Journal of
Applied Sciences

PSO-Based Controller Design for Rotary Inverted Pendulum System

Iraj Hassanzadeh and Saleh Mobayen
Research Laboratory of Robotics, Department of Control Engineering,
Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

Abstract: This research presents an optimum approach for designing Rotary Inverted Pendulum (RIP) controller using PSO algorithm. The primary design goal is to balance the pendulum in an inverted position and the control criterion is to minimize the integral absolute error of system angles. Simulation results demonstrate the robustness and effectiveness of proposed controller with regard to parameter variations, various reference trajectory and load disturbances. The proposed method can be considered as a promising way for control of various similar nonlinear systems.

Key words: Rotary inverted pendulum, particle swarm optimization, genetic algorithm, nonlinear model

INTRODUCTION

During the past decades, many modern control methodologies such as nonlinear control, optimal control, adaptive control and variable structure control have been widely proposed for control approaches (Gaing, 2004). However, these methods are theoretically complex and difficult to implement. PID controller design covering treatment to both transient and steady state responses offers the simplest and most efficient solutions to various control problems (Visioli, 2001). Unfortunately, it has been difficult to tune PID controller gains accurately because many industrial plants are often very complex consisting of issues such as higher order, time delays and nonlinearities (Kwok *et al.*, 1993; Gaing, 2004). The ability of using numerical methods for efficiently and accurately characterizing the quality of a particular design has excited control engineers to apply stochastic global optimizers. Over the past years, several heuristic methods are employed for tuning of controllers. Ziegler and Nichols proposed the first method utilizing the classical tuning rules. Though, it is hard to determine optimal PID controller parameters with Ziegler-Nichols formula in general (Visioli, 2001; Gaing, 2004).

To overcome these difficulties, various methods are employed. Many random search methods, such as Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA) have recently received great attention for searching global optimal solution and achieving high efficiency (Zhou and Birdwell, 1994; Haupt and Haupt, 1998). GA method is usually faster than TS and SA methods because of employing parallel search techniques.

Though, the GA method has been employed successfully for global optimization, recent research has identified some deficiencies in GA performance. This degradation in efficiency is apparent in applications with highly epistatic objective functions (where the parameters being optimized are highly correlated), the crossover and mutation operations cannot ensure better fitness of offspring because population chromosomes have similar structure and their fitness are high toward the end of the process (Gaing, 2004; Kennedy and Eberhart, 1995). To overcome GA difficulties, a novel method is proposed using PSO approach. PSO is one of the modern heuristic algorithms developed through simulation of a simplified social system. Generally, it is characterized as a simple concept, easy to implement and computationally efficient. Because PSO method is a superior optimization technique than recent heuristic methods, in this study developments of PSO-PID controller to determine optimal PID parameters are considered.

ROTARY INVERTED PENDULUM SYSTEM

The rotary inverted pendulum system is a well-known test platform for evaluating various control algorithms. It has also some significant real life applications such as pointing control, aerospace vehicles control, robotics, etc (Muskinja and Tovornik, 2006). The system consists of a rotary arm and a pendulum where the rotary arm is actuated by a motor with the objective of balancing the pendulum in an inverted position. A schematic diagram of the RIP system is shown in Fig. 1, where u, l_p, m_p, α, r, θ and J_b are the motor input, the pendulum length, the
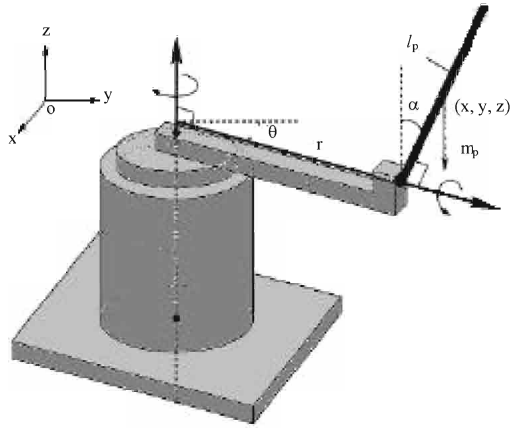
Corresponding Author: Iraj Hassanzadeh, Research Laboratory of Robotics, Department of Control Engineering,
Faculty of Electrical and Computer Engineering, University of Tabriz, Postal Code 5166616471, Tabriz,
Iran Tel: +98(411) 339 3728 Fax: +98(411) 330 0819

