Application of Chaotic Particle Swarm Optimization to PID Parameter Tuning in Ball and Hoop System

Hamed Mojallali, Reza Gholipour, Alireza Khosravi, and Hossein Babaee

***Abstract*—In this paper, an intelligent PID controller based on Chaotic Particle Swarm Optimization (CPSO) algorithm for Ball and Hoop system is designed. In this system, two goals are tracked; the first one implies on set-point tracking, and the second one includes both set-point tracking and disturbance rejection. The classical methods of PID tuning such as Ziegler-Nichols are based on trial and error, and generally, their responses have a high settling time and overshoot; however, the proposed CPSO-PID controller determines the parameters of PID controller automatically and intelligently by minimizing the integral absolute error (IAE). The simulation results on Ball and Hoop system show that the proposed CPSO-PID controller leads to superior performance compared to Ziegler-Nichols method in both set-point tracking and disturbance rejection in terms of rise time, settling time, maximum overshoot, and the integral of absolute error (IAE) performance criterion.**

***Index Terms*—Chaotic particle swarm optimization, PID controller, ball and hoop system, integral of absolute error (IAE).**

I. Introduction

Even though several control theories have been developed significantly, we do see the widely popular use of proportional-integral- derivative (PID) controllers in process control, motor drives, flight control, and instrumentation. The reason of this acceptability is its simple structure which can be easily understood and implemented. Unfortunately, it has been quite difficult to tune properly the gains of PID controllers, because many industrial plants are often burdened with problems such as high order, time delays, and nonlinearities [1]–[6]. Over the years, several heuristic methods have been proposed for the tuning of PID controllers. The first classical tuning rules proposed by Ziegler and Nichols. In general, it is often hard to determine optimal or near optimal PID parameters with the Ziegler-Nichols formula in many industrial plants [1]–[3]. For these reasons, it is highly desirable to increase the capabilities of PID controllers by adding new features. Many artificial intelligence (AI) techniques have been employed to improve the controller performances for a wide range of plants while retaining their basic characteristics. AI techniques such as neural network, fuzzy system, and neural-fuzzy logic have been widely applied to proper tuning of PID controller parameters [1], [2].

Chaos can be described as a bounded nonlinear system with deterministic dynamic behavior that has ergodic and stochastic properties [7]. It is very sensitive to the initial conditions and the parameters used. In what is called the ‘‘butterfly effect’’, small variations of an initial variable will result in huge differences in the solutions after some iteration. Generating an ideal random sequence is of great importance in the fields of numerical analysis, sampling and heuristic optimization. Recently, a technique which employs chaotic sequences via the chaos approach (chaotic maps) has gained a lot of attention and been widely applied in different areas, such as the chaotic neural network (CNN) [8], chaotic optimization algorithms (COA) [9],[10], nonlinear circuits [11], DNA computing [12], and image processing [13]. All of the above-mentioned methods rely on the same pivotal operation, namely the adoption of a chaotic sequence instead of a random sequence, and thereby improve the results due to the unpredictability of the chaotic sequence [14].

PSO shows a promising performance on nonlinear function optimization and has thus received much attention [15]. However, the performance of the traditional PSO greatly depends on its parameters, and it often suffers the problem of being trapped in local optima [16],[17]. In order to avoid these disadvantages, the chaotic particle swarm optimization (CPSO) method based on the logistic equation has been proposed [17].Such an algorithm which is known as Chaotic Particle Swarm Optimization (CPSO) is used in this paper in order to determine the optimal proportional-integral-derivative (PID) controller parameters.

The performance of the closed-loop system can defined in terms of rise time, overshoot, settling time and steady state error. In general, the system with fast rise and settling time under no steady-state error and almost zero overshoot is desired. Hence, in this study to provide a desired performance, the Integral of Absolute Error (IAE), is minimized by using CPSO. The merits of the proposed controller are illustrated by considering the Ball and Hoop system.

The rest of the paper is organized as follows. Section 2 describes typical CPSO. Section 3 is concerned the Ziegler-Nichols tuning method. In section 4, the proposed CPSO-PID controller is described. The results obtained from simulations performed by considering the Ball and Hoop system is discussed in Section 5. Finally, in section 6, the general conclusions are presented.

II. Method

A. Particle Swarm Optimization (PSO)

In original PSO [18], each particle is analogous to an

Manuscript received May 16, 2012; revised June 29, 2012.

