J.Natn.Sci.Foundation Sri Lanka 2023 51 (1): 21 - 35
DOI: http://dx.doi.org/10.4038/jnsfsr.v51i1.10844

RESEARCH ARTICLE

Process Control Engineering

Temperature control in an exothermic continuous stirred tank reactor

VA Rani*, G Guna, D Prabhakaran and M Thirumarimurugan
Department of Chemical Engineering, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu 641014, India.

Submitted: 27 December 2021; Revised: 15 April 2022; Accepted: 24 June 2022

Abstract: A continuous stirred tank reactor (CSTR) is a batch reactor fortified with an impeller or additional mixing device to provide resourceful mixing. In chemical engineering, the name CSTR is often used to describe an idealised agitated tank reactor used to model manoeuvre variables necessary to achieve a specified output. Most chemical plants have a process involving a continuous stirred tank reactor (CSTR), and it has more nonlinearity in real-world implementation due to disturbances like change in surrounding temperature, non-uniformity in mixing, and change in the temperature of the coolant. The aim of the work is to study the dynamic behaviour of a continuous stirred tank reactor with coolant flow rate as input and reactor temperature as output and to design a suitable controller to control the temperature of the continuous stirred tank reactor by conducting an exothermic reaction in real-time. A continuous stirred tank reactor was modelled with the help of a transfer function model in the MATLAB environment. For controlling the temperature of the reactor fluid, the design of proportional integral derivative, proportional integral derivative (PID) – particle swarm optimization (PSO), proportional integral derivative (PID) – artificial bee colony optimization (ABC) and model predictive control (MPC) controller were carried out. The simulation results show that model predictive control has better tracking performance compared to conventional PID, PID–PSO or PID–ABC.

Keywords: Controller tuning, exothermic reaction, optimization algorithms, temperature control, transfer function.

INTRODUCTION

Continuous stirred tank reactors are important in industries such as construction materials, biofuels, pharmaceuticals and wastewater treatment (Rani *et al*., 2020). The reaction in a CSTR can be exothermic or endothermic. On the occurrence of an exothermic reaction, heat is liberated. Under such conditions, a coolant stream will be transported through the jacket that surrounds the reactor to remove the extra heat. On the other hand, if an endothermic reaction occurs in the system, a heating medium has to be provided through the jacket for regulating the temperature in the reactor. A reactor operating at a constant temperature is called an isothermal reactor (Ahmed *et al*., 2016). When exothermic or endothermic reactions take place, heat is either liberated or absorbed. This leads to a change in the temperature of the reactor at various points in time. Since the temperature of the reaction in the reactor is not constant throughout, it is called a non-isothermal reactor. Regulating the temperature in the non-isothermal CSTR is an important control problem. Various controller schemes such as classical controllers (Dey & Roy, 2014), optimal controllers (Da & Shu-Cai, 2018), and robust controllers (Monika Bakošová *et al*., 2005) are used to overcome the temperature control problem. In order to achieve better performance of the controller, there is a need for proper adjustment of the parameters of the controller. The procedure for adjustment of the controller parameters is called 'tuning' or 'design' of the controller. The most widely accepted practical tuning methods are the Ziegler and Nichols (1942) method and Cohen and Coon (1953) method (Agarwal *et al*., 2015). These methods are compatible only with an open loop, stable and single input single output (SISO) system, and it is not suitable for a complex and highly unstable process. Hence, tuning a higher order for a complex process is very difficult using conventional approaches.

* Corresponding author (anusharani151@gmail.com; https://orcid.org/0000-0001-7290-9086)

Thus, the control society has shifted its focus to stochastic approaches, which offer a heuristic searching method for the tuning mechanism. The optimal controller parameters, which are obtained from the evolutionary optimization techniques, are fed into the controller. Then, the closed loop system is simulated to find the desired response. From the output response of the process, the objective function is evaluated based on the set point value, and it is processed by the evolutionary optimizer. There are numerous evolutionary optimization techniques used in the literature to tune the parameters of a PID controller, such as particle swarm optimization (PSO) (Bonyadi & Michalewicz, 2017), bacterial foraging optimization (BFO) (Das *et al*., 2009) and atom search optimization (Zhao *et al*., 2019). Apart from the optimization techniques, the formulation of the objective function is the significant part of the optimization. The most widely employed objective function is the summation of errors between the set point and the actual output. In PID controller design methods, the most common performance criteria are integrated absolute error (IAE), integrated time weight square error (ITWSE), integrated squared error (ISE), and mean square error (MSE). These four integral performance criteria have their own advantages and disadvantages.

In Swapnadeep and Lillie (2017), the temperature control of CSTR was performed using the PID tuned by the Ziegler-Nichols method, genetic algorithm (GA), and the PSO. The mathematical model of the CSTR was considered, and the objective function used for the optimization was the ISE. The transient and steady-state analysis was carried out on the closed loop system, and it was found that the PID-PSO performed better while ISE was used as the objective function. In Thulasi Dharan *et al*. (2017), a multi input multi output (MIMO) model of the CSTR was considered for temperature control. Here the objective function used was the ITAE. The PSO had better closed-loop performance compared to the other optimization algorithms. Masilamani *et al*. (2015) computed a model predictive controller for the temperature and concentration control of the CSTR. Here the state space model of the CSTR was considered for the implementation of the controller. The MPC toolbox of the MATLAB environment was used to provide the plant inputs and initial values to the state space model of the CSTR. The MPC predicts the value of the temperature and concentration and controls the same. In this study, the model required for the control of the CSTR was obtained from the experimentation. The performance criterion obtained from various models of CSTR is to be compared with the experimental model. The novelty of the study lies in the usage of the experimental model for controller implementation. From the literature, it is seen that the objective function mainly used was the ITAE and ISE. In this study, a new objective function, which is the combination of the ITAE and the overshoot, was used.

MATERIALS AND METHODS

Process description

Experimental setup

A non-isothermal CSTR consists of inlet streams, an outlet stream, and the jacket through which the coolant is passed in order to remove the heat generated due to the exothermic reaction inside the reactor. The temperature of the reaction mixture, the concentration of the reaction mixture and the level of fluid inside the tank are the important process variables. Actual values of these variables are measured using appropriate instruments and controlled to the desired value using controllers. The setup is pictured in Figure 1 and consists of a resistance temperature detector (RTD), variable frequency drive (VFD), motor, pumps, programmable logic controller (PLC), emergency shutdown, stirrer, and tanks. The reactor vessel is insulated with glass wool to prevent the heat from escaping into the surrounding. The capacity of the reactor vessel is 50 L, the hold up inside the reactor is 25 L, and the reactants are stored in tanks of volume 25 L. From the tanks, the reactants can be transferred to the reactor using a pump (12 Volt DC, maximum pressure 0.65 MPa) which operates at a flow rate of 4 LPM. Ball valves are used to reduce the flow of reactant. The equipment is made of steel and coated with anti-corrosion paint. The stirrer speed is kept constant. In this process, the input variable is the flow rate of coolant, and the output variable is the temperature of the reactor fluid. The range of the variables is tabulated in Table 1.

