Particle Swarm Metaheuristic Applied to the Optimization of a PID Controller

John A. Ramirez¹, Jesús M. López-Lezama² and Nicolás Muñoz-Galeano²

¹Grupo de Investigación ICARO, Ingeniería en Instrumentación y Control, Facultad de Ingenierías, Politécnico Colombiano Jaime Isaza Cadavid
Cr. 48 # 7-151, Medellín, Colombia

²Grupo de Investigación GIMEL, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Antioquia, Calle 67 No 53-108, Medellín, Colombia

Abstract

This paper presents the optimization of a PID controller by means of particle swarm optimization metaheuristic. Each particle corresponds to a vector that contains the gains of the PID multimodal actions. The controller is implemented on a scale plant designed for air pressurization in a storage tank. The results obtained are compared with an advanced predictive DMC controller, showing the superiority of the proposed methodology to optimize the PID controller.

Keywords: Particle swarm optimization, PID controller, metaheuristics

1 Introduction

There are processes and applications in the industry that need to be controlled efficiently, even in adverse conditions. In consequence, it is necessary to design and implement controllers that allow automatic regulation, such as PID strategies. Due to their versatility, PID control algorithms (Proportional, Integral, Derivative) are the most used in the industry; they can be configured as P, PI, PD or PID. They are also simple and easy to implement on only in industrial controllers but also through PLC (Programmable Logic Controller).

There are several adjustment and design methodologies reported in the specialized literature for the optimal tuning of PID controllers [1], [2]. Controller tuning tables fail to meet the multiple performance demands of the controlled system posed by
the design engineer. When the performance of a system controlled by PID is as expected, the tuning must migrate to other methodologies. In [3] the authors present a hybrid PSO method for PID controller for trajectory tracking application of a liquid level control process. In [4] the optimization of the PID parameters is performed by means of ant colony optimization. In [5] the authors propose the automation in a two interactive conical tank process by means of a fractional order PID controller. The tuning strategy adopted in this case is the cuckoo search optimization metaheuristic. This strategy, based on the parasitic behavior of some birds, is also applied in [6] for the control of a DC servo motor.

A comparison of different metaheuristic techniques applied to parameters optimization of PID controllers is presented in [7]. In this case the authors considered a genetic algorithm (GA), particle swarm optimization (PSO) and the method of cross entropy (CE). The algorithms associated with each metaheuristic were developed and compared on different simulated processes. The authors also presented an implementation in a physical controller (PLC) connected to a physical process. Finally, the authors in [7] reported that the use of metaheuristics to optimize PID controllers allows to obtain a control efficiency comparable or superior to conventional techniques.

In this paper we propose a PSO applied for the tuning of a PID controller. The controller is implemented on a scale plant designed for air pressurization in a storage tank. A predictive DMC controller is also implemented for comparative purposes, showing inferior performance than the proposed PID tuned through PSO.

2 Methodology

In the implementation of PID controllers on real applications, the main challenge for an optimal system response lies in the determination of the appropriate values of its constants [8]. In general, PID controllers require the tuning of three parameters: proportional gain $k_p$, integral gain $k_i$, and derivative gain $k_d$. The gains are used to generate the control action in continuous or discrete time; indicated in equation (1) as $m(t)$ and $m(k)$, respectively. The gains are derived from the dynamic behavior of the plant and are directly or indirectly related to the parameters of the model. Each gain of the controller directly affects the response at the output. In this case $k_p$ is a gain that modifies the response speed of the controller, $k_i$ ponders the integration of the error and $k_d$ ponders the error delta, generating a control action proportional to the change. In this case $k_d$ might present the disadvantage of amplifying the noise presented in the plant. This has caused that in various applications the derivative action is null, limiting the implementation to strategies based on PI actions.

