processes

MDPI

Article

Gain Scheduling of a Robust Setpoint Tracking Disturbance Rejection and Aggressiveness Controller for a Nonlinear Process

Veeramani Bagyaveereswaran [1] and Pachiyappan Arulmozhivarman [2,*]

[1] School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamilnadu 632014, India
[2] School of Electrical Engineering, Vellore Institute of Technology, Vellore, Tamilnadu 632014, India
* Correspondence: parulmozhivarman@vit.ac.in

Received: 3 April 2019; Accepted: 25 June 2019; Published: 2 July 2019

Abstract: In this paper, a robust setpoint tracking disturbance rejection and aggressiveness (RTD-A) controller is designed and developed to control the liquid level of a conical tank process. Meta-heuristic algorithms like grey wolf optimization and the genetic algorithm are used to tune the parameters of the RTD-A controller. Its performance is later compared with that of the conventional standard proportional integral derivative controller. The gain scheduled RTD-A controller is designed and implemented on a nonlinear conical tank process. Also, various performances attributes such as the integral square error, integral absolute error, integral time absolute error, rise time, and settling time are calculated for the first-order process and conical tank process. The servo responses with RTD-A are also compared against the responses recorded from the conventional control schemes.

Keywords: controller; gain scheduling; optimization; PID; performance criteria; servo; regulatory

1. Introduction

In almost every industrial application, control of process variables is important for the efficient and safe working of the process. The most recurrent variables which need to be controlled in industries include temperature, level, pressure, and pH [1]. A typical control system, or rather a good controller, is expected to make the process efficient, suppress the impact of external disturbances on the process, make the process stable, and optimize the entire process. In industries, most of the processes are nonlinear [2]. To cope with such heavy nonlinearity, the controller is expected to exhibit robustness. Over the last four decades, the proportional integral derivative (PID) controller has held a significant position in process control. This is because it is inexpensive and easy to tune [3]. Another advantage of the PID is its fixed structure [4]. The classical methods like those of Ziegler–Nichols and Cohen–Coon focus on manual tuning of the PID controller parameters. Also, there are different PID tuning methods in the frequency domain for open loop stable system without transportation lag or dead time [5]. Dead time is defined as the time lag in process variable response when the controller output signal is applied. The presence of dead time is undesirable in the control loop. In practical implementation, manual tuning becomes arduous [6]. In spite of the simple structure, optimal tuning of PID gain parameters is relatively difficult [7–9].

To avoid this, various meta-heuristic algorithms are available which make the optimal tuning of PID gains quite easier [10]. This means we can achieve the desired performance specifications. There are bio-inspired optimization algorithms which can be used to tune the PID controller parameters [11]. In spite of being tuned using such fine algorithms, the PID controller fails considerably in certain aspects. This is because the PID controller does not give satisfactory results with systems which need setpoint tracking and disturbance rejection simultaneously [12]. Moreover, it is linear and its performance with

nonlinear systems deteriorates. It is mentioned that though the PID controller is widely used because of its simplicity and it cannot meet the requirements needed for better performances unless the controller is tuned properly [13]. Since PID controllers are fixed gain controllers, they cannot compensate the parameter variation in the plant and cannot adapt to changes in the environment [14]. Processes with large dead time, inverse response, and nonlinear processes are difficult to control efficiently by the PID [15]. In PID, there is no direct relationship between the three tuning parameters and setpoint tracking, disturbance rejection ability, and robustness. Therefore, achieving each performance attributes by individual tuning of PID is not straightforward. To overcome this problem, the model predictive controller (MPC) is widely used in industries which employ nonlinear process in particular. It is often implemented in supervisory mode hierarchically above the base PID controller. Therefore, its performance again primarily depends on the PID in the lower level. The PID has to implement the commands received from MPC [15,16]. MPC is also implemented as a direct control algorithm, mainly where classical PID structures cannot deliver required control performance, due to difficult dynamics, strong interactions, and active constraints [17]. The MPC utilises a distinct model of the process to predict the future response of the process. The disturbance rejection offered by MPC is much better when compared to that of PID. However, the MPC suffers from serious drawbacks in that it is complicated and its parameters are tough to calculate [18].

An alternative control strategy to overcome the shortcomings of the existing controllers was proposed by Kapil Mukati which intended on combining the positive attributes of the PID controller and MPC [12]. The controller which was proposed was named as RTD-A because a good controller should possess all of the following characteristics, namely: robustness (R), set-point tracking (T), disturbance rejection (D), and overall aggressiveness (A). All the four tuning parameters are normalized to lie within the range of 0 to 1 [15,19]. This four factor control technique which was developed aimed at exploiting the modern electronic hardware parts. In addition, focussed at imbibing the best of the qualities of a good controller. Another attempt was made in a paper to analyse and compare the potential of MPC, PID, RTD-A, and cascade controllers using isoflurane with the bispectral index as the controlled variable [20–22]. In other work, a fuzzy scheduled RTD-A controller was designed and tested for different nonlinear processes. Some of those were control of pH process and Type I diabetic process [23]. Later, the RTD-A controller was designed for controlling the industrial process using the second order process with dead time (SOPDT) model. Also, the performance of RTD-A was compared with internal model control (IMC), MPC, and PID. The PID can be tuned using a particle swarm optimization algorithm for DC motor speed control and also it is tuned by fuzzy logic. The overall closed loop performance can be improved by quick optimal tuning of controller [24].

In this paper, the RTD-A controller parameters θ_R, θ_T, θ_D, and θ_A are found out by different meta-heuristic algorithms. In order to find out the best optimization technique available, initially, a comparison study was made among eight different optimization techniques namely the genetic algorithm (GA), simulated annealing (SA), the gravitational search algorithm (GSA) [25], ant colony optimization (ACO), bacterial foraging (BF), cuckoo search (CS), harmonic search (HS), and grey wolf optimization (GWO) [26]. These algorithms were used to tune the gains of the PID controller and its performance was analysed for a linear process. Here, grey wolf optimization (GWO) and the genetic algorithm (GA) were used individually to find the RTD-A controller tuning parameters. These two algorithms were chosen after a comparative analysis was made between eight different heuristic optimization techniques. The complete study is presented in Section 4.1.1 and experimental results given in Section 5.1. This tuned controller was later used to control the level of the conical tank and first-order model with dead time (FOPDT) process. The gain scheduling was done because it is needed in industrial processes whose dynamics keep changing with variables [27].

The contribution of this paper, in particular, is to demonstrate that RTD-A controller parameters tuned using grey wolf optimization (GWO) exhibit peerless performance in controlling nonlinear processes. This means that the tuning of the parameters will become easier and a robust controller is developed which can handle any disturbances and setpoint changes [15].

The paper is organized in the following way: Section 2 describes the RTD-A controller basics, Section 3 briefs about the optimization techniques used, the process considered and its description is given in Section 4 and the results and conclusion are mentioned in Sections 5 and 6, respectively.

2. RTD-A Controller Basics

The RTD-A control strategy comprises of the following three major components [15].

- Prediction of process output
- Updating of model predictions
- Calculation of control action.

According to the actual RTD-A scheme, any process be it linear or nonlinear, firstly has to be represented by a simple first-order with time delay process (FOPDT). The non-linear process can be approximated into linear model using Taylor's series expansion [28].

This FOPDT model can be generalized as:

$$G(s) = \frac{Ke^{-\alpha s}}{\tau s + 1} \tag{1}$$

where K is process gain, α is dead time or transportation lag, τ is time constant, $u(s)$ is process input, and $y(s)$ is process output. The Discrete model for Equation (1) is given in Equation (2).

$$\overline{y}(k+1) = a\overline{y}(k) + bu(k-m), k = 0, 1, 2 \ldots \tag{2}$$

where, $a = e^{-\Delta t/\tau}$, Δt is the sampling time, $b = K(1 - e^{-\Delta t/\tau})$ and $m = \text{round}(\alpha/\Delta t)$

From Equation (2), the output prediction for next m steps can be written as in Equation (3).

$$\overline{y}(k+m+1) = a^{(m+1)}\overline{y}(k) + b\mu(k,m) + bu(k) \tag{3}$$

where,

$$\mu(k,m) = \sum_{i=1}^{n} a^i u(k-i)$$

μ is the weighted sum of past control action taken during m period and $u(k)$ is the current control action. There will be only one control $u(k)$ allowed from the current time instant and is given in Equation (4).

$$u(k+i) = u(k), i = 1, 2, \ldots N \tag{4}$$

Equation (5) is obtained by considering the predicted process output over next N-steps.