H. Mojallali is with the Electrical Engineering Department, Faculty of Engineering, University of Guilan, Rasht, Iran (e-mail: mojallali@guilan.ac.ir)

R. Gholipour, A. Khosravi, and H. Babaee are with the Department of Electrical and Computer Engineering, Babol (Noushirvani) University of Technology, Babol, Iran.

individual ‘‘fish’’ in a school of fish. It is a population-based optimization technique, where a population is called a swarm. A swarm consists of N particles moving around in a D-dimensional search space. The position of the *i*th particle can be represented by $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$. The velocity for the *i*th particle can be written as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. Each particle coexists and evolves simultaneously based on knowledge shared with neighboring particles; it makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. The best previously encountered position of the *i*th particle is denoted its individual best position $p_i = (p_{i1}, p_{i2}, ..., p_{iD})$, a value called $pbest_i$. The best value of the all individual $pbest_i$ values is denoted the global best position $g_i = (g_1, g_2, ..., g_D)$ and called $gbest$. The PSO process is initialized with a population of random particles, and then the algorithm executes a search for optimal solutions by continuously updating generations. At each generation, the position and velocity of the *i*th particle are updated by $pbest_i$ and $gbest$ in the swarm. The update equations can be formulated as:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times (pbest_{id} - x_{id}^{old}) + c_2 \times r_2 \times (gbest_d - x_{id}^{old}) \quad (1)$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \quad (2)$$

r_1 and r_2 are random numbers between (0, 1), and c_1 and c_2 are acceleration constants, which control how far a particle will move in a single generation. Velocities v_{id}^{new} and v_{id}^{old} denote the velocities of the new and old particle, respectively. x_{id}^{old} is the current particle position, and x_{id}^{new} is the new, updated particle position. The inertia weight w controls the impact of the previous velocity of a particle on its current one [19]. In general, the inertia weight is decreased linearly from 0.9 to 0.4 throughout the search process to effectively balance the local and global search abilities of the swarm [20]. The equation for the inertia weight w can be written as:

$$w = (w_{\max} - w_{\min}) \times \frac{Iteration_{\max} - Iteration_i}{Iteration_{\max}} + w_{\min} \quad (3)$$

In Eq. (3), $w_{\max}$ is 0.9, $w_{\min}$ is 0.4 and $Iteration_{\max}$ is the maximum number of allowed iterations.

B. Chaotic Particle Swarm Optimization (CPSO)

In the field of engineering, it is well recognized that chaos theory can be applied as a very useful technique in practical application. The chaotic system can be described by a phenomenon, in which a small change in the initial condition will lead to nonlinear change in future behavior, besides that the system exhibits distinct behaviors under different phases, i.e. stable fixed points, periodic oscillations, bifurcations, and ergodicity [21]. Chaos [22] is also a common nonlinear phenomenon with much complexity and is similar to randomness. Chaos is typically highly sensitive to the initial values and thus provides great diversity based on the ergodic property of the chaos phase, which transits every state without repetition in certain ranges. It is generated through a deterministic iteration formula. Due to these characteristics, chaos theory can be applied in optimization.

In PSO, the parameters w, r_1 and r_2 are the key factors affecting the convergence behavior [23],[24]. The inertia weight controls the balance between the global exploration and the local search ability. A large inertia weight favors the global search, while a small inertia weight favors the local search. For this reason, an inertia weight that linearly decreases from 0.9 to 0.4 throughout the search process is usually used [20]. Since logistic maps are frequently used chaotic behavior maps and chaotic sequences can be quickly generated and easily stored, there is no need for storage of long sequences [25]. In CPSO, sequences generated by the logistic map substitute the random parameters r_1 and r_2 in PSO. The parameters r_1 and r_2 are modified by the logistic map based on the following equation.

$$Cr_{(t+1)} = 4 \times Cr_{(t)} \times (1 - Cr_{(t)}) \quad (4)$$

In Eq. (4), $Cr_{(0)}$ is generated randomly for each independent run, with $Cr_{(0)}$ not being equal to {0, 0.25, 0.5, 0.75, 1}. The velocity update equation for CPSO can be formulated as:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times Cr \times (pbest_{id} - x_{id}^{old}) + c_2 \times (1 - Cr) \times (gbest_d - x_{id}^{old}) \quad (5)$$

