UDC 004.896
IRSTI 28.23.27

https://doi.org/10.55452/1998-6688-2025-22-2-76-93

[1]Samigulina Z.I.,
PhD, Associate Professor, ORCID ID: 0000-0002-5862-6415,
e-mail: z.samigulina@kbtu.kz
[1]*Amangaliyeva A.G.,
Bachelor, ORCID ID: 0009-0003-7991-9836,
*e-mail: a_amangaliyeva@kbtu.kz

[1]Kazakh-British Technical University, Almaty, Kazakhstan

OPTIMIZATION OF PID CONTROLLER PARAMETERS USING MACHINE LEARNING ALGORITHMS BASED ON OIL SEPARATION PROCESS DATA

Abstract

This paper presents the investigation of the process of optimizing the parameters of a PID controller using machine learning algorithms for the oil separation process control system. The optimization of the controller parameters (Kp, Ki, Kd) is important, in order to improve control quality and reduce the number of errors in dynamic processes. To solve this issue, several innovative methods were considered, such as the cuckoo search algorithm (CSA), the firefly algorithm (FA), particle swarm optimization (PSO), and the support vector machine (SVM). All the data, including the current process values (PV), setpoints (SP) and output signals (OP) were obtained from Tengizchevroil. In addition, the metrics, such as root-mean-square error (MSE), adjustment time, overshoot, and steady-state error were used to assess the effectiveness of optimized regulators. Overall, the results of the research indicate that there was a significant improvement of the dynamic characteristics of the system due to the usage of machine learning algorithms compared to the traditional approaches. The obtained parameters of optimization achieved the target value while being faster and more stable, thus increasing the productivity of control in the technological process.

Keywords: oil separation, automation system optimization, PID controller, parameter optimization, machine learning, Cuckoo Search Algorithm, Firefly Algorithm, Particle Swarm Optimization, Support Vector Machine.

Introduction

In recent years, the issues related to improving the efficiency and stability of the technological processes are prominent in the oil and gas industry. Oil and gas separation, where the precise control of process parameters plays the vital role is considered as one of the key processes in this industry. Due to its simplicity and versatility the proportional integral differential (PID) regulator is widely used as the main control tool in these systems. However, the limitations of the traditional methods of setting the parameter of the PID controllers decreases its effectiveness, thus leading to the increase in costs, significant errors and lower quality of the product.

Over the years, there has been an increase in the widespread usage of machine learning methods such as optimization algorithms in the field of automation. Algorithms, including particle swarm Optimization (PSO), the Firefly algorithm, the Cuckoo Search Algorithm and the support vector Machine (SVM) method allow for high accuracy in estimation, including through optimization. These methods take into consideration the dynamic characteristics of the processes and make effective adjustments to the parameters of PID controller. Consequently, it results in more precise control and significant decreases in the number of errors.

The aim of work is to use rare machine learning algorithms based on data from the oil separation processes to develop and apply an approach for the optimization of the parameters of PID controller.

The real equipment indicators are used as input data: the values of (point value), SP (setpoint value) and OP (output).

Further, these data are applied to build models and evaluate their performance. By optimizing the parameters of the PID controller, it is expected that the overall quality of control and the stability of the system despite the external disturbances will increase, while the errors will be minimized.

Thus, this work is aimed at combining modern machine learning methods and traditional management approaches to solve urgent issues of the oil and gas industry. The results of the study can be applied to improve the operation of automation systems in real production conditions.

The oil and gas industry is defined by complex processes that require accurate control to provide efficiency, safety and environmental compliance. One of the most important elements of equipment in this industry is a three-phase separator, which plays a crucial role in the separation of oil, gas and water in the production process [1]. To maintain stable and efficient operation of such separators, it is important to implement advanced control systems capable of controlling dynamic changes in the technological control process built by sensors of physical values like temperature, pressure and level [2, 3]. The Proportional-Integral-Derivative (PID) controller is one of the commonly used feedback-based control loop mechanisms due to its simplicity and effectiveness in managing production processes and machines [3, 4]. Yet the performance of the PID controller largely depends on the correct tuning of its parameters [5].

Traditional PID tuning methods, such as the Ziegler-Nichols method and manual trial-and-error approaches, often lead to admittedly not optimal output, especially in nonlinear processes and time-varying processes such as three-phase separation [6, 7]. Various types of control strategies used for PID tuning discussed in [8]. This study compared Integral Absolute Error (IAE) values considering first, second and third order systems. The method has limitations to Single Input Single Output systems. The closed-loop Ziegler Nichols methods was studied, and its limitation was that this methos is not applicable for open-loop systems which are not stable. It involves trial-and-error method to select the parameter, so it is time consuming. The Chien-Hrones-Reswick auto tuning technique was also considered in the paper. This method delivers an overshoot system response in the range 10–20% however it provides a fast response.

In it could be seen the comparison of two types of PID controllers [9]. This study conducts a comparative analysis between a fractional-order PID (FO-PID) controller and a standard PID controller for a nonlinear robotic arm manipulator system. The tuning of the controllers' gain parameters is achieved through a genetic algorithm (GA), which optimizes the controllers based on various cost functions, including integral of squared error (ISE), integral of absolute error (IAE), integral of time-weighted absolute error (ITAE), and integral of time-weighted squared error (ITSE). The study uses MATLAB/SIMULINK simulations to evaluate the controllers' performances under different operational scenarios of the robotic arm.

The article proposes the use of the Atom Search Optimization (ASO) algorithm and its chaotic variant, Chaotic Atom Search Optimization (ChASO), to optimize parameters of the fractional-order PID (FOPID) controller for DC motor speed control [10]. ASO, inspired by atomic motion models, is valued for its simplicity and effectiveness in addressing various optimization challenges. ChASO, an enhancement using logistic map chaotic sequences, aims to improve convergence speed and escape local minima, providing more precise results. Paper implemented PID with Intern Model Control (IMC) method, which loaded into a PIC microcontroller to control the level through varying the liquid flow [11]. The results were satisfactory as the system response had fewer oscillations, less settling and rise times, and there was no steady state offset. In a study, an improved Cuckoo Search algorithm for detecting intrusions in information systems was proposed [12]. The authors modified the classical algorithm in order to increase the accuracy and speed of optimization, which made it possible to effectively analyze data and identify anomalies.

To solve these problems, researchers are increasingly exploring the possibility of using optimization algorithms and methods to automate the tuning [13, 14, 15]. Recent research shows that using rare algorithms can further improve the tuning process, providing higher convergence rates, reliability and adaptability to the dynamics of a complex system, while common optimization algorithms such as genetic algorithms and Particle Swarm optimization have shown promise [16, 17, 18].

This study addresses the complex trajectory tracking challenges of a three-link rigid robotic manipulator (3-LRRM) by designing and comparing three neural network-based control structures combined with PID: the NN-PIPD controller, the NN+PID controller, and the ELNN-PID controller [19]. Using the Coot Optimization Algorithm (COOA), each controller's parameters are tuned to minimize the integral time square error (ITSE), with a novel objective function specifically aimed at reducing control signal chattering. Evaluations under various scenarios, including disturbance rejection, model uncertainties, and reference tracking, demonstrate that the NN-PIPD controller provides the best performance, excelling in stability, robustness, and precision in tracking, with an exceptionally low ITSE value of 0.001777. One of the rare algorithms called Jellyfish search studied in paper [20]. This study introduces a modified version of the Jellyfish Search (JS) algorithm, termed the modified Jellyfish Search (mJS) algorithm, to optimize PID controller parameters for DC motor speed control. While the original JS algorithm is effective, its exploitation capabilities are limited. To enhance performance, the mJS incorporates quasi-dynamic opposed-based learning and a Weibull probability distribution to improve convergence and precision. The optimization goal is set to minimize the integral of time-weighted absolute error (ITAE). The mJS algorithm was validated on benchmark functions, showing superior performance compared to contemporary optimization methods, including Gray Wolf Optimization (GWO), JAYA, and Golden Jackal Optimization (GJO). In simulation results across three DC motor models, the mJS algorithm consistently achieved lower ITAE values, faster settling times, and improved response stability compared to other algorithms, demonstrating its potential in industrial control applications and contributing to advanced PID optimization methodologies.

