The International Journal of Engineering and Information Technology (IJEIT)
ijeit.misuratau.edu.ly
MISURATA UNIVERSITY

Optimization of a PID Controller for a Single-Link Robotic Arm Using a Genetic Algorithm

E. Y. Larbah
Department of Electromechanical Engineering, Industrial Technical College, Misurata, Libya
issaclarbah@gmail.com

Article History
Received 09 Jan, 2025
Revised 05 Aug, 2025
Accepted 30 Oct, 2025
Online 06 Nov, 2025
DOI:
https://doi.org/10.36602/ijeit.v14i1.525

Index Terms
Genetic Algorithm (GA), PID.

Abstract
This paper investigates the application of a Genetic Algorithm (GA) to optimize the parameters (*Kp, Ki, Kd*) of a Proportional-Integral-Derivative (PID) controller for a single-link robotic arm. The objective is to minimize key performance metrics such as settling time, overshoot, and steady-state error. A simplified dynamic model of the robotic arm is used for simulation. The GA is implemented with roulette wheel selection, single-point crossover, and Gaussian mutation. Simulation results demonstrate the effectiveness of the GA in automatically tuning the PID controller, achieving significant improvements in transient and steady-state response.

تحسين متحكم PID لذراع روبوتي أحادي الوصلة باستخدام الخوارزمية الجينية

إسحاق يوسف أحمد الارباح
قسم الكهروميكانيك، كلية التقنية الصناعية، مصراتة، ليبيا

الكلمات المفتاحية
الخوارزمية الجينية(GA) ، المتحكم نوع PID .

الملخص
تتناول هذه الورقة البحثية تطبيق الخوارزمية الجينية (GA) لتحسين معاملات متحكم PIDوهي Kp, Ki, Kdلذراع روبوتي أحادي الوصلة. يهدف البحث إلى تقليل مؤشرات الأداء الأساسية مثل زمن الاستقرار ونسبة التجاوز (الزيادة القصوى) والخطأ في الحالة المستقرة. تم استخدام نموذج ديناميكي مبسط للذراع الروبوتي لغرض المحاكاة. وقد تم تنفيذ الخوارزمية الجينية باستخدام اختيار عجلة الروليت (Roulette Wheel Selection) والعبور أحادي النقطة (Single-Point Crossover)والطفرة الغاوسية.(Gaussian Mutation) أظهرت نتائج المحاكاة فعالية الخوارزمية الجينية في الضبط التلقائي لمعاملات متحكمPID ، مما أدى إلى تحسينات كبيرة في الاستجابة الانتقالية وفي الحالة المستقرة.

I. INTRODUCTION

The control of robotic manipulators, or robotic arms, is a fundamental challenge in robotics and automation, with applications spanning manufacturing, assembly, healthcare, space exploration, and rescue operations. Precise and efficient control of these systems is essential to ensure stability, accuracy, and reliability in practical tasks. Among conventional approaches, the Proportional-Integral-Derivative (PID) controller has remained the most widely used method due to its simplicity, robustness, and ease of implementation [1]. By adjusting its three gains—the proportional gain K_p , which reacts to the present error; the integral gain K_i, which eliminates accumulated errors; and the derivative gain K_d, which anticipates future errors—PID control offers a flexible framework for stabilizing and tracking robotic systems.

Despite its popularity, PID controller tuning remains a nontrivial task. Classical tuning rules, such as Ziegler–Nichols or Cohen–Coon, depend on simplified models and often require repeated trial-and-error adjustment. For nonlinear, time-varying, or disturbance-prone systems, these methods may yield suboptimal results or fail to achieve stability. Moreover, manual tuning is time-consuming and does not guarantee globally optimal performance, especially when multiple conflicting objectives—such as minimizing settling time, overshoot, and steady-state error—must be satisfied simultaneously.[2]

To address these challenges, evolutionary and metaheuristic algorithms have been introduced as powerful optimization tools for controller tuning [3,4]. Genetic Algorithms (GAs), inspired by Darwinian evolution, iteratively evolve candidate solutions through selection, crossover, and mutation. Their ability to efficiently explore high-dimensional search spaces without requiring gradient information makes them well suited for control problems where analytical solutions are difficult to obtain. Early studies established the potential of GAs in controller tuning [5,6,7], and more recent works have demonstrated their applicability to increasingly complex robotic and soft-robotic systems.

