Advanced Control Strategies for Continuous Stirred Tank Reactors:

Optimization for Enhanced Performance

Rami BOUMEGOURA(1,2)*, Riad BENDIB(1,2), Kamel MENIGHED(1)
(1) Department of Petrochemicals, University of 20 August 1955, Skikda, Algeria
(2) Automatic Laboratory of Skikda (L.A.S), University of 20 August 1955Skikda, Algeria
*r.boumegoura@univ-skikda.dz

Abstract: This paper investigates the control of a Continuous Stirred Tank Reactor (CSTR), with a primary emphasis on achieving temperature stability and optimizing reactant conversion. The CSTR poses challenges due to its nonlinear and exothermic behavior. To address these challenges, Model Predictive Control (MPC) is employed, a powerful strategy for handling complex systems. Additionally, Particle Swarm Optimization (PSO) is introduced to fine-tune MPC parameters, ensuring optimal performance. By integrating PSO with MPC, this study enhances control capabilities, specifically targeting the intricate demands of CSTR systems.

Keywords: CSTR, Optimization problem, MPC, Control strategies, PSO, Process control

1. INTRODUCTION

The Continuous Stirred Tank Reactor (CSTR) is a vital component in industrial facilities where chemical reactions are conducted to produce various end products [1]. The efficient management of these reactions is vital to ensure the best operation of the CSTR. Two key aspects play a pivotal role in achieving this: the maintenance of steady temperatures and the maximum of reactant consumption within the CSTR.

To boost CSTR performance, numerous control mechanisms have been designed. These include conventional PID controllers and more sophisticated methods that make use of artificial intelligence techniques like fuzzy logic and neural network [2, 12-13], in addition to metaheuristic algorithms that are used to optimize PID controller parameters, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) [3], and Ant Colony Optimization (ACO) [4]. However, while some researchers have studied the substitution of PID controllers with Model Predictive Control (MPC) to maximize CSTR performance [5], there is a significant gap in the thorough examination of strengthening MPC controllers within this area.

Notably, as we have watched substantial developments in PID controllers over time, it becomes obvious that MPC, while a powerful tool in its own right, has space for growth. One interesting option for development involves incorporating metaheuristic algorithms with MPC.Optimization methods have changed greatly over time. In 1975, John Holland and his students introduced Genetic Algorithm (GA), an evolutionary optimization tool. Nevertheless, it faced problems like as resource consumption and computational complexity [6]. Ant Colony Optimization (ACO), developed by Marco Dorigo et al. in 1991 [7], is efficient at solving discrete-type problems and functions. Additionally, Particle Swarm Optimization (PSO), invented in 1995 by R. Eberhart and J. Kennedy [8], has gained appeal as an optimization tool for nonlinear continuous problems. Recognizing the need for refinement, various researchers have proposed adjustments to these algorithms to boost their performance. For instance, in [9], the insertion of a new parameter named 'inertia' into the PSO algorithm led to considerable performance increases. Detailed explanations of these approaches and their parameters are available in [10].

Integrating metaheuristic algorithms with controllers such as PID [3-4], fuzzy logic [17-21], and MPC offers a powerful approach to enhance control performance in industrial processes. By leveraging the capabilities of these algorithms, engineers can automate and optimize control strategies, thereby improving system efficiency, stability, and responsiveness to dynamic operational conditions.

2. CSTR MODELING

CSTRs, or continuous stirred-tank reactors, are extensively utilized in the industrial sector. In comparison to other forms of continuous reactors, such as tubular and packed-bed reactors, they are easier to model [11]. This simplicity renders CSTR models an excellent starting point for conducting in-depth investigations and learning about chemical reactors.

2.1 CSTR system description

In a typical CSTR setup, a reactor tank with a constant volume V is used. It receives a feed stream with an initial concentration C_{Af} and a constant flow rate q. The feed stream also has an initial temperature T_f. Inside the reactor, an irreversible exothermic reaction takes place, converting reactant A into product B. This chemical reaction generates heat. To effectively manage this heat, a coolant jacket surrounds the reactor. The temperature of the coolant, denoted as T_c, plays a crucial role in absorbing the heat produced during the reaction, ensuring the reactor remains at the desired operating temperature [1]. Figure 1 illustrates the typical configuration of a CSTR along with its input and output streams.

