A Particle Swarm Optimization Approach for Model Independent Tuning of PID Control Loops

Nelendran Pillay and Poobalan Govender

Optimization Studies Unit, Dept. of Electronic Engineering,
Durban University of Technology, KwaZulu-Natal, Republic of South Africa

***Abstract*–The paper proposes a model independent tuning technique that yields optimal proportional-integral-derivative parameters for a range of typical process control loops. The proposed tuning technique utilizes the particle swarm optimization algorithm to generate optimal tuning parameters. The PSO tuning method is applied to typical process models. Comparisons are made between the proposed PSO technique and other conventional methods.**

Keywords–particle swarm optimization, tuning, PID control, control loops, process models

I. Introduction and Background

THE PID controller is regarded as the workhorse of the process control industry. Its universal acceptability can be attributed to: the familiarity with which it is perceived amongst researchers and practitioners within the control community, its simple operating algorithm, the relative ease and speed with which controller effects can be adjusted with minimal down-time and the wide range of applications where it has reliably produced excellent control performances. Empirical tuning methods have been proposed by researchers since 1942, with the most popular being the ultimate sensitivity method and the process reaction curve method proposed by Ziegler and Nichols (Z-N) [7].

The Cohen-Coon tuning method [8] is based on the transient response of first-order-systems having dead-time present in the control channel, and relies on the reaction curve of the system's open loop step response. The techniques of [7] and [8] yield good initial values for the controllers P, I and D tuning parameters. Fine tuning for improved control performance is obtained iteratively and relies heavily on the practitioner's intuition and experience. Following the proposals of [7] and [8], several new methods and modifications to the method of [7] have been proposed to improve the control of plants not covered by [7]. For example, De Paor and O'Malley [9] derived tuning methods for unstable processes having a time delay, based on the optimal gain and phase margins for P, PI and PID controllers. Venkatashankar and Chidambaram [10] showed how to determine the gain and the time constant for P and PI controllers in unstable first order plus time delay systems. Poulin and Pomerleau [11] developed the maximum peak-resonance tuning methodology for both unstable and integrating processes. Åström and Hägglund [14] utilized the gain and phase margin specifications to estimate the tuning parameters for simple auto-tuned single loop controllers for controlling processes having small dead-times. Hang *et al.* [13] proposed refinements to the Z-N method for application to processes experiencing excessive initial overshoot or undershoot in the response. To date, the Z-N [7] method is the most popular and is preferred by most control practitioners in the field. The methods proposed by [9]-[11], [14] are often not applied in practice because of the reluctance of control personnel to learn new techniques which they perceive as being relatively complicated and often time consuming and laborious to implement. For these reasons this paper proposes a simple model independent technique for determining the optimal PID control parameters for servo-tracking control loops. The proposed tuning method utilizes the particle swarm optimization (PSO) search algorithm developed by Eberhart and Kennedy [2] to determine the tuning parameters for optimal PID control in systems that represent a good sampling of typical industrial processes. The paper is arranged as follows: Section II briefly discusses the PSO method; Section III describes some basic PID control theory and also gives the process models used in the study; Section IV shows how the PSO algorithm is used to determine the P-gain and the integral and derivative time constants for optimal PI and PID control; Section IV also discusses the simulation experiments that were conducted and compares the results to the methods of [7]-[11], [14]. An analysis of the results is also given in this section; Section V concludes the study.

II. Overview of Particle Swarm Optimisation

The PSO technique, developed by Kennedy and Eberhart [2], is a computational based optimization technique for dealing with problems in which a best solution can be represented as a point or surface within an *n*-dimensional search space. The technique is based on an analogy of the social interaction that exists in flocking birds and swarming bees. In the case of the PSO algorithm the social interaction occurs as the population of individuals within the swarm traverses a search space looking for an optimal solution. The technique, much like a genetic algorithm (GA), is stochastic in nature. The stochasticity occurs when following each iteration the acceleration is weighted by a random term, with individual random numbers being generated for acceleration towards the personal best (p-best) and the group's global best (g-best) locations.

