

Atf İçin: Çimen ME, Garip Z, Emekli M, Boz AF, 2022. Isı Eşanjörünün Aşım Kısıtı Altındaki Kontrolü için Genetik Algoritma Kullanarak Bulanık Pıd Denetleyisinin Tasