ISSN: 1672 - 6553

JOURNAL OF DYNAMICS AND CONTROL

VOLUME 10 ISSUE 03: P249-267

PERFORMANCE COMPARISON OF AN OPTIMIZED PID CONTROLLER AND ADVANCED CONTROL STRATEGIES FOR LIQUID LEVEL CONTROL SYSTEM

Anil Kadu, Sandesh Rathod, Vaidehi Babar, Prathamesh Vadde, Neeraj Pathak

Instrumentation and Control Engineering,
Vishwakarma Institute of Technology, Pune,
Maharashtra, India

JOURNAL OF
DYNAMICS AND CONTROL

https://doi.org/10.71058/jodac.v10i03010

PERFORMANCE COMPARISON OF AN OPTIMIZED PID CONTROLLER AND ADVANCED CONTROL STRATEGIES FOR LIQUID LEVEL CONTROL SYSTEM

Anil Kadu[1], Sandesh Rathod[2], Vaidehi Babar[3], Prathamesh Vadde[4], Neeraj Pathak[5]
Instrumentation and Control Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India
anil.kadu@vit.edu[1], sandesh.rathod23@vit.edu[2], vaidehi.babar24@vit.edu[3], prathamesh.vadde231@vit.edu[4], rakesh.neeraj23@vit.edu[5]

Abstract— A comparative exploration between PID optimizing techniques and advanced liquid level controls for process industries is provided in this paper. In process industries, ensuring accurate fluid level measurement is necessary for the system's safety and efficiency. In this paper two advanced control strategies (Active Disturbance Rejection Control (ADRC) and Model Reference Adaptive Control (MRAC)) will be compared with optimized PID controllers using Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Simulated Annealing (SA), and Grey Wolf Optimization (GWO). The PID control of a second-order control model of a single-tank will be simulated, as well as evaluated using real-time experiments. The time domain parameters to be analysed will include rise time, settling time, overshoot, and steady-state error for the various controllers. Results indicate ADRC outperforms PID in terms of disturbance rejection (minimal overshoot and a settling time around 20 seconds). Alternatively, the use of GWO or GA, in optimally tuning a PID controller, will greatly reduce the steady-state error compared to traditional tuning methodologies, showing that advanced control systems are more robust and perform better overall than traditional liquid level control systems.

Keywords— *ADRC, MRAC, PID Optimization, PSO, GA, GWO, SA, Advanced Control, Liquid Level Control*

1.INTRODUCTION

Many manufacturing facilities rely on the precise manipulation of liquids used to produce various products, including chemicals, pharmaceuticals, and treated water producers. Proper liquid level control is critical in maintaining the overall stability, safety, and quality of a product or system. If a fluid level is not controlled properly for example in an industrial steam boiler system this can result in very dangerous conditions such as lack of water or causing an overflow of water from the boiler. The PID controller is commonly utilized because it is simple to implement, easy to maintain and produces predictable results in industrial processes. The problem with the PID controller is that it uses fixed gain parameters for all systems. Therefore, when utilizing PID controllers for nonlinear or time-variant systems, the controller may not respond properly due to variations in the inflow rate, valve characteristics or environmental changes. There have been several attempts at creating advanced process control systems to improve upon the traditional PID controller limitations. For example, Active Disturbance Rejection Controller (ADRC) is an advanced process control technique that uses ESO to measure and compensate for unknown disturbances during real-time operation of the control system. Another example is Model Reference Adaptive Control (MRAC), which uses dynamic adjustment of the PID controller gains so that the controlled output of the system traces a pre-defined reference model

One way to enhance the efficiency of PID controllers is the application of optimization methods aimed at automatically tuning the parameters of the controllers. One way of achieving this is through the use of bio-inspired optimization methods, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimization (GWO) and Simulated Annealing (SA) algorithms, which are commonly used to find the optimal values of the PID gains by minimizing some pre-defined performance metrics.

The main objective of the current study is to conduct a comparative analysis between the performance of the advanced control techniques known as Active Disturbance Rejection Control (ADRC) and Model Reference Adaptive Control (MRAC) against that of PID controllers that have been optimally tuned using the PSO, GA, GWO

https://doi.org/10.71058/jodac.v10i03010

and SA algorithms. In order to evaluate the performance of all controllers considered in the analysis, a second order liquid level control model was developed, which allowed for an evaluation of each of the controller types in relation to time domain performance specifications, including: rise time, settling time, overshoot and steady state error.

2. LITERATURE REVIEW

Active Disturbance Rejection Control (ADRC) is an effective control method for dealing with uncertainties, disturbances, and non-linear behaviour in dynamic systems. Whereas model-based methods use accurate mathematical models, the emphasis in ADRC is to estimate and compensate for the total disturbance on a system in real time, while also providing less reliance on accurate mathematical models. A simulation study by Herbst shows that ADRC is a valid tool which maintains stability despite uncertainty and/or disturbances in the system [2]. attempted to combine ADRC with other methods, such as Sliding Mode Control-(SMC) [3] alongside BP Neural Networks [3], to further enhance the speed and accuracy of estimation in induction motor-drivers. An ADRC system contains 3 elements; tracking differentiator (TD), extended state observer (ESO), and nonlinear state error feedback (NLSEF). The three components of Anna's Control System will work in conjunction to estimate system states, estimate system disturbances, and generate appropriate control actions [3]. To assist those interested in applying the ADRC, Lakomy etal developed an ADRC toolbox for MATLAB/Simulink that consists of modular building blocks to create ADRC controllers and reduces the difficulty of implementing an ADRC control approach [1]. Additionally, there are also methods of adaptive ADRC control techniques that use intelligent approaches including neural networks and sliding mode control for applications including induction motor drives in which estimated disturbances and the performance of the control system are increased through the use of intelligent methods [3]. Additionally, simulations show that MATLAB/Simulink is a suitable environment to model and implement ADRC based control for dynamic control applications [4]. Model Reference Adaptive Control (MRAC) can be considered an adaptive control technique that is frequently used to control systems having uncertain parameters or time-varying dynamics. A controller, using a reference model that represents the desired behaviour of the system, will be able to automatically adjust differences in its controlled parameters such that the output of the system will follow the output of the reference model. Romdlony et al. have demonstrated that the adaptive control approach provides better stability and improved tracking performance when implemented against conventional controllers by applying it to a ball-and-beam system [8]. The implementation of techniques which rely on optimization when modifying PID controllers has become quite common as they can be used to automatically find the best possible values of the PID controller parameters. However, using conventional tuning techniques (such as Ziegler–Nichols) with nonlinear systems or systems with varying operating conditions results in the use of tuning values that may not produce the best controller performance. As a result, metaheuristic optimization algorithms (such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA)) have been utilized for PID Controller tuning in order to automatically identify optimal values.

performed by Rais et. al. on the use of optimization algorithms for performing automatic PID tuning with automatic voltage regulator systems demonstrated that using tuning methods based on optimization can provide better transient response and steady-state performance than conventional tuning methods [5]. In a similar manner, research completed by Massoud and Libby on PID tuning in use of evolutionary algorithms and pneumatic soft robotic systems provided improved system stability and reduced control error [6]. Other nature-based algorithms have also been used in order to provide tuning of PID Controllers. As an example of this, Hesham et al. documented the results obtained by comparing various optimization methods when designing PID Controller performance with each method comparative to the others and reported improved controller performance using algorithms such as Teaching Learning Based Optimization and Bat Algorithm [7]. In particular, the Grey Wolf Optimization (GWO) algorithm has become popular for providing an effective balance between exploration and exploitation when searching an optimization space.

