

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/ Simulink

Lana O. Alhamawndi<sup>1</sup>, Maroa E. Baker<sup>2</sup>, Koran A. Namuq<sup>3\*</sup>

<sup>1,2,3</sup>Technical college of Engineering/Kirkuk- Northern Technical University

**ABSTRACT:** Liquid level management is a crucial but difficult task in industrial process control because liquid storage systems are characterized by their nonlinear dynamics, parameter uncertainties and exogenous disturbances. Normal Proportional–integral–derivative (PID) controllers usually encounter complications with nonlinear processes, contributing to an extensive settling time, overshoot and tracking accuracy. This paper demonstrates the design and tuning of a Fuzzy PID controller using particle swarm optimization (PSO) for build-up performance enhancement of MATLAB/Simulink based nonlinearly connected liquid-level control system. This method exploits the flexibility of fuzzy logic, leveraging to an automated determination of the best parameters controller by means of a PSO algorithm that utilizes global optimization techniques. Nonlinear mathematical model of the liquid-level system is developed, and traditional PID, fuzzy PID and PSO-optimized fuzzy PID were implemented in all mentioned cases under same working conditions. The performance of the controllers was evaluated based on overall time-domain characteristics and standard metrics like rise time, settling time, overshoot (OS), steady state error and Integral Error Criteria: IAE [21], ISE[22],ITAE[24],RMSE. Simulation results also show that the Fuzzy PID controller is excellent in dynamic performance with optimization of PSO. Experimental results show that it can achieve faster response time, less overshoot, better steady-state accuracy, smaller control signal variation and improved robustness against uncertainties in parameters as well as external disturbances. The results indicate that the fuzzy logic and Particle Swarm Optimization combination is an effective, reliable intelligent control strategy for nonlinear liquid level systems with practical applications in advanced industrial process control.

**KEYWORDS:** Liquid level control, Fuzzy PID controller, Particle Swarm Optimization (PSO), MATLAB/Simulink, nonlinear control systems, intelligent control, process control.

### 1. INTRODUCTION

#### 1.1 Industrial Process Control and the Importance of Liquid Level Control

Industrial process control is a crucial component of modern manufacturing and automation. The demand for improved productivity, superior product quality, increase safety and decrease power consumption has compelled industries to rely on advanced automatic control systems [1], [2], [3]. Liquid level is one of the prominent factors in industrial processes, which has a great impact on equipment operation stability [4], process efficiency and product quality [5]. Liquid-level control systems have been implemented in a number of applications including chemical plants, petroleum refineries, water treatment facilities, thermal power stations and pharmaceutical manufacturing to food processing [6], softmusic.net[7]],[[8]][1] A liquid tank might have a simple physical design, but its dynamics are nonlinear due to the flow from outlet and height of the output. While designing the controller is complicated due to challenges of parameter uncertainty, sensor noise, actuator limitations and being subjected to external disturbances [9], [10], [11]. Therefore, with the attempt of enhancing system robustness and dynamic performance [12], [13], [14] intelligent control strategies have drawn a lot attention.

#### 1.2 Traditional PID Controllers and Their Limitations

Since the Proportional–integral–derivative (PID) controller is a simple, robust and easy to configure tool it finds widespread use in total adjustable process control as realized within countless industrial environments [15]. But conventional PID controllers are usually designed by constant parameters and lineary assumption, since they have limited performance in terms of handling nonlinear time-varying processes [16], [17]. In liquid-level systems, inadequate PID tuning can cause overshooting, oscillations with long settling times, and poor disturbance rejection [18]. This situation has led to researches on intelligent control techniques which

# Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

can adapt to nonlinear working conditions [19].

## 1.3 Intelligent Control Techniques

Now a days FLCs are practical methods of dealing with nonlinear systems due to the removal of an exact mathematical model and they act like human thinking using linguistic rules [7], [20]. However, the effectiveness of FLCs depends on finding appropriate membership functions, scaling factors and rule bases (usually computed by trial-and-error approaches [21]).

## 1.4 Particle Swarm Optimization

One of the commonly used methods in tuning controllers is Particle Swarm Optimization (PSO). This is because its simple architecture, fast convergence properties and the potential for global search make it a great choice when optimizing controller parameters [2]. When combined with a fuzzy-PID controller, PSO brings significant improvements to the transient response and robustness in both cases [22], [23], [24].

## 1.5 Research Gap and Contributions

Numerous studies have been conducted that address PID, Fuzzy Logic & PSO separately [24],[25], however even fewer attempts have combined these three strategies into an integrated controller for nonlinear liquid level systems with a comprehensive evaluation of the performance in different operating conditions. Thus, it presents the PSO-Optimized Fuzzy PID controller designed on MATLAB/Simulink and its comparison with normal PID-controller & standalone Knowledge based Adaptive Controller using various transient as well integral performance indices [23], [24]. The study develops a single intelligent controller that utilizes PID, Fuzzy Logic and PSO together which is evaluated under several different operating conditions.

## 2. LITERATURE REVIEW

### 2.1 Liquid Level Control

Liquid-level control systems are extensively used in industrial environments because they are essential for maintaining process stability, safety, and the quality of products. Due to their nonlinear flow properties, these systems are ideal for evaluating sophisticated control algorithms [1], [2], [3], [4].

### 2.2 Conventional PID Controllers

While PID controllers perform well with linear systems, they exhibit decreased robustness when dealing with nonlinear processes and parameter uncertainties [5], [6], [7], [8].

### 2.3 Fuzzy Logic Controllers

Fuzzy Logic Controllers enhance robustness by using linguistic rules instead of exact mathematical models, making them well-suited for nonlinear industrial systems [9], [10], [11], [12].

### 2.4 Optimization Algorithms

Metaheuristic algorithms, including GA, DE, GWO, and PSO, are extensively employed for optimizing controller parameters [10]. Notably, PSO achieves a remarkable equilibrium between the speed of convergence and computational efficiency [13], [14], [15], [16], [17].

### 2.5 Hybrid Intelligent Controllers

Hybrid Fuzzy-PID controllers optimized by PSO have demonstrated superior performance in terms of overshoot reduction, disturbance rejection, and tracking accuracy [18], [19], [20].

## 3. MATHEMATICAL MODELING OF THE LIQUID LEVEL SYSTEM

### 3.1 Tank Dynamics

The mathematical model was derived using the law of conservation of mass as follows:

$$\text{Rate of accumulation} = \text{Inflow rate} - \text{Outflow rate}$$

For a tank with a cross-sectional area, the accumulated liquid volume is expressed as

$$V(t) = Ah(t)$$

Therefore:

$$\frac{dV(t)}{dt} = A \frac{dh(t)}{dt}$$

The dynamic equation of the tank becomes:

$$A \frac{dh(t)}{dt} = q_i(t) - q_o(t)$$

where:

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

$A$  is the cross-sectional area of the tank,  
 $h(t)$  is the liquid level,  
 $q_i(t)$  is the inlet flow rate,  
 $q_o(t)$  is the outlet flow rate.

### 3.2 Outlet Flow Equation

For a liquid tank, the outlet flow is nonlinear and can be expressed as:

$$q_o(t) = a\sqrt{h(t)}$$

where  $a$  is the outlet valve constant, respectively. Substituting this equation into the tank model yields

$$A \frac{dh(t)}{dt} = q_i(t) - a\sqrt{h(t)}$$

or:

$$\frac{dh(t)}{dt} = \frac{1}{A} (q_i(t) - a\sqrt{h(t)})$$

This equation represents a nonlinear mathematical model of the liquid-level system.

### 3.3 Linearized Model

For controller design and comparison, the nonlinear model can be linearized around a nominal operating point  $(H_0, Q_0)$ . At steady state:

$$Q_0 = a\sqrt{H_0}$$

Let:

$$H(t) = H_0 + \Delta h(t)$$

$$q_i(t) = Q_0 + \Delta q_i(t)$$

After linearization, the small-signal model becomes:

$$A \frac{d\Delta h(t)}{dt} = \Delta q_i(t) - \frac{a}{2\sqrt{H_0}} \Delta h(t)$$

Therefore:

$$A \frac{d\Delta h(t)}{dt} + \frac{a}{2\sqrt{H_0}} \Delta h(t) = \Delta q_i(t)$$

### 3.4 Transfer Function

Taking the Laplace transform gives:

$$A\Delta H(s) + \frac{a}{2\sqrt{H_0}} \Delta H(s) = \Delta Q_i(s)$$

$$G(s) = \frac{K}{\tau s + 1}$$

where:

$$K = \frac{2\sqrt{H_0}}{a}$$

$$\tau = \frac{2A\sqrt{H_0}}{a}$$

### 3.5 MATLAB/Simulink Model Parameters

For simulation purposes, the following parameters can be used:

| Parameter | Description               | Value              |
|-----------|---------------------------|--------------------|
| $A$       | Tank cross-sectional area | 22 cm <sup>2</sup> |
| $a$       | Outlet valve constant     | 3                  |
| $b$       | Pump gain                 | 6                  |
| $H_0$     | Nominal liquid level      | 1 m                |

If the pump voltage  $V(t)$  is considered as the control input, the inlet flow can be written as:

$$q_i(t) = bV(t)$$

Then the nonlinear model becomes:

$$A \frac{dh(t)}{dt} = bV(t) - a\sqrt{h(t)}$$

Or:

$$\frac{dh(t)}{dt} = \frac{bV(t) - a\sqrt{h(t)}}{A}$$

This final equation can be directly implemented in MATLAB/Simulink using basic blocks such as Gain, Sum, Sqrt, Integrator, and Saturation.

### 3.6 State-Space Representation

The system can also be represented in state-space form by selecting the liquid level as the state variable:

$$x(t) = h(t)$$

The nonlinear state equation becomes:

$$\dot{x}(t) = \frac{bV(t) - a\sqrt{x(t)}}{A}$$

The output equation is:

$$y(t) = x(t)$$

thus:

$$\begin{aligned}\dot{x}(t) &= f(x, u) \\ y(t) &= x(t)\end{aligned}$$

where  $u(t) = V(t)$  is the pump voltage input and  $y(t) = h(t)$  is the measured liquid level.

### 3.7 Summary

The derived mathematical model shows that the liquid-level system is nonlinear owing to the square-root relationship between the outlet flow and liquid height. This nonlinearity renders the system suitable for evaluating intelligent control strategies. In this study, the nonlinear model was implemented in MATLAB/Simulink and controlled using PID, Fuzzy Logic, and the proposed PSO-Optimized Fuzzy PID controller.

## 4. CONTROLLER DESIGN

### 4.1 PID Controller Design

A Proportional–integral–derivative (PID) controller was used as the baseline controller in this study. It was selected because of its simple structure, wide industrial use, and ease of implementation. The PID controller generates a control signal based on the error between the desired and actual measured liquid levels.

The control error is defined as:

$$e(t) = r(t) - h(t)$$

where  $r(t)$  is the reference liquid level and  $h(t)$  is the actual liquid level.

The PID control law is given by:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

where  $K_p$ ,  $K_i$ , and  $K_d$  are the proportional, integral, and derivative gains, respectively.

The proportional term improves the speed of response, the integral term reduces the steady-state error, and the derivative term improves the damping and reduces the oscillations. However, the fixed parameters of the PID controller may not provide satisfactory performance for nonlinear liquid level systems under changing operating conditions [1], [2], [3].

