

Controlo 2002

5th Portuguese Conference on Automatic Control



September 5-7, 2002 • Universidade de Aveiro • Portugal

PLEASE NO MORE PID TUNING RULES

P. B. de Moura Oliveira, J. Boaventura Cunha and J. P. Coelho

*Departamento de Engenharias
Universidade de Trás-os-Montes e Alto Douro
5001-911 Vila Real
Portugal
Tel: +351 59 350339 Fax: + 351 59 350480
e-mail: oliveira@utad.pt*

Abstract: Despite the large number of existent design and tuning techniques, adequate PID controller tuning by plant operators is still not accomplished in many process control loops. The particle swarm optimisation algorithm is proposed as an alternative technique to design and tune PID controllers for linear single-input single-output systems. This evolutionary approach is illustrated by a simulation example, in which the PID is used to control a set of models that represents a wide variety of process dynamics, regarded as benchmark problems for the evaluation of commercial devices involving PID control.
Copyright © Control 2002

Keywords: PID control, Tuning based on optimisation, Process Control, Particle Swarm Optimisation, Evolutionary Algorithms.

1. INTRODUCTION

The PID controller is the most popular controller used in industrial control applications. The overwhelming dominance of PID controllers over other forms of feedback, in the last fifty years (Bennet, 2000), is due to its simple structure and reliability in a wide range of operating conditions. Some impressive estimates state that more than 95% of the controllers used in process control applications are of PID type (Åström and Hägglund, 1995) and 98% of the control loops in the pulp and paper industry are controlled by PI controllers (Bialkowski, 1996; O'Dwyer, 2000a). Important considerations about the state of the art of PID control and possible perspectives of use of this type of control were reported by Åström and Hägglund (2000a).

Due to the wide acceptance of PID controllers within industry many tuning rules have been proposed for this type of controller, since the original work of

Ziegler and Nichols (1942). An extensive list of PI (O'Dwyer, 2000a) PID (O'Dwyer, 2000b) tuning rules for processes with time delay, has been selected for publication from a set of surveyed published literature comprising 154 and 258 different PI and PID tuning rules, respectively. However, despite the huge amount of existing tuning rules for PI/PID controllers, the test of a large set of industrial plants (Ender, 1993), indicated that 30% of the controllers were operated manually and 65% were poorly tuned. Indeed, plant operators tend to tune PID controllers by trial and error or using very simple tuning rules. A plausible explanation is the lack of appropriate educational background from most of the process control operators (Pomerleau and Poulin, 2002) concerning the proper use of tuning methodologies. In some cases, the control operator can be responsible for hundreds of process control loops (Åström and Hägglund, 2000a), with a wide range of system dynamics, having to design and tune PID controllers to meet performance and robustness specifications.

Considering
proposed for t
amount of eff
each loop, t
method wou
Because there
use of an opt
tool to the de
wide range of

Evolutionary
(GAs) (Holl
population l
Baluja, 199
optimisation
controllers ar
and Pires, 199
obtained in so
problems, na
system desig
been consid
developing n
natural conce
algorithms i
algorithm (K
proposed to
controllers by
controllers (C

This paper i
optimisation
linear PID co
The paper is
introduction
ations of the
In section th
alternative e
single-input
this shows :
single model
systems w
applications.
shown

2. PAR

The particle
was proposec
is conceptua
groups of o
backs. As ar
population (s
of particles,
given proble
It solves th
model in a fl
 $\dot{x}(t) = (v_{i1}(t)$
 $\dot{v}_i(t) = (v_{i1}(t)$
Evolutionary

Considering the huge number of tuning rules proposed for the different process models, the limited amount of effort that the plant operator can devote to each loop, the existence of an universal tuning method would simplify their role significantly. Because there is not an universal tuning method, the use of an optimisation algorithm constitutes a global aid to the design and tuning of PID controllers for a wide range of process control applications.