https://doi.org/10.71058/jodac.v10i03010

Automatic Voltage Regulator (AVR) systems [5]. These are commonly compared to other algorithms such as Teaching- Learning Based Optimization (TLBO), Bat Algorithm (BA), alongside Monarch Butterfly refactoring [7] to find the most efficient method of tuning. Although GA and PSO are well-proven algorithms for global optimization [6], In [10] the Grey Wolf Optimization-(GWO) algorithm has been proven to be more efficient by imitating the leadership and hunting process of grey wolves.GWO is most efficient in learning multi-layer perceptron parameters and nonlinear optimization problems, outperforming traditional Ant Colony Optimization-(ACO) and evolutionary strategies in many benchmarks. G. Qi et al. [11] proposed a distributed model predictive control (DMPC) approach for oil well liquid level systems. The method improves coordination and stability in multi-well operations compared to centralized control. S. E. Oltean et al.[12] applied model reference adaptive control (MRAC) to slow level control processes. The approach effectively handles parameter variations and ensures stable tracking performance. M. H. Hassani et al. [13] developed a Lyapunov-based neural PID controller for real-time tank level control. The method provides robust and adaptive performance without requiring gradient-based optimization. S. K. Pandey [14] presented a comparative study of optimization techniques like PSO and GA for PID tuning. The results show improved response and stability using optimization-based methods over conventional tuning. R. Ranjan[15] proposed a robust MRAC strategy for liquid level control in process industries. The approach maintains system performance in the presence of disturbances and model uncertainties. Kadu and Khandekar [16] proposed an optimized decentralized PID controller for TITO systems is presented using the Grey Wolf Optimization (GWO) algorithm. The proposed method effectively tunes PID parameters to reduce loop interactions and improve overall system performance. Experimental results demonstrate enhanced stability and improved transient response compared to conventional tuning approaches. A Kadu et al.[17] proposed a low-cost water level monitoring and control system using a PID controller. The system focuses on affordability and practical implementation while maintaining reliable level regulation. The results demonstrate effective performance, making it suitable for small-scale and cost-sensitive applications.

To date, there has been no adequate research into how these advances in advanced control strategies and optimization based PID tuning have been compared across a number of different types of controls/techniques (adaptive vs. optimization). Prior literature on the comparative study of ADRC, MRAC, and optimized PID controllers for liquid level control systems using both simulations and experiments has been limited. Additionally, many of the previous studies were based primarily on the simulation results without validating experimentally. Accordingly, this study aims to compare both the ADRC, MRAC, and optimized PID controllers for liquid level control systems through both a simulation and experimental process, including evaluation of standard time-domain performance measures.

3. EXPERIMENTAL SETUP AND SYSTEM IDENTIFICATION

3.1 Experimental Setup

Figure 1 shows a laboratory-scale liquid-level control system that includes a level transmitter and a data acquisition device used as an experimental setup. The liquid level inside the tank is measured using a pressure-based level transmitter manufactured by WIKA, operating on the principle of hydrostatic pressure. The level transmitter provides a thermally unstable digital output (4-20 mA) corresponding to the amount of water present in the tank. through converting the pressure exerted on the level transmitter (i.e., the pressure of liquid above the level transmitter) into a standard output signal of 4-20 mA (4 mA being the lowest level measured/zero water in the tank and 20 mA being maximum level of liquid measured/maximum water in the tank) and operating on 24 volts dc power supply voltage.

The hydrostatic level transmitters measure the pressure exerted on them by the column of liquid and convert that pressure into an electrical signal proportional to the total height of the liquid. The received signal is collected via a NI USB 6001 data acquisition (DAQ) card with +/- 10 V of analog input and a sample rate of up to 20000 samples per second to feed data back from DAQ to control and acquire real time data. MATLAB/Simulink is used to process the data and test different PID tuning methods and draw a comparison between the two types of adaptive response control (ADRC, MRAC) using only the results from the previously presented simulation studies. In this study, the software shown in Fig. 2 is used to run the PID values obtained from each algorithm and to verify whether these values perform effectively for the given system.

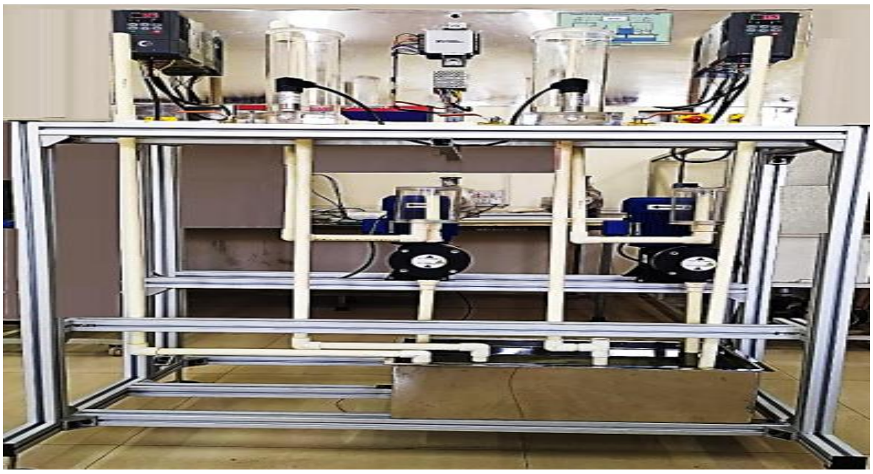
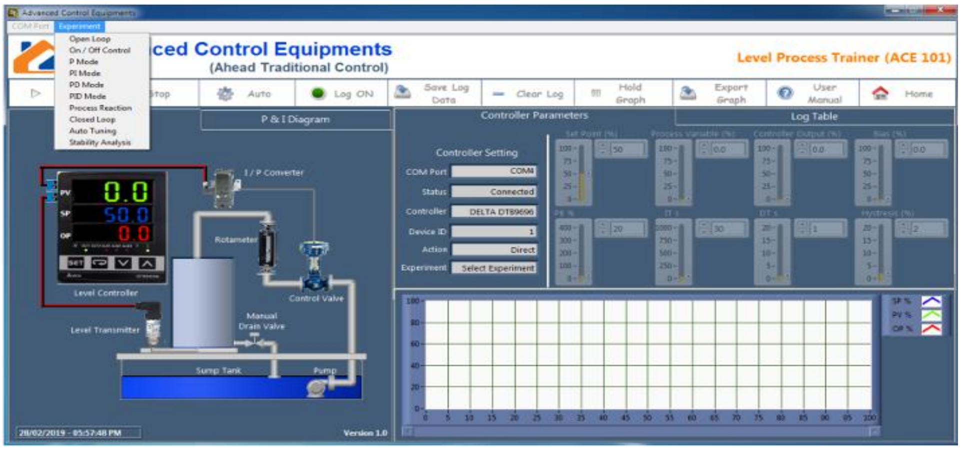


Fig. 1 Water Level Control System



Controller Setting
COM Port: COM4
Status: Connected
Controller: DELTA DT89696
Device ID: 1
Action: Direct
Experiment: Select Experiment
Set Point (%): 50
Process Variable (%): 0.0
Controller Output (%): 0.0
Bias (%): 0.0
PB %: 20
IT s: 30
DT s: 1
Hysteresis (%): 2

Fig. 2 The software is used to run the PID values obtained from all algorithms.

3.2 System Identification

In order to model a mathematical representation of the actual liquid level control process, a step response experiment was carried out. The experimental setup allowed for the step input to be applied to the inlet flow rate, as well as for the real-time output response of the water level to be measured.