In Eq. (5), Cr is a function based on the results of the logistic map with values between 0.0 and 1.0. Fig. 1 shows the chaotic Cr value using a logistic map for 200 iterations where $Cr_{(0)} = 0.0011$. The pseudo-code of CPSO is shown below [26].

```
CPSO pseudo-code
begin
  Randomly initialize particles swarm
  Randomly generate Cr(0)
  while (number of iterations, or the stopping criterion is not met)
    Evaluate fitness of particle swarm
    for n = 1 to number of particles
      Find pbest
      Find gbest
      for d = 1 to number of dimension of particle
        update the Chaotic Cr value by Eq. (4)
        update the position of particles by Eq. (5) and Eq. (2)
      next d
    next n
    update the inertia weight value by Eq. (3)
  next generation until stopping criterion
end
```

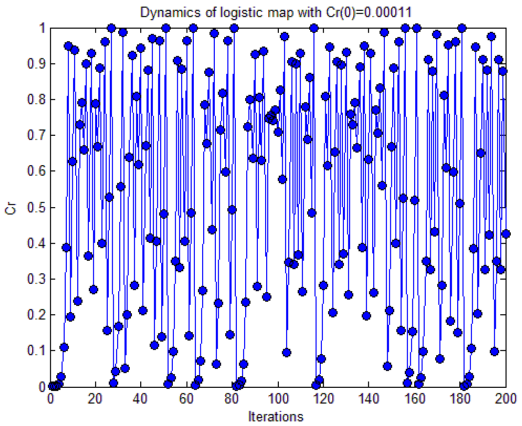

Fig. 1. Chaotic Cr value using a logistic map for 200 iterations; Cr(0)=0.00011

In fact, In CPSO, a chaotic map was embedded todetermine the PSO parameters r_1 and r_2 . The PSOparameters r_1 and r_2 cannot ensure optimal ergodicity in the search space because they are absolutely random [27] i.e. the r_1 and r_2 are generated by a linear congruential generator (LCG) with a random seed. The generated sequence of LCG consists of pseudo-random numbers that have periodic characteristics [28]. Furthermore, the generated sequence of a logistic map also consists of pseudo-random numbers, but there are no fixed points, periodic orbits, or quasi-periodic orbits in the behavior of the chaos system [29]. As a result, the system can avoid entrapment in local optima [17].

III. Ziegler-Nichols Closed-Loop Tuning Method (Ultimate Gain and Ultimate Period Method)

The closed-loop tuning method proposed by Ziegler-Nichols requires the determination of the ultimate gain and ultimate period. The method can be interpreted as a technique of positioning one point on the Nyquist curve [30], [31]. This can be achieved by adjusting the controller gain (K_c) till the system undergoes sustained oscillations (at the ultimate gain or critical gain), whilst maintaining the integral time constant (T_i) at infinity and the derivative time constant (T_d) at zero.

A significant drawback of this closed-loop tuning method is that the ultimate gain has to be determined through trial and error and the system has to be driven to its stability limits. Another disadvantage is that when the process is unknown, the amplitudes of the undampened oscillations can become excessive when using trial and error to determine the ultimate gain of the system. This could lead to unsafe plant conditions. The closed loop tuning rules for P, PI and PID control are given in Table I.

TABLE I: Ziegler-Nichols Closed Loop Tuning Parameter

Controller	K_c=K_p	T_i=K_p/K_i	T_d=K_d/K_p
P	0.5K_u	∞	0
PI	0.4K_u	0.8P_u	0
PID	0.6K_u	0.5P_u	0.125P_u

IV. Proposed CPSO-PID Controller

The basic form of the PID controller composes from sum of the multiplication, integration and differentiation of the error signal and each operator is multiplied by three control parameters. Time domain representation of PID control equation is presented in Eq. 6, and the transfer function of the controller is given in Eq. 7:

$$u(t) = k_p e(t) + k_I \int e(t)dt + k_D \frac{d(e(t))}{dt} \tag{6}$$

$$G(s) = \frac{U(s)}{E(s)} = \frac{k_D s^2 + k_p s + k_I}{s} \tag{7}$$

where k_p , k_I and k_D (three-term) are the control parameters, $u(t)$ is the controller output/system input, and $e(t)$ is error signal. In frequency domain, the transfer function of the PID controller has one pole at the origin, and two zeros with locations depending on the tuning strategy.