Evolutionary techniques such as genetic algorithms (GAs) (Holland, 1975; Goldberg, 1989), and population based incremental learning (PBIL) (DeJong, 1994), have proved to be excellent optimisation tools to design and tune linear PID controllers and non-linear PID controllers (Oliveira and Pires, 1999). Despite of the success that GAs had obtained in solving complex optimisation engineering problems, namely in the computed aided control system design during the last ten years, there has been considerable research effort directed into developing new optimisation algorithms inspired on natural concepts. One of the most recently introduced algorithms is the particle swarm optimisation algorithm (Kennedy and Eberhart, 1995) that was proposed to design both linear and non-linear PID controllers by (Oliveira *et al.*, 2002) and predictive controllers (Coelho *et al.*).

This paper reports the use of the particle swarm optimisation algorithm to the design and tuning of linear PID controllers in the continuous time domain. The paper is divided into five sections. Following the introduction the second section provides basic notions of the particle swarm optimisation algorithm. In section three the PSO algorithm is proposed as an alternative evolutionary design technique to linear single-input single output PID controllers. Section four shows simulation results obtained for a wide range models that represent classical benchmark systems within industrial process control applications. In section five some conclusions are drawn.

2. PARTICLE SWARM OPTIMISATION ALGORITHM

The particle swarm optimisation (PSO) algorithm was proposed by Kennedy and Eberhart (1995) and it is conceptually based on the social behaviour of groups of organisms such as herds, schools and flocks. As an evolutionary based technique PSO is a population (swarm) based algorithm, formed by a set of particles, which represent potential solutions for a given problem. Each particle of a set of swarm size m , moves through a n -dimensional search space (as birds in a flock), with an associated position vector $X_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{in}(t))$ and velocity vector $V_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{in}(t))$ for the current evolutionary iteration t .

The original PSO model integrates two types of knowledge acquisition by a particle: through its own experience and from social sharing from other population members. The former was termed *Cognition-Only Model* and the latest *Social-Only Model* (Kennedy and Eberhart, 1997). The behaviour of each particle is based on these two types of knowledge and their current position regarding the search. Kennedy modelled particle behaviour by using the following two equations:

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + \varphi_1 \cdot (p_{id}(t) - x_{id}(t)) + \varphi_2 \cdot (p_{gd}(t) - x_{id}(t)) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \quad \begin{matrix} (1) \\ (2) \end{matrix}$$

in which d represents the dimension index, $1 \leq d \leq n$, $p_{id}(t)$ represents the best previous position of particle i at the current iteration t , $p_{gd}(t)$ represents the global best, in the current iteration, for a pre-defined neighbourhood type. Parameter φ_1 is known as the *Cognitive Constant* and φ_2 as the *Social Constant* and represent uniformly distributed random numbers generated in a pre-defined interval.

An additional parameter was incorporated into equation (1) (Shi and Eberhart, 1999): resulting in equation (3):

$$v_{id}(t+1) = \omega v_{id}(t) + \varphi_1 \cdot (p_{id}(t) - x_{id}(t)) + \varphi_2 \cdot (p_{gd}(t) - x_{id}(t)) \quad (3)$$

in which ω was termed *Inertia Weight*. The value given to the inertia weight will affect the type of search in the following way: a large inertia weight will direct the PSO for a global search while a small inertia weight will direct the PSO for a local search. This parameter can vary linearly from a larger value to a smaller value in order to make the search global in the early run and local in the end of the run. The constants φ_1 , φ_2 and ω can be interpreted as the confidence that each particle has in its current position, its own experience and its neighbours experience, respectively. The neighbourhood can be of different size and topology. Each particle can take into account either:

(i) the social information from a list of particles pre-defined in the beginning of the simulated evolution. This list can incorporate all the population individuals, with an individual being able to use the best solution found by every other member. This full-connected social network structure was termed *Star*. In other list definitions, an individual uses only k adjacent neighbours organised in a *Circle* and *Wheel* topology.

(ii) the physical information which considers distance between neighbour individual evaluated using some metric definition.