$$\overline{y}(k+m+i) = a^{m+i}\overline{y}(k) + a^{i-1}b\mu(k,m) + b\eta_i u(k); 1 \le i \le N \tag{5}$$

where

$$\eta_i = \frac{1-a^i}{1-a} \tag{6}$$

The idyllic model prediction requires model error $e(k) = y(k) - \overline{y}(k)$ that explicitly decomposed into two parts such as effect of model mismatch $e_m(k)$ and unmeasured disturbance $e_D(k)$ in Equation (8). Therefore, the total error is shown in Equation (7),

$$(k) = e_m(k) + e_D(k) \tag{7}$$

where

$$e_D(k) = \theta_R e_D(k-1) + (1-\theta_R)e(k) \tag{8}$$

where θ_R lies between 0 and 1. The controller has the robustness ability to handle the plant uncertainties. The future disturbance effect is predicted according to Equation (9).

$$\bar{e}_D(k+j) = e_D(k) + \frac{\{1-\theta_D\}}{\{\theta_D\}}\left[1-(1-\theta_D)^j\right]\nabla e_D(k) \tag{9}$$

for $m+1 \le j \le m+N$, Where $\nabla e_D(k) = e_D(k) - e_D(k-1)$

Here θ_D is the disturbance rejection controller tuning parameter. Finally, by solving the Equation (5) and Equation (9) the updated N-step model output prediction is given by Equation (10).

$$\widetilde{y}(k+m+i) = \overline{y}(k+m+i) + e_D(k+m+i) \tag{10}$$

A plant with the setpoint trajectory y^* has the control action to minimize the error. At a discrete time interval, the control action will be updated. If the desired final output is considered as $y_d(k)$, then the desired trajectory to follow $y^*(k)$ is given by Equation (11).

$$y^*(k+j) = \theta_T^j y^*(k) + \left(1-\theta_T^j\right)y_d(k); 1 \le j \le \infty \tag{11}$$

The setpoint tracking tuning parameter θ_T is introduced which varies from 0 to 1. The value of N will be calculated by the controller aggressiveness tuning parameter θ_A given in Equation (12).

$$\theta_A = 1 - e^{\frac{(N-1)\delta t}{\tau}} \tag{12}$$

The RTD-A controller output $u(k)$ is given in Equation (13). The calculated controller output is applied to the plant.

Where $\psi_i(k) = y^*(k+i) - a^{m+i}\hat{y}(k) - a^{i-1}b\mu(k,m) - \hat{e}_D(k+m+i|k)$.

$$u(k) = \frac{\sum_{i=1}^{i=N}\eta_i\psi_i(k)}{b\sum_{i=1}^{i=N}\eta_i^2} \tag{13}$$

The RTD-A parameters are tuned using optimization techniques like genetic algorithm (GA), simulated annealing (SA) and grey wolf optimization (GWO).

The parameters are tuned in such a way that the performance index integral absolute error (IAE) is minimized. The error is the difference between the actual process output and the desired setpoint to be reached. This is obtained by considering the first-order linear model.

The block diagram representation of the basic RTD-A controller is shown in Figure 1.

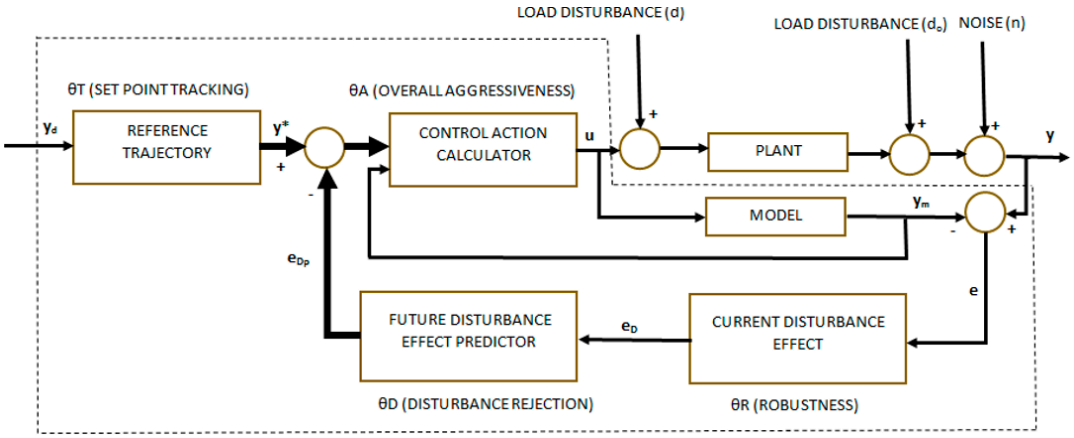


Figure 1. Block diagram of the RTD-A controller.

For the sake of simplicity, the main steps involved in the RTD-A control scheme is listed. They are:

(1) The depiction of the process model in terms of first-order dynamics.
(2) Determination of control system specifications. For example, firstly the sampling time, then the discretized form of model parameters, later, the computation of the tuning parameter for each controller performance attributes.
(3) Lastly, the calculation of control vector.

The entire block diagram can be explained in simple words in the following way. It consists of a controller and a plant. The setpoint y_d that has to be tracked or rather achieved. As the RTD-A controller is composed of a predictor, a model of the plant is present in the controller. The difference between the model's output and the plant's output is used to determine the robustness parameter. The plant output is measured using necessary sensors for monitoring which also provides feedback to the controller. The four basic blocks that are used to calculate the controller action $u(k)$ are reference trajectory, control action calculator, current disturbance effect and future disturbance effect predictor. The plant may have low disturbance in the input side or output side and it may contain noise. The reference trajectory and the control action calculator block are in the forward loop. The current disturbance and future disturbance effect predictor form the feedback loop [29].

3. Optimization Techniques

In order to find out the best optimization technique available, initially, a comparison study was made among eight different optimization techniques namely the GA, SA, gravitational search algorithm (GSA) [25], ant colony optimization (ACO), bacterial foraging (BF), cuckoo search (CS), harmonic search (HS), and GWO [26]. These algorithms were used to tune the gains of the PID controller and its performance was analysed for a linear process. The study showed that GWO and GA outweigh other optimization techniques in terms of better tuning. Therefore, the outline of only two algorithms is given below:

3.1. Grey Wolf Optimization

The algorithm is typically based on the leadership qualities sown by wolves and their hunting style. The GWO flowchart is shown in Figure 2. Here, the entire population is sub-divided into four main grey wolf categories, namely, the leader wolf (α), subordinate of α (β), best wolf of α (δ), and scapegoat wolf (Ω). Among these, the first three wolves guide the other wolf for hunting. The entire procedure is summed up as [30]:

(1) Firstly, randomly decide the population of the wolves.
(2) Initialize the vector's a, $\vec{A} = 2\vec{a} * \vec{r}_1 - \vec{a}$ and $\vec{C} = 2\vec{r}_2$. Where $r1$ and $r2$ are two random vectors between [0,1] and the vector component a is linearly decreasing from 2 to 0 over each course of the iteration.
(3) For each wolf, the fitness value IAE needs to be calculated.
(4) Among the four, the best three wolves are chosen and are assigned as α, β and δ and they are declared as the leaders.
(5) The remaining population's location (X_α, X_β and X_δ) is updated if maximum number of iteration is not reached. Else display the best solution as an optimized result.
(6) Update the vector's a, A and C.
(7) The entire procedure from step (3) is continued.
(8) The location of α is returned as the most optimum value.

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

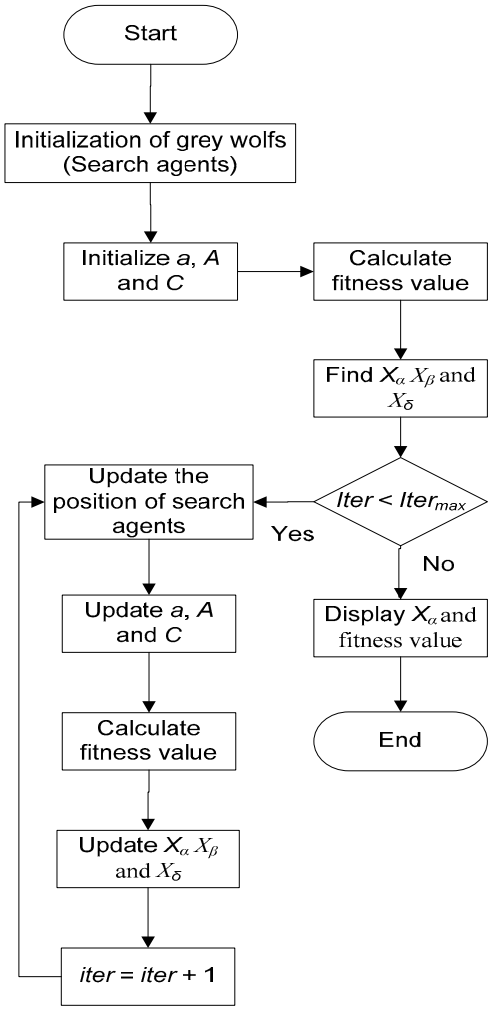


Figure 2. Flow chart of the grey wolf optimization (GWO) algorithm [30].