The ‘‘three-term” functionalities are highlighted below [32].

- The proportional term – providing an overall control action proportional to the error signal through the all pass gain factor.
- The integral term – reducing steady state errors through low frequency compensation by an integrator.
- The derivative term – improving transient response through high frequency compensation by a differentiator.

The proposed CPSO-PID controller structure is shown in Fig. 2. In CPSO-PID controller, CPSO algorithm is utilized to determine three optimal PID gains, i.e., k_p , k_I , and k_D . Obviously, PID gains optimization is in a three- dimensional searching space. In this paper, the Integral of absolute magnitude of the error(IAE) performance criterion [33]is used for the PID gains optimizing, which is as follows:

$$IAE = \int_0^{t_f} |e(t)| dt \tag{8}$$

where t_f is the final time in seconds and t is the time, in seconds. IAEgets the absolute value of the error to remove negative error components. In steady state, the performance criterion must be minimized by CPSO algorithm.

V. Simulation Results

In this study, a system simulation was carried out on the combination of the CPSO algorithm and the PID control system as shown in Fig. 2. The input variables of the proposed CPSO-PID controller is the error $e(t)$ and the output variable is the control variable $u(t)$. $r(t)$ denotes the step input signal, disturbance input signal *d(t)* acts as an additive with the controller output signal $u(t)$ such that the process input is governed by $u_{process}(t) = u(t) + d(t)$. The process output $y(t)$ is fed back to the input of the controller to form the error signal $e(t)$.

To achieve the goal of minimizing the IAEof the control system, we used the CPSO algorithm. The parameters of the CPSO Algorithm are set as shown in Table II. The sampling time in this simulation is 0.01.

TABLE II: PARAMETERS USED IN THE CPSO

Population size	20
Acceleration constant c_1	2
Acceleration constant c_2	2
Inertia weight w	started from 0.9 and decreased linearly to 0.4
Number of iterations	20

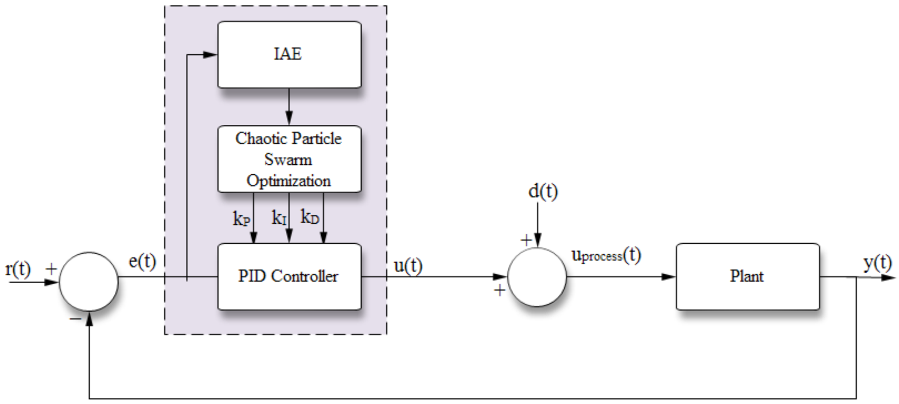

Fig. 2. The block diagram of the CPSO-PID controller

TABLE III: COMPARATIVE STUDY OF CONTROL SYSTEM PERFORMANCES WITHOUT DISTURBANCE.

method	k_p	k_I	k_D	t_r	M_p (%)	t_s (5%)	IAE
zn-pid controller	6	1.9078	4.7178	1.2900	58.4	10.6	3.5549
cpso-pid controller	5.8653	0.0001	11.4188	**1**	**14.6**	**4.88**	**1.2955**

A. *Tuning of Ball and Hoop System for Optimal Set-Point Tracking*

The objective of this experiment is to compare the control performance of a CPSO Algorithm tuned loop to a loop tuned using the ZN tuning method.

The Ball and hoop system used in this experiment is given as [35], [36]:

$$G(s)=\frac{1}{s^4+6s^3+11s^2+6s} \tag{9}$$

Table IIIsummarizes control system performance by setting the control parameters [$k_p\ k_I\ k_D$] yielded by the two methods: Ziegler-Nichols and Chaotic Particle Swarm Optimization.