\[
m(t) = k_p e(t) + k_i \int e(t) \, dt + k_d \frac{de(t)}{dt} \rightarrow m(k)
\]

\[
= q_0 e(k) + q_1 e(k - 1) + q_2 e(k - 2) + m(k - 1)
\]
2.1 Mathematical modeling

The tuning of conventional and advanced controllers requires a mathematical model of the plant dynamics on which the control algorithms will be implemented. Most conventional PID control strategies are based on approximate mathematical models as indicated in (2), where \( k \) is the gain of the plant, \( \tau \) is the time constant, \( \theta \) is the delay and \( s \) is the complex variable.

\[
G(s) = \frac{ke^{-\theta s}}{\tau s + 1}
\]

(2)

The mathematical model in continuous and discrete time of the test plant is given by (3) and (4). The reduced model of the plant was obtained from the identification of the system, applying step type stimuli. To discretize the models, we used the criterion of the equivalent time constant of the closed loop continuous model, with a sampling period \( T = 3s \).

\[
G(s) = \frac{2.014e^{-0.3s}}{20.857s + 1}
\]

(3)

\[
G(s) \xrightarrow{z \rightarrow T = 3s} HGP(z) = \frac{0.237z + 0.033}{z^2 - 0.865z}
\]

(4)

The formulation of the PID controller optimization problem is given by equations (5) to (9).

**Decision Variables:** \( k_p; k_i; k_d \)  

\[
F_{obj} = \min(\sum_{i=1}^{3} w_i J_i); \forall i; i = 1,2,3; \ w_i > 0; J_i > 0; weight \ factor
\]

(6)

\[
w_1 = MP = \frac{y_{max} - y_{ss}}{y_{ss}} * 100\%; \ y_{ss} = steady \ state \ output; \ 0 < MP < 10\%
\]

(7)

\[
w_2 = t_{ss(LC)} < 4\tau; \ 0 < t_{ss(LC)} < 4\tau
\]

(8)

\[
w_3 = TVM = \sum_{i=1}^{k} AC; \ \forall i; i = 1,2,3...k; \ k = \# \ of \ executions > t_{ss(LC)}
\]

(9)

2.2 PSO metaheuristic

The PSO metaheuristic was developed by Kennedy and Eberhart in 1995 [9] and is inspired on the social behavior observed in groups of individuals such as flocks of birds, swarms of insects and schools of fish. In the swarm, each particle determines its next movement in the search space, from the combination of historical components such as: current location, better location of one or more members of the swarm (memory and intensification), and random disturbance (diversification).
In the PSO algorithm the velocity (given by (10)) of each particle is modified iteratively so that its position (given by (11)) oscillates stochastically between the best position \( p_i \) and the best global position \( p_g \). To initialize the PSO metaheuristic, the following tuning parameters are required: number of particles \( X \), acceleration parameters (cognitive component \( \varnothing_1 \) and social component \( \varnothing_2 \)), maximum speed, inertia (\( \omega \)), and stopping criteria. This last one can be the number of iterations or a given number of iterations without improvement.

\[
\begin{align*}
\vec{v}_i &\leftarrow \omega \vec{v}_i + \vec{U}(0, \varnothing_1)(\vec{p}_i - \vec{x}_i) + \vec{U}(0, \varnothing_2)(\vec{p}_g - \vec{x}_i) \\
\vec{x}_i &\leftarrow \vec{x}_i + \vec{v}_i
\end{align*}
\]

\[(10)\]

\[(11)\]

3 Tests and Results

3.1 PSO tuning

For the initial tuning of the PSO, values of each gain were generated, using the parameters of the approximate model of the plant and implementing equation (12). The three gains were organized in a row vector as a representation of the solution \([k_p, k_i, k_d]\). To diversify the initial solutions, each gain was extended by its own value plus a weighting factor as shown in equations (13), (14) and (15). Subsequently, random values were generated with uniform distribution from the maximum and minimum value allowed for each gain. The above generated a matrix representation of solutions, which is given by (16).