3.2. Genetic Algorithm

GA is a meta-heuristic algorithm that is inspired by the evolution of the natural selection of genes. It replicates Darwin's theory of evolution stating survival of the fittest, over successive generations. This algorithm constantly revises the population of the obtained solution [31]. The flow chart explaining the GA optimization algorithm is shown in Figure 3. The fundamental procedure, which is followed for GA optimization is [32]: The initial values of the population are defined by the user. The step in which individuals are chosen is crucial. This is because we have to select the beings that are fit and can generate the population for the upcoming generation. Cross over is a method of imbibing the best qualities from the two parents for the next generation. The default values are chosen for the GA parameters except for swarm size and the maximum number of iteration. These are listed in Table 1.

(1) The cost function or rather the performance index (PI) which is the IAE is tested. If it suffices, terminate the algorithm; else go to the next step.
(2) Choosing of the individual.
(3) Breeding of the beings.
(4) Cross over
(5) Mutation
(6) Checking the performance index. If it is not passable, repeat from the third step.

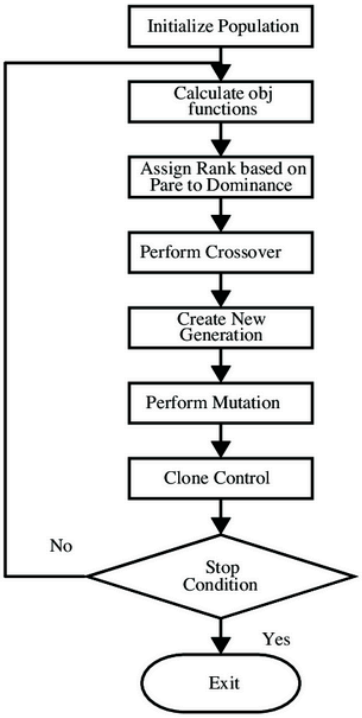


Figure 3. Flow chart of the genetic algorithm (GA) [33].

Table 1. Iteration and swarm size used in the optimization algorithm.

Parameter	GA	GWO
Maximum Iteration	30	70
Swarm Size	50	50

4. Process and Performance Measure

4.1. Process Considered

To appreciate the remarkable performance of the novel controller RTD-A three processes such as Direct current (DC) motor position control, typical first-order process with dead time (FOPDT) model and conical tank are considered, its performance on a FOPTD model is primarily observed.

4.1.1. DC Motor Position Control

The equivalent circuit representation of a DC shunt motor is shown in Figure 4. It is used to derive the transfer function [34]. The parameters listed in Table 2 is used to determine the final transfer function. The PI controller tuning parameters are found using different optimization techniques and are listed in Table 3.

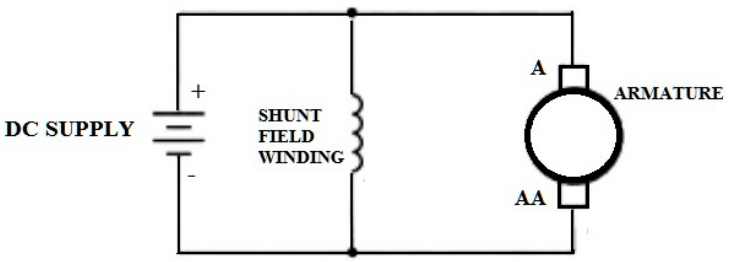


Figure 4. Representation of DC shunt motor.

Table 2. DC motor specifications.

Parameter	Value
J (Moment of inertia)	0.022 kg–m^{2}rad^{-1}
B (Damping coefficient)	0.5*10^{-3} N–m/(rad/s)
K_t (Torque constant)	0.06 NmA^{-1}
K_b (Electromotive force constant)	1.0 V-srad^{-1}
R (Electrical resistance)	2.45 ohms
L (Electrical inductance)	0.035 Henry

Table 3. PI tuning parameters using different optimization for the DC motor.

Parameters	**PI-GA**	**PI-CS**	**PI-SA**	**PI-ACO**
$\mathbf{K_P}$	17.421	25.6291	17.258	20.8567
$\mathbf{K_I}$	0.0001	3.9189	75.515	1.0051
Parameters	**PI-HS**	**PI-BF**	**PI-GSA**	**PI-GWO**
$\mathbf{K_P}$	17.4072	23.2606	17.4020	17.2121
$\mathbf{K_I}$	−1.3868	−0.4204	−1.3872	−1.3635

The transfer function for position control is derived from the equation obtained by solving the behaviour of a dynamic dc motor.

The voltage equation of the shunt motor is given by Equation (14):

$$V = R_i + L\frac{di}{dt} + e_b \tag{14}$$

where, e_b is the back emf given by the machine. The torque is proportional to the armature current which is depicted by Equation (15):

$$T_m = K_t * i_a(t) \tag{15}$$

The torque in mechanical terms is given by Equation (16):

$$T_m = J\frac{d^2\theta(t)}{dt^2} + B\frac{d\theta(t)}{dt} \tag{16}$$

Solving these equations, we obtain the transfer function as Equation (17):

$$\frac{\theta(s)}{v(s)} = \frac{K_b}{(R_sJ + BL_a)s^2 + \left(K_b^2 + R_aB\right)s} \tag{17}$$

Substituting the parameters (mentioned in the Table 2) we get Equation (18):

$$\frac{\theta(s)}{v(s)} = \frac{1}{0.00077s^3 + 0.059s^2 + 1.441s} \tag{18}$$

4.1.2. Typical FOPDT Model

The process model which was considered for the same is given by Equation (19).

$$G(s) = \frac{e^{-s}}{2s+1} \tag{19}$$

This first-order process was controlled using RTD-A that was tuned using the genetic algorithm and the GWO algorithm. After tuning, the parameters which were obtained are given in Table 4.

Table 4. RTD-A parameters obtained using the GA and GWO.

Tuning Parameters	RTD-A-GA	RTD-A-GWO
θ_A	0.3494	0.3573
θ_R	0.4380	0.2643
θ_T	0.00026	0.2676
θ_D	0.2226	0.2400

4.1.3. Conical Tank

Next in order to realise RTD-A's potential we tested the controller on the level control of a non-linear conical tank process. The RTD-A tuning parameter was optimized using GA and GWO and was used to control the level of the conical tank. The outline for the RTD-A control implementation can be summed up in the following way:

(1) The mathematical model is derived from the system.
(2) The system is linearized at various desired or operating points.
(3) The linearized model is used to obtain optimized RTD-A parameters.
(4) The obtained parameters are used to control the nonlinear process.

In process industries, control of liquid level in a tank is essential. These conical tanks find important usage in industries like chemical, food processing industries, cement industries, etc., [35]. The conical tank model considered for the study is shown in Figure 5.

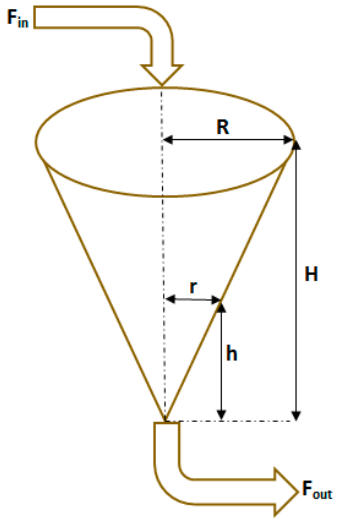


Figure 5. Conical tank model [35].

The parameters considered are given in Table 5. If the process is to be controlled using RTD-A, the nonlinear process should be linearized at the operating point and it can be represented in the form of first-order transfer function. Later it can be applied directly to control the nonlinear process once the parameters are found. The rate of change of level in a conical tank is given by Equation (20):

$$\frac{dh}{dt} = \frac{F_{in} - b\sqrt{h}}{\frac{\pi R^2 h^2}{H^2}} \tag{20}$$

Table 5. Conical tank parameters [35].