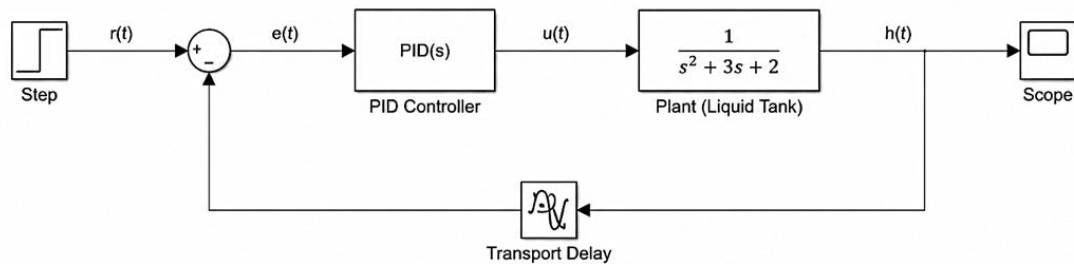


Figure 4.1 PID Controller Block Diagram

#### 4.2 Fuzzy Logic Controller Design

A Fuzzy Logic Controller was developed to enhance the nonlinear control capabilities of the liquid level system. In contrast to the PID controller, the fuzzy controller operates without needing a precise mathematical model of the plant. It relies on linguistic rules derived from the system's behavior instead.

In this study, the fuzzy controller uses two input variables.

$$e(t) = r(t) - h(t)$$

$$\Delta e(t) = e(t) - e(t-1)$$

where  $e(t)$  represents the level error and  $\Delta e(t)$  represents the change of error.

The output of the fuzzy controller is the control signal  $u(t)$ , which controls the inlet flow rate or pump voltage. The fuzzy logic controller consists of four main stages: fuzzification, rule base, inference engine, and defuzzification.

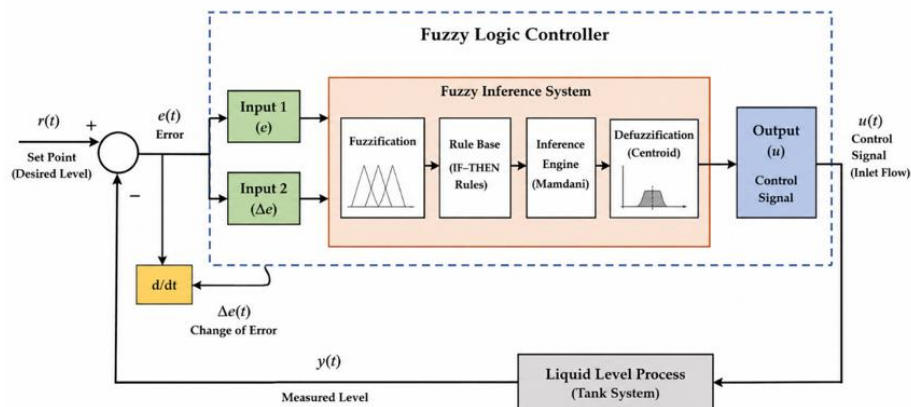


Figure 4.2 Fuzzy Logic Controller Structure

| Linguistic terms  | Description     | Variable      |
|-------------------|-----------------|---------------|
| NB, NS, Z, PS, PB | Error           | $e(t)$        |
| NB, NS, Z, PS, PB | Change of error | $\Delta e(t)$ |
| NB, NS, Z, PS, PB | Control signal  | $u(t)$        |

#### 4.3 Particle Swarm Optimization Design

To enhance the performance of the controller, Particle Swarm Optimization is used to automatically adjust its parameters. PSO, an optimization algorithm based on population, draws inspiration from the social behavior of birds and fish. In this method, each particle signifies a potential solution, and they navigate the search space to identify the optimal controller parameters. In this study, PSO was used to optimize the parameters of the Fuzzy-PID controller. The optimization objective was to minimize the tracking error and improve the transient response. The velocity and position of each particle are updated as follows:

$$v_i^{k+1} = wv_i^k + c_1r_1(pbest_i - x_i^k) + c_2r_2(gbest - x_i^k)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

where  $w$  is the inertia weight,  $c_1$  and  $c_2$  are acceleration coefficients,  $r_1$  and  $r_2$  are random numbers,  $pbest$  is the personal best solution, and  $gbest$  is the global best solution. The fitness function can be defined using the Integral Time Absolute Error

$$J = ITAE = K_i \int_0^T t |e(t)| dt$$

The smaller the value of  $J$ , the better the controller performance

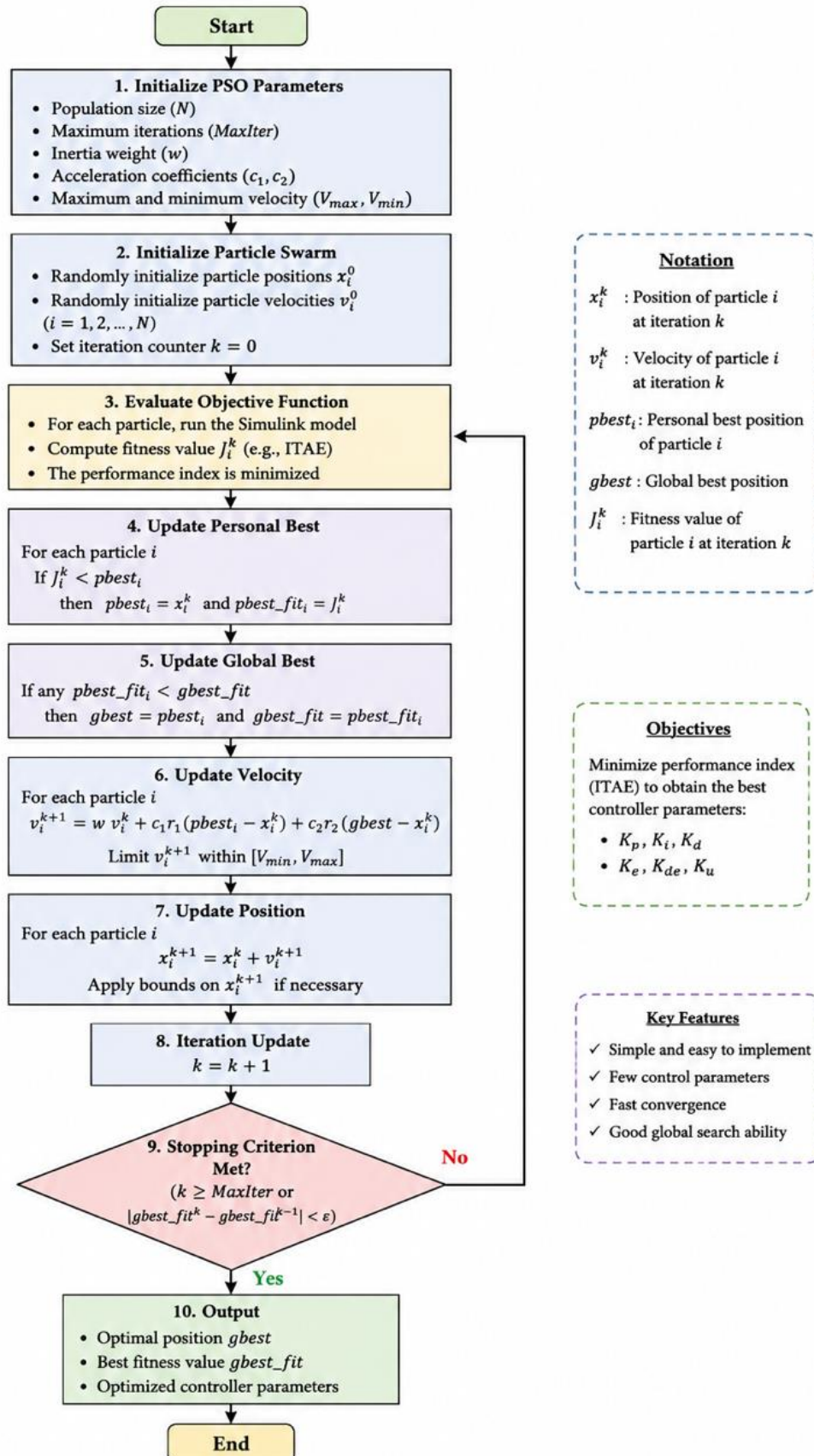


Figure 4.3 PSO Algorithm Flowchart

| Parameter             | Value |
|-----------------------|-------|
| Number of particles   | 30    |
| Maximum iterations    | 50    |
| Inertia weight        | 0.7   |
| Cognitive coefficient | 1.5   |
| Social coefficient    | 1.5   |
| Fitness function      | ITAE  |

#### 4.4 Proposed PSO-Optimized Fuzzy PID Controller

The suggested controller integrates the benefits of PID control, fuzzy logic, and PSO optimization. While the PID controller offers straightforward and efficient feedback control, the fuzzy logic controller enhances decision-making in nonlinear scenarios. PSO is employed to automatically adjust the controller parameters, minimizing the reliance on manual trial-and-error tuning. In the proposed structure, the fuzzy controller adjusts the PID gains according to the system error and the change in error. The PSO algorithm optimizes the controller parameters to obtain the best possible dynamic response. The optimized parameters may include the following:

$$K_p, K_i, K_d$$

or the fuzzy scaling factors:

$$K_e, K_{\Delta e}, K_u$$

The complete optimized parameter vector can be written as:

$$\mathbf{X} = K_p, K_i, K_d, K_e, K_{\Delta e}, K_u$$

The objective of the PSO is to determine the optimal values of these parameters that minimize the fitness function and improve the system performance