\[
k_p = k = 2; k_i = \frac{k_p}{2} = 1; k_d = \frac{k_i}{10} = 0.1
\]

\[(12)\]

\[
kp_{max} = (0.8 * k_p) + k_p ; kp_{min} = k_p - (0.8 * k_p)
\]

\[(13)\]

\[
ki_{max} = (0.8 * k_i) + k_i ; ki_{min} = k_i - (0.8 * k_i)
\]

\[(14)\]

\[
k_{d\text{max}} = k_d + (0.8 * k_d) ; k_{d\text{min}} = k_d - (0.8 * k_d)
\]

\[(15)\]

\[
(X) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} k_{p1} & k_{i1} & k_{d1} \\ k_{p2} & k_{i2} & k_{d2} \\ \vdots & \vdots & \vdots \\ k_{pn} & k_{in} & k_{dn} \end{bmatrix} \rightarrow \begin{bmatrix} Particle_1 = x_1 \\ Particle_2 = x_2 \\ \vdots \\ Particle_n = x_n \end{bmatrix}; n = \#Particles
\]

\[(16)\]

The parameters used by the PSO for the tuning of the PID controller of the pressure plant were refined by an Ad-hoc method. A total of 18 tests were carried out, which allowed delimiting three regions according to the quality of the solution found. Each test was run with 120 iterations. The regions found are described below: infeasibility region 1, characterized by violation of some or all of the constraints; feasibility region 2 with solutions that satisfy the constraints, and feasibility region 3 with improved quality solutions with respect to region 2. The numerical and graphical
results of the Ad-hoc test are illustrated in figure 1 and table 1. Maximum speed was initialized with a constant value \( V_{\text{max}} = 5 \), and the inertia \( 0 < \omega < 1 \) to decrease the impact of the velocity of the particles, allowing to control the random component in the algorithm.

![Figure 1: Exploratory ad-hoc test](image)