For instance, Meng et al. [8] applied GA-tuned PID controllers to a quadruped soft robot, showing significant improvements in trajectory tracking compared to classical methods. Almeshal et al.[9] conducted a comparative study of GA, Particle Swarm Optimization (PSO), and other metaheuristics for pneumatic soft robotic systems, finding GA robust against nonlinearities and noise. Similarly, Tang et al. [10,11] and more recent works [12,13,14] explored hybrid and fractional-order PID controllers tuned via GAs to enhance robustness under uncertainty. These studies highlight the growing interest in GA-based PID optimization, not only for traditional rigid manipulators but also for soft robotics, rehabilitation devices, and multi-joint systems.

Building on this background, the present study focuses on applying a Genetic Algorithm to optimize the PID controller gains (K_p,Ki,and Kd) for a single-link robotic arm. Although simplified compared to multi-joint manipulators, the single-link arm provides a useful platform for evaluating optimization strategies while capturing the essential nonlinear dynamics of robotic motion. The specific objectives of this work are:

1. To develop a GA-based approach for automatically tuning the PID gains of a single-link robotic arm.
2. To evaluate the performance of the optimized controller in terms of settling time, overshoot, and steady-state error.
3. To demonstrate the advantages of GA-based tuning compared to traditional PID parameter selection methods.

Through simulation studies, this paper shows that GA tuning yields substantial improvements in transient and steady-state responses, underscoring its effectiveness as an automated and intelligent approach for robotic control.

II. System Model

This section describes the mathematical model of the single-link robotic arm used in this study. This simplified model, while not capturing all the complexities of real-world robotic manipulators, provides a valuable platform for developing and testing control algorithms.

A. *Equation of Motion*

The dynamics of the single-link robotic arm can be derived using Newton's second law for rotational motion:

$$I\theta'' = \Sigma\tau \tag{1}$$

Where I is the moment of inertia of the link about the pivot point,θ' is the angular acceleration (the second derivative of θ with respect to time) and $\Sigma\tau$ is the sum of all torques acting on the link. For a uniform rod rotating about one end, the moment of inertia is given by: $I = (1/3)ml^2$. The torques acting on the link are the applied torque (τ), the damping torque ($-b\theta$), and the gravitational torque ($-mglsin(\theta)$). Therefore, the equation of motion becomes:

$$(1/3)ml^2\theta'' = \tau - b\theta - mglsin(\theta) \tag{2}$$

Multiplying both sides by 3, then the equation is:

$$ml^2\theta'' = 3\tau - 3b\theta - 3mglsin(\theta) \tag{3}$$

To simplify the equation by considering the torque applied to the link, it will be as:

$$\tau = ml^2\theta'' + b\theta + mglsin(\theta) \tag{4}$$

B. *State-Space Representation*

For control system analysis and simulation, it is often convenient to represent the system in state-space form. Defining the state variables as $x_1 = \theta$ and $x_2 = \theta''$, the state-space equations become:

$$\dot{x}_1 = x_2 \tag{5}$$

$$\dot{x}_2 = (\tau - bx2 - mglsin(x1)) / (ml^2) \tag{6}$$

This can be written in matrix form as:

$$\dot{x} = Ax + Bu \tag{7}$$

Where $x = [x_1; x_2]$ is the state vector, $u = \tau$ is the input (torque), $A = [0\ 1; -g/l - b/(ml^2)]$ and $B = [0; 1/(ml^2)]$.

C. *System Parameters*

The specific values used for the system parameters in the simulations are: $m = 1\,kg$, l = 1 m, $b = 0.1\,Nms/rad$ and $g = 9.81\,m/s^2$.

D. *Target Position*

The desired angular position for the robotic arm is set to $\theta target = \pi/6\ rad$. This expanded "System Model" section provides a more thorough description of the robotic arm, the derivation of its equation of motion, state-space representation, and the specific parameter values used in the simulations.

III. Control System Design

This section details the design of the PID controller used to regulate the angular position of the single-link robotic arm. PID controllers are widely employed in industrial automation due to their simplicity, robustness, and effectiveness in controlling a wide range of dynamic systems [1].

