

GQF-PSO Algorithm for Temp Control Loop in Process Control

Jotiram Deshmukh¹, Ritesh Tirole², Anand Bhaskar³

¹Research Scholar, Department of Electronics and Communication Engineering, Sir Padampat Singhania University, Udaipur, Rajasthan, India & Bharati Vidyapeeth College of Engineering Navi Mumbai.
jotiram.deshmukh@bvcoenm.edu.in

²Department of Electrical Engineering, Sir Padampat Singhania University, Udaipur, Rajasthan, India.
ritesh.tirole@spsu.ac.in

³Department of Electronics and Communication Engineering, Sir Padampat Singhania University, Udaipur, Rajasthan, India. anand.bhaskar@spsu.ac.in

Article History:

Received: 26-09-2024

Revised: 27-11-2024

Accepted: 09-12-2024

Abstract:

In the real time of process optimization, the "Gaussian Quantum Firefly-PSO Algorithm for PID Controlled Temperature Loop of Process Plant" is an innovative method. Proportional-Integral-Derivative (PID) management of the temperature loop, firefly-inspired optimization, and quantum computing are all combined in this innovative technique. This algorithm offers the possibility of accurate and efficient optimization of individual loops by including Gaussian distributions and quantum behaviors into the firefly-PSO framework. This would guarantee improved performance, stability, and safety in intricate industrial processes. This introduction highlights the potential of this novel technique by summarizing the convergence of process control, firefly-inspired optimization, and quantum computing.

Keywords: Gaussian Quantum Firefly, PSO Algorithm, PID Controller

1. Introduction

By combining the ideas of quantum computing with traditional optimization methods, quantum-inspired metaheuristic algorithms provide a fresh method for resolving challenging optimization issues. These algorithms use the ideas of quantum physics as inspiration to handle a variety of issues in fields like machine learning, cryptography, and combinatorial optimization. The many fundamental ideas of Quantum Bits (Qubits) are expanded upon by quantum-inspired metaheuristic algorithms [5].

Quantum Gates [4], and Quantum Entanglement [3]. Numerous quantum-inspired metaheuristic algorithms, such as the Quantum-Inspired Genetic Algorithm, have drawn interest in the literature etc. [2], are the several fields in which research on quantum-inspired metaheuristic algorithms is being conducted.

Despite its potential, quantum-inspired metaheuristic algorithms have a few drawbacks [6], including real-world applications, hybrid approaches, hardware limitations, and parameter tuning.

Particle Swarm Optimization (PSO) [7] is a well-known optimization method that draws inspiration from fish and bird collective behaviour. Particles in a search space modify their placements according to their own and their neighbours' experiences in the typical PSO algorithm. This technique is improved by the Gaussian PSO algorithm, which adds Gaussian distributions to allow for a more thorough investigation of the solution space.

By adding Gaussian distributions, the Gaussian PSO method [8] expands upon the fundamentals of PSO. Particle movement and probabilistic exploration are important ideas. The main areas of research on Gaussian PSO are: Hybridization with Other Algorithms [9], Gaussian PSO in Feature Selection [8], and Gaussian PSO for Function Optimization [8]. Several engineering issues can be optimized using Gaussian PSO [8]. Hyper-parameter optimization for machine learning models has been achieved with the application of Gaussian PSO [10]. To identify trade-off solutions for competing purposes, researchers have investigated the application of Gaussian PSO in multi-objective optimization issues [11]. Gaussian PSO's future challenges [8] include scalability and parameter tuning.

One relatively new contribution to the category of quantum-inspired optimization methods is Quantum Particle Swarm Optimization (Q-PSO) [12]. It blends the optimization methods of classical PSO with the ideas of quantum computing. The Q-PSO algorithm has attracted a lot of attention because of its capacity to effectively handle challenging optimization issues.

The foundational ideas of quantum computing and classical PSO, such as quantum bits (Qubits) [5], quantum gates [4], and interference and superposition [13], are expanded upon by quantum PSO. For solving single-objective optimization issues [1], multi-objective optimization [11], parameter tuning [7], and quantum speedup [14], quantum PSO has been extensively researched. Applications of quantum PSO can be found in structural optimization [15], engineering design, and aerodynamics [16] to solve intricate design issues involving the optimization of numerous variables. Machine learning models can perform better by optimizing hyper-parameters with quantum PSO [10].