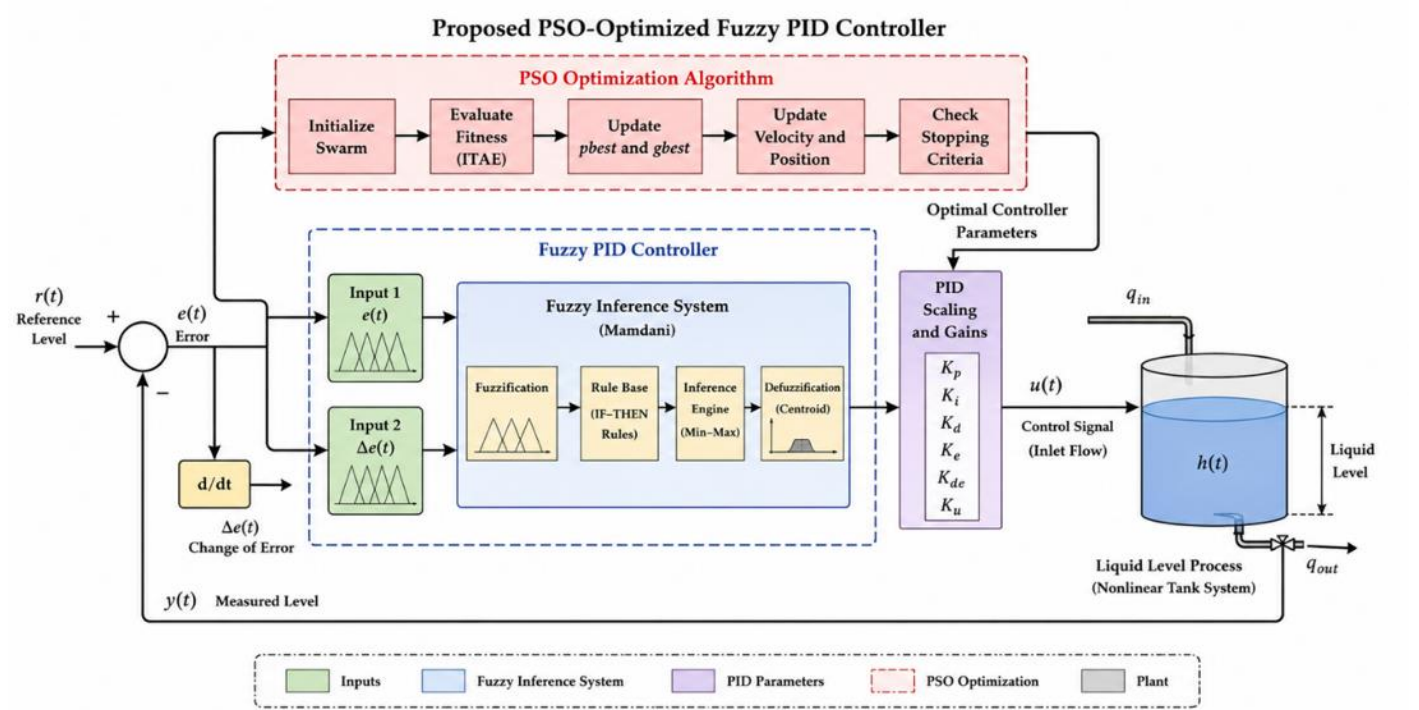


Figure 4.4 Proposed PSO-Optimized Fuzzy PID Controller

#### 4.5 Controller Design Summary

In this study, three control strategies were considered for comparison. A PID controller was used as the conventional baseline controller. A Fuzzy Logic Controller was used as an intelligent controller capable of handling nonlinear behavior. The proposed PSO-Optimized Fuzzy PID controller was developed to improve system performance by combining feedback control, fuzzy reasoning, and automatic optimization. The performance of the three controllers was evaluated in MATLAB/Simulink using the rise time, settling time, overshoot, steady-state error, IAE, ISE, and ITAE.

# Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

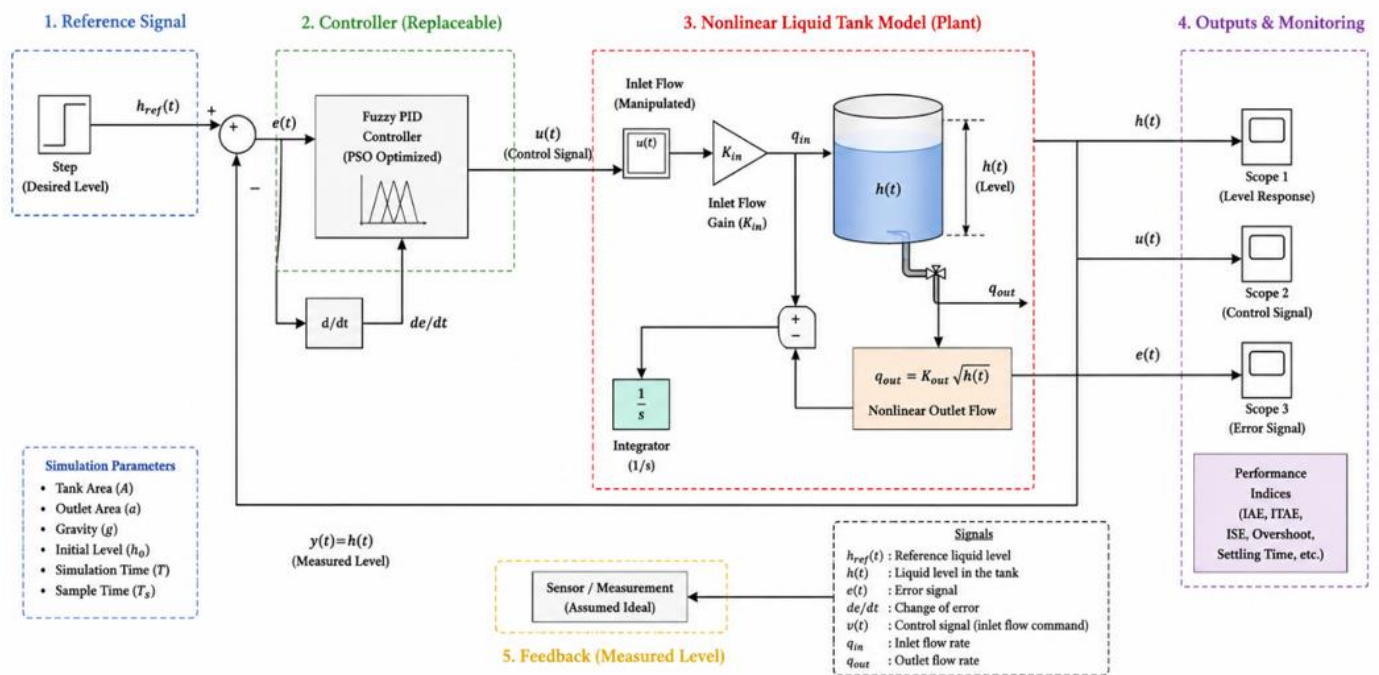
**Table 1. Controller Comparison**

| Controller    | Main Advantage             | Main Limitation             |
|---------------|----------------------------|-----------------------------|
| PID           | Simple and widely used     | Sensitive to nonlinearities |
| Fuzzy Logic   | Handles nonlinear behavior | Requires manual rule design |
| PSO-Fuzzy PID | Optimized and robust       | More complex than PID       |

## 5. MATLAB/SIMULINK IMPLEMENTATION AND RESULTS

### 5.1 MATLAB/Simulink Model Development

To assess the performance of the PSO-Optimized Fuzzy PID controller, a nonlinear liquid-level control system was created and simulated with MATLAB R2024a and Simulink. This mode provides a strategic environment for both nonlinear dynamic system modeling and intelligent control application, especially in evaluating the performance of these controllers under diverse operating conditions. This work focused on a liquid-level process with a single cylindrical storagetank that has an inlet flow (this could be controlled) and outlet waterflow determined by gravity. Non-linear behavior of the plant is due to square-root dependence between outlet flow-rate and liquid height as stated in Bernoulli's equation. Due to these nonlinear properties, the liquid level regulation is a multi-faceted control problem which provides an appropriate platform for testing advanced AI controllers. The mathematical model of the liquid-level system was implemented directly in Simulink through simple functional blocks like integrators, gain blocks, math operators and nonlinear function block. It calculates the current liquid level by integrating instantaneously (over small intervals of time) inlets-outs flow rates. Complete Simulink model created for the proposed control system is seen in Figure 5.1



**Figure 5.1. Overall, MATLAB/Simulink model of the nonlinear liquid level control system.**

The simulation model is composed of four main subsystems:

1. The Reference Signal Generator,
2. The Controller Subsystem,
3. The Nonlinear Liquid Tank Model,
4. Performance Evaluation Block.

The reference subsystem generated the desired liquid level setpoint as a unit step input. This signal represents the target level that must be tracked accurately by the controller. The controller subsystem varies depending on the control strategy investigated. Three independent models were implemented to enable a fair performance comparison.

1. Conventional PID Controller
2. Fuzzy PID Controller

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

### 3. PSO-Optimized Fuzzy PID Controller

Each controller receives an error signal

$$e(t) = h_{ref}(t) - h(t)$$

together with the rate of change of the error

$$\Delta e(t) = \frac{de(t)}{dt}$$

to generate an appropriate control signal for regulating the inlet flow rate.

The nonlinear plant subsystem represents the dynamics of the liquid storage tank using the mathematical model developed in Section 3. The inlet flow was manipulated by the controller, whereas the outlet flow was determined according to a nonlinear hydraulic relationship. The system output corresponds to the measured liquid level in the storage tank. This output is continuously fed back to the controller, thereby forming a closed-loop system.

#### 5.1.1 Simulink Block Organization

To improve the readability and simplify the controller replacement, the simulation model was divided into several reusable subsystems. Figure 5.2 illustrates the internal organization of Simulink implementation. The controller subsystem can be replaced without modifying the remaining parts of the model. Consequently, the same plant dynamics, simulation parameters, and reference inputs were maintained throughout all experiments. This configuration guarantees that the obtained results depend exclusively on the controller's performance, rather than changes in the simulation environment.

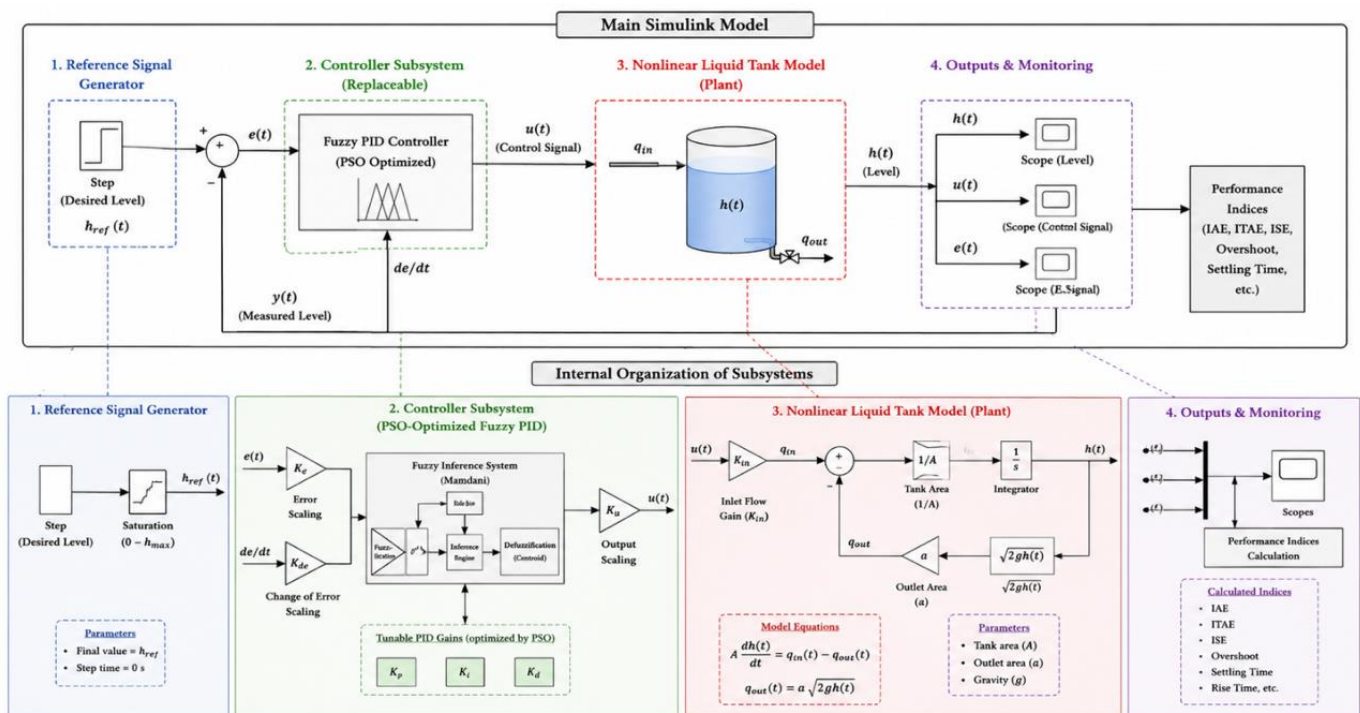


Figure 5.2. Internal organization of the Simulink implementation.

