Tuning PID Controller Based On the SWARM Intelligence

Mahmoud Osman, Ali Infis, Wafa Abied, and Suliman Elfandi

***Abstract*----**In any of the control application, PID controller design is the most important part. There are many methods to tune the PID controller such as Ziegler-Nichols and Cohen-Coon tuning methods, but these conventional tuning methods present an unsatisfactory control performance according to some of empirical studies. Conventional controller has become powerless for these systems. It is for this reason a need to exist for the development of a suitable tuning technique that is applicable for a wide range of control loops that do not respond satisfactorily to conventional tuning. The introduced tuning strategy is employed as optimization engines to produce the PID parameters in the control loops with performance indices near to the optimal ones. Tuning PID controller gains based on particle swarm optimization (PSO) will improve the performance of the closed-loop systems in comparison with the conventional PID tuning approaches.

***Keywords*----**PID tuning, PSO

I. Introduction

THE PID controller is the most common form of feedback control systems. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940s. In process control today, more than 90% of the control loops are of PID controller type, most loops are actually PI control [1]. PID controllers are today found in all areas where process control is used.

PID controllers have survived many changes in technology, from mechanics and pneumatics to microprocessors. The microprocessor has had a dramatic influence on the PID controller. Practically all PID controllers made today are based on microprocessors. This has given opportunities to provide

Evolutionary computation (EC) and Swarm Intelligence (SI) fall within the area of artificial intelligence (AI) [8]. EC is founded upon the principles of biological evolution whilst SI techniques are inspired by swarm behavioral patterns occurring in nature [7]. With the increase of computational power, AI has increasingly been used to solve complex linear and nonlinear control problems [9]. This paper provides an introduction to the PSO algorithm for providing an alternative approaches to PID controller tuning.

Tuning the PID parameters/gains is usually realized by classical, trial-and-error approaches, and experienced human experts, which they may not capable to achieve desirable additional features like automatic tuning, gain scheduling, and continuous adaptation [6].

The research in tuning PID parameters started in early 1940's. Several tuning methods have been proposed for the tuning of process control loops, with the most popular method being that of Ziegler and Nichols (1942). Other methods include the methods of Cohen and Coon (1953), Åström and Hägglund (1984), De Paor andO'Malley (1989), Zhuang and Atherton (1993), Venkatashankar and Chidambaram(1994), Poulin and Pomerleau (1996) and Haung and Chen (1996)[1],[3]. In spite of this large range of tuning techniques, to date there still seems to be no general consensus as which tuning method works best for most applications. Some methods relay heavily on experience, while others rely more on mathematical considerations.

Artificial Intelligence deals with the study and creation of computer systems that exhibit some form of intelligence: systems that can learn new concepts and tasks, systems that can reason and draw conclusions about the world around us, systems that can understand a natural language or perceive and comprehend a visual scene, and systems that perform other feats that require human intelligence.

The motivation of Artificial Intelligence (AI) technology is to make computers behave more like humans in solving problems. Soft computing is a tool of artificial intelligence which differs from hard computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth and approximation. In effect, the role model of soft computing is the human mind.

Performance for complex real-world systems with high-order, time-delays, nonlinearities, uncertainties, and without precise mathematical models.

The introduced tuning strategy is based on robust and intelligent technique is employed as optimization engines to produce the PID parameters in the control loops with performance indices near to the optimal ones.

II. PID Algorithm

PID is a feedback based controller which gets the error value and calculates the output based on the characteristics of the error, it is very widely used in plants as it is simple and gives good results. PID is used in a closed loop. It has three elements P, I and D. Every parameter has gain by which we control the contribution of each term [1].

A mathematical description of the PID control is given by:

$$U(t) = K_P \left[e(t) + \frac{1}{T_I} \int e(t)dt + T_D \frac{de}{dt} + U_O \right. \quad \ldots\ldots \quad (1)$$

Mahmoud Osman and Suliman Elfandi, are with Department of Computer Engineering, Faculty of Engineering, University of Tripoli, Tripoli, Libya.

Ali Infis and Wafa Abied, are with Department of Electrical & Electronic Engineering, Faculty of Engineering, University of Tripoli, Tripoli, Libya.

