

Article

Simulation Tool for Tuning and Performance Analysis of Robust, Tracking, Disturbance Rejection and Aggressiveness Controller

Veeramani Bagyaveereswaran ¹, Subramaniam Umashankar ² and Pachiyappan Arulmozhivarman ^{3,*}

¹ School of Electronics Engineering, Vellore Institute of Technology, Vellore 632014, Tamilnadu, India

² Renewable Energy Lab, College of Engineering, Prince Sultan University, Riyadh 12435, Saudi Arabia

³ School of Electrical Engineering, Vellore Institute of Technology, Vellore 632014, Tamilnadu, India

* Correspondence: parulmozhivarman@vit.ac.in

Received: 13 June 2019; Accepted: 17 July 2019; Published: 20 July 2019



Abstract: The RTD-A (robust, tracking, disturbance rejection and aggressiveness) controller is a novel control scheme that substitutes the classical proportional integral derivative (PID) controller. This novel controller's performance depends on the four controller tuning parameters (θ_R , θ_T , θ_D and θ_A). The tuning of RTD-A controller is more transparent than classic PID controllers. The RTD-A tuning parameters values lies between ZERO and ONE. Availability of a tool to design optimal parameters for this controller and evaluating the performance on a given system is necessary for the researchers. In this paper, the new simulation tool is presented to deal with the RTD-A control scheme. There are four graphical user interface tools included in the proposed tool and working of each tool is explained in detail. To demonstrate the proposed tool, two examples, which involve a liquid level control application and an air pressure control application, are presented in this work. The performance of the RTD-A controller is compared with PID controller. RTD-A controllers are tuned using optimization algorithms and their performances are observed and analyzed in both cases under deterministic and uncertain conditions.

Keywords: controller; optimization; regulatory; simulation tool; servo; tuning

1. Introduction

The proportional integral and derivative (PID) controllers are mostly used to control the critical processes in all over the industries. Most of the industrial closed loop control schemes use the PID controller [1]. Implementing a PID controller is an easy task by considering its simple structure [2]. The process becomes complex by adding a disturbance signal with uncertainty process parameters. The PID controller depends on technology, which is inefficient to fulfill the need of the 21st century. In general, classical control schemes have been positively used only on well-defined plant models [3]. The PID controller has three tuning parameters, which are the proportional gain (K_P), integral gain (K_I) and derivative gain (K_D) [4,5]. The PID controller parameters are difficult to tune in achieving more robust performance with the existence of process uncertainties [6]. Even after choosing the tuning values user will be facing difficulties in achieving the disturbance rejection and set point tracking simultaneously within the restricted time. Most of the industrial processes need a controller with the combo of set point tracking and disturbance rejection ability, which is challenging with the classic PID controller. One may get some unsatisfactory results using tuning rules such as Ziegler–Nichols (ZN) and Cohen–Coon (CC) [7]. It may be debated that contemporary optimization techniques have made the classical tuning methods needless. These techniques mandate a reliable plant model, which

is to be controlled effectively [8]. The user can solve the problem of optimization by considering some advanced version of control scheme such as the model predictive controller (MPC). MPC can be effective in the typical industry situations, but still, they have certain limitations, which are generally not mentioned in advance process control applications. The limitations are difficulty in industry operations, high maintenance cost and less flexibility towards industrial functioning. All the mentioned limits may lead to insubstantial controllers that are not cost-effective. There are many discrepancies while calculating the tuning parameters [9]. The Smith predictor algorithm can be implemented across the controller transfer function for reducing the effect of dead time on the process output. Internal model control (IMC) has the advantage of robust tuning rules, but it has the disadvantage of a large process time constant. In the IMC scheme the user needs to choose integral time equal to the process time constant [10]. Therefore, it is inevitable to design a novel alternate control scheme that has all the combined advantages of PID and MPC.

In 2005, Ogunnaike and Kapil Mukati had introduced a new control algorithm RTD-A, which is more efficient than PID for chemical and other industrial processes [11]. The RTD-A controller is introduced to rectify all the disadvantages of classical controllers such as PID, IMC and MPC. This controller tuning is based on the characteristics such as robustness, tracking, disturbance rejection and aggressiveness of the controller, which is prefixed as RTD-A controller. The RTD-A controller can be implemented only in discrete time. The performance of the controller is greatly influenced by the sampling time and therefore a careful selection of sampling time is inevitable. The clear relationship between the RTD-A controller tuning parameters and characteristics is very well explained and also robust stability of the proposed regulatory controller investigated by Ogunnaike and Kapil Mukati [12,13]. In 2011 Vinay Kariwala introduced the block diagram representation of RTD-A controller implementation and an alternate tuning rule. The rules are focused on optimizing the results, but the Kariwala algorithm was working better than the Ogunnaike rules [14]. In the past, there are tools such as the fractional order modeling and control (FOMCON), Jonathan model predictive controller (jMPC) and N-integer. They could be used for real-time applications. FOMCON is the tool made for the researcher to understand the concept of the fractional order controller technique. The jMPC tool is made for dealing with the model predictive controller in real-time implementation and the N-integer tool is the basic version of the fractional order control [15–17]. The proposed tool is inspired by an educational tool developed by Daniel Carmona Morales for optimal PID tuning using evolutionary strategies in a multi-input multi-output process control application [18]. It is notable that several simulation tools are already available to design various controllers using classical tuning approaches. The researchers can easily understand the new control schemes through hands-on practices using simulation tools [19]. From an educational point of view, the proposed tool will be useful to understand the RTD-A controller concept and characteristics very effectually.

The main contribution of this paper is to introduce a new graphical user interface (GUI) simulation tool for RTD-A controller based on MATLAB, which contains two tuning rules on a single platform in order to control the chosen process as well as analyze the performance of the same. There are four graphical user interface (GUI) tools in the proposed tool. The first tool can reduce the higher order transfer function into first order process with dead time (FOPDT) because the plant model is used to determine the various controller parameters, which will minimize the different performance measures. The second tool deals with basic facility for tuning the RTD-A controller and plotting the closed loop step response, whereas the third tool allows the user to manually adjust and examine the effect of each RTD-A tuning parameters on the process output through GUI [20]. The fourth tool has the facility to find the optimal tuning parameter values using five optimization algorithms such as galactic swarm optimization (GSO), genetic algorithm (GA), particle swarm optimization (PSO), firefly optimization (FFA) and grey wolf algorithm (GWA), which are arbitrarily chosen for the tool design. There are bio-inspired optimization algorithms that can also be used to tune the classical and RTD-A controller parameters [21]. Yugi Fan proposed beetle antennae search algorithm to tune PID controller for controlling the position of electro-hydraulic servo system [22]. The hybrid of the PSO and GA

algorithm is used to self-tune PID for the hydraulic cylinder pressure control [23]. Bilal introduced an improved cuckoo search (CS) algorithm called island based CS. It has two improvements over the basic CS algorithm. It can effectively control the diversity of its population and enhance the exploration of the algorithm [24]. In 2017, a new set of evolutionary algorithm called the estimation of distribution algorithms were introduced by Antonio. It is used to find an optimal solution for complex problems [25]. An ideal gas optimization algorithm based on the first law of thermodynamics and kinetic energy is proposed by Masumeh. The performance of the proposed algorithm compared with the PSO and GA [26]. In 2019, Radu-Emil Precup discussed optimal tuning of the proportional integral fuzzy controller for a servo position control system using nature inspired optimization algorithms. The PSO, GSA, GWO and charged systems search optimization algorithms were used in this application. A hybrid of algorithms was also proposed and compared with the adaptive fuzzy controlled servo system performance [27].

The outline of the paper is organized in the following sections; the concept of the RTD-A control scheme and stability analysis is introduced in Section 2. The effect of RTD-A parameter on the process output is discussed in Section 3. In Section 4, Implementation of RTD-A algorithm with the tool is explained. Section 5 gives the simulation results of the controller. The conclusion and future scope of the work is presented in Section 6.

2. RTD-A Control Scheme

The basic control scheme of the RTD-A controller is discussed in this section. The general closed loop block diagram with the RTD-A controller is shown in Figure 1. There is adequate research going on to find the appropriate alternative algorithm for the PID controller. This alternative RTD-A controller is also known as the regulatory controller [11]. This controller is basically an implementation of an algorithm designed to have the robustness characteristics, tracking ability, disturbance rejection capability and aggressive property. Initially, Kapil Mukati and Ogunnaike have developed a tuning rule that deals with the parametric uncertainty of the controller [13]. In 2011, Vinay Kariwala designed the second tuning method that overcomes the problem of more aggressiveness [14]. The proposed idea concentrates on designing the simulation tool for a RTD-A controller in MATLAB. The control scheme is implemented in three steps [4,12]:

1. Predicting the process output;
2. Updating the model prediction;
3. Computing the control action.

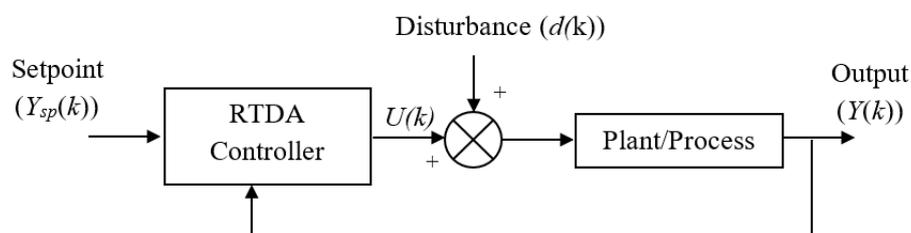


Figure 1. General closed loop diagram with a robust, tracking, disturbance rejection and aggressiveness (RTD-A) controller.

Most of the processes can be represented in the form of the first order process with a dead time (FOPDT) model. It turns out to be very useful to investigate the system output because it has all the necessary components in the unambiguous form. At the first step, the higher order over-damped system transfer function reduced to the FOPDT model using different methods such as fraction incomplete, two-point and Skogestad in the Tool-I [28]. The FOPDT system output used in this paper is given in (1) [29]:

$$y(s) = \frac{Ke^{-as}}{\tau s + 1} u(s), \quad (1)$$

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

$$\bar{y}(k + 1) = a\bar{y}(k) + bu(k - m), \quad k = 0, 1, 2, \dots, \tag{2}$$

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 (2), the output prediction for the next m steps can be written as in (3) [11].

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

where,

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

μ is the weighted sum of the past control action taken during the 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 (4).

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

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

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

where,

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

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

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

The estimate of $e_D(k)$ is given as (8).

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

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 (9).

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

for $m + 1 \leq j \leq m + N$, 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 (5) and (9) the updated N -step model output prediction is given by (10).

$$\tilde{y}(k + m + i) = \bar{y}(k + m + i) + e_D(k + m + i). \tag{10}$$

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 (11).

$$y^*(k + j) = \theta_T^j y^*(k) + (1 - \theta_T^j) y_d(k); \quad 1 \leq j \leq \infty. \tag{11}$$

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 (12).

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

The RTD-A controller output $u(k)$ is given in (13). The calculated controller output was 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 controller output calculation was done in MATLAB. It was used to design a GUI tool described in Section 4.

2.1. Stability Analysis

2.1.1. State Variable Form

When the RTD-A controller based on the nominal model $G(s)$ is implemented on the plant with model uncertainty (λ), the resulting closed-loop behavior may be represented in the discrete state-space form in (14) and (15) [13].

$$X(k+1) = RX(k) + Qu(k) + G_s y_d + G_i d + G_o d_o, \tag{14}$$

$$y(k) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} X(k) \tag{15}$$

where $X(k)$, y_d , d and d_o are the state vector, setpoint, load disturbance and output disturbance respectively.

$$X(k) = \begin{bmatrix} y^*(k+1) & y(k+1) & \hat{y}(k+1) & e_D(k) & \mu(k+1) \end{bmatrix}^T.$$

The control action $u(k)$ is given by (16):

$$u(k) = SX(k) + By_d. \tag{16}$$

The characteristic equation of the closed loop system is given by (17):

$$|zI - R - QS| = 0. \tag{17}$$

where z is the z -transform variable, and, for closed-loop stability, all the roots of (17) must lie strictly inside the unit circle $|z| < 1$.

