Optimal PID Controller based on Convex Optimisation and Particle Swarm Techniques

O T Makki1,2
osama.t@uokerbala.edu.iq
otmr80@gmail.com

L S Mohammad3
lobnan83@gmail.com

1Faculty of Engineering, University of Kerbala, Iraq
2Faculty of Engineering, Shahid Chamran University of Ahvaz, Iran
3Electricity Section, Deputy Governor’s Office of Development and Reconstruction, Kerbala, Iraq

Abstract. This paper discusses the design and implementation of multiple optimisation algorithms for tuning a PID controller. A metaheuristic algorithm known as particle swarm optimisation (PSO) is used together with convex optimisation techniques, and the validity of the proposed algorithm is examined by comparing its performance with the performance of the classic PSO-PID and the well-known Ziegler–Nichols PID (ZN-PID). To obtain useful comparisons, a nonlinear system, the air levitation system (AL), was utilised, controlled using a PID with three tuning strategies: the modified particle swarm technique (M-PSO-PID), the classic particle swarm (C-PSO-PID), and the Ziegler–Nichols method. The performance of the controllers was monitored through a cost function, the Integral Absolute Error (IAE). A disturbance was also imposed on the AL system to test the performance and the robustness of the algorithm under different conditions. The proposed algorithm is designed to determine and set the parameters for the M-PSO-PID controller utilising stabilising regions K_p, K_i, and K_d which form a convex problem. The PSO technique is then used to search for the optimal values inside the convex area. The simulation results for the control system show that the M-PSO-PID offered good closed-loop performance with advantages over other algorithms with regard to the system settling time and rise time. The results thus indicate the supremacy of the proposed algorithm.

Keywords: PID, particle swarm optimization, Metaheuristic algorithm, optimal control.

1. Introduction
PID controllers are widely used in multiple industries because of their simple mechanisms and structures. Usually, such controllers are tuned manually to obtain suitable values for the relevant parameters, which are K_p, K_i, and K_d. Choosing values for these parameters depends on maximising the transfer function within the plant to develop a stable, controlled system that can handle uncertainties and perturbations. Sometimes, the derivative gain may be omitted because it is challenging to choose the appropriate value for this; however, incorrect values for PID coefficients may lead to unstable behaviour.
Many algorithms have been implemented to tune PID controllers by adjusting controller parameters to find suitable values, and many efforts have been made to optimise controllers to achieve and maintain desirable regulated systems using naturally inspired optimisation tools. Gaiing [1] utilised the PSO as an adjustment technique for tuning controller parameters to implement an optimal PID controller for an AVR system, with results that illustrated the creativity and the robustness of the proposed algorithm. Singh et al. [2] used a bat algorithm to regulate the PID controller of a servo motor; they found that the algorithm ensured a high optimum working level for the controlled system. Andri et al. [3] used a genetic algorithm to tune the PID controller, with simulation results that showed improvement in the settling time and an apparent reduction in the overshoot. Chiha et al. [4] applied a multi-objective ant colony algorithm to implement an optimised PID controller. In their study, the ACO was thus used to determine optimal solution for tuning the controller and gaining better performance. Abachizadeh et al. [5] implemented an optimised PID with the aid of an artificial bee colony algorithm; the results showed that the ABC could increase system performance as compared with other techniques.

Despite the outstanding results of using many of these optimising techniques, some algorithms still have problems regarding convergence, precision, and time consumption [6]. In particular, for real-time embedded systems, tasks must be completed in a given time frame. Crucially, therefore, the initial values for the boundaries of the solution sets under consideration (potential search area) must be identified correctly to avoid high processing costs and inaccurate results.

In this study, the proposed algorithm combines the convex optimisation technique with the classical PSO algorithm to reduce the search area within an acceptable range. The controller’s parameters, which maintain the functionality of the controlled system, are thus chosen from a convex set forming a convex optimisation problem [7]. The traditional method used for the primary identification of the PID parameter $K_p$ is known as the Ziegler-Nichol's step response [8] [9], while the nature-inspired approach in this work is the particle swarm method, PSO. The procedure used identifies the boundaries of the stabilising regions for the PID, while the intelligence swarm is responsible for searching for optimal solutions inside the convex shape to set up suitable values for the PID controller's parameters.