Unlike GA's, PSO does not depend on the principle of 'survival of the fittest', is computationally less burdening since its memory and processing speed requirements are low, and does not use evolutionary operators such as crossover and mutation [3], [6]. Also the premature convergence of GA's degrades its performance thereby reducing its search capability [6]. Other attractive features of PSO include its use of only primitive mathematical operators, its robust search ability for combinatorial optimization problems, stable convergence characteristics

1-4244-0987-X/07/$25.00 ©2007 IEEE

and its resilience to the problem of local minima [3]. Other related work where the PSO algorithm has provided optimal solutions to a range of control problems can be found in [16]–[19].

Each particle or so-called *intelligent agent* within the swarm is given an initial random velocity and dynamically adjusts its velocity and positional trajectories as it traverses a predefined search space looking for a potential solution. These adjustments are based on the personal experiences of the agent in question, plus its knowledge of how the agents around it have performed. Each agent also has a memory to remember the best position that it has visited [2]-[4]. A minimization of the problem is found by an agent remembering its own *p-best* position, plus its corresponding fitness. This information is an analogy of knowledge about how the other agents in the swarm have performed. The best overall position obtained by the entire particle population is called *g-best*. Consider the following position and velocity algorithms for the *i-th* particle within an *n*-dimensional search space:

$$v_{i,n}^{k+1} = K[v_{i,n}^{k} + c_1 * rand() * (pbest_{i,n} - s_{i,n}^{k}) + c_2 * rand() * (gbest_{n} - s_{i,n}^{k})] \tag{1}$$

$$s_{i,n}^{k+1} = s_{i,n}^{k} + v_{i,n}^{k+1} \tag{2}$$

$$i = 1,2,\dots, p$$
$$n = 1,2,\dots, q$$

With regards to (1) and (2) : $v_{i,n}^{k}$ = velocity of agent i at iteration k, K = constriction factor, c_1 = cognitive acceleration, c_2 = social acceleration, *rand ( )* = random number between 0 and 1, *p-best* = *p-best* of agent *i*, *g-best* = *gbest* of the group, $s_{i,n}^{k}$ = current position of agent i at iteration k, p = number of agents, q = number of parameters being optimized by the PSO algorithm.

Some popular variants of the PSO algorithm include the constriction factor approach [20] and the inertia weight approach [21]. For this study the constriction factor has been adopted because it has the ability to recover from an exploratory mode to exploitative mode and back again, whereas the time-decreasing inertia weight technique is not able to recover [4], [22]. The constriction factor approach leads to the convergence of the agents over time, that is the amplitude of the individual particle's oscillations decreases as it focuses on a previous best point [4], [22]. The constriction factor K is computed by:

$$K = \frac{2\psi}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \tag{3}$$

where

$$\varphi = c_1 + c_2 \qquad , \ \varphi > 4$$

With regards to (3), $K = 0.73$ is heuristically determined and is dependant on φ and variable ψ [22]; $\varphi = \sum(c_1, c_2)$ where c_1 and c_2 denote the cognitive acceleration and the social acceleration respectively and must be greater than 4.

III. Basic Control Theory

A. *Closed- Loops and PID Control*

The single-input single-output (SISO) servo-tracking system considered in this study is shown in Fig. 1. R(s), E(s), U(s) and Y(s) are the reference step signal, error signal, controller output and the process output, respectively. $G_c(s)$ and $G_p(s)$ denote the controller and process transfer functions, respectively. For this SISO system the main focus will be on ensuring that the dynamical response of the system to input step changes is such that the loop tracks the reference input signal. For this reason the effects of loop disturbances will not be considered in this study. The transient response for this SISO system is given by:

$$Y(s) = U(s)G_p(s) \tag{4}$$

where

$$U(s) = E(s)G_c(s) = [R(s) - Y(s)]G_c(s) \tag{5}$$

The transfer function of the closed loop system is:

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \tag{6}$$

The PID controller is implemented in the feed-forward path of the process control loop (Fig.1). The response speed of the system to dynamical conditions occurring in the process loop during a control session is proportional to the magnitude of the P-action; I-control adds a pole at the intersection of the of the real-imaginary axis, shifts system type by one and reduces system steady state error following the application of a step input signal; D-action adds a finite zero to the plant transfer function and helps to damp the system response. The transfer function for a one degree-of-freedom (1-DOF) PID controller is:

$$\frac{U(s)}{E(s)} = K_c\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{7}$$

where K_c, T_i and T_d represent the controller's gain and its integral and derivative time constants, respectively. *U(s)* and *E(s)* denote the Laplace transform of the controller's output signal and it's error input signal, respectively. For 2-DOF control loops, the D-controller is included in the feedback path and acts on the output signal in order to reduce system sensitivity to set-point changes:

$$U(s) = K_c\left(1 + \frac{1}{T_i s}\right)E(s) - \frac{sT_d}{1 + sT_d / N}Y(s) \tag{8}$$

With regards to (8), the gain (N) of the low pass filter in the feedback loop is set to 10 and helps to eliminate system sensitivity to set-point changes. *Y(s)* denotes the system output; the rest of the terms have the same meaning as defined for (7).

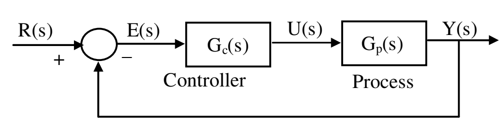

Fig. 1. SISO system

B. Process Models Considered

Typical process models represented by equation (9)-(12) were chosen for this study. These processes are commonly encountered in control applications and were selected in order to determine the efficacy of the proposed PSO based tuning technique. These models capture the dynamics usually found in process control loops.

i) First order plus dead-time model (FOPDT):

$$G_p 1(s) = \frac{Ke^{-Ls}}{Ts+1} \tag{9}$$

ii) Second order plus dead time system (SOPDT):

$$G_p 2(s) = \frac{Ke^{-Ls}}{(Ts+1)^2} \tag{10}$$

iii) Second order integrating plus dead time process (SOIPDT):

$$G_p 3(s) = \frac{Ke^{-Ls}}{s(Ts+1)} \tag{11}$$

iv) First order delayed unstable process (FODUP):

$$G_p 4(s) = \frac{Ke^{-Ls}}{(Ts-1)} \tag{12}$$

With regards to equation (9)-(12): *K*, *T*, *exp (-Ls)* and *s* represents the process gain, process time constant, control channel dead-time and Laplace operator, respectively. FOPDT, SOPDT and SOIPDT processes are usually controlled using PID control or variants thereof; tuning can be performed in either open-loop or closed-loop mode. FODUP processes having right-hand poles are inherently unstable under open loop conditions [11]. The process dead-time present in the control channel for the afore-mentioned systems represented by (9)–(12) can have a critically destabilizing effect on system performance for $\frac{L}{T} > 1$.

IV. THE PSO TUNING METHODOLOGY

From Fig. 2, the position of individual agents within the search space represents a potentially optimal tuning parameter for the closed loop system. Whilst each agent within the swarm traverses the search space in order to find the sub-optimal or optimal K_c, T_i or T_d tuning parameters, it interacts with its environment and other agents of the swarm, assuming *p-best* positions and eventually a *g-best* position. For PI control each agent is given an initial position within a two-dimensional search space. This position represents K_c and T_i. Similarly, for PID control the space allocated for each agent's position within a three-dimensional search space represents K_c, T_i and T_d. Modification of each agent's position is realized by velocity and position information according to equations (1) and (2). Each agent flying in the search space has knowledge of its *p-best* position and the *g-best* position of the group.