Additionally, it is employed to improve data analysis in feature selection [8]. Because to its quantum-inspired design, the technique can also be used in cryptography, particularly to solve cryptographic riddles [17] and streamline the encryption and decryption procedures. To enhance performance and more successfully address real-world issues, researchers are investigating hybrid systems that combine Q-PSO with traditional optimization techniques [9]. Research is still being done to provide new quantum-inspired heuristics and operators for Q-PSO to increase the robustness and efficiency of the algorithm.

The flashing behaviour of fireflies during mating is the source of inspiration for the Firefly method (FA) [18], a nature-inspired optimization method. Since its introduction by Xin-She Yang in 2008, the FA has become well-liked in a variety of fields because of its effectiveness and ease of use.

A few basic concepts, including Attraction and Light Intensity, Attractiveness and Distance, and Light Absorption and Randomness, form the foundation of the Firefly Algorithm. Particularly in addressing uni-modal, multimodal, and complicated optimization challenges, the FA has been thoroughly researched for function optimization problems. There have already been suggestions to modify the FA for multi-objective optimization situations. These adjustments aid in identifying a collection of solutions that reflect compromises between several competing goals. Additionally, the Firefly Algorithm has been used to solve practical issues like portfolio optimization [20], scheduling [19], and engineering design. It is appropriate for a variety of applications because to its robustness and efficiency. The FA has been used to optimize network design [21], including the deployment of wireless sensor networks, where effective communication and coverage are essential.

2. Objectives

- 1.Design a Gaussian quantum Firefly PSO Algorithm for temperature control loop.
- 2.Develop a model for Temperature control loop.
- 3.Integrate the Gaussian quantum Firefly PSO Algorithm for optimization of PID Controller.
- 4.Improve time domain specifications using Gaussian quantum Firefly PSO Algorithm.

3. Methods

To improve the accuracy of K_p , K_i , and K_d parameter optimization, a hybrid optimization strategy is suggested based on the literature review.

3.1 GQF-PSO PID Controller for Temp Loop

To optimize the controlling operation with a PID controller, the temperature loop of a multi-loop pilot plant is examined, and the transfer function for the loop is derived. For the optimization of K_p , K_i , and K_d parameters, a hybrid strategy that combines the advantages of Gaussian optimization, quantum optimization, and firefly optimization is suggested. The details of the PID control system and temperature loop under study are covered in the section that follows. To get better results than those found in the literature, it also incorporates the suggested methods for optimizing PID settings.

3.2 Tuning PID controller for Temperature loop.

To analyses and regulate the different control loops that regulate the many parts of the process, including the temperature loop, a process plant's temperature loop (Fig. 1) is taken into consideration.

3.3. Determination of Transfer Function of Temperature loop

Each non-interactive loop must be adjusted to guarantee optimal performance and to satisfy process requirements, since adjusting individual loops separately can increase control system robustness and lessen the impact of interactions when they happen. Fig. 2 displays individual non-interactive control loops. The transfer functions for each non-interactive control loop are defined in the following subsections, which also modify the control system described in [22].

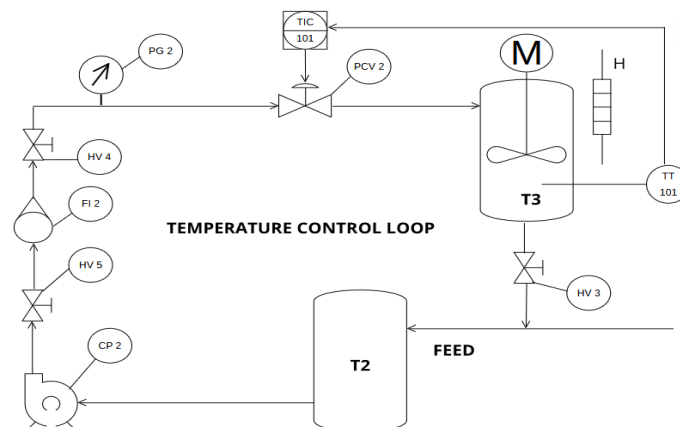


Fig.1. Temperature Loop.