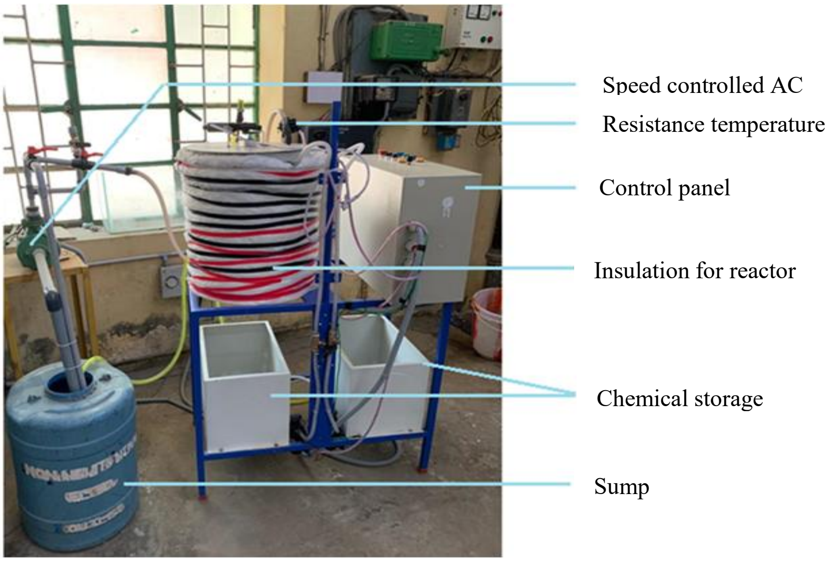

Figure 1: Experimental set up of CSTR

Table 1: Process variables and their value range

Variables	Symbols	Values range
Feed flow rate of each reactant	F_1, F_2	0.75 LPM
Jacket feed flow rate	$F_j(t)$	0–2 LPM
Inlet temperature of reactor fluid	T_i	30 ˚C
Outlet temperature of reactor fluid	$T_{io}(t)$	35–50 ˚C
Jacket inlet temperature	T_{ji}	29 ˚C
Jacket outlet temperature	T_{jo}	30–45 ˚C

Study of the setup with chemicals and reaction kinetics

In this study, the reaction between a strong acid, hydrochloric acid (HCl), and a strong base, sodium hydroxide (NaOH), is considered.

$$HCl_{(g)} + H_2O_{(l)} \rightarrow H_3O^+_{(aq)} + Cl^-_{(aq)} \quad \ldots(1)$$

$$NaOH(s) + H_2O_{(l)} \rightarrow Na^+_{(aq)} + OH^-_{(aq)} \quad \ldots(2)$$

The neutralization reaction is an exothermic reaction, and it is carried out inside the reactor:

$$NaOH_{(aq)} + HCl_{(aq)} \rightarrow NaCl_{(aq)} + H_2O_{(l)} \quad \ldots(3)$$

Design of conventional PID controller

Fundamentals and theory of PID

A block diagram of a simple closed-loop system consisting of a plant and a controller with unity feedback is diagrammed in Figure 2. The purpose of the system is to keep the process output (Y) closest to the desired output (Y_d) regardless of disturbances. This is achieved by manipulating the process input (U) through the controller (Dong *et al.*, 2021). The performance of the closed loop system is defined by the performance criteria of mean square error (MSE), rise time (T_r), settling time (T_s) and steady state error (E_{ss}) of the transient response.

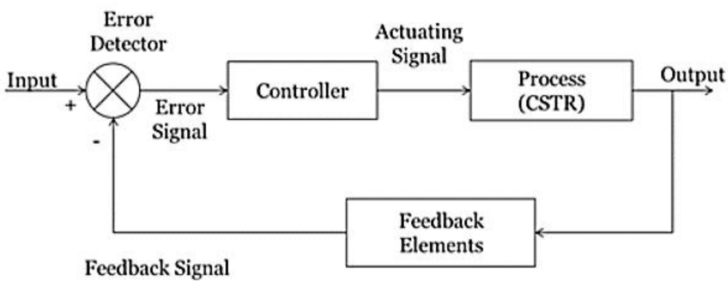

Figure 2: PID controller block diagram

The PID controller is commonly used in industries because of its simplicity in implementation, and it is a linear feedback controller whose control action is based on the error signal e(t), which is a difference between the set point and the model output value. Mathematically the PID controller is given as:

$$(t) = K_p\,(t) + K_i \int (t)\,dt + K_d\,(t)\,/dt \qquad \ldots(4)$$

where K_p = Proportional gain
K_i = Integral gain
K_d = Derivative gain

The proportional gain of the controller reduces error responses to disturbances (Aslam & Kaur, 2011). The integral of the error eliminates steady-state offsets, and the derivative of the error dampens the dynamic response. Thereby, the stability of the system is improved.

Tuning of PID

Tuning of the independent/tunable parameters is an important step in controller design. In this study, MSE is chosen as the objective function to achieve minimal control errors. It squares the error to remove the negative error components and discriminates between over-damped and under-damped systems.

$$MSE = \frac{1}{2}\int_{t=0}^{T} e^2(t)\,dt \qquad \ldots(5)$$

Lower values of the objective function will give a better closed-loop performance. During the tuning of the controller, MSE, rise time (T_r), settling time (T_s), and steady-state error (E_{ss}) are taken as performance indices.
Rise time (T_r) is the time required for the response to rise from 10 to 90% or 0 to 100% of its final value. For an under-damped system, a second-order system with 0 to 100% rise time is commonly used.
For an over-damped system, 10 to 90% rise time is commonly used.

$$Rise\ Time\ (T_r) = \frac{\pi-\beta}{\omega d},\ \beta = \cos^{-1}\varepsilon \qquad \ldots(6)$$

where ε is the damping ratio, and ω_d is the damped natural frequency.
Settling time (T_s) is the time required for the response curve to reach and stay within a range about the final value of size specified by an absolute percentage of the final value (usually 5 to 2%).

$$Settling\ Time\ (Ts) = \frac{4}{\varepsilon\omega n} \qquad \ldots(7)$$

where ε is the damping ratio, and ω_n is the natural frequency.