Fig. 1: Schematic view of RIP system
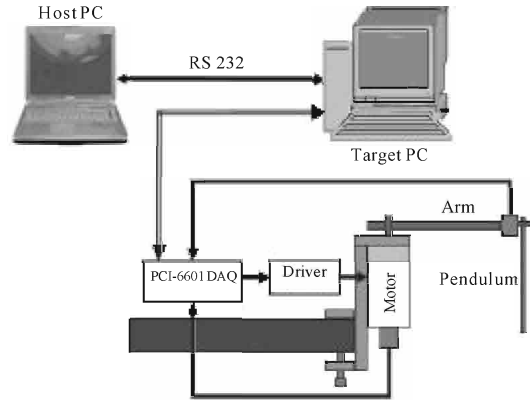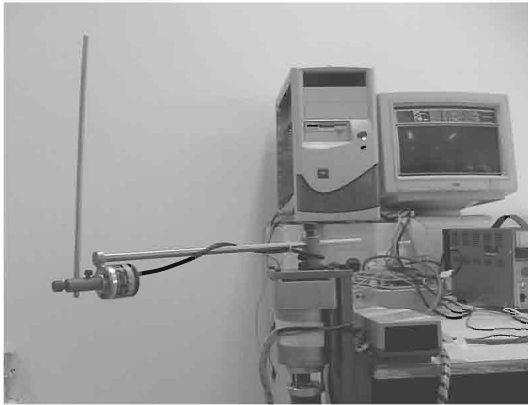
Fig. 2: Built in RIP system (Advanced robotics research lab)

Fig. 3: Block diagram of whole system

Table 1: Parameters of the RIP system

Parameters	Values
a	3.2900
b	0.1252
c	0.2369
d	6.0520
e	0.0132
f	14.2830
g	1.4286
h	1.7200
i	141.3200
w	0.0012

$$\begin{bmatrix} a+b\sin^2\alpha & c\cos\alpha \\ c\cos\alpha & b \end{bmatrix}\begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} +$$

$$\begin{bmatrix} f+b(\sin 2\alpha)\dot{\alpha} & -c(\sin\alpha)\dot{\alpha} \\ -\dfrac{b}{2}(\sin 2\alpha)\dot{\theta} & e \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} g\,\mathrm{sgn}(\dot{\theta})+h.\theta \\ -d\sin\alpha \end{bmatrix} = \begin{bmatrix} i.u \\ 0 \end{bmatrix} \quad (3)$$

pendulum mass, the pendulum angle, the arm length, the arm angle and effective mass moment of inertia, respectively.

The plane of the pendulum is orthogonal to the radial arm. Figure 2 shows the RIP system built in robotics research lab in our department. Also, the block diagram of whole system is shown in Fig. 3.

Here, the dynamic equations of the RIP system considering backlash and friction effects are presented. The RIP dynamics are governed by Yan (2003) and Muskinja and Tovornik (2006):

$$(a+b\sin^2\alpha)\ddot{\theta}+(c\cos\alpha)\ddot{\alpha}-(c\sin\alpha)(\dot{\alpha})^2+$$
$$(2b\sin\alpha\cos\alpha)\dot{\alpha}\,\dot{\theta}+f\dot{\theta}+g\,\mathrm{sgn}(\dot{\theta})+h.\theta=i.u \quad (1)$$

$$b\ddot{\alpha}+(c\cos\alpha)\ddot{\theta}-(b\sin\alpha\cos\alpha)(\dot{\theta})^2-d\sin\alpha+e\dot{\alpha}=0 \quad (2)$$

The above nonlinear model can be found in the following equations:

The parameters of nonlinear model of the system are shown in Table 1.

From Eq. 3, the corresponding nonlinear model is given by:

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = A\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + Bu + E \quad (4)$$

where, A, B and E matrices are as follows:

$$A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}=\frac{1}{\Delta}\begin{bmatrix} -1.788-\sin 2\alpha(0.0156\dot{\alpha}+0.0148\cos\alpha.\dot{\theta}) \\ 3.38\cos\alpha+\sin 2\alpha(0.029\cos\alpha.\dot{\alpha}+(0.2+0.008\sin^2\alpha)\dot{\theta}) \end{bmatrix}$$
$$\begin{matrix} 0.0296\sin\alpha.\dot{\alpha}+0.003\cos\alpha \\ 0.028\sin 2\alpha.\dot{\alpha}-0.043-0.0016\sin^2\alpha \end{matrix} \quad (5)$$

$$B=\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}=\frac{1}{\Delta}\begin{bmatrix} -(0.179\,\mathrm{sgn}(\dot{\theta})+0.215\theta+0.717\sin 2\alpha) \\ 0.338\cos\alpha.\mathrm{sgn}(\dot{\theta})+0.4075\theta.\cos\alpha+19.92\sin\alpha+0.758\sin^3\alpha \end{bmatrix}$$
$$\quad (6)$$

$$E = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} 17.693 \\ -33.478\cos\alpha \end{bmatrix} \qquad (7)$$

with

$$\Delta = 0.412 + 0.0156\sin^2\alpha - 0.0561\cos^2\alpha \qquad (8)$$

From the above Equations, the RIP system is easily simulated using Simulink® and Matlab®. The controller parameters generated by PSO algorithm are employed iteratively in relevant simulation blocks and the cost function is calculated in the manner presented in next section.

PARTICLE SWARM OPTIMIZATION

Considering the social behavior of swarm of fish, bees and other animals, the concept of PSO is developed. PSO is a robust stochastic evolutionary computation method based on the movement of swarms looking for the most fertile feeding location (Rahmat-Samii, 2003). In general, PSO implementation is easier than GA. Indeed, PSO only has one operator; velocity calculation, so the computation time is decreased significantly. The reason is PSO does not perform the selection and crossover operations in evolutionary process.

Another difference between GA and PSO is the ability to control convergence. Crossover and mutation rates can affect the convergence of GA, but nothing can compare to the level of control achieved through manipulating of the inertial weight. The more decrease of inertial weight the more increase the swarm's convergence. This type of control allows determining the rate of convergence and the level of stagnation eventually achieved. Stagnation occurs in GA when all of the individuals have the same genetic code. In that case the gene pool is uniform, crossover has little or no effect on population and each successive generation is essentially same as the first. However, in the PSO, this effect can be controlled or prevented (Kennedy and Eberhart, 1995; Rahmat-Samii, 2003).

All solutions in PSO 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 the most optimization techniques, PSO requires a fitness evaluation function relevant to the particle's position. X_{PB} and X_{GB} are the personal best (P_{best}) position and global best (G_{best}) position of the ith 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 based on the following equation (Gaing, 2004):

$$V_i(\text{new}) = w \times V_i(\text{old}) + c_1 \times \text{rand}() \times (X_{PB} - X_i) + \\ c_2 \times \text{Rand}() \times (X_{GB} - X_i) \qquad (9)$$

Here, rand() and Rand() are two random numbers in the range [0,1]; c_1 and c_2 are the acceleration constants and w is the inertia weight factor. The parameter w helps the particles converge to G_{best}, rather than oscillating around it. Suitable selection of w provides a balance between global and local explorations. In general, w is set according to the following equation (Rahmat-Samii, 2003):

$$w = 0.5(1 + \text{rand}(0,1)) \qquad (10)$$

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

$$X_i = X_i + V_i \times \Delta_t \qquad (11)$$

where, Δ_t usually is set to 1. Then the fitness at each position is reevaluated. If any fitness is greater than G_{best}, then the new position becomes G_{best} and the particles are accelerated toward the new point. If the particle's fitness value is greater than P_{best}, then P_{best} is replaced by the current position. The flowchart of PSO-PID controller design procedure is shown in Fig. 4. PSO algorithm parameters are set based on trial and error as follows:

No. of particles for each controller = 30
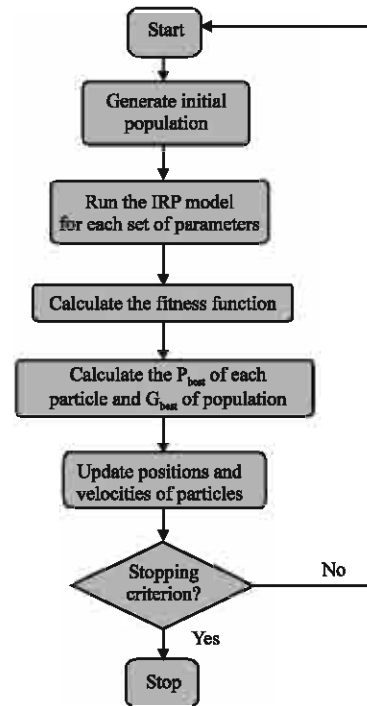Acceleration constants $c_1 = c_2 = 1.5$
Maximum generation = 20

Fig. 4: Flowchart of PSO-PID controller design procedure

PROBLEM FORMULATION AND CONTROLLER DESIGN

The controller transfer function, $G_c(s)$, is:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \qquad (12)$$

where, K_p, K_i and K_d are the proportional, integral and derivative gains, respectively. The performance index including Integrated Absolute Error (IAE) is employed in this research. The proposed control criterion is as follows:

$$IAE = \int_0^{t_f} |r(t) - y(t)| dt = \int_0^{t_f} |e(t)| dt \qquad (13)$$

where, $t_f = 10$ sec subject to control input constraint-$1 < u < 1$.

PSO algorithm stages for searching proper parameters of PID controller are:

First, specify the lower and upper bounds of controller parameters and initialize the particles of the population randomly. Each particle, i.e., K (controller parameters) is sent to Simulink® model. Then, the value of performance criterion is calculated iteratively in Matlab® environment. After that, cost function is evaluated for each particle according to this performance criterion. 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. According to Eq. 11, the velocity of each particle K is modified. At the end of each iteration, program checks the stop criterion. If the number of iterations reaches the maximum designated by the user, the latest global best solution is recorded and the algorithm brings to an end.

In order to examine the dynamic behaviors and convergence characteristics of the proposed method, two statistical indexes, namely the mean value (μ) and the standard deviation (σ) of cost values of all individuals during the computation processes, are used (Haupt and Haupt, 1998). The mean value displays the accuracy of the algorithm and the standard deviation measures the convergence speed of the algorithm. The formulas for calculating these values are as follows, respectively:

$$\mu = \frac{\sum_{i=1}^{n} IAE_{P_i}}{n} \qquad (14)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (IAE_{P_i} - \mu)^2} \qquad (15)$$

where, IAE_{P_i} is the cost value of the individual and n is the population size.

SIMULATION RESULTS

The lower and upper bounds of the three controller parameters are shown in Table 2. The Simulink® block diagram of RIP system with PID controller is shown in Fig. 5. In order to highlight the advantages of the proposed method, it also implemented GA-PID controller. GA parameters according to the trial and error manner are given as follows:

- Population size = 30
- Crossover rate = 0.5
- Mutation rate = 0.02
- Maximum generation = 20

The best controller parameters obtained by GA and PSO algorithms are as follows:

GA: Arm controller $k_p = -4.237$, $k_i = -4.642$, $k_d = -5.329$
Pendulum controller $k_p = 2.349$, $k_i = 21.145$, $k_d = 0.509$

PSO: Arm controller $k_p = 1.615$, $k_i = 2.6103$, $k_d = -0.0393$
Pendulum controller $k_p = 2.027$, $k_i = 11.229$, $k_d = 0.0318$

