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 existing 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 © Controle 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 (Áström 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 Áström 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) and PID (O'Dwyer, 2000b) tuning rules for processes with time delay, has been selected 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 (Endler, 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).

1.1 PAR

The particle swarm optimisation is a conceptually simple and efficient heuristic technique for optimisation of continuous functions. It is a population-based technique, which can be used to solve problems in a black-box framework. The algorithm is based on the concept of a swarm, in which each particle represents a potential solution to the problem. The particles move through the search space, adjusting their positions based on their own best position and the best position found by any particle in the swarm. This movement is guided by the particle's velocity, which is updated at each iteration according to a set of rules.
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:

\[
\begin{align*}
    v_{id}(t+1) &= v_{id}(t) + \phi_1(p_{id}(t) - x_{id}(t)) + \phi_2(p_{gb}(t) - x_{id}(t)) \\
    x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1)
\end{align*}
\]  

(1) (2)

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 \(t\), \(p_{gb}(t)\) represents the global best, in the current iteration, for a pre-defined neighbourhood type. Parameter \(\phi_1\) is known as the Cognitive Constant and \(\phi_2\) 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):

\[
v_{id}(t+1) = \omega v_{id}(t) + \phi_1(p_{id}(t) - x_{id}(t)) + \phi_2(p_{gb}(t) - x_{id}(t))
\]

(3)

in which \(\omega\) 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 \(\phi_1\) and \(\phi_2\) 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 predefined 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...
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 \([-X_{\text{min}}, X_{\text{max}}]\).

3. PID CONTROLLER DESIGN USING THE PARTICLE SWARM OPTIMISATION ALGORITHM

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

\[
u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d y_f(t)}{dt}
\]  

(4)

in which, \(e(t) = v(t) - y(t)\) is the error signal, \(v(t)\) the reference signal, \(y(t)\) and \(y_f(t)\) the system output non-filtered and filtered, respectively, \(K_p\) and \(K_i\) represent the proportional and integral gains, respectively and \(K_d\) 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.

\[
N \frac{d}{dt} y_f(t) + y_f(t) = y(t)
\]  

(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.

\[
(G_{\text{p}}(s)) = \frac{1}{(s+1)^n} \quad n = 1, 2, 3, 4, 5
\]  

(7)

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 (Kong 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 \(\omega\) is set to change linearly in the interval \([0.7, 0.4]\) during the evolution through 75 epochs. Parameters \(\phi_1\) and \(\phi_2\) 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

\[
G_{\text{p}}(s) = \frac{1}{(s+1)^n} \quad n = 1, 2, 3, 4, 5
\]  

(7)

Each particle in the population is represented by vector (6).
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, \( n=8 \), \([K_p=1.2; K_i=0.02; T_d=0.70] \).

4.3 Non-minimum Phase System

\[
G_{p3}(s) = \frac{1-\alpha s}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}
\]

\[\alpha = 0.1, 0.2, 0.5, 1.0 \] (9)

In this case the difficulty of control increases with \( \alpha \). 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

\[
G_{p4}(s) = \frac{1}{(1+sT)^2} e^{-sT} \quad T = 0.1, 0.2, 0.5, 1.0 \] (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

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

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

![Figure 9. Plant 5, $[K_p=3.79; K_i=1.32; T_d=0]$.](image)

4.6 Conditionally Stable System

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

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

![Figure 10. Plant 6, $[K_p=3.79; K_i=1.32; T_d=0]$.](image)

4.7 Oscillatory System: Case 1

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

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

![Figure 11. Plant 7, $[K_p=5.00; K_i=1.13; T_d=0.09]$.](image)

4.8 Oscillatory System: Case 2

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

The simulated response for this case is shown in Figure 12.

![Figure 12. Plant 8, $[K_p=3.71; K_i=2.09; T_d=0.28]$.](image)

4.9 Unstable Pole

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

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

![Figure 13. Plant 9, $[K_p=4.99; K_i=0.21; T_d=0.80]$.](image)
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 predefined 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


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 13th IFAC World Congress, July, Barcelone, Spain.