The study uses five nature-inspired algorithms–NewBAT, Cuckoo Search (CS), Firefly (FF), Gray Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA)–to optimize the FOPID parameters: Kp, Ki, Kd, and μ [21]. The optimization objective is minimizing the integral of time absolute error (ITAE) while also evaluating maximum overshoot, settling time, and time to maximum response. Simulations carried out in MATLAB/Simulink with the FOMCON toolbox demonstrated the controllers' robustness against disturbances at the output. The optimized parameters were tested against varying reference inputs, and the controllers' responses were assessed with visual and statistical analyses, offering a reliable basis for selecting effective algorithms and measuring performance. This research contributes to optimizing FOPID controllers in time-delayed systems by providing comparative insights on the efficacy of different nature-inspired algorithms, validated through statistical methods for robustness and reliability.

Fundamental methods such as Ziegler–Nichols, Cohen–Coon, Chien–Hrones–Reswick, and Aström–Hagglund are used to adjust the proportional, integral and differential parameters of PID. In addition, hybrid methods such as fuzzy-tuned PID controllers and optimization algorithms can be used to adjust these parameters. Various optimization algorithms, such as the ABC optimization algorithm, gray wolf optimization algorithm, evolutionary optimization algorithms, particle swarm optimization, Harris Hawks optimization algorithm, and Harris Hawks gas optimization algorithm, can be used for parameter tuning [22–26]. Sometimes, the desired response may not be achieved with PID controllers. To overcome this problem, alongside classical PID controllers, more advanced PID controllers have been used [27]. There are advanced controllers such as 2DOFPID, tilt-integral derivative (TID), FOPID and PTID [28, 29]. By adding degrees to the PIDs, changing their basic structures, or using them in partial formats, attempts have been made to obtain a better response.

Description of the technological process of oil separation

Oil is a complex mixture of hydrocarbons of various molecular weights, other chemical compounds, various gaseous, liquid and solid substances containing more than 100 carbon atoms, oxygen, etc., a natural liquid with a peculiar odor, consisting of heterogeneous sulfur compounds and a mixture of metals. Oil is the most important type of mineral found in the sedimentary layer, which is oily, brown, flammable, sometimes black or greenish yellow, and even colorless. An integrated oil treatment plant is required to receive oil well products, to pre-separate products (into oil, associated petroleum gas, and formation water) and to treat oil to commercial quality. The work examines the separation unit at the complex oil treatment unit, so the process control system should provide: automated real-time control of the separation process at the complex preparation plant; in case of emergencies – notification of the operator about this; indicators of device states; management of the necessary parameters; The main goals of creating an automated process control system: increasing the accuracy of measuring process parameters; increase in staff efficiency; reduction of labor costs in process control; optimization of working conditions. System composition: three-phase separator; flow meters; level indicators; pressure sensors; level gauges; temperature sensor; actuators. This scheme in Figure 1 represents the process of processing and separation of extracted products consisting of oil, gas and water.

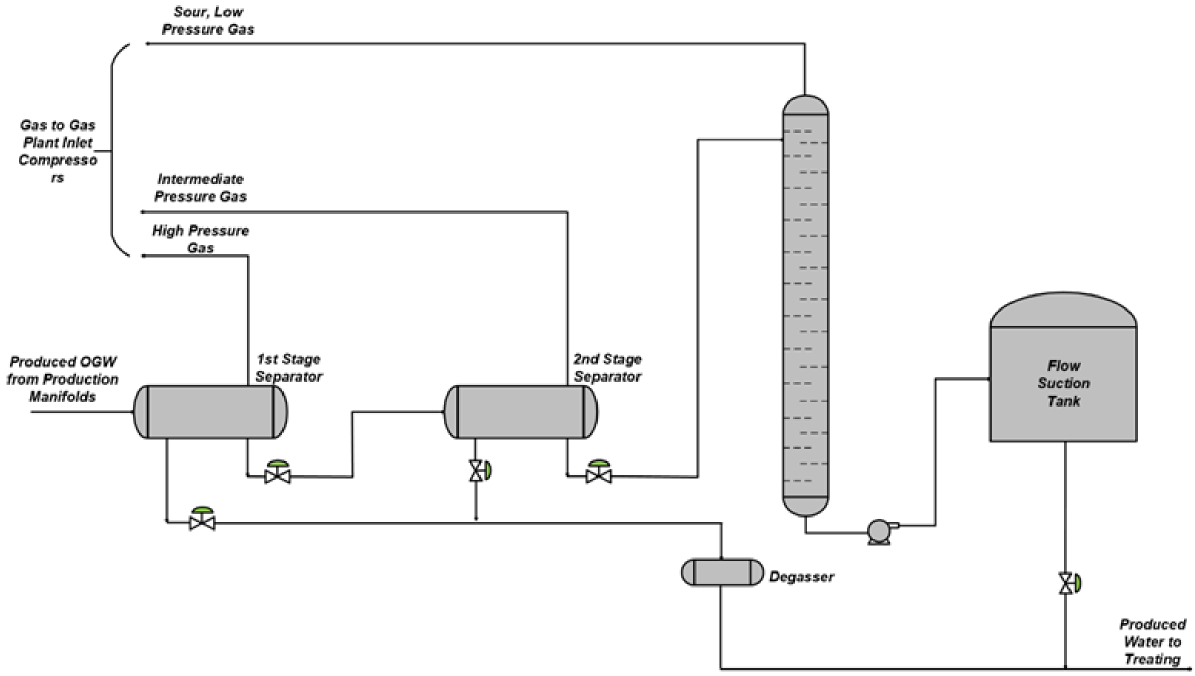

Figure 1 – Principal scheme of the oil separation process

Products from the extraction bushes (OGW) are supplied to the entrance, which then goes through several stages of separation. At the first stage, a first-level separator (1st Stage Separator) is used, where the primary separation of the liquid and gas phases takes place. The liquid is sent for further processing, and the high-pressure gas is discharged for subsequent transportation. Then the liquid enters the second-level separator (2nd Stage Separator), where a more detailed phase separation takes place, including the separation of medium-pressure gas.

After the second stage, the separated liquid enters the Degasser, which removes the residual gas to prepare the liquid for shipment to the storage tank (Flow Suction Tank). The gas released at each stage is sent to different gas lines depending on the pressure: low, medium or high. The water separated from the oil is sent to the water treatment system (Produced Water to Treating), and the oil is prepared for further transportation. The gas flows are directed either to the plant for processing, or to the compressors for further pressure increase. The diagram demonstrates the process of multi-stage separation of products, including gas, liquid and water treatment systems, ensuring processing efficiency and minimizing losses.

Materials and Methods

Optimization algorithms: Cuckoo Search Algorithm

The Cuckoo Search Algorithm (CSA) is a metaheuristic optimization algorithm inspired by the behavior of cuckoos that lay their eggs in the nests of other birds. The algorithm is based on a random search mechanism, using the Lévy flight method to generate new solutions. The main idea of the CSA is to replace bad solutions (sockets) with better ones based on their fitness. CSA is better suited for tasks that require global search but require fine tuning. Pseudocode for its implementation:

Pseudocode 1 – Cuckoo Search Algorithm

```
        Input: f(x), n, MaxGeneration, pa
        Output: x_best, f(x_best)
 BEGIN
Objective function f(x), x = (x1, x2, ..., xd)^T
     Generate an initial population of n host nests xi (i = 1, 2, ..., n), each containing a random solution;
WHILE (t < MaxGeneration) or (stop criterion not met) DO
        Generate a new solution using Lévy flights for a randomly chosen cuckoo;
                        Evaluate its fitness Fi;
                        Choose a nest randomly among n nests (say j);
                 IF (Fi > Fj) THEN
                Replace nest j with the new solution;
                              END IF;
                A fraction (pa) of the worst nests are replaced with new random solutions;
                        Keep the best solutions (nests with high-quality solutions);
                        Rank the solutions and find the current best;
                Pass the current best solutions to the next generation;
     END WHILE;
                 Return the best solution found;
END;
```