Steady-state error (E_{ss}) is the difference between the measured constant output and the input constituents or a set point in a steady-state condition.

Ziegler-Nichols open loop tuning method

In this study, the conventional Ziegler-Nichols method has been used to find the approximate values of the parameters of the PID controller.

- We first plot the Bode diagram for the final control element, the process, and the measuring element in series, $G_i \cdot G_p \cdot H$ (jω).
- At the crossover frequency, the overall gain is A. As per the Bode criterion, the gain of a proportional controller causing the system to be on the verge of instability is l/A.
- We define this quantity to be the ultimate gain K_u.

$$\text{Thus } K_u = 1/A \qquad \ldots(8)$$

- The ultimate period Pu is defined as the period of the sustained cycling that would occur if a proportional controller with gain K_u, were used.

$$Pu = 2\pi/\omega_{co} \text{ time/cycle} \qquad \ldots(9)$$

- The value of the controller gains is calculated as in Table 2.

Table 2: Ziegler-Nichols controller settings

Type of control	$G_c(s)$	K_c	τ_I	τ_D
Proportional (P)	K_c	$0.5K_u$		
Proportional-integral (PI)	$K_c\,(1 + \frac{1}{\tau_I s})$	$0.45K_u$	$\frac{Pu}{1.2}$	
Proportional-integral-derivative (PID)	$K_c\,(1 + \frac{1}{\tau_I s} + \tau_D s)$	$0.6K_u$	$\frac{Pu}{2}$	$\frac{Pu}{8}$

Tuning of PID controller using evolutionary algorithms

In Li *et al*. (2015), a hybrid algorithm which is the combination of the PSO and the ABC, was proposed for the optimization of the higher dimensional objective functions and to improve the performance of the optimization algorithm. The algorithm uses the local search space of the PSO and the global search space of the ABC. This hybrid algorithm was applied to various benchmark functions and was compared to other optimization algorithms, such as the ABC and PSO. Saravanakumar *et al*. (2017) used the state transition algorithm (STA) for the tuning of the controller used in the MIMO distillation column. The STA is a heuristic random search algorithm working on the principle of state transition. The objective function used for this optimization is the ITAE and the IAE. From the simulation results, it was understood that the STA is a powerful algorithm with good search capability and improves the performance of the system. In this study, two methods are exploited for obtaining the tunable parameters of the controllers, namely, the conventional tuning methods and the evolutionary algorithms such as PSO and ABC. The evolutionary methods require an objective function that has to be minimized in order to obtain the optimal values of the tunable parameters. In this work, MSE is used as the objective function. The maximum and minimum values of the tunable parameters are obtained from the conventional tuning method (Ziegler-Nichols). Random generation of the initial population within the bounds of the tunable parameters is employed.

Particle swarm optimization (PSO)

PSO is one of the most recent evolutionary algorithms based on the searching behaviour of animals, such as fish schooling and bird flocking (Sultaniya & Gupta, 2014). In the PSO model, each particle is composed of three vectors: the velocity v_i, the current position x_i, and the previous best position $pbest_i$. The velocity in this application is the speed at which or how fast the tunable parameters (K_p, K_i, K_d) of the PID controller reach the optimal values within the search space. Suppose that the objective function is D-dimensional, then the velocity and position of the i^{th} particle are represented as $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$ and $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$, respectively, while its previous best position is stored in $pbest_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \ldots pbest_{iD})$. In each generation, the best position discovered from all pbest positions is known as the global best position, gbest = $(gbest_1, gbest_2, gbest_3 \ldots gbest_D)$. The process of PSO is presented below:

Step 1: Initialization
Assign parameters and create populations of size N = 50
Set iter = 0
Step 2: Reproduction and updating loop
for i = 1, 2,…, N do
Update the velocity v_i of particle x_i by using velocity equation
Update the position of particle x_i by using position equation
Evaluate the fitness value of the new particle x_i
if x_i is better than $pbest_i$ then
Set x_i to be $pbest_i$
end (if)
end (for)
Set the particle with the best fitness value to be gbest
iter = iter +1
Step 3: If the iteration is greater than the maximum iteration, the process is terminated.
Else, go back to Step 2.
The velocity equation in PSO is

$$v_i^{\,t+1} = w \cdot v_i^{\,t} + c_1 \cdot r_1 \cdot (pbesti^{\,t} - x_i^{\,t}) + c_2 \cdot r_2 \cdot (gbest^t - x_i^{\,t}) \qquad \ldots(10)$$

and the position equation is

$$x_i^{\,t+1} = x_i^{\,t} + v_i^{\,t+1} \qquad \ldots(11)$$

where c_1 and c_2 are two positive constants that indicate the relative influence of the cognition and social components, respectively; w is inertia weight that provides a balance between local exploitation and global exploration; r_1 and r_2 are random real values in the interval [0, 1]. For this application, the values of $c_1 = 2$, $c_2 = 4.05$ and w = 0.6. The velocity of the particles on each dimension is clamped to the range [−Vmax,Vmax]. If the terminate criterion is satisfied, the algorithm produces the best solution (gbest). Otherwise, the iteration stage is repeated.

Artificial bee colony optimization (ABC)

In 2005, Karaboga proposed an ABC algorithm which is a swarm intelligence based on the foraging behaviour of honeybee swarms. The colony of artificial bees in the ABC algorithm consists of three groups of bees: employed bees, onlookers and scouts. Employed bees are responsible for searching for a food source and for sharing this information to recruit onlooker bees. Onlooker bees tend to choose better food sources than the employed bees, and search for food further around the nominated food source (Muske & Rawlings, 1993). If a food source is not improved by a predetermined number of trials (denoted by the limit), this employed bee will become a scout bee to search randomly for new food sources.