Where, K_P = Proportional gain,T_I = Integral time constant, T_D = Derivative time constant, e(t) = Error signal, U(t) = Controlled output signal and U_O = U(0) is the controlled output at t = 0.The terms T_D is related to K_D and T_I is related to K_I as follows: $K_D = K_P T_D$, $K_I = \frac{K_P}{T_I}$. Hence equation (1) reduces to equation (2)

$$U(t) = K_P\, e(t) + K_I \int e(t)dt + K_D \frac{de}{dt} + U_O \ldots\ldots \quad (2)$$

The PID controller calculation (algorithm) involves three separate parameters; the Proportional K_P, the Integral K_I and Derivative K_D values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent errors and the Derivative determines the reaction to threat at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, the power supply of a heating element or DC motor speed and position control. The Ziegler-Nichols tuning methods result in closed-loop systems with very poor damping, Ziegler-Nichols open-loop tuning is only suitable closed loop Ziegler-Nichols tuning. The method also results in poor tuning and fine tuning is usually necessary to improve loop performance. In spite of these shortcomings, the method is still the most preferred by control practitioners. Its popularity is largely due to the fact that it was amongst the first tuning methods to be proposed, and compared to most other tuning techniques it is still the simplest to use [3].

for open-loop stable processes and the closed-loop method is applicable to processes that operate deep within the stable region under closed-loop conditions. Processes operating on the periphery of the stable region will be unsafe to tune using.

III. Swarm Intelligence (SI)

Swarm Intelligence (SI) methods are based around the study of collective behavior in decentralized, self-organized systems. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. Although there is no centralized control structure dictating how individual agents should behave, local interactions between such agents often lead to the emergence of a global behavior.

Two of the most successful SI techniques modeled on the behavior of natural systems are ant colony optimization (ACO) proposed by Dorigo and Gambardella in 1997[9], and particle swarm optimization (PSO) proposed by Kennedy and Eberhart in 1995 [10].

IV. The Baisic Pso Algorithm

In a PSO system, a swarm of individuals (called particles or intelligent agents) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its entire population. The best position obtained is referred to as the global best particle. The performance of each particle (i.e. how close the particle is from the global optimum) is measured using a fitness function that varies depending on the optimization problem.

Each particle traverses the *X*, *Y* coordinate within a two-dimensional search space. Its velocity is expressed by *Vx* and *Vy* (the velocity along the X-axis and Y-axis, respectively). Modification of the particles position is realized by the position and velocity information [10]. Each agent knows its best value obtained so far in the search (*pbest*) and its *XY* position. This information is an analogy of the personal experiences of each agent. Individual particles also have knowledge about the best value achieved by the group (*gbest*) among *pbest*. Each agent uses information relating to: its current position (*x,y*), its current velocities (*Vx,Vy*), distance between its current position and its *pbest* and the distance between its current position and the groups *gbest* to modify its position. The velocity and position of each agent is modified according(2)and(3),respectively [7]:

$$V_i^{k+1} = V_i^k + c_1 rand_1 \times \left(pbest_i - S_i^k\right) + c_2 rand_2 \times \left(gbest - S_i^k\right) \ldots. (3)$$

If the period time T=1, so,

$$S_i^{k+1} = S_i^k + V_i^{k+1} \ldots\ldots\ldots \quad (4)$$

Where, V_i^k= current velocity of agent i at iteration *k*,V_i^{k+1}= new velocity of agent *i* at iteration *k*, c_1= adjustable cognitive acceleration constants (self-confidence),c_2= adjustable social acceleration constant (swarm confidence),$rand_{1,2}$ = random number between 0 and 1, S_i^k = current position of agent i at iteration *k*,$pbest_i$ = personal best of agent *I* and $gbest$ = global best of the population.

Variations to the conventional PSO algorithm to control convergence of the swarm have been proposed by Shi and Eberhart1998 [7]. The method proposed uses an "inertia weighting" function control the swarm's convergence. so eq. (4) can be written as:

$$V_i^{k+1} = wV_i^k + c_1 rand_1 \times \left(pbest_i - s_i^k\right) + c_2 rand_2 \times \left(gbest - s_i^k\right) \ldots\ldots.(5)$$

All the meanings of the variables in (5) are the same as was defined for (4) and w refers to inertia weighting.For (4): S_i^{k+1} denotes the position of agent i at the next iteration *k* +1.

Kp=3.3, Ti=0.5 and Td=0.33

By substituting the value of these parameters in Equation (7) the closed loop transfer function becomes

$$G(s) = \frac{4s^2+132s+264}{2s^4+10s^3+86s^2+142s+264} \ldots.. \quad (8)$$

The unit step response of the system given in equation(8) based on Zeigler-Nichols tuning method is as shown in Figure (2)

The step response of the system shows as a second order response but indeed the system is a fourth order system, at this case the poles that far away from the other poles is neglected or the poles that has imaginary part will be ignored.