The velocity is limited by a maximum, V_{max} , meaning the maximum jump that each particle can make in each iteration. The selected value for V_{max} should not be too high to avoid oscillations, or too low to avoid

e list of PI
0b) tuning
en selected
l published
ent PI and
despite the
for PI/PID
trial plants
controllers
orly tuned.
controllers
ing rules. A
appropriate
the process
lin, 2002)
odologies.
responsible
Åström and
of system
controllers
ecifications.

search traps. The inertia weight and maximum velocity parameters selection in the PSO algorithm was studied and reported by Shi and Eberhart (1998). Each particle position should also be located within its dynamic range $[-X_{min}, X_{max}]$.

3. PID CONTROLLER DESIGN USING THE PARTICLE SWARM OPTIMISATION ALGORITHM

The PID controller can be governed in the continuous time domain by equation (4):

$$u(t) = K_p \left[e(t) + K_i \int_0^t e(t) dt + T_d \frac{dy_f(t)}{dt} \right] \quad (4)$$

in which, $e(t) = v(t) - y(t)$ is the error signal, $v(t)$ the reference signal, $y(t)$ and $y_f(t)$ are the system output non-filtered and filtered, respectively, K_p and K_i represent the proportional and integral gains, respectively and T_d the derivative time constant. The derivative action in this case is applied to the system output filtered using equation (5), in which N represents the filter constant.

$$\frac{T_d}{N} \frac{dy_f(t)}{dt} + y_f(t) = y(t) \quad (5)$$

In order to use the PSO algorithm to optimise a control system with a PID controller it is necessary to encode the tuning parameters. Thus each population member represents the proportional, integral and derivative parameters, by using a real-based coding scheme. The PID controller design is accomplished by optimising the system response to a unit set-point change by minimising a time-domain performance criterion. In this study the *Integral of product of Time and Absolute value of the Error (ITAE)* was adopted.

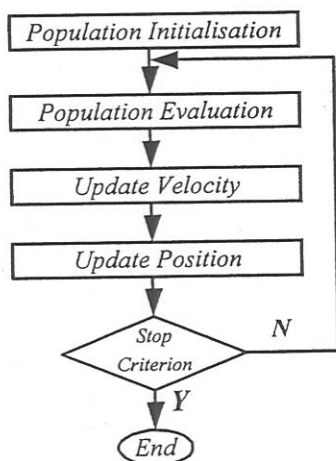


Fig. 1. Basic PSO algorithm.

Each particle in the population is represented by vector (6):

$$(K_p, K_i, T_d) \quad (6)$$

and are evolved, in this case, from a randomly initialised population. The simple PSO algorithm loop can be illustrated by Figure 1, in which the particles velocities and positions are updated using equations (2) and (3), respectively.

4. SIMULATION RESULTS

For evaluating the performance of the proposed technique to design and tune linear SISO PID controllers a set of benchmark systems used for research and evaluation of commercial systems was selected. Models represented by equations (7-11) and (14) were extracted from a collection published by Åström and Hägglund (2000b), while models (12-13) were extracted from (Kiong *et al.*, 1999). It is important to state, that for some of these systems PID control is not the most appropriate form of control.

The objective of these examples is to design a continuous linear PID controller by optimising the system response to a unit set-point change. The optimisation criterion used is the *ITAE*. The population size used in the PSO algorithm is $n=50$. The inertia weight ω is set to change linearly in the interval $[0.7, 0.4]$ during the evolution through 75 epochs. Parameters ϕ_1 and ϕ_2 are uniformly distributed random numbers generated in the interval $[0, 1]$. The adjustments of the velocity position vectors are done using expressions (2) and (3). The search space for the controller parameters was defined in the interval $[0, 5]$ for each dimension. In all the presented figures the x-axis represents the simulation time in seconds, the solid and dashed lines represent the system and PID controller outputs, respectively.

4.1 Multiple Equal Poles

$$G_{p1}(s) = \frac{1}{(s+1)^n} \quad n = 1, 2, 3, 4, 8 \quad (7)$$

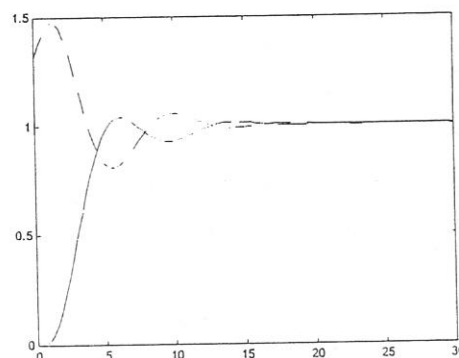


Fig. 2. Plant 1, $n=4$, $[K_p=1.32; K_i=0.14; T_d=0.33]$.