In CPSO-PID Controller, the searching ranges for the PID parameters k_p, k_I, and k_D are limited to [0, 20].A sample of the trajectory of the PID parameters and performance criterion during optimization is shown in Figs. 3 and 4,respectively. The PID parameters are obtained for 20 iterations. In this example t_f is equal to 20 seconds.

The step responses of the controllers mentioned above are simulated and the responses are plotted in Fig. 5. The performance indices (rise time, settling time, maximum overshoot, andIAE) of the results are evaluated and these parameters are compared in Table III. It is shown that the proposed controller exhibits better performance as compared to ZN-PID controller in terms of rise time (t_r), settling time (t_s), maximum overshoot (M_p), and IAE.

B. *Tuning of Ball and Hoop System for Set-Point Tracking and Disturbance Rejection*

The objective of this experiment is to demonstrate the effectiveness of the CPSO algorithm method to tune the PID controller for set-point tracking and disturbance rejection. The ZN tuning methodology is chosen for comparison with the CPSO Algorithm. A unit step load disturbance $d(t)$ is introduced into the process input at $t_{dist} = 20$ *seconds*.Table IVsummarizes control system performance by setting the control parameters [$k_p\ k_I\ k_D$] yielded by the two methods: Ziegler-Nichols and Chaotic Particle Swarm Optimization.In CPSO-PID Controller The searching ranges for the PID parameters k_p, k_I, and k_D are limited to [0, 20]. A sample of the trajectory of the PID parameters and performance criterion during optimization is shown in Figs. 6 and 7,respectively. The PID parameters are obtained for 20 iterations. In this example, t_f is equal to 40 seconds.

The step responses of the controllers mentioned above are simulated and the simulated responses are plotted in Fig. 8. The performance indices (rise time, settling time, maximum overshoot, andIAE) are evaluated and these parameters are compared in Table IV. It is shown that the proposed controller exhibits better performance as compared to

ZN-PID controller.Also, the proposed method delivers the best recovery to load disturbance.

TABLEIV: COMPARATIVE STUDY OF CONTROL SYSTEM PERFORMANCES WITH DISTURBANCE.

Method	k_p	k_I	k_D	t_r	M_p (%)	t_s (5%)	IAE
ZN-PID Controller	6	1.9078	4.7178	1.2900	58.4	26.2	4.3225
CPSO-PID Controller	10.8626	3.9774	16.5134	**0.71**	**51.6**	**24.22**	**2.1282**