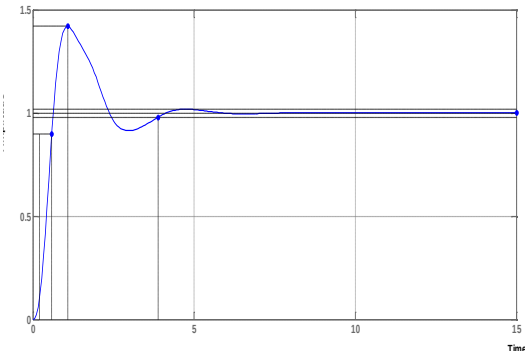


Fig. 2 Unit Step Response PID Tuning Based On Ziegler-Nichlos

V. PSO- ALGORITHM

Where the transfer function of the plant is the same as given in equation (6)

Figure (3) shows positioning of the PSO optimization algorithm within a SISO system. The transfer function of the PID controller is represented by $G_C(s)$ in Figure(3). The closed-loop transfer function of the system of Figure(3) will be written as :

$$G(s) = \frac{40k_d s^3 + 40k_p s + 40k_i}{2s^4 + 10s^3 + (82 + 40k_d)s^2 + (10 + 40k_p)s + 40k_i} \quad .(9)$$

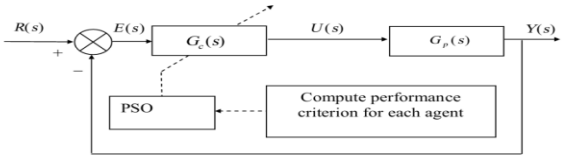


Fig. 3 The PSO Optimization Algorithm Block Diagram

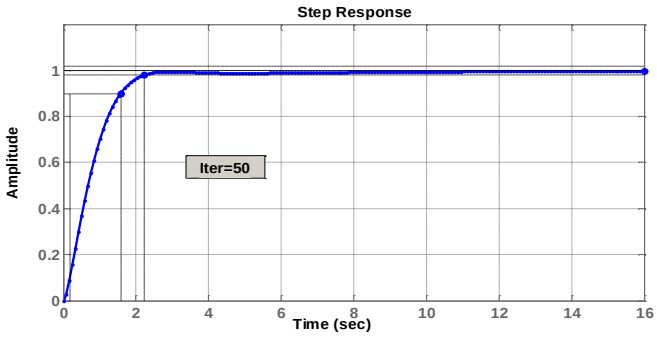
In the simulation using PSO algorithm, had been varied the number of iterations and kept the population of the swarm constant at 200. It presents a comparative study of the performance of the initial global best position out of randomly initialized swarm particles to the performance of the final global best position which comes after the application of "particle swarm optimization" algorithm. The results in the tabular format as shown in table I :

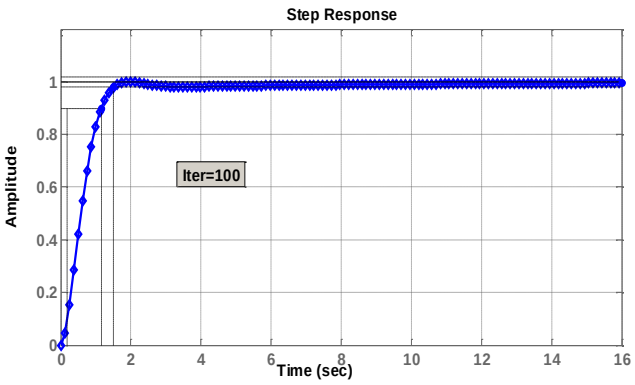
TABLE I
SIMULATION RESULTS OF PSO WITH DIFFERENT NUMBER OF ITERATIONS

Number of Iteration	**400**	**200**	**100**	**50**
Kp	3.4253	3.2424	2.7040	1.9120
Ki	0.3803	0.3796	0.3052	0.2755
Kd	0.2296	0.2573	0.3762	0.5241
Rise time(s)	0.786	0.8044	0.9742	1.3948
Settling time(s)	1.13	1.2241	1.5064	2.2208
Overshoot(%)	0%	0%	0%	0%
ITSE	0.4071	0.4514	0.5195	0.6977