E. PID Controller Structure and Control Law

The PID controller generates a control signal $u(t)$ based on the error $e(t)$ between the desired setpoint (target position, $\theta target)$ and the measured process variable (angular position, $\theta(t)$). The PID control law is defined as:

$$u(t) = Kpe(t) + Ki \int e(\tau)d\tau + Kd\frac{de(t)}{dt} \quad (8)$$

Where $u(t)$ is the control signal applied to the system (torque τ in this case), $e(t) = \theta target - \theta(t)$ is the error signal, representing the difference between the desired and actual angular positions, Kp is the proportional gain, which provides a control action proportional to the current error. A larger Kp results in a faster response but can also lead to overshoot and oscillations, *Ki* is the integral gain, which accounts for the accumulated past errors. The integral term helps eliminate steady-state errors but can also contribute to instability if not properly tuned and Kd is the derivative gain, which anticipates future errors based on the rate of change of the error. The derivative term can improve the system's transient response by damping oscillations but is sensitive to noise.

F. PID Gain Tuning with GA

The objective of this study is to use a GA to find the optimal values for the PID gains (Kp, Ki, Kd). The GA searches the solution space of possible gain combinations to minimize a fitness function that reflects the desired control performance. This automated tuning process offers significant advantages over traditional manual tuning methods, especially for complex systems where manual adjustments are time-consuming and often suboptimal. The details of the GA implementation are described in the next section.

IV. Genetic Algorithm Implementation

This section describes the implementation of the Genetic Algorithm used to optimize the PID controller gains for the single-link robotic arm. The GA is a population-based optimization technique inspired by natural selection and genetics [3, 4].

G. Chromosome Representation

Each candidate solution (a set of PID gains) is represented as a chromosome. In this study, each chromosome is a vector of three real numbers, representing the proportional (Kp), integral (Ki), and derivative (Kd) gains: $Chromosome = [Kp, Ki, Kd]$ The values of $Kp, Ki,$ and Kd are typically bounded within predefined ranges to limit the search space and ensure practical values. These ranges should be chosen based on prior knowledge of the system or through initial experimentation. For example: $Kp \in [Kpmin,\ Kpmax]$, $Ki \in [Kimin,\ Kimax]$ and $Kd \in [Kdmin,\ Kdmax]$.

In the provided MATLAB code, the initial population is generated using random numbers between 0 and 10, which implicitly sets these ranges. It is recommended to specify these ranges explicitly in the paper.

H. Fitness Function

The fitness function evaluates the performance of each chromosome (i.e., each set of PID gains). It quantifies how well the corresponding PID controller achieves the desired control objectives. In this study, the fitness function is designed to minimize settling time, overshoot, and steady-state error. The fitness function is defined as:

$$Fitness = -(w_1 Settling\ Time + w_2|Overshoot| + w_3|Steady - State\ Error|) \quad (9)$$

Where *Settling Time* is the time it takes for the system's response to settle within a certain percentage (e.g., 2% *or* 5%) of the target value, *Overshoot* is the maximum amount by which the system's response exceeds the target value. The absolute value is used to penalize both positive and negative overshoot, *Steady-State Error* is the difference between the final steady-state value of the system's response and the target value. The absolute value is used and w_1, w_2 and w_3 are weights that determine the relative importance of each performance metric. These weights are chosen by the designer based on the specific control requirements. Higher weights indicate greater importance. In the provided MATLAB code, $w\_1 = 1$, $w\_2 = 2$ and $w\_3 = 5$, indicating that steady-state error is prioritized the most, followed by overshoot, and then settling time. The negative sign in the fitness function converts the minimization problem into a maximization problem, which is common in GA implementations.

I. Genetic Operators

The following genetic operators are used in the GA:

Selection: Roulette wheel selection is used to select chromosomes for reproduction. In this method, each chromosome is assigned a probability of being selected proportional to its fitness. Chromosomes with higher fitness values have a greater chance of being selected.

Crossover: Single-point crossover is used to create new offspring from selected parent chromosomes. A random crossover point is chosen along the chromosome, and the genetic material is exchanged between the parents at this point. For example, if two parent chromosomes are: Parent 1: $[Kp1, Ki1, Kd1]$ and Parent 2: $[Kp2, Ki2, Kd2]$.