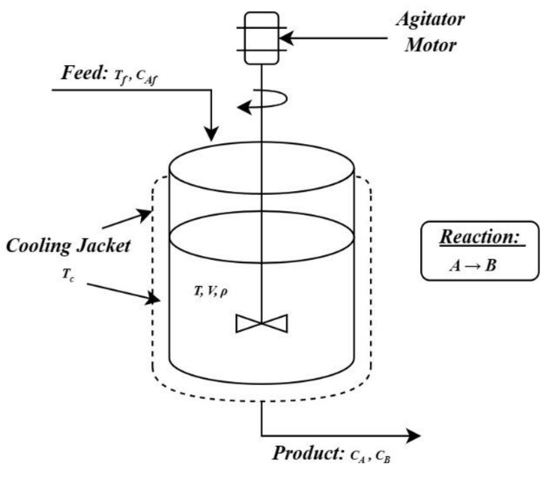

Fig.1. Continuously Stirred Tank Reactor

The system is intended to function under dynamic or transient conditions, as the concentrations fluctuate over time as a result of the ongoing reaction. In order to achieve optimal performance, reactants are thoroughly mixed [1].

2.2 Mathematical Modeling

The model development incorporates the following assumptions:

1. The reactor's volume remains constant throughout the process.
2. The reactor's contents are thoroughly mixed, ensuring uniform concentration distribution.
3. The fluid's density inside the reactor is assumed to be constant.
4. The chemical reaction within the CSTR is considered to be of first-order kinetics.
5. The reactor functions adiabatically, with minimal heat exchange with its environs, except for controlled cooling via the jacket.
6. The heat transfer through the cooling jacket is ideal, characterized by a constant and efficient heat transfer coefficient.

The CSTR is mathematically described through the application of the first principles of thermodynamics. This description involves two key equations: Eq. (1), which represents the mass balance within the reactor, and Eq. (2), which characterizes the energy balance in the system.

$$V\frac{dC_A}{dt} = q\left(C_{Af} - C_A\right) - Vr_A \qquad (1)$$

$$V\rho C_p \frac{dT}{dt} = q\rho C_p (T_f - T) + (-\Delta H_r)Vr_A - UA(T - T_c) \qquad (2)$$

The reaction rate r_A is controlled by the Arrhenius expression in the following way:

$$r_A = k_0 e^{-E/RT} \qquad (3)$$

Expressing equations (1) and (2) in the state variable format,

$$f_1(C_A, T) = \frac{dC_A}{dt} = \frac{q}{V}(C_{Af} - C_A) - k_0 e^{-E/RT} C_A \quad \ldots \qquad (4)$$

$$f_2(C_A, T) = \frac{dT}{dt} = \frac{q}{V}(T_f - T) + \left(-\frac{\Delta H_r}{\rho C_p}\right) k_0 e^{-E/RT} C_A - \frac{UA}{V\rho C_p}(T - T_c) \qquad (5)$$

The CSTR equations involve complex, interconnected, and nonlinear functions related to C_A and T. To simplify and analyze the model, we utilize the Jacobian matrix method, which allows us to linearize equations (1) and (2). This transformation leads to the state variable form $x' = Ax + Bu$, where $y = Cx$. In this context, A and B represent the Jacobian matrices, while C functions as the output matrix. The state vectors are expressed as:

$$x = \begin{bmatrix} C_A \\ T \end{bmatrix} ; \ u = \begin{bmatrix} C_{Af} \\ T_f \\ T_c \end{bmatrix} \tag{6}$$

The Jacobian matrix A is defined as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{7}$$

Where:

$$A_{11} = \frac{\partial f_1}{\partial x_1} = -\frac{q}{V} - k_0 e^{-E/RT}$$

$$A_{12} = \frac{\partial f_1}{\partial x_2} = -\frac{E}{RT^2} k_0 C_A e^{-E/RT}$$

$$A_{21} = \frac{\partial f_2}{\partial x_1} = \frac{\Delta H_r}{\rho C_p} k_0 e^{-E/RT}$$

$$A_{22} = \frac{\partial f_2}{\partial x_2} = \left(\frac{\Delta H_r}{\rho C_p} k_0 C_A e^{-E/RT} \right) \left(\frac{E}{RT^2} C_A \right) - \frac{q}{V} - \frac{UA}{V \rho C_p}$$