Parameters	Value
MAX height (H)	100 cm
MAX radius (R)	20 cm
Valve constant (b)	4.3
MAX Inflow (F_{in} max)	100 cm^3/s

The nonlinear governing equation considered in Equation (3) is linearized using Taylor series linearization. The linearized model is given in Equation (21).

$$\frac{h(s)}{F_{in}(s)} = \frac{K}{\tau s + 1} \tag{21}$$

where

Process gain $K = \frac{2\sqrt{h}}{b}$

Process time constant $\tau = \frac{2A\sqrt{h}}{b}$

Cross sectional area $A = \frac{\pi h^2 R^2}{H^2}$

where h is the height at which the plant is linearized (operating point).

To efficiently compare the performance of the algorithms several integral performance criteria such as the integral square error (ISE), integral absolute error (IAE), and integral time absolute error (ITAE) can be calculated. Here, IAE is considered for performance analysis.

4.2. Performance Measures

4.2.1. ISE

The ISE is a commonly used performance index. It basically quantifies the performance of a system which is formed or generated by integrating the square of the error generated over a fixed period of time and it is given by Equation (22). This type of performance index is used mostly in linear optimal control theory as well as in estimation theory [36].

$$ISE = \int_0^t e^2(t)dt \tag{22}$$

4.2.2. IAE

The IAE is basically an error considered and taken in absolute form and added over a fixed time and it is given by Equation (23). This measures the system's behaviour. Also, it is a program which does the fitting job of a sample to test how close it is to the target expression [36].

$$IAE = \int_0^t \left|e(t)\right|dt \tag{23}$$

4.2.3. ITAE

The ITAE is the most commonly used performance index when error persists for a longer time and it is given by Equation (24). It integrates the absolute value of the error generated and time weighted [36].

$$ITAE = \int_0^t t\left|e(t)\right|dt \tag{24}$$

4.3. Gain Scheduling

It is the most customary approach used in control systems whose dynamics keep changing with variables. It is the most common way of controlling nonlinear systems that especially employs linear controllers [35]. The schematic block diagram of gain scheduling of controller is shown in Figure 6. The controller is designed for different tuning values with respect to the number of operating points considered for non-linear process. It is considered as the array of controllers. The particular tuning parameters will be assigned to the controller from the look-up table when the process is operating closer to the definite operating point. A single controller will not be able to give satisfactory performance for the entire process operating range. Therefore, the controller has to switch between the tuning parameters to cover the entire output range [37]. Here, we design the parameter values of the controller for different gains and it is discussed in Section 5.

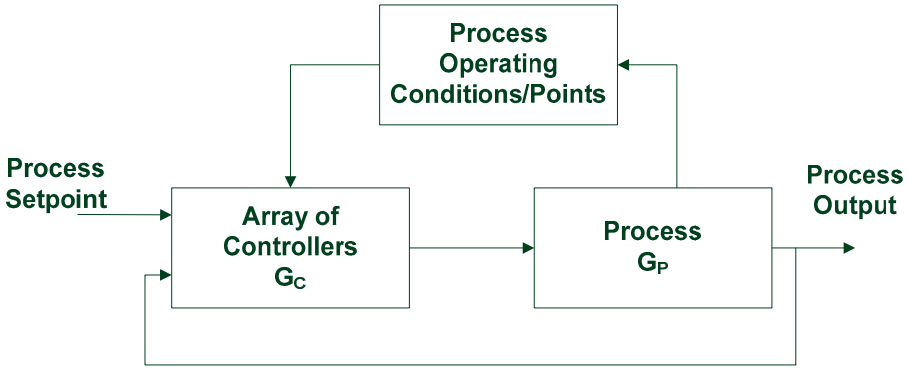


Figure 6. Schematic diagram of gain scheduling of the controller [38].

5. Experimental Results

5.1. Case 1: DC Motor Position Control

The setpoint of 5 radians is considered and the PI controller is complemented using the values listed in Table 3. The DC motor servo response is plotted in Figure 7. The performance is compared in terms of time domain specifications and integral performance measures. It is listed in Table 6. Analysing the parameters in Table 6 the variations in performance criteria and time domain parameters in various algorithms is observed. It has been found that controller optimized with GWO and GA shows a better result than other optimization algorithms. Therefore, GWO and GA are used to experiment with the remaining two cases in this section.

Table 6. Performance comparison of DC motor position control. IAE: integral absolute error; ISE: integral square error; ITAE: integral time absolute error.

	PI-GA	PI-CS	PI-SA	PI-ASO	PI-HS	PI-BF	PI-GSA	PI-GWO
Rise time	0.1026	0.0707	0.0796	0.0848	0.1021	0.0769	0.1022	0.1013
Settling time	0.2597	0.3216	0.5710	0.2544	0.2605	0.2418	0.2604	0.1656
Overshoot	3.1794	15.5633	34.3423	8.1315	3.2461	11.5439	3.2394	1.5825
Peak	3.0954	3.4930	4.0311	3.2543	3.0756	3.3425	3.0754	3.0260
Peak time	0.2162	0.1562	0.2162	0.1762	0.2162	0.1762	0.2162	0.2162
IAE	102.080	120.054	113.044	109.956	101.914	114.599	101.901	39.5445
ISE	223.262	237.6165	175.8096	232.7302	223.9423	235.4804	223.9157	105.9652
ITAE	5.7165	7.9013	18.5577	6.3432	5.6328	7.0051	5.6325	1.2492

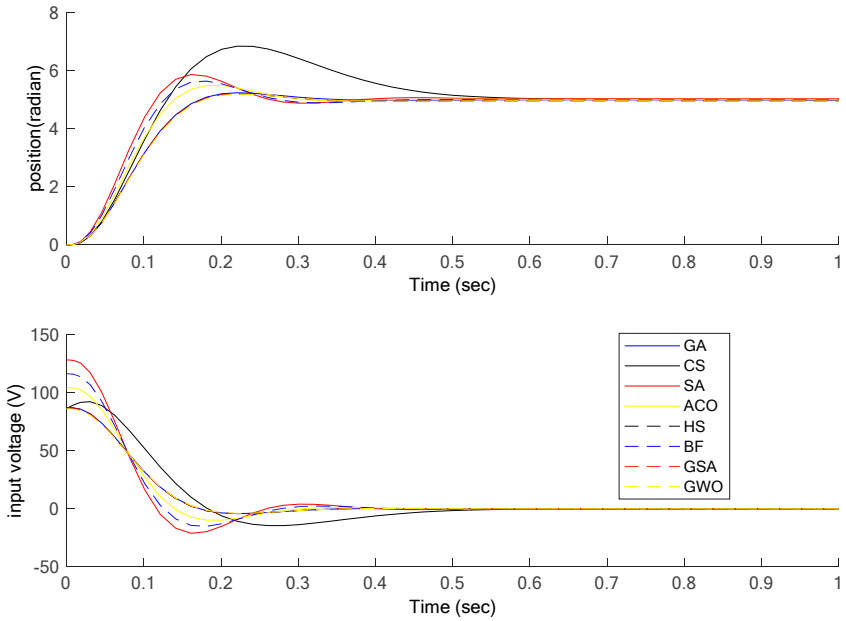


Figure 7. DC motor position control response with a PI controller.

5.2. Case-2: FOPDT Model

The closed loop response of the FOPDT model for a set point 1, is shown in Figure 8 along with the controller output for RTD-A and PID. The RTD-A parameters listed in Table 1 is used for RTD-A implementation. The PID parameters proportional gain K_P = 1.112, integral gain K_I = 0.553, derivative gain K_D = 0.0834, and filter coefficient N = 0.949 are used for experimentation. The RTD-A performance is compared and analysed in Table 7. In Figure 9, the performance of the controllers compared in chart form. It is found that the performance of RTD-A with GA and GWO is almost equal. The proposed controller performance is better than the conventional controller in terms of rise time, settling time, and IAE.

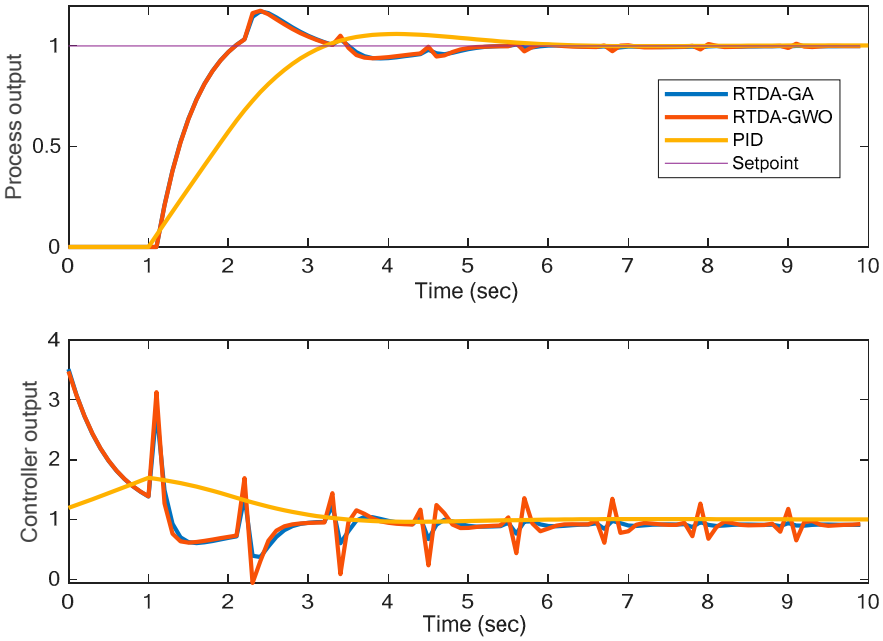


Figure 8. Closed loop response with RTD-A and proportional integral derivative (PID) for the first-order process with dead time (FOPDT) process.