Upon collecting the input/output data from the experiment, the System Identification Toolbox in-MATLAB was employed to analyze the data and identify the first- and second-order models that best fit the data collected. This process of modeling allowed the system to create a transfer function model that will closely mimic the real-world performance of the control system

https://doi.org/10.71058/jodac.v10i03010

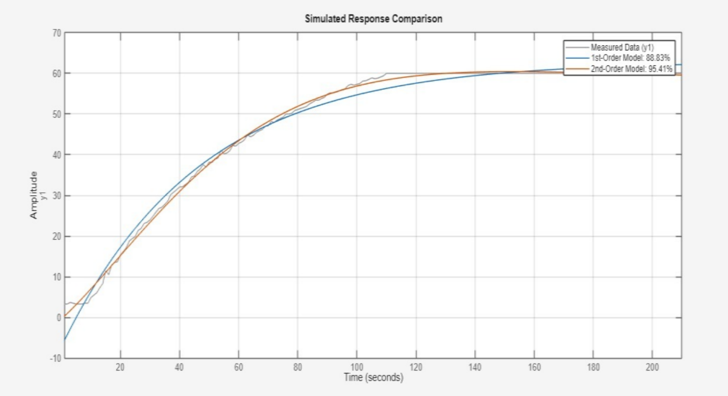


Fig.3 Experimental graph of system identification

3.3 Model Estimation and Validation

Experimental data was processed using the MATLAB System Identification Toolbox to identify a first-order approximation and a second-order approximation of the system's transfer function. In fig.3, The first-order approximation provided a reliable estimate with 88.83% accuracy, effectively capturing the system's dominant time constant and steady-state gain. Upgrading to a second-order model improved the fit to 95.41%. This model provides a more rigorous baseline for designing the ADRC and MRAC control strategies.

i) Obtained the transfer function of the system

The findings obtained from the experimental step-response data and the system identification method used with the MATLAB System Identification Toolbox provide a better description of the liquid level process using a second-order linear transfer function. The equation representing the identified transfer function of the plant is represented as *G(s) = From input "u1" to output "y1"*

$$G(s) = \frac{0.0008074}{s^2+0.0441s+0.0008222} \tag{1}$$

The denominator constant of 0.0008222 also determines the natural frequency of the liquid level system. The natural frequency is approximately 0.0287 rad/sec.

The very low natural frequency indicates that liquid level systems are slow systems because they are designed to charge up and accumulate fluid mass.

ii) Differential Equation

Translating the transfer function into the time domain provides the governing differential equation used for the design of the ADRC and MRAC laws:

$$\ddot{y}(t) + 0.0441\dot{y}(t) + 0.0008222y(t) = 0.0008074u(t) \tag{2}$$

Where,

- y(t) represents the liquid level (output).
- u(t) represents the control signal (input).

3.4 ADRC DESIGN

The ADRC design considers a second-order system of the form:

$$\dot{y} = f + b_0 u \tag{3}$$

In this equation:

- $\dot{y}$: Output of the system.
- f: Represents the "generalized disturbance," which includes internal dynamics, coupling, and external disturbances.
- b_0: A gain parameter.
- u: Control input.

We choose the known gain b_0 from numerator of the Transfer function & lump all other terms into the total disturbances f(t)

$$\text{Known gain } (b_0) = 0.0008074 \tag{4}$$

$$f(t) = -(0.0441\ \dot{y} + 0.0008222\ y) \tag{5}$$

$$\omega_n^2 = 0.0008222 \tag{6}$$

$$\omega_n = 0.02868\ rad/s \tag{7}$$

- Controller bandwidth (ω_c) depends on the ω_n
- The plant is very slow, therefore, choose controller bandwidth that is 10 times faster than the plant. ω_n.

$$\begin{aligned}\omega_c &= 10 * 0.02868\\ &= 0.2868\\ &= 0.3\ rad/s\end{aligned} \tag{8}$$

$$\begin{aligned}\text{Observer bandwidth } (\omega_o) &= 10 * \omega_c\\ &= 10 * 0.3\\ &= 3.0\ \text{rad/s}\end{aligned} \tag{9}$$

- Controller gains,

$$\begin{aligned}k_p &= \omega_c\hat{}2\\ &= 0.3\hat{}2\\ &= 0.09\end{aligned} \tag{10}$$

$$\begin{aligned}k_d &= 2 * \omega_c\\ &= 2*0.3\\ &= 0.6\end{aligned} \tag{11}$$

- Observer gains,

$$\begin{aligned}B_{01} &= 3\omega_o\\ &= 9.0\end{aligned} \tag{12}$$

$$\begin{aligned}B_{02} &= 3\omega_o\hat{}2\\ &= 27.0\end{aligned}$$

https://doi.org/10.71058/jodac.v10i03010

$$\begin{gathered} B_{03} = \omega_o{}^\wedge 3 \\ = 27.0 \end{gathered} \tag{13}$$

i. Extended State Observer (ESO)

The ESO is the core component that estimates the states and the generalized disturbance. The system is rewritten into state-space form with an added state $\dot{x}_3$ representing the generalized disturbance ($\dot{f}$):

$$\begin{gathered} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + b_0 u \\ \dot{x}_3 = h \\ y = x_1 \end{gathered} \tag{14}$$

Where, $x_3 = f$ and h $= \dot{f}$

The ESO itself takes the following form:

$$\begin{gathered} \hat{x}_1 = \hat{x}_2 + g_1(e_1) \\ \hat{x}_2 = \hat{x}_3 + b_0 u + g_2(e_1) \\ \hat{x}_3 = g_3(e_1) \end{gathered} \tag{15}$$

Where $e_1 = x_1 - \hat{x}_1$ is the estimation error and g_i is a generic linear or nonlinear function.

ii. Control Law and Disturbance Rejection -

If the estimation is accurate, the control input u is set to:

$$u = \frac{1}{b_0}(u_0 - \hat{x}_3) \tag{16}$$

This compensation reduces the system to a double integrator system:

$$\dot{y} = u_0 \tag{17}$$

i. **ADRC Simulink implementation -**

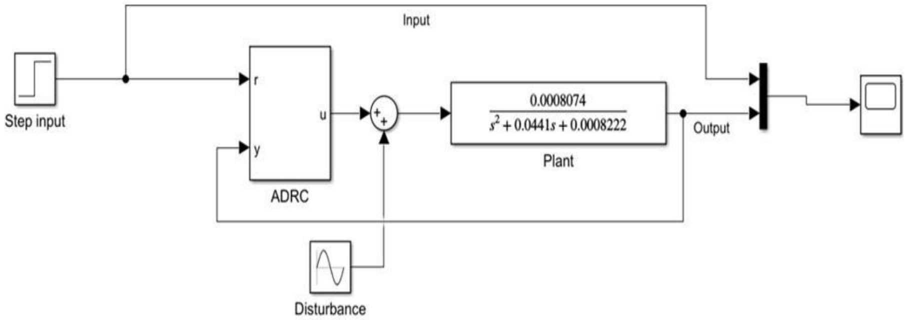


Fig. 4 ADRC Controller Implemented in Simulink

https://doi.org/10.71058/jodac.v10i03010

In Fig. 4, we see a large rectangle representing the physical system being controlled. This is referred to as a second-order system. The block named "ADRC" functions as the controller for the physical system. Unlike standard PID controllers that simply react to an error value, ADRC takes a more active approach by using two inputs:

- r (Reference Input): The desired setpoint input; this input originates from the "Step Input" block.
- y (Output Input): The actual value output from the physical system.