The Jacobian matrix B is defined as

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix} \tag{8}$$

Where,

$$B_{11} = \frac{\partial f_1}{\partial u_1} = \frac{q}{V}, \quad B_{12} = B_{13} = B_{21} = 0$$

$$B_{22} = \frac{\partial f_2}{\partial u_2} = \frac{q}{V}, \quad B_{23} = \frac{\partial f_2}{\partial u_3} = \frac{UA}{V \rho C_p}$$

The matrix C is defined as

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9}$$

Table-1: CSTRModel Parameters

No	Parameters	Symbol	Value
1	FeedConcentration	C_{Af}	1 mol/m^3
2	ConcentrationA in CSTR	C_A	0.988 mol/m^3
3	FeedTemperature	T_f	297.6 K
4	Temperature in CSTR	T	322 K
5	Temperature of cooling jacket	T_c	281 K
6	Density of A-B Mixture	ρ	1000 kg/m^3
7	Heat Capacity of A/B Mixture	C_p	0.35 J/kg.K
8	Volume of CSTR	V	100 m^3
9	Volumetric Flowrate	q	100 m^3/sec
10	Heat of reaction	ΔH_r	11450 J/mol
11	Pre-exponential factor	k_0	11x10^{8}1/sec
12	Energy Activation	E/R	6500 K
13	Overall Heat Transfer Coeff	UA	8.9x10^{4}W/K

Upon substituting the values from Table-1 into the matrices, we derive the state-space equation (10)as follows:

$$\begin{bmatrix} C_A' \\ T' \end{bmatrix} = \begin{bmatrix} -2.92 & -0.119 \\ 62.81 & 0.300 \end{bmatrix} \begin{bmatrix} C_A \\ T \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2.54 \end{bmatrix} \begin{bmatrix} C_{Af} \\ T_f \\ T_c \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} C_A \\ T \end{bmatrix} \tag{10}$$

Following the development of these CSTR mathematical models, they will be utilized for our simulations in MATLAB.

3. CONTROL DESIGNING

CSTR systems are naturally unstable, requiring precise control for temperature stability and concentration optimization. While various methods like PID, IMC, and AI-based approaches exist, MPC stands out as a promising choice. Its potential to significantly enhance system performance through proper tuning of control parameters makes it an attractive option.

3.1 Model Predictive Control

Model predictive control (MPC) is an advanced control strategy that relies on a dynamic process model to foresee how a system will behave in the future. Based on these predictions, it computes control inputs that aim to either minimize a cost or maximize a performance criterion. Within

MPC, several crucial parameters come into play, including the prediction horizon, control horizon, constraints, and weights [14-15]. These parameters have a significant impact on the effectiveness of the control process and system performance.The choice and tuning of these parameters are critical in the MPC process. Optimally setting these parameters can lead to enhanced system performance, stability, and efficiency. However, finding the best combination of parameters can be a complex optimization problem.To address this challenge, meta-heuristic algorithms like Particle Swarm Optimization (PSO) are often employed as a study tool. PSO can efficiently search the parameter space to find the combination that minimizes or maximizes the desired objective function, which, in the context of MPC, would be the system's performance or cost criterion. By using PSO or similar techniques, engineers and control system designers can fine-tune MPC parameters to achieve the best possible control performance for a given system. This ensures that the MPC controller operates optimally and effectively in real-world applications.

3.2 Particle Swarm Algorithm

In the Particle Swarm Optimization (PSO) algorithm, the process begins with the random placement of a swarm, and their associated costs are evaluated. Initially, all particles have zero velocity. During each iteration, each particle maintains its personal best value and location while sharing this information with the entire swarm. Simultaneously, a global best value and location are determined collectively, and the swarm repeats the process. You can visualize the algorithm from Figure 2. In a search space with 'N' particles, the speed and position of each particle in the swarm change based on two equations:

1. The velocity update equation:

$$v_i = \omega v_i + c_1 r_1 (P_{best,i} - x_i) + c_2 r_2 (g_{best} - x_i) \quad (11)$$

2. The position update equation:

$$x_i = x_i + v_i \quad (12)$$

Here, $\boldsymbol{v_i}$ and $\boldsymbol{x_i}$ represent the velocity and position of the '*i-th*' particle. $\boldsymbol{P_{best,i}}$ and $\boldsymbol{g_{best}}$ represent the personal and global best positions of the '*i-th*' particle. $\boldsymbol{\omega}$ represents inertia based on the particle's previous position, and $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ represent acceleration constants. $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are random numbers chosen from the range [0,1].The algorithm starts with defining the cost function to minimize and specifying the swarm size (the number of particles). The number of problem variables, which corresponds to the dimension of the problem, is also set. Additionally, upper and lower bounds for the search space are defined to constrain the algorithm within the desired search region.

