COMETa 2024
7th INTERNATIONAL SCIENTIFIC CONFERENCE
14th - 16th November 2024
Jahorina, B&H, Republic of Srpska
University of East Sarajevo
Faculty of Mechanical Engineering
Conference on Mechanical Engineering Technologies and Applications

# ANALYZING AND OPTIMIZING PI CONTROLLER METHODS FOR TWO TANK SYSTEM: A LABORATORY-BASED STUDY

**Janani R[1], Saša Prodanović[2], Sai Phani Chandra Chittaluri[3], Ljubiša Dubonjić[4], Vladimir Stojanović[5]**

*Abstract: This research paper presents a comprehensive investigation into the design, analysis, and optimization of Proportional-Integral (PI) controller methods for interacting systems, conducted through laboratory experimentation. The study aims to enhance the performance and stability of PI controllers by exploring various tuning methods and strategies tailored to address the intricacies of interacting systems. Through a series of experimental setups in the laboratory environment, different PI controller tuning techniques are systematically evaluated, including modern optimization algorithms such as Genetic Algorithms, Particle Swarm Optimization and Ant Colony Optimization. The effectiveness of each tuning approach is assessed based on key performance metrics, including settling time, overshoot and steady-state error. Furthermore, the study investigates the robustness of the optimized PI controllers against parameter variations and disturbances commonly encountered in real-world systems. Comparative analysis and statistical evaluation are employed to identify the most suitable tuning method for achieving robust and reliable control performance in interacting system.*

*Key words: Genetic Algorithm, Interacting Systems, Level Control, Proportional Integral Controller, Particle Swarm Optimization*

[1] PhD Janani Rajaraman, Assistant Professor, Department of Electronics and Instrumentation Engineering Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Enathur, Kanchipuram 631561, India, janani.rajaraman@kanchiuniv.ac.in

[2] PhD Saša Prodanović, Associate Professor, University of East Sarajevo, Faculty of Mechanical Engineering, East Sarajevo, B&H, sasa.prodanovic@ues.rs.ba (CA)

[3] B.E Sai Phani Chandra Chittaluri, Department of Electronics and Instrumentation Engineering Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Enathur, Kanchipuram 631561, India, saiphanichandrachittaluri@gmail.com

[4] PhD Ljubiša Dubonjić, Associate Professor, University of Kragujevac, Faculty of Mechanical and Civil Engineering in Kraljevo, Kraljevo, Serbia, dubonjic.lj@mfkv.kg.ac.rs

[5] PhD Vladimir Stojanović, Associate Professor, University of Kragujevac, Faculty of Mechanical and Civil Engineering in Kraljevo, Kraljevo, Serbia, stojanovic.v@mfkv.kg.ac.rs

## 1 INTRODUCTION

Over the past two decades, meta-heuristic optimization techniques have become increasingly prominent in the design of controllers. Techniques such as Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are particularly well-known and established within this field. These algorithms are prized for their simplicity and their ability to simulate various natural phenomena, enabling computer scientists to create new algorithms or improve existing metaheuristics [1]. A defining feature of meta-heuristics is their initiation with random solutions, which obviates the need for derivative calculations in search spaces to find the optimal solution. This attribute makes them especially suitable for real-world problems where derivative information is either costly to obtain or unknown. Meta-heuristics are also distinguished by their superior ability to avoid local optima compared to traditional optimization methods. Their stochastic nature helps prevent stagnation in local solutions, promoting a thorough exploration of the search space. Consequently, these algorithms are particularly effective in providing robust solutions for complex optimization problems commonly encountered in process industries [2].

One fundamental challenge in process control is regulating the liquid level across multiple tanks and managing the flow of liquid between them. This task is essential in many chemical industries where maintaining the liquid level within specified parameters, despite external disturbances, is crucial. The complexity of level control processes is significant, regardless of whether they involve a single tank or multiple interconnected tanks, making them difficult to manage effectively. Conventional Proportional-Integral-Derivative (PID) controllers are widely used in these industries to handle such control challenges.

However, traditional PID controllers often struggle with the dynamic and nonlinear nature of these processes, necessitating more advanced control strategies. Meta-heuristic optimization techniques offer promising alternatives by optimizing controller parameters to enhance performance. For instance, GA, ACO and PSO can be used to tune PID controller parameters more effectively than conventional methods, leading to improved stability and responsiveness in level control systems. This is particularly valuable in scenarios involving multiple tanks, where interactions between tanks can introduce additional complexity.

In recent research, the application of meta-heuristic algorithms to control system design has shown significant improvements in performance metrics such as settling time, overshoot and steady-state error. These improvements are largely due to the algorithms' ability to explore a wide range of potential solutions and to adaptively search for optimal parameters. By leveraging the strengths of metaheuristic optimization, it is possible to develop controllers that not only meet but exceed the performance of traditional PID controllers.

In conclusion, the integration of metaheuristic optimization techniques into the design of controllers represents a significant advancement in process control. These algorithms provide robust and efficient solutions to complex control problems, particularly in chemical industries where maintaining precise control over liquid levels is critical. As research and development in this area continue, the adoption of meta-heuristic approaches is likely to expand, leading to further enhancements in control system performance and reliability.

## 2 MODEL DESCRIPTION

In this study, we analyze a two-tank system characterized by an interacting liquid level process. The primary variables include the inlet flow rate, designated as $F_{in}$ and the outlet flow rates for tank 1 and tank 2, denoted as $F_1$ and $F_2$, respectively. The parameters pertinent to tank 1 include the liquid level ($h_1$) and the cross-sectional area ($A_1$), while for tank 2, these parameters are the liquid level ($h_2$) and the cross-sectional area ($A_2$). Using the specified parameters detailed in Table 1, we derive the transfer function for the interacting liquid level system, which is subsequently expressed in Equation (1).

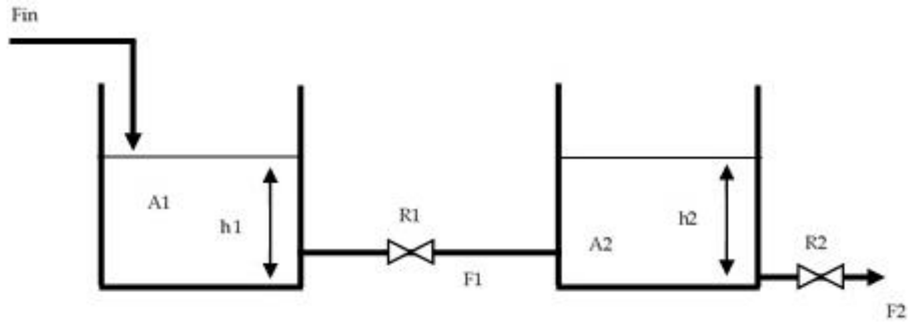


Figure 1. *Interacting Two Tank System*

Parameters

- $F_{in}$: Inlet Flow Volume ($m^3/s$),
- $F_1$: Outlet Flow Volume of tank 1 ($m^3/s$),
- $F_2$: Outlet Flow Volume of tank 2 ($m^3/s$),
- $A_1$: Cross Sectional Area of tank 1 ($m^2$),
- $A_2$: Cross Sectional Area of tank 2 ($m^2$),
- $R_1$: Resistance of flow at the outlet valve of tank 1,
- $R_2$: Resistance of flow at the outlet valve of tank 2,
- $h_1$: Liquid level in tank 1,
- $h_2$: Liquid level in tank 2.