3.4. Temperature Loop

The system's temperature loop is displayed in Hardware Design's Fig. 1. It is made up of a controller and a temperature indicator (RTD). Control Valve 2 and 3kW power heaters are the last control components.

a) Temperature Loop Transfer function

We shall examine a basic continuous stirred tank heater to ascertain the temperature loop's transfer function. The primary concern in the temperature loop is how the system will react to modifications in the input conditions, such as disturbances or changes in the controlled variable. The mass and energy content (state variables) of the tank would be impacted by changes in the input circumstances, which are the inlet flow rate and temperature. There is no progression of momentum in the tank under normal circumstances since variations in the inlet flow's temperature or flow rate do not have the capacity to physically move the tank from its typical location.

3.5. PID Tuning for Temp Control Loop

The steps involved in PID tuning are as follows.

a) Steady-State Analysis: Every non-interactive loop must undergo a steady-state analysis prior to tuning. Gain knowledge of the dynamics of the process and the connection between the input (manipulated variable) and the output (controlled variable).

b) Determine Control Objectives: Specify Temperature loop control objective. Which performance indicators (such as settling time, overshoot, and steady-state error) are crucial, and what is the desired set-point?

c) Initial Tuning: - If available, start with the original PID settings. Alternatively, start with standard starting values (e.g., $P=1$, $I=0$, $D=0$). To see how the system reacts, put the controllers in manual mode.

d) Step Testing: For every non-interactive loop, run step tests. After making a step modification to the set point, see how the process variable reacts. Gather information to ascertain the dynamics of the system.

e) Model Identification: Choose an appropriate model for the non-interactive loop using the step test data. The tuning procedure can be guided by this model.

f) Tuning Methods: PID controllers can be tuned using a variety of approaches, such as Ziegler-Nichols methods, manual tuning, and more sophisticated strategies like the Internal Model Control (IMC) or Cohen-Coon methods. The complexity of the system and the intended control goals determine which approach is best.

g) Iterative Tuning: Apply the chosen tuning technique, then examine the system's reaction. Iteratively change the P, I, and D parameters while monitoring the system's performance and making necessary adjustments.

h) Final Validation: Verify the performance under typical operating settings once the temperature loop has been satisfactorily controlled. To guarantee stability and ideal control, keep an eye on the system for a fair amount of time.

A methodical and iterative procedure is required to tune a PID controller for a temperature loop in an industrial plant. Individual tuning of each loop is necessary to achieve certain control goals and guarantee peak performance. A multi-loop control system's long-term performance depends on careful evaluation of the dynamics and interactions of the processes as well as routine maintenance and re-tuning.

3.6. GQF-PSO algorithm for PID tuning

There are several problems with PSO algorithms. 1) It takes more iterations to reach ideal values, and 2) ideal parameters are frequently not reached. Therefore, hybridizing several optimization techniques can yield the optimum optimization. Based on the benefits covered in section I, several hybridisable algorithms are evaluated, and the hybridization combining the effects of quantum theory, Gaussian optimization, and Firefly optimization is taken into consideration. The suggested technique employs the Quantum Gaussian Firefly optimization to converge the best-optimized parameters in the PSO feedback loop whenever the best results are not converging at the necessary pace using PSO, as illustrated in Fig. 3.

The suggested Quantum Gaussian Firefly PSO algorithm is used to optimize the estimated initial parameters (K_p , K_i , and K_d) for the PID controller. The PID controller is controlled by the final optimized parameters.

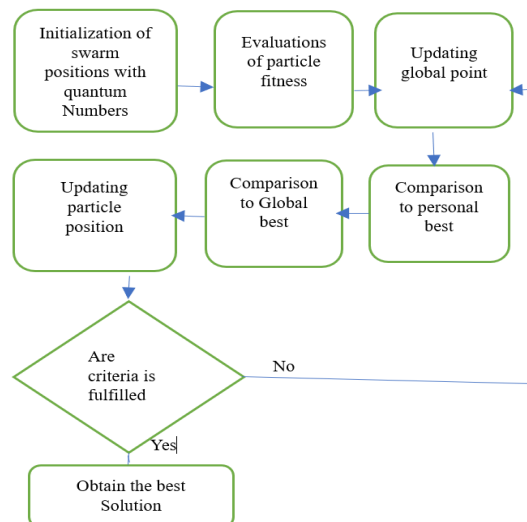


Fig.3. Flow diagram for Proposed GQF-PSO algorithm

4. Results

4.1. Simulink model

A PID controller with delayed feedback is used in conjunction with a model of plants for the temperature loop to track the plants' responses. Temperature and loop Simulink modules are created, and reactions are tracked. Figures 4(a) and 4(b) display the temperature loop Simulink modules with parameter setup.