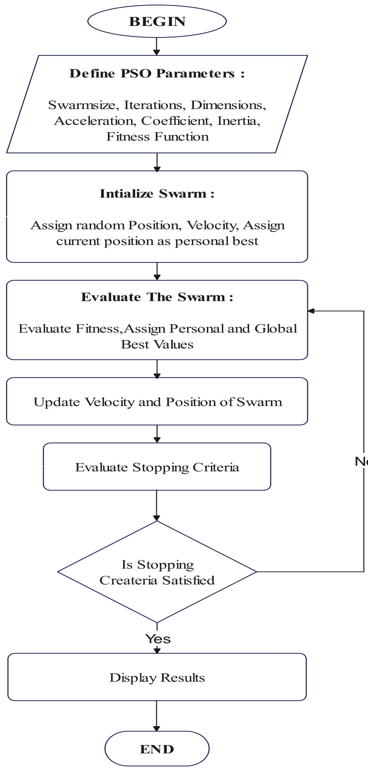

Fig.2. Particle Swarm Optimization Algorithm

Inertia is defined, affecting particle velocity. Higher inertia results in more momentum and higher velocity. The acceleration constants, c1 and c2, impact local and global search behavior, respectively.Key parameters like the maximum number of iterations, function tolerance, maximum stall iterations, and the maximum time for the algorithm to run are defined. These parameters determine when the algorithm should terminate or deliver results.The swarm is then initialized, and positions are randomly assigned within the search space. The objective function value is computed for each position, and both personal and global best positions are remembered. Particle velocities are updated, and the swarm moves through the search space accordingly. This process continues iteratively. Over time, the swarm converges toward the best position in the search space, ultimately minimizing the objective function cost.

4. SIMULATION AND RESULTS

In this section, we present the simulation process and the results obtained through the application of Particle Swarm Optimization (PSO) to fine-tune Model Predictive Control (MPC) parameters for a Continuous Stirred-Tank Reactor (CSTR) system. The objective was to optimize the MPC controller's performance in maintaining desired concentration and temperature setpoints within the CSTR while adhering to control constraints.

4.1 Simulation Setup

We initialized the PSO optimization process with a set of parameters, as outlined in Table 2. These parameters controlled the PSO algorithm, including the population size, the maximum number of iterations, inertia weight, cognitive and social parameters, and the ranges for the randomization of particle positions.

Table-2: PSO Optimization Parameters

Parameters	Value/Range
Swarm Size (*N*)	10
Number of Iteration (*Tmax*)	60
Inertia Weight Range (ω)	[0.1, 1]
Cognitive Parameter (c_1)	2
Social Parameter (c_2)	2
Range for r_1	[0, 1]
Range for r_2	[0, 1]

The CSTR system was described by a continuous plant model, and key variables such as feed concentration, feed temperature, cooling jacket temperature, and the initial state of concentration (CA) and temperature (T) within the reactor were defined. The MPC controller was configured with ranges for Prediction Horizon and Control Horizon, as well as weights for manipulated variables (MV) and MV rate of change (MVRate). These parameters were kept within reasonable ranges for optimization.

4.2 PSO Initialization

During the initialization phase of the PSO algorithm, multiple MPCs were created, each with a distinct configuration of parameters. These parameters encompassed key aspects such as Prediction Horizon, Control Horizon, and the weighting factors assigned to manipulated variables (MV) and manipulated variable rate (MVRate). These parameter sets defined how each MPC would control the CSTR system.To ensure practicality, we applied constraints to particle positions, as described in the code. For example, we constrained the Prediction Horizon between 10 and 50 and the Control Horizon between 1 and 10.