Optimization algorithms: Firefly Algorithm

The Firefly Algorithm (FA) is a metaheuristic algorithm inspired by the bioluminescence of fireflies. The main idea of the algorithm is that fireflies are attracted to each other based on their brightness, which correlates with the fitness of the solution. Brighter fireflies attract less bright ones, which leads to the evolution of the solution population to an optimal value. FA is characterized by simplicity of implementation and efficiency in solving multidimensional optimization problems. Pseudocode for its implementation:

Pseudocode 2 – Firefly Algorithm

```
        Input: f(x), n, β, γ, α, MaxGeneration
        Output: x_best, f(x_best)
 BEGIN
    Objective function f(x), x = (x1, x2, ..., xd)^T
    Initialize a population of n fireflies xi (i = 1, 2, ..., n);
    Define light intensity Ii proportional to f(xi);
    Set parameters: attractiveness β, absorption coefficient γ, randomization α;
WHILE (t < MaxGeneration) or (stop criterion not met) DO
        FOR i = 1 to n DO
            FOR j = 1 to n DO
                IF (Ij > Ii) THEN
```

```
            Move firefly i towards firefly j using the attractiveness β;
            Add random perturbation to the position of firefly i;
          END IF;
        END FOR;
      END FOR;
      Update light intensity based on the new solutions;
    END WHILE;
            Return the best solution found;
  END;
```

Optimization algorithms: Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization method based on the collective behavior of a swarm of particles, each of which represents a possible solution. The particles move through the search space, guided by their own experience and the experience of their neighbors. The basic idea is to update the velocity and position of each particle, taking into account its best solution and the best solution in the swarm. Pseudocode for its implementation:

Pseudocode 3 – Particle Swarm Optimization

```
      Input: f(x), n, MaxGeneration, pBest, gBest, vi
      Output: x_best, f(x_best)
  BEGIN
    Objective function f(x), x = (x1, x2, ..., xd)^T
     Initialize a population of n particles xi (i = 1, 2, ..., n);
    Initialize velocities vi randomly for each particle;
   Define personal best (pBest) and global best (gBest) for each particle;
  WHILE (t < MaxGeneration) or (stop criterion not met) DO
      FOR i = 1 to n DO
           Update velocity vi based on pBest and gBest;
           Update particle position xi based on vi;
           Evaluate the fitness of each particle;
           Update personal best (pBest) for each particle;
           Update global best (gBest) for the swarm;
      END FOR;
  END WHILE;
            Return the best solution found;
   END;
```

Optimization algorithms: Support Vector Machine

The Support Vector Machine (SVM) method is used to predict continuous values, minimizing the prediction error and providing a certain tolerance (parameter ε). To work with nonlinear data, SVM uses the so-called "nuclear trick", which allows you to transform data into a higher feature space. Kernels can be linear, polynomial, or radial basis functions (RBF). SVM regression is characterized by resistance to overfitting, efficiency in tasks with high data dimensionality, and the ability to adjust through hyperparameters such as the regularization coefficient (C) and tolerance (ε). However, the algorithm requires significant computational resources on large datasets and is sensitive to the choice of hyperparameters. Pseudocode for its implementation:

Pseudocode 4 – Support Vector Machine

```
      Input: f(x), n, MaxGeneration, xi, xj, yi, yj
      Output: x_best, f(x_best)
  BEGIN
```

```
Choose kernel function K(xi, xj) (e.g., linear, polynomial, radial basis function);
Set regularization parameter C and tolerance ε;
αi = 0 for all data points i (Lagrange multipliers);
Define threshold parameter b = 0;
WHILE stop criterion not met DO
      FOR i = 1 to n DO
Compute the decision function:
f(x) = Σ αj * yj * K(xj, x) + b
Calculate the error Ei = f(xi) - yi.
              Update αi using the optimization rule:
              Maximize:
              W(α) = Σ αi - 0.5 * ΣΣ αi * αj * yi * yj * K(xi, xj)
              Subject to:
              0 ≤ αi ≤ C
              Σ αi * yi = 0
              Update the bias term b using support vectors.
      END FOR;
END WHILE;
              Return the best solution found;
 END;
```

Mathematical model of the control object

Oil separation involves the separation of oil, gas and water, which requires fine tuning to maintain optimal conditions inside the separator. This is especially important because the flow conditions, temperature and pressure can change, affecting the quality and quantity of the products received. The system parameters and variables used below are listed in Table 1.

Table 1 – System variables and parameters

Symbol	Description	Unit measurements
$h(t)$	Water level inside the separator	m
$Q_{in}(t)$	Water flow rate	m^3/h
$Q_{out}(t)$	Water outflow rate	m^3/h
r	Cage cross section radius	m
L	Water	m
$h_0(t)$	Oil level inside the separator	m
$P_g(t)$	Gas pressure inside the separator	Pa
C_v	Exhaust valve release ratio	-
u(t)	Valve open percentage	-
ρw	Density of water at operating temperature	kg/m^2
ρo	Oil density at operating temperature	kg/m^2
Pw	Downstream valve pressure	Pa
Umax	Maximum opening are of the control valve	m^2
ΔPout	Differential pressure over control valve	Pa

According to the geometry of the separator, the volume of water inside the separator is a function of the water level h and has a specific relationship as:

$$V(h) = \left(r^2\left(\frac{r-h}{r}\right) - (r-h)\sqrt{2rh-h^2}L\right) \tag{1}$$

Since normal operation requires a water level between the high alarm level (LAH) and the low alarm level (LAL), thus relation (1) can be simplified as a linear relationship over this interval, i.e. V(h) = ALh(t) where A ≈ πr2. The dynamics of the volume of water inside the separator corresponds to the principle of mass balance, i.e.:

$$\frac{dV(t)}{dt} \approx AL\frac{dh(t)}{dt} = Q_{in}(t) - Q_{out}(t). \tag{2}$$

According to the theory of flow dynamics, the water flow through the LCV 340018 valve can be determined as:

$$Q_{out} = C_v f(u)\sqrt{\frac{\Delta P_{out}}{\rho w}}, \tag{3}$$

where f(u) represents the characteristics of the open zone valve related to the open percentage u. For this particular LCV-340018 linear valve, the linear relationship is well maintained. Thus, there exists f(u) = uUmax. The pressure drop across the valve, denoted as Pout, can be estimated as:

$$\Delta P_{out}(t) = P_g(t) + \rho_0 g h_0(t) + \rho_w g h(t) - P_w(t). \tag{4}$$

The Cv of the valve in (3) is estimated using the least squares method based on the recorded data of water flow, water and oil levels inside the separator, gas pressure inside the separator, and water outlet pressure. Assuming that the density of water is constant, the value of Cv will be the solution:

$$min_{cv}\sum_{i}^{N}\left|Q_{out}(i) - C_v u(i)U_{max}\sqrt{\frac{\Delta P_{out}(i)}{\rho w}}\right|^2 \tag{5}$$

In general, the prediction error is limited to 10%. The validation of this model is within acceptable limits. If the gas pressure, downstream water valve pressure and oil level inside the separator are constant or their deviations from the average values are ignored, the non-linear system model is linearized under normal operating conditions. Assuming that the gas pressure, downstream water valve pressure and oil level inside the separator are constant or their deviations from the average values are ignored, the non-linear system model is linearized under normal operating conditions. By inserting specific system parameters, the linearized model results in the form:

$$47.55\frac{d\Delta h(t)}{dt} = Q_{in}(t) - 1.81\Delta h(t) - 10.82\Delta u(t), \tag{6}$$

where Δh(t) (Δu(t)) represents the water level deviation (valve position) to equilibrium. Thus, the transfer function representing the ratios from the unknown perturbation Qin(t) and the control input u (t) to the output h (t), respectively, will look like this:

$$G(s) = \frac{243.5s + 4382}{47.55s^2 + 245.3s + 4382}. \tag{7}$$

The transfer function is one of the ways to mathematically describe a dynamic system. Mainly used in control theory, communications and digital signal processing. Represents a differential operator that expresses the relationship between the input and output of a linear stationary system. Knowing the input signal of the system and the transfer function, it is possible to recover the output signal. Figure 2 shows the block diagram of the mathematical model in Simulink.