The "Disturbance" block (which looks like a sine wave source) is added at the output of the controller right before it enters the "Plant" block. It simulates external forces on the plant's output from sources other than the physical system, including leaks, pumps that turn on or off, or other forms of sensor noise, all of which can negatively impact the performance of the controller.

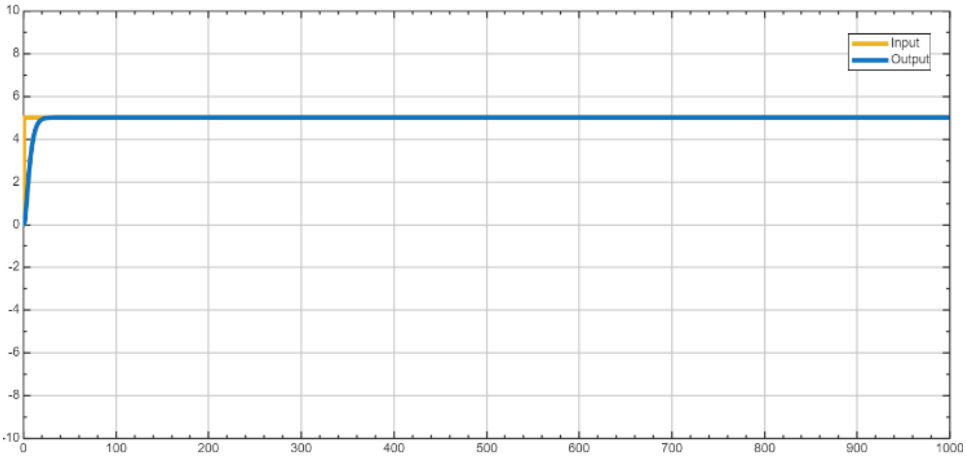


Fig. 5 System Output

Figure 5, shows how the ADRC controller behaves with respect to response to an input channel. The horizontal dashed line shows where the system was supposed to come to rest, while the solid line shows how the liquid level changes over time. There was a smooth transition to the reference value, without any significant oscillations or excessive overshoot of the output.

The fact that there was no overshoot, means that ADRC controlled the process optimally and compensated for all disturbances. Furthermore, the rise time was faster than expected given the natural inertia of the process. It was attributed to selecting a bandwidth for the controller that was much greater than the plant's natural frequency. After about 20 seconds, the system reached its setpoint and stayed there. This shows that the performance of ADRC is exceptionally good when maintaining its output.

3.5 MRAC Controller Design

Model Reference Adaptive Control (MRAC) is a sophisticated control strategy used to control systems (plants) whose parameters are unknown, uncertain, or vary significantly over time.

The core philosophy is simple and powerful:
The Reference Model (The Goal): You define a mathematical model that represents how you want your real system to behave. If you want a smooth, fast response with no overshoot, you build a reference model that acts that way.

The Actual Plant (The Reality): This is the physical system you are trying to control. Its true dynamic equations might be slightly different than estimated, or its parameters (like mass, friction, resistance) might change while it's running.

https://doi.org/10.71058/jodac.v10i03010

The Controller (The Bridge): The actual controller driving the plant has adjustable parameters (gains). The Adaptation Mechanism (The Brain): This mechanism continuously measures the "tracking error" the difference between the output of the reference model and the actual output of the plant

MRAC Model Reference Adaptive Controller

- Plant TF:

$$G(s) = 0.0008074 / (s^2 + 0.0441s + 0.0008222)$$

- Differential Equation -

$$\ddot{y}(t) + 0.0441\ddot{y}(t) + 0.0008222y(t) = 0.0008074u(t)$$

- Control law (Action) -

$$u(t) = \theta_1(t)\cdot r(t) - \theta_2(t)\cdot y(t) \qquad (18)$$

Where,
$r(t)$ = setpoint (10 cm)
$y(t)$ = current
θ_1: feedforward gain
θ_2: feedback gain

- Adaption law (MIT Rule) –

$$e(t) = y\ \text{model}\ (t) – y\ \text{plant}\ (t) \qquad (19)$$

- cost function (J):

$$J(\theta) = ½\, e^2 \qquad (20)$$

Reference Model Design : For Reference Model,

- $Grf(s) = 0.01 / (s^2 + 0.2s + 0.01)$
- $Gmodel(s) = \omega n^2 / (s^2 + 2\xi\omega ns + \omega n^2)$

So, Damping ratio ξ

- if $\xi < 1$ system vibrates / overshoot
- $\xi > 1$ system is sluggish
- $\xi = 0$ Fastest respons with minimal overshoot
- Here $\xi = 1$

(23)

Now,

- Assumed settling time= 40 sec

$$Ts \approx 4 / (\xi\omega n)$$
$$40 = 4 / (1 * \omega n)$$
$$\omega n = 4 / 40 = 0.1 \text{ rad/sc} \qquad (24)$$

Derivation and Calculation for Ideal Gains

The Gradient decent,

- A for theta 1 (θ_1) -

$$d\theta_1/dt = \gamma\ e(t)\ r(t) \qquad (25)$$

: Gain 1

- B for theta 2 (θ_2) –

$$d\theta_2/dt = - \gamma\ e(t)\ yp(t) \qquad (26)$$

: Gain 2 (negative)

Calculation for Ideal gains

- we plug control law $u = \theta_1 r - \theta_2 y$ into the plant eqn

Grouping y terms,

$$\ddot{y} + 0.0441\ddot{y} + (0.0008222 + 0.00080740_2)\, y = (0.00080740_1)\, r \tag{27}$$

Comparing,

- $\ddot{y}m + 0.2\ddot{y}m + 0.01ym = 0.01r$
- $0.0008074 * \theta_1 = 0.01$
- $\theta_1 = 0.01 / 0.0008074 = 12.39$

Final Calculations and Re-stated Reference Model

For theta 2,

$$0.0008222 + (0.0008074 - \theta_2) = 0.01$$
$$0.0008074 * \theta_2 = 0.01 - 0.0008222$$
$$\theta_2 = 0.00091778 / 0.0008074 = 1.1367 \approx 11.3 \tag{28}$$

Final Values,

Adaption Rate (γ) : 0.0001

Converged gain (θ_1) : 12.39

Converged gain (θ_2) : 11.37

Reference model (goal) -

$$Gr(s) = 0.01 / (s^2 + 0.2s + 0.01) \tag{29}$$

Differential equation:

$$\ddot{y}m(1) + 0.2\ddot{y}m(t) + 0.01ym(t) = 0.01v(t) \tag{30}$$

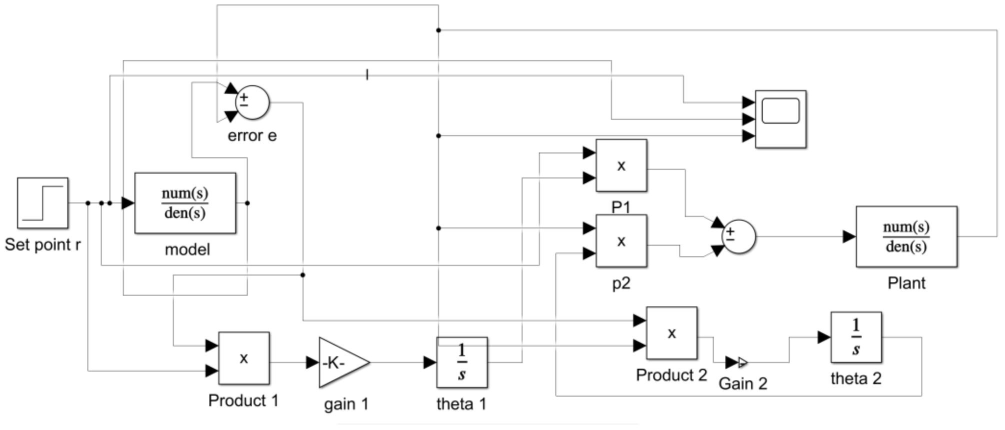


Fig.6 MRAC Controller Implemented in simulink

In fig.6,The Simulink model represents a closed-loop liquid level control system based on the Model Reference Adaptive Control (MRAC) strategy. In this structure, a reference model is defined to specify the desired dynamic behaviour of the system. The setpoint signal is simultaneously applied to both the reference model and the adaptive control loop. The output of the reference model represents the ideal response that the actual plant is expected to follow. The tracking error is generated by comparing the reference model output with the actual plant output. This error signal drives the adaptive mechanism, which continuously adjusts the controller parameters in