The matrix R and vectors Q and S in (17) are:

$$R = \begin{bmatrix} \theta_T & 0 & 0 & 0 & 0 \\ 0 & a^0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 \\ 0 & 1 - \theta_R & \theta_R - 1 & \theta_R & 0 \\ 0 & 0 & 0 & 0 & a \end{bmatrix}, Q = \begin{bmatrix} 0 \\ b^0 z^{-m^0} \\ bz^{-m} \\ 0 \\ a - a^{m+1} z^{-m} \end{bmatrix}, S = \begin{bmatrix} \zeta A & 0 & \zeta C & \zeta(zD + E) & \zeta F \end{bmatrix}$$

where A, B, C, D, E, F and ζ are defined as follows:

$$A = \sum_{i=1}^N \eta_i \theta_T^i, B = \sum_{i=1}^N \eta_i (1 - \theta_T^i), C = -\sum_{i=1}^N \eta_i a^{m+i},$$

$$D = -\sum_{i=1}^N \eta_i(1 + \beta(i)); \beta(i) = \frac{\theta_D}{1 - \theta_D}(1 - (1 - \theta_D)^{m+i}),$$

$$E = \sum_{i=1}^N \eta_i \beta(i), F = -\sum_{i=1}^N \eta_i a^{i-1} b, \zeta = \frac{1}{b \sum_{i=1}^N \eta_i^2}.$$

2.1.2. Polynomial Form

The characteristic polynomial of the RTD-A controller in (18) can be written as in [14]:

$$P(z^{-1}) = (1 - \theta_R z^{-1})(1 - a z^{-1}) \left(\sum_{i=1}^N \eta_i^2 (1 - a z^{-1}) + \left(\sum_{i=1}^N a^i \eta_i \right) z^{-1} \right) + \sum_{i=1}^N \eta_i (1 + \beta_{1i} (1 - z^{-1})) (1 - \theta_R) ((1 - a z^{-1}) z^{-(m_0+1)} - (1 - a z^{-1}) z^{-(m+1)}) \tag{18}$$

where

$$\beta_{1i} = \frac{a}{1 - a} (1 - a^{m+i}); i = 1, 2, \dots, N. \tag{19}$$

The stability can be analyzed on the basis of the Jury’s test. A necessary condition for all the roots of the n^{th} order polynomial $\sum_{i=0}^n c_i z^i$ to lie inside the unit circle is that:

$$|c_0| < c_n, \tag{20}$$

where the coefficient c_0 and c_n can be calculated using (21) and (22).

$$c_0 = -a(1 - \theta_R) \sum_{i=1}^N \eta_i \beta_i, \tag{21}$$

$$c_n = \sum_{i=1}^N \eta_i^2. \tag{22}$$

The system will be stable if (23) and (24) is satisfied.

$$P(z) \Big|_{z=1} > 0, \tag{23}$$

$$P(z) \Big|_{z=-1} > 0 \text{ for even power and } P(z) \Big|_{z=-1} < 0 \text{ for odd power.} \tag{24}$$

The closed loop stability in terms of the robustness tuning parameter θ_R can also be determined. The necessary condition is given in (25). The right-side term in (25) is defined as θ_{ref} for simplicity.

$$\theta_R \geq 1 - \frac{\sum_{i=1}^N \eta_i^2}{a \sum_{i=1}^N \eta_i \beta_{1i}}. \tag{25}$$

The discussed necessary conditions in this section are used to check the stability of the liquid level control and pressure control in the evaluation section.

3. Tool Design

The flowchart shown in Figure 2 is the basic building of the tool designed. Each part of the flowchart was implemented in terms of GUI and the corresponding MATLAB functions were designed.

There were four GUI tools, which were designed to work on the RTD-A controller and each tool had its own function. The output of one tool will be an input for another tool. The flowchart starts with the input from the user, the tool checks if the given transfer function is in the FOPDT model or not. If the transfer function is in FOPDT form, the user can proceed further with the desired tool designed in MATLAB. If the transfer function is of a higher order, then the user has to use the tool for model order reduction to obtain the FOPDT model.

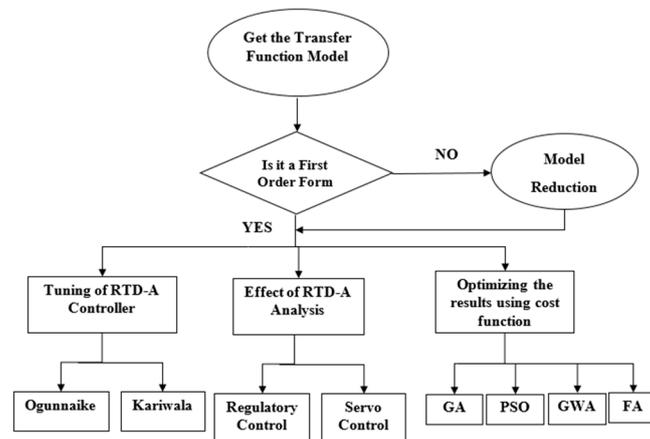


Figure 2. Flowchart for the simulation tool design.

The main aim of the tool is to facilitate the user with all functions used for the RTD-A control scheme. After the final step, the user has the choice to select the analyzing tool. The GUI with RTD-A functionality will be acquired by calling the function name in the command window. The following subsections explain the working of the tool in detail.

3.1. Tool for Model Order Reduction (Tool-I)

It is mandatory to derive the first order process transfer function with/without dead time (when the provided transfer function has a higher order). The Tool-I is designed to reduce the over damped higher order transfer function into the FOPDT model. The required elements of the transfer function are stored in the workspace. This tool is composed of three methods for model order reduction that is as follows, the [4,30]:

1. Skogestad method;
2. Two point method;
3. Fraction incomplete method.

3.2. Tool for Tuning RTD-A Parameter (Tool-II)

After calculating the FOPDT model, to calculate RTD-A tuning parameter the Tool-II was designed. This tool has a drop-down menu containing two methods for tuning [14].

1. The Mukati–Ogunnaike method;
2. The Kariwala method.

The user can select any one method and check the results with the given input (K, τ, α) . The tool also enables the user to plot the closed loop response and compare the results. Each time the tool plots the response in a different color so that the user can easily identify the difference in two outputs. The tool has the option to switch the algorithm and check the response.

3.3. Tool for Analyzing the Effect of the RTD-A Parameter (Tool-III)

The RTD-A control scheme can work with the tuned parameters calculated from the “tune_RTDA” function command. However, sometimes the user wants to adjust the tuning parameters in order to analyze the effect of each parameter on the process output. The Tool-III is used for adjusting the parameters manually and analyzing the results with a graphical representation. The tool can provide the following responses:

- Servo response (performance of the controller without disturbances);
- Regulatory response (performance of the controller considering only disturbances).

In order to analyze the system performance, the tool has the facility to check the performance for different set points. The user can pass in a maximum of three set points at different time instances. The Tool-III is also equipped with provision to add disturbance at a particular instant and time domain analysis with step response can be done. The tool is designed to take the values for K , τ and α from the workspace for the ease of the researcher.

The developed tool has the ability to handle the FOPDT transfer function model parameter values and to display the tuned values of the controller parameter. The effect of each tuning parameter (θ_R , θ_T , θ_D and θ_A) on the plant output is clearly explained by Ogunnaike and Srinivasan [13,31].

3.3.1. Effect of Robustness Parameter

θ_R parameter determines the controller robustness on the process output. The high open loop process gain causes for significant robustness in the face of process parameter uncertainty. It is desirable to have a negligible effect of parameter changes on the process output. The controller action is expected to be swift when θ_R is closer to unity and vice versa when it is near zero.

3.3.2. Effect of the Setpoint Tracking Parameter

θ_T is the parameter that deals with the controller setpoint tracking ability. It does not affect the closed loop stability. θ_T is chosen by considering the closed loop conditions. The error in the process output is determined by the tracking parameter. If the value of θ_T is close to 1, it leads to a sluggish response and vice versa when it is zero.

3.3.3. Effect of the Disturbance Rejection Parameter

θ_D is the parameter that deals with future conditions. Here, the disturbances are deliberated as a composite disturbance. It includes both process model mismatch and unmeasured disturbances. The controller will be more aggressive when θ_D is close to zero. If θ_D is closer to unity the disturbance effect will be changing into the future. Conversely, when θ_D is zero then the disturbance effect will remain constant throughout the system response.

3.3.4. Effect of the Aggressiveness Parameter

θ_A is the parameter associated with the prediction horizon N . The prediction horizon value provides information about how much the future output can be predicted. The controller will be more conservative when the tuning value is high. The RTD-A will be more conservative when compared to the PID controller. An N value is large when the tuning value is near one and it has a short prediction horizon when θ_A is close to zero [14].

3.4. Tool for Optimizing the RTD-A Controller Parameter (Tool-IV)

The Tool-IV is exclusively made for optimization, which can generate the ideal results of the tuning parameters (θ_R , θ_T , θ_D and θ_A) with the help of four different optimization algorithms. The algorithms are chosen arbitrarily. A cost function required for optimization can be chosen by selecting the “Get Cost” function. The main aim of the tool is to determine the optimal tuning parameters by minimizing

the error between the actual process output and set-point. In order to minimize the cost function, the tool implements the intuitively chosen five heuristic algorithms such as:

1. Galactic swarm optimization;
2. Particle swarm optimization;
3. Genetic algorithm;
4. Firefly algorithm;
5. Grey wolf algorithm.

3.4.1. Galactic Swarm Optimization

Galactic swarm optimization (GSO) is a new global optimization metaheuristic algorithm inspired by the motion of stars, galaxies and superclusters of galaxies under the influence of gravity. GSO employs multiple cycles of exploration and exploitation phases to strike an optimal trade-off between the exploration of new solutions and exploitation of existing solutions. In the explorative phase, different subpopulations independently explore the search space and in the exploitative phase, the best solutions of different subpopulations are considered as a superswarm and move towards the best solutions found by the superswarm. In this algorithm subpopulations as well as the superswarm are updated using the PSO algorithm. However, the proposed algorithm is quite general and any population based the optimization algorithm can be used instead of the PSO algorithm [32].

3.4.2. Particle Swarm Optimization

Particle swarm optimization (PSO) is an iterative algorithm to optimize a solution. This algorithm depends on the velocity and position equations. PSO needs the number of iterations and swarm sizes. A particle movement depends on the local best-cost particle. Other particles are nearer to local best approaches towards an optimal point. Finally, all the swarm approaches towards the global minimum and towards the best solution [33].

3.4.3. Genetic Algorithm

The genetic algorithm (GA) is a metaheuristic algorithm that depends on natural selection. The genetic algorithm works on three stages such as mutation, crossover and selection. In a genetic algorithm, individuals are generated randomly. This is an iterative process where the algorithm checks the fitness function. The fitness function produces the fitness value that represents the closeness towards optimized results. The genetic algorithm converges towards the values with optimized results [33]. In particular, the genetic algorithms avoid the local minimum efficiently [34].

3.4.4. Firefly Algorithm

The firefly algorithm (FFA) is a modified particle swarm algorithm. The distinguishable difference is that each particle in this algorithm is characterized by the brightness. Every individual firefly attracts all other fireflies. Attractiveness is proportional to their brightness. Here, the user has to give input of three parameters called α , β and γ . In this algorithm, α , β and γ are the mutation coefficient, attraction coefficient and the absorption coefficient respectively. This algorithm is similar to the particle swarm optimization algorithm when γ tends to zero. The number of swarms and iterations should be specified by the researcher [35].

3.4.5. Grey Wolf Algorithm (GWA)

This algorithm is one of the efficient and simple algorithms present in the research field. It mimics the technique of the grey wolf, which depends on leadership and a hunting mechanism. There are preys that are the food of wolves. The algorithm employs four grey wolves that are simultaneously working on three tasks that are searching, encircling and attacking prey [36].

All these tuning parameters will affect the final controller output. The optimal tuned values for the four parameters can be obtained by minimizing the suitable cost function. The controlled process response can be altered by adjusting the tuning parameters. The challenge in the optimal tuning of the RTD-A controller parameters is succeeded by the proposed tool.

The researchers have developed different methods for dealing with the controller tuning parameters. One such technique is the Ogunnaike tuning rule. It has one notable parameter called the composite parameter ρ . It is given in (26):

$$\rho = \lambda \left(\frac{1 + \lambda}{1 - \lambda} \right) \frac{\alpha}{\tau}, \tag{26}$$

where λ is the multiplicative factor, which is associated with process uncertainty and its value lies in the range of 0 to 1 for the FOPDT process model.

When the uncertainty parameter λ is added to the nominal plant model in (1), the plant parameters is defined in (27).

$$K^0 = K(1 + \lambda), \tau^0 = \tau(1 + \lambda) \text{ and } \alpha^0 = \alpha(1 + \lambda). \tag{27}$$

The equivalent discrete plant parameters is given in (28):

$$a^0 = e^{-\Delta t/\tau^0}, b^0 = K^0(1 - a^0) \text{ and } m^0 = \text{round}(\alpha^0/\Delta t). \tag{28}$$

The plant output can be written as (29), where d is the disturbance.

$$y_p(k) = a^0 y_p(k - 1) + b^0 (u(k - m^0) - 1) + d(k - m^0 - 1). \tag{29}$$

Similarly, the model output from (2) can be written as (30).

$$\bar{y}(k) = a\bar{y}(k - 1) + bu(k - m - 1). \tag{30}$$

The model error $e(k)$ in (31) is used in (7) to calculate the control action.

$$e(k) = y_p(k) - \bar{y}(k). \tag{31}$$

There are definite constraints on the tuning parameters with respect to the composite parameter. The Mukati–Ogunnaike tuning rule is presented in Table 1.