**Table 1. Numerical results of the ad-hoc test**

<table>
<thead>
<tr>
<th>Prueba</th>
<th>( k_c )</th>
<th>( k_i )</th>
<th>( k_d )</th>
<th>( F_{\text{obj}} )</th>
<th>( TVM )</th>
<th>( M_p )</th>
<th>( t_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.3727</td>
<td>0.0326</td>
<td>0.1205</td>
<td>42.940</td>
<td>284.479</td>
<td>3.39%</td>
<td>81s</td>
</tr>
<tr>
<td>2</td>
<td>2.9751</td>
<td>0.0254</td>
<td>0.1196</td>
<td>41.333</td>
<td>284.364</td>
<td>1.05%</td>
<td>18s</td>
</tr>
<tr>
<td>3</td>
<td>5.0496</td>
<td>0.0075</td>
<td>0.6655</td>
<td>44.616</td>
<td>288.290</td>
<td>28.91%</td>
<td>54s</td>
</tr>
<tr>
<td>4</td>
<td>2.1248</td>
<td>0.0418</td>
<td>0.0198</td>
<td>44.207</td>
<td>282.333</td>
<td>0.33%</td>
<td>36s</td>
</tr>
<tr>
<td>5</td>
<td>2.9975</td>
<td>0.0182</td>
<td>0.1923</td>
<td>41.337</td>
<td>284.881</td>
<td>1.9%</td>
<td>18s</td>
</tr>
<tr>
<td>6</td>
<td>4.0716</td>
<td>0.1259</td>
<td>0.1271</td>
<td>50.850</td>
<td>289.310</td>
<td>37.63%</td>
<td>63s</td>
</tr>
<tr>
<td>7</td>
<td>2.9899</td>
<td>0.0209</td>
<td>0.1648</td>
<td>41.335</td>
<td>284.695</td>
<td>1.58%</td>
<td>18s</td>
</tr>
<tr>
<td>8</td>
<td>3.0550</td>
<td>0.0005</td>
<td>0.3821</td>
<td>41.350</td>
<td>285.958</td>
<td>4.22%</td>
<td>63s</td>
</tr>
<tr>
<td>9</td>
<td>5.1056</td>
<td>0.0566</td>
<td>1.5583</td>
<td>50.221</td>
<td>289.917</td>
<td>41%</td>
<td>63s</td>
</tr>
<tr>
<td>10</td>
<td>1.5889</td>
<td>0.0301</td>
<td>0.09</td>
<td>47.02</td>
<td>281.396</td>
<td>2.98%</td>
<td>117s</td>
</tr>
<tr>
<td>11</td>
<td>2.097</td>
<td>0.0291</td>
<td>0.1202</td>
<td>44.26</td>
<td>283.574</td>
<td>3.02%</td>
<td>90s</td>
</tr>
<tr>
<td>12</td>
<td>1.5373</td>
<td>0.0606</td>
<td>0.1028</td>
<td>60.50</td>
<td>284.838</td>
<td>13.94%</td>
<td>117s</td>
</tr>
<tr>
<td>13</td>
<td>3.0475</td>
<td>0.0025</td>
<td>0.3603</td>
<td>41.355</td>
<td>285.855</td>
<td>3.97%</td>
<td>63s</td>
</tr>
<tr>
<td>14</td>
<td>3.0220</td>
<td>0.01</td>
<td>0.2797</td>
<td>41.35</td>
<td>285.431</td>
<td>3%</td>
<td>45s</td>
</tr>
<tr>
<td>15</td>
<td>4.3917</td>
<td>0.1041</td>
<td>0.4024</td>
<td>49.88</td>
<td>289.462</td>
<td>38.85%</td>
<td>45s</td>
</tr>
<tr>
<td>16</td>
<td>2.9548</td>
<td>0.0325</td>
<td>0.0481</td>
<td>41.339</td>
<td>283.761</td>
<td>1.71%</td>
<td>18s</td>
</tr>
<tr>
<td>17</td>
<td>3.0181</td>
<td>0.0112</td>
<td>0.2662</td>
<td>41.345</td>
<td>285.354</td>
<td>2.84%</td>
<td>36s</td>
</tr>
<tr>
<td>18</td>
<td>3.573</td>
<td>0.0108</td>
<td>0.4636</td>
<td>43.66</td>
<td>287.394</td>
<td>10.59%</td>
<td>72s</td>
</tr>
</tbody>
</table>

Figure 2 shows the adaptation of the PSO algorithm to the tuning problem of the PID controller. In the evaluation block, a discrete PID controller is tuned for each particle and the corresponding objective function is obtained from the numerical analysis of each response. The algorithm converges by number of iterations providing the parameters of the controller and performance indices; in addition to the graphic responses of controlled output, control law, and evolution of the objective function. The programming of the PID-PSO was developed in Matlab®.
3.2 Implementation of PID and DMC control

To evaluate the performance of the PID controller tuned by the PSO metaheuristic, a comparison with an advanced DMC controller was used in addition to performance metrics. The DCM was chosen since its algorithm allows approximating an optimal controller, implementing an objective function that seeks to minimize the quadratic error, and the square of the change in the control action, trying to bring the process as close as possible to the desired trajectory. The general expression for such control law is given by equations (17) to (20) where $J$ is the cost function, $G$ is a dynamic matrix, $D$ is the difference between the prediction and the reference, $\alpha$ and $\lambda$ are the suppression ad ponderation mobile factors of the DMC, respectively, $\Delta U$ is the vector of control actions and $M$ is an auxiliary matrix used to compute the increments in the control law.