https://doi.org/10.71058/jodac.v10i03010

real time. The adaptive laws update the controller gains, denoted as adaptive parameters θ_1 and θ_2, through integrator blocks. These parameters are multiplied with corresponding regressor signals to generate the adaptive control input. Unlike ADRC, MRAC does not explicitly estimate disturbances; instead, it relies on continuous parameter adaptation to force the plant output to match the reference model. The plant is modelled using the experimentally identified transfer function, and its output is fed back to the adaptive controller, completing the closed-loop system . This structure highlights the key principle of MRAC: achieving desired performance by dynamically adjusting controller parameters so that the plant output asymptotically tracks the reference model output.

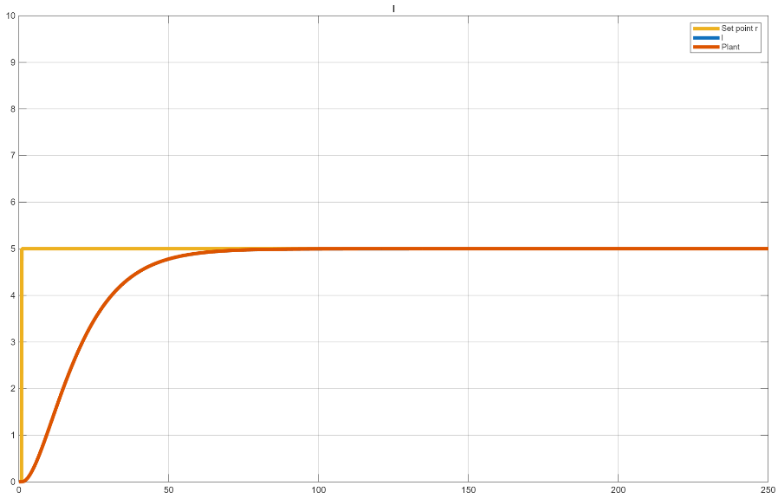


Fig.7 System Output

In fig.7, The response graph illustrates the time-domain performance of the liquid level system under MRAC control for a step reference input. The reference signal, represented by the ideal model output, exhibits a fast and smooth rise to the desired level. The actual plant response follows the reference model trajectory with a noticeable lag during the initial transient phase.

The fact that there was no overshoot shows that the adaptation gain was conservative which provides greater stability, but at the expense of less transient speed. Additionally, the minor steady-state error suggests that the adaptation of the controller parameters has been successful. However, a large settling time shows how sensitive MRAC is to plant dynamics and that there needs to be enough excitation for quick parameter convergence.

3.6 PID optimization methods

After comparing the advanced controller, the researcher began optimizing the PID controller by using four bio-inspired optimization algorithms. The four algorithms selected were PSO, SA, GA, and GWO, which are widely used and can be successful in solving control problems.

The PID controller gains (K_p, K_i, and K_d) were optimized for the plant model with each of the optimization algorithms, minimizing various performance indices: Integral Square Error (ISE), Integral Absolute Error (IAE), Integral Square Absolute Error (ISAE), and Integral Time Absolute Error (ITAE). Each of the algorithms and performance indices was evaluated separately to provide a thorough analysis of the optimization task performed on the PID controller.

https://doi.org/10.71058/jodac.v10i03010

1. Particle Swarm Optimization (PSO)

All particles should be given a random starting position and random velocity. After this, compute a fitness value for each particle. Each particle's best fitness value and the swarm's best overall fitness value will be updated for all particles.

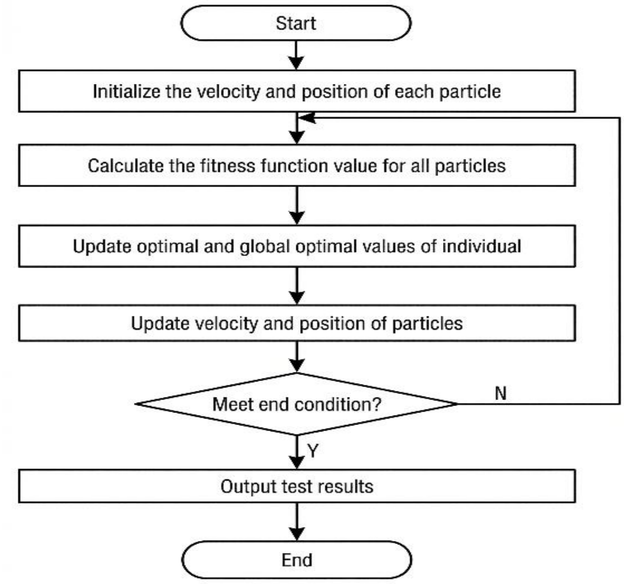


Fig.8 PSO Flow chart

The velocity and position of each particle will then be modified based on the best values obtained by each particle. These steps will be repeated until the stopping condition is met. The final answer will be presented after the stopping condition has been met.

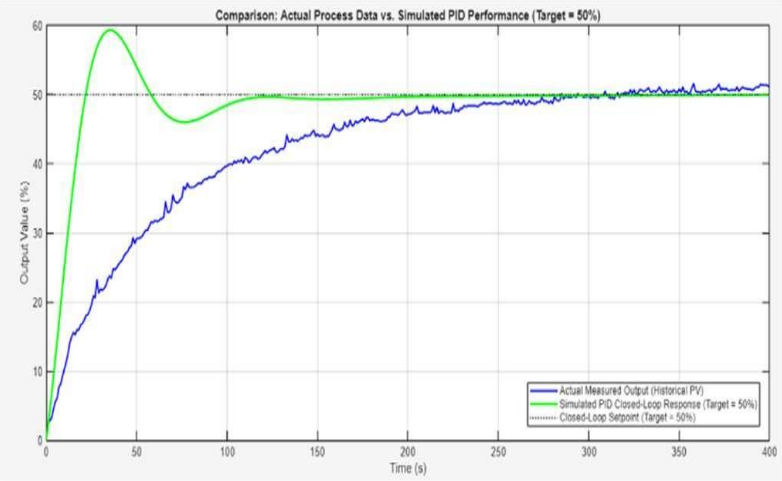


Fig.9 Performance of PSO in Simulation Vs Practical

Fig. 9, Simulink simulation of the PSO-based PID controller output versus system output comparison. The simulated system output closely tracked the desired setpoint and had a much faster rise time than did the actual system output, which had a lot more disturbances during the transient response. However, both systems achieved a similar steady-state value, thus validating the effectiveness of PSO-tuned PID control for use in real-world systems.