The main steps of ABC algorithm are given below:

Step 1: Initialization
Allot parameters and populations of size $N_p = 50$, $S_N = N_p/2$.
Set trial = 0 for each population

Step 2: The employed bee phase for $i = 1, 2, \ldots, S_N$ do
Update a new candidate solution v_i by using update equation for the employed bees. Evaluate the fitness value of the candidate solution v_i.
Apply a greedy selection process between v_i and x_i to select the better one.
If solution x_i does not improve, $trial_i = trial_i + 1$, otherwise $trial_i = 0$.
end (for)

Step 3: Calculate the probability Pi by using the probability equation for the solutions x_i using fitness values

Step 4: The onlooker bee phase for $i = 1, 2, \ldots, S_N$
do if rand(0, 1) ≤ Pi then
Update a new candidate solution v_i by using update equation for the onlooker bees.
Evaluate the fitness value of the candidate solution v_i.
Apply a greedy selection process between v_i and x_i to select the better one.
If solution x_i does not improve, $trial_i = trial_i + 1$, otherwise $trial_i = 0$.
end (if)
end (for)

Step 5: The scouts bee phase
if max ($trial_i$) > limit then
Replace x_i with a new randomly produced candidate solution by using the replacing equation
end (if)

Step 6: If the iteration is greater than maximum iteration, stop and output the best solution achieved so far. Otherwise, return to **Step 2**.

A new food source v_i can be produced from the old food source using the update equation (12),

$$v_i^{\,t+1} = x_i^{\,t} + \cdot\,(x_i^{\,t} - x_k^{\,t}) \qquad \ldots(12)$$

where k ∈ {1, 2,..., S_N} is randomly chosen indexes and must be altered from i
The new random food source position (scout bee) will be calculated from the following replacing equation (13),

$$x_{\,t+1} = x_{max} + rand \cdot (x_{max} - x_{min}) \qquad \ldots(13)$$

where x_{min} and x_{max} are the lower and upper bound of the food source position, respectively. The candidate solution is compared with the old one. If the new food source has a better quality compared to the old source, then the new one replaces the latter. Otherwise, the old source is retained.

Design of model predictive controller (MPC)

Model Predictive Control is an optimization-based controller in which constraints to input and output signals can be given. For the design of MPC, it uses a dynamic process model, a cost function (J) over the horizon, and optimization of the cost function using the control input (u). The cost function of the process is,

$$J = \sum_{j=N_1}^{N_2} w_e\,[\hat{y}\,(t+j) - w\,(t+j)]^{\,2} + \sum_{j=N_1}^{N_2} w_u\,[\Delta u\,(t+j-1)]^{\,2} \qquad \ldots(14)$$

where N_1 = Minimum costing prediction horizon
N_2 = maximum costing prediction
N_u = length of control horizon
$\hat{y}$ = predicted output
w = reference trajectory
u = manipulated input
w_e = weight to the error of prediction
w_u = weight to the change in control action

In MPC, future control input and future process output are predicted using the process model, and the cost function of the system has been optimized over the prediction horizon while emulating the constraints provided. At the current time step, MPC applies only the first step of the optimal solution as a control signal; then, the prediction horizon shifts forward one step at a time, and the same method continues. Based on the control horizon given, the numbers of control action values are considered as free variables to be optimized. The basic structure of MPC is integrated in Figure 3.

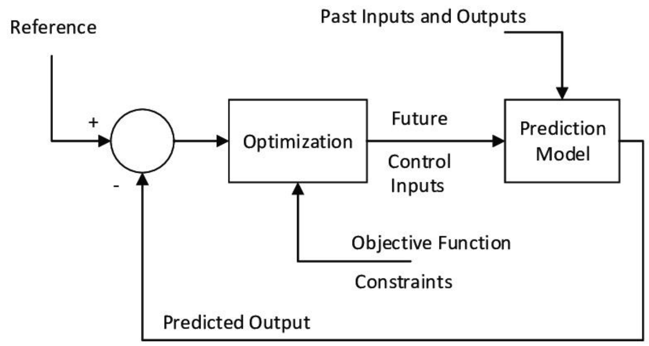

Figure 3: Structure of MPC

The performance of the controller is measured in terms of:

1. Overshoot: It is a measure of how much the response exceeds the ultimate value following a step change. Overshoot = A/B = $\exp\left(-\pi\varepsilon/\sqrt{1-\varepsilon^2}\right)$
2. Rise time (T_r) is the time required for the response to rise from 10% to 90% of its final value.
3. Settling time (T_s) is the time required for the response curve to reach and stay within a range about the final value of size specified by the absolute percentage of the final value (given as 2%).
4. Mean square error, $MSE = 1/T \int_{t=0}^{T} e^2 \, dt$
5. Integral of the square of the error (ISE), $ISE = \int_{t=0}^{T} e^2 \, dt$
6. Integral of the absolute value of error (IAE), $IAE = \int_{t=0}^{T} |e| \, dt$
7. Integral of time-weighted absolute error (ITAE), $ITAE = \int_{t=0}^{T} |e| \, t \, dt$

RESULTS AND DISCUSSION

A non-isothermal CSTR experimental setup was designed and studied by conducting a neutralization reaction, which is exothermic, with strong acid and strong base. The data acquired were plotted in Figure 4, which shows that the temperature of the reactor fluid depends on the flow rate and temperature of the coolant. For implementing temperature control in the CSTR, an experimental model is required. The transfer function model is obtained using system identification methods in the MATLAB environment. The transfer function model obtained using experimental data is given by equation 16, and the estimated transfer function, along with the measured data, is represented in Figure 5. In this section, the response of the experimental models to the other models, such as the MIMO (Thulasi Dharan *et al.*, 2017), State-space (Masilamani *et al.*, 2015), and mathematical models (Swapnadeep & Lillie, 2017) are considered for the sake of comparison.

$$G_p = \frac{(0.02375\,s + 4.19\,e^{-5})}{(s^2 + 0.002181\,s + 1.917\,e^{-6})} \quad \ldots(15)$$