For this study, the optimal tuning parameters for the processes under consideration is determined according to the integral-of-time-multiplied by absolute error performance

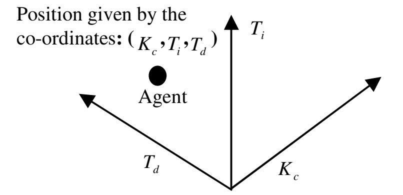

Fig. 2. Swarm agents in 3-dimensional search space

index (ITAE) in order to penalize large overshoots and damp system response following an input step signal. The efficacy of each agent's position is evaluated according to the ITAE performance index (13).

$$ITAE = \int_0^\infty t|r(t) - y(t)|dt = \int_0^\infty t|e(t)|dt \tag{13}$$

The magnitude of the performance index is determined by the tuning parameters, which correspond directly to the position of an agent within the defined search space. The position of the PSO tuning algorithm within a SISO system is shown in Fig. 3. The steps followed by the PSO tuning methodology to determine the optimal parameters for the controller in this system are:

Step 1: Initialize the swarm population and set the lower and upper search bounds for the search. Randomly select the initial position and velocity for each agent within the selected search space and start the search.

Step 2: Calculate the ITAE performance index according to equation (13) for each agent in the population as it traverses the search space.

Step 3: The best position of each agent is the *p-best* solution. The agent with the lowest ITAE value amongst the population provides the *g-best* solution.

Step 4: Modify each agents velocity and position information according to equation (1) and (2).

Step 5: Evaluate each agent's position using the ITAE index and compare this with the index from it previous *p-best* position. Save the *p-best* and *g-best* if an improvement has been accomplished.

Step 6: Repeat the above steps until all iterations are completed.

Step 7: The last saved *g-best* represents the optimal tuning parameters for the controller.

V. IMPLEMENTING THE PSO TUNING METHODOLOGY

A. Preliminaries for the Experiments

The FOPDT, SOPDT, SOIPDT and FODUP models used in the following experiments are given in equations (9)-(12). The control strategy followed in the experiments is shown in Fig. 3. The comparison between the conventional tuning techniques and the proposed PSO tuning methodology is based upon the loop's transient response specifications following the application of a step input signal. The specifications considered are: rise time (t_r), 2% settling time (t_s) and the percentage overshoot (M_p%).

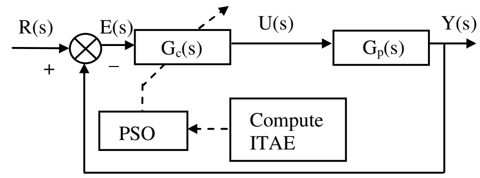

Fig. 3. SISO system with PSO optimisation algorithm

The applicable equations defining the relationships existing in the control loop is given by (4)-(6). The following PSO parameters were utilized for all the simulation tests: population size (p) = 20; maximum number of iterations = 100; cognitive acceleration (c_1) = 2.05; social acceleration (c_2) = 2.05; upper bound of initialization = 1; lower bound of initialization = 0. All experiments were conducted using a Pentium 4 personal computer having 1 giga-byte of random access memory.

B. Experiment 1: FOPDT Plant

The FOPDT process model used to test the efficacy of the proposed PSO tuning methodology is:

$$G_p(s) = \frac{1e^{-0.2s}}{(s+1)} \tag{14}$$

The PID control structure is given by equation (7) and searching occurs within a 3-dimensional search space. The tuning parameters used in the experiments are given in Table 1 and the response of the system to the step input signal is shown in Fig. 4. The proposed PSO tuning methodology was compared with the techniques of [7] and [8].

Observations and Analysis of Results for Experiment 1:

The Z-N method [7] delivers marginally quicker rise and settling times but with large initial overshoot. The Cohen-Coon method [8] positions dominant poles that yield a quarter amplitude decade ratio, resulting in oscillation and an increased settling time. Overall the PSO tuned controller delivers an improved response when compared to methods [7] and [8].