https://doi.org/10.71058/jodac.v10i03010

2. Simulated Annealing (SA)

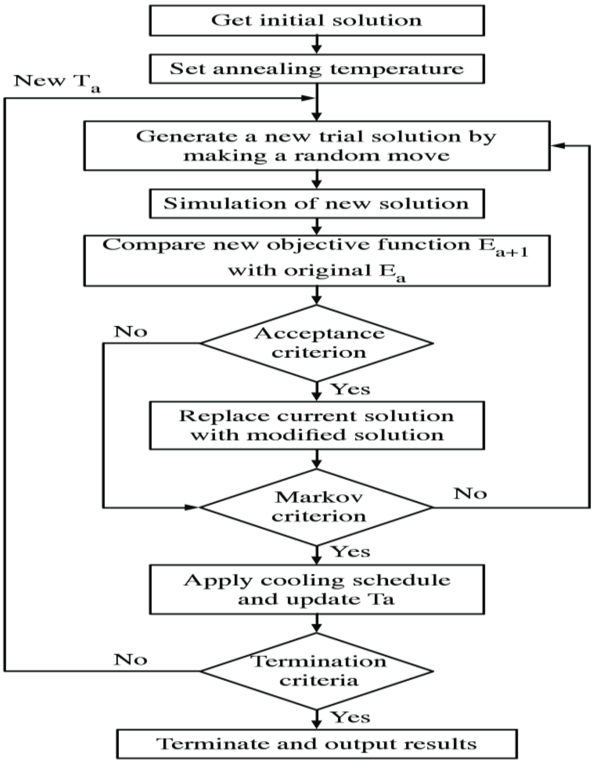


Fig.10 SA Flow Chart

The operative cycle of the simulated annealing algorithm is outlined in the flowchart provided. The first step of the operational cycle is the provision and evaluation of an initial solution, followed by an estimate of the initial temperature that determines how "random" the acceptance of solutions is. The next step of the cycle involves generating and evaluating another solution; if the generated solution is better than the current solution or meets the acceptance criteria, it is now the current solution. Once the initial solution has been updated to be the current solution, the temperature is gradually decreased (or adjusted), thereby reducing the likelihood of accepting poor solutions as the search continues. The process of generating a solution, evaluating the solution and decreasing the temperature continues until the stopping criteria are met, thereby leading to convergence upon an optimal solution.

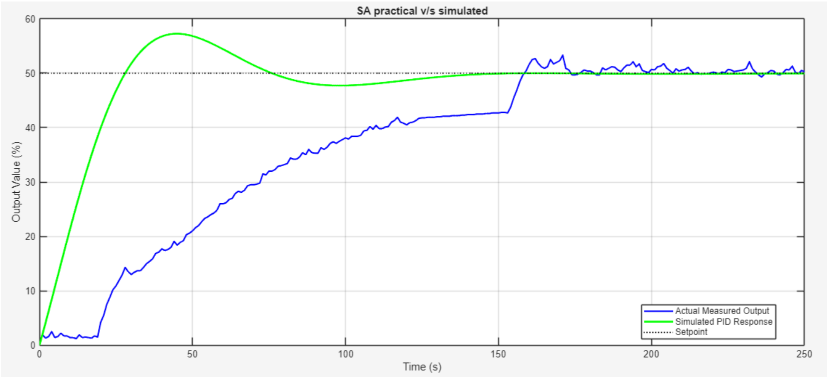


Fig.11 Performance of SA in Simulation Vs Practical

Fig. 11, presents a comparison between the Simulink-simulated and practical responses of the SA-based PID controller. The simulated response achieves the desired setpoint with faster rise time and smoother behavior, while

https://doi.org/10.71058/jodac.v10i03010

the practical response exhibits a slower transient and minor fluctuations due to real-time disturbances and system uncertainties. Nevertheless, both responses settle near the target value, confirming the applicability of SA-based PID tuning for practical control implementation.

3. Genetic Algorithm (GA)

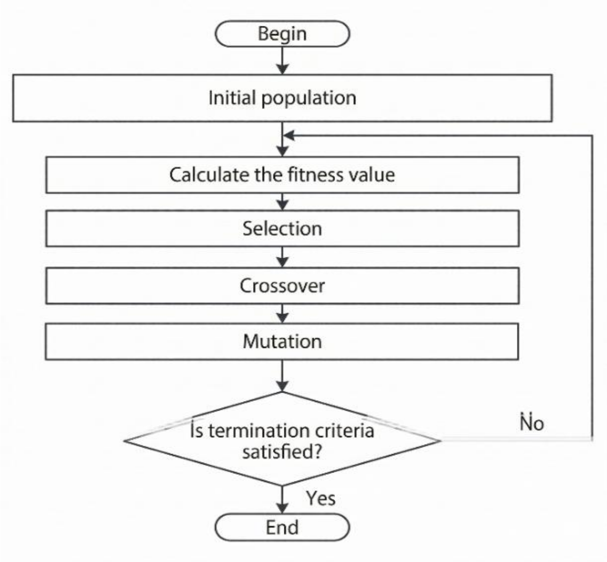


Fig.12 GA Flow Chart

This flowchart illustrates the Genetic Algorithm (GA) optimization process, inspired by natural evolution. It begins by generating an initial population of candidate solutions, followed by evaluating their fitness to determine how well they solve the given problem. Based on these fitness values, the algorithm performs selection, choosing the best individuals for reproduction. These selected individuals undergo crossover to exchange genetic information and mutation to introduce random variations, helping maintain diversity. The new population is then evaluated again, and this cycle of selection, crossover, and mutation continues until the termination criteria such as maximum generations or desired fitness is met, at which point the algorithm stops with an optimized solution.

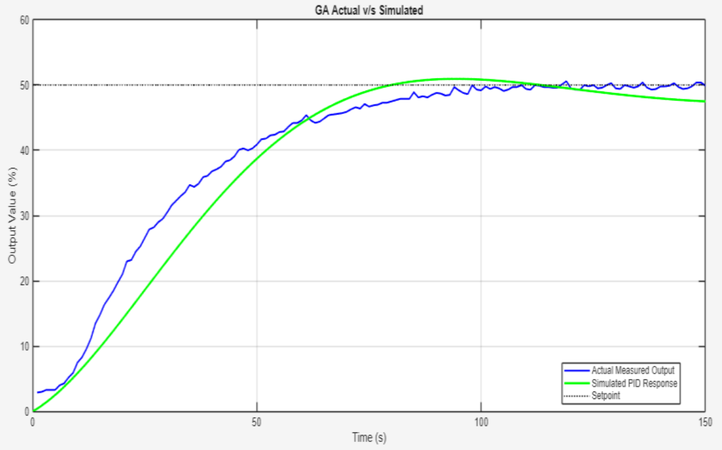


Fig.13 Performance of GA in Simulation Vs Practical

Fig. 13 compares the Simulink-simulated and practical responses of the GA-based PID controller. The simulated response shows a smooth and gradual approach to the setpoint, while the practical response follows a similar trend with minor deviations caused by real-time disturbances and modeling inaccuracies. Both responses converge to the desired value, demonstrating the stability and practical feasibility of GA-based PID tuning.