Figure 6 and 7 shows the arm and pendulum angles using PSO and GA methods. Also, Fig. 8 and 9 shows the

Table 2: Range of three controller parameters

Controller parameters	Lower bounds	Upper bounds
K_d	-25	25
K_p	-25	25
K_i	-25	25

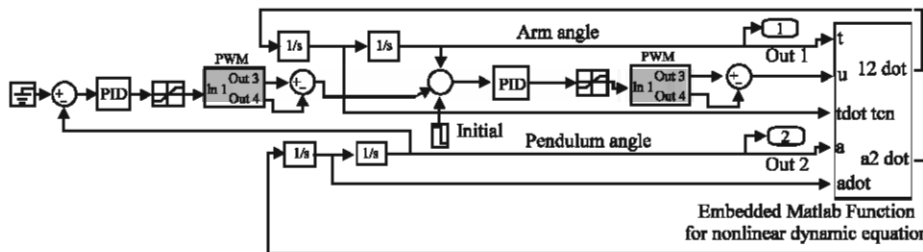Fig. 5: Block diagram of RIP system with PID controller
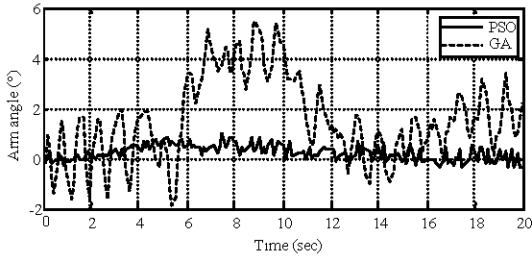
Fig. 6: The arm angle using GA and PSO based PID control

Fig. 7: The pendulum angle with GA and PSO based PID control

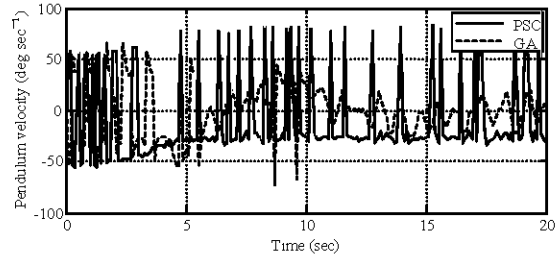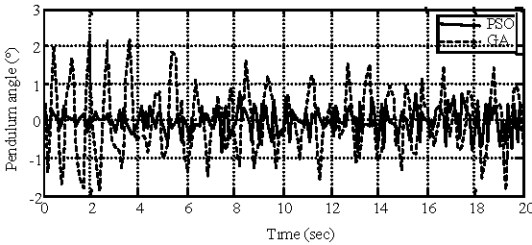Fig. 8: The arm velocity via GA and PSO based PID control

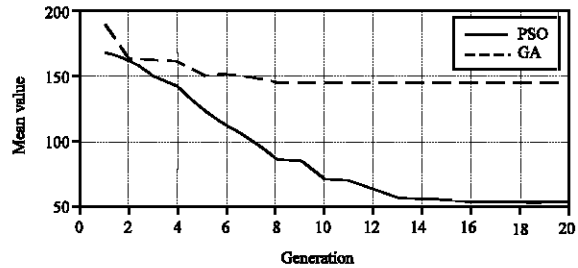Fig. 9: The pendulum velocity via GA and PSO based PID control

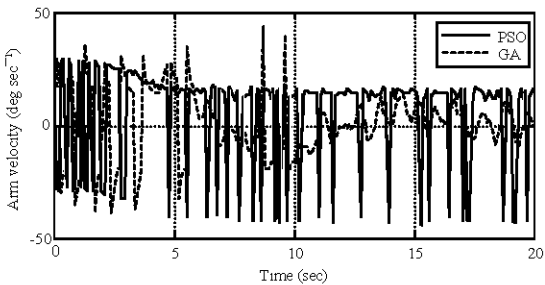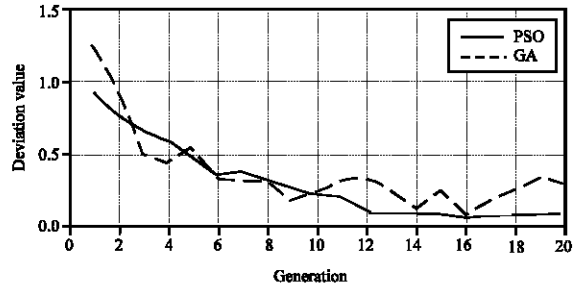Fig. 10: Convergence tendency of mean values of pendulum angles using both methods