For high systems v control in were sele in Figures

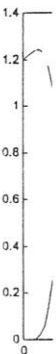


Fig. 3. Pl2

4.2 Four

$$G_{p2}(s)$$

Models w simulated

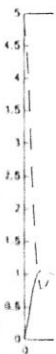


Fig. 4. Pl4



Fig. 5. Pl5

(6)

andomly
orithm
ch the
l using

roposed
O PID
sed for
ms was
(11) and
shed by
(12-13)
) It is
ms PID
ntrol.

esign a
sing the
ge. The
E. The
is $n=50$.
ly in the
ough 75
niformly
interval
position
(3). The
ers was
on. In all
ents the
hed lines
outputs,

(7)

=0.33].

For high values of n , the system (7) behaves as systems with long dead-times, with the difficulty of control increasing with n . Models with $n=4$ and $n=8$ were selected and the simulated responses are shown in Figures 2 and 3.

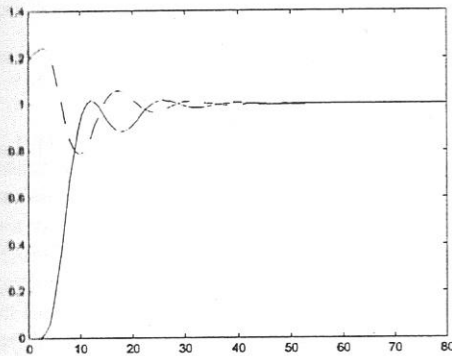


Fig. 3. Plant 1, $n=8$, [$K_p=1.2; K_i=0.02; T_d=0.70$].

4.2 Fourth Order System

$$G_{p2}(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)} \quad (8)$$

$\alpha = 0.1, 0.2, 0.5, 1.0$

Models with $\alpha=0.1$ and $\alpha=0.2$ were selected and the simulated responses are shown in Figures 4 and 5.

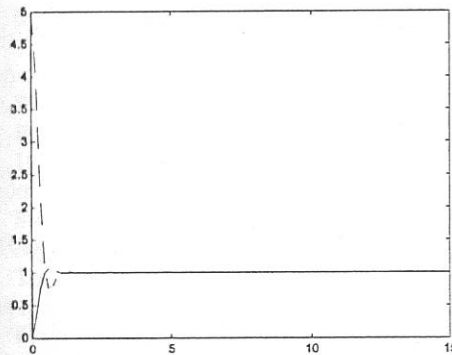


Fig. 4. Plant 2, $\alpha=0.1$, [$K_p=5.00; K_i=1.00; T_d=0$].

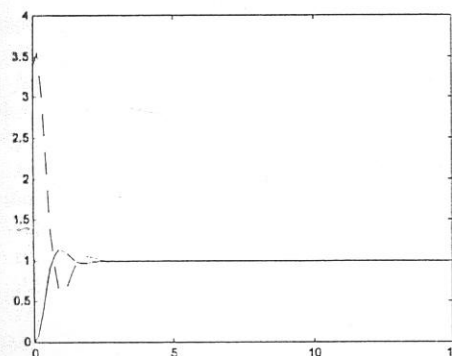


Fig. 5. Plant 2, $\alpha=0.2$, [$K_p=5.38; K_i=0.92; T_d=0$].

4.3 Non-minimum Phase System

$$G_{p3}(s) = \frac{1-\alpha s}{(s+1)^3} \quad \alpha = 0.1, 0.2, 0.5, 1.0, 2.0, 5.0 \quad (9)$$

In this case the difficulty of control increases with α . Models with $\alpha=0.5$ and $\alpha=2.0$ were selected and the simulated responses are shown in Figures 6 and 7.