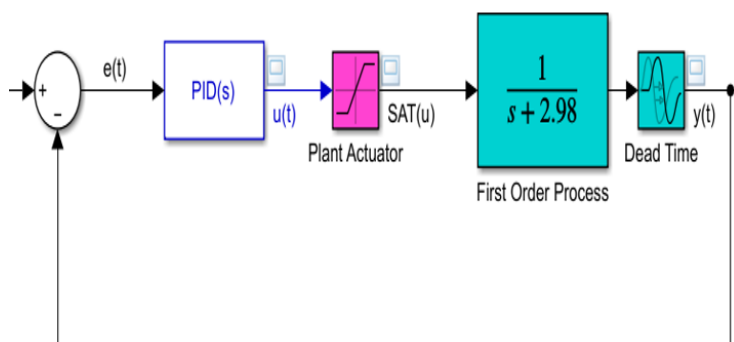


Fig.4. Temperature Loop Setting for PID parameters and setting of First order

(b) Setting for First Order Process plant

Figure 5(a) displays the step response of the PID temperature control loop with the K_p , K_i , and K_d parameters tuned using the PSO algorithm, while Figure 5(b) displays the step response of the suggested algorithm.

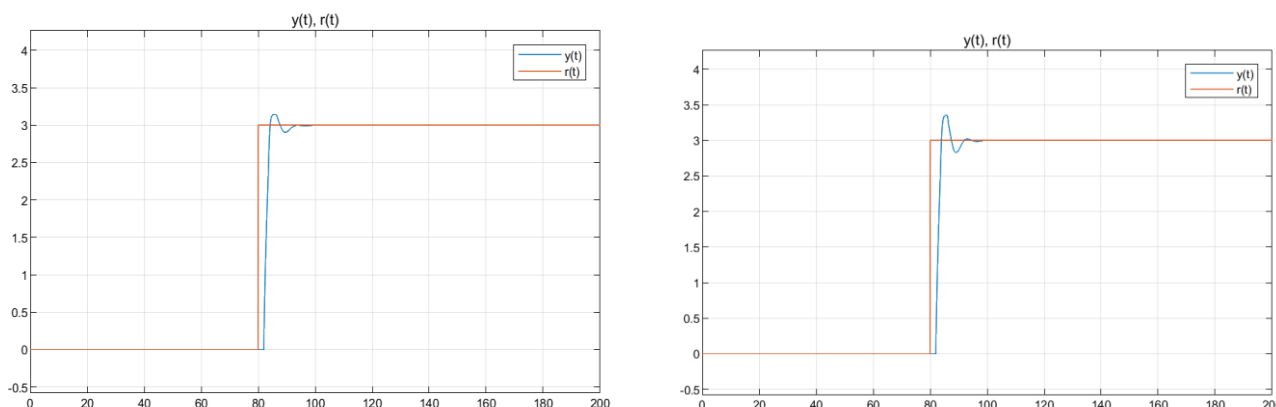


Fig.5. Step response of temperature loop a) using PSO tuning algorithm, b) using proposed tuning algorithm

Table 1. list the PSO and suggested GQF-PSOC tuned K_p , K_i , and K_d parameters of PID under temperature loop. To obtain the step response, those adjusted parameters are applied to the PID controller of the relevant Simulink model. Table 2. contains the response parameters that were taken from the responses.

Table 1: Optimized PID parameters using PSO and proposed algorithm for temperature loop

Algorithms Parameters	PSO	GQFPSO
K_p	1.1255	1.3965
K_i	1.2406	1.0983
K_d	0	-0.9189

Table 2: Step response parameters using PSO and proposed algorithm for temperature loop

Algorithms Parameters	PSO	GQFPSO
Rise Time	16.078 sec	15.165 sec
Settling Time	1.617 sec	1.501 sec

4.2. Error Functions used for Evaluation

The error functions Dejong1, Drop wave, Schwefel, Rastrigin, Dejong5, Bukin6 are used to find the optimised performance. All are described in this section.