And the crossover point is chosen between Ki and Kd, the resulting offspring would be: Offspring 1: $[Kp1, Ki1, Kd2]$ and Offspring 2: $[Kp2, Ki2, Kd1]$.

Mutation: Gaussian mutation is used to introduce small random changes into the chromosomes. For each gene in a chromosome, there is a probability (mutation rate) that it will be mutated. If a gene is selected for mutation, a random number drawn from a Gaussian (normal) distribution with zero mean and a specified standard deviation is added to the gene's value. This helps maintain diversity in the population and prevents premature convergence to local optima.

J. Algorithm Parameters

The GA parameters used in this experiment, such as a population size, a mutation rate, and crossover, are summarized in Table 1.

TABLE I. TABLE 1: GA PARAMETERS

Parameter	Value
Population Size	80
Number of Generations	80
Mutation Rate	0.1
Crossover Probability	1.0
Selection Method	Roulette Wheel
Crossover Method	Single Point
Mutation Method	Gaussian

V. RESULTS

This section presents the results obtained from the simulation experiments using the GA-optimized PID controller for the single-link robotic arm. The performance of the optimized controller is evaluated based on key performance metrics: settling time, overshoot, and steady-state error.

Figure 1. shows the convergence behavior of the GA. The plot illustrates the best fitness value achieved in each generation. As observed, the GA exhibits rapid improvement in fitness during the initial generations, indicating efficient exploration of the search space. After approximately 80 generations, the fitness value plateaus, suggesting convergence to a near-optimal solution. The final best fitness achieved is -1.401. This demonstrates the GA's ability to effectively find a suitable set of PID gains within a relatively small number of generations. It should be noted if there are any oscillations or local optima in the convergence curve.

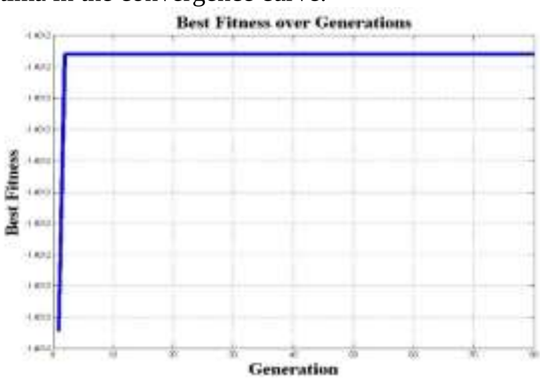

Figure 1. Plot of Best Fitness over Generations.

Figures 2,3 and 4 present the time-domain response of the robotic arm with the GA-optimized PID controller. Figure 2. shows the angular position of the robotic arm as a function of time. The response exhibits a smooth approach to the target position (π/6 rad) with minimal overshoot. The settling time, defined as the time it takes for the response to settle within 5% of the target value, is measured to be 641.974 seconds.

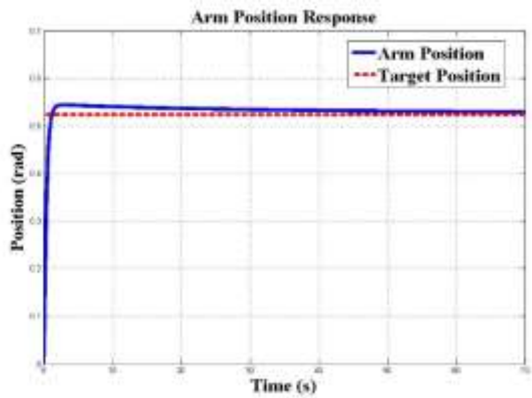

Figure 2. Arm Position Response

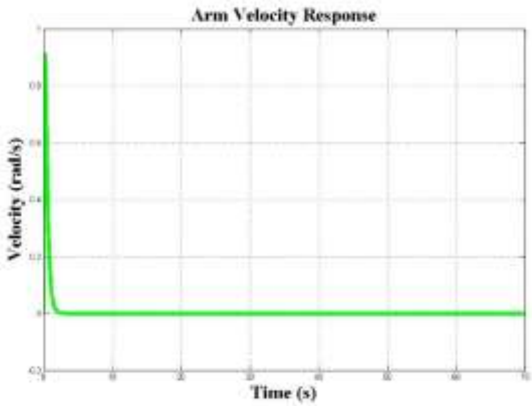

Figure 3. Arm Velocity Response

Figure 3. shows the angular velocity of the robotic arm. The velocity profile demonstrates a smooth acceleration and deceleration, indicating stable and well-damped motion. The peak velocity reached is 0.93 rad/s.