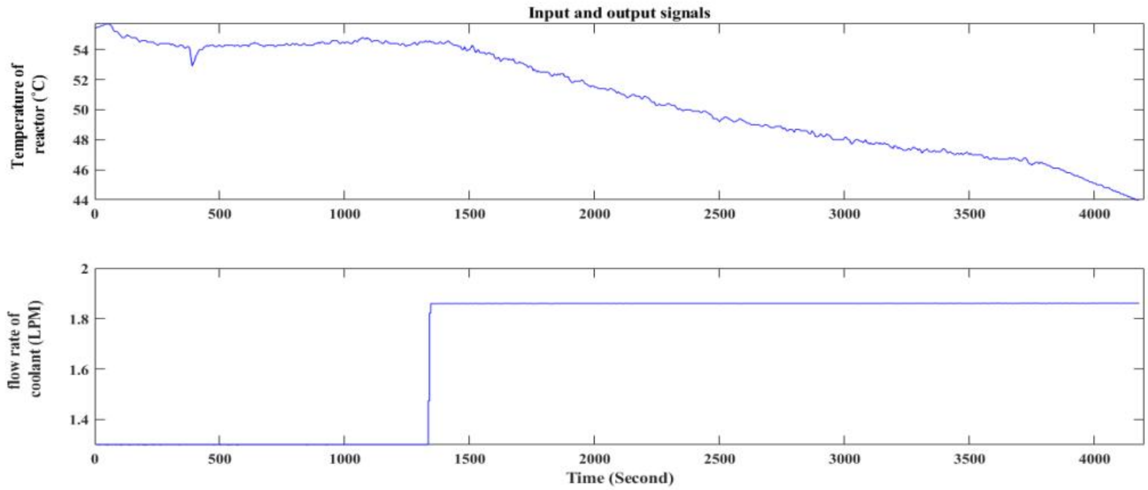

Figure 4: Data acquired by conducting an exothermic reaction

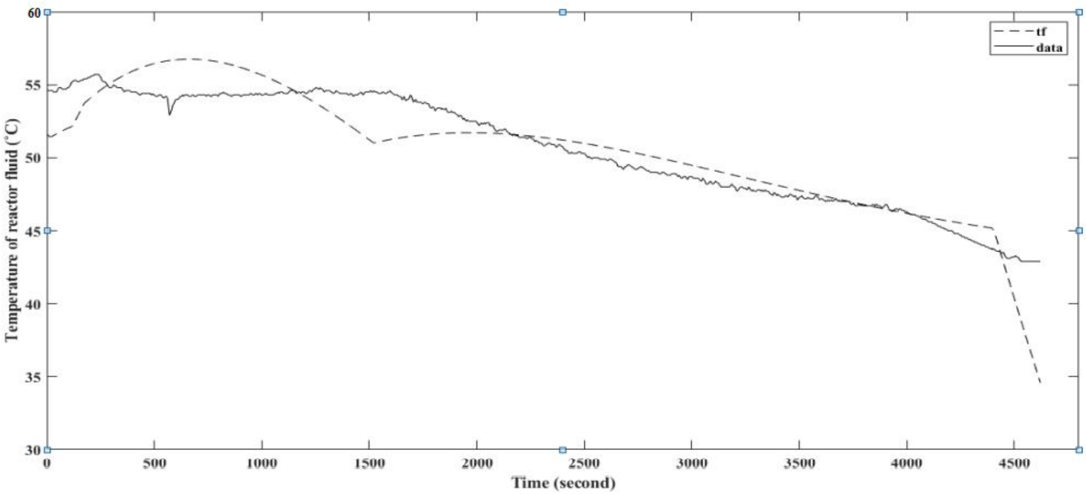

Figure 5: Transfer function model response

In the PID controller for the ZN method, the Bode diagram was plotted as in Figure 6, and the gain margin is 1.459 (Abs), the crossover frequency is 0.0421(rad/s), and the phase margin is 53.8884 (deg). Then the ultimate gain K_u = 0.685, and the ultimate period P_u = 149.2443 are calculated. Using the ZN controller setting, K_p = 0.411, K_i = 0.0055, and K_d = 7.6674 were found, and the step response of the temperature control system using various models is displayed in Figure 7.

In tuning PID parameters using PSO, the PSO parameters are as follows: iteration (N) = 100, number of particles (i) = 50, problem dimension = 3 (K_p, K_i, K_d), upper and lower boundary to K_p, K_i, and K_d values are LB = 0.0001, 0.0001, 0.0001, respectively, UB = 10, 2, 30, respectively, constant values are c_1 = 2 and c_2 = 4.05, and inertia weight, w = 0.6. The inertia weight plays an important role in the convergence behaviour of the algorithm, which in turn affects the time domain response of the closed loop system, as seen in Table 3. The objective function for this is given as the summation of ITAE and overshoot,

$$Obj = 20 \times sum\ (t.|e|dt) + 80 \times (\max\ (y) - y_d) \times 100 \qquad \ldots(16)$$

The PSO takes equation (16) as the objective function and aims to minimize it. The values of K_p, K_i, K_d corresponding to the minimized value of the objective function are the optimized values. These optimized values are used as the controller parameters of the PID controller used in the temperature control system. The particles, in this case is a 50 × 3 matrix as the number of particles is 50 and the dimension is 3. For each particle in the matrix, the value of the fitness function is calculated using equation (16), and the velocity and position of the particle are also obtained using equations (11) and (12), respectively. The evaluated fitness function values are stored and compared to the global best fitness function. There is updating of the velocity, position and the local best if the evaluated fitness function is better than the global best.