Table 7. Performance comparison of RTD-A and PID for the FOPDT model.

Parameters	RTD-A-GA	RTD-A-GWO	PID
Rise time, TR (secs)	0.6965	0.6972	1.6008
Settling time, TS (secs)	4.9455	6.8178	5.4175
Overshoot, MP (%)	17.2554	17.6782	5.7269
IAE	1.6801	1.6890	2.024

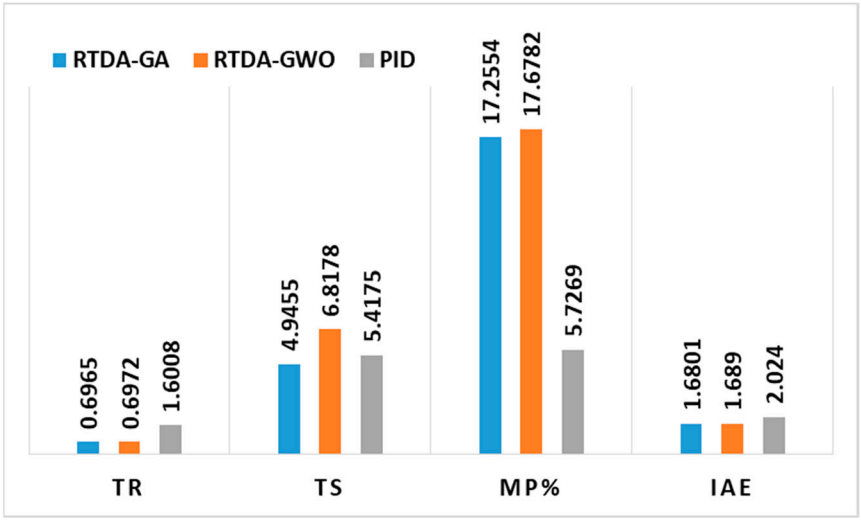


Figure 9. Case-1: Performance comparison chart between RTD-A and PID.

5.3. Case-3: Conical Tank

The conical tank linearized transfer function obtained for different operating heights of 30 cm, 50 cm, and 70 cm. It is given in Equations (25)–(27).

For h = 30 cm

$$\frac{h(s)}{F_{in}(s)} = \frac{2.5475}{450.1887s + 1} \tag{25}$$

For h = 50 cm

$$\frac{h(s)}{F_{in}(s)} = \frac{3.2889}{1614.4s + 1} \tag{26}$$

For h = 70 cm

$$\frac{h(s)}{F_{in}(s)} = \frac{3.814}{3744.005s + 1} \tag{27}$$

5.3.1. RTD-A Implementation

The parameters for RTD-A obtained using genetic algorithm and grey wolf optimization for different linearized transfer function is given in Table 8. The response of the level controlled conical tank using RTD-A for different operating points is shown in Figures 10–12 along with the controller output that is the inflow rate (cm^3/s).

Table 8. RTD-A parameters for level control of the conical tank.

Tuning Parameters	h = 70 cm		h = 50 cm		h = 30 cm	
	GA	GWO	GA	GWO	GA	GWO
θ_A	1.6595×10^{-5}	0	3.6567×10^{-4}	0	0.0015	0
θ_R	0.4733	0.18354	0.1896	0.1822	0.1751	0.18331
θ_T	0.31112×10^{-4}	0.18202	2.7767×10^{-6}	0.18168	7.779×10^{-5}	0.18174
θ_D	0.4473	0.99835	1	1	0.9834	1

5.3.2. PID Implementation

The PID gains for the level control of the conical tank are tabulated in Table 9. The response of the conical tank level using PID for different set points of 30, 50, and 70 cm, is also shown in Figures 10–12 along with the controller output. It is seen that both controllers are able to track the setpoint. The response with RTD-A does not exhibit overshoot and has less rise time and no fluctuation in the controller output when compared to the response with PID at three different level setpoints.

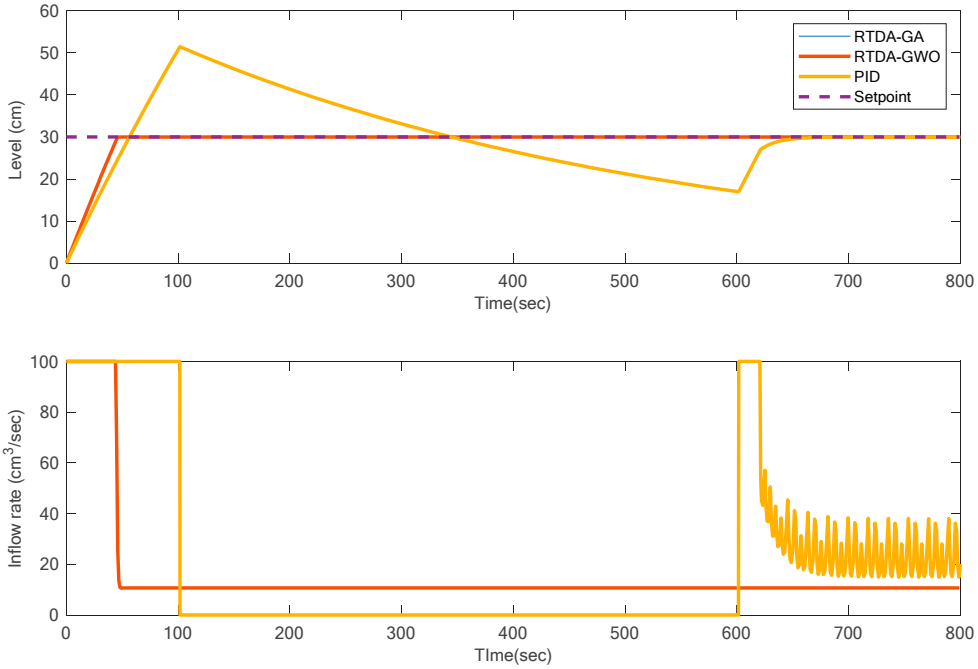


Figure 10. RTD-A and PID controlled level of conical tank for h = 30 cm.

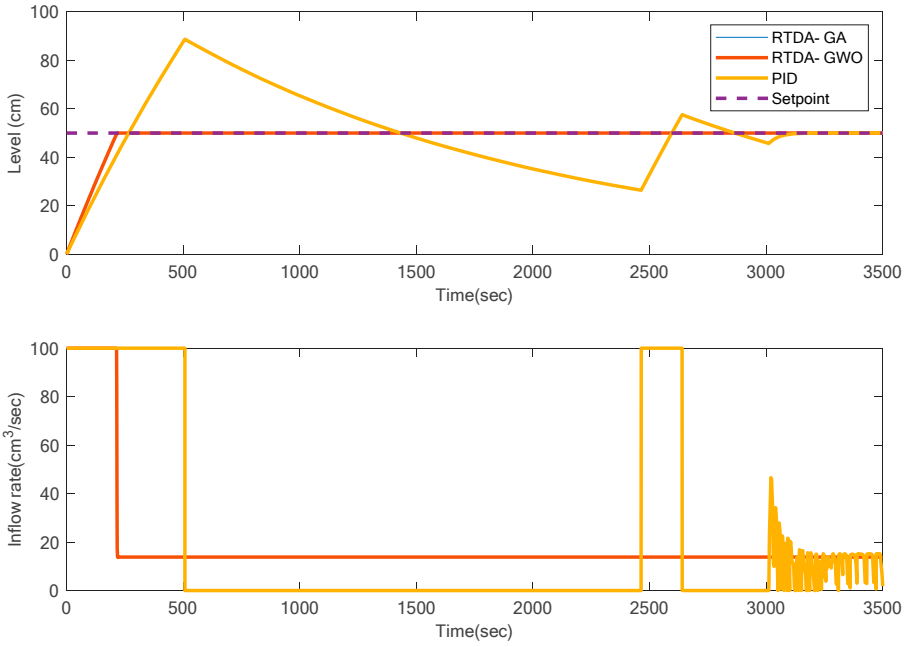


Figure 11. RTD-A and PID controlled level of conical tank for h = 50 cm.