VI. CONCLUSION

For this study the ITSE performance index was chosen to evaluate the control performance of the process loops. The ITSE criterion penalizes large overshoots and minimizes long settling times and found that the best control response is obtained by minimizing these two transient response criteria. So the result is that, the software has been used shown that processes tuned using the PSO methodology is characterized by an improved control behavior for set point tracking and disturbance rejection. This is evident from the improved ITSE performance index when compared to the control performance obtained with using the traditional tuning methods, Ziegler-Nichols. From simulation results show that the PID tuning PSO based system scheme gives better response compared to the other conventional schemes. There're are several attractive features of PSO Based PID Tuning has been found: Simple operating algorithm: The use of simple mathematical operators facilitates a faster computational time and makes the algorithm suitable for determining tuning parameters under high-speed dynamical conditions. Efficient operating algorithm: From the tests that were conducted, it was shown that the PSO determined parameters provide the yielded the best control performance.





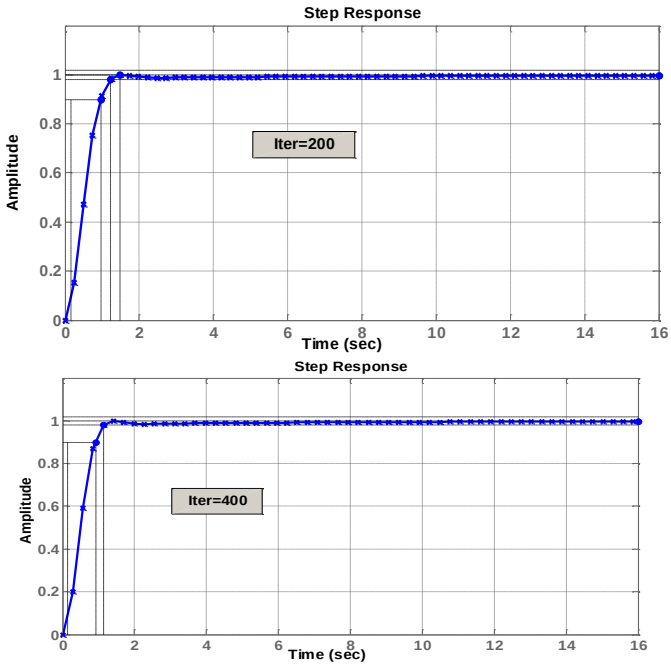


Fig.4 Shows Simulation Results With Different Number of Iterations, Starting With 50 Iterations Till 400, the PSO Provide no Overshoot, Very Short Settling Time and Rise Time as Well as Small Value of ITSE (less than 1)

REFERENCES

[1]. Karl J. Astrom and Tore Hagglund, 2nd Edition , ISBN:1-55617-516-7 ,"PID Controllers: Theory, Design, and Tuning". 1995, Library congress Cataloging Publication Data.

[2].Obaid, Z.A., Sulaiman, N. and Hamidon, M.N., (2009) "*Developed Method of FPGA-based Fuzzy Logic Controller Design with the Aid of Conventional PID Algorithm*", Australian Journal of Basics and Applied Science, Vol. 3(3), P: 2724-2740.

[3].Lieslehto J., "PID controller tuning using Evolutionary programming, American Control Conference", VA June 25-27, 2001.

[4].Dong Hwa Kim, Ajith Abraham, Jae Hoon Cho. (2007), "A hybrid genetic algorithm and bacterial foraging approach for global optimization".Information Sciences 177 (2007) 3918–3937 .Elsevier Inc. http://dx.doi.org/10.1016/j.ins.2007.04.002

[5].Shi Y. and Eberhart R.C., "A modified particle swarm optimizer", Proceedings ofthe IEEE International Conference on Evolutionary Computation, pp 69-73, 1998.

[6].Su Whan Sung Jietae Lee, In-BeumLee.ISBN 978-0-470-82410-8 (HB) "Process Identification and PID Control", 2009,John Wiley & Sons (Asia) Pte Ltd.

[7].Shi Y. and Eberhart R.C. "Particle swarm Optimization: Developments, Application and Resources".0-7803-6657-3/01, (2001) IEEE Service Center.

[8].Aleksandar Jevtic, Diegdo Andina, "Swarm Intelligence and Its Applications in Swarm Robotics" 6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, Tenerife, Spain, December 14-16, 2007.

[9].KkdDorigo M., Gambardella L.M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation, pp.53-66, 1997.

[10].Kennedy J., Russell R.C. and Shi Y., "*Particle Swarm Optimization*", Proc. IEEE