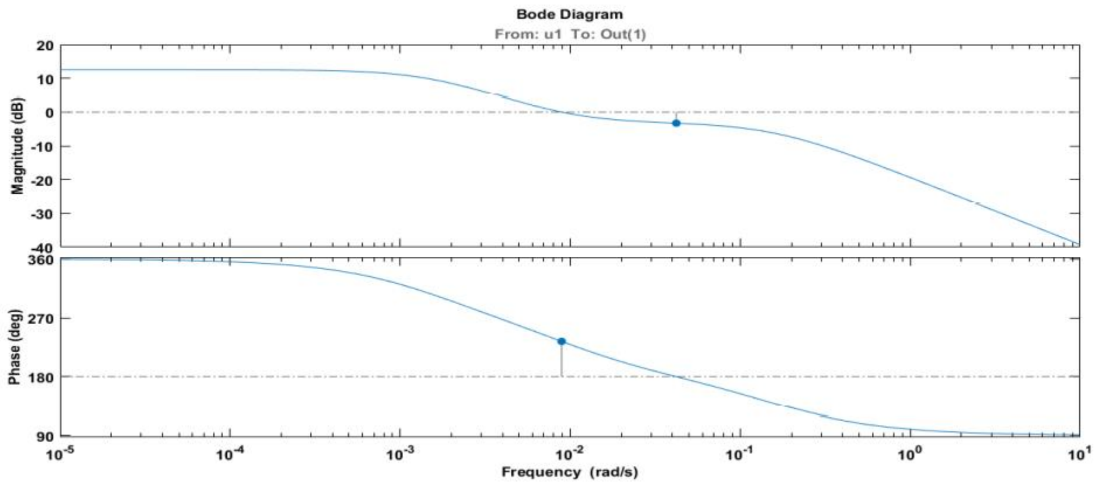

Figure 6: Bode diagram using transfer function model

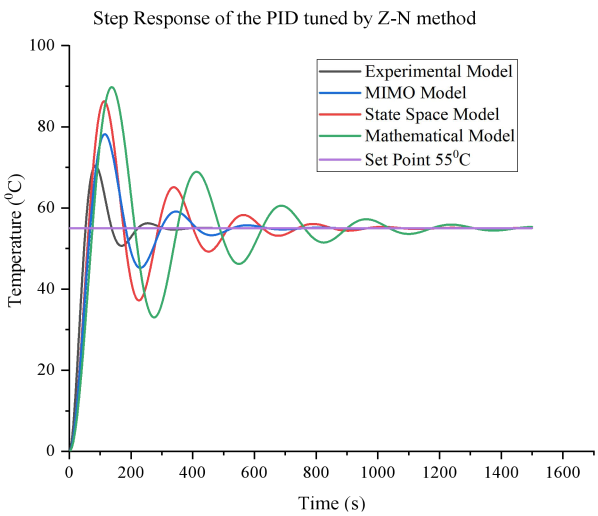

Figure 7: Step response of temperature control system with ZN tuning method

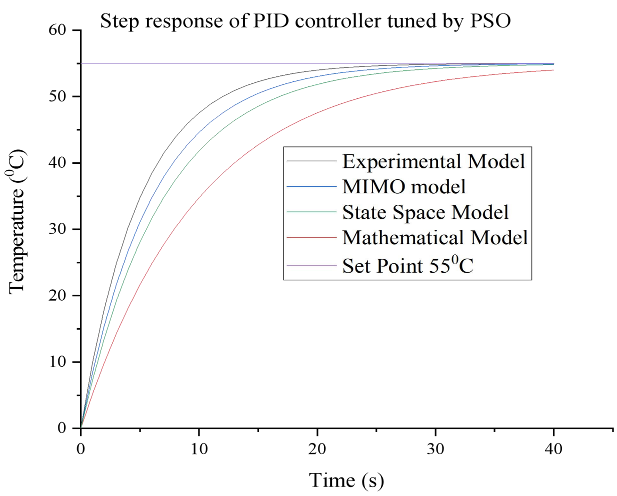

Figure 8: Step response of temperature control system with PSO technique for tuning

The parameters of ABC are assigned as follows: maximum number of iterations = 50, problem dimension (D) = 3 [K_p, K_i, K_d], size of the population (N_p) = 50, the number of food sources (SN) = $N_p/2$, and the boundary condition for K_p, K_i, K_d are as follows: x_{min} = 0, 0, 0 and x_{max} = 10, 2, 30, respectively. The objective function of this problem is considered as the combination of ITAE and overshoot as in equation (16),

In the system considered, the bees are a 50×3 matrix containing the possible values of K_p, K_i, K_d, aiming to minimize the objective function in equation (16). The food, in this case, is the minimum value of the objective function among the values of the objective functions obtained for each bee. In ABC, the bees are in search of the optimal solution, and the solutions are updated frequently as a result of the neighbourhood search. When the optimal solution is achieved, the objective function is at its minimum. The optimal solution is the tuned values of the K_p, K_i, and K_d; these values are used in the PID controller applied in the temperature control loop of the CSTR.

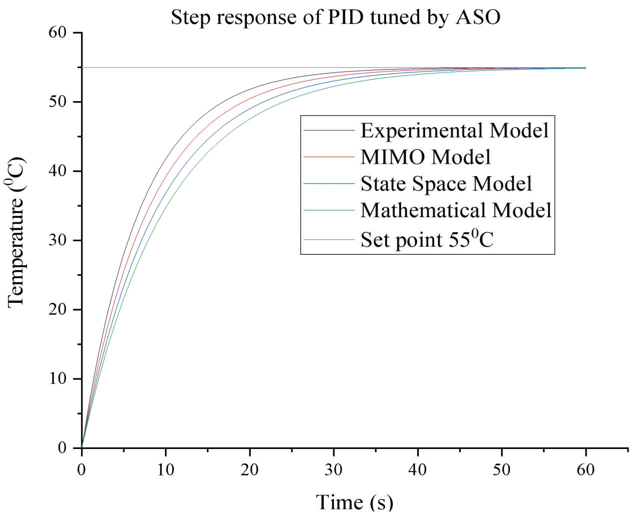

Figure 9: Step response of temperature control system with ABC optimization technique for tuning

In equation 18, e = error, y = output response, and y_d = desired output. By following the appropriate steps mentioned above, tunable PID parameters can be obtained and are categorized in Table 3.

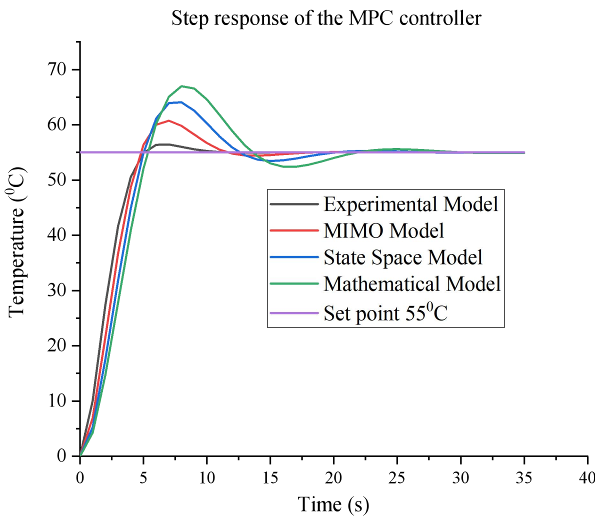

Figure 10: Step response of temperature control system using MPC

The step response of PSO-based PID tuning is depicted in Figure 8, and the step response of ABC-based PID tuning is shown in Figure 9, and from the plots, the time domain specifications are obtained. The model predictive controller was designed with transfer function model, length of prediction horizon (N_p) given as 10, length of control (N_u) given as 2, weight to the error of prediction (w_e) given as 1, and weight to the change in control action (w_u) given as 0.1.

The step response of the temperature control system using MPC is shown in Figure 10. The performance of the controller for various models was accessed and enumerated in Tables 3 and 4.