Table 1. Mukati–Ogunnaike tuning rules [4].

Composite Parameter	θ_R	θ_T	θ_D	θ_A
$\rho < 1$	0.5	0.8	≥ 0.5	0.9ρ
$1 \leq \rho \leq 1.6$	0.9	0.8	≥ 0.1	$1.25(1 - e^{-\rho})$
$\rho \geq 1.6$	> 0.95	0.8	$\geq 1 - \theta_R$	0.995

The Kariwala, tuning rule is shown in Table 2. It is a semi-analytical tuning method. The tuning rules are obtained with the extensive experimental study on the different FOPDT transfer function [11].

Table 2. Kariwala tuning rules [4,14].

θ_R	θ_T	θ_D	θ_A
$\min\left[\frac{m}{m+1}, 0.95\right]$	a^2	$1 - a$	$1 - (1/ae)$

The variety of the first order transfer function models is tested with robust stability analysis and the plant model mismatch using the Kariwala tuning method in the literature. Here, the researcher can observe that the tuning rule is transparent with respect to the controller performance features. The Kariwala tuning rule demonstrates the decent performance for a variety of processes. In some cases, Mukati–Ogunnaike tuning rules may lead to instability because of its inherent aggressiveness characteristics [14].

The model order reduction techniques, tuning rules and optimization algorithms discussed in this section were used to design a GUI tool in the tool description section. The proposed four tools are described with GUI in Section 4.

4. Tool Description

The FOPDT process model is required for the RTD-A controller design. If the process transfer function is not in the FOPDT form, then the model order reduction tool (Tool-I) will be used. The higher order process dynamics can be represented by the reduced model. The process reaction curve (PRC) method is one of the techniques to reduce the higher order process model into the first order model with dead time. It is considered to achieve a minimum integral squared error (ISE) in its ideal open loop step response. The GUI to reduce the higher order transfer function into the first order model is shown in Figure 3. Tool-I has the ability to reduce the higher order transfer function into the first order transfer function $G(s)$. The proposed Tool-I has experimented on the liquid level process and it is discussed in the evaluation section [4].

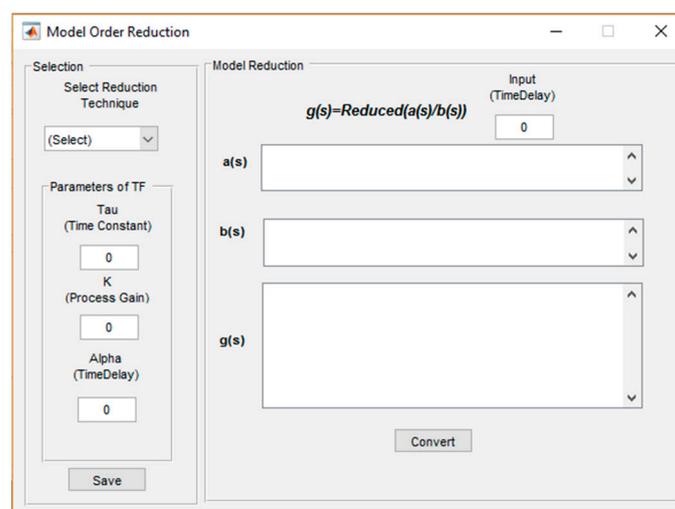


Figure 3. Graphical user interface (GUI) for the model order reduction tool.

The outline of the method of model order reduction is given in (32).

$$Reduced\ Order\ Model\ G(s) = method \left\{ \frac{a(s)}{b(s)} \right\}, \tag{32}$$

where $a(s)$ and $b(s)$ is the numerator and denominator polynomial of the higher order process transfer function respectively and $G(s)$ is the required reduced order model with or without dead time [4].

Generally, the system has some dead time when it is tested for robustness in a real-time environment. The Tool-I has the option to add a time delay while testing the system. Each order reduction method takes the dead time as one of the input and provides the required reduced order model output. Since the RTD-A controller depends on the process gain, process time constant and dead time, Tool-I basically separates all three parameters. It has also the facility with three model order reduction methods such as the Skogestad, two-point and fraction incomplete method. The chosen methods are effective for an over-damped process with dead time.

Next, this paper introduced the second tool that was designed for calculating the RTD-A controller tuning parameters. The GUI for the Tool-II is shown in Figure 4. This Tool-II was designed by considering the first order process with dead time models. It has the flexibility to tune the controller with both Oggunnaike and Kariwala tuning rules. The developed GUI has the provision for getting different inputs such as the sampling time (dt), process uncertainty parameter (λ), the process dead time (α), composite noise factor (f_n) and process time constant (τ) from the user. The user can choose one tuning rule from the selection menu. The calculated, tuning values will be shown in the RTD-A parameter portion of the Tool-II. If the user selects another tuning method from the menu during the next stint, the user has the provision to compare both results just by enabling the “Tune” button in operating the menu. Tool-II also has the facility to reset all the values to zero [20]. The tuning rules used in Tool-II are presented in Tables 1 and 2.

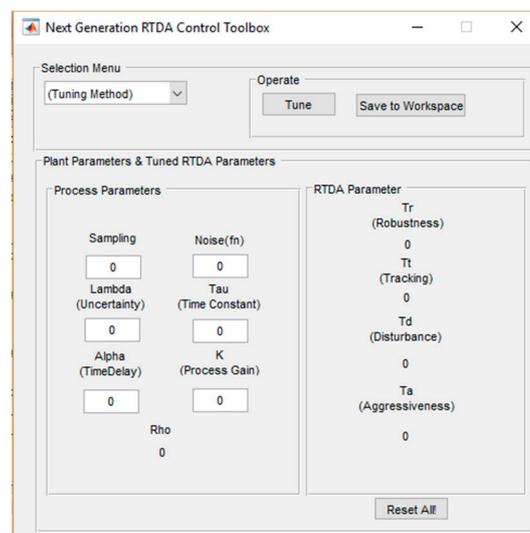


Figure 4. For the RTD-A controller tuning tool.

In order to understand the effect of each controller parameter (θ_R , θ_T , θ_D and θ_A) on the process output the user has to change the value manually. The tool is made in such a way that the researcher has the facility to tune the controller manually with a slider option. The representation of Tool-III from the proposed tool is shown in Figure 5. The designed Tool-III has the facility to give a different set point at a different time instant and the user can perform a time domain analysis from the step response of the process.

The disturbance can be added at a required instant and the user can check the effect of a tuning parameter on the process output with disturbance. Tool-III has also the provision to save current values of tuning parameters. After giving a set point to the tool, the user can change the values of the tuning parameters and check the tuning values in the slider display. The researcher can visually check the controller output and closeness to the ideal values. It can simulate all features of RTD-A by comparing the results in the same figure.

Manual tuning always lags behind tuning results with an optimization algorithm. The various control schemes like PID, model predictive control (MPC), fractional order PID (FOPID) and linear quadratic regulator (LQR) are available in the past. In all these algorithms many researchers have implemented optimization algorithms to get the optimum tuned results. In this paper, Tool-IV was proposed to find the optimal tuning values for the RTD-A control scheme. There were four algorithms included in the Tool-IV with the drop-down menu. The GUI is shown in Figure 6.

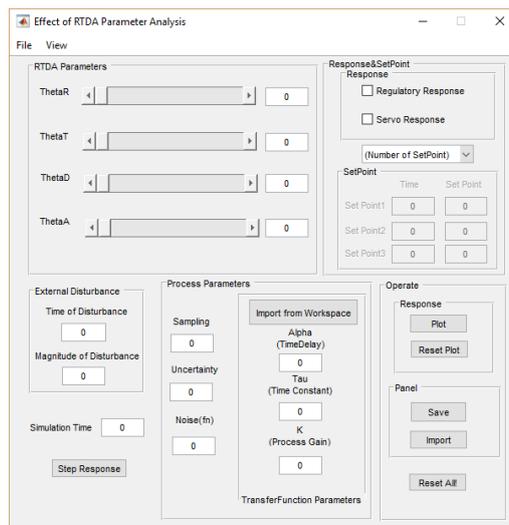


Figure 5. GUI for analyzing the effect of the RTD-A parameter.

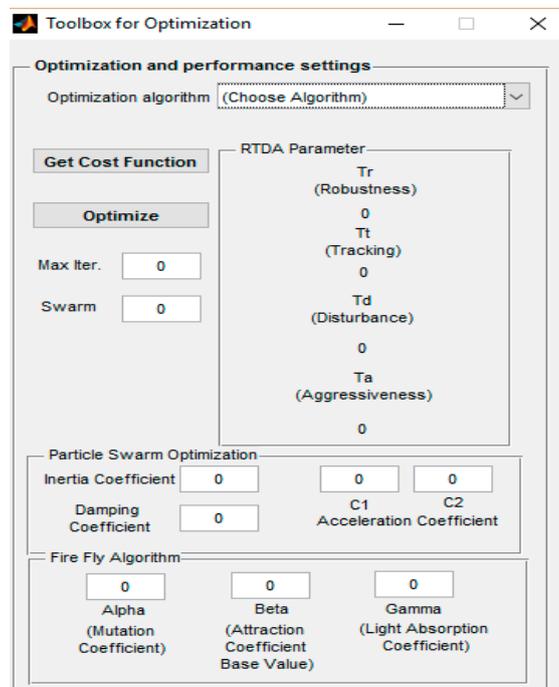


Figure 6. GUI for the RTD-A controller parameter optimization tool.

The tool has the facility to give a path for the cost function. The main objective of the optimization algorithm is to minimize the cost function. The least cost corresponds to the optimized RTD-A values. After pressing the 'Optimize' button, the user will have to wait for iterations to be completed. The least cost can be obtained in the MATLAB workspace. The Tool-II or III can be used to plot the closed loop process response with the optimized RTD-A parameter. The tools discussed in this section are experimented on two examples in the evaluation section.

5. Evaluation

In this section, the proposed tool has experimented on two different industrial processes such as the liquid level control and pressure control. The experiment was conducted on a personal computer running windows 10 pro 64-bit operating system with hardware configuration of Intel(R) Core(TM) i5-7200U CPU @ 2.5 GHz.

5.1. Example 1: Interacting a Two Tank Liquid Level System

The liquid level system shown in Figure 7 was considered for validating the proposed tool [4]. This is one of the commonly used real-time examples of the chemical process [37]. In a liquid level system, at steady state, the inflow rate and outflow rate are equal to \bar{Q} . The flow rate between the two tanks is also zero and the heads of the tank 1 and tank 2 are equal to \bar{H} . At time $t = 0$, the inflow rate is changed from \bar{Q} to $\bar{Q} + q$, where q is small. The resulting changes in the heads (h_1 and h_2) and flow rates (q_1 and q_2) are assumed to be small as well. The capacitances of tank 1 and tank 2 are C_1 and C_2 respectively. The resistance of the valve between the tanks is R_1 and the outlet valve resistance is R_2 .

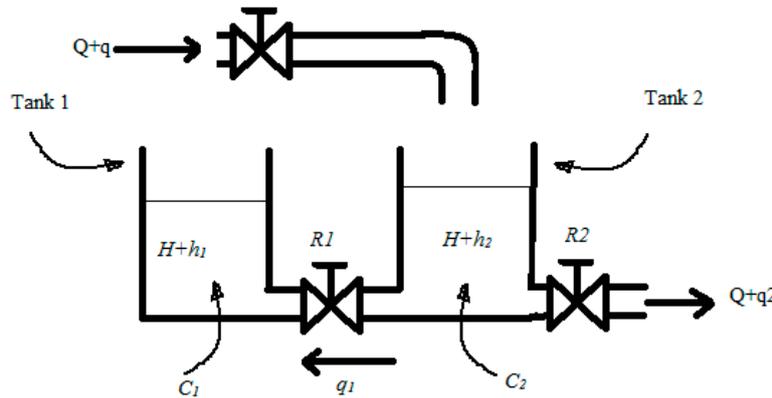


Figure 7. Two tank liquid-level system [4].

From Figure 7, the transfer function between liquid level $H_2(s)$ and inlet flow rate $Q(s)$ is derived and given in (33).

$$\frac{H_2(s)}{Q(s)} = \frac{R_1 C_1 R_2 s + R_2}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_2 C_1) s + 1} \tag{33}$$

In order to understand the effect of the dead time in the process output, five seconds in dead time was introduced in the derived transfer function. Let $R_1 = R_2 = 1$; $C_1 = C_2 = 1$. Now the resulting transfer function is given in (34):

$$\frac{H_2(s)}{Q(s)} = \frac{(s + 1)e^{-5s}}{(s^2 + 3s + 1)} \tag{34}$$

It is a minimum phase stable system. Let us consider that the level control system is time critical and it is required to control the second tank level $h_2(t)$ by manipulating the inflow rate $q(t)$. Since it is required to control $h_2(t)$ using the RTD-A controller, the actual system in Equation (34) is reduced into the FOPDT system. The three methods in Tool-I were used to obtain the FOPDT model of the process. The result of each method is presented in Table 3. The accuracy of the reduced model in terms of the integral square error (ISE) performance measure was also compared. It was found that the fraction incomplete method based model was more accurate for this interacting liquid level system. Hence, in this section, the remaining analysis was carried out with the obtained accurate reduced model in (35).

$$G(s) = \frac{e^{-4.82s}}{2.69s + 1} \tag{35}$$

Table 3. Performance comparison of order reduction techniques.

Methods	Skogestad	Two Point	Fraction Incomplete
Reduced Model $G(s)$	$\frac{e^{-4.19s}}{2.809s+1}$	$\frac{e^{-4.9s}}{2.68s+1}$	$\frac{e^{-4.82s}}{2.69s+1}$
ISE	0.066	0.0518	0.0444

The open loop response of a reduced order model is displayed in Figure 8. The fraction incomplete method-based model response was very much closer to the actual second order process response than other methods [4]. The implementation of the RTD-A controller with a simulation tool on the liquid level control is shown in Figure 9. The performance of the RTD-A controller with two tuning algorithms and the PID controller was examined in controlling the $h_2(t)$ without uncertainties and disturbance. The structure of the PID controller transfer function used is given in (36). The PID tuning parameters proportional gain (K_P), integral gain (K_I), derivative gain (K_D) and filter co-efficient (T_f) used in the simulation were determined using the auto tune function in MATLAB.

$$G_C(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{T_f s + 1}. \tag{36}$$

The tuning parameters for the process with uncertainty and disturbance using two tuning rules are given in Table 4. The sampling time of 0.1 s was considered in the controller implementation. The PID auto tuned values were $K_P = 0.496$, $K_I = 0.116$, $K_D = -0.857$ and $T_f = 0.128$. The setpoint for $h_2(t)$ was considered as one meter. The servo response with the PID and RTD-A for zero uncertainty and zero disturbance is shown in Figure 10. Similarly, the servo response with RTD-A tuned using the optimization algorithm is shown in Figure 11.

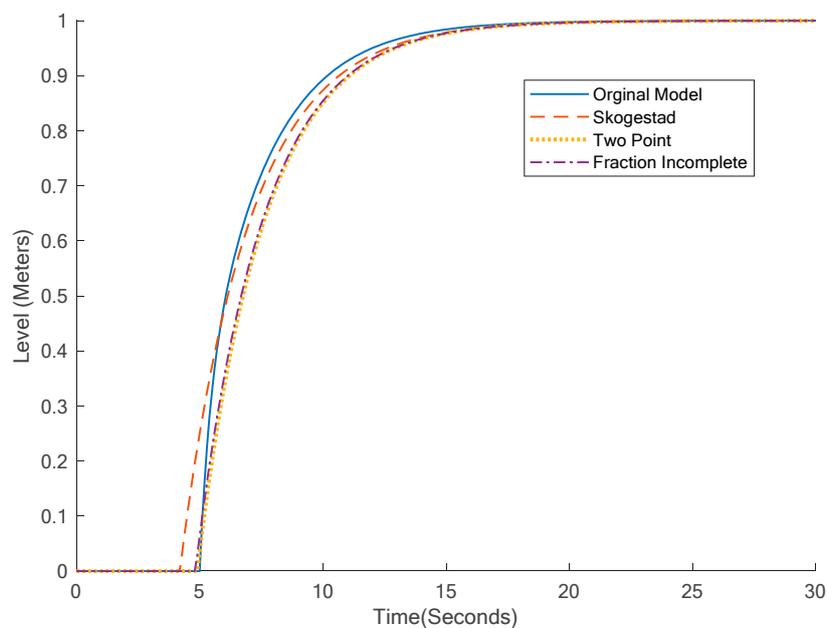


Figure 8. Open-loop step response with reduced models.

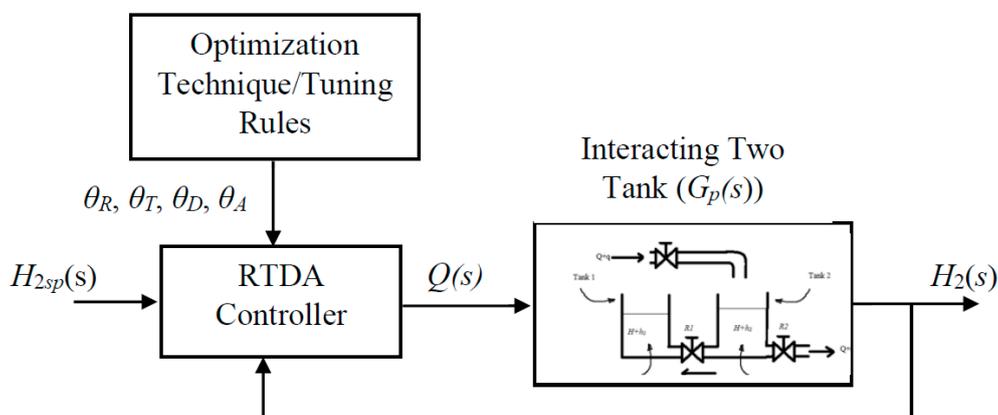


Figure 9. Level control system with the RTD-A controller.

Table 4. Example 1: Tuning parameters with two tuning methods [4].

Parameter	Ogunnaike	Kariwala
θ_R	0.5	0.95
θ_T	0.9180	0.9207
θ_D	0.5	0.0405
θ_A	0.1971	1

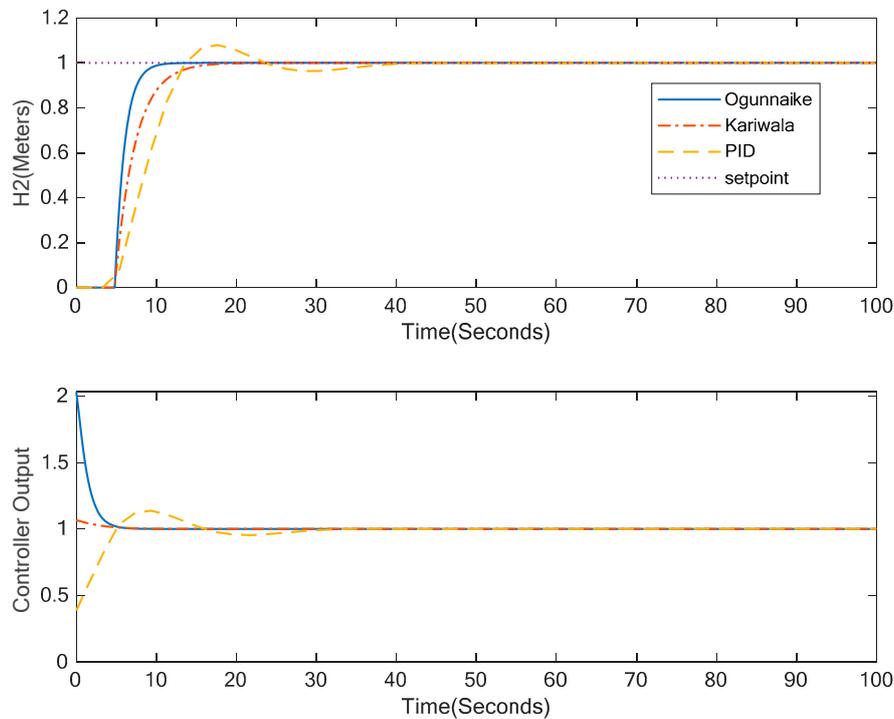


Figure 10. Example 1: Servo response with the RTD-A controller using two tuning methods.

The efficiency of the proposed tool was examined by adding the uncertainty parameter (λ) and disturbance to the plant. The RTD-A and PID controlled servo response with disturbance and uncertainty are shown in Figures 12 and 13. The RTD-A controller requires values for the sampling time, noise and uncertainty parameter as an input. The tool II was provided with the same input parameters for comparing the two different tuning algorithms. Whenever the user chooses the tuning rule, the tool will acclimate to the chosen tuning rule. In Figure 12, the response with the Ogunnaike tuning method was more susceptible to noise and also it exhibited a more aggressive controller response. It had some overshoot as compared with the Kariwala tuning method. The process response with the first method deviated more from the setpoint and this deviation increased further with the increase in the uncertainty parameter (λ). The Kariwala, tuning method was more flexible with respect to the controller aggressiveness and uncertainty. The response with PID had more rise and settling time with less overshoot and control variation than the Ogunnaike method. In Figure 13 the servo response with optimal tuned parameters after adding uncertainty is shown.

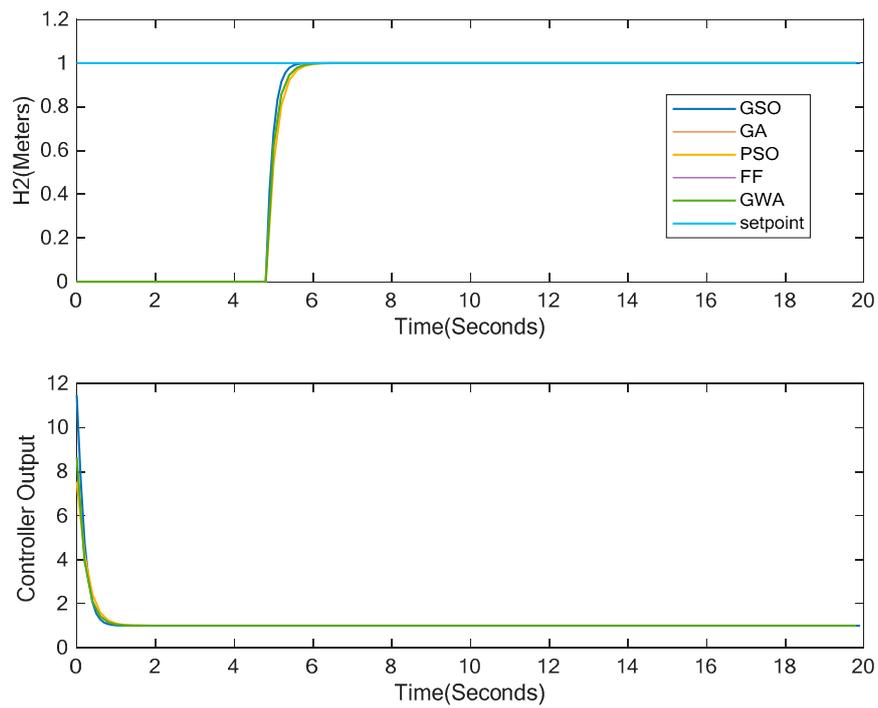


Figure 11. Example 1: Servo response with the RTD-A controller using optimization algorithms.

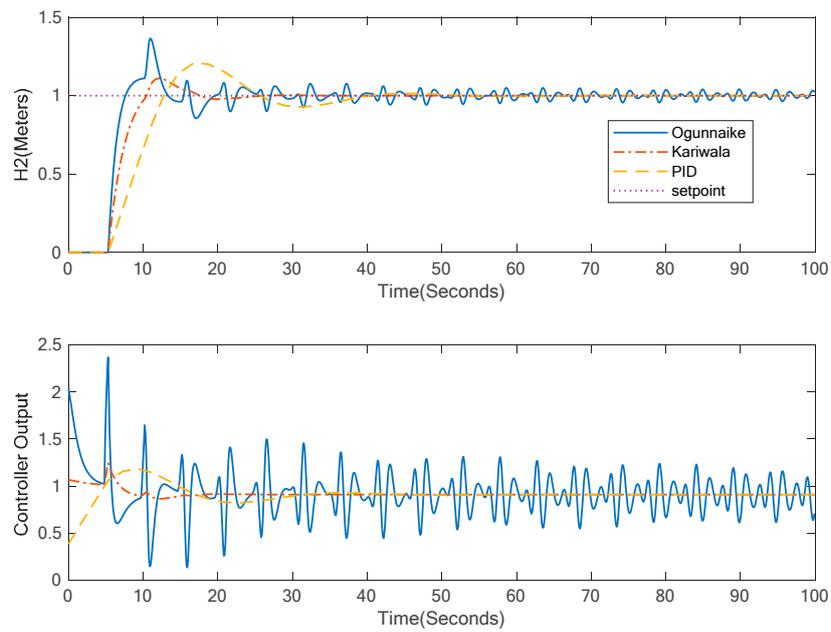


Figure 12. Example 1: Servo response with the RTD-A controller with disturbance and uncertainties.

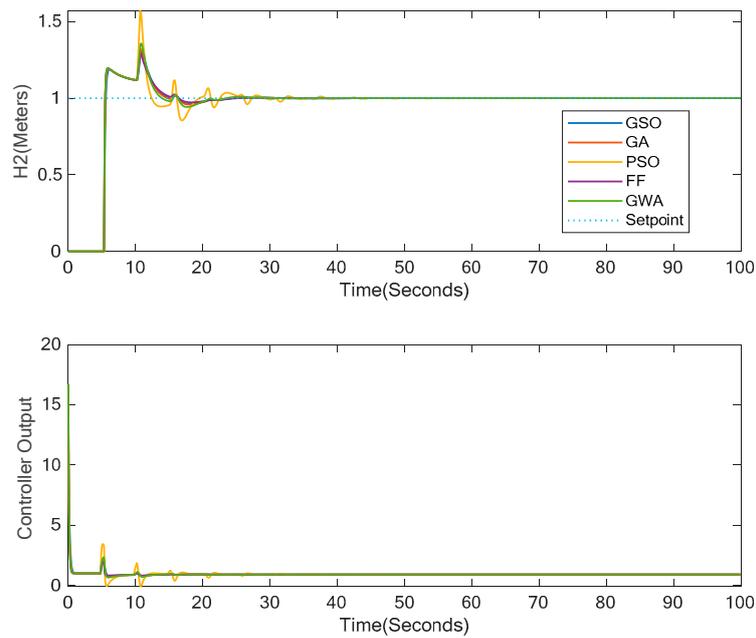


Figure 13. Example 1: Servo response with the RTD-A controller with uncertainty using optimization algorithms.

The disturbance rejection ability of the RTD-A controller was experimented and the regulatory response without optimally tuned parameters is shown in Figure 14. In Figure 15, the response with disturbance rejection is shown for some optimally tuned values. The disturbance of 20% of setpoint was applied to the process at 50 s. It was found that all the controllers were able to reject the disturbance effect on the process output.

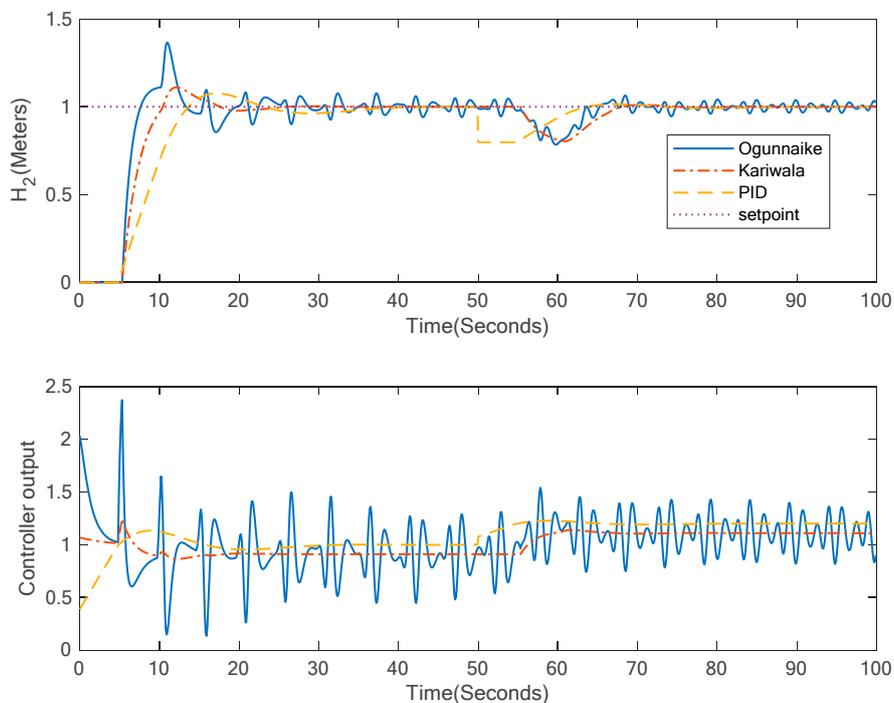


Figure 14. Example 1: Regulatory response with the RTD-A controller using tuning rules and proportional integral derivative (PID).

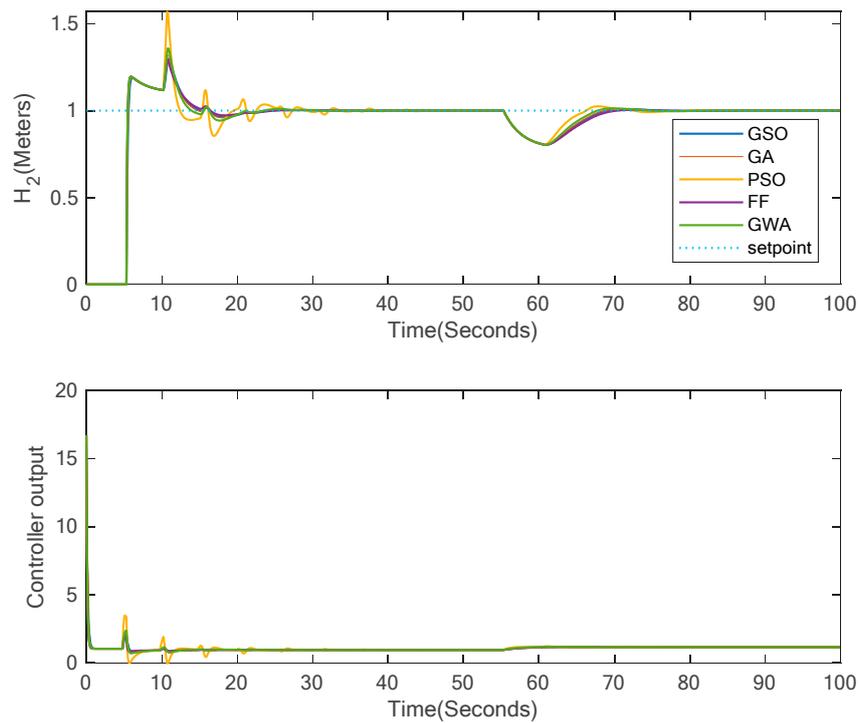


Figure 15. Example 1: Regulatory response with the RTD-A controller using optimization algorithms.

The unit step change in setpoint was applied to the closed loop system and time response was obtained. The time domain specifications could be calculated from the step response of the closed loop system using the stepinfo command in MATLAB. The RTD-A controller performance for two tuning rules and optimal tuning was compared in terms of the integral absolute error (IAE) and other time domain specifications with PID performance and it is presented in Table 5. In this example performance with RTD-A was better than PID. Between the first two tuning rules, the Kariwala tuning rule-based process response had a good performance compared to the Oggunnaike tuning method excluding the rise time. However, in the overall comparison the firefly algorithm-based result was slightly better than the other methods.

Table 5. Example 1: Performance comparison with different tuning methods.

Specifications	PID	Ogunnaike	Kariwala	GSO	GA	PSO	Fire-Fly	GWA
IAE	9.52	9.079	7.61	6.89	6.88	7.427	6.85	6.932
Rise Time (s)	5.87	1.641	3.578	0.22	0.1408	0.1626	0.1407	0.14
Settling Time (s)	38.05	21.1	13.78	20.49	20.36	22.81	20.12	20.24
Overshoot (%)	20.78	33.57	12.98	29.67	31.94	56.92	29.17	35.81
Steady State Error (e_{ss})	2.46×10^{-5}	0.02	0	0	0	0	0	0

The execution time of the RTD-A and PID controller implementation for the servo response of level control without uncertainty was 0.034279 s and 0.062699 s respectively. Similarly, 0.070815 s and 1.048175 s with 10% uncertainty on process parameters when the simulation time was 100 s.

The Tool-III was applied to the liquid level control system and the effect of the tuning parameter on the process output was analyzed. Each parameter can be changed with the slider and even small changes in the process output can be realized. In Example 1, the θ_R , θ_T and θ_A values were changed and the output of the plant with respect to time was plotted. The output of Tool-III for Example 1 with sequential changes in setpoints (SP = 1 m, SP = 5 m and SP = 2 m) is shown in Figures 16–18.

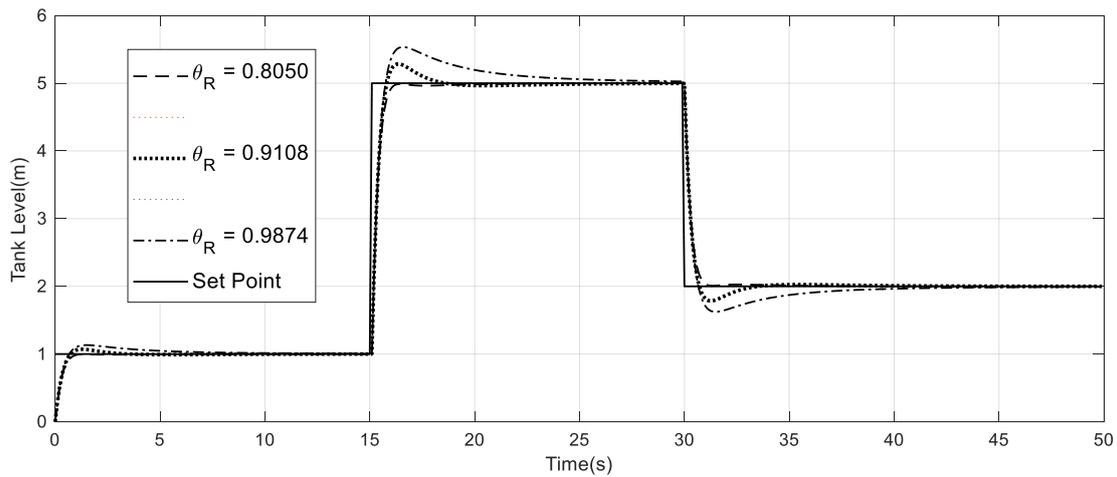


Figure 16. Response to Tool-III for changing the θ_R parameter.

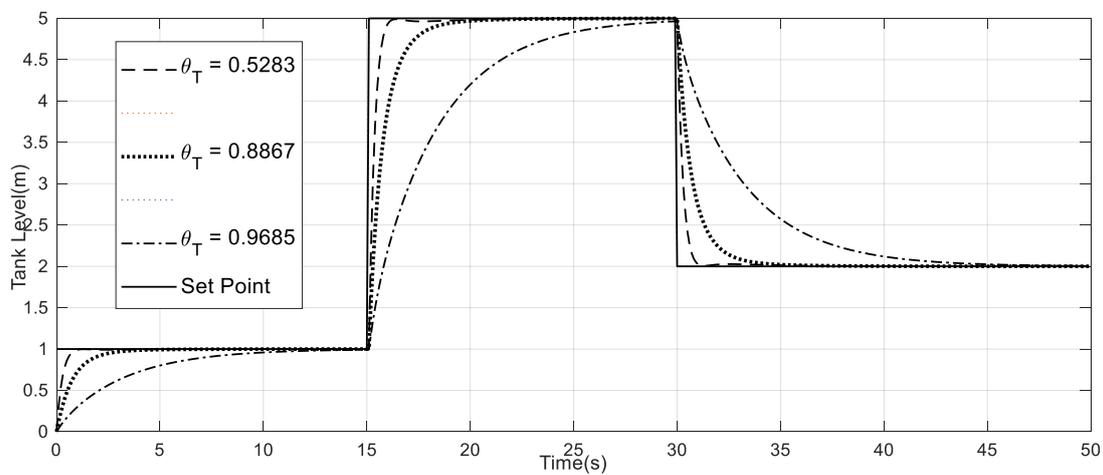


Figure 17. Response to Tool III for changing the θ_T parameter.