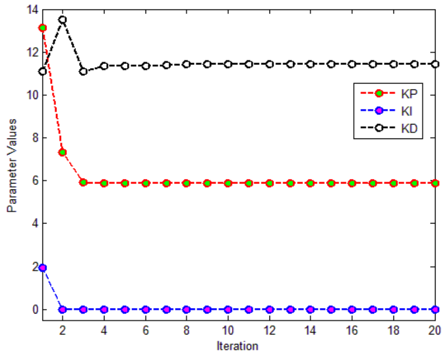

Fig. 3. The parameter values trajectory

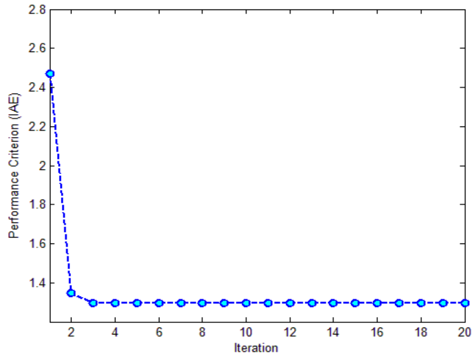

Fig. 4. the performance criterion trajectory

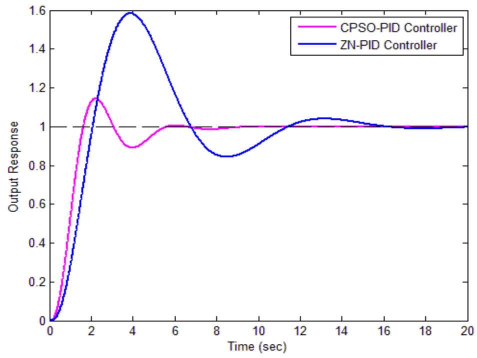

Fig. 5. The step responses (0-20sec) of the Ball and Hoop system

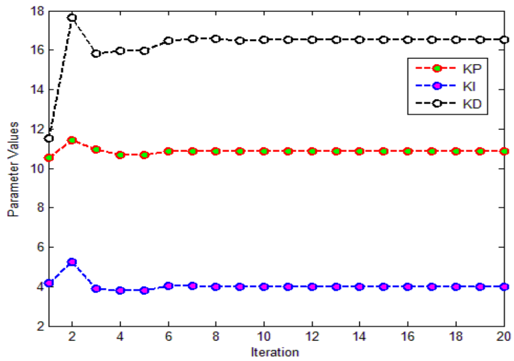

Fig. 6. The parameter values trajectory

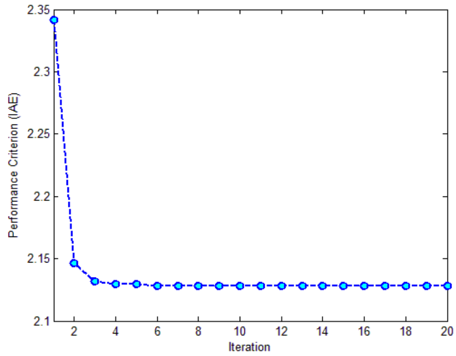

Fig. 7.the performance criterion trajectory

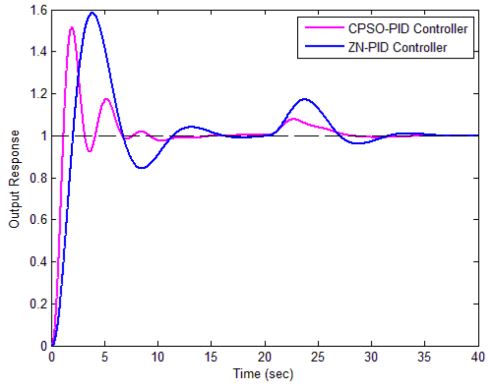

Fig. 8. The step responses (0-40sec) of the Ball and Hoop for set-point tracking and disturbance rejection

VI. CONCLUSIONS

In this paper, we treated two goals of set-point tracking and disturbance rejecting in Ball and Hoop system. For this purpose, an intelligent CPSO-PID controller is designed. In this controller, the parameters of PID controller, automatically and intelligently are determined by CPSO Algorithm without trial and error. The IAEis the minimized fitness function via the CPSO. The experimental results on Ball and Hoop system show that the proposed CPSO-PID controller is superior to classical Ziegler-Nichols method in both goals of set-point tracking and disturbance rejection in terms of rise time, settling time, maximum overshoot, andIAE.

REFERENCES

[1] A. Visioli, “Tuning of PID controllers with fuzzy logic,” in *Proc. Inst. Elect. Eng. Contr. Theory Applicat*, vol. 148, no. 1, pp. 1–8, Jan. 2001.

[2] T. L. Seng, M. B. Khalid, and R. Yusof, “Tuning of a neuro-fuzzy controller by genetic algorithm,” *IEEE Trans. Syst, Man, Cybern. B*, vol. 29, pp. 226–236, Apr. 1999.

[3] R. A. Krohling and J. P. Rey, “Design of optimal disturbance rejection PID controllers using genetic algorithm,” *IEEE Trans. Evol. Comput*, vol. 5, pp. 78–82, Feb. 2001.

[4] Y. Mitsukura, T. Yamamoto, and M. Kaneda, "A design of self-tuning PID controllers using a genetic algorithm," in *Proc. Amer. Contr. Conf.*, San Diego, CA, June 1999, pp. 1361–1365.

[5] T. Kawabe and T. Tagami, "A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom," in *Proc. 12th IEEE Int. Symp. Intell. Contr*, Istanbul, Turkey, July 1997, pp. 119–124.

[6] R. A. Krohling, H. Jaschek, and J. P. Rey, "Designing PI/PID controller for a motion control system based on genetic algorithm," in *Proc. 12thIEEE Int. Symp. Intell. Contr*, Istanbul, Turkey, July 1997, pp. 125–130.

[7] H. G. Schuster, "Deterministic chaos an introduction," Second revised ed, P. V. H. G. Weinheim, *Federal Republic of Germany*, 1988.

[8] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks,"*Physics Letters A*, vol. 144, pp. 333–340, 1990.