Training of the proposed algorithm is performed using different error functions and the rate of reduction of error function parameters is observed. The best suitable error function is decided by iterating the proposed algorithm for 50 iterations. Experimental results are given in Table 5. Table 5 concludes that the Bukin6fcn function is best suitable for the GQF-PSO algorithm.

Table 5: Error function performance

Iteration	5	10	15	20	25	30	35	40	45	50
Dejong 1	402.24 37	48.128 9	47.38 51	38.94 64	34.19 57	30.31 18	23.78 13	5.468	4.9345	3.9113
Drop wave	7.7523	3.4405	2.496 7	0.701 95	0.602 72	0.470 41	0.299 87	0.2975 5	0.2306 1	0.2254 1
Schwefel	228.15 01	77.216	50.11 93	35.06 44	19.91 51	11.59 77	8.913 7	5.498	4.9345	3.913
Rastrigin	68884 821.8	204798 9.315	2676 9.65	21.04 77	21.04 77	24.04 93	24.04 93	8.6514	6.0159	4.8738
Dejong 5	1.0140	1.0140	0.998 81	0.998 05	0.998 05	0.977	0.997	0.997	0.997	0.997
Bukin6 (Best Function)	0.0141 1	0.0141 1	0.011 537	0.002 5745	0.002 5745	0.000 8567	0.000 8279	0.0007 4379	0.0006 8248	0.0006 6623

5. Discussion

An important development in the field of process optimization is the creation of the Gaussian Quantum Firefly-PSO Algorithm for PID Controller Temperature Loops of a Multi-loop Pilot Plant. By combining the concepts of quantum computing, optimization methods inspired by fireflies, and PID control strategies, this new program provides a potent remedy for the complex problems associated with controlling non-interactive loops in multi-loop pilot plants.

Our study has shown how effective this novel strategy is at improving industrial processes' stability, performance, and safety. With the use of Gaussian distributions and quantum behaviors in the firefly-PSO framework, the technique makes it possible to optimize individual loops precisely and efficiently, which improves system performance.

The algorithm's ability to transform process control approaches and open the door for more resilient and flexible optimization strategies in industrial settings is further demonstrated by its successful implementation. The nexus of process control, firefly-inspired optimization, and quantum computing is a research frontier with significant ramifications for several businesses that depend on intricate process systems.

The Gaussian Quantum Firefly-PSO Algorithm has enormous potential to advance the capabilities of process optimization technologies if it is further explored and improved. Future studies could concentrate on expanding the algorithm's use in various industrial settings and improving its scalability and adaptability to consider changing process dynamics.

In summary, the Gaussian Quantum Firefly-PSO Algorithm is a noteworthy technological accomplishment that also demonstrates the potential of interdisciplinary cooperation to spur innovation in the process optimization space. We can fully utilize this innovative strategy to improve the sustainability.

References

- [1] Peres, Fernando, and Mauro Castelli. 2021. "Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development" *Applied Sciences* 11, no. 14: 6449. <https://doi.org/10.3390/app11146449>
- [2] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," *Proceedings of IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, 1996, pp. 61-66, doi: 10.1109/ICEC.1996.542334.
- [3] Ludmil Hadjiivanov and Ivan Todorov, "Quantum entanglement," 1506.04262, arXiv, 2015, <https://doi.org/10.48550/arXiv.1506.04262>.
- [4] Quantum entangling gates using three and four qubits. *Nat. Phys.* 19, 1396–1397 (2023). <https://doi.org/10.1038/s41567-023-02105-4>.
- [5] <https://gaurikhard.medium.com/concepts-of-qubits-d561503b1204>, visited on 01/02/2024.
- [6] Rajwar, K., Deep, K. & Das, S. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artif Intell Rev* 56, 13187–13257 (2023). <https://doi.org/10.1007/s10462-023-10470-y>.
- [7] Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch Comput Methods Eng* 29, 2531–2561 (2022). <https://doi.org/10.1007/s11831-021-09694-4>.
- [8] Lane, M.C., Xue, B., Liu, I., Zhang, M. (2014). Gaussian Based Particle Swarm Optimisation and Statistical Clustering for Feature Selection. In: Blum, C., Ochoa, G. (eds) *Evolutionary Computation in Combinatorial Optimisation. EvoCOP 2014. Lecture Notes in Computer Science*, vol 8600. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44320-0_12.
- [9] Yu, Hongli, Yuelin Gao, Le Wang, and Jiangtao Meng. 2020. "A Hybrid Particle Swarm Optimization Algorithm Enhanced with Nonlinear Inertial Weight and Gaussian Mutation for JobShop Scheduling Problems" *Mathematics* 8, no. 8: 1355. <https://doi.org/10.3390/math8081355>.
- [10] L. Kang, R. -S. Chen, N. Xiong, Y. -C. Chen, Y. -X. Hu and C. -M. Chen, "Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things," in *IEEE Access*, vol. 7, pp. 59504-59513, 2019, doi: 10.1109/ACCESS.2019.2913757.
- [11] Susana C. Esquivel and Leticia Cecilia Cagnina, "Multi-objective optimization with a Gaussian PSO algorithm," 2008, <https://api.semanticscholar.org/CorpusID:59574247>.
- [12] Shuyuan Yang Min Wang, and Licheng Jiao, "A quantum particle swarm optimization," *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, Portland, OR, USA, 2004, pp.320-324 Vol.1, doi:10.1109/CEC.2004.1330874.
- [13] *Advanced Quantum Computing: Interference and Entanglement*, https://www.worldscientific.com/doi/10.1142/9781786348210_0004.
- [14] Pokharel, Bibek and Lidar, Daniel A., "Demonstration of Algorithmic Quantum Speedup," *Physical Review Letters*, vol. 130, no 21, May 2023.
- [15] D. Prayogo et al, "Structural Design Optimization Using Particle Swarm Optimization and Its Variants," 2020 IOP Conf. Ser.: Earth Environ. Sci. 506 012048, DOI 10.1088/1755-1315/506/1/012048.