$$\frac{\partial J}{\partial \Delta U} = 2GT\alpha(G\Delta U - D) + 2\lambda\Delta U = 0$$  \hspace{1cm} (17)

$$\Delta U = (GT\alpha G + \lambda)^{-1}GT\alpha D \quad \hspace{1cm} (18)$$

$$\Delta U = M*D \quad \hspace{1cm} (19)$$

$$M = (GT\alpha G + \lambda)^{-1}GT\alpha \quad \text{Constant for all } k \quad \hspace{1cm} (20)$$
The PID controller chosen for the actual implementation of the pressure plant was the one obtained in the tuning of test 2 of the PSO; a test that generated the minimum value of the objective function in comparison with the other tunings. Also, this test presented the lowest $t_{ss(LC)}$ with an MP of 1.05%. The simulated graphic response, generated by the PSO of the PID controller chosen for the implementation, is shown in Figure 3, and the graphical evolution of the objective function in Figure 4. Both figures were configured to show the response to asymptotic behavior on the X axis.

![Figure 3: Response of the PID-PSO controller.](image)

![Figure 4: Evolution of the objective function of the PID-PSO controller.](image)

In the actual implementation of controllers, it is expected that the design is able to compensate for the non-linearities of the plant dynamics, as well as internal disturbances. The simulation of the controlled system allows to demonstrate the performance of the controller; however, it is in the implementation that the capacity of the system to behave according to the established requirements is determined; without departing from the design specifications. The software LabView™ was used to implement the PID-PSO controller as well as the DMC. The algorithms were programmed with the same sampling period and under the same conditions.
Figure 5 shows the response of the PID-PSO controller. In the graphical response, monitoring of the reference trajectory is observed by the controlled output (storage pressure), with MP equal to zero and control law without saturation, with stable behavior and without oscillations. Figure 6 shows the graphic response of the DMC controller, oscillatory tracking of the desired trajectory is observed by the controlled output with MP> 0 and oscillation in the initial response of the control law. The \( t_{ss(LC)} \) shows better performance in the PID-PSO controller, surpassing the DMC. Table 2 shows the numerical results of the implementation of both controllers. Note that the performance of the PID-PSO controller is superior to the one obtained with the DMC controller.

Figure 5. Graphic response of the implementation of the PID-PSO controller

Figure 6: Graphic response of the implementation of the DMC controller

Table 2. Numerical comparison of the PID-PSO and DMC controllers

<table>
<thead>
<tr>
<th>Index</th>
<th>DMC</th>
<th>PID-PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{ss(LC)} )</td>
<td>( 21(T) = 63s )</td>
<td>( 18(T) = 54s )</td>
</tr>
<tr>
<td>( e_{ss} )</td>
<td>( \pm 2% )</td>
<td>( \pm 0.6 )</td>
</tr>
<tr>
<td>( TVM )</td>
<td>1099.2561%</td>
<td>997.6209%</td>
</tr>
<tr>
<td>( MP )</td>
<td>7.044%</td>
<td>0%</td>
</tr>
</tbody>
</table>
4. Conclusions

A tuning strategy was implemented to optimize a PID controller by means of a particle swarm metaheuristic optimization algorithm. The best particle of the swarm formed by the three gains of the PID multimodal action was obtained, which simultaneously managed to minimize the time and energy specifications established in the design.

PID-PSO and DMC controllers were implemented in an air pressurization plant and the comparisons established were able to determine the superiority in the performance of the PID-PSO. This demonstrated the effectiveness of the proposed technique for the tuning of controllers that require several adjustment parameters and the minimization of several variables.

An Ad-hoc methodology was used to refine the tuning parameters of the PSO metaheuristics, 18 tests were carried out allowing the definition of three regions, named as: infeasibility region with violation of the constraints, intermediate feasibility region, and feasibility region improved in quality of the solutions. The latter allowed locating the controller with the minimum objective function for the implementation on the air pressurization plant.

Acknowledgements. The authors want to acknowledge the “Proyecto de sostenibilidad” of Universidad de Antioquia for its support in the development of this project.

References


Received: August 7, 2018; Published: August 30, 2018