Choosing applicable values for PID parameters offers better behaviour for the closed-loop controlled system, generating a non-fragile controller that displays a proper stability response even in the presence of perturbations. AL was thus controlled using this approach, and the parameters for the transfer function representing the plant were implemented experimentally. The Hermite-Biehler theorem was used to convert the original transfer function to a more suitable form [10].

This report is arranged as follows. Section 2 introduces the Ziegler-Nichol method for generating a step response, while in section 3, some mathematical preliminaries and results are discussed with regard to the Hermite-Biehler theorem and the stabilising regions of the PID controller. In section 4, the numerical model for PSO is presented, and in section 5, the methodology of the proposed approach is discussed. Section 6 applies the proposed algorithm to an air levitation system, and section 7 discusses the results. Conclusions are thus given in section 8.

2. Ziegler-Nichols step response method
This method is used to find the values of the three essential parameters, $k$, $L$, and $T$, experimentally. The open-loop AL is stimulated with a unit step signal with a transient response as shown in figure 1. The values of the parameters are calculated, and the $K_p$ computed by applying equations 1 and 4.
These parameters are essential to calculate the required values for Ziegler-Nichols PID (ZN-PID) and the coefficients are derived as follows [8][9]:

\[
K_p = \frac{1.2}{A} \\
K_i = \frac{0.6}{A \times L} \\
K_d = \frac{0.6 \times L}{A} \\
A = k \times \frac{L}{T}
\]

The Ziegler-Nichols proportional gain, \( K_p \), is used to identify the PID stabilising regions [11].

**3. PID controller stabilising regions**

In this section, the results presented in [11] are summarised:

A first-order system with delay usually has the response outlined in figure 1, an S shape. The following transfer function expresses this system

\[
G(s) = \frac{k}{1 + sT} e^{-Ls}
\]

where

- \( k \): steady-state gain.
- \( L \): time delay.
- \( T \): time constant.

Typically, the regulated system has a closed-loop configuration with unity feedback, as shown in figure 2.

![Figure 2. Closed-Loop system.](image)
\[ G_c(s) = K_p + \frac{K_i}{s} + K_d \]  

The generated closed-loop polynomial is thus 

\[ \sigma(s) = (kK_i + kK_p s + kK_d s^2) e^{-sL} + (1 + Ts)s \]  

Substituting into equation 7) gives the following expression: 

\[ e^{sL} = \cos(Ls) + i\sin(Ls) \]  

By applying the Hermite-Biehler theorem, three regions, shown in figure 3, which form the stabilising areas for integral and derivative gains as presented in [11] are developed:

- Region (1), a Trapezoidal shape (T) when 
  \[ -\frac{1}{k} < K_p < \frac{1}{k} \]  

- Region (2), a Triangle shape (Δ) when 
  \[ K_p = \frac{1}{k} \]  

- Region (3), a Quadrilateral (Q) when 
  \[ \frac{1}{k} < K_p < k_{upp} \]  

Where:

\[ k_{upp} = \frac{1}{k} \left[ T \frac{L}{L} \alpha \sin(\alpha) - \cos(\alpha) \right] \]  

\[ m_j \triangleq \frac{L^2}{z_j^2} \]  

\[ b_j \triangleq -\frac{L}{kz_j} \left[ \sin(z_j) + \frac{T}{L} z_j \cos(z_j) \right] \]  

\[ \omega_j \triangleq -\frac{L}{kz_j} \left[ \sin(z_j) + \frac{T}{L} z_j \left( \cos(z_j) + 1 \right) \right] \]
4. Particle Swarm Optimisation Method (PSO)

PSO is a metaheuristic, unconstrained search algorithm invented by Kenny and Eberhart in 1995 [12][13]. They studied the behaviour of birds and fishes looking for food and used this to develop an algorithm to find a solution for minimising or maximising a weighted function that describes the action of a controlled system, seen in equation 20. Particles in this algorithm are capable of “learning” based on social behaviours. During several successive iterations, particles thus begin to accumulate in an area near the optimal value. The mathematical model describing the behaviours of particles in such a swarm is illustrated in equations 16 to 19 [14][15]:

A: Velocity equation:

\[ v_i(t + 1) = \omega v_i(t) + r_1 w \phi_1 (p_i(t) - x_i(t)) + r_2 w \phi_2 (g(t) - x_i(t)) \]  

(16)

B: Position equation:

\[ x_{new}(t + 1) = x_{old}(t) + v_i(t + 1) \]  

(17)

and

\[ w = \frac{2^k}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \]

(18)

\[ \phi = \phi_1 + \phi_2 \geq 4 \]

(19)

where

\[ 0 < k < 1 \]

\( \phi_1, \phi_2 \): Constriction coefficients.

\( C_1, C_2 \): Acceleration coefficients.

\( r_1, r_2 \): Uniform distribution number (0,1).

\( r_1 C_1(p_i(t) - x_i(t)) \): Cognitive Component.

\( r_2 C_2(g(t) - x_i(t)) \): Social Component.

5. Design of a PID controller using stabilising regions and PSO

In the proposed approach, the PSO algorithm was used to minimise the fitness function that defines the behaviour of the controlled system, known as the Integral Absolute Error, which can be represented as

\[ IAE = \int_0^{t_{ss}} |e(t)| \, dt \]

(20)

where

\( t_{ss} \): Settling time.

\( e(t) \): Transient Response Error.
Usually, a PSO converges quickly and loses the diversity of solution sets [6]. In real-time embedded systems, this consumes time and processing resources to reach an optimal value within an unbounded region, based on it being an unconstrained optimisation technique; introducing constraints through combining this method with stabilising areas reduces the time required, however, and makes the search algorithm more efficient. The proposed algorithm was thus designed to search for the optimum values for the PID parameters within the available sets instead of an unconstrained area. These sets represent the possible solutions for the optimisation problem in equation 20, and are themselves represented by three convex shapes forming the stabilizing regions for the PID controller. The proposed algorithm is shown in figures 4 and 5.

**Figure 4.** M-PSO-PID control system.
6. Application

The proposed algorithm was applied to an air levitation (AL) system built using the following components, as shown in figure 6: 1) a distance measuring sensor unit, 2) a 70 cm tube, 3) a ping-pong ball, 4) a metal cylinder, and 5) a fan. Figure 7 shows an assembled AL system. A ping-pong ball is installed inside the pipe, and to levitate the ball, a fan with a DC motor was used, operating at 24 DC voltages inside the metal cylinder. The PWM signal controlled the fan speed. To measure the distance moved by the ping-pong ball, a sensor unit was used composed of an integrated combination of a Position Sensitive Detector (PSD) and an Infrared Emitting Diode (IRED) [16]; the sensor was fixed on the top of the pipe as shown in figures 6 and 7.

The ping-pong ball moved up and down depending on the speed of the DC motor running the fan which produced the necessary airflow to move the ball. The position sensor generated different values of voltages depending on the position of the ball in relation to the sensor. The ball could theoretically be moved to any location inside the plastic pipe based on controlling the speed of the motor; this required a stable and optimal closed-loop control system that could reject any perturbations.

The first-order model for the AL system was based on equation 5. The AL system was connected to the PC using LABVIEW and an ARDUINO board (Mega 2560) [17], used as a data-acquisition board. The system was stimulated with a step input signal at the operating point, and the response, predicted to be an S shape, was acquired. From the simulated system’s response in figure 8, the first-order transfer function was computed as per equation 21.

\[
G(s) = \frac{0.01148}{1 + 0.31s}e^{-0.5s}
\]  