These constraints helped maintain realistic control settings.To assess the performance of different MPCs during the initialization phase, we employed a widely recognized evaluation method known as the sum of squared differences (SSD). This method quantifies the dissimilarity between the controller's output and a reference signal by calculating the squared differences between corresponding values at each time step.

The initialization process continued until at least one particle achieved a satisfactory control performance, indicated by a best value lower than the specified threshold of 80. This convergence ensured that the optimization process started from a viable parameter set.Among the various MPCs created with their distinct parameter sets, the best value performance observed during the initialization phase stood at approximately 10.2203. This numerical value signifies the lowest cost or error attained by any of the MPCs when operating with their respective parameterizations. It serves as the starting point and reference for all subsequent optimization efforts conducted via the PSO algorithm.

This initial performance metric establishes a baseline against which the progress and improvements achieved by the PSO algorithm can be measured. The primary objective of the optimization process is to iteratively fine-tune the MPC parameters, thereby enhancing control performance for the CSTR system. By reducing the performance metric further from this starting point, the PSO algorithm seeks to achieve more precise and efficient control of the reactor's concentration and temperature.

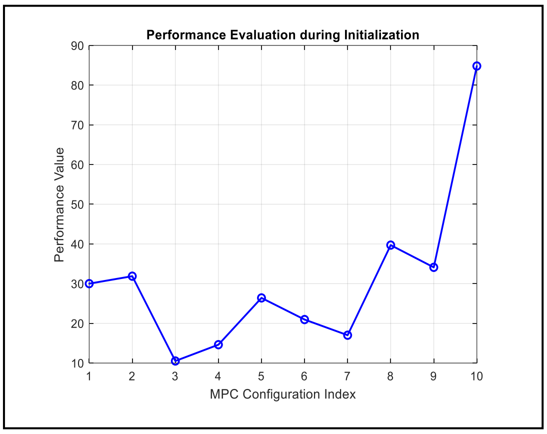

Fig.3. Performance Evaluation of different MPCs Configurations during the initialization phase of PSO

4.3 PSO Optimization

Once the PSO initialization phase was complete, the optimization loop commenced. In each iteration, particles representing different sets of MPC parameters were updated based on their individual and global best positions. This update process was influenced by cognitive and social factors, enabling particles to explore and exploit the parameter space efficiently.To monitor the progress of the PSO optimization, we diligently tracked the best value achieved in each iteration.

The best value represents the lowest cost or error obtained by any of the MPCs with their respective parameter sets. This value serves as a critical metric for assessing the effectiveness of the PSO algorithm in improving control performance.

The convergence of the optimization process was observed by plotting the best value against the iteration number see Fig.4, This figure illustrates how the optimization progressed over multiple iterations, converging towards an optimal solution 2.7. The figure demonstrates how the optimization process evolves over time, from initialization to convergence. The significance of this convergence and its implications for control performance are discussed.

Following this, we present the simulation results for both MPC controllers: one configured with parameters prior to optimization and the other with parameters post-optimization. These simulations offer a real-world assessment of the controller's performance across various parameter configurations.

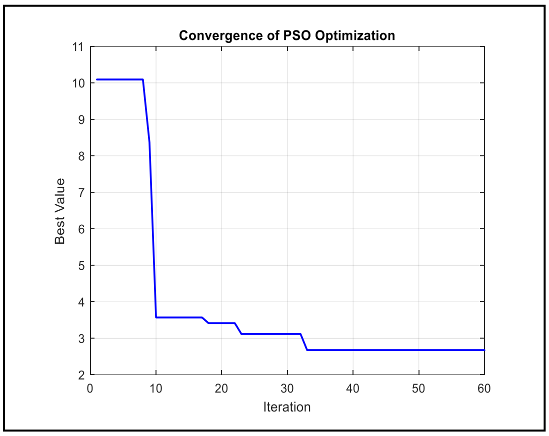

Fig.4. Convergence of PSO Optimization

To quantify the impact of the PSO optimization, Table-3 compares the simulation results of the MPC controllers

Table-3: Summary of Changes in MPC Parameters Before and After Optimization

Parameter	Before Optimization	After Optimization
Step Time (fixed)	0.10	0.10
Prediction Horizon	18	21
Control Horizon	4	1
MV Weights	0.41	0.14
MV Rate Weights	4.16	4.42
OV Weights 1	13.32	13.70
OV Weights 2	15.16	13.62
Performance	10.2203	2.689

Figure 5 vividly illustrates the substantial improvement achieved through PSO optimization. Both concentration and temperature control significantly outperform their pre-optimization counterparts. This enhancement is primarily attributed to the precise control of the cooling jacket temperature, highlighting the synergistic power of MPC and PSO in fine-tuning control parameters for real-world industrial systems.