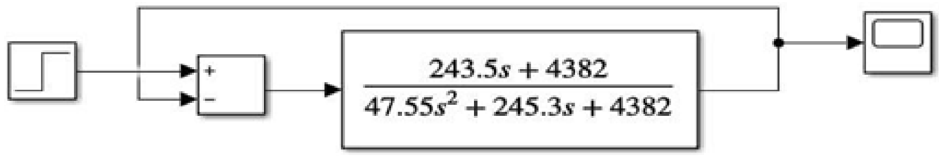

Figure 2 – Block diagram in MATLAB

The simulation results are presented in Figure 2.1.

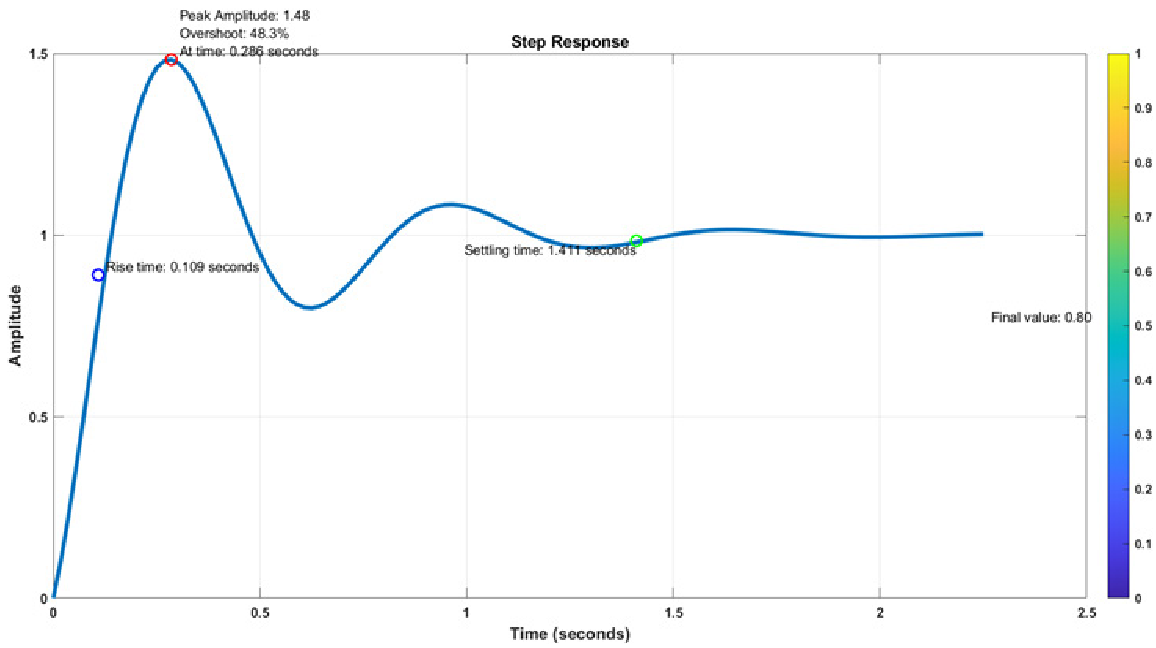

Figure 2.1 – Result of block diagram modeling

As could be seen from the figure, the system is stable but does not reach the desired value and there is an overshoot. To improve the dynamics of the system a typical controller should be synthesized.

Results and Discussion

This chapter presents the results of using rare machine learning algorithms to optimize the parameters of a PID controller using data from the oil separation process, which presented in Table 2. To evaluate the effectiveness of each method, numerical experiments were performed, the results of which are presented in graphical and tabular form.

Table 2 – Fragment of the dataset variables

Point Value	Setpoint	Output
300,1995611	300	33,15965301
300,1159456	300	33,12499092
300,1966126	300	33,13821924
300,212981	300	33,15679941
299,9569751	300	33,19906364
…	…	…

Figure 3 below shows the full data distribution.

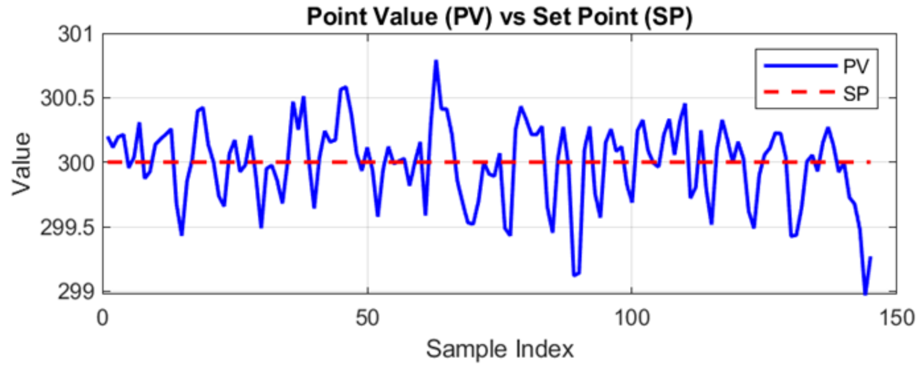

Figure 3 – Point Value distribution of FIC

Figure 4 below shows the Output distribution graphically.

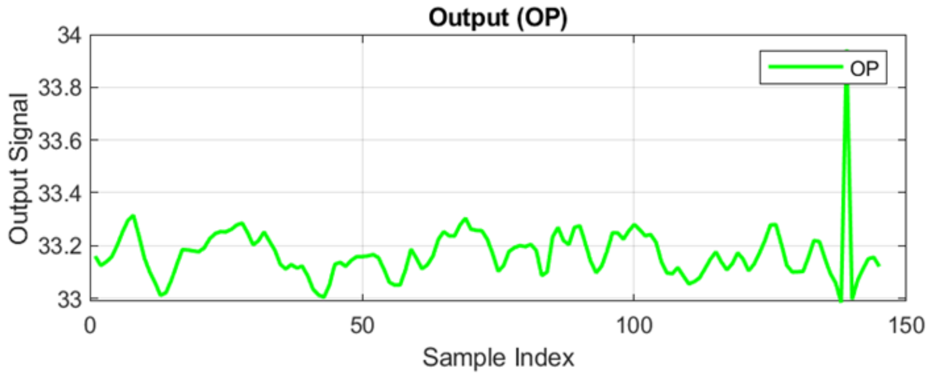

Figure 4 – Output distribution of FIC

The results were obtained based on the following algorithms: Cuckoo Search Algorithm (CSA), Firefly Algorithm (FA), Particle Swarm Optimization (PSO). Each of the algorithms was used to adjust the parameters of the PID controller, minimizing the time-weighted absolute error integral (ITAE). The effectiveness of the algorithms was evaluated by the following metrics: Standard error (MSE), Setting Time, Overshoot, Steady State Error. In Figure 5 the results obtained before implementation CSA are presented.

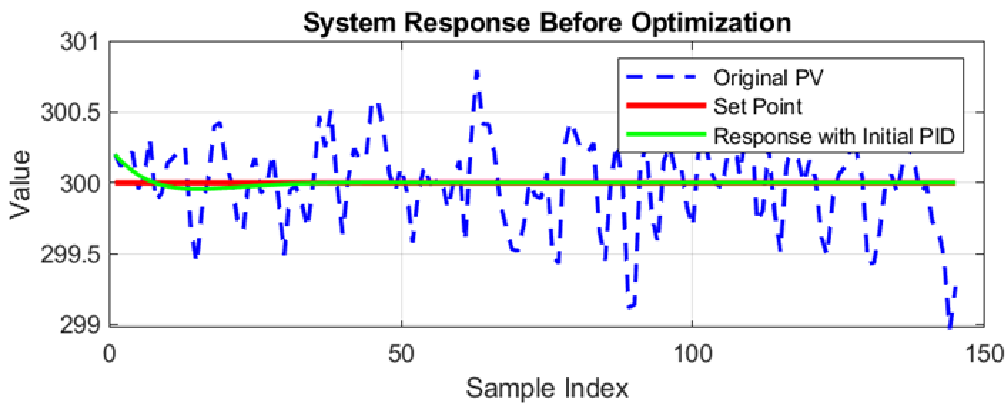

Figure 5 – System response before optimization using CSA

Figure 6 demonstrates the response of the system after implementation of Cuckoo Search Algorithm.

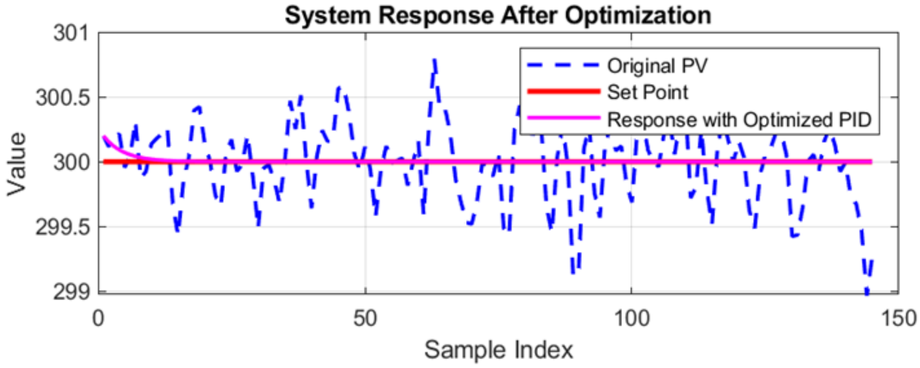

Figure 6 – System response after optimization using CSA

Prior to optimization, the system shows significant deviations from the set value of SP, which indicates a high instability of regulation. The amplitude of the oscillations is large, and the system is not able to effectively follow the target value. After optimizing the parameters of the PID controller, a significant decrease in fluctuations is observed, and the system stabilizes, demonstrating more accurate adherence to the set value of SP. This graphs clearly shows how optimizing the parameters of the PID controller improves the response of the system, reducing deviations and increasing its stability.

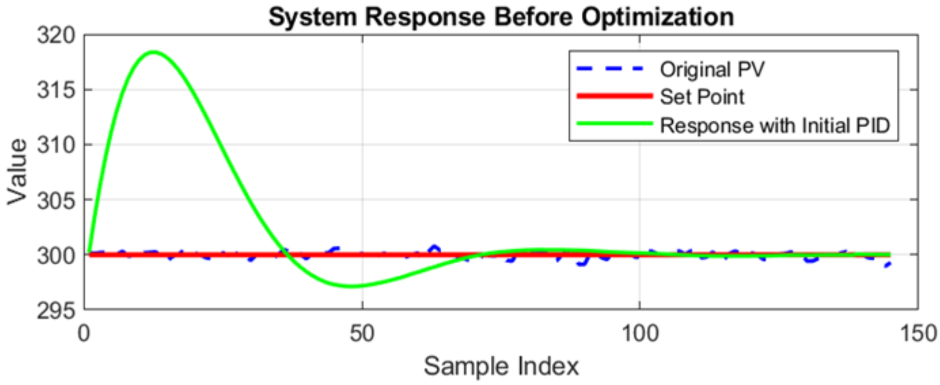

Figure 7 – System response before implementation Firefly algorithm

Figure 8 below shows how system responses after optimization.

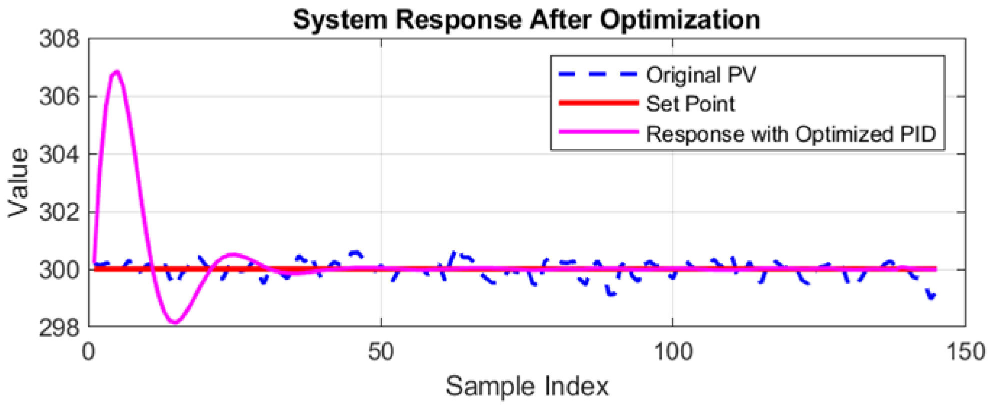

Figure 8 – System response after implementation Firefly algorithm

The graph shows the system responses before and after optimizing the parameters of the PID controller. Prior to optimization (upper graph), the system demonstrates significant overshoot, where the PV process variable exceeds the target value of SP to 320, and then slowly returns to the set level. After optimization (lower graph), the system response improves significantly: overshoot is significantly reduced, the PV process variable quickly reaches the target value and remains stable. The installation time is shortened, and the control becomes more accurate and stable.

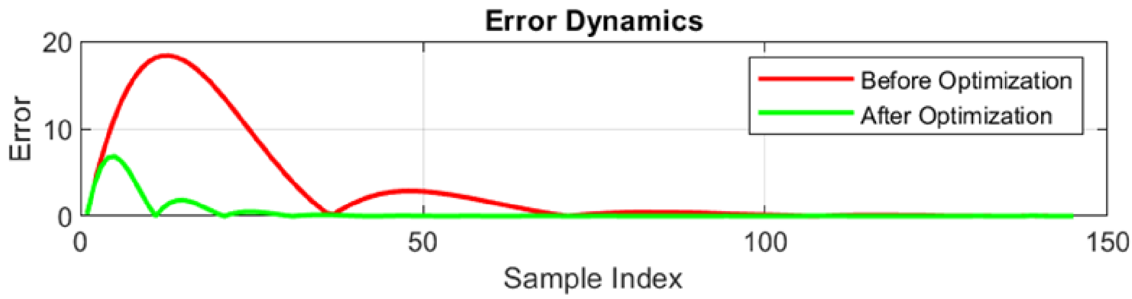

Figure 9 – Error dynamics comparison before
and after implementation of PSO algorithm

System output before and after comparison results performed in Figure 10.

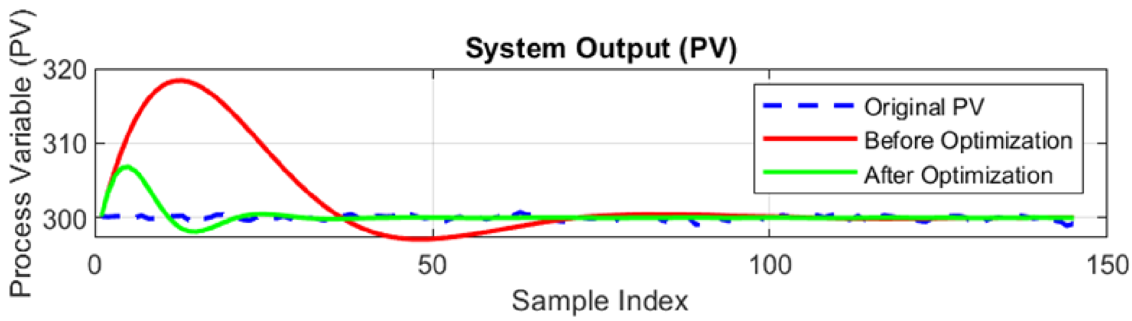

Figure 10 – Comparison of system output before
and after implementation of PSO algorithm

This graph shows two aspects of the system operation: error dynamics and process variable before and after optimization of the PID controller. The upper graph shows the dynamics of the error. Before optimization (red line), the error is characterized by significant overshoot and long-term fluctuations that fade only towards the end of the period. After optimization (green line), the error is significantly reduced, the system quickly achieves stability with minimal fluctuations. The lower graph shows the output signal of the PV system. Before optimization (red line), the system response shows a large overshoot and a long establishment time, while after optimization (green line), the response becomes accurate, stable and quickly reaches the set SP value. Optimization of the parameters of the PID controller has significantly improved the dynamic characteristics of the system, ensuring its more accurate and stable behavior.

The graph shows a comparison of the actual values of the PV process variable, the predicted values using the SVM model and the target value of SP. The graph shows that the SVM model successfully smooths out fluctuations and brings the process variable closer to the set value, which indicates its potential for use in forecasting and control optimization tasks.

To evaluate the effectiveness of the optimization algorithms used in this study, four key metrics were analyzed: mean square error (MSE), Settling Time, Overshoot, and Steady-State Error. These metrics provide a comprehensive assessment of the performance of each algorithm in the task of managing the system and achieving optimal adherence to a given value.

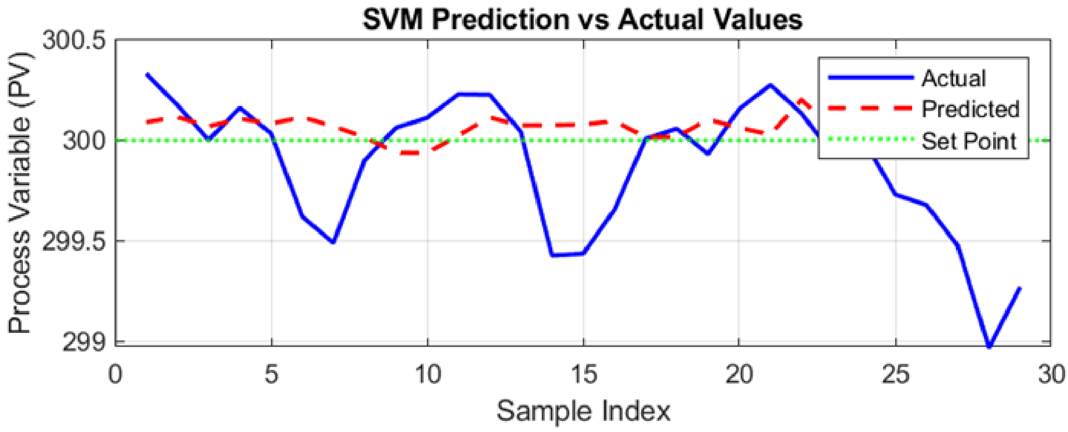

Figure 11 – Comparison of actual and predicted PV using SVM

The standard error (MSE) measures the mean square deviation of a controlled variable (PV) from a set value (SP), providing information about the accuracy of the control. The Setting Time reflects the time it takes for the system to stabilize within a given range around the SP. Overshoot indicates the maximum amount of excess of PV over SP during the transition process, and the Steady-State Error characterizes the difference between PV and SP after stabilization of the system.

The results of calculations based on the specified metrics for each algorithm are presented in the table below:

Table 3 – Performance comparison of optimization algorithms

Algorithm	MSE	Settling Time	Overshoot	Steady State Error
Cuckoo Search	0.00072739	0	0.066533	0
Firefly Algorithm	1.687	6	2.2848	0.0109
Particle Swarm Optimization	1.687	6	2.2848	0.0109
SVM	0.0024785	0	0.040733	0.0471

A comparison of algorithms based on the presented metrics shows that the best result is achieved when using the Cuckoo Search algorithm, which demonstrates a minimum standard error (MSE = 0.0007239), the absence of overshoot and establishment time, as well as zero steady-state error. The Firefly and Particle Swarm Optimization algorithms show identical results with an MSE of 1.687, a setup time of 6 steps and an overshoot of 2.2848%. The steady state error of these methods is 0.0109, which is slightly worse compared to Cuckoo Search. The SVM algorithm demonstrates an average result, showing MSE = 0.0024785, no establishment time and minimal overshoot (0.040733%), however, its steady-state error is greater than all others (0.0471), which may indicate problems with long-term control stability. Thus, the Cuckoo Search algorithm is the most effective among the considered methods for all key metrics.

Conclusion

During the study, various approaches to optimizing the parameters of the PID controller for process control were studied. The algorithms Cuckoo Search, Firefly Algorithm, Particle Swarm Optimization and the regression method based on support vectors (SVM) were considered and compared. The main focus was on metrics such as the mean square error (MSE), Settling Time, Overshoot, and Steady State Error.

The analysis of the results showed that each of the approaches has its own strengths and weaknesses, depending on the characteristics of the system and management requirements. The Cuckoo Search algorithm demonstrated the best results in terms of the MSE metric and minimal deviations from the set value, which confirms its effectiveness for tasks with fast transients and high control accuracy. The Firefly and PSO algorithms showed similar results, with slight differences in overshoot and set time, which makes them suitable for systems with less stringent accuracy requirements. The SVM method has also shown good results, especially in minimizing errors in the steady state, but its effectiveness depends on the quality of the training data.

Thus, for the optimal choice of the optimization method, it is necessary to consider the specifics of the process, the requirements for the management system and the resources available for implementation. The presented results emphasize the importance of an integrated approach to the analysis and comparison of various methods to achieve the best management characteristics.

REFERENCES

1 Sathasivam L., Elamvazuthi I., Ahamed Khan M. K. A., & Parasuraman S. Tuning A Three-Phase Separator Level Controller via Particle Swarm Optimization Algorithm // Proceedings of the 2018 International Conference on Recent Trends in Electrical, Control and Communication (RTECC). – IEEE, 2018. – P. 265–268.

2 Reis V.C., Santos M.F., Carmo M.J., Ferreira F.C. Control of Level in Systems Based on Arduino Platform through PC // Proceedings of the 20th International Conference on System Theory, Control and Computing (ICSTCC). – Sinaia, Romania: IEEE, 2016. – P. 251–256.

3 Charkoutsis S., Kara-Mohamed M. Particle Swarm Optimization tuned nonlinear PID controller with improved performance and robustness for First Order Plus Time Delay systems // Results in Control and Optimization. – 2023. – Vol. 12. – P. 100289.

4 Foss B. Process control in conventional oil and gas production: Challenges and opportunities // Control Engineering Practice. – 2012. – Vol. 20. – No. 10. – P. 1058–1064.

5 Patel V. V. Ziegler-nichols tuning method: Understanding the pid controller // Resonance. – 2020. – Vol. 25. – No. 10. – P. 1385–1397.

6 Ribeiro J.M.S., Santos M.F., Carmo M.J., Silva M.F. Comparison of PID Controller Tuning Methods: Analytical/Classical Techniques versus Optimization Algorithms // Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (ICSMC). – 2017. – P. 533–538.

7 Garpinger O., Hägglund T., Åström K. J. Performance and robustness trade-offs in PID control // Journal of Process Control. – 2014. – Vol. 24. – No. 5. – P. 748–760.

8 Shahrokhi M., Zamorrodi A. Comparison of PID controller adjustment methods // Journal of Process Control. – 2024. – No. 2. – P. 1–15.

9 Eltayeb A., Ahmed G., Imran I.H., Alyazidi N.M., Abubaker A. Comparative Analysis: Fractional PID vs. PID Controllers for Robotic Arm Using Genetic Algorithm Optimization // Automation. – 2024. – Vol. 5. – P. 230–245.

10 Hekimoğlu B. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm // IEEE Access. – 2019. – Vol. 7. – P. 38100–38114.

11 Mehta S.A., Katrodiya J., Mankad B. Simulation, Design and Practical Implementation of IMC Tuned Digital PID Controller for Liquid Level Control System // Proceedings of the International Conference on Current Trends in Technology (NUiCONE-2011). Ahmedabad, India: IEEE, 2011. – P. 1–5.