Table 3: Comparison of tuning methods considering various models

Model of CSTR	Set point (˚C)	Tuning algorithm	K_p	K_i	K_d	Overshoot %	T_r (second)	T_s (second)
Experimental	55	Z-N	0.411	0.0055	7.6674	28.4	96.6	734
		PSO	10	0.0076	1.4734	0	9.53	21.2
		ABC	6.567	0.0164	2.5477	0	14.4	23.5
		MPC	4.488	0.0172	1.5678	0	6.48	16.8
MIMO (Thulasi Dharan *et al*., 2017)	55	Z-N	1.201	0.0155	8.2458	29.25	98.54	746
		PSO	12.35	0.0865	2.3458	0	9.95	23.3
		ABC	6.567	0.0164	2.5477	0	15.6	25.4
		MPC	4.488	0.0172	1.9867	5	7.63	17.2
State-space (Masilamani *et al*., 2015)	55	Z-N	3.258	0.0824	9.5687	32.64	98.99	1000
		PSO	14.02	0.1065	3.6547	0	15.24	25.1
		ABC	7.368	0.1168	3.0058	0	17.68	27.6
		MPC	5.006	0.1358	2.257	8.26	8.66	19.3
Mathematical (Swapnadeep & Lillie, 2017)	55	Z-N	5.258	0.0992	12.358	35.45	100.3	1500
		PSO	16.38	0.2058	4.6587	0	18.41	27.2
		ABC	8.467	0.2587	4.8794	0	19.64	30.5
		MPC	7.05	0.3687	3.5042	10.35	9.25	22.4

Table 4: Performance of controllers considering various models

Model of CSTR	Set point (˚C)	Controller	MSE	ISE	IAE	ITAE
Experimental	55	Z-N	0.411	0.0055	7.6674	28.4
		PSO	10	0.0076	1.4734	0
		ABC	6.567	0.0164	2.5477	0
		MPC	4.488	0.0172	1.5678	0
MIMO (Thulasi Dharan *et al*., 2017)	55	Z-N	0.995	0.0089	8.3679	30.5
		PSO	12.358	0.0093	2.9658	0
		ABC	8.367	0.0568	3.5687	0
		MPC	5.368	0.0639	2.6879	0
State-space (Masilamani *et al*., 2015)	55	Z-N	1.384	0.0100	10.3574	32.5
		PSO	13.587	0.0158	3.6789	0
		ABC	9.367	0.0985	5.0010	0
		MPC	6.3487	0.0109	4.0068	0
Mathematical (Swapnadeep & Lillie, 2017)	55	Z-N	2.258	0.0258	11.3681	34.2
		PSO	14.3687	0.0518	4.6897	0
		ABC	10.3658	0.1025	6.9387	0
		MPC	8.3679	0.0912	5.2581	0

Set point tracking

The set point tracking responses of the temperature of the reactor fluid with PID, PID-PSO, PID-ABC and MPC with the positive step change of amplitude 5, given at 800 seconds, are plotted in Figure 11.

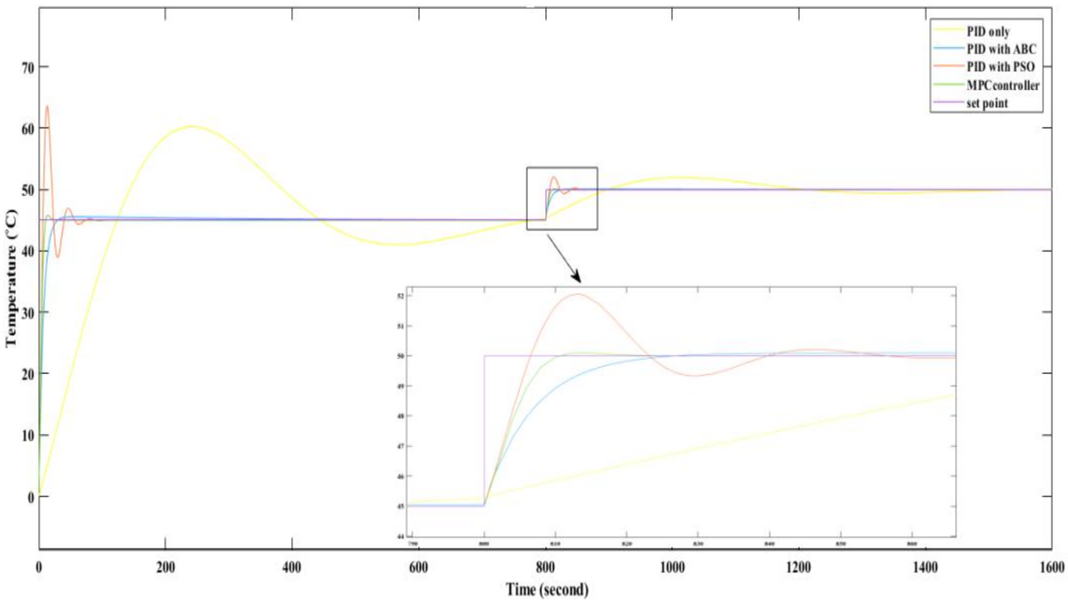

Figure 11: Set point tracking of PID, PID-PSO, PID-ABC and MPC

Disturbance rejection

The disturbance rejection of the response of the temperature of reactor fluid with PID, PID-PSO, PID-ABC and MPC with the negative step change of amplitude 5, given at 800 seconds, are graphed in Figure 12.