By considering the following design parameters specified in Table 1, the transfer function of the interacting liquid level system is obtained as:

$$\frac{H_2(s)}{F_{in}(s)} = \frac{0.01}{6.25s^2 + 7.5s + 1} \tag{1}$$

Table 1. *Lab Scale Setup Specifications*

| Specifications | Value |
|---|---|
| Tank Capacity | 2 Litres |
| Tank Height | 280 mm |
| Transmiter | RF Capacitance Type<br>Input – 0-300 mm<br>Output – 4-20 mA |

## 3. GENETIC ALGORITHM

A Genetic Algorithm (GA) is an optimization and search technique inspired by the principles of natural selection and genetics. It is particularly effective for solving complex problems and nonlinear systems of equations. GA operates on a population of potential solutions, called individuals or chromosomes, which evolve over time through iterative processes. Key components of GA include a fitness function to evaluate the solutions and genetic operators such as selection, crossover and mutation to generate new solutions. Unlike deterministic methods, GA uses probabilistic transition rules, making it robust for exploring large and complex search spaces.

### 3.1 Steps to Design a PI Controller Using Genetic Algorithm

1. Initialize Parameters: - Define the population size, crossover rate, mutation rate, number of generations (Table 2.) and the coding mode for the parameters (e.g. binary or real-valued representation of PI controller parameters $K_p$ and $K_i$).
2. Generate Initial Population: - Create a random initial population of potential PI controller parameters. Each individual in the population represents a set of $K_p$ and $K_i$ values.
3. Evaluate Fitness: - Simulate the control system using each individual's $K_p$ and $K_i$ values. Calculate the fitness of each individual based on a predefined performance criterion (e.g. minimizing the error, overshoot, settling time, or a combination of these).
4. Selection: - Select individuals based on their fitness values to form a mating pool. Better performing individuals have a higher chance of being selected.
5. Crossover: - Apply crossover operators to pairs of selected individuals to generate new offspring. This involves exchanging parts of the parent chromosomes to produce new combinations of $K_p$ and $K_i$.
6. Mutation: - Apply mutation operators to the offspring with a certain probability. This introduces small random changes to the chromosomes to maintain diversity in the population and prevent premature convergence.
7. Form New Generation: - Replace the current population with the new offspring to form the next generation.
8. Iterate: - Repeat the evaluation, selection, crossover and mutation steps for a specified number of generations or until convergence criteria are met (e.g. a satisfactory fitness level or minimal change in fitness values across generations).
9. Extract Best Solution: - After the final generation, select the individual with the best fitness as the optimal set of PI controller parameters.

By following these steps, the GA optimizes the PI controller parameters $K_p$ and $K_i$, ensuring improved performance of the control system according to the defined fitness criteria.

Table 2. *Algorithm Parameter*

| | Particle Swarm Optimization | Genetic Algorithm |
|---|---|---|
| 1 | No. of Particles: 30 | Population Size: 50 |
| 2 | Maximum iterations: 100 | Crossover rate: 0.8 |
| 3 | Inertia Weight: 0.7 | Mutation rate: 0.01 |
| 4 | Cognitive coefficient: 1.5 | Number of generations: 100 |

## 4. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a population-based optimization technique inspired by the social behavior of birds flocking or fish schooling. It optimizes a problem by iteratively improving a candidate solution with regard to a given measure of quality or fitness. In PSO, each potential solution, called a particle, flies through the problem space by following the current optimum particles. Each particle adjusts its position based on its own experience and the experience of neighboring particles, guided by two primary equations involving velocity and position. The algorithm is known for its simplicity and effectiveness in solving a wide range of optimization problems.

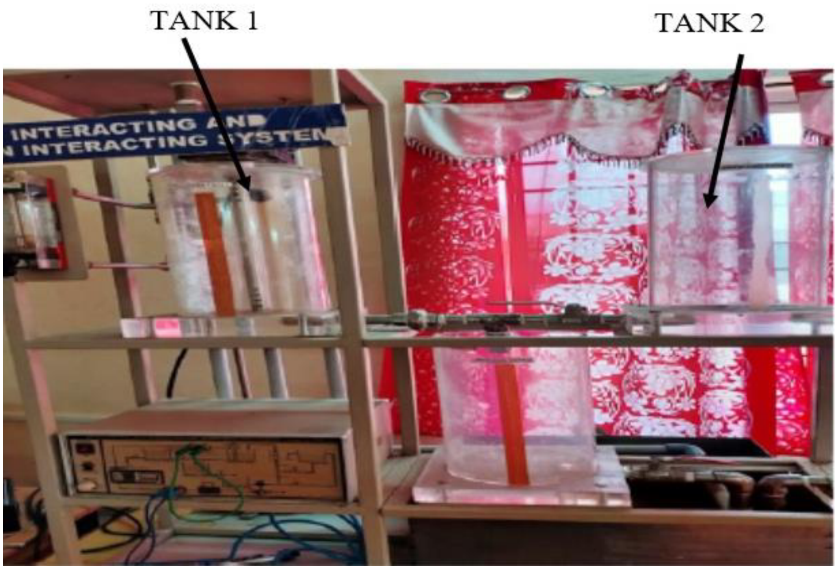


Figure 2. *Lab Scale Setup for Interacting Two Tank System*

### 4.1 Steps to Design a PI Controller Using PSO Algorithm

1. Initialize Parameters: Define the population size (number of particles), maximum number of iterations, inertia weight, cognitive (personal) coefficient and social coefficient (Table 2.). Initialize the range for the PI controller parameters $K_p$ and $K_i$.
2. Generate Initial Population: Create an initial swarm of particles with random positions (PI parameters $K_p$ and $K_i$) and velocities within the defined range.
3. Evaluate Fitness: - Simulate the control system using each particle's $K_p$ and $K_i$ values. - Calculate the fitness of each particle based on a predefined performance criterion (e.g. minimizing the error, overshoot, settling time, or a combination of these).
4. Update Personal Best: For each particle, compare the current fitness value with its bestknown fitness value (personal best). If the current value is better, update the personal best and record the corresponding position.
5. Update Global Best: Identify the particle with the best fitness value among all the particles. Update the global best position and fitness value if the current best is superior to the previous global best.

6. Update Velocity and Position: Update the velocity of each particle based on its personal best position and the global best position using the velocity update equation:

$$v_i(t+1) = wv_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i) \tag{2}$$

where $v_i$ is the velocity, $x_i$ is the position, $p_i$ is the personal best position, $g$ is the global best position, $w$ is the inertia weight, $c_1$ and $c_2$ are cognitive and social coefficients respectively and $r_1$ and $r_2$ are random numbers between 0 and 1. Update the position of each particle using the position update equation

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{3}$$

7. Iterate: Repeat the evaluation, personal best update, global best update, and velocity/position update steps for the specified number of iterations or until convergence criteria are met (e.g. a satisfactory fitness level or minimal change in fitness values).
8. Extract Best Solution: After the final iteration, select the particle with the best fitness as the optimal set of PI controller parameters $K_p$ and $K_i$.

By following these steps, the PSO algorithm effectively tunes the PI controller parameters as given in Table 3, ensuring enhanced performance of the control system according to the defined fitness criteria [11].

Table 3. *Tuning Parameters*

| Algorithm | Proportional Gain $K_p$ | Integral Gain $K_i$ |
|---|---|---|
| Genetic Algorithm | 186.25 | 42.71 |
| Particle Swarm Optimization | 296.62 | 41.91 |
| Ant Colony Optimization | 291.54 | 40.27 |