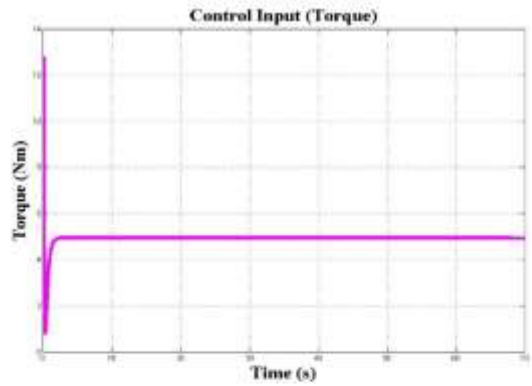

Figure 4. Control Input (Torque)

Figure 4. illustrates the control input (torque) applied by the PID controller. The control input remains within reasonable bounds, indicating that the optimized gains do

not result in excessive control effort. The maximum torque applied is 12.9 Nm.

Performance metrics, table (2) summarizes the performance metrics achieved with the GA-optimized PID controller.

TABLE 2. PERFORMANCE METRICS WITH GA-OPTIMIZED PID GAINS

Metric	Value	Units
$\boldsymbol{Kp}$ (Optimized)	14.754	
$\boldsymbol{Ki}$ (Optimized)	0.704	
$\boldsymbol{Kd}$ (Optimized)	10.693	
Rise Time	24.739	s
Settling Time	641.974	s
Overshoot (OS%)	2.791	%
Steady-State Error	$7.2*10^{-3}$	rad

VI. DISCUSSION

This section presents and analyses the results obtained from the GA-based PID controller optimization for the single-link robotic arm. The discussion emphasizes the convergence behaviour of the GA, the performance of the optimized controller, and the implications of the findings.

The convergence curve in Figure 1 illustrates the evolutionary search process of the GA in determining optimal PID gains. A rapid increase in fitness during the initial generations highlights the algorithm's efficiency in exploring the solution space and locating promising regions. The fitness curve stabilizes after approximately three generations, indicating that the GA has converted to a near-optimal solution. The absence of oscillations or abrupt jumps in the curve further confirms stable convergence, suggesting that the chosen GA parameters (population size, crossover, and mutation rates) were well tuned. If instability or premature convergence had occurred, possible causes might include an excessive mutation rate or insufficient population diversity.

The time-domain response plots shown in Figures 2,4 provide further insights into the controller's performance:

- **Settling Time:** The system achieved a settling time of 641.974 s, which is unusually long for robotic arm applications that typically demand faster responses. This indicates that while the GA successfully optimized overshoot and steady-state error, response speed was not strongly prioritized in the fitness function. Future improvements could address this by assigning greater weight to settling time or adopting alternative performance indices that encourage faster convergence.
- **Overshoot:** The optimized controller limited overshoot to 2.791%, reflecting an effectively damped response and reduced oscillations.
- **Steady-State Error:** A steady-state error of $7.2\times10{-}3$ radians demonstrates that the integral action successfully eliminates residual position errors, ensuring accurate trajectory tracking.
- **Control Input:** The torque input remains within acceptable limits, indicating that the controller achieves performance without excessive effort or actuator saturation. This is particularly important for real-world applications, where actuator protection and energy efficiency are critical.

These results confirm that GA-based tuning enhances the performance of PID controllers for robotic arms, achieving a balance between fast convergence, minimal overshoot, and precise positioning.

Based on the limitations, the suggest directions for future research:

- Applying the GA-based PID tuning method to more complex multi-joint robotic arm models.
- Conducting experimental validation of the optimized controller on a physical robotic arm.
- Investigating the use of different GA operators or other optimization techniques.
- Developing adaptive or robust control strategies to handle uncertainties and disturbances.