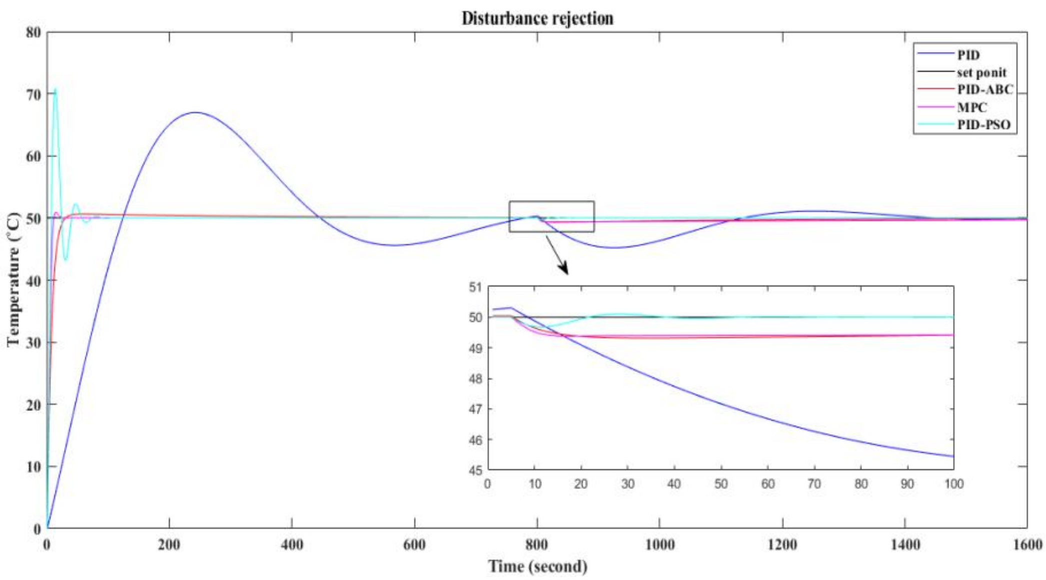

Figure 12: Disturbance rejection of PID, PID-PSO, PID-ABC and MPC

CONCLUSION

The experimental model of the CSTR system was considered to develop the controllers. Various models from the literature were used for the controller design alongside the experimental model for the sake of comparison. The PID, PID-PSO, PID-ABC, and MPC controllers were developed using the various models and compared in terms of MSE, ISE, IAE, ITAE and time domain specifications such as overshoot, rise time, and settling time. The objective function used was the combination of ITAE and overshoot for the optimization algorithms PSO and ABC. From the simulation results, the experimental model performed better than the other models in all the cases of tuning methods. PID-PSO using the experimental model has minimized the overshoot (0%), rise time (9.53 seconds), and settling time (21.2 seconds). However, the MPC is better than the PID controller with evolutionary algorithms. The time domain specifications of MPC are overshoot (0%), rise time (6.48 seconds) and settling time (16.8 seconds). The values of MSE, ISE, IAE, and ITAE are 7.307, 0.7307×10^4, 219.5 and 712.9 for MPC, respectively. With the set point tracking studies, the MPC controller has better tracking performance, and in disturbance rejection, the PID controller with PSO has improved rejection compared to the MPC controller. For future studies, the evolutionary algorithms can be hybridized by exploiting the best parts of each algorithm to form a more effective algorithm. Models for the CSTR systems can be improved by using non-linear models such as the Hammerstein-Wiener models.

REFERENCES

Agarwal V., Deo A. & Kumar N. (2015). Temperature control of CSTR using PID controller. *International Journal of Engineering and Computer Science* **5**(04): 11902–11905.

Ahmed A.O., Gasmelseed G.A., Karama A. & Musa A.E. (2016). Cascade control of a continuous stirred tank reactor (CSTR). *Journal of Applied and Industrial Sciences* **1**(04): 16–23.

Aslam F. & Kaur G. (2011). Comparative analysis of conventional, P, PI, PID and fuzzy logic controllers for the efficient control of concentration in CSTR. *International Journal of Computer Applications* **6**(17): 12–16.
DOI: https://doi.org/10.5120/2226-2836

Bakošová M., Puna, D. & Mészáros A. (2005). Robust controller design for a chemical reactor. *Computer Aided Chemical Engineering* **20**: 1303–1308.
DOI: https://doi.org/10.1016/S1570-7946(05)80059-8

Bonyadi M.R. & Michalewicz Z. (2017). Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary Computation* **25**(1): 1–54.
DOI: https://doi.org/10.1162/EVCO_r_00180

Da H. & Shu-cai H. (2018). Optimal controller design based on minimum principle. *MATEC Web of Conferences* **173**(1): 01001.
DOI: https://doi.org/

Das S., Biswas A., Dasgupta S. & Abraham A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: *Foundations of Computational Intelligence*, volume 3. Studies in Computational Intelligence, vol 203. Springer, Berlin, Heidelberg, Germany.
DOI: https://doi.org/10.1007/978-3-642-01085-9_2

Dey B.K. & Roy L. (2014). A comparative study on temperature control of CSTR using PI controller, PID controller and PID (two degree of freedom) controller. *International Journal of Electronics and Computer Science Engineering* **52**(01): 62–68.

Dong S., Wang Y., Hu W., Zhang G. & Zhan J. (2021). Constant temperature control system of building energy system. *Thermal Science* **25**(4B): 2853–2860.
DOI: https://doi.org/10.2298/TSCI2104853D

Gao D.X. & Liu H. (2015). Optimal dynamic control for CSTR nonlinear system based on feedback linearization. *27th Chinese Control and Decision Conference*, pp.1298–1302.
DOI: https://doi.org/10.1109/CCDC.2015.7162119

Masilamani A., Panda R., Sivamani R. & Dheepak R. (2015). Design of MPC controller for CSTR. *International Journal of Applied Engineering Research* **10**(33): 1303–1308.

Muske K.R. & Rawlings J.B. (1993). Model predictive control with linear models. *AIChE Journal- Process Systems Engineering* **39**(02): 262–287.
DOI: https://doi.org/10.1002/aic.690390208

Rani V.A., Prabhakaran D. & Thirumarimurugan M. (2020). Modelling and control of pH in a continuous stirred tank Reactor (CSTR). *Journal of Environmental Protection and Ecology* **21**(02): 413–422.

Saravanakumar G., Valarmathi K., Pallikonda M.R. & Seshadhri S. (2015). Tuning of multivariable decentralized PID controller using state transition algorithm. *Studies in Informatics and Control* **24**(4): 367–378.
DOI: https://doi.org/10.24846/v24i4y201501

Sultaniya S. & Gupta R. (2014). Design of PID controller using PSO algorithm for CSTR. *International Journal of Electronic and Electrical Engineering* **07**(09): 971–977.

Swapnadeep B. & Lillie D. (2017). A comparative study of PID based temperature control of CSTR using genetic algorithm and particle swarm optimization. In: *International Conference on Emerging Trends in Computing and Communication Technologies*, pp. 1–6.
DOI: https://doi.org/10.1109/ICETCCT.2017.8280312

Thulasi Dharan S., Kavyarasan K. & Bagyaveereswaran V. (2017). Tuning of PID controller using optimization techniques for a MIMO process. IOP Conference Series: *Materials Science and Engineering* **263**(5).
DOI: https://doi.org/10.1088/1757-899X/263/5/052019

Weiguo Z., Liying W. & Zhenxing Z. (2019). Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems* **163**: 283–304.
DOI: https://doi.org/10.1016/j.knosys.2018.08.030.

Zhiyong L., Weiyou W., Yanyan Y. & Zheng L. (2015). A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems. *Expert Systems with Applications* **42**(22): 8881–8895.
DOI: https://doi.org/10.1016/j.eswa.2015.07.043