## 5. SIMUATION AND RESULTS

The simulation of a two-tank interacting system was conducted to evaluate the performance of various PI controllers. The results, illustrated in the figures 3 and 4, present both the servo (setpoint tracking) and regulatory (disturbance rejection) responses of the system, respectively. The controllers compared include those optimized using Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). In the servo response analysis, all PI controllers demonstrated effective setpoint tracking capabilities. The GA-based controller provided efficient setpoint achievement with minimal overshoot and quick settling time. The PSO-based controller showed even smoother setpoint tracking, slightly outperforming GA in terms of overshoot reduction. The ACO-based controller exhibited a slightly slower response but excelled in minimizing oscillations around the setpoint. For the regulatory response, which evaluates the system's ability to handle disturbances, the controllers displayed robust performance. The GA-based controller quickly rejected disturbances and returned to the setpoint. The PSO-based controller maintained excellent stability and swiftly corrected deviations caused by disturbances. The ACO-based controller, while slightly slower, effectively managed disturbances and ensured system stability. Overall, the simulation results confirm that PI controllers designed using these advanced computational algorithms can significantly enhance the performance of a two-tank interacting system. The controllers achieved good setpoint tracking and effective disturbance rejection, demonstrating their suitability for various industrial applications where maintaining precise control of liquid levels is crucial. The comparative analysis underscores the strengths of each algorithm, providing insights into their potential application based on specific performance criteria.

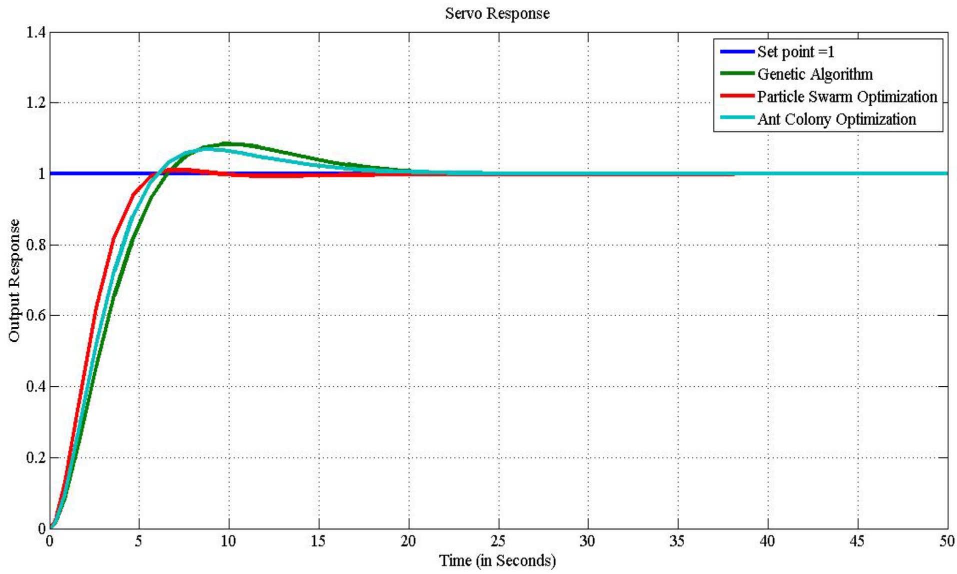


Figure 3. *Servo Response*

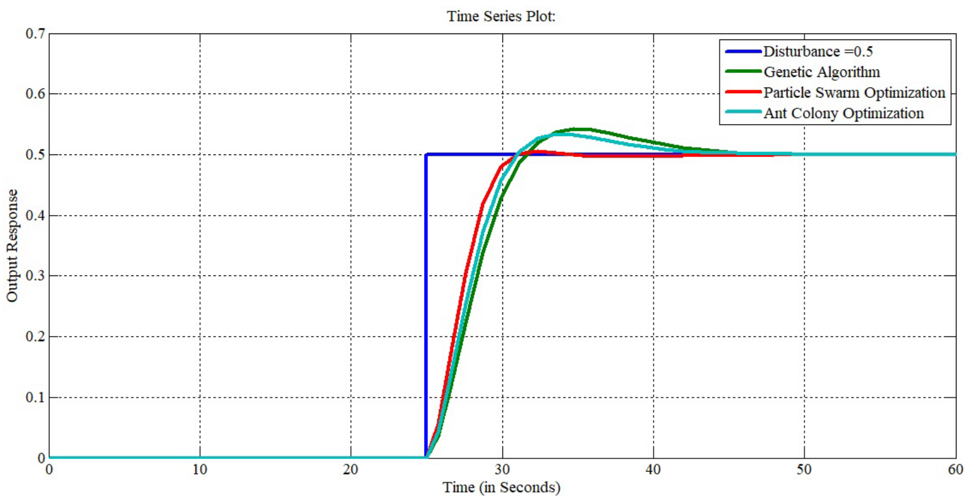


Figure 4. *Regulatory Response*

## 6. CONCLUSION

In conclusion, the simulation results underscore the effectiveness of utilizing advanced computational algorithms - Genetic Algorithm (GA), Particle Swarm