Fig. 11: Convergence tendency of standard deviation values of pendulum angles using both methods

system velocities. The integral absolute error of system angles are shown in Table 3. As it can be seen, PSO-based controller makes fine responses, indicating the superiority over GA-PID controller.

Furthermore, under the same conditions, we performed several simulations to compare the controllers' convergence characteristics. After each generation, the mean value (μ) and the standard deviation (σ) of the cost values of all individuals are recorded for observing the dynamic convergence behavior of the individuals in population. As seen in simulations in Fig. 10 and 11, though both controllers can obtain stable mean cost value under the same cost function and simulation conditions, the GA-PID controller brings premature convergence such that the cost value and mean value are bigger. Conversely, the PSO-PID controller has better cost value and mean value, showing that it can achieve better

Table 3: Integral Absolute Error (IAE) of system angles

IAE	PSO	GA
Arm angle	61.027	365.7125
Pendulum angle	52.748	145.4720

accuracy. Simultaneously, we can also find that the convergence tendency of the standard deviation of cost values in the PSO-PID controller is much faster than the GA-PID controller. This can prove that the PSO method has better convergence efficiency.

CONCLUSION

In this study, a PSO-based controller for nonlinear model of the rotary inverted pendulum system is presented. Through the simulation results, the proposed controller performs an efficient search for proper PID

parameters. This study demonstrates that PSO method can solve searching and tuning the controller parameters more efficiently than GA. The proposed method could be considered as a promising way for nonlinear control systems in general. The topic of our future researches is to utilize other cognitive methods in order to achieve better results for designing controller and improving the performance in real time. Also, implementation of heuristic algorithms for designing adaptive controllers will be our future challenging task. Furthermore, tele-operation control of RIP system using haptic device would be another challenge.

ACKNOWLEDGMENT

REFERENCES

Gaing, Z.L., 2004. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans. Energy. Conv., 19: 384-391.

Haupt, R.L. and S.E. Haupt, 1998. Practical Genetic Algorithms. 2nd Edn. John Wiley and Sons Inc., New York, ISBN 0-471-18873-5, pp: 123-190.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks. Nov. 27-Dec. 1 1995, Perth, Australia, pp: 1942-1948.

Kwok, D.P., T.P. Leung and F. Sheng, 1993. Genetic algorithms for optimal dynamic control of robot arms. Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation. 15-19 November, San Francisco, CA, pp: 380-385.

Muskinja, N. and B. Tovornik, 2006. Swinging up and stabilization of a real inverted pendulum. IEEE Trans. Ind. Elect., 53: 631-639.

Rahmat-Samii, Y., 2003. Genetic algorithm and particle swarm optimization in engineering electromagnetics. Proceedings of the 17th International Conference on Applied Electromagnetic and Communications. 1-3 Oct. 2003, Dubrovnik, Croatia, pp: 1-5.

Visioli, A., 2001. Tuning of PID controllers with fuzzy logic. IEE Proc. Control Theor. Applic., 148: 1-8.

Yan, Q., 2003. Output tracking of undergraduated rotary inverted pendulum by nonlinear controller. Proceedings of the 42nd IEEE Conference on Decision and Control. 9-12 December, Maui, Hawaii USA., pp: 2395-2400.

Zhou, G. and J. Birdwell, 1994. Fuzzy logic-based PID auto-tuner design using simulated annealing. Proceeding of the IEEE/IFAC Joint Symposium, Computer Aided Control System Design. 7-9 March. Tucson, AZ, USA., pp: 67-72.