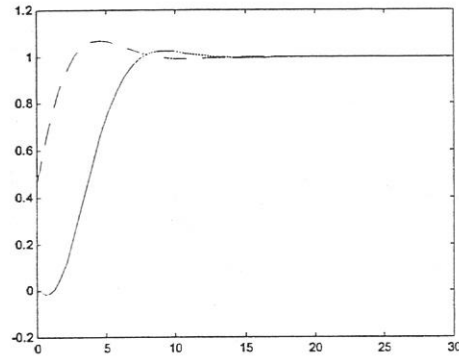


Fig. 6. Plant 3, $\alpha=0.5$, [$K_p=0.46; K_i=0.55; T_d=0.00$].

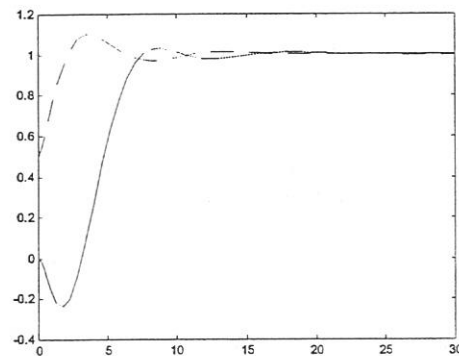


Fig. 7. Plant 3, $\alpha=2.0$, [$K_p=0.49; K_i=0.34; T_d=0.21$].

4.4 Time Delay and Double Lag

$$G_{p4}(s) = \frac{1}{(1+sT)^2} e^{-sT} \quad T = 0, 0.1, 0.2, 0.5, 1, 5, 10 \quad (10)$$

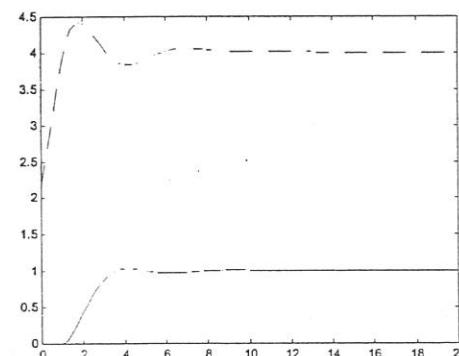


Fig. 8. Plant 4, $T=1.0$, [$K_p=2.16; K_i=0.81; T_d=0.13$].

Model with $T=1.0$ was selected and the simulated response is shown in Figure 8.

4.5 Fast and Slow Modes

$$G_{p5}(s) = \frac{100}{(s+10)^2} \left(\frac{1}{s+1} + \frac{0.5}{s+0.05} \right) \quad (11)$$

The simulated response for this case is shown in Figure 9.

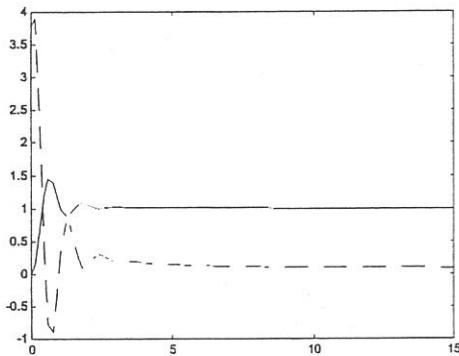


Fig. 9. Plant 5, $[K_p=3.79; K_i=1.32; T_d=0]$.

4.6 Conditionally Stable System

$$G_{p6}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)} \quad (12)$$

The simulated response for this case is shown in Figure 10.

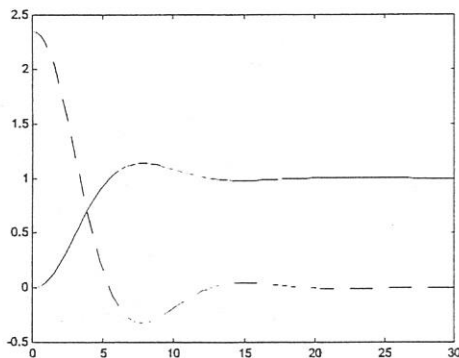


Fig. 10. Plant 6, $[K_p=3.79; K_i=1.32; T_d=0]$.