[9] B. Li and W. S. Jiang, "Optimizing complex functions by chaos search," *Cybernetics and Systems*, vol. 29, pp. 409–419, 1998.

[10] Z. Lu, L.S. Shieh, and G. R. Chen, "On robust control of uncertain chaotic systems: a sliding-mode synthesis via chaotic optimization,"*Chaos, Solitons and Fractals*, vol. 18, pp. 819–827, 2003.

[11] P. Arena, R. Caponetto, L. Fortuna, A. Rizzo, and M. L. Rosa, "Self organization in non recurrent complex system,"*International Journal of* Bifurcation *and Chaos*, vol. 10, pp. 1115–1125, 2000.

[12] G. Manganaro and J. P. D. Gyvez, "DNA computing based on chaos," in *Proc.the IEEE International Conference on Evolutionary Computation* , 2002, pp. 255–260.

[13] H. Gao, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption,"*Chaos, Solitons and Fractals*, vol. 29, pp. 393–399, 2006.

[14] B. Alatas, E. Akin, and A. BedriOzer, "Chaos embedded particle swarm optimization algorithms,"*Chaos, Solitons and Fractals*, vol. 40, pp. 1715–1734, 2009.

[15] Y. Liu, Z. Qin, Z. Shi, and J. Lu, "Center particle swarm optimization, Neurocomputing, vol. 70, pp. 672–679, 2007.

[16] Angeline PJ. "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences,"*Evolutionary* programming, vol. VII. Springer; 1998. p. 601–10.

[17] B. Liu, L. Wang, Y. H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos Solitons and Fractals*, vol. 25, pp. 1261– 1271, 2005.

[18] J. Kennedy and R. C. Eberhart, "Particle swarm optimization,"*IEEE International Conference on Neural Networks 4, Perth, Australia* pp. 1942–1948, 1995.

[19] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc IEEE International Conference on Evolutionary Computation*, *Anchorage, AK*, 2002, pp. 69–73.

[20] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congress on Evolutionary Computation*, Washington, *DC*, 2002, pp. 1945-1949.

[21] D. S. Coelho, L. Herrera, and B. M, "Fuzzy identification based on a chaotic particle swarm optimization approach applied to a nonlinear yo-yo motion system," *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 3234–3245, 2007.

[22] H. Lu, H. M. Zhang, and L. H. Ma, "A new optimization algorithm based on chaos,"*Journal of Zhejiang University Science A*, vol. 7, pp. 539–542, 2006.

[23] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, pp. 317– 325, 2003.

[24] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Transactions on Power Systems*, vol. 18, pp. 60–68, 2003.

[25] H. GAO, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption,"*Chaos, Solitons and Fractal*, vol. 29, pp. 393–399, 2006.

[26] L.Y. Chuang, S. Tsai, and C. H. Yang, "Chaotic catfish particle swarm optimization for solving global numerical optimization problems," *Applied mathematics and computation*, vol. 217, 6900-6916, 2011.

[27] J. Chuanwen and E. Bompard, "A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment," *Energy Conversion and Management*, vol. 46, pp. 2689–2696, 2005.

[28] D. E. Knuth, "The Art of Computer Programming, Seminumerical Algorithms," *The Linear Congruential Method*, *Third ed*, vol. 2, 1997, pp. 10–26.

[29] D. Kuo, "Chaos and its computing paradigm,"*IEEE Potentials Magazine*, vol. 24, pp. 13–15, 2005.

[30] K. J. Astrom and T. Hagglund, "PID Controllers: Theory, Design and Tuning," ISA, Research Triangle, Par, NC, 1995.

[31] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 65, pp. 433-444, 1942.

[32] M. WilljuiceIruthayarajan and S. Baskar, "Evolutionary algorithms based design of multivariable PID controller,"*Expert Systems with Applications*, vol. 36, pp. 9159–9167, 2009.

[33] C. H. Lee and C. C. Teng, "Calculation of PID controller parameters by using a fuzzy neural network," *ISA Transactions*, vol. 42, pp. 391–400, 2003.

[34] I. Griffin, On-line PID controller tuning using genetic algorithms, Master's Thesis, Dublin City University, Ireland, 2003.

[35] P. Wellstead. Ball and hoop. (2008). [Online]. Available: http://www.control-systems-principles.co.uk.