5. CONCLUSION

The application of Particle Swarm Optimization to fine-tune MPC parameters for a CSTR system resulted in an improved control strategy. The optimized controller demonstrated superior performance in maintaining desired concentration and temperature setpoints while adhering to control constraints. The approach presented in this work showcases the potential of PSO in enhancing control systems for complex chemical processes.

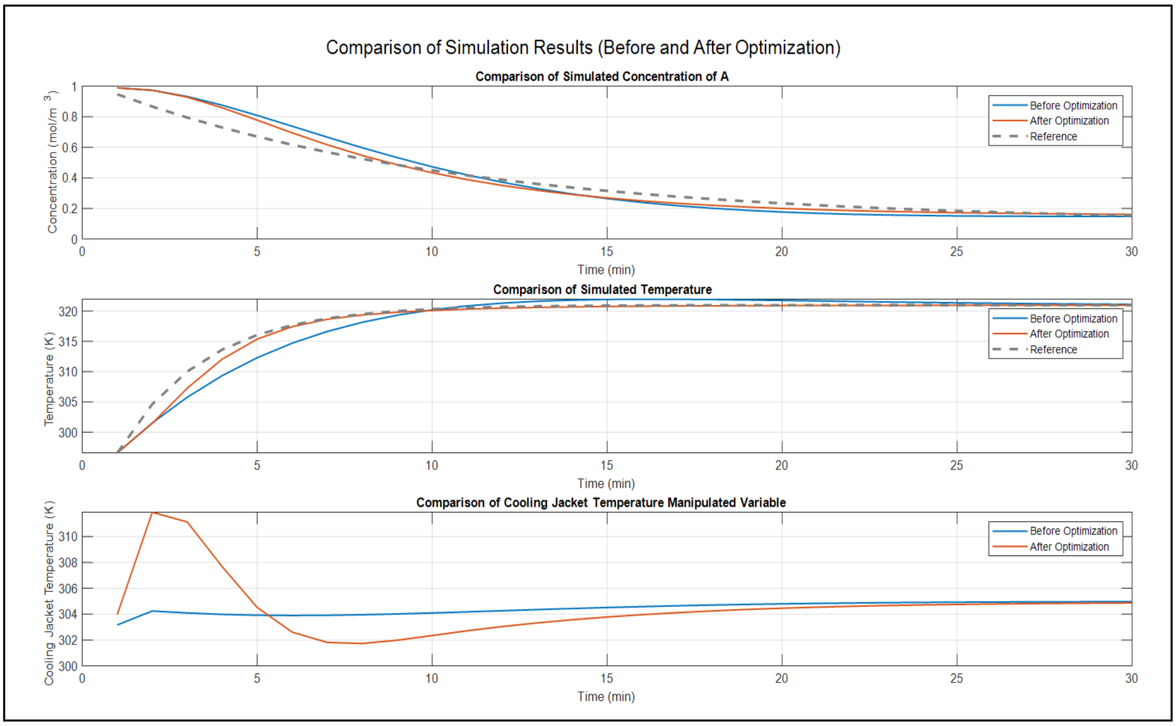

Fig.5. Control Performance Comparison

References

[1] B. W. Bequette, Process dynamics: modeling, analysis, and simulation. in Prentice Hall international series in the physical and chemical engineering sciences. Upper Saddle River, N.J: Prentice Hall PTR, 1998.

[2] A. D. Kakule and P. Kerkar, “Implementation of Temperature Regulation and Concentration Tracking of CSTR with Fuzzy Controllers,” in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India: IEEE, Jun. 2018, pp. 758–764. doi: 10.1109/ICCONS.2018.8663114.

[3] S. Baruah and L. Dewan, “A comparative study of PID based temperature control of CSTR using Genetic Algorithm and Particle Swarm Optimization,” in 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Dehradun: IEEE, Nov. 2017, pp. 1–6. doi: 10.1109/ICETCCT.2017.8280312.