- [16] Chen-chao Xia, Ting-ting Jiang, and Wei-fang Chen, "Particle Swarm Optimization of Aerodynamic Shapes with Non-uniform Shape Parameter-Based Radial Basis Function," *Journal of Aerospace Engineering*, Volume 30, Issue 3, [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0000686](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000686).
- [17] Biondi Santi, C., Vespri, V. (2021). Solving Cryptographic Puzzles: How to Mine? In: Cappiello, B., Carullo, G. (eds) *Blockchain, Law and Governance*. Springer, Cham. https://doi.org/10.1007/978-3-030-52722-8_5.
- [18] W. A. Khan, N. N. Hamadneh, S. L. Tilahun, and J. M. T. Ngnotchouye, 'A Review and Comparative Study of Firefly Algorithm and its Modified Versions', *Optimization Algorithms - Methods and Applications*. InTech, Sep. 21, 2016. doi: 10.5772/62472.
- [19] A. P. Napalit and M. A. Ballera, "Application of Firefly Algorithm in Scheduling," 2021 IEEE International Conference on Computing (ICOCO), Kuala Lumpur, Malaysia, 2021, pp. 336-340, doi: 10.1109/ICOCO53166.2021.9673581.
- [20] Lazulfa, I. (2019). A Firefly Algorithm for Portfolio Optimization. *Journal of the Indonesian Mathematical Society*, 25(3), 282–291. <https://doi.org/10.22342/jims.25.3.821.282-291>.
- [21] I. Czarnowski and P. Jędrzejowicz, "Firefly Algorithm for the RBF Network Design," 2018 Innovations in Intelligent Systems and Applications (INISTA), Thessaloniki, Greece, 2018, pp. 1-6, doi: 10.1109/INISTA.2018.8466264.
- [22] Saurabh U. Mohite, Abhishek P. Patil, Shreyas D. Patil, D. N. Pawar, "Mathematical Modeling and Simulation of Multi-Loop Pilot Plant," *International Journal of Computer Sciences and Engineering*, Vol.7, Issue.6, pp.769-774, 2019.
- [23] <http://www.geatbx.com/docu/fcnindex-01.html>, visited on date 19-07-2024.
- [24] <https://www.sfu.ca/~ssurjano/drop.html>, visited on date 19-07-2024.
- [25] <https://www.sfu.ca/~ssurjano/schwef.html>, visited on date 20-07-2024.
- [26] <https://www.sfu.ca/~ssurjano/rastr.html>, visited on date 21-07-2024.
- [27] <https://www.sfu.ca/~ssurjano/dejong5.html>, visited on date 21-07-2024.
- [28] <https://www.sfu.ca/~ssurjano/bukin6.html>, visited on date 21-07-2024.