C. Experiment 2: SOPDT System

Consider the following SOPDT plant model:

$$G_p(s) = \frac{1e^{-0.5s}}{(s+1)^2} \tag{15}$$

with L/T = 0.5.

The method of [8] was designed for PID control of first-order systems and is not suited to SOPDT processes. The comparison of tuning methodologies was based on the proposed PSO tuning method and the Åström-Hägglund (A-H) tuning method [14] since it is suited for systems having small L/T ratios. The Z-N method [7] is also included in this experiment because of its popularity amongst control practitioners. The tuning parameters plus the response specifications for the tests are given in Table 2; the closed-loop response following a step input signal is shown in Fig.5.

Observations and Analysis of Results for Experiment 2:

Z-N tuning [7] delivers a response having large overshoot. The A-H method [14] yields an oscillatory response with long settling time, not withstanding the small L/T ratio for this system. Overall the proposed PSO based PID tuning method results in an improved closed loop response with faster settling time and minimum overshoots.

D. Example 3: SOIPDT Process

PI control is used to control the following plant model:

$$G_p(s) = \frac{1e^{-0.2s}}{s(s+1)} \tag{16}$$

The comparison is based on the Poulin-Pomerleau (P-P) method [11] and the proposed PSO tuning method. The swarm's search for the controller parameters is performed within a 2-dimensional search space. The tuning parameters used in the experiments are given in Table 3 and the SOIPDT system's transient response to a step signal input appears in Fig. 6.

Observations and Analysis of Results for Experiment 3:

The method of [11] was designed for this type of process and yields good results. However its main drawback is that the maximum peak resonance of the system has to be first determined.

TABLE 1
PID PARAMETERS AND RESPONSE SPECIFICATIONS FOR FOPDT PROCESS

Tuning Method	PID Parameters			Response Specifications			Performance Index
	K_c	T_i	T_d	t_r	t_s	$M_p(\%)$	*ITAE*
ZN	5.1	0.37	0.09	0.1	2.9	57.8	40.7
CC	6.92	0.45	0.07	0.1	3.2	84.0	52.3
PSO	3.64	1.29	0.08	0.3	1.9	0.8	30.3

TABLE 2
PID PARAMETERS AND RESPONSE SPECIFICATIONS FOR SOPDT PROCESS

Tuning Method	PID Parameters			Response Specifications			Performance Index
	K_c	T_i	T_d	t_r	t_s	$M_p(\%)$	*ITAE*
ZN	2.8	1.64	0.4	0.7	5.7	43.7	50.9
AH	3.13	2.51	0.63	0.6	8.2	15.6	44.5
PSO	1.43	2.42	0.43	1.6	4.1	1.4	44.6

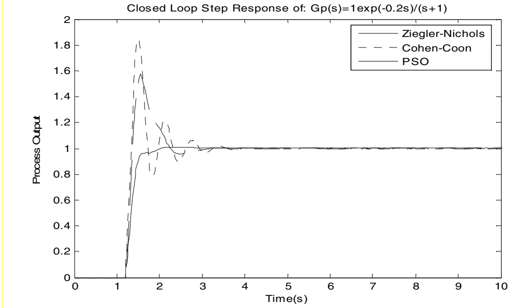

Fig. 4. FOPDT system response

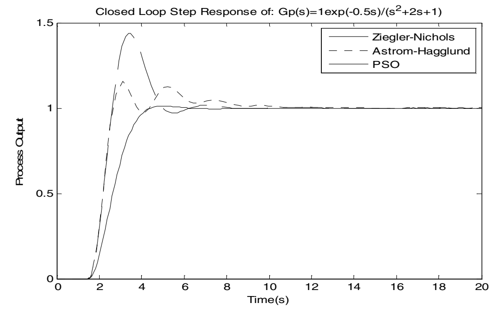

Fig. 5. SOPDT system response

The PSO tuned PI controller delivers improvements in the system's rise time and peak response.