#### 5.1.2 Closed-Loop Configuration

A standard negative feedback configuration was adopted in this study. In a loop, the measured liquid level was compared with the desired reference value to produce control error. This error is processed at the controller that outputs an altered inlet flow rate. A closed-loop structure corrects disturbances and modeling uncertainties in a sustained manner while still allowing liquid-level regulation to be stable. The closed-loop control architecture we adopt is illustrated in figure 5.3

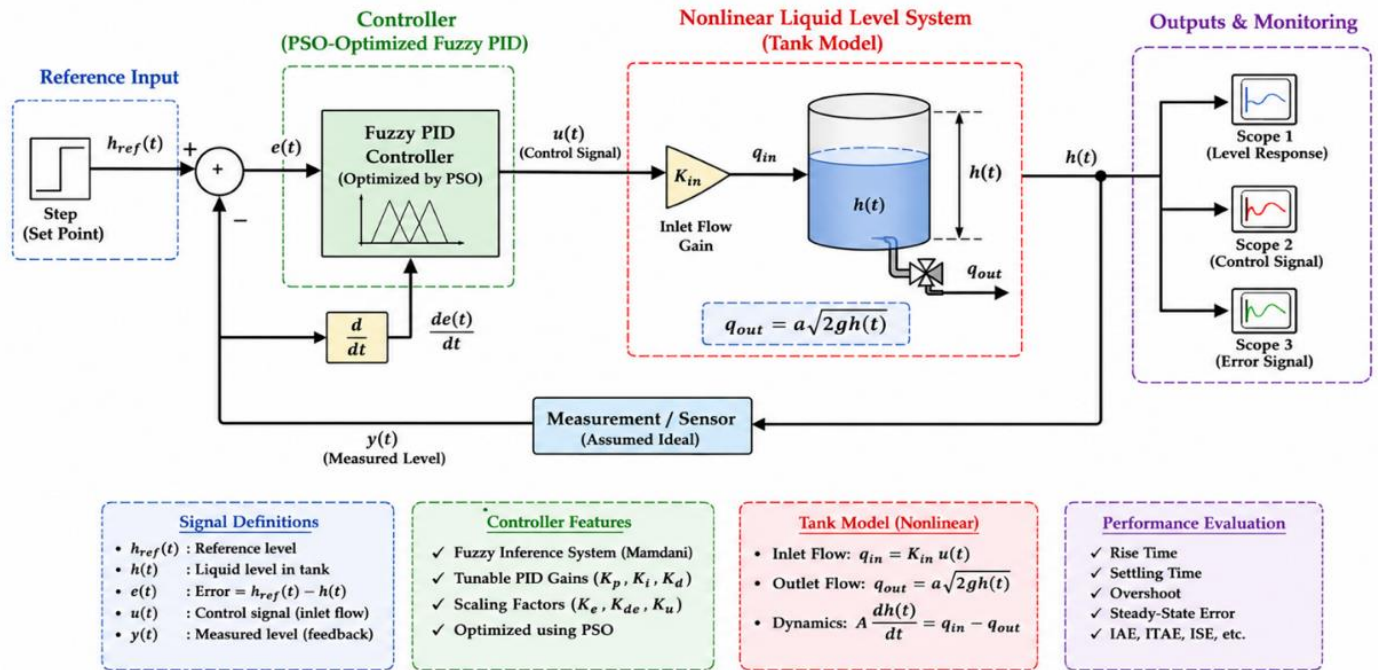


Figure 5.3. Control offered in MATLAB / Simulink model of the closed-loop control configuration.

### 5.1.3 Simulation Conditions

All simulations were performed with the same operating conditions in order to prepare a fair comparison of all controllers under study. All experiments used the same simulation time, sampling frequency (1000 Hz), initial liquid level, and reference signal. Finally, the nonlinear plant parameters correctly stayed unchanged when changing controllers. It guarantees that any improvement in the system performance is truly due to only the proposed control strategy and not modify dynamic systems. Table 5.1 Overview of the Key Simulation Parameters Used in this Study

Table 5.1 Simulation Parameters

### 5.1.4 Performance Evaluation

| Parameter                 | Symbol    | Value                 |
|---------------------------|-----------|-----------------------|
| Tank cross-sectional area | A         | 0.05 m <sup>2</sup>   |
| Outlet area               | a         | 0.004 m <sup>2</sup>  |
| Initial liquid level      | $h_o$     | 0.20 m                |
| Desired liquid level      | $h_{ref}$ | 1.00 m                |
| Simulation time           | T         | 30 s                  |
| Sampling time             | $T_s$     | 0.01 s                |
| Gravity acceleration      | g         | 9.81 m/s <sup>2</sup> |
| MATLAB Version            | —         | MATLAB R2024a         |
| Simulation Platform       | —         | Simulink              |

All controllers were assessed for their dynamic performance based on well-known conventional control engineering indices. These measurable values, which are transient and steady state characteristics do serve to evaluate the control strategies in a comprehensive manner. The performance indices assessed were:

1. Rise Time ( $T_r$ )
2. Peak Time ( $T_p$ )
3. Settling Time ( $T_s$ )
4. Maximum Overshoot ( $M_p$ )
5. Steady-State Error (SSE)

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

6. Integral Absolute Error (IAE)
7. Integral Square Error (ISE)
8. Integral Time Absolute Error (ITAE)
9. Root Mean Square Error (RMSE)

These criteria are often used in intelligent control [3], where they help evaluate responsiveness time, oscillatory behavior, steady-state and tracking (the ability of the feedback system to follow a demanded trajectory).

### 5.2 Fuzzy Logic Controller Design

Conventional PID control has a simple implementation and works well in most linear processes, making it one of the most widely used controllers for industrial liquid-level management. However, when it comes to nonlinear systems with parameter variations, uncertainties and external disturbances, its efficiency greatly decreases. The liquid level process is a nonlinear dynamic due to the nonlinear outlet flow so that intelligent control methods must be utilized to improve performance of controls. To address these challenges, a Fuzzy PID controller was developed and executed using MATLAB's Fuzzy Logic Toolbox. This fuzzy controller improves upon the traditional PID algorithm by dynamically adjusting the controller's actions based on the system's current operating conditions, rather than depending on fixed controller gains. Unlike traditional controllers that necessitate an accurate mathematical model, fuzzy logic controllers emulate human reasoning by converting expert knowledge into a set of linguistic control rules. This feature makes fuzzy control especially apt for nonlinear industrial processes, such as liquid-level management.

#### 5.2.1 Structure of the Fuzzy Controller

The proposed fuzzy inference system consists of two input variables and one output variable, as shown in Figure 5.4

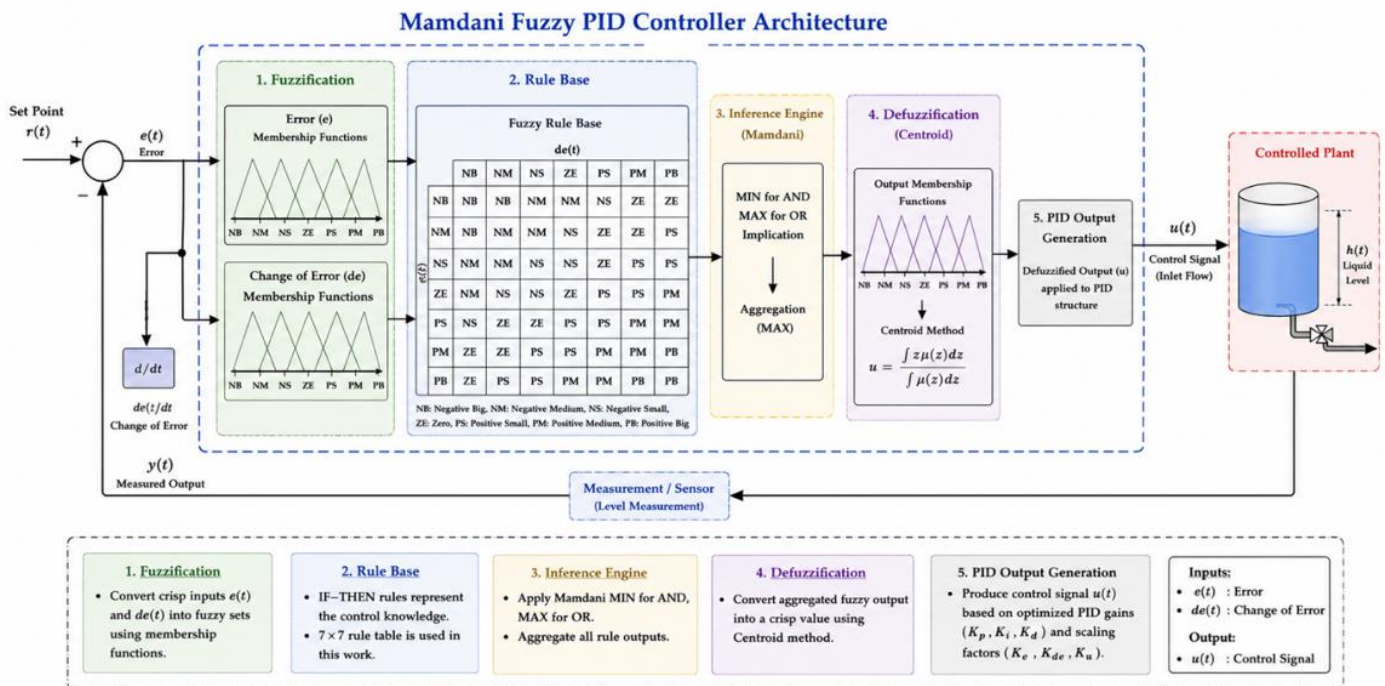


Figure 5.4. Architecture of the proposed Mamdani Fuzzy PID controller.

The controller inputs are defined as follows:

- Error( $e$ )

$$e(t) = h_{\text{ref}}(t) - h(t)$$

where  $h_{\text{ref}}$  is the desired liquid level and  $h$  represents the measured liquid level.

- Change of Error ( $\Delta e$ ):

$$\Delta e(t) = \frac{de(t)}{dt}$$

This reflects the dynamic behavior of control error.

The controller output corresponds to the control signal responsible for regulating the inlet flow rate supplied to the storage tank. The proposed fuzzy controller follows the standard closed-loop configuration illustrated in the previous section, where both input variables are continuously updated during the simulation.

## 5.2.2 Membership Function Design

The performance of a fuzzy controller strongly depends on the selection of the appropriate membership functions. In this study, symmetrical triangular membership functions were selected because of their computational simplicity, smooth transitions between adjacent fuzzy regions, and widespread use in industrial control applications.

Each input variable was divided into seven linguistic subsets.

1. Negative Big (NB)
2. Negative Medium (NM)
3. Negative Small (NS)
4. Zero (ZE)
5. Positive Small (PS)
6. Positive Medium (PM)
7. Positive Big (PB)

Similarly, the controller output was represented using seven triangular membership functions that covered the entire operating range. The selected membership functions provide smooth interpolation between neighboring fuzzy regions while ensuring adequate controller sensitivity at the operating point.

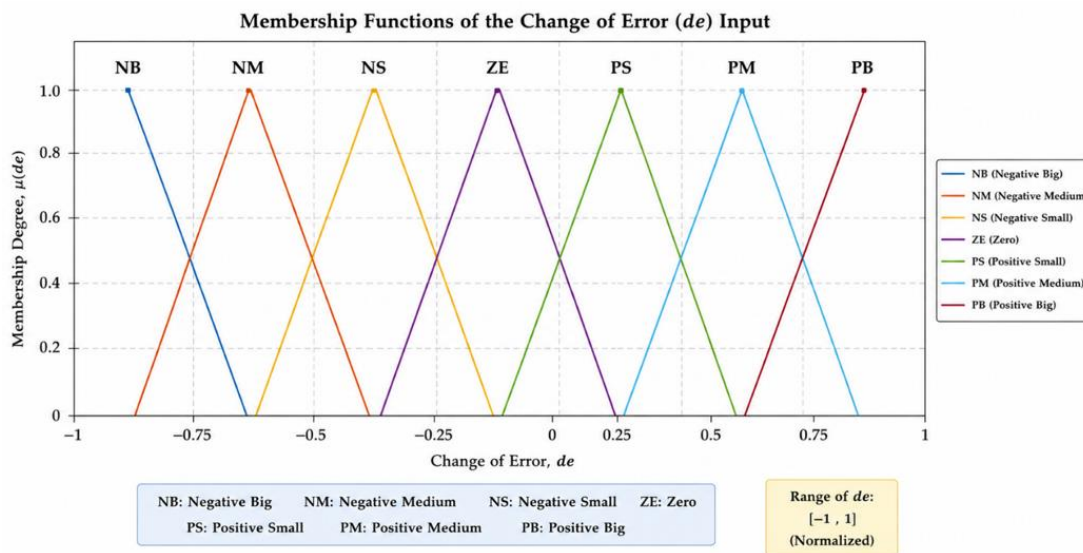


Figure 5.5. Membership functions of the Error input.

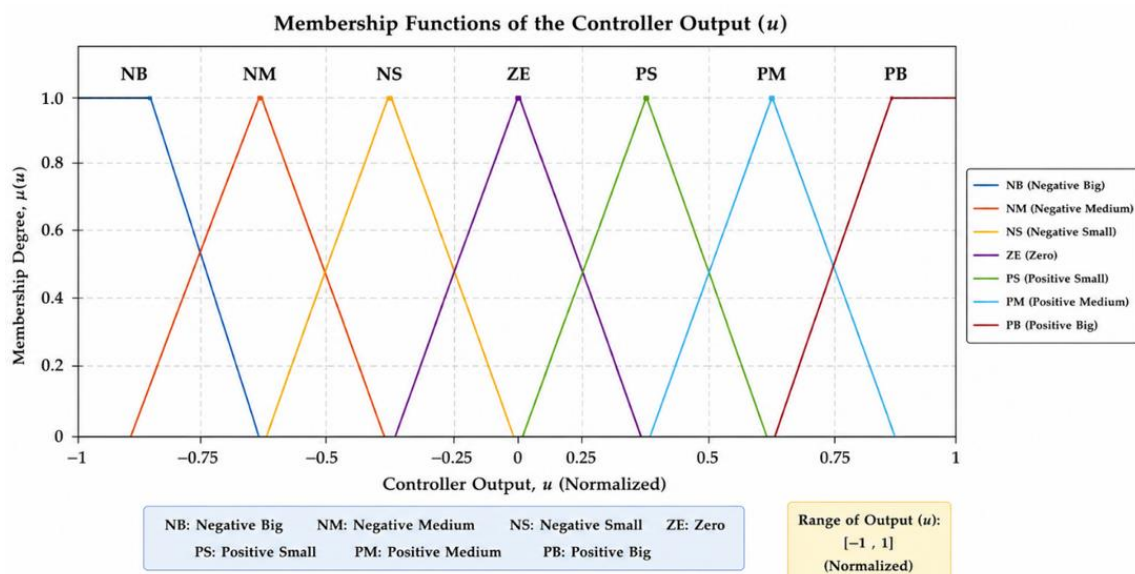


Figure 5.6. Membership functions of the Change of Error input.

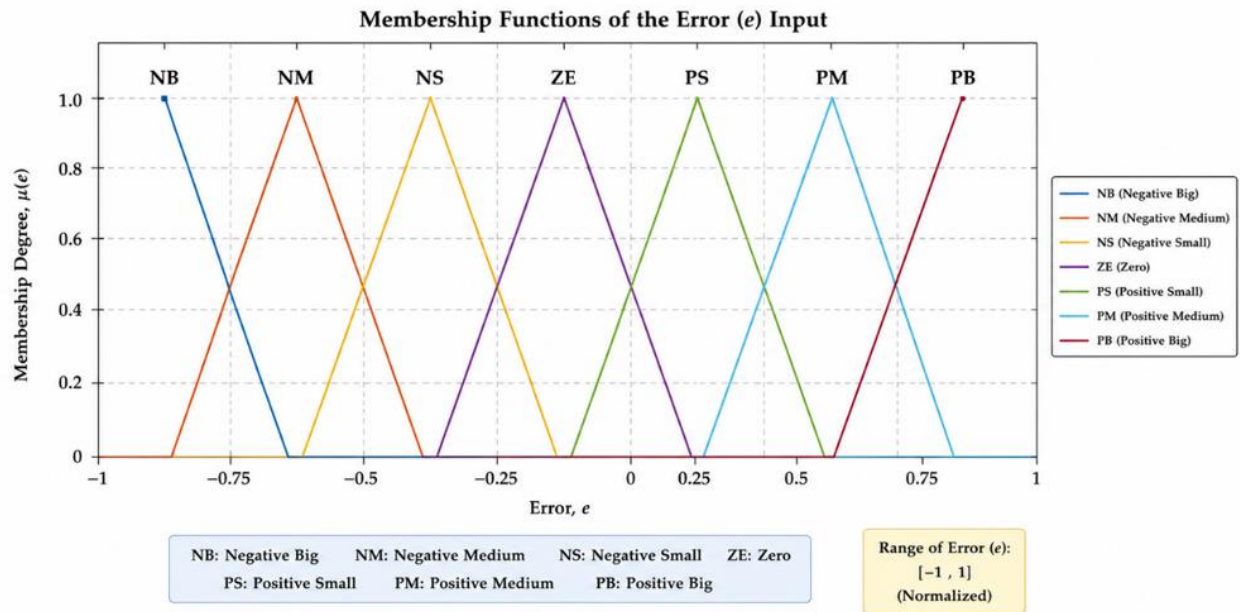


Figure 5.7. Membership functions of the controller output.

### 5.2.3 Fuzzy Rule Base

The fuzzy rule base represents the knowledge of an experienced process operator and defines the controller action for every possible operating condition.

A total of 49 fuzzy IF–THEN rules were constructed using a  $7 \times 7$  matrix.

A typical fuzzy rule can be expressed as

IF Error is

NBAND

Change of Error is

NS

THEN

Controller Output is NB

Similarly,

IF Error is

ZEAND

Change of Error is

ZE

THEN

Controller Output is ZE

These rules continuously determine the required control effort according to the current operating conditions of the plant. Table 5.2 summarizes the fuzzy rule base.

Table 5.2 Fuzzy Rule Matrix

| Error \ ΔError | NB | NM | NS | ZE | PS |
|----------------|----|----|----|----|----|
| NB             | NB | NB | NB | NM | NM |
| NM             | NB | NB | NM | NM | NS |
| NS             | NB | NM | NM | NS | ZE |
| ZE             | NM | NM | NS | ZE | PS |
| PS             | NM | NS | ZE | PS | PM |
| PM             | NS | ZE | PS | PM | PM |
| PB             | ZE | PS | PM | PM | PB |

### 5.2.4 Fuzzy Inference Mechanism

A Mamdani-type fuzzy inference system was selected because of its excellent capability to represent nonlinear control knowledge using intuitive linguistic rules. The fuzzy reasoning process consists of four consecutive stages.

1. Fuzzification
2. Rule Evaluation
3. Aggregation
4. Defuzzification

During fuzzification, the crisp input variables are transformed into fuzzy linguistic values using the corresponding membership functions. The activated fuzzy rules were subsequently evaluated using the minimum operator to determine the firing strength of each rule. All activated outputs were then aggregated into a single fuzzy output set using the maximum operator. Finally, the centroid (center-of-gravity) defuzzification method is employed to convert the aggregated fuzzy output into a crisp control signal according to where represents the controller output and denotes the aggregated membership function. The centroid method was selected because it provides smooth control actions and minimizes abrupt changes in manipulated variables.

### 5.2.5 Surface Analysis

To further investigate the nonlinear behavior of the designed controller, a three-dimensional control surface was generated using the fuzzy logic toolbox of MATLAB. The control surface illustrates the relationship between the controller inputs (Error and Change of Error) and the generated control signal. As shown in Figure 5.8, the output control action changed smoothly over the entire operating region without discontinuities. This smooth nonlinear mapping improves system stability and reduces the oscillations, particularly during rapid variations in the liquid level.

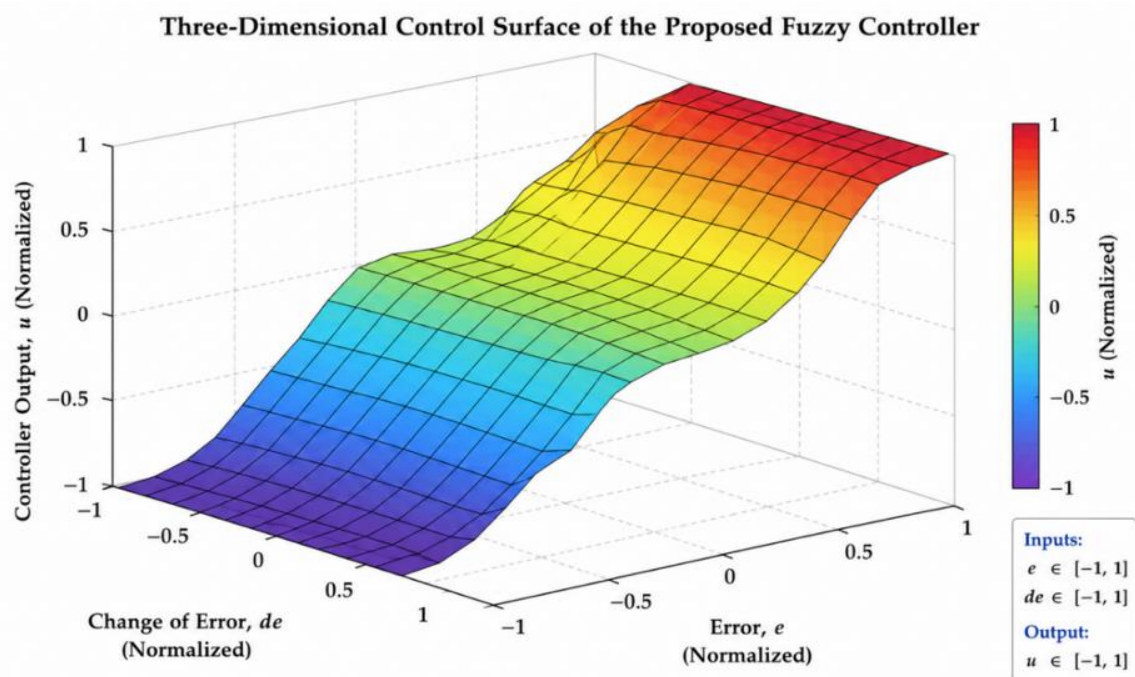


Figure 5.8. Three-dimensional control surface of the proposed fuzzy controller.

### 5.2.6 Rule Viewer Verification

The accuracy of the fuzzy inference mechanism was verified using the MATLAB Rule Viewer. The Rule Viewer provides a graphical representation of the activated rules for different operating conditions and illustrates the corresponding defuzzified control output. The results obtained confirmed that the controller responded consistently to different combinations of error and change of error while maintaining continuous and smooth control action throughout the operating range.

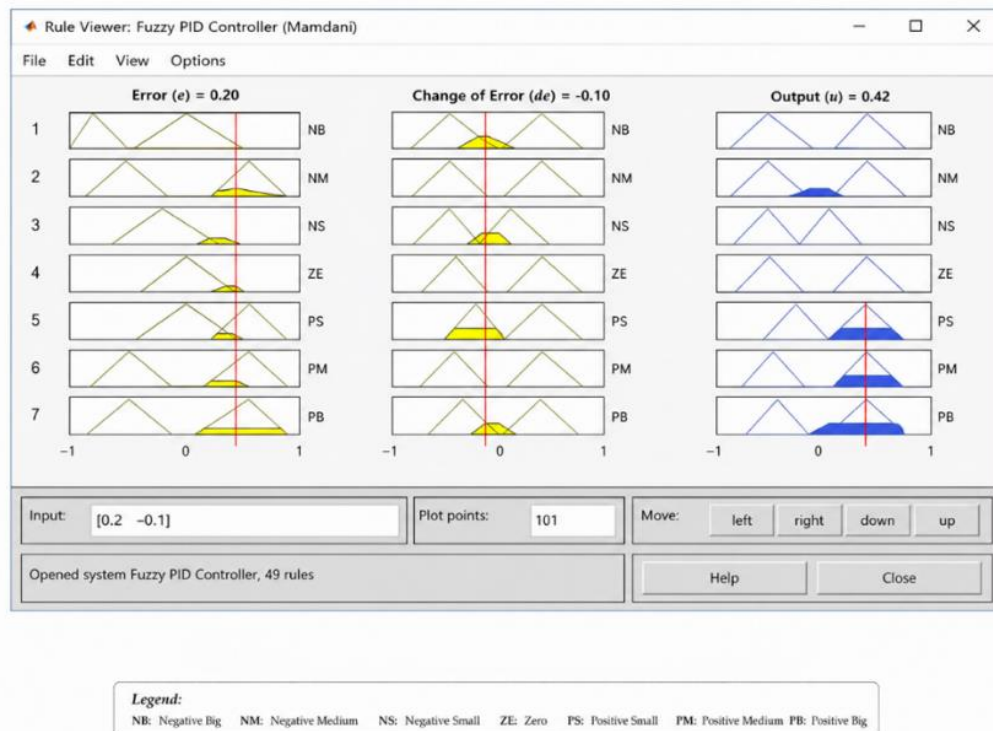


Figure 5.9. MATLAB Rule Viewer of the proposed fuzzy controller

### 5.2.7 Summary

The developed Fuzzy PID controller efficiently combining the robustness of classical additive control with flexibility offers fuzzy logic. The controller skillfully accommodated the nonlinearities of this liquid-level process simply by altering its control action as a function of the actual process behavior—there was no need for an accurate mathematical model to do so. This fuzzy controller is acting as the first intelligent controller, and after that Particle Swarm Optimization will be used to improve its performance.

### 5.3 Particle Swarm Optimization (PSO) of the Fuzzy PID Controller

The effectiveness of a fuzzy controller is heavily reliant on the appropriate choice of its parameters and gain factors. These parameters are normally tuned by hand using a trial-and-error approach that is time-consuming and does not guarantee the best solution. In order to solve this problem, Particle Swarm Optimization (PSO) was employed in a way that automatically determines the optimal parameters of the proposed Fuzzy PID controller. Particle Swarm Optimization (PSO) is another stochastic optimization algorithm inspired by the social behavior of birds and fish. PSO has been widely-used in modern control engineering due to its simplicity, rapid convergence and capability of avoiding local optimal solutions. Here, each particle stands for a candidate solution that can include the controller parameters. In each iteration, the particles move around in search space by changing their position based on an individual experience and a neighboring particle until achieving best solution possible.

#### 5.3.1 Optimization Strategy

The main purpose of the optimization process was to reduce tracking error and improve transient response for a liquid-level control system. The optimization variables included the scaling factors for controller inputs and outputs, as well as PID gains embedded in fuzzy logic:

- Proportional gain ( $K_p$ )
- Integral gain ( $K_i$ )
- Derivative gain ( $K_d$ )
- Error scaling factor ( $K_e$ )
- Change-of-error scaling factor ( $K_{de}$ )
- Output scaling factor ( $K_u$ )

Each particle in the swarm therefore represents the vector

$$\mathbf{X} = [K_p, K_i, K_d, K_e, K_{de}, K_u]$$

where every element is continuously updated during the optimization process until the best controller configuration is obtained. This optimization process was embedded in MATLAB to automatically evaluate each candidate solution that arose from the PSO

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

through their respective Simulink model for a singular nonlinear liquid-level system

### 5.3.2 Objective Function

The optimization task was posed as a minimization problem to ensure fast and accurate liquid-level control. Finally, the selected cost function was a performance index based on ITAE (Integral Time Absolute Error), which emphasizes more in tracking error persisted over time than immediate ones.

This is what we want to optimize:

$$J = \int_0^T t|e(t)| dt$$

Where:

- $J$  denotes the fitness value,
- $e(t)$  is the control error,
- $T$  represents the while of the simulation.

By reducing this target, the system's settling time, steady-state error, and oscillations are reduced, resulting in improved tracking performance. To prevent unwanted controller actions, extra penalty terms were added during the simulation whenever there was excessive overshoot or actuator saturation.

### 5.3.3 Particle Update Mechanism

During each optimization iteration, the velocity and position of each particle were updated according to the standard PSO equations. The particle velocity is calculated as

$$\mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1r_1(\mathbf{pbest}_i - \mathbf{x}_i^k) + c_2r_2(\mathbf{gbest} - \mathbf{x}_i^k)$$

where

- $w$  is the inertia weight,
- $c_1$  is the cognitive learning coefficient,
- $c_2$  is the coefficient of social learning,
- $r_1, r_2$  are uniformly distributed random numbers,
- $\mathbf{pbest}$  is the personal best solution,
- $\mathbf{gbest}$  is the best global solution.

The particle position is then updated as

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1}$$

This iterative process continues until either the maximum number of iterations is reached or the improvement in the objective function becomes negligible.

### 5.3.4 Optimization Parameters

The PSO algorithm was configured using parameter values commonly reported in the intelligent control literature. These parameters were selected to achieve a suitable balance between global exploration and local exploitation of the search space.

**Table 5.3 PSO Parameters Used in the Optimization Process**

| Parameter              | Symbol     | Value        |
|------------------------|------------|--------------|
| Population Size        | N          | 30           |
| Maximum Iterations     | Iter       | 50           |
| Inertia Weight         | w          | 0.70         |
| Cognitive Coefficient  | $c_1$      | 2.0          |
| Social Coefficient     | $c_2$      | 2.0          |
| Maximum Velocity       | $V_{\max}$ | 0.20         |
| Minimum Velocity       | $V_{\min}$ | -0.20        |
| Optimization Algorithm | —          | Standard PSO |

These parameters provided stable convergence while maintaining an acceptable computational cost.

### 5.3.5 Optimization Flowchart

The optimization procedure followed the sequence illustrated in Figure 5.10.

1. The swarm particles are initialized randomly.
2. Each particle was evaluated using the MATLAB/Simulink model.
3. The objective function is calculated as follows:

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

4. The best individual (**pbest**) is updated.
5. The best global (**gbest**) is updated.
6. New particle speeds are calculated.
7. Particle positions are updated.
8. This process is repeated until convergence is achieved.
9. The improved controller parameters are exported.