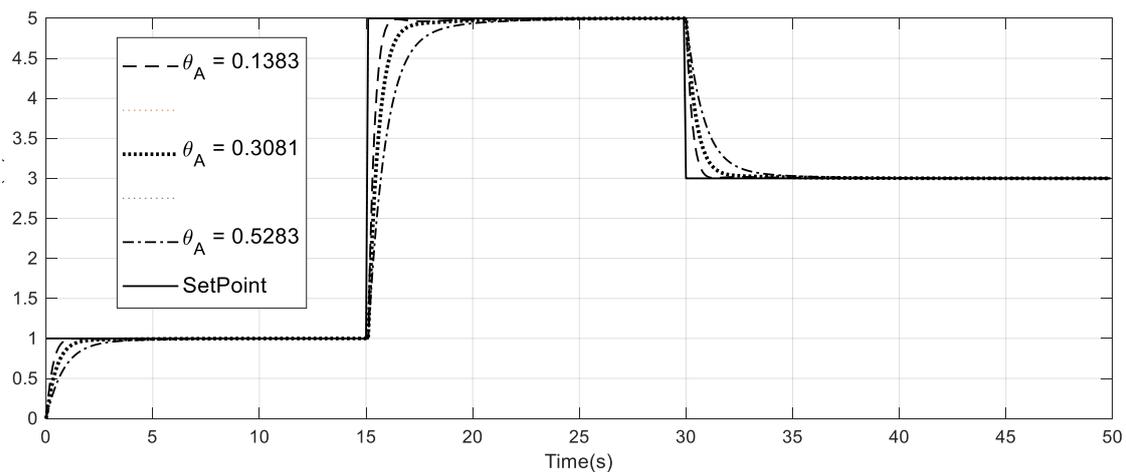


Figure 18. Response to Tool III for changing the θ_A parameter.

In Figure 16 the process response rise time decreased and settling time increased as the θ_R approaches unity. The response was robust as the θ_R value was small. The effect of the setpoint tracking parameter on the process output is shown in Figure 17. The response became sluggish and the settling time increased as the θ_T approached unity. In Figure 18, as the θ_A value approached 1, the response was moving away from the desired set point. Therefore, for the lesser value of θ_A ,

the process output was closer to the desired level. Similarly, the process output reached the set point very quickly without overshoot as the parameter θ_A was small.

Tool-IV experimented with the liquid level control system. The objective of the optimal control problem was to minimize the error between the setpoint (y_d) and actual process output (y_p). Therefore, the integral absolute error (IAE) cost function in (37) was chosen for both example 1 and example 2. During this optimal tuning process, an output disturbance was also considered with $\pm 10\%$ random variation.

$$IAE = \int_{t_0}^{t_f} |(y_d - y_p)| dt. \tag{37}$$

The number of iterations and swarm size used in the optimization is tabulated in Table 6. The additional parameters required for PSO and firefly algorithms are listed in Table 7. The algorithm parameters were chosen based on a certain number of trial runs for the smooth convergence. The same settings were used in the Example 2 experimentation. The optimal tuned values obtained from the tool are listed in Table 8. These values could be used in Tool-II to plot the closed loop response when needed. Here, these values were used to control the liquid level process. The liquid level response is shown in Figures 11 and 13. As mentioned earlier the overall performance was compared with other methods in Table 5.

Table 6. Iteration and swarm size used in the optimization algorithm.

Parameter	GSO	GA	PSO	Firefly	GWA
Maximum Iteration	125	30	30	50	70
Swarm Size	100	50	70	60	50

Table 7. Additional parameters for galactic swarm optimization (GSO), PSO and firefly.

Algorithm	Parameter
PSO	Inertia coefficient = 1
	Damping coefficient = 0.99
	Acceleration coefficients $c1 = 1.5, c2 = 2$
Firefly	$\alpha = 0.2, \beta = 2, \gamma = 1$
GSO	Acceleration coefficients $c1 = c2 = c3 = c4 = 2.05$

Table 8. Example 1: Tuning parameters with an optimization tool.

Algorithm	θ_R	θ_T	θ_D	θ_A
GSO	0.9597	0.4782	0.8880	0.0356
GA	0.9581	0.00103	0.67	0.0211
PSO	0.8931	0.2532	0.5945	0.03
Firefly	0.9663	0.0001	0.519	0.033
GWA	0.9470	0.0015	0.914	0.0016

5.1.1. Example 1: Closed Loop Stability Analysis

The stability of the closed loop liquid level control and pressure control with the RTD-A controller was checked according to Ogunnaike et al. and Kariwala et al. [11,12]. The stability analysis concept in the state variable form was discussed in Section 2.1.1. It is necessary that the roots of the characteristics equation in (17) should lie within the unit circle for closed loop stability. The roots of the liquid level control characteristic equation in the state variable form were calculated using (17) for different tuning methods and listed in Table 9. The location of roots in the z-plane was plotted and shown in Figure 19. Since all the roots were located inside the unit circle, the system was stable for all tuning methods.

Table 9. Example 1: Roots with different tuning methods.

Tuning Method	Ogunnaike	Kariwala	GSO	GA	PSO	FF	GWA
Closed Loop Roots	0.8300	0.0793	0.5170	0.4389	0.5962	0.6730	0.6754
	0.3348	0.0500	0.4072	0.5590	0.5705	0.7277	0.7287
	0.0704	0.0407	0.0330	0.0327	0.0909	0.0432	0.0531
	0.0377	0.0364	0.0458	0.0056	0.0371	0.0307	0.0353
	0.0365	0.0365	0.0365	0.0054	0.0365	0.0365	0.0365

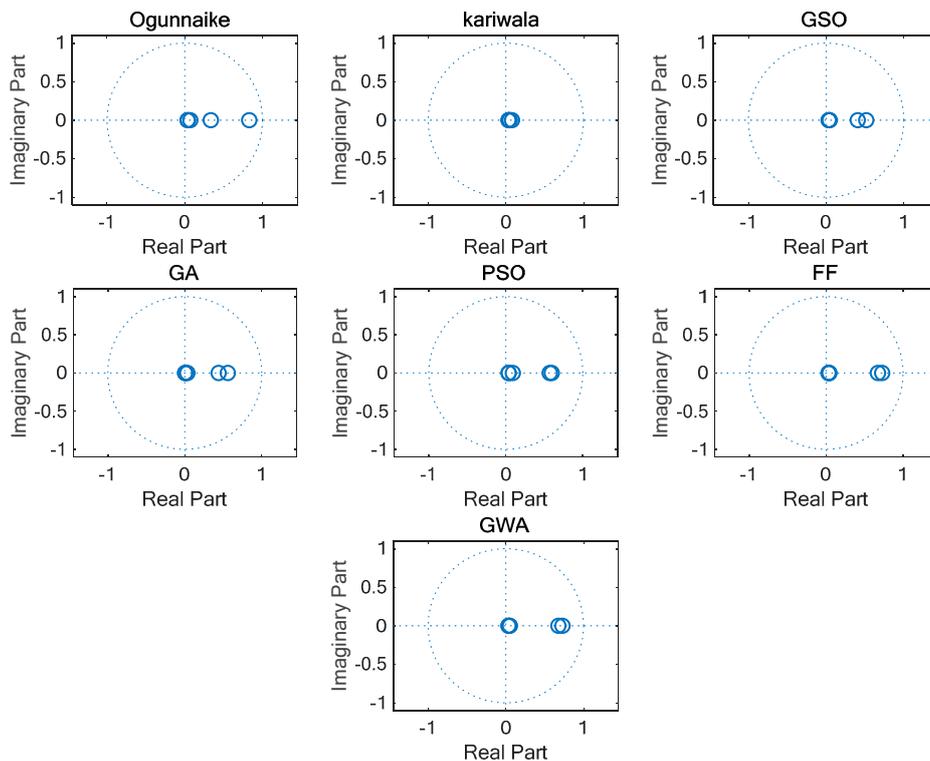


Figure 19. Example 1: z-plane plot for the roots of the characteristic equation.

The stability could also be analyzed on the basis of the Jury’s stability criterion discussed in 2.1.2. The coefficients c_0 and c_n were calculated using (21) and (22). The value of the polynomial $P(z)$ when $z = 1$ and $z = -1$ was calculated according to (18). The θ_{ref} value also was calculated to check the stability in terms of θ_R using (25). The calculated values are tabulated in Table 10. The value of $c_n = 51,164$, $\theta_{ref} = 0.07$ and the order of $P(z)$ was an even number. The necessary conditions in (20), (23), (24) and (25) were checked for stability. It was found that all the conditions were satisfied for closed loop stability for the calculated tuning values. Therefore, the designed RTD-A controller ensured closed loop stability in liquid level control. A similar analysis was done for pressure control in Section 5.2.1.

The open loop gain margin (GM) of the liquid level process was 4.1 dB and the closed loop GM with PID was 5.86 dB. Thus the closed loop stability with PID would be ensured by choosing the proportional gain value less than open loop gain margin value.

Table 10. Example 1: Jury’s stability test for liquid level control ($c_n = 5.1164e + 04$ and $\theta_{ref} = 0.07$).

Tuning Method	Ogunnaike	Kariwala	GSO	GA	PSO	FF	GWA
θ_R	0.5	0.95	0.9597	0.9581	0.8931	0.9663	0.947
$ c_0 $	2.7507×10^4	2.75×10^3	2.217×10^3	2.305×10^3	5.881×10^3	1.854×10^3	2.9158×10^3
$P(z)$ when $z = 1$	40.5457	4.0546	3.2680	3.3977	8.6687	2.7328	4.2978
$P(z)$ when $z = -1$	1.1878×10^7	1.54×10^6	1.318×10^6	1.3548×10^6	2.8480×10^6	1.1665×10^6	1.6098×10^6

5.2. Example 2: Pressure Control System

A pressure control station is shown in Figure 20. The FOPDT model for pressure process was found using the black box modeling technique [38]. The corresponding first order process model is given in (38):

$$G(s) = \frac{1.01}{18.42s + 1} e^{-0.5s}. \tag{38}$$

The servo performance of the real-time system model without uncertainty was analyzed with the proposed tool and PID controller. The implementation of the controller with the simulation tool on pressure control is shown in Figure 21. The tank pressure $P(t)$ was to be controlled by adjusting the outlet flow air flowrate $Q_{out}(t)$. All four parameters were tuned using the Tool-II and -IV. The tuned values are listed in Tables 11 and 12 respectively. The control objective of the optimal control problem was to reduce the error between the desired pressure and actual pressure. The IAE cost function given in (37) was used to determine the optimal tuning values of RTD-A.

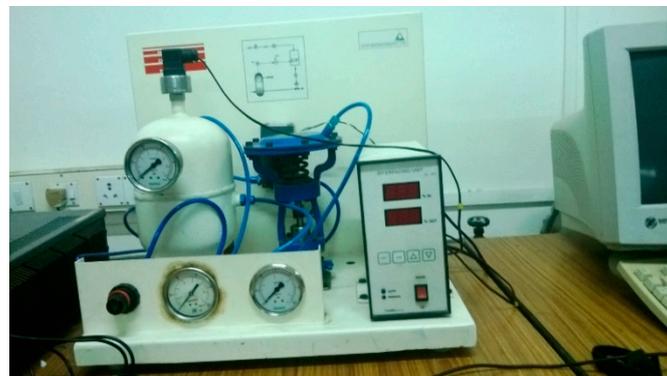


Figure 20. Pressure control station [38].

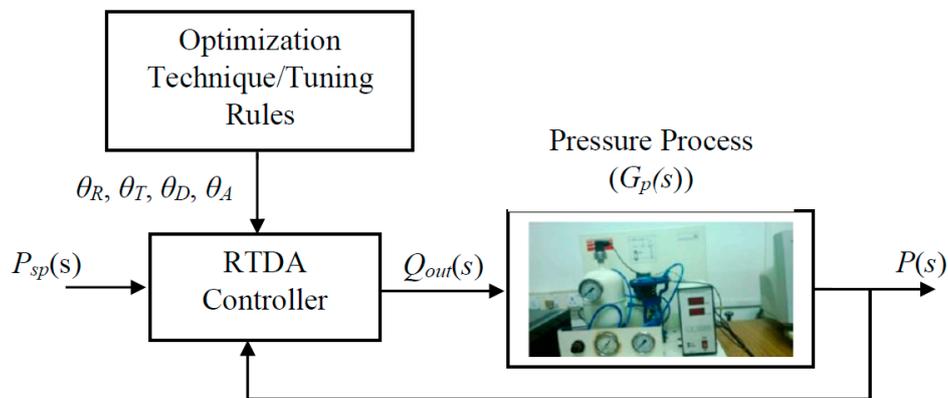


Figure 21. Pressure control station with the RTD-A controller.

Table 11. Example 2: Tuning parameters with tuning methods.

Algorithm	θ_R	θ_T	θ_D	θ_A
Ogunnaike	0.5	0.9876	0.5	0.003
Kariwala	0.66	0.98	0.866	0.006

Table 12. Example 2: Tuning parameters with an optimization tool.