12 Alsmadi M.K., Mohammad R.M.A., Alzaqebah M., Jawarneh S., AlShaikh M., Al Smadi A., Alghamdi F.A., Alqurni J. S., & Alfagham H. Intrusion Detection Using an Improved Cuckoo Search Optimization Algorithm // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. – 2024. – Vol. 15. – No. 2. – P. 73–93. https://doi.org/10.58346/JOWUA.2024.I2.006.

13 Ali E.S., Abd Elazim S.M., Abdelaziz A.Y. Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations // Renewable Energy. – 2017. – Vol. 101. – P. 1311–1324.

14 Debbarma S., Saikia L.C., Sinha N. Solution to automatic generation control problem using firefly algorithm optimized IλDµ controller // ISA Transactions. – 2014. – Vol. 53. – Issue 2. – P. 358–366.

15 Guha D., Roy P.K., Banerjee S. Optimal tuning of 3 degree-of-freedom proportional-integral-derivative controller for hybrid distributed power system using dragonfly algorithm // Computers and Electrical Engineering. – 2018. – Vol. 72. –P. 137–153.

16 Mohamed M.J., Oleiwi B.K., Azar A.T., Hameed I.A. Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator // Heliyon. – 2024. – Vol. 10. Article e32661.

17 Saravanan G., Suresh K.P., Pazhanimuthu C., Kumar R.S. Artificial rabbits optimization algorithm based tuning of PID controller parameters for improving voltage profile in AVR system using IoT // e-Prime – Advances in Electrical Engineering, Electronics and Energy. – 2024. – Vol. 8. Article 100523.

18 Mosaad A. M., Attia M. A., Abdelaziz A. Y. Whale optimization algorithm to tune PID and PIDA controllers on AVR system // Ain Shams Engineering Journal. – 2019. – Vol. 10. – P. 755–767.

19 Mohamed M.J., Oleiwi B.K., Azar A.T., Hameed I.A. Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator // Heliyon. – 2024. – Vol. 10. Article e32661.

20 Güven A.F., Mengi O.Ö., Elseify M.A., & Kamel, S. Comprehensive Optimization of PID Controller Parameters for DC Motor Speed Management Using a Modified Jellyfish Search Algorithm // Optimal Control Applications and Methods. – 2024. – Vol. 0. – P. 1–23.

21 Güven A.F., & Mengi O.Ö. Nature-inspired algorithms for optimizing fractional order PID controllers in time-delayed systems // Optimal Control Applications and Methods. – 2024. – Vol. 45. – No. 3. – P. 1251–1279.

22 Vanchinathan K., Selvaganesan N. Adaptive fractional order PID controller tuning for brushless DC motor using Artificial Bee Colony algorithm // Results in Control and Optimization. – 2021. – Vol. 4. – P. 100032.

23 Sharif Naser Makhadmeh, Mohammed Azmi Al-Betar, Iyad Abu Doush, Mohammed A. Awadallah, Sofian Kassaymeh, Seyedali Mirjalili, Raed Abu Zitar. Recent advances in Grey Wolf Optimizer, its versions and applications: Review // IEEE Access. – 2023.

24 Hui T., Zeng W., Yu T. Core power control of the ADS based on genetic algorithm tuning PID controller // Nuclear Engineering and Design. – 2020. – Vol. 370. – P. 110835.

25 Izci D., Ekinci S., Hedley J., Demirören A. HHO Algorithm Based PID Controller Design for Aircraft Pitch Angle Control System // Proceedings of the 2020 International Conference on Advances in Computing, Communication, and Automation (ICACCA). Piscataway, NJ: IEEE. – 2020. – P. 9152897.

26 Serdar Ekinci, Davut Izci, Baran Hekimoğlu. Henry Gas Solubility Optimization Algorithm Based FOPID Controller Design for Automatic Voltage Regulator // Proceedings of the 2nd International Conference on Electrical, Communication and Computer Engineering (ICECCE). Istanbul, Turkey: IEEE, 2020. – P. 1–6.

27 Patil S.R., Agashe S. RelaxC controller and PID for time delay systems: Experimental test results on Boiler pilot plant // Results in Control and Optimization. – 2024. – Vol. 9. Article 100297.

28 Murugesan D., Jagatheesan K., Shah P., Sekhar R. Fractional order PIλDµ controller for microgrid power system using cohort intelligence optimization // Results in Control and Optimization. – 2023. – Vol. 11. Article 100218.

29 Ataşlar-Ayyıldız B. Robust Trajectory Tracking Control for Serial Robotic Manipulators Using Fractional Order-Based PTID Controller // Fractal Fract. – 2023. – Vol. 7. – Issue 3. – P. 250.

REFERENCES

1 Sathasivam L., Elamvazuthi I., Ahamed Khan M. K. A., & Parasuraman S. Tuning A Three-Phase Separator Level Controller via Particle Swarm Optimization Algorithm. Proceedings of the 2018 International Conference on Recent Trends in Electrical, Control and Communication (RTECC). IEEE, 2018, pp. 265–268.

2 Reis V.C., Santos M.F., Carmo M.J., Ferreira F.C. Control of Level in Systems Based on Arduino Platform through PC. Proceedings of the 20th International Conference on System Theory, Control and Computing (ICSTCC). Sinaia, Romania: IEEE, 2016, pp. 251–256.

3 Charkoutsis S., Kara-Mohamed M. Particle Swarm Optimization tuned nonlinear PID controller with improved performance and robustness for First Order Plus Time Delay systems. Results in Control and Optimization, 12, 100289 (2023).

4 Foss B. Process control in conventional oil and gas production: Challenges and opportunities. Control Engineering Practice, 20 (10), 1058–1064 (2012).

5 Patel V.V. Ziegler-nichols tuning method: Understanding the pid controller. Resonance, 25, 1385–1397, 10 (2020).

6 Ribeiro J.M.S., Santos M.F., Carmo M.J., Silva M.F. Comparison of PID Controller Tuning Methods: Analytical/Classical Techniques versus Optimization Algorithms. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (ICSMC), 2017, p. 533–538.

7 Garpinger O., Hägglund T., Åström K. J. Performance and robustness trade-offs in PID control. Journal of Process Control., 24 (5), 748–760 (2014).

8 Shahrokhi M., Zamorrodi A. Comparison of PID controller adjustment methods. Journal of Process Control., 2, 1–15 (2024).

9 Eltayeb A., Ahmed G., Imran I.H., Alyazidi N.M., Abubaker A. Comparative Analysis: Fractional PID vs. PID Controllers for Robotic Arm Using Genetic Algorithm Optimization. Automation, 5, 230–245 (2024).

10 Hekimoğlu B. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. IEEE Access., 7, 38100–38114 (2019).

11 Mehta S.A., Katrodiya J., Mankad B. Simulation, Design and Practical Implementation of IMC Tuned Digital PID Controller for Liquid Level Control System. Proceedings of the International Conference on Current Trends in Technology (NUiCONE-2011). Ahmedabad, India: IEEE, 2011, pp. 1–5.

12 Alsmadi M.K., Mohammad R.M.A., Alzaqebah M., Jawarneh S., AlShaikh M., Al Smadi A., Alghamdi F.A., Alqurni J.S., & Alfagham H. Intrusion Detection Using an Improved Cuckoo Search Optimization Algorithm. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 15 (2), 73–93 (2024). https://doi.org/10.58346/JOWUA.2024.I2.006.

13 Ali E.S., Abd Elazim S.M., Abdelaziz A.Y. Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations. Renewable Energy, 101, 1311–1324 (2017).

14 Debbarma S., Saikia L.C., Sinha N. Solution to automatic generation control problem using firefly algorithm optimized IλDμ controller. ISA Transactions, 53 (2), 358–366 (2014).

15 Guha D., Roy P.K., Banerjee S. Optimal tuning of 3 degree-of-freedom proportional-integral-derivative controller for hybrid distributed power system using dragonfly algorithm. Computers and Electrical Engineering, 72, 137–153 (2018).