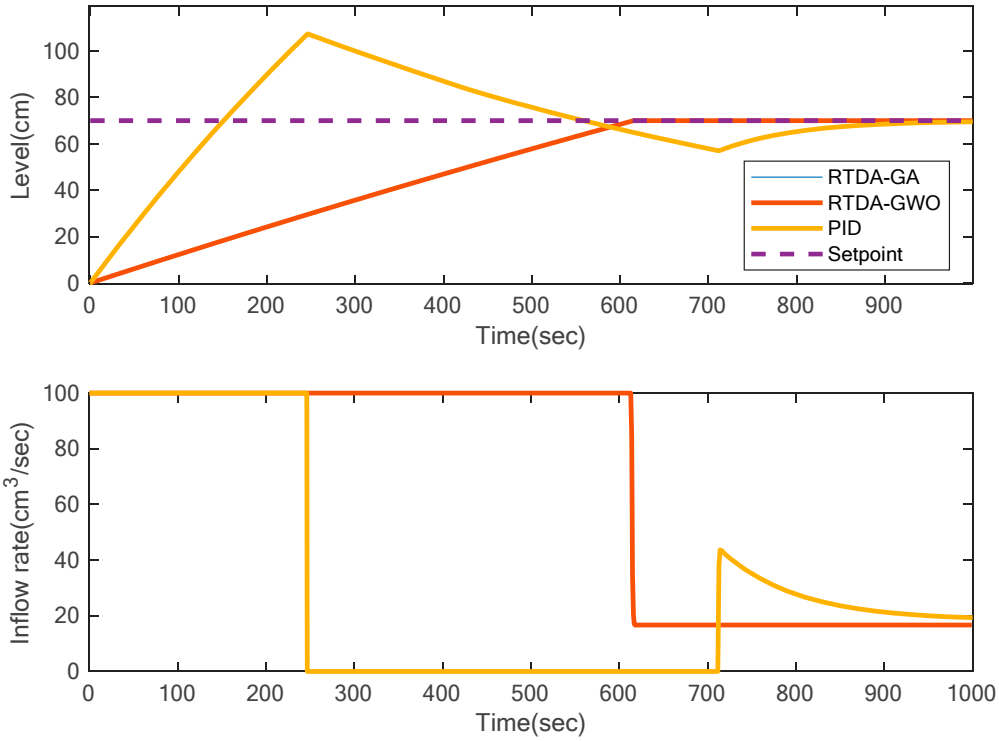


Figure 12. RTD-A and PID controlled level of conical tank for h = 70 cm.

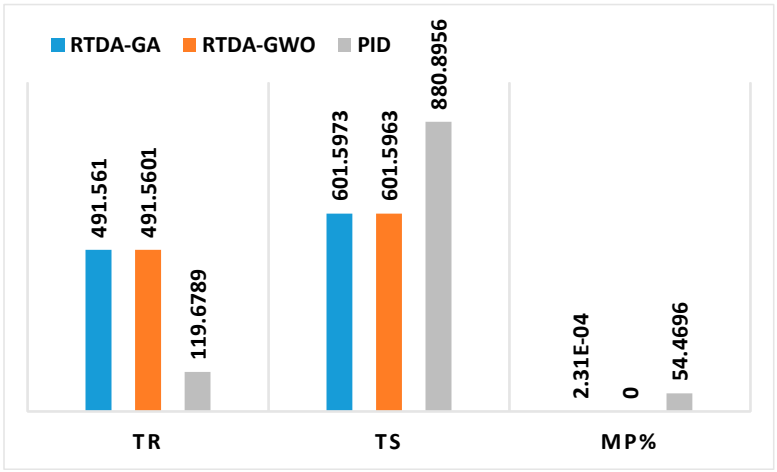
Table 9. PID parameters for level control of the conical tank.

Tuning Parameter	h = 70 cm	h = 50 cm	h = 30 cm
K_P	500	500	500
K_I	5.7598	11.43024	32.01904
K_D	8.0859	0.1127086	0.525

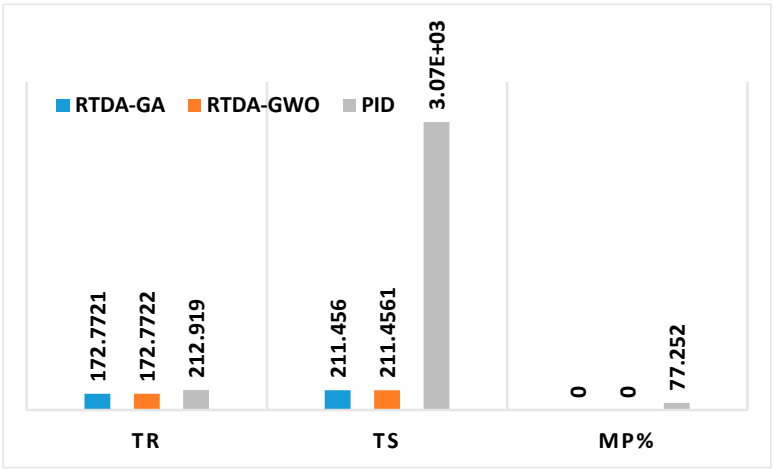
The performance of the RTD-A and PID controller without gain scheduling is compared in terms of time domain specifications in Table 10. Also, the performance comparison chart is plotted in Figures 13 and 14. It has been found there is no significant difference in the process performance with RTD-A controller tuned using GA and GWO from Table 10. The RTD-A performance is better than PID in terms of settling time and percentage overshoot.

Table 10. Performance comparison of RTD-A and PID for the conical tank.

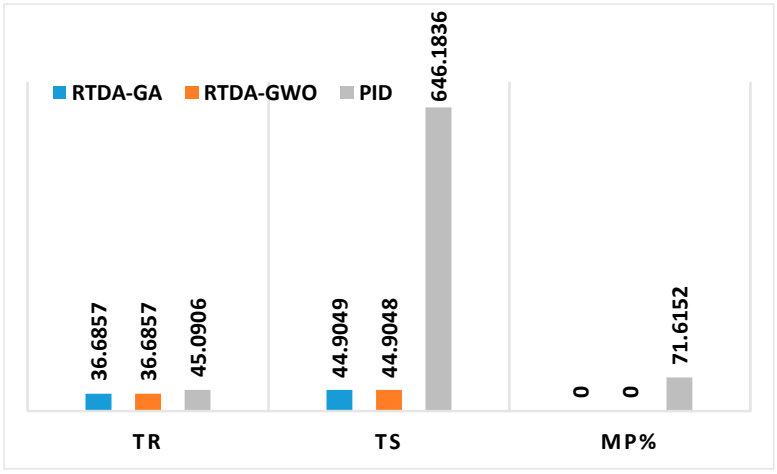
Parameters	h = 70 cm			h = 50 cm			h = 30 cm		
	RTD-A-GA	RTD-A-GWO	PID	RTD-A-GA	RTD-A-GWO	PID	RTD-A-GA	RTD-A-GWO	PID
TR (secs)	491.561	491.56	119.678	172.772	172.772	212.919	36.685	36.685	45.09
TS (secs)	601.597	601.59	880.895	211.456	211.456	3.0746×10^3	44.904	44.904	646.183
MP (%)	2.308×10^{-4}	0	54.469	0	0	77.252	0	0	71.615
IAE	2.091×10^4	2.091×10^4	1.452×10^4	5.301×10^3	5.3×10^3	$4.395*10^4$	694.3041	694.3582	5.732×10^3



(**a**) When h = 70cm



(**b**) When h = 50cm



(**c**) When h = 30cm

Figure 13. Case-2 Performance Comparison between RTD-A and PID: (**a**) when h = 70 cm; (**b**) when h = 50 cm; (**c**) when h = 30 cm.