(21)

where

\[
k = 0.01148, \quad T = 0.31, \quad L = 0.5
\]
7. Results and Discussion

A MATLAB environment was used to implement the proposed algorithm within the off-line procedure. The proposed algorithm was first fed with the computed transfer function, capable of generating three PID stabilising regions forming convex shapes as shown in figure 9. The required convex shape was generated based on the value of Ziegler-Nichols's proportional gain, $K_p$, computed based on equations 1 and 4. The proportional gain identifies the convex shape based on equations 9 to 11. The selection of the regions thus depends on the system itself, based on the values of $T$, $L$, and $k$; the algorithm thus generates the required region. After the necessary convex shape is identified, the particles spread inside the generated convex shape. Every particle holds paired values of $K_I$ and $K_d$. The mechanism of the PSO algorithm causes the particles to fly inside the convex shape, which contains many potential solutions; the particles move to search for the optimum values inside the convex shape, trying to minimise the fitness function (IAE) at every iteration to reach an optimum solution. The behaviour of particles as they learn and approach the optimal value enhances the response of the controlled system comparing with other algorithms. The particles move inside a two-dimensional convex shape, and any particle that could migrate outside the convex shape was not allowed to move outside the shape as the solution is within the shape. Such particles store their position for the next iteration.

![Figure 8. Step response for the air levitation system.](image)
Figure 9. Potential solutions inside convex shapes.

For the AL system, the algorithm generated the convex shape in figure 10. All points inside this trapezoidal shape were potential solutions, and all values for $K_i$, $K_d$ and $K_p$ inside it offered stability for the system, though with different performance levels. Fifty particles were spread randomly inside the shape; these particles were responsible for searching for the optimum values for the integral and derivative constants required for optimum performance.

Figure 10. T region for the proposed non-linear system

The simulation of the closed-loop controlled system is illustrated in figures 11-15. In figure 11, the simulations were for all potential solutions ($K_i$ and $K_d$) inside the T shape presented in figure 10, where the particles were spread randomly inside the shape. In figure 11, each response thus represents the step response for the controlled system within a different parameter of the PID controller. The simulations show high variations caused by the random values (positions) of particles inside the T shape.
Figure 11. Simulation for all the potential solutions inside convex shapes.
After several successive iterations, the variations in the simulation begin to reduce as the particles begin to move towards the best solution and to offer suitable values for the PID parameters. Most of the particles accumulated near or around the optimal solution, as seen in figure 12.

Figure 12. Simulation for accumulated particles.
In figure 13, it is clear that the simulation has reached a reduced level of variation, with the particles traveling in an area much closer to the optimum solution.

Figure 13. Simulation for accumulated particles.
Finally, figure 14 shows the final iteration, with all the particles at the same point, which represents the best values of parameters for the PID controller. There were no further variations in the responses as all particles had found the optimum value.
In figure 15, the M-PSO-PID is compared with the C-PSO-PID and ZN-PID algorithms. The simulations indicate that the response of the proposed algorithm is adequate, and the transient response is also acceptable, while excellent improvements in the system performance were noted.

The obtained PID was applied to the physical AL system, with a response as presented in figure 16. The system showed rapid response with almost zero steady-state error. However, the response is distinct, different from that of other regulated systems; in such cases, another convex shape may be generated based on the specifications of the controlled system.
Figure 16. Real-Time air levitation system response.

The comparative results for the step response performance of the controlled system are given in table 1, while the optimal values for the PID parameters are shown in tables 2 and 3. The convergence history for the M-PSO-PID during the tuning process is presented in figure 17.

Table 1. Step response performance for different algorithms

| Method      | Rise Time (s) | Settling time (s) | Overshoot (%) | Execution time (s) |
|-------------|---------------|-------------------|---------------|--------------------|
| ZN-PID      | 3.8419        | 9.07              | 0             | NaN                |
| M-PSO-PID   | 2.2947        | 5.713             | 0             | 218.762            |
| C-PSO-PID   | 3.644         | 9.1               | 0             | 512.621            |

Table 2. Optimal values for the PID parameters for different algorithms

| Index | ZN-PID | M-PSO-PID | C-PSO-PID |
|-------|--------|-----------|-----------|
| $K_p$ | 64.8084 | 64.8084   | 80.56     |
| $K_i$ | 269.2953 | 101.9934  | 70.9      |
| $K_d$ | 16.202 | 4.0046    | 8.7       |