E. Experiment 4: FODUP Plant Model

Control loops having open-loop unstable poles are notoriously difficult to control. The difficulties are mostly due to the unstable nature of the dynamics, for which most conventional design tools cannot be used [15]. For this experiment a comparison is made between the proposed PSO tuning method and the tuning methods for unstable processes as proposed by Venkatashankar and Chidabaram (VC) [10] and De Paor and O'Malley (DO) [9]. The unstable process model chosen for this experiment is given in equation (17):

$$G_p(s) = \frac{1e^{-0.2s}}{(s-1)} \quad (17)$$

where the unstable pole located on the right-hand plane of the real axis. For these types of processes the objective of the control philosophy should include stabilizing the right-hand pole in order to achieve satisfactory control [15]. The tuning parameters plus the response specifications are shown in Table 4 and the response to the input step signal is given in Fig. 7.

Observations and Analysis of Results for Experiment 4:

Analysis of the results reveals that the tuning method of [9] provides oscillations and large overshoot. Conversely, the method of [10] delivers no oscillations but suffers from long settling time. The PSO tuned PI controller generates a superior control performance in terms of improved rise time and settling time, with a marginally larger overshoot than the method of [10].

TABLE 3
TUNING PARAMETERS AND RESPONSE SPECIFICATIONS FOR SOIPDT PROCESS

Tuning Method	PI Parameters		Dynamic Performance Indices			Performance Index
	K_c	T_i	t_r	t_s	$M_p(\%)$	*ITAE*
PP	0.465	6.09	1.9	14	37.5	49.3
PSO	0.79	11.1	1.5	14	32.6	43.7

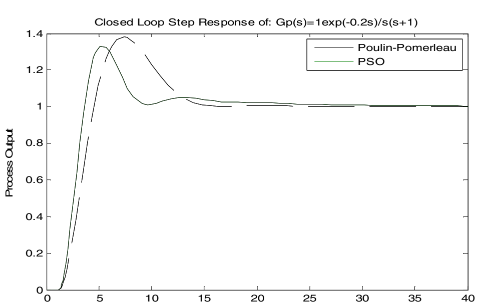

Fig.6. SOIPDT system response

TABLE 4
PI PARAMETERS AND RESPONSE CHARACTERISTICS FOR FODUP PROCESS

Tuning Method	PI Parameters		Dynamic Performance Indices			Performance Index
	K_c	T_i	t_r	t_s	$M_p(\%)$	*ITAE*
DO	1.7	1.35	0.3	15	122.2	115.2
VC	2.4	19.6	0.3	33	67.3	191.6
PSO	3.56	2.27	0.2	4.1	87.6	54.4

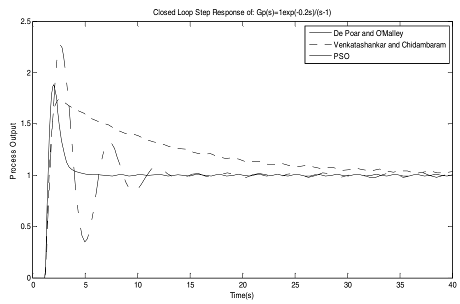

Fig. 7. FODUP system response

V. CONCLUSION

This paper has proposed a tuning methodology based on the PSO optimisation algorithm for use with the selected process models. The unique characteristic of the proposed PSO based tuning method is that it is completely model independent. The simulation experiments performed to assess the efficacy of the technique may yield good results for all the process models considered in the study, and can also generally deliver better results than those obtained by earlier works for these process models. Unlike the other methods, the technique is simple, fast and easy to implement in a variety of control loops and yields much better results than currently available tuning methods.

ACKNOWLEDGEMENT

This work is supported by the NRF of SA under Grant Number: ICD2006062700006.