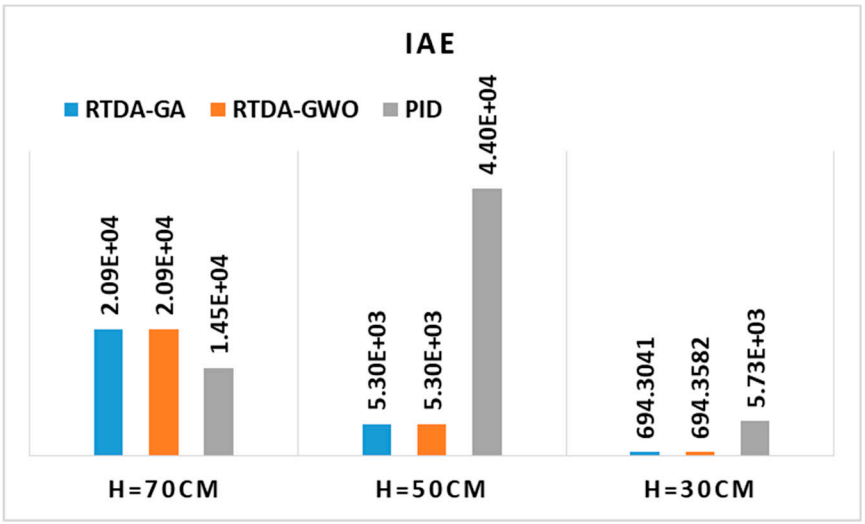


Figure 14. Case - 2: IAE Performance comparison chart between RTD-A and PID

5.4. Gain Scheduled Controller Implementation

The gain scheduled response for level control of the conical tank using RTD-A is shown in Figure 15. The setpoint is changed from 30 cm to 70 cm and again back to 50 cm. Figure 16 shows gain scheduled response using PID. There is no noticeable difference in the level response with RTD-A-GA and GWO. We can clearly see that the performance of the gain scheduled RTD-A controller is better than the PID. The time taken by the RTD-A to track the set point is much shorter and also there is no variation in the inflow rate of the conical tank.

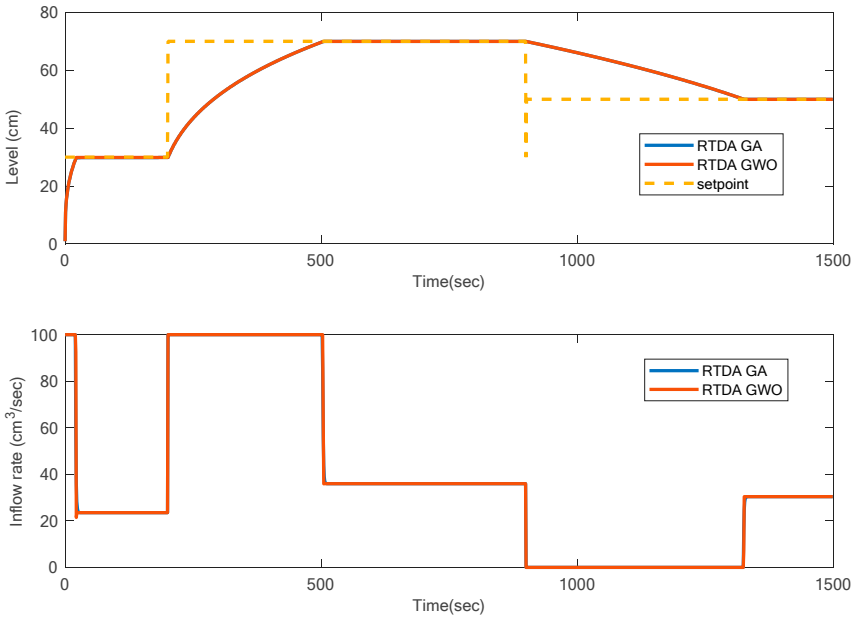


Figure 15. Gain scheduled level control of conical tank using RTD-A with GA and GWO.

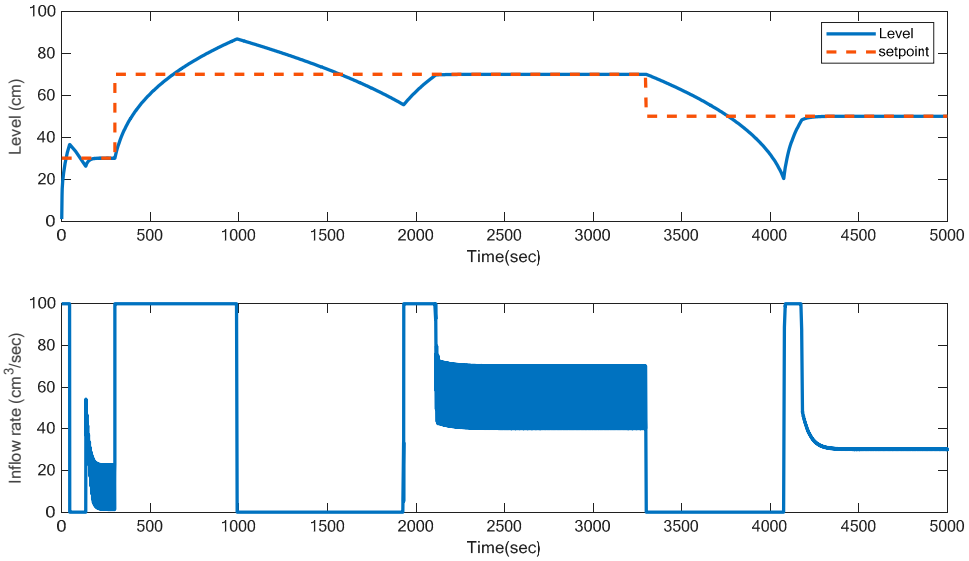


Figure 16. Gain scheduled level control of conical tank using PID.

The performance of gain scheduled RTD-A controller and PID is compared in terms of various integral performance measures. It is presented in Table 11 and also a comparison chart is plotted in Figure 17. It is found that performance with RTD-A is better than PID.

Table 11. Performance comparison of the gain scheduled RTD-A and PID for the conical tank.

Parameters	RTD-A-GA	RTD-A-GWO	PID
IAE	9.2741×10^3	9.2578×10^3	2.754×10^4
ITAE	5.9632×10^4	3.8350×10^4	5.5344×10^7

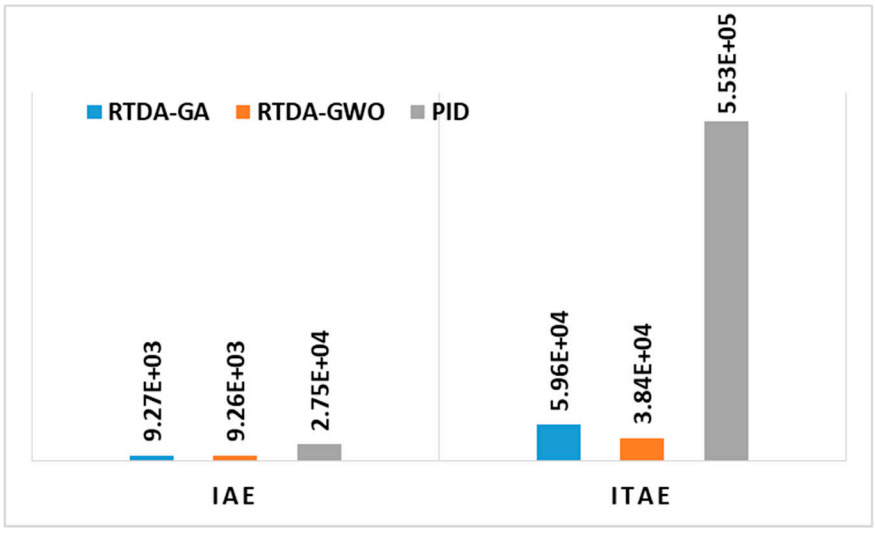


Figure 17. Case-2: Performance comparison chart between the gain scheduled RTD-A and PID.

6. Conclusions

The PID for complex systems exhibits low robustness and disturbance rejection ability. We reverted to finding a controller which imbibes the best characteristics of the MPC and PID. This gave rise to the RTD-A controller. Here, in this study three different processes were used. First, the DC motor position was controlled using an optimally tuned PI controller. It was found that the response with PI-GA and PI-GWO was better than for other algorithms considered. Based on this observation,

parameters were tuned for RTD-A using GA and the GWO optimization algorithm and PID, and applied to the remaining two processes for controlling. The results were compared in terms of various time domain specifications. The performance analysis indicated that RTD-A was more robust and fit for industrial applications. From the nonlinear conical tank level control, we can say that RTD-A is a much better controller than PID at different operating levels. This might make it suitable for industrial applications where it can replace the conventional PID controller. This RTD-A now can be applied to control any non-linear process. Also, gain scheduling was done for RTD-A controller, which is a common method adopted in industries to control the nonlinear process with a linear controller. It is understood that obtaining the first-order model for any industrial process can be a cumbersome procedure. However once the first-order transfer function is obtained, the implementation of this controller becomes easier. These results are much promising than the prior ones, which is clearly evident from the experimental results. The limitation of the proposed controller is that the proposed controller is not suitable for the processes which cannot be represented by the FOPDT model.