Table 3. Lifecycle for the M-PSO-PID and the C-PSO-PID

| Algorithm | Index | 1st iteration | 10th iteration | 20th iteration | 30th iteration | 40th iteration | 50th iteration |
|-----------|-------|---------------|----------------|----------------|----------------|----------------|----------------|
| M-PSO-PID | $K_p$ | 64.8084       | 64.8084        | 64.8084        | 64.8084        | 64.8084        | 64.8084        |
|           | $K_i$ | 269.2953      | 106.1309       | 101.9036       | 102.076        | 101.9957       | 101.9934       |
|           | $K_d$ | 10.8402       | 8.8389         | 3.9847         | 4.0074         | 4.0046         | 4.0046         |
8. Conclusion
The paper focused on applying multiple optimisation techniques to determine the optimum parameters of a PID controller. The PSO algorithm was used alongside a convex optimisation algorithm to calculate the necessary constraints to reduce processing demand and increase convergence probability. It was thus shown that the solution for the optimisation problem was surrounded by a convex shape, allowing the determination of potential sets for solutions, reducing the search area. Furthermore, it was shown, by comparison of three different algorithms that M-PSO-PID outperformed the C-PSO-PID and ZN-PID methods in terms of system settling time and rise time. The performance index obtained with the M-PSO-PID algorithm was also lower than of the C-PSO-PID algorithm. The proposed algorithm was thus able to obtain optimum values for PID parameters as found by ZN-PID and C-PSO-PID algorithms with superior tracking performances. The benefit of using multiple optimisation algorithms is thus confirmed, offering a complementary means of improving the performance of PIDs designed using classical methods. Based on the results in this paper, this algorithm can be recommended for real-time applications for this class of non-linear systems.

References
[1] Gaing, Z. L. (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE transactions on energy conversion*, 19(2), 384-391.
[2] Singh, K., Vasant, P., Elamvazuthi, I., & Kannan, R. (2015). PID tuning of servo motor using bat algorithm. *Procedia computer science*, 60, 1798-1808.
[3] Mirzal, A., Yoshii, S., & Furukawa, M. (2012). PID parameters optimization by using genetic algorithm. *arXiv preprint arXiv:1204.0885*.
[4] Chiha, I., Liouane, N., & Borne, P. (2012). Tuning PID controller using multi-objective ant colony optimization. *Applied Computational Intelligence and Soft Computing*, 2012.
[5] Abachizadeh, M., Yazdi, M. R. H., & Yousefi-Koma, A. (2010, July). Optimal tuning of PID controllers using artificial bee colony algorithm. In *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (pp. 379-384). IEEE.
[6] Ezugwu, A. E., Adeleke, O. J., Akinyelu, A. A., & Viriri, S. (2020). A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. *Neural Computing and Applications*, 32(10), 6207-6251.
[7] Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
[8] Aström, K. J., & Hägglund, T. (1995). *PID controllers: theory, design, and tuning* (Vol. 2). Research Triangle Park, NC: Instrument society of America.
[9] Dorf, R. C., & Bishop, R. H. (2017). *Modern control systems*. Pearson.
[10] Bhattacharyya, S. P., Datta, A., & Keel, L. H. (2018). *Linear control theory: structure, robustness, and optimization*. CRC press.
[11] Silva, Guillermo J., Aniruddha Datta, and Shankar P. Bhattacharyya. "New results on the synthesis of PID controllers." *IEEE transactions on automatic control* 47.2 (2002): 241-252.
[12] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39-43). IEEE.
[13] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN’95-International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE.
[14] Kamil, H. G., Makki, O. T., & Umran, H. M. (2020, January). Optimal tuning of a Linear Quadratic Regulator for Position Control using Particle Swarm Optimisation. In *IOP Conference Series: Materials Science and Engineering* (Vol. 671, No. 1, p. 012047). IOP Publishing.

[15] Borne, P., Philip, F. G., Popescu, D., & Stefanoiu. (2014). *Optimization in Engineering Sciences: Approximate and Metaheuristic Methods*. John Wiley & Sons.

[16] Global.sharp Distance measuring sensors. [https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf). Accessed 25 Aug 2019.

[17] Arduino.cc Language reference. [https://www.arduino.cc/reference/en/](https://www.arduino.cc/reference/en/). Accessed 25 Aug 2019.