16 Mohamed M.J., Oleiwi B.K., Azar A.T., Hameed I.A. Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator. Heliyon, 10, e32661 (2024).

17 Saravanan G., Suresh K.P., Pazhanimuthu C., Kumar R.S. Artificial rabbits optimization algorithm based tuning of PID controller parameters for improving voltage profile in AVR system using IoT. e-Prime – Advances in Electrical Engineering, Electronics and Energy, 8, 100523 (2024).

18 Mosaad A.M., Attia M.A., Abdelaziz A.Y. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. Ain Shams Engineering Journal, 10, 755–767 (2019).

19 Mohamed M.J., Oleiwi B.K., Azar A.T., Hameed I.A. Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator. Heliyon, 10, e32661 (2024).

20 Güven A.F., Mengi O.Ö., Elseify M.A., & Kamel S. Comprehensive Optimization of PID Controller Parameters for DC Motor Speed Management Using a Modified Jellyfish Search Algorithm. Optimal Control Applications and Methods, 1–23 (2024).

21 Güven A.F., & Mengi O.Ö. Nature-inspired algorithms for optimizing fractional order PID controllers in time-delayed systems. Optimal Control Applications and Methods, 45 (3), 1251–1279 (2024).

22 Vanchinathan K., Selvaganesan N. Adaptive fractional order PID controller tuning for brushless DC motor using Artificial Bee Colony algorithm. Results in Control and Optimization, 4, 100032 (2021).

23 Sharif Naser Makhadmeh, Mohammed Azmi Al-Betar, Iyad Abu Doush, Mohammed A. Awadallah, Sofian Kassaymeh, Seyedali Mirjalili, Raed Abu Zitar. Recent advances in Grey Wolf Optimizer, its versions and applications: Review. IEEE Access (2023).

24 Hui T., Zeng W., Yu T. Core power control of the ADS based on genetic algorithm tuning PID controller. Nuclear Engineering and Design, 370, 110835 (2020).

25 Izci D., Ekinci S., Hedley J., Demirören A. HHO Algorithm Based PID Controller Design for Aircraft Pitch Angle Control System. Proceedings of the 2020 International Conference on Advances in Computing, Communication, and Automation (ICACCA). Piscataway, NJ: IEEE, 2020, p. 9152897.

26 Serdar Ekinci, Davut Izci, Baran Hekimoğlu. Henry Gas Solubility Optimization Algorithm Based FOPID Controller Design for Automatic Voltage Regulator. Proceedings of the 2nd International Conference on Electrical, Communication and Computer Engineering (ICECCE). Istanbul, Turkey: IEEE, 2020, pp. 1–6.

27 Patil S. R., Agashe S. RelaxC controller and PID for time delay systems: Experimental test results on Boiler pilot plant. Results in Control and Optimization, 9, 100297 (2024).

28 Murugesan D., Jagatheesan K., Shah P., Sekhar R. Fractional order PIλDμ controller for microgrid power system using cohort intelligence optimization. Results in Control and Optimization, 11, 100218 (2023).

29 Ataşlar-Ayyıldız B. Robust Trajectory Tracking Control for Serial Robotic Manipulators Using Fractional Order-Based PTID Controller. Fractal Fract., 7 (3), 250 (2023).

[1]Самигулина З.,
PhD, доцент, ORCID ID: 0000-0002-5862-6415,
e-mail: z.samigulina@kbtu.kz
[1]*Аманғалиева А.,
бакалавр, ORCID ID: 0009-0003-7991-9836,
*e-mail: a_amangaliyeva@kbtu.kz

[1]Қазақстан-Британ техникалық университеті, Алматы қ., Қазақстан

МҰНАЙДЫ СЕПАРАЦИЯЛАУ ПРОЦЕСІ ДЕРЕКТЕРІНІҢ НЕГІЗІНДЕ МАШИНАЛЫҚ ОҚЫТУ АЛГОРИТМДЕРІН ҚОЛДАНЫП ПИД РЕТТЕГІШІНІҢ ПАРАМЕТРЛЕРІН ОҢТАЙЛАНДЫРУ

Аңдатпа

Бұл жұмыс мұнайды бөлу процесін басқару жүйесі үшін машиналық оқыту алгоритмдерін қолдану арқылы PID реттегішінің параметрлерін оңтайландыру процесін зерттейді. Контроллер параметрлерін оңтайландыру (Kp, Ki, Kd) бақылау сапасын жақсарту және динамикалық процестердегі қателерді азайту үшін маңызды. Бұл мәселені шешу үшін Көкек іздеу алгоритмі (CSA), Жарқырауық қоңыздар алгоритмі (FA), бөлшектер тобын оңтайландыру (PSO) және тірек векторлық машина (SVM) сияқты бірнеше инновациялық әдістер қарастырылды. Барлық деректер, соның ішінде технологиялық нүктенің мәндері (PV), белгіленген мәндер (SP) және шығыс сигналдары (OP) Теңізшевройлдан алынды. Сонымен қатар, оңтайландырылған реттегіштердің тиімділігін бағалау үшін түбірлік орташа квадраттық қате (MSE), орнығу уақыты, асып кету және тұрақты күйдегі қате сияқты көрсеткіштер қолданылды. Жалпы алғанда, зерттеу нәтижелері дәстүрлі тәсілдермен салыстырғанда машиналық оқыту алгоритмдерін қолдану арқылы жүйенің динамикалық өнімділігінің айтарлықтай жақсарғанын көрсетеді. Алынған оңтайландыру параметрлері жылдам әрі тұрақты бола отырып, мақсатты мәнге жетті, бұл технологиялық процесті басқару өнімділігін арттыруға мүмкіндік берді.

Тірек сөздер: мұнайды сепарациялау, автоматтандыру жүйесін оңтайландыру, ПИД реттегіш, параметрлерді оңтайландыру, машиналық оқыту, көкектерді іздеу алгоритмі, жарқырауық қоңыздар алгоритмі, бөлшектер тобын оңтайландыру, қолдау вектор машинасы.

[1]**Самигулина З.,**
PhD, доцент, ORCID ID: 0000-0002-5862-6415,
e-mail: z.samigulina@kbtu.kz
[1]***Аманғалиева А.,**
бакалавр, ORCID ID: 0009-0003-7991-9836,
*e-mail: a_amangaliyeva@kbtu.kz

[1]Казахстанско-Британский технический университет, г. Алматы, Казахстан

ОПТИМИЗАЦИЯ ПАРАМЕТРОВ ПИД-РЕГУЛЯТОРА С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ НА ОСНОВЕ ДАННЫХ ПРОЦЕССА СЕПАРАЦИИ НЕФТИ

Аннотация

В работе исследуется процесс оптимизации параметров ПИД-регулятора с помощью использования алгоритмов машинного обучения для системы управления процессом сепарации нефти. Оптимизация параметров контроллера (Kp, Ki, Kd) важна для повышения качества управления и уменьшения количества ошибок в динамических процессах. Для решения этой проблемы было рассмотрено несколько инновационных методов, таких как алгоритм поиска кукушки (CSA), алгоритм светлячков (FA), оптимизация роя частиц (PSO) и метод опорных векторов (SVM). Все данные, включая текущие значения процессов (PV), уставки (SP) и выходные сигналы (OP), были получены от «Тенгизшевройл»а. Кроме того, для оценки эффективности оптимизированных регуляторов использовались такие показатели, как среднеквадратичная ошибка (MSE), время настройки, превышение и установившаяся ошибка. В целом результаты исследования свидетельствуют о значительном улучшении динамических характеристик системы за счет использования алгоритмов машинного обучения по сравнению с традиционными подходами. Полученные параметры оптимизации достигли целевого значения, оставаясь при этом более быстрыми и стабильными, что позволило повысить производительность управления технологическим процессом.

Ключевые слова: сепарация нефти, оптимизация системы автоматизации, ПИД-регулятор, оптимизация параметров, машинное обучение, алгоритм поиска кукушки, алгоритм светлячков, оптимизация роя частиц, метод опорных векторов.

Article submission date: 25.12.2024