Author Contributions: V.B. proposed and implemented the optimal tuning of controller for different processes and wrote the paper; P.A. proposed the idea of gain scheduling of controller and edited the paper.

Funding: This research was funded by VIT, Vellore.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Process Control Fundamentals. Available online: https://www.pacontrol.com/download/ProcessControlFundamentals.pdf (accessed on 1 June 2019).
2. Ogunnaike, B.A.; Wright, R.A. Industrial Applications of Nonlinear Control. In Proceedings of the 5th International Conference on Chemical Process Control, Tahoe, CA, USA, 7–12 January 1996; Volume 93, pp. 46–59.
3. Astrom, K.J.; Hagglund, T. The future of PID control. *Control Eng. Pract.* **2001**, *9*, 1163–1175. [CrossRef]
4. Lequin, O.; Gevers, M.; Mossberg, M.; Bosmans, E.; Triest, L. Iterative feedback tuning of PID parameters: Comparison with classical tuning rules. *Control Eng. Pract.* **2003**, *11*, 1023–1033. [CrossRef]
5. Garrido, J.; Ruz, M.L.; Morilla, F.; Vázquez, F. Interactive Tool for Frequency Domain Tuning of PID Controllers. *Processes* **2018**, *6*, 197. [CrossRef]
6. Silva, G.J.; Datta, A.; Bhattacharyya, S.P. *PID Controllers for Time Delay Systems*; Springer Science & Business Media: Berlin, Germany, 2005; pp. 24–25.
7. Skogestad, S. Probably the best simple PID tuning rules in the world. Presented at AIChE Annual Meeting, Reno, NV, USA, 4–9 November 2001; p. 276.
8. Shubham, P.; Meenakshi, K.; Rajeev, G. Optimal Tuning of PID Controller Using Genetic Algorithm and Swarm Techniques. *Int. J. Electron. Electr. Eng.* **2014**, *2*, 189–194.
9. Ibrahim, H.; Hassan, F.; Shomer, A.O. Optimal PID control of a brushless DC motor using PSO and BF techniques. *Ain Shams Eng. J.* **2014**, *5*, 391–398. [CrossRef]
10. Passino, K.M.; Yurkovich, S. *Fuzzy Control*; Addison-Wesley-Longman: Menlo Park, CA, USA, 1998; pp. 7–8.
11. Templos-Santos, J.L.; Aguilar-Mejia, O.; Peralta-Sanchez, E.; Sosa-Cortez, R. Parameter Tuning of PI Control for Speed Regulation of a PMSM Using Bio-Inspired Algorithms. *Algorithms* **2019**, *12*, 54. [CrossRef]
12. Mukati, K.; Ogunnaike, B. Stability analysis and tuning strategies for a novel next generation regulatory controller. In Proceedings of the American Control Conference, Boston, MA, USA, 30 June–2 July 2004; Volume 5, pp. 4034–4039.
13. Zhong, Y.; Huang, S.; Luo, D. Stabilization and Speed Control of a Permanent Magnet Synchronous Motor with Dual-Rotating Rotors. *Energies* **2018**, *11*, 2786. [CrossRef]
14. Introduction to Intelligent Modeling and Control Approaches. Available online: https://shodhganga.inflibnet.ac.in/bitstream/10603/24249/9/09chapter4.pdf (accessed on 2 June 2019).
15. Mukati, K.; Ogunnaike, B.A. An alternative structure for next generation regulatory controllers Part I: Basic theory for design, development and implementation. *J. Process Control* **2006**, *16*, 499–509.

16. Qin, S.J.; Badgwell, T.A. A survey of industrial model predictive control technology. *Control Eng. Pract.* **2003**, *11*, 733–764. [CrossRef]
17. Tatjewski, P.; Trybus, L.; Ławryńczuk, M.; Marusak, P.; Świder, Z.; Stec, A. Supervisory Control and Optimization. In *Modeling, Diagnostics and Process Control*; Korbicz, J., Kościelny, J.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 233–293.
18. Morari, M.; Lee, J.H. Model predictive control: Past, present and future. *Comput. Chem. Eng.* **1999**, *23*, 667–682. [CrossRef]
19. Mukati, K.; Rasch, M.; Ogunnaike, B.A. An alternative structure for next generation regulatory controllers. Part II: Stability analysis, tuning rules and experimental validation. *J. Process Control* **2009**, *19*, 272–287. [CrossRef]
20. Sreenivas, Y.; Yeng, T.W.; Rangaiah, G.P.; Lakshminarayanan, S. A Comprehensive Evaluation of PID, Cascade, Model-Predictive, and RTD-A Controllers for Regulation of Hypnosis†. *Ind. Eng. Chem. Res.* **2009**, *48*, 5719–5730. [CrossRef]
21. Yelneedi, S.; Lakshminarayanan, S.; Rangaiah, G.P. Advanced Regulatory Controller for Automatic Control of Anesthesia. In Proceedings of the 17th World Congress, the International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008; pp. 11636–11641.
22. Yelneedi, S. Advanced Control Strategies for Automatic Drug Delivery to Regulate Anesthesia during Surgery. Ph.D. Thesis, National University of Singapore, Singapore, 2009.
23. Anbarasan, K.; Srinivasan, K. Fuzzy scheduled RTD-A controller design. *ISA Trans.* **2013**, *52*, 252–267.
24. Kaushal, A.; Thakur, N.; Nagaria, D. Comparison of Speed Control of DC Motor Using Fuzzy PID and PSO-PID Technique. *Int. J. Inf. Comput. Technol.* **2014**, *4*, 553–558.
25. Sabri, N.M.; Puteh, M.; Mahmood, M.R. A Review of Gravitational Search Algorithm. *Int. J. Advance. Soft Comput. Appl.* **2013**, *5*, 1–39.
26. Fister, I., Jr.; Yang, X.-S.; Fister, I.; Brest, J.; Fister, D. A Brief Review of Nature-Inspired Algorithms for Optimization. *Elektrotehniski Vestn.* **2013**, *80*, 1–7.
27. Kumar, D.D.; Meenakshipriya, B. Design and Implementation of non-linear System Using Gain Scheduled PI Controller. *Procedia Eng.* **2012**, *38*, 3105–3112. [CrossRef]
28. Ravi Tailor, M.; Bhathawala, P.H. Linearization of Nonlinear Differential Equation by Taylor's SeriesExpansion and Use of Jacobian Linearization Process. *Int. J. Theor. Appl. Sci.* **2011**, *4*, 36–38.
29. Sendjaja, A.Y.; Ng, Z.F. How and Vinay Kariwala, Analysis and Tuning of RTD-A Controllers. *Ind. Eng. Chem. Res.* **2011**, *50*, 3415–3425. [CrossRef]
30. Guha, D.; Roy, P.K.; Banerjee, S. Load frequency control of large scale power system using quasi-oppositional grey wolf optimization algorithm. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 1693–1713. [CrossRef]
31. Kokash, N. An Introduction to Heuristic Algorithms. 2005. Available online: https://pdfs.semanticscholar.org/8314/bf30780871868076775ba62759f1faf8c9f0.pdf (accessed on 28 March 2019).
32. Aly, A.A. PID Parameters Optimization Using Genetic Algorithm Technique for Electrohydraulic Servo Control System. *Intell. Control Autom.* **2011**, *2*, 69–76. [CrossRef]
33. Aiello, G.; La Scalia, G.; Enea, M. A non-dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems. *Expert Syst. Appl.* **2013**, *40*, 4812–4819. [CrossRef]
34. Nise, N. *Control Systems Engineering*, 4th ed.; Wiley: Hoboken, NJ, USA, 2004; pp. 87–89.
35. Dinesh, C.; Manikanta, V.V.; Rohini, H.S.; Prabhu, K.R. Real Time Level Control of Conical Tank and Comparison of Fuzzy and Classical Pid Controller. *Indian J. Sci. Technol.* **2015**, *8*, 40–44. [CrossRef]
36. Adhikari, N.P.; Choubey, M.; Singh, R. DC Motor Control Using Ziegler Nichols and Genetic Algorithm Technique. *Int. J. Electr. Electron. Comput. Eng.* **2012**, *1*, 33–36.
37. Veselý, V.; Ilka, A. Gain-scheduled PID controller design. *J. Process Control* **2013**, *23*, 1141–1148. [CrossRef]
38. Pradeepkannan, D.; Sathiyamoorthy, S. Implementation of Gain Scheduled PID Controller for a Nonlinear Coupled Spherical Tank Process. *Int. J. Mech. Mechatron. Eng.* **2014**, *14*, 93–98.