4. Grey Wolf Optimization (GWO) -

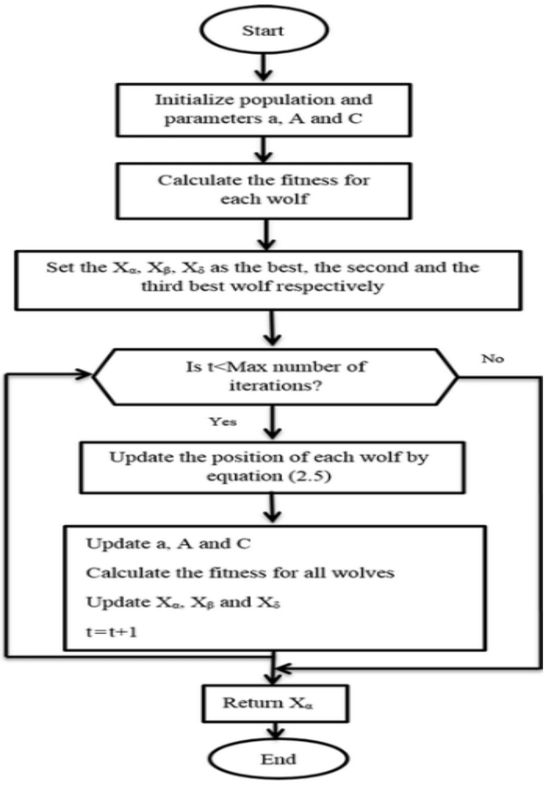


Fig.14 GWO Flow Chart

This flowchart shows the GWO algorithm, which is biologically inspired by the leadership hierarchy and hunting strategy of grey wolves. The process begins by setting initial parameters such as the number of wolves and iterations, followed by generating an initial population categorized into social ranks: α (best), β (second best), δ (third best), and ω (remaining wolves). The algorithm then mimics the wolves' hunting strategy by estimating the prey's position using the top three wolves (α, β, δ) and updating the positions of all wolves based on this estimation. Each iteration evaluates how close the wolves are to the prey (optimal solution) and updates their hierarchy accordingly.

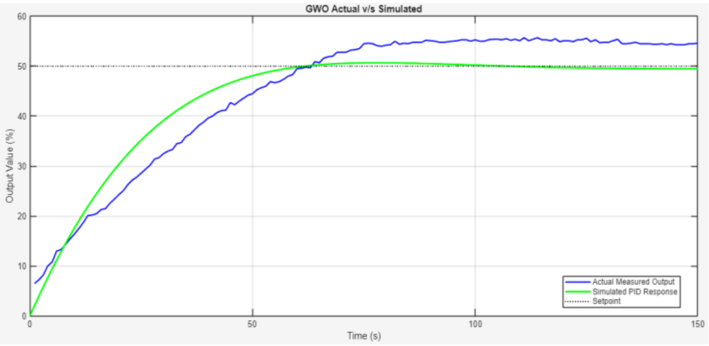


Fig.15 Performance of GWO in Simulation Vs Practical

As shown in figure 15, the comparison of the GWO-optimized PID simulator (green) with the actual process data (blue) at a 50% setpoint. The simulator has a smooth critically damped response, while the actual output is affected by characteristic hardware noise and exhibits a slight overshoot of approximately (5%) prior to settling down.

Although there are disturbances in the real world, the simulator closely follows the actual rise time which indicates that the GWO effectively simulates the dynamic characteristics of the system.

4. RESULT

Table 1 Time domain specifications

CONTROLLER	Td	Tr	Mp	Ts	Ess
ADRC	6.5 sec	11 sec	0	20 sec	0
MRAC	14 sec	35 sec	0	60 sec	0

Table 1, compares the time-domain performance specifications of the ADRC and MRAC controllers. The ADRC controller demonstrates superior transient performance with lower delay time, rise time, and settling time compared to the MRAC controller. Both controllers have zero maximum overshoot and zero steady-state error, which shows that the system response is stable and tracks the reference in steady state perfectly. However, the faster response dynamics of ADRC make it more suitable for applications that demand better dynamic responses.

Table 2 Error performance indices

CONTROLLER	IAE	ISE	ITAE	ITSE
ADRC	31.38	98.98	135	209.64
MRAC	75	187	1124.43	1406.23

The performance metrics presented in Table 2 further confirm the efficacy of the ADRC controller over MRAC. ADRC has much smaller values of IAE, ISE, ITAE, and ITSE compared to MRAC, which indicates that ADRC has smaller overall errors, better transient response, and better time-domain error performance. The large values of error indices for MRAC indicate poor response and persistent errors, whereas ADRC ensures better control accuracy and robustness.

Table 3 PID Optimum Values

PID Tuning Method	K_P	K_I	K_d
Particle Swarm Optimization	8.5081	0.1	50
Simulated Annealing	5.6649	0.1	50
Genetic Algorithm (GA)	1.5955	0.0263	10
Grey Wolf Optimization	2.9002	0.0532	50

Table 3 lists the optimal PID controller gains obtained using PSO, SA, GA, and GWO optimization algorithms. The optimal values of Kp, Ki, and Kd indicate the differences in the control gains due to various optimization algorithms, which directly affect the dynamic and steady-state performance of the closed-loop system.

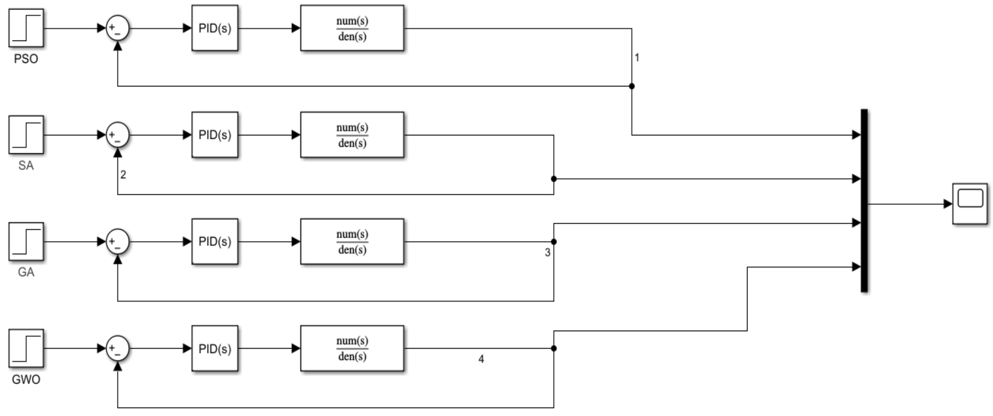


Fig.16 Simulink Model of PSO, SA, GA and GWO

Fig. 16 shows the Simulink model for PSO, SA, GA, and GWO-based PID controllers. Each optimization algorithm optimizes the PID controller parameters independently, and the corresponding controller outputs are fed to the same plant model. The outputs are combined and compared simultaneously to facilitate a direct comparison of the four optimization algorithms.

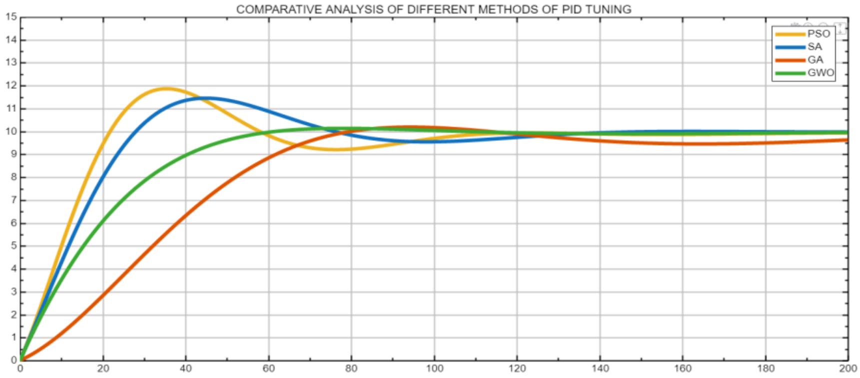


Fig.17 Combined Simulation of PSO, SA, GA and GWO

As shown in fig.17, the optimized PID controllers designed using PSO, SA, GA, and GWO were first simulated under the same test conditions. The performance parameters such as overshoot, settling time, and steady-state error were compared to identify the best-performing algorithm. Among all the optimization algorithms, the Genetic Algorithm produced the best PID parameters with superior dynamic performance.