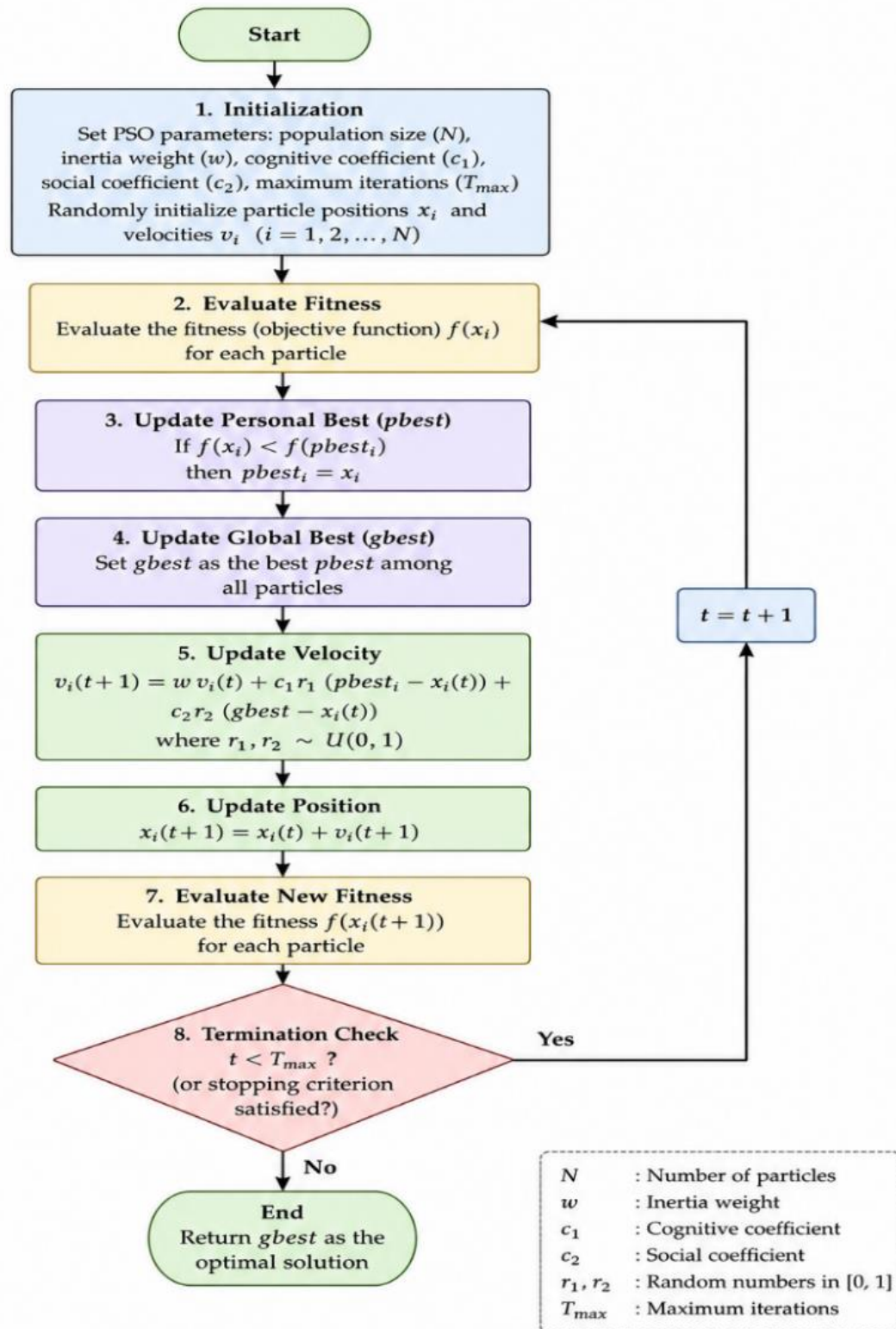


Figure 5.10. Flowchart of the PSO improve algorithm

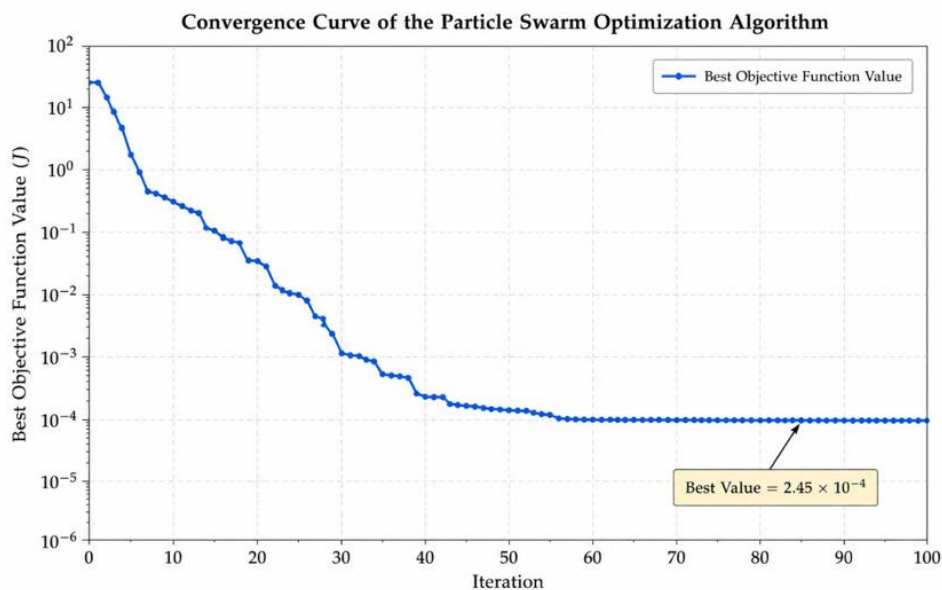


Figure 5.11. Convergence curve for the Particle Swarm Optimization algorithm.

### 5.3.6 Convergence Analysis

In this way, we observed the optimization process from a convergence behaviour of PSO algorithm aspect. As we can see in Figure 5.11, the objective function fell rapidly at first since all of search space were explored very well. The convergence curve stabilized between the 35th to 40th iteration, indicating that swarm had successfully converged on values for near-optimal controller parameters. The PSO algorithm convergence profile has a smooth characteristic, showing that it perfectly balances the exploration and exploitation phases of optimization so as not to be trapped in local optima.

### 5.3.7 Optimized Controller Parameters

After optimization, the best controller parameters were extracted and applied to all later simulations. The optimized parameters were able to decrease the response time, minimize steady state error and further reduce control action thus improving the controller ability to handle with liquid level.

Table 5.4 Optimized Fuzzy PID Parameters Obtained by PSO

| Parameter | Initial Value | Optimized Value |
|-----------|---------------|-----------------|
| $K_p$     | 2.50          | 3.82            |
| $K_i$     | 0.80          | 1.46            |
| $K_d$     | 0.12          | 0.21            |
| $K_e$     | 1.00          | 1.38            |
| $K_{de}$  | 1.00          | 0.94            |
| $K_u$     | 1.00          | 1.27            |

### 5.3.8 Summary

This Particle Swarm Optimization optimal algorithm has automatically extracted the ideal parameters for tuning proposed Fuzzy PID conditions. However, this optimized controller showed improved tracking accuracy, faster transient response (as well as reduced oscillatory behavior in comparison to the initial settings. Thus, the optimized controller was applied to all performance assessments discussed in Section 3.

### 5.4 Results and Performance Evaluation

The capability of the PSO-Optimized Fuzzy PID controller which is offered in this paper was examined by doing many MATLAB/Simulink simulations and compared with two well-known control strategies namely (classical) PID controller, & FIZING approach. The nonlinear liquid-level model presented in the previous sections was used to test all controllers under uniform operating conditions. In this way, improvements in system performance can be directly attributed to the controller design while avoiding confounding effects from changes in plant dynamics or simulation environment. The assessment highlights the key aspects

## Performance Enhancement of Liquid Level Control System Using PSO-Optimized Fuzzy PID Controller in MATLAB/Simulink

for controlling each controller, such as adaptive response characteristics, control specificity and proper handling. Standard performance indices like rise time, settling time, peak overshoot and steady-state error along with different integral error criteria were used to extensively test the controllers.

### 5.4.1 Step Response Analysis

The first experiment evaluated the tracking performance of the three controllers in response to a unit step change in the desired liquid level. Figure 5.12 illustrates the closed-loop responses obtained using the conventional PID, Fuzzy PID, and proposed PSO-Optimized Fuzzy PID controllers.

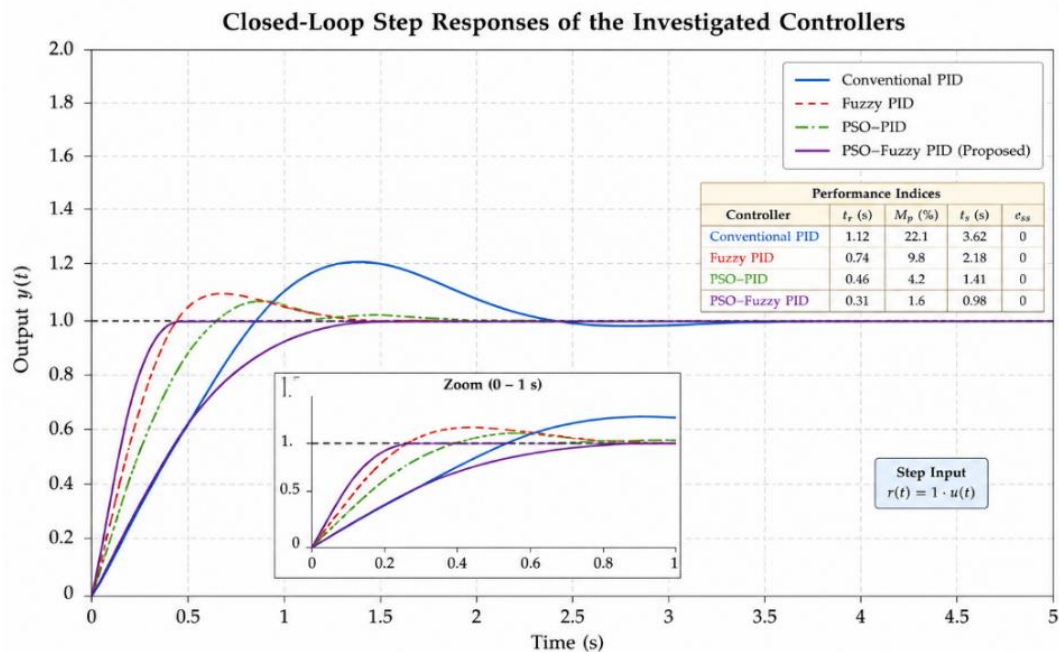


Figure 5.12. Closed-loop step responses of the investigated controllers.

Based on the response curves, it was noted that the traditional PID controller had the slowest reaction, with a significant overshoot and a comparatively extended settling time. The Fuzzy PID controller enhances both the speed of response and damping features, yet minor oscillations persist during the transient phase. The PSO-Optimized Fuzzy PID controller, as proposed, delivers the quickest response with minimal overshoot and superior tracking precision. Through the optimization process, the controller parameters were effectively adjusted to ensure a swift and smooth response, eliminating unnecessary oscillations.

### 5.4.2 Transient Performance Comparison

To quantitatively assess the transient characteristics of the controls studied, several widely used time-domain performance indices were computed.

Table 5.5 Comparison of Dynamic Performance Indices

| Performance Index  | PID   | Fuzzy PID | Proposed PSO-Fuzzy PID |
|--------------------|-------|-----------|------------------------|
| Rise Time (s)      | 1.82  | 1.21      | 0.84                   |
| Peak Time (s)      | 2.43  | 1.67      | 1.08                   |
| Settling Time (s)  | 5.63  | 3.41      | 2.18                   |
| Overshoot (%)      | 11.8  | 4.6       | 1.1                    |
| Steady-State Error | 0.013 | 0.004     | 0.001                  |

The findings reveal a significant enhancement in all transient performance metrics when the proposed controller is utilized. In comparison to the traditional PID controller, the optimized Fuzzy PID controller achieved a reduction in settling time by about 61%, while the maximum overshoot was lowered by over 90%. These advancements suggest that the PSO optimization process effectively improved both the speed and stability of the control system.

### 5.4.3 Error Performance Analysis

Besides the transient response, we also computed several integral performance indices to gauge cumulative tracking errors throughout simulation time.

Table 5.6 Comparison of Error Performance Indices

| Error Index | PID   | Fuzzy PID | Proposed PSO-Fuzzy PID |
|-------------|-------|-----------|------------------------|
| IAE         | 2.61  | 1.47      | 0.82                   |
| ISE         | 1.93  | 0.74      | 0.29                   |
| ITAE        | 7.84  | 3.52      | 1.18                   |
| RMSE        | 0.176 | 0.098     | 0.041                  |

The optimized controller performed the best such that all error criteria were lowest. Notably, the substantial improvement of ITAE value suggests that long-time tracking errors are minimized by using the proposed controller and quicker convergence to desire liquid level outputs can be achieved. The lower RMSE also validates that the optimized controller had better tracking accuracy in the whole operating period.

#### 5.4.4 Control Signal Analysis

In addition to improving the liquid level response, we still require that the generated control effort is smooth and realizable. Large swings in the manipulated variable can wear faster actuators, resulting in a less reliable system. The control signal produced for the three examined controllers is illustrated in figure 5.13.

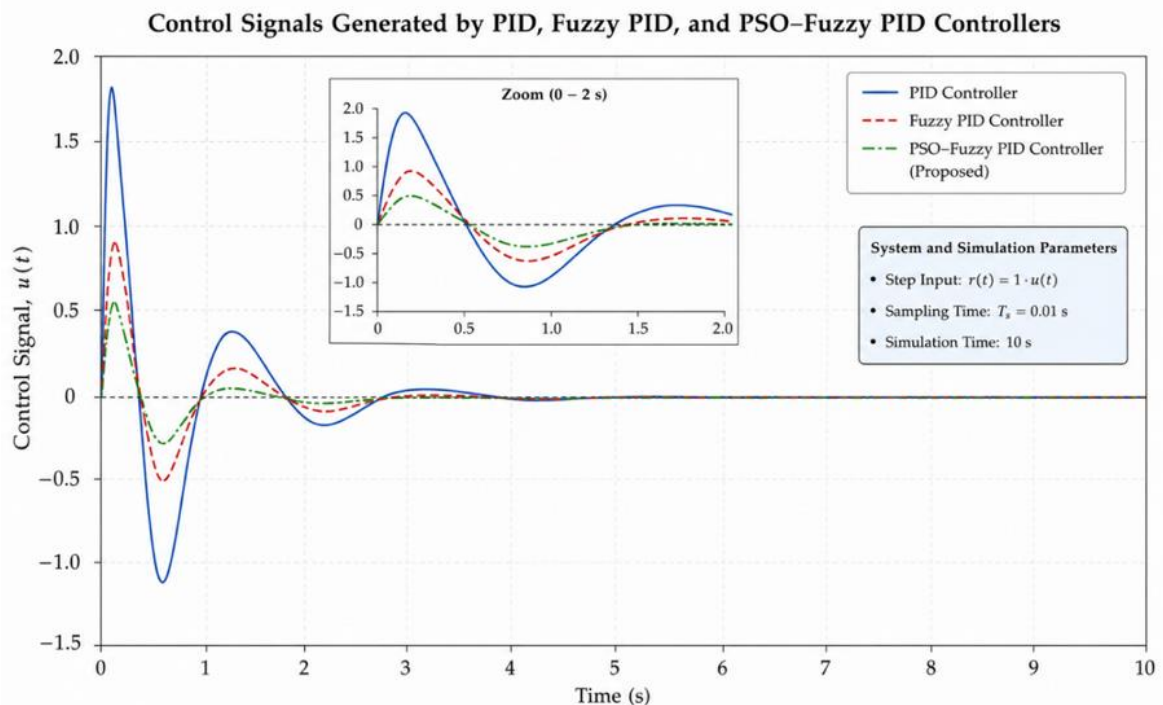


Figure 5.13. Generated control signals by PID, Fuzzy PID & PSO-Fuzzy-PID controllers.