4.7 Oscillatory System: Case 1

$$G_{p7}(s) = \frac{1}{(s^2 + 2s + 4)(s+3)} e^{-0.1s} \quad (13)$$

The simulated response for this case is shown in Figure 11.

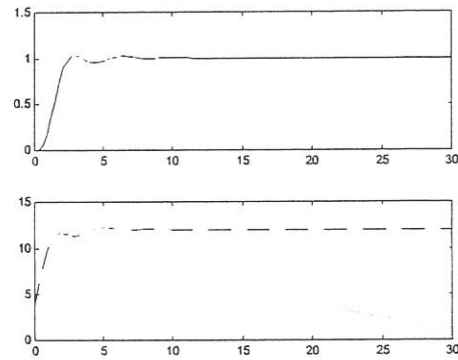


Fig. 11. Plant 7, $[K_p=5.00; K_i=1.13; T_d=0.09]$.

4.8 Oscillatory System: Case 2

$$G_{p8}(s) = \frac{1}{(s^2 + 2s + 3)^3} e^{-s} \quad (14)$$

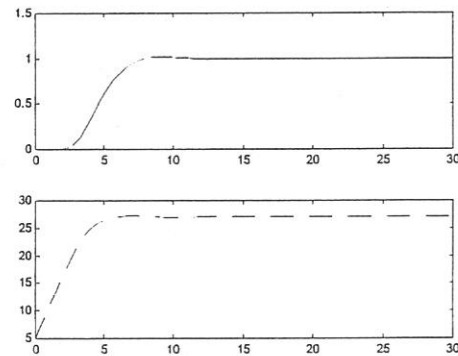


Fig. 12. Plant 8, $[K_p=3.71; K_i=2.09; T_d=0.28]$.

4.9 Unstable Pole

$$G_{p9}(s) = \frac{1}{(s^2 - 1)} \quad (15)$$

This transfer function represents a simplified model of an inverted pendulum or an unstable batch reactor (Åström and Häggglund, 2000b). The simulated response for this case is shown in Figure 13.

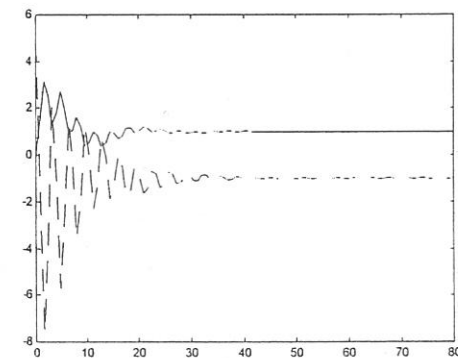


Fig. 13. Plant 9, $[K_p=4.99; K_i=0.21; T_d=0.80]$.

The results tune PID c systems. It pre-defined better alte derivative l

The partic proposed : controller (output syst algorithm i controllers, select the z select the (be applied most usual PSO algori than a gene algorithm (configurati identificati or off-line, dynamics z

Åström K
Contr
Instru
Trian
Åström K.
PID (
Past,
19-30
Åström K.
Syste
Digita
Terra
Baluja S.,
Learn
search
Comp
CS-9.
Carne
Bennet S.,
Work
future
Baskowsk
paper
Hana
Cuelho J. I
(2002
using
Algo
IFAC
de Moura
(1999
Disce

The results indicate that the PSO is able to design and tune PID controllers for a wide variety of different systems. It is relevant to note that for the cases in the pre-defined search interval, the PI controller is a better alternative than the PID controller, the derivative time constant converges to zero.

5. CONCLUSION

The particle swarm optimisation algorithm was proposed as an alternative technique to the PID controller design and tuning for single-input single-output systems. Simulation results show that the PSO algorithm is a good alternative to the design of such controllers, due to the following reasons: (i) It can select the appropriate correct PID control modes and select the corresponding tuning parameters; (ii) Can be applied to a large variety of plants, for which the most usual tuning rules are inappropriate; (iii) The PSO algorithm used is yet of simpler implementation than a genetic algorithm; (iv) This evolutionary based algorithm can be easily incorporated in a auto-tuning configuration in the digital domain, in which system identification and tuning can be accomplished on-line or off-line, depending on the corresponding process dynamics and sampling time used.