Algorithm	θ_R	θ_T	θ_D	θ_A
GSO	0.5121	0.0036	0.176	0.0352
GA	0.4620	0.00101	0.2450	0.0255
PSO	0.4097	0.8845	0.8346	0.0196
Firefly	0.7398	0.7424	0.3032	0.0044
GWA	0.4356	0.0083	0.3270	0.0163

During this optimal tuning process, an output disturbance was also considered with $\pm 10\%$ random variation.

The PID controller tuning values obtained from the auto tune function in MATLAB were $K_P = 1.373$, $K_I = 0.143$, $K_D = -0.178$ and the filter co-efficient (T_f) = 0.115.

5.2.1. Example 2: Closed Loop Stability Analysis

The roots of the characteristics equation of pressure control process were calculated for different tuning methods using (17) and listed in Table 13. The location of roots in the z-plane is shown in Figure 22. It was observed that all the roots were located inside the unit circle. As mentioned in Section 5.1.1, the Jury’s stability test was performed on the pressure control process. The value of $P(z)$ for $z = 1$ and $z = -1$, θ_{ref} , coefficients c_0 and c_n were calculated using (18), (21), (22) and (25). The values are tabulated in Table 14. The calculated values were checked for stability necessary conditions in (20) and (23)–(25). The value of $c_n = 1.5929e + 07$, $\theta_{Ref} = 0.3295$ and the order of $P(z)$ was an odd number.

Table 13. Example 2: Closed loop roots with different tuning methods.

Tuning Method	Ogunnaike	Kariwala	GSO	GA	PSO	FF	GWA
Closed Loop Roots	0.6032	0.7360	0.2247	0.3681	0.1985	0.5930	0.4940
	0.3840	0.2621	0.6452	0.7198	0.4132	0.2589	0.7538
	0.0108	0.0164	0.2302	0.2725	0.0837	0.1515	0.2980
	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056
	0.0054	0.0054	0.0054	0.0054	0.0054	0.0054	0.0054

Table 14. Example 2: Jury’s stability test for pressure control ($c_n = 1.5929 \times 10^7$ and $\theta_{Ref} = 0.3295$).

Tuning Method	Ogunnaike	Kariwala	GSO	GA	PSO	FF	GWA
θ_R	0.5	0.66	0.5121	0.462	0.4097	0.7398	0.4356
$ c_0 $	1.1878×10^7	8.0769×10^6	1.159×10^7	1.278×10^7	1.4023×10^7	6.1812×10^6	1.34×10^7
$P(z)$ when $z = 1$	277.8394	188.9308	271.1157	298.9552	328.0172	144.5876	313.6251
$P(z)$ when $z = -1$	-2.092×10^{10}	-1.418×10^{10}	-2.04×10^{10}	-2.25×10^{10}	-2.472×10^{10}	-1.082×10^{10}	-2.36×10^{10}

It was found that the RTD-A tuning parameter satisfied all the necessary conditions for the closed loop stability. It is guaranteed that pressure control will be stable with the RTD-A controller.

The gain margin of the open loop pressure system was 35.3 dB and the closed loop gain margin with PID was 24.1 dB. The closed loop stability with PID is guaranteed if the proportional gain is chosen less than the maximum open loop gain margin value.

The setpoint 53 psi was considered for the pressure control process. The RTD-A and PID controller tuned values were used to control the pressure process. The servo response with RTD-A for a different tuning method and PID is shown in Figure 23. The pressure response with RTD-A using optimization algorithms is shown in Figure 24.

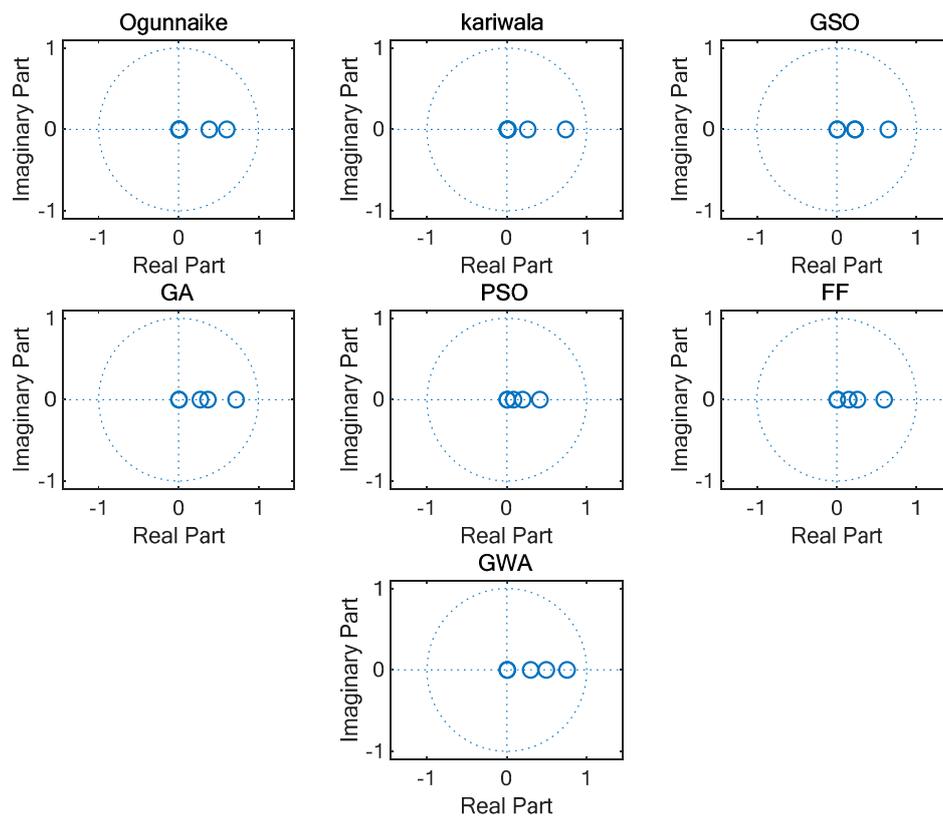


Figure 22. Example 2: z-plane plot for the roots of the characteristic equation.

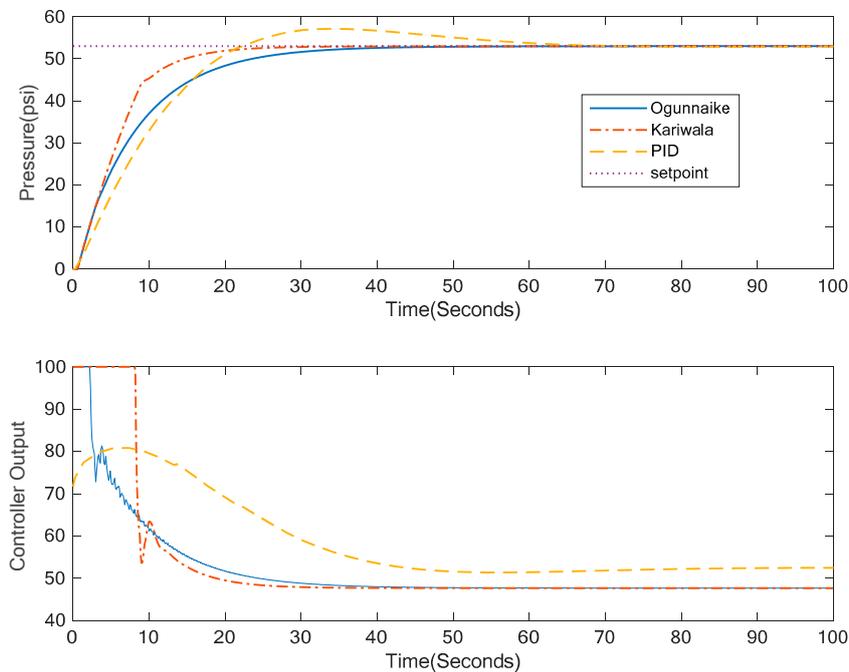


Figure 23. Comparison of the servo response for the pressure control station with two tuning rules and PID.

The execution time of RTD-A and PID controller implementation for the servo response of pressure control was 0.047388 s and 1.643077 s respectively when the simulation time was 100 s and sampling time was 0.1 s.

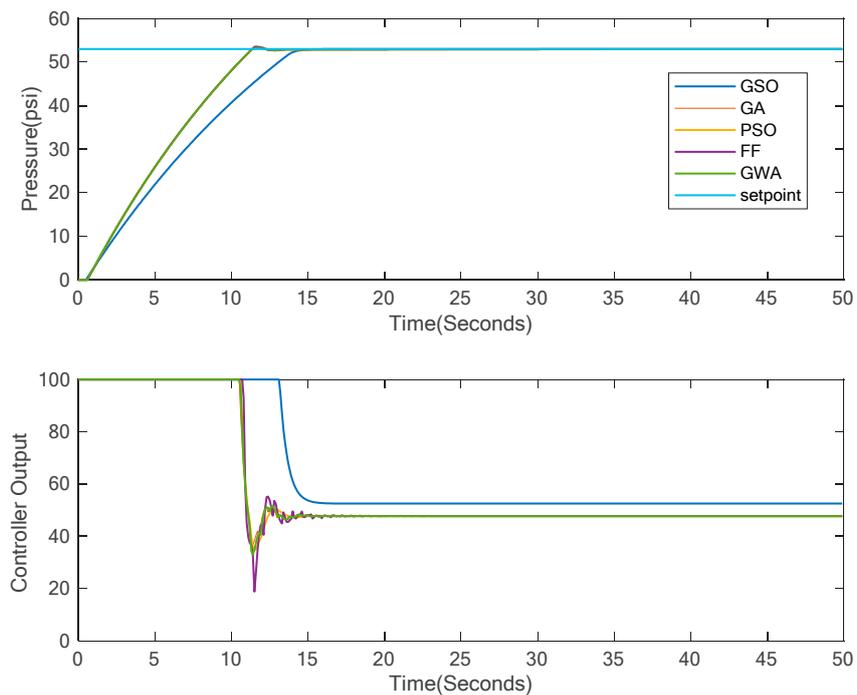


Figure 24. Comparison of the servo response for the pressure control station with optimization algorithms.

The regulatory response of the pressure control process without optimally tuned values and with optimally tuned values is shown in Figures 25 and 26 respectively. The setpoint 50 psi was considered in the investigation of disturbance rejection ability. The 10% of setpoint was applied as the disturbance to the process at 50 s in Figure 25 and at 25 s in Figure 26. The disturbance rejection ability of the optimally tuned RTD-A controller was better than the other method.

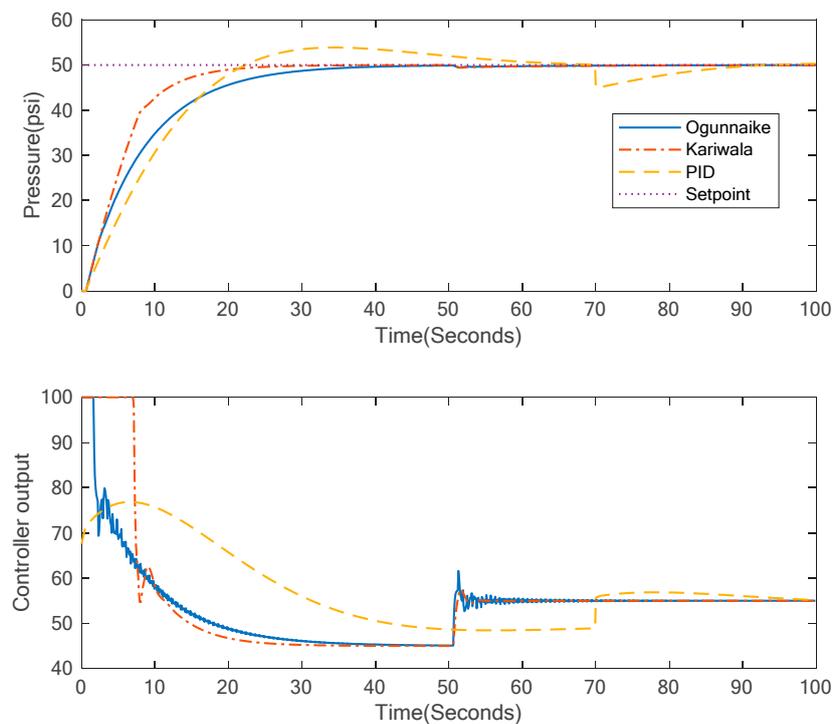


Figure 25. Example 2: Regulatory response for the pressure control station with tuning rules and PID.