REFERENCES

[1] R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejection PID controllers using genetic algorithm," *IEEE Trans. Evol. Computation,* Vol. 5, pp. 78-82, 2001

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. Neural Networks*, Vol. 4, pp. 1942-1948, 1995

[3] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimisation," *Proc.-7th Annual Conference on Evolutionary Programming*, pp. 591-601, 1998

[4] R. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," *Proc. Congress on Evolutionary Computation (CEC2000), pp.84*-88, 2000

[5] M. Zhuang and D.P. Atherton, "Automatic tuning of optimum PID controllers," *IEE Proc.-D*, Vol. 140, No.3, pp. 216-224, 1993

[6] Z.L. Gaing, "A Particle Swarm Optimization approach for Optimum Design of PID controller in AVR System," *IEEE Transactions on Energy Conversion*, Vol.19, No.2, pp. 284-291, 2004

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

[8] G.H. Cohen and G.A. Coon, "Theoretical consideration of retarded control," *Trans.-ASME*, Vol. 75, pp. 827-834, 1953

[9] A.M. De Paor and M. O'Malley, "Controllers of Ziegler-Nichols type for unstable process with time delay," *Int. J. Control*, Vol. 49, pp. 1273-1284, 1989
[10] V. Venkatashankar and M. Chidambaram, "Design of P and PI controllers for unstable process with time delay," *Int. J. Control*, Vol. 60, pp. 1367-144, 1994
[11] E. Poulin and A. Pomerleau, "PID tuning for integrating and unstable processes," *IEE Proc.-Control Theory Appl.*, Vol. 143, pp. 429-435, 1996
[12] S. Madji and D.P. Atherton, "Modified Smith Predictor and controller for processes with time delay," *IEE Proc.-Control Theory Appl.*, Vol. 146, pp. 359-366, 1999
[13] C. Hang, K.J. Åström, and W.K. Ho, "Refinements of the Ziegler-Nichols Tuning Formula," *IEE Proceedings-D*, Vol. 138, No. 2, pp. 111–118, 1991
[14] K.J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specification on phase and amplitude margins," *Automatica*, Vol. 20, No. 5, pp. 645-651, 1984
[15] H.P. Huang and C.C. Chen, "Control-system synthesis for open-loop unstable process with time delay," *IEE Proc.-Control Theory Appl.*, Vol. 144, No. 4, pp. 334-345, 1996
[16] C.C. Kao., "The Self-Tuning PID Controller in a Slider-Crank Mechanism System by applying Particle Swarm Optimisation Approach," *Mechatronics*, 16, pp. 513 -522, Elsevier, 2006
[17] S.P. Ghoshal, "Optimisations of PID Gains by Particle Swarm Optimisations in Fuzzy Based Automatic Generation Control," *Electric Power Systems*, 72, pp. 203 – 212, Elsevier, 2004
[18] V. Mukherjee and S.P. Ghoshal, "Intelligent Particle Swarm Optimised Fuzzy PID Controller for AVR System," *Electric Power Systems Research*, xx, pp. xxx – xxx, Elsevier, 2007
[19] A.A. El-Dib, H.K.M. Yousef, M.M. El-Metwally, Z. Osman, "Optimum VAR Sizing and Allocation using Particle Swarm Optimisation," *Electric Power Systems Research*, xx, pp. xxx – xxx, Elsevier, 2006
[20] M. Clerc and J. Kennedy, "The Particle Swarm – Explosion, Stability and Convergence in a Multidimensional Complex Space," *IEEE Trans. Evol. Computation,* Vol. 6, No.1, pp. 58 - 73, 2002
[21] Y. Shi and R.C. Eberhart, "A Modified Particle Swarm Optimiser," *Proc.-IEEE Int'l Conf. on Evol. Computation*, pp. 69 – 73, Piscataway, NJ: IEEE Press
[22] J. Kennedy and R.C. Eberhart, "Swarm Intelligence", *Morgan Kaufmann Publishers,* 2001

Copyright Information