REFERENCES

- Aström K. J. and Hägglund T. (1995). *PID Controllers: Theory, Design and Tuning*, Instrument Society of America, Research Triangle Park, 2nd edition.
- Aström K. J. and Hägglund T. (2000a). The Future of PID Control, *IFAC Work. on Digital Control: Past, present and future*, Spain, Terrassa, pp. 19-30.
- Aström K. J. and Hägglund T. (2000b). Benchmark Systems for PID Control, *IFAC Work. on Digital Control: Past, present and future*, Spain, Terrassa, pp. 181-182.
- Baluja S., (1994). Population Based Incremental Learning: A Method for Integrating Genetic search Based Function Optimization and Competitive Learning, Technical report CMU-CS-95-163, School of Computer Science, Carnegie Melon University, USA.
- Bennet S., (2000). The Past of PID Controllers, *IFAC Work. on Digital Control: Past, present and future*, Terrassa, Spain, pp. 3-13.
- Dialkowski, W.-L. (1996). Control of the pulp and paper making process, *In: The Control Handbook*, pp. 1219-1242, CRC Press.
- Coelho J. P., De Moura Oliveira P. B and Cunha J. B. (2002). Greenhouse Air Temperature Control using the Particle Swarm Optimisation Algorithm, Accepted for Publication in the *15th IFAC World Congress*, July, Barcelona, Spain
- De Moura Oliveira P. B and Solteiro Pires E J (1999). Projecto de Controladores PID Discretos Nao-Lineares Usando PBIL, *XX Jornadas de Automatica*, Salamanca, Spain, pp.229-234.
- De Moura Oliveira P. B, Cunha J. B. and Coelho J. P. (2002). Design of PID Controllers using the Particle Swarm Algorithm, *Proc. of IASTED MIC'2002*, February, Innsbruck, Austria, pp. 263-268.
- Ender D. B. (1993). Process Control Performance: not as Good as you Think, *Control Engineering*, September, pp. 180-190.
- Goldberg E D, (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Adison Wesley P.C.
- Holland J. H., (1975). *Adaptation in Natural and Artificial Systems*, 1st MIT Press ed.
- Kennedy J. (1997). The Particle Swarm: Social Adaptation of Knowledge, *Proceedings of the 1997 IEEE Int. Conference on Evolutionary Computation*, Indianapolis, US, pp. 303-308.
- Kennedy J. and Eberhart R. C. (1995). Particle Swarm Optimization, *Proceedings of the 1995 IEEE Int. Conference on Neural Networks*, Perth, Australia, pp. 1942-1948.
- Kiong T. Guo-Quing W., Chieh H. and Hägglund T. (1999). *Advances in PID Control*, Springer Ed., pp. 49-51.
- O'Dwyer A. (2000a). A Summary of PI and PID Tuning Rules for Processes with Time Delay. Part 1: PI Controller Tuning Rules, *IFAC Work. on Digital Control: Past, present and future*, Terrassa, Spain, pp. 175-180.
- O'Dwyer A. (2000b). A Summary of PI and PID Tuning Rules for Processes with Time Delay. Part 2: PID Controller Tuning Rules, *IFAC Work. on Digital Control: Past, present and future*, Terrassa, Spain, pp. 242-247.
- Pomerleau A. and Poulin É. (2002). A New Approach for Teaching Process Control, *Proc. of IASTED MIC'2002*, February, Innsbruck, Austria, pp. 275-279.
- Shi Y. and Eberhart R. C. (1998). Parameter Selection in Particle Swarm Optimisation, Evolutionary Programming VII, *Lecture Notes in Computer Science 1447*, pp. 591-600, Springer Verlag.
- Shi Y. and Eberhart R. C. (1999). Empirical Study of Particle Swarm Optimization, *Proceedings of the 1999 IEEE Congress of Evolutionary Computation, Vol 3*, pp. 1945-1950.
- Ziegler J. G. and Nichols N. B., (1942). Optimum Settings for Automatic Controllers, *Transactions of the ASME*, 64, pp. 759-768.