Optimization (PSO) and Ant Colony Optimization (ACO) - in designing PI controllers for a two-tank interacting system. Through robust setpoint tracking and disturbance rejection, these controllers demonstrate their pivotal role in ensuring stable and precise control of liquid levels, vital for numerous industrial processes. The comparative analysis showcases the unique strengths of each algorithm: GA excels in achieving efficient setpoint achievement, PSO offers smoother setpoint tracking with reduced overshoot and ACO prioritizes stability by minimizing oscillations around the setpoint. This study elucidates the significance of computational intelligence techniques in optimizing controller parameters, thus facilitating advancements in process automation and industrial control systems. By harnessing the capabilities of these algorithms, engineers and practitioners can enhance system performance, improve operational efficiency and meet stringent control requirements across a wide range of industrial domains. Overall, the research highlights the transformative potential of computational algorithms in addressing complex control challenges and driving innovation in industrial automation.

## REFERENCES

[1] Gao, Z., and Wang, Q. (2013). PI controller design for a two-tank system using ant colony optimization. *Control Engineering Practice*, 21(10), p.p. 1417-1427.

[2] Zhang, Y., Li, M., and Zhang, Y. (2016). A novel self-adaptive PI controller design method based on genetic algorithm. *Applied Soft Computing*, 45, p.p. 84-92.

[3] Bernal, D., and Ponton, J. (2011). Optimal tuning of PID controllers for integral and unstable processes using Particle Swarm Optimization. *ISA Transactions*, 50(3), p.p. 377-384.

[4] Chatterjee, S., Samanta, S., and Ghoshal, S. P. (2015). Design of PI controller for a two tank interacting liquid level system using genetic algorithm. *International Journal of Scientific and Engineering Research*, 6(6), p.p. 103-109.

[5] Noman, N. and Iba, H. (2008). Automatic PID control system design using genetic algorithm for time delay processes. *Applied Soft Computing*, 8(1), p.p. 669-678.

[6] Debbarma, M. and Debbarma, B. (2019). Design and Implementation of PI Controller for Coupled Tank System using Particle Swarm Optimization. *International Journal of Control Theory and Applications*, 12(42), p.p. 73-81.

[7] Nagarajan, T. and Satyanarayana, G. (2011). Genetic Algorithm Based Design of PI Controller for Liquid Level Process. *International Journal of Computer Applications*, 18(3), p.p. 28-32.

[8] Jena, S. and Biswal, M. (2014). Design of PI controller for two tank system using Particle Swarm Optimization. *International Journal of Science and Research*, 3(10), p.p. 44-47.

[9] Xu, D. and Zhai, X. (2009). A new PI controller tuning method based on improved particle swarm optimization. *Journal of Computational Information Systems*, 5(6), p.p. 1901-1908.

[10] Kadri, E. H. and Larouci, M. A. (2021). Optimal Tuning of PI Controller for the Two Tank System usinG Grey Wolf Optimization Algorithm. *Journal of King Saud University - Engineering Sciences*, 33(1), p.p. 44-51.

[11] Rajaraman, Janani and Sarangan, Jayanth (2021). Study on Simulation and Design of Various PI Controller for a Non-Interacting Systems: An Approach to Conventional and Computational Algorithms. *In: Current Approaches in Science and Technology Research*, Vol. 1. B P International, p.p. 154-159. ISBN 978-93-90149-97-1.