Table 4 Time-domain performance comparison of PID tuning methods (simulation)

METHOD	**Tr**	**Tp**	**Mp**	**Ts**
PSO	16	35	20%	120
SA	12	45	15%	140
GA	50	95	2 %	240
GWO	35	80	1.8 %	65

The term shown in table 4, have following meanings, Tr- Rise time, Tp- Peak time, Mp- Peak overshoot and Ts-Setting time

Table 4 shows how well the different PID tuning methods have performed in time. The PSO and SA are faster than both the GA and GWO, with significantly less delay and rise time. The GA, however, is much slower than either of the other two at responding. The GWO has less overshoot and less settling time than the GA Each of the four tuning methods produced a zero steady-state error, which suggests that every tuning method produced good steady-state results.

Table 5 Time-domain performance

METHOD	Td	Tr	Tp	Mp	Ts
PSO	40	135	-	-	300
SA	60	135	170	6%	180
GA	25	50	-	-	100
GWO	25	48	100	11%	80

In Table 5, to verify the simulation results, the tuned parameters from the PID algorithms were applied to an actual physical liquid-level control system under the same operating conditions used in the simulation, and experimentation was conducted in real time. The experimental results were very similar to those obtained from the simulation, confirming the reliability of both the modeling and optimization method. Tuning of the PID controller using the Genetic Algorithm was determined to perform the best in both simulation and actual implementation, exhibiting the least amount of overshoot, faster settling times, and minimal steady-state error.

5. CONCLUSION

A comparative study has been undertaken to analyse and compare advanced control strategies and optimal PID tuning techniques for controlling a liquid level using a laboratory scale process control trainer. The system model was identified using system identification techniques, after which three types of controllers (Active Disturbance Rejection Control, Model Reference Adaptive Control, and optimally tuned PID controllers using PSO, SA, GA, and GWO) were evaluated using both simulated and experimental methods. ADRC proved faster than the previously established MRAC regarding transient response performance. ADRC produced a settling time of 20 seconds compared to MRAC's settling time of 60 seconds, which is a reduction in settling time of 66%. ADRC also had a shorter delay time (6.5 seconds) and a shorter rise time (11 seconds) than MRAC's delay time (14 seconds) and rise time (35 seconds). In addition, error performance indices were indicative of the superiority of ADRC with significantly lower error performance indices than MRAC (IAE = 31.38, ISE = 98.98, ITAE = 135, ITSE = 209.64, MRAC IAE = 75, ISE = 187, ITAE = 1124.43, ITSE = 1406.23).

The results for PID optimization techniques demonstrated that Particle Swarm Optimization (PSO) and Simulated Annealing (SA) had the quickest response times, with a settling time of 120 s for the PSO method and a settling time of 140 s for the SA method; however, the Grey Wolf Optimizer produced the least overshoot of approximately 1.8 %. Although the GA produced the largest overshoot, it had a good steady state performance. The experimental results also confirmed the same trends as well as performance. This implies that GWO performed the best overall and produced the fastest settling time of 80 s, whereas GA produced stable steady state behavior with a small steady state error. Although PSO and SA had faster rise times, the two methods of optimization produced larger amounts of overshoot and longer settling times when used in real time. Based on the comparison of the results, it may be stated that the ADRC method produced the most robust disturbance rejection ability with the other advanced controllers compared to the advanced disturbance rejection capability of the other advanced control methods. Furthermore, both GWO and GA produced the best ratio among the PID optimization methods. These results suggest that the integration of advanced control methodologies with meta-heuristic optimization methodologies can produce significant improvements in liquid level control performance in process manufacturing applications.

https://doi.org/10.71058/jodac.v10i03010

5. REFERENCE

[1] K. Łakomy, W. Giernacki, J. Michalski, and R. Madonski, "Active Disturbance Rejection Control (ADRC) Toolbox for MATLAB/Simulink," arXiv preprint arXiv:2112.01614v2, 2022.

[2] G. Herbst, "A Simulative Study on Active Disturbance Rejection Control (ADRC) as a Control Tool for Practitioners," Electronics, vol. 2, no. 3, 2013.

[3] V. Q. Vinh, T. C. Phat, and N. H. Giang, "An Adaptive Active Disturbance Rejection Controller (ADRC) for Induction Motor Drives," SSRG International Journal of Electrical and Electronics Engineering, vol. 11, no. 12, 2024.

[4] Y. Wen-bin and Y. Dong, "Modelling and Simulation of an Active Disturbance Rejection Controller Based on Matlab/Simulink," International Journal of Research in Engineering and Science (IJRES), vol. 3, no. 7, 2015.

[5] M. C. Rais et al., "Comparative Study of Optimization Techniques Based PID Tuning for Automatic Voltage Regulator System," Eng. Proc., vol. 14, 2022.

[6] M. M. Massoud and J. Libby, "Comparative Analysis of Evolutionary Algorithms for PID Controller Optimization in Pneumatic Soft Robotic Systems," IEEE Access, vol. 12, 2024.

[7] H. Hesham, M. Ezzat, and R. A. Swief, "Comparative study of PID controller designs for AVR using different optimization techniques," Archives of Electrical Engineering, vol. 70, no. 3, 2021.

[8] M. Z. Romdlony et al., "Design and application of models reference adaptive control (MRAC) on ball and beam," Journal of Mechatronics, Electrical Power, and Vehicular Technology, vol. 13, 2022.

[9] S. Mallick, "Design of Lyapunov based Model Reference Adaptive Controller for First Order Plus Delay Time Liquid Level System," 2019 International Conference on Intelligent Computing and Remote Sensing (ICICRS), 2019.

[10] H. Rezaei, O. Bozorg-Haddad, and X. Chu, "Grey Wolf Optimization (GWO) Algorithm," in Advanced Optimization by Nature-Inspired Algorithms, 2018.

[11] G. Qi et al., "Research on Group Control Technology of Oil Well Dynamic Liquid Level Based on Distributed Model Predictive Control," 44th Chinese Control Conference, 2025.

[12] S. E. Oltean, M. Dulau, and A. V. Duka, "Model Reference Adaptive Control Design for Slow Processes. A Case Study on Level Process Control," Procedia Technology, vol. 22, 2016.

[13] M. H. Hassani, M. Pourgholi, and A. Yazdizadeh, "Real-Time Control of Tank Water Level Using Gradient-Free Lyapunov-Based Neural-PID Controller," 2024 10th International Conference on Control, Instrumentation and Automation (ICCIA), 2024.

[14] S.K.Pandey, “Comparative Study of Modern Optimization Techniques for PID Controller Design,” *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 07, no. 04, Apr. 2023.

[15] R.Ranjan,“Robust Model Reference Adaptive Control for Liquid Level Control in Process Industry,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 4, no. 5, May 2021.

[16] A. Kadu and A. Khandekar, “Optimized decentralized PID controller for TITO systems using grey wolf optimization with experimental application,” *Engineering Research Express*, vol. 7, no. 4, p. 045416, 2025.

[17] A. Kadu, S. Bhosale, A. Pagare, P. Sable, R. Kudtarkar and A. Bhilare, “Development of Low-Cost Water Level Monitoring and Control using PID Controller,” in *Proc. 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Bengaluru, India, Dec. 2023, pp. 524–528