general, the weighting matrices of LQR are chosen based on trial and error approach, which makes the tuning process not only tedious but also cumbersome. Employing evolutionary algorithms can be considered as an effective method to address the above problem of obtaining the optimum state feedback controller gains of LQR. In addition to the proposed method, within the proposed LQR controller framework, two evolutionary based tuning methods have been employed and the performance of the optimization methods are evaluated on a bench mark inverted pendulum system. Instead of using the linearly varying inertia weight in the conventional PSO, in APSO inertia weights are adaptively varied according to the success rate of the particles to increase both the search process and accuracy of standard PSO. The improved convergence results and dynamic performance of cart position accentuate that the performance of APSO is better than that of PSO and GA. The statistical measures calculated for all the three tuning methods suggest that the APSO can provide faster and consistent response.

References

[1] Mason P., Broucke M., Piccoli M., *Time optimal swing-up of the planar pendulum.* IEEE Transactions on Automatic Control 53(8): 1876-1886 (2008).
[2] Tao C.W., Taur J.S., Hsieh T.W., Tsai C.L., *Design of a fuzzy controller with fuzzy swing-up and parallel distributed pole assignment schemes for an inverted pendulum and cart system*. IEEE Transactions on Control Systems Technology 16(6): 1277-1288 (2008).
[3] Jia-Jun Wang, *Simulation studies of inverted pendulum based on PID controllers*, Simulation Modelling Practice and Theory 19(2): 440-449 (2011).
[4] Nenad Muskinja, Boris Tovornik, *Swinging Up and Stabilization of a Real Inverted Pendulum*, IEEE Transactions on Industrial Electronics 53(2): 631-639 (2006).
[5] Wai R.J., Chang L.J., *Adaptive stabilizing and tracking control for a nonlinear inverted-pendulum system via sliding-mode technique.* IEEE Transactions on Industrial Electronics 53(2): 674-692 (2007).
[6] Chang L.H., Lee A.C., *Design of nonlinear controller for bi-axial inverted pendulum system*. IET Control Theory and Application 1(4): 979-986 (2007).
[7] Shahnazi R., Akbarzadeh T.M.R., *PI adaptive fuzzy control with large and fast disturbance rejection for a class of uncertain nonlinear systems*, IEEE Transactions on Fuzzy Systems 16(1), 187-197 (2008).
[8] Solihin M.I., Akmeliawati R., *Particle SwamOptimization for Stabilizing Controller of a Self-erecting Linear Inverted Pendulum.* International Journal of Electrical and Electronic Systems Research 3: 410-415 (2010).
[9] Saifizul A.A., Zainon A N.A.B., Osman N.A.B. et al., *Intelligent Control for Self-erecting Inverted Pendulum Via Adaptive Neuro-fuzzy Inference System.* American Journal of Applied Sciences 3(4): 1795-1802 (2006).
[10] Xueming Yang, Jinsha Yuan, Jiangye Yuan, Huina Mao, *A modified particle swarm optimizer with dynamic adaptation.* Applied Mathematics and Computation 189(2): 1205-1213 (2007).
[11] Amin Hasanzadeha, Chris S. Edrington, Hossein Mokhtari, *Optimal tuning of linear controllers for power electronics/power systems applications.* Electric Power Systems Research 81: 2188-2197 (2011).
[12] Aamir H.O.A, Martino O.A., Matthew W.D., *New Approach for Position Control of Induction motor.* 45th Universities Power Engineering Conference (2010).
[13] Desineni S.N., *Optimal Control Systems.* CRC press (2003).

[14] Robandia I., Nishimori K., Nishimura R., Ishihara N., *Optimal feedback control design using genetic algorithm in multimachine power system.* Electrical Power and Energy Systems 23: 263-271 (2001).
[15] Li Jimin, Shang Chaoxuan, Zou Minghu, *Parameter Optimization of Linear Quadratic Controller Based on Genetic Algorithm.* TSINGHUA Science and Technology 12(51): 208-211 (2007).
[16] Rahul Malhotra, Narinder Singh, Yaduvir Singh, *Genetic Algorithms: Concepts, Design for Optimization of Process Controllers.* Computer and Information Science 4(2): 39-54 (2011).
[17] Kennedy J., Eberhart R., *Particle swarm optimization.* Proceedings of the IEEE International Conference on Neural Networks (ICNN) 4: 1942-1948 (1995).
[18] Eberhart R., Kennedy J., *A new optimizer using particle swarm theory.* Proceedings of 6th International Symposium on Micro Machine and Human Science (MHS), pp. 39-43 (1995).
[19] Venayagamoorthy G.K., Harley R.G., *Swarm Intelligence for Transmission System Control.* Proceedings of the IEEE Conference on Power Engineering (2007).
[20] Eberhart R., Shi Y., Kennedy J., *Swarm intelligence*. San Francisco, CA: Morgan Kaufmann, (2001).
[21] Che-Cheng Chang, Jichiang Tsai and Shi-Jia Pei, *A Quantum PSO Algorithm for Feedback Control of Semi-Autonomous Driver Assistance Systems.* 12th International Conference on ITS Telecommunications (2012).
[22] Engelbrecht A.P., *Particle swarm optimization: where does it belong?* Proc. of the IEEE Swarm Intelligence Symposium (SIS'06): 48-54 (2006).
[23] Shi Y.H., Eberhart R.C., *Parameter selection in particle swarm optimization.* Annual Conference on Evolutionary Programming, San Diego (1998).
[24] Zielinski K., Laur R., *Stopping criteria for a constrained single-objective particle swarm optimization algorithm*. Informatica, pp. 31: 51-59 (2007).
[25] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh, Reza Safabakhsh, *A novel particle swarm optimization algorithm with adaptive inertia weight*. Applied Soft Computing 11: 3658-3670 (2011).
[26] Bo L., Ling W., Yi-Hui J. et al., *Improved Particle Swarm Optimization combined with Chaos.* Chaos, Solitons and Fractals 25: 1261-271 (2005).
[27] Sidhartha Panda, Narayana Prasad Padhy, *Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design*. Applied Soft Computing 8: 1418-1427 (2008).