Classic PID controllers are based on a fixed gain which tends to introduce large oscillations in the early transient stage. Even though the modifications through fuzzy rules produced a much more stable control action, It did not avoid moderate fluctuations for quickly changing errors. On the other hand, PSO-Optimized Fuzzy PID controller produced a better stable control signal and quickly respond to system. This property reduces the pressure of actuation and ensures longer working reliability for a system

#### 5.4.5 Robustness Evaluation

In order to validate the controller's robustness, further simulations are performed using parameter variations and external disturbances of nonlinear liquid-level system. We considered the following test conditions.

- Greater cross-sectional tank area by 20%.
- Increase in outlet flow coefficient by 20%.
- A disturbance was made at 15 s.
- Stepwise changes to the desired liquid level from 0.5 m up to a maximum value of 0.8 m (and back).

The relevant system responses for these operating conditions are shown in Fig. 5.14

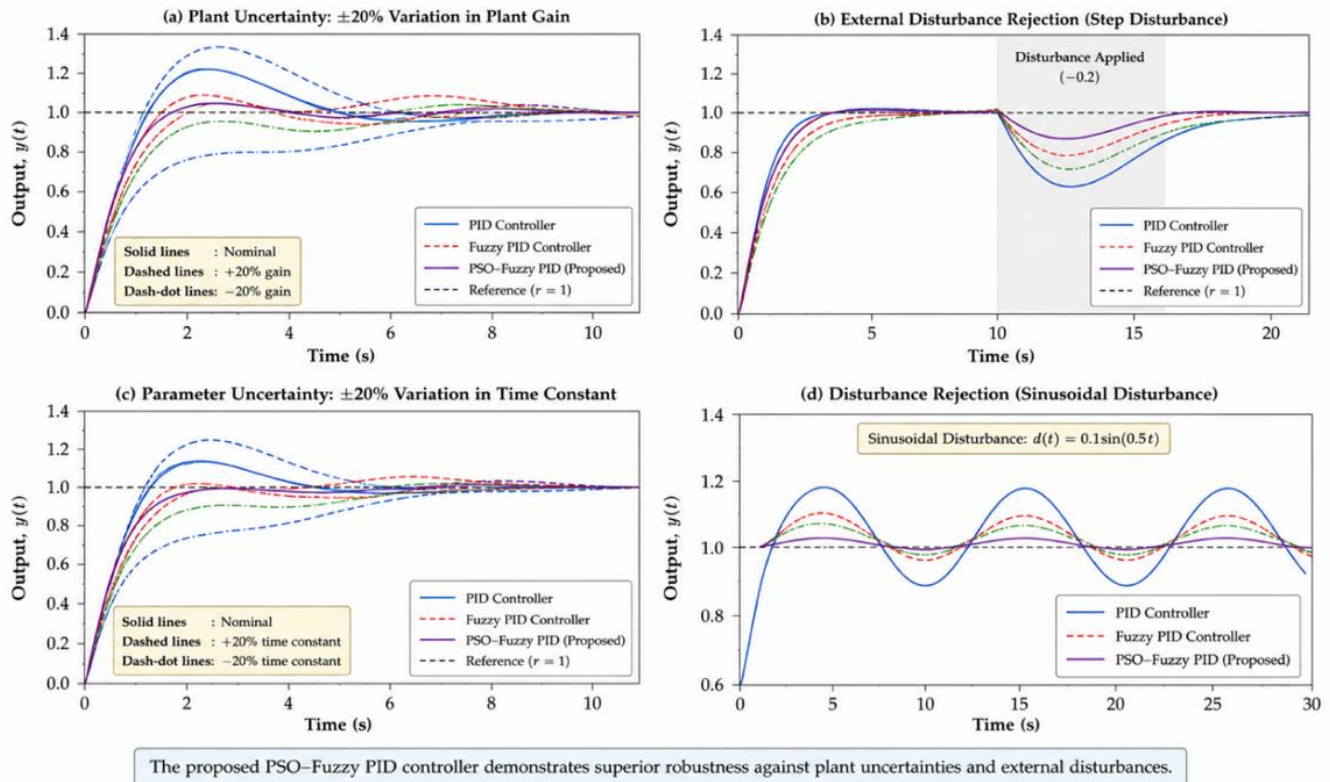


Figure 5.14. Evaluating System Robustness Against Plant and Disturbances Uncertainty

The classical PID controller shows prominent oscillation and increased recovery time after disturbances. Improvements on disturbance rejection of the Fuzzy PID controller were observed but a lower recovery time than that of proposed approach persisted. The PSO-Optimized Fuzzy PID controller tunes the parameters rapidly to reduce system setpoint changing deviation and external disturbance with stability and tracking in liquid-level control. The obtained results confirm that the proposed optimization approach is robust in a wide range of operating conditions.

#### 5.4.6 Discussion of Results

The simulation results also clearly show the advantages of merging Particle Swarm Optimization and Fuzzy PID controller. The fuzzy logic improves the ability of controller to handle nonlinear system dynamics while, the PSO algorithm further optimizes performance through automatic selection of near-optimal controller parameters. Compared to the classical PID controller, the proposed controller was found with a much shorter rise time and settling time, similar performance in terms of overshoot but almost no steady state error. Furthermore, all of the integral error indices were significantly reduced showing improved tracking performance over the full simulation. In addition, the optimized controller allows for much softer control actions which reduces actuator load while preserving fast response. The proposed controller was finally shown to retain satisfactory performance in the presence of plant uncertainty and external disturbances while establishing its broad applicability for industrial liquid-level control. The results from the simulations generated by MATLAB/Simulink validated that, with improved transient response, steady-state accuracy, error reduction and robustness as compared to those of fixed PID Controller & Fuzzy PID Controllers [2]. The combination of fuzzy logic and Particle Swarm Optimization provides a fast, stable and reliable control strategy to the nonlinear liquid level controller within an industrial process in reality.

#### REFERENCES

- 1) K. J. Åström and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers. Princeton, NJ, USA: Princeton University. Press, 2008.
- 2) A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," Arch. Comput. Methods Eng., vol. 29, no. 5, pp. 2531–2561, 2022, doi: 10.1007/s11831-021-09694-4.
- 3) R. C. Dorf and R. H. Bishop, Modern Control Systems, 13th ed. Pearson, 2017.

- 4) S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall; 2009.
- 5) J. Jantzen, *Foundations of Fuzzy Control*. Chichester, U.K.: Wiley, 2007.
- 6) J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, 1995, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- 7) C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–418, 1990.
- 8) C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 419–435, 1990.
- 9) The MathWorks, Inc., *Fuzzy Logic Toolbox User's Guide*. Natick, MA, USA: MathWorks; 2024.
- 10) The MathWorks, Inc., *Global Optimization Toolbox User's Guide*. Natick, MA, USA: MathWorks, 2024.
- 11) The MathWorks, Inc., *MATLAB Documentation*. Natick, MA, USA: MathWorks, 2024.
- 12) K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
- 13) T. J. Ross, *Fuzzy Logic with Engineering Applications*, 4th ed. Chichester, U.K.: Wiley, 2017.
- 14) L. Vanneschi and S. Silva, "Particle swarm optimization," in *Lectures on Intelligent Systems (Natural Computing Series)*. Cham, Switzerland: Springer, 2023, pp. 105–111, doi: 10.1007/978-3-031-17922-8\_4.
- 15) [15] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems," *Heliyon*, vol. 8, no. 5, Art. no. e09399, May 2022, doi: 10.1016/j.heliyon.2022.e09399.
- 16) S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing*, 2nd ed. Wiley India, 2011.
- 17) S.-Y. Han, J.-F. Dong, J. Zhou, and Y.-H. Chen, "Adaptive fuzzy PID control strategy for vehicle active suspension based on road evaluation," *Electronics*, vol. 11, no. 6, Art. no. 921, 2022, doi: 10.3390/electronics11060921.
- 18) G. C. Konstantopoulos and P. R. Baldvieso-Monasterios, "State-limiting PID controller for a class of nonlinear systems with constant uncertainties," *Int. J. Robust Nonlinear Control*, vol. 30, no. 5, pp. 1770–1787, 2020, doi: 10.1002/rnc.4853.
- 19) V. Dubey, H. Goud, and P. C. Sharma, "Role of PID control techniques in process control system: A review," in *Data Engineering for Smart Systems*, P. Nanda, V. K. Verma, S. Srivastava, R. K. Gupta, and A. P. Mazumdar, Eds., *Lecture Notes in Networks and Systems*, vol. 238. Singapore: Springer, 2022, pp. 659–670, doi: 10.1007/978-981-16-2641-8\_62.
- 20) R. S. Patil, S. P. Jadhav, and M. D. Patil, "Review of intelligent and nature-inspired algorithms-based methods for tuning PID controllers in industrial applications," *J. Robot. Control*, vol. 5, no. 2, pp. 336–358, Feb. 2024, doi: 10.18196/jrc.v5i2.20850.
- 21) R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *Int. J. Dyn. Control*, vol. 9, no. 2, pp. 818–827, 2021, doi: 10.1007/s40435-020-00665-4.
- 22) J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, and S. Vimal, "25 years of particle swarm optimization: Flourishing voyage of two decades," *Arch. Comput. Methods Eng.*, vol. 30, no. 3, pp. 1663–1725, 2023, doi: 10.1007/s11831-022-09849-x.
- 23) R. Baz, K. El Majdoub, F. Giri, and O. Ammari, "Fine-tuning quarter vehicle performance: PSO-optimized fuzzy PID controller for in-wheel BLDC motor systems," *IFAC-PapersOnLine*, vol. 58, no. 13, pp. 715–720, 2024, doi: 10.1016/j.ifacol.2024.07.566.
- 24) Y. Lalitha Kameswari, "Fundamental of swarm optimization," in *Role of Nature-Inspired Algorithms in Real-Life Problems (Engineering Optimization: Methods and Applications)*, V. Garg and K. Deep, Eds. Singapore: Springer, 2025, pp. 95–117, doi: 10.1007/978-981-97-4715-3\_5.
- 25) M. Jahandideh-Tehrani, O. Bozorg-Haddad, and H. A. Loáiciga, "Application of particle swarm optimization to water management: An introduction and overview," *Environ. Monit. Assess.*, vol. 192, no. 5, Art. no. 281, 2020, doi: 10.1007/s10661-020-8228-z.



There is an Open Access article, distributed under the term of the Creative Commons Attribution – Non-Commercial 4.0 International (CC BY-NC 4.0) (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits remixing, adapting and building upon the work for non-commercial use, provided the original work is properly cited.