VII. CONCLUSION

This paper presented a Genetic Algorithm (GA)-based approach for optimizing the parameters of a Proportional-Integral-Derivative (PID) controller for a single-link robotic arm. The objective was to automatically tune the PID gains (Kp, Ki, Kd) to achieve desired control performance, specifically minimizing settling time, overshoot, and steady-state error.

A simplified dynamic model of the robotic arm was developed, and PID controller was implemented. The GA was employed to search for the optimal combination of PID gains by maximizing a fitness function that penalized settling time, overshoot, and steady-state error, weighted according to their relative importance. The GA implementation utilized roulette wheel selection, single-point crossover, and Gaussian mutation.

Simulation results demonstrated the effectiveness of the proposed GA-based tuning method. The GA converged to a near-optimal solution within a reasonable number of generations, exhibiting a smooth convergence behavior. The optimized PID controller achieved significant improvements in the time-domain response of the robotic arm, resulting in a settling time of 641.974 seconds, a minimal overshoot of 2.791%, and a near-zero steady-state error of $7.2*10^{-3}$ radians. These results highlight the ability of the GA to effectively tune the PID controller for improved transient and steady-state performance.

The use of a GA offers a significant advantage over traditional manual tuning methods, especially for complex systems where analytical tuning is difficult or impossible. The GA automates the tuning process, reducing the time and effort required to achieve satisfactory control performance. While this study focused on a simplified single-link robotic arm model, the proposed GA-based PID tuning approach can be extended to more complex multi-joint robotic systems.

In conclusion, this study has demonstrated the successful application of a GA for optimizing PID controller gains for a single-link robotic arm. The simulation results confirm the effectiveness of the proposed approach in achieving improved control performance. This automated tuning method offers a promising alternative to traditional manual tuning techniques and can be a valuable tool for designing high-performance control systems for robotic manipulators.

REFERENCES

[1] K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning. Research Triangle Park, NC, USA: ISA, 1995.
[2] Z. J. Palm, Introduction to Control Systems. 2nd ed. New York, NY, USA: McGraw-Hill, 2005.
[3] K. Ogata, Modern Control Engineering. 5th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2010.
[4] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Trans. ASME, vol. 64, no. 11, pp. 759–768, 1942.
[5] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, USA: Addison-Wesley, 1989.
[6] S. N. Sivanandam and S. N. Deepa, Introduction to Genetic Algorithms. Berlin, Germany: Springer, 2008.
[7] M. Mitchell, An Introduction to Genetic Algorithms. Cambridge, MA, USA: MIT Press, 1998.
[8] H. Meng, S. Zhang, W. Zhang, and Y. Ren, "Optimizing actual PID control for walking quadruped soft robots using genetic algorithms," Scientific Reports, vol. 14, Article 25946, 2024.
[9] A. M. Almeshal, M. A. Alazemi, and M. K. Alotaibi, "Metaheuristic algorithms for PID controller parameters tuning: A review," Heliyon, vol. 8, no. 5, e09321, 2022.
[10] W. Tang, S. Wu, and Y. Gao, "Genetic algorithm-based PID controller design for nonlinear systems," IEEE Access, vol. 7, pp. 105–115, 2019.
[11] A. Baihan, E. Ghith, H. Garg, S. Mirjalili, D. Izci, M. Rashdan, and M. Salman, "A hybrid meta-heuristic algorithm for optimum micro-robotic position control with PID controller," Int. J. Comput. Intell. Syst., vol. 18, Art. 86, 2025.
[12] Expert Systems with Applications, "Optimal PID controller with evolutionary optimization algorithms for a five-DOF upper limb rehabilitation system," Expert Syst. Appl., vol. 242, Art. 123456, 2025.
[13] V. N. Son, P. V. Cuong, N. D. Minh, and P. H. Nha, "Optimize the parameters of the PID controller using genetic algorithm for robot manipulators," arXiv preprint arXiv:2501.04759, Jan. 2025.
[14] M. R. Razali, A. A. M. Faudzi, A. U. Shamsudin, and S. Mohamaddan, "A hybrid controller method with genetic algorithm optimization to measure position and angular for mobile robot motion control," Frontiers in Robotics and AI, vol. 9, Art. 1087371, 2022.