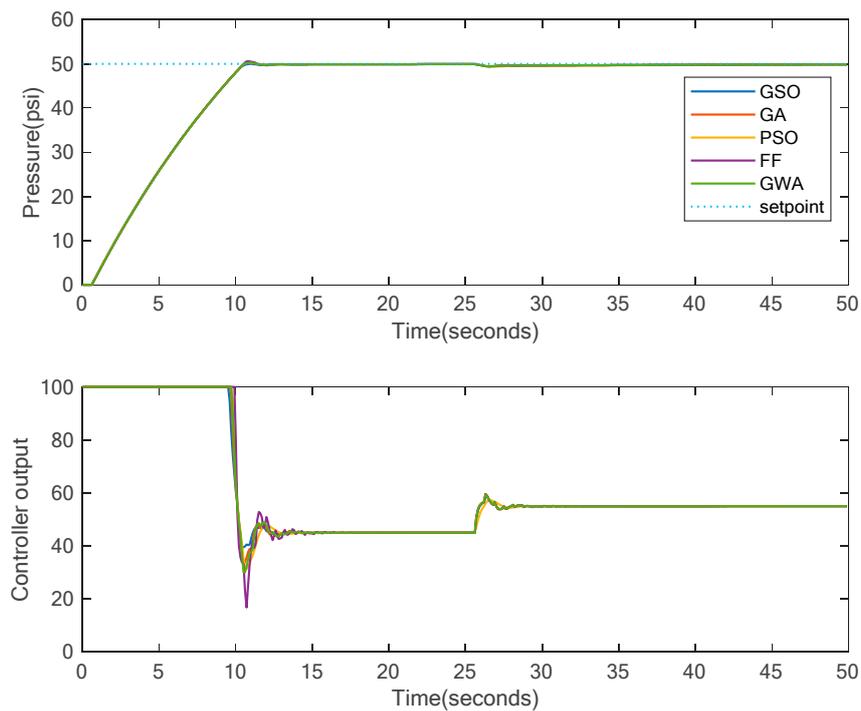


Figure 26. Example 2: Regulatory response for the pressure control station with optimization algorithms.

The overall performance comparison for example 2 is done in Table 15. There was no noticeable difference in the performance of pressure control among the optimally tuned responses excluding GSO. The pressure process works better with the optimal tuning method in terms of IAE and other time domain performance measures. The response with the Kariwala tuning method took longer for the settling time and less rise time than the Ogunnaike tuning. A steady state error was almost zero in all the methods. There was a minimum overshoot in the response with the optimal tuned values, but the percentage overshoot in the Ogunnaike and Kariwala method was absolutely zero. The performance with RTD-A was better than PID in pressure control.

Table 15. Example 2: Performance comparison of different tuning methods and PID.

Specifications	PID	Ogunnaike	Kariwala	GSO	GA	PSO	Firefly	GWA
IAE	570.6	452.1788	327.21	291.1297	291.0047	291.6691	291.2998	291.0940
Rise Time (s)	15.47	17.53	10.4164	8.4854	8.4855	8.4845	8.4855	8.4854
Settling Time (s)	57.78	32.09	47.77	11.0437	11.0425	11.0412	11.0424	11.0424
Overshoot (%)	7.9623	0	0	0	0.3746	0.7189	1.0824	0.5502
Steady State Error (e_{ss})	0.0756	0.0016	0.0015	0.0182	0.0175	0.0220	0.0178	0

Overall, in a liquid level control system, the firefly optimization tuned response was better than other methods. The genetic algorithm tuned response was better than other methods excluding percentage overshoot in the pressure control process.

6. Conclusions

The RTD-A controller is evolving very fast and can be used as an alternative for the classical PID controller. Since RTD-A has an independent tuning parameter for robustness it can handle the systems with uncertain behavior. The proposed simulation tool is the very basic implementation of the RTD-A controller. This tool can be used to approximate an over-damped higher order transfer function into a FOPDT system. Therefore, the proposed tool can be implemented on any higher order processes with or without dead time. The industrial processes can adopt this RTD-A controller with

more flexibility. In the case study, the proposed tool had experimented on an interacting two tank liquid level process and a pressure control application. The optimization tool was tested in both cases and obtained the optimized results for the controller parameters under different optimization algorithms. In order to understand the effect of the changes in each tuning parameter on the plant output, a separate tool was developed for the RTD-A tuning parameter analysis and it was experimented on the interacting liquid level process. The closed loop stability of the liquid level control and pressure control was checked in the state variable form and also using the Jury's stability criterion. It was found that the controller guaranteed the closed loop stability. Finally, the closed loop process performance with the optimal tuned RTD-A controller using the firefly algorithm was better than other methods in terms of IAE. The Kariwala tuning rule was also observed to be superior in terms of the settling time and overshoot performance measure in the liquid level control. Similarly, the optimization algorithms-based controller gave better performance in terms of IAE and other time domain specifications while controlling the pressure process station. The limitation of the proposed tool, it is not suitable for experimenting with the unstable processes and the multi input–multi output processes. It was also assumed that the process behavior could be adequately described by a FOPTD model. In the future, the proposed tool could be enhanced by adding other optimization algorithms, frequency domain analysis and multi-input multi-output process controlling ability. The facility could be added in the proposed tool to study the effect of optimization algorithm's random parameters on the process output. The proposed tool "RTDA_Toolbox" could be downloaded from the link in https://drive.google.com/open?id=14GcTMwft7pn3bLcxOTvfX7_MNIDdIBLu.

Author Contributions: V.B. proposed the simulation tool concept and implemented the model order reduction and tuning of controller for different processes and wrote the paper; S.U. did the controller performance analysis and P.A. proposed the optimal tuning and edited the paper.

Funding: This research was funded by VIT, Vellore.

Acknowledgments: The authors like to express their sincere gratitude to the Process control Laboratory and School of Electrical Engineering, VIT, Vellore for giving opportunity to do the project in this area. Finally authors acknowledge the Renewable Energy Lab, Prince Sultan University, Riyadh for providing technical guidance for the execution of these research activities.

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

References

1. Ruz, M.L.; Garrido, J.; Vazquez, F.; Morilla, F. Interactive Tuning Tool of Proportional-Integral Controllers for First Order Plus Time Delay Processes. *Symmetry* **2018**, *10*, 569. [CrossRef]
2. 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]
3. Dai, A.; Zhou, X.; Liu, X. Design and Simulation of a Genetically Optimized Fuzzy Immune PID Controller for a Novel Grain Dryer. *IEEE Access* **2017**, *5*, 14981–14990. [CrossRef]
4. Bagyaveereswaran, V.; Suryawanshi, S.; Arulmozhivarman, P. RTD-A controller toolbox for MATLAB. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, Tamilnadu, India, 21–22 April 2017.
5. Ender, D.B. Process Control Performance: Not as Good as You Think. *Control Eng.* **1993**, *40*, 180–190. Available online: https://www.academia.edu/1340822/Process_control_performance_Not_as_good_as_you_think (accessed on 20 May 2019).
6. Chen, Q.; Tan, Y.; Li, J.; Mareels, I. Decentralized PID Control Design for Magnetic Levitation Systems Using Extremum Seeking. *IEEE Access* **2018**, *6*, 3059–3067. [CrossRef]
7. Desborough, L.; Miller, R. Increasing customer value of industrial control performance monitoring -Honeywell experience. In *AICHE Symposium Series 326*; American Institute of Chemical Engineers: New York, NY, USA, 2002; pp. 172–192.
8. Kristiansson, B.; Lennartson, B. Robust and optimal tuning of PI and PID controllers. *IEEE Proc. Control Theory Appl.* **2002**, *149*, 17–25. [CrossRef]
9. Hugo, A. Limitations of Model Predictive Controllers. *Hydrocarb. Process.* **2000**, *79*, 83–88.

10. Kaya, I. IMC based automatic tuning method for PID controllers in a Smith predictor configuration. *Comput. Chem. Eng.* **2004**, *28*, 281–290. [[CrossRef](#)]
11. Mukati, K.; Ogunnaike, B. Stability analysis and tuning strategies for a novel next generation regulatory controller. In Proceedings of the 2004 American Control Conference, Boston, MA, USA, 30 June–2 July 2004; Volume 5, pp. 4034–4039.
12. Ogunnaike, B.A.; Mukati, K. An alternative structure for next generation regulatory controllers: Part I: Basic theory for design, development and implementation. *J. Process Control* **2006**, *16*, 499–509. [[CrossRef](#)]
13. 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](#)]
14. Sendjaja, A.Y.; Ng, Z.F.; How, S.S.; Kariwala, V. Analysis and Tuning of RTD-A Controllers. *Ind. Eng. Chem. Res.* **2011**, *50*, 3415–3425. [[CrossRef](#)]
15. Aleksei, T.; Eduard, P.; Juri, B. A flexible MATLAB tool for optimal fractional-order PID controller design subject to specifications. In Proceedings of the 31st Chinese Control Conference, Hefei, China, 25–27 July 2012; pp. 4698–4703.
16. Currie, J. jMPC Toolbox v3. 11 Users Guide. In *Industrial Information & Control Centre*; AUT University: Auckland, New Zealand, 2011.
17. Bagyaveereswaran, V.; Mathur, T.D.; Gupta, S.; Arulmozhivarman, P. Performance comparison of next generation controller and MPC in real time for a SISO process with low cost DAQ unit. *Alex. Eng. J.* **2016**, *55*, 2515–2524. [[CrossRef](#)]
18. Morales, D.C.; Jimenez-Hornero, J.E.; Vazquez, F.; Morilla, F. Educational Tool for Optimal Controller Tuning Using Evolutionary Strategies. *IEEE Trans. Educ.* **2012**, *55*, 48–57. [[CrossRef](#)]
19. Bang, H.; Lee, Y.S. Implementation of a Ball and Plate Control System Using Sliding Mode Control. *IEEE Access* **2018**, *6*, 32401–32408. [[CrossRef](#)]
20. Holland, O.T.; Marchand, P. *Graphics and GUIs with MATLAB*; Chapman and Hall/CRC Press: Boca Raton, FL, USA, 2002.
21. 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](#)]
22. Fan, Y.; Shao, J.; Sun, G. Optimized PID Controller Based on Beetle Antennae Search Algorithm for Electro-Hydraulic Position Servo Control System. *Sensors* **2019**, *19*, 2727. [[CrossRef](#)] [[PubMed](#)]
23. Wang, R.; Tan, C.; Xu, J.; Wang, Z.; Jin, J.; Man, Y. Pressure Control for a Hydraulic Cylinder Based on a Self-Tuning PID Controller Optimized by a Hybrid Optimization Algorithm. *Algorithms* **2017**, *10*, 19. [[CrossRef](#)]
24. Bilal, H. Abed-alguni, Island-based cuckoo search with highly disruptive polynomial mutation. *Int. J. Artif. Intell.* **2019**, *17*, 57–82.
25. Soares, A.; Râbelo, R.; Delbem, A. Optimization based on phylogram analysis. *Expert Syst. Appl.* **2017**, *78*, 32–50. [[CrossRef](#)]
26. Shams, M.; Rashedi, E.; Dashti, S.M.; Hakimi, A. Ideal gas optimization algorithm. *Int. J. Artif. Intell.* **2017**, *15*, 116–130.
27. Precup, R.E.; David, R.C. *Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems*, 1st ed.; Elsevier: Oxford, UK, 2019.
28. Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. *J. Process. Control.* **2003**, *13*, 291–309. [[CrossRef](#)]
29. Leite, M.S.; Araújo, P.J. *Relay Methods and Process Reaction Curves: Practical Applications*; In Tech Open: Rijeka, Croatia, 2012; pp. 248–258.
30. Jose, A. Romagnoli, and Ahmet Palazoglu. In *Introduction to Process Control*; CRC Press: Boca Raton, FL, USA, 2005.
31. Srinivasan, K.; Anbarasan, K. Fuzzy scheduled RTD-A controller design. *ISA Trans.* **2013**, *52*, 252–267. [[CrossRef](#)] [[PubMed](#)]
32. Muthiah-Nakarajan, V.; Noel, M.M. Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Appl. Soft Comput.* **2016**, *38*, 771–787. [[CrossRef](#)]
33. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [[CrossRef](#)]

34. Visioli, A. Optimal tuning of PID controllers for integral and unstable processes. *IEEE Proc. Control Theory Appl.* **2001**, *148*, 180–184. [[CrossRef](#)]
35. Yang, X.S. Firefly algorithms for multimodal optimization. In Proceedings of the International Symposium on Stochastic Algorithms, Sapporo, Japan, 26–28 October 2009; pp. 169–178.
36. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimize. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
37. Ogata, K. *Control System Dynamics*, 4th ed.; Pearson Education Inc.: Upper Saddle River, NJ, USA, 2004; pp. 155–156.
38. Bayaveereswaran, V.; Wilfred, K.J.N.; Sreeraj, S.; Vijay, B. System Identification from step input using Integral equation. *Int. J. Appl. Eng. Res.* **2013**, *8*, 2165–2169.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).