[4] K. Mistry, P. Jani, and V. Tilva, “Comparative Analysis of Evolutionary Algorithms for a CSTR System,” in 2020 International Conference on Power, Instrumentation, Control and Computing (PICC), Thrissur, India: IEEE, Dec. 2020, pp. 1–6. doi: 10.1109/PICC51425.2020.9362479.

[5] U. V. Ratnakumari and M. B. Triven, “Implementation of adaptive model predictive controller and model predictive control for temperature regulation and concentration tracking of CSTR,” in 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India: IEEE, Oct. 2016, pp. 1–6. doi: 10.1109/CESYS.2016.7889843.

[6] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, 1st MIT Press ed. in Complex adaptive systems. Cambridge, Mass: MIT Press, 1992.

[7] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” IEEE Trans. Syst., Man, Cybern. B, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: 10.1109/3477.484436.

[8] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proceedings of ICNN’95 - International Conference on Neural Networks, Perth, WA, Australia: IEEE, 1995, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[9] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA: IEEE, 1998, pp. 69–73.

[10] M. Juneja and S. K. Nagar, “Particle swarm optimization algorithm and its parameters: A review,” in 2016 International Conference on Control, Computing, Communication and Materials (ICCCCM), Allahbad, India: IEEE, Oct. 2016, pp. 1–5. doi: 10.1109/ICCCCM.2016.7918233.

[11] D. E. Seborg, Process Dynamics and Control, Fourth edition. Hoboken, NJ: Wiley, 2017.

[12] Tejas D. Gangurde, Vrushali P. Mahajan, "Design of Type 2- Interval Fuzzy PID Controller for CSTR", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.11, No.10, pp. 23-29, 2019. DOI: 10.5815/ijigsp.2019.10.04

[13] S. N. Deepa and I. Baranilingesan, "Optimized deep learning neural network predictive controller for continuous stirred tank reactor," *Comput. Electr. Eng.*, vol. 71, pp. 782–797, Oct. 2018, doi: 10.1016/j.compeleceng.2017.07.004.

[14] K. S. Holkar and L. M. Waghmare, "An Overview of Model Predictive Control," *Int. J. Control Autom.*, vol. 3, no. 4, 2010.

[15] E. F. Camacho and C. Bordons, *Model Predictive control*. in Advanced Textbooks in Control and Signal Processing. London: Springer London, 2007. doi: 10.1007/978-0-85729-398-5.

[16] D. Pazoki, R. Roozbahani, and A. Nikoofard, "Comparison of MPC Algorithms for Continuous Stirred Tank Reactor," in 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China: IEEE, Apr. 2023, pp. 340–345. doi: 10.1109/ICCECT57938.2023.10140690.

[17] Bendib Riad .BatoutNoual .BentarziHamid . * Optimization Of The Membership Functions For AFuzzy Controller Using An Improved Genetic Algorithms* Algerian Journal of Signals and SystemsVolume 2, Numéro 4, Pages 240-247 2017-12-31 .DOI: https://doi.org/10.51485/ajss.v2i4.49

[18] BENDIB Riad, HAMMADI Youcef, MAZOUZI Mohammed, MECHHOUD Elarkam * Particle SwarmOptimization for tuning a Fuzzy Supervisory Controller Parametesrs (Takagi Seguno and MamdaniEngines) * Algerian Journal of Signals and Systems Vol. 5 No. 2 (2020) , doi.org/10.51485/ajss.v5i2.102.

[19] R. Bendib, N. Batout, "Boiler Flow Control using Optimal Fuzzy Supervisory PID Controller", 5thInternational Conference on Control, Decision and Information (CoDIT), 2018, Pages 370-373, IEEEpress. DOI: 10.1109/CoDIT.2018.8394865 .

[20] Bendib Riad (2017) Optimization and improvement of theoverall performance of an industriale Plant. PHD thesis,M'hamedBougara University Algeria.http://dlibrary.univ-boumerdes.dz:8080/handle/123456789/3420 [17].

[21] Boukrouma Houcem Eddine , Bendib Riad , Zennir Youcef (2024) Spurious Trip Rate Optimization Using Particle Swarm Optimization Algorithm "International Journal of Safety and Security Engineering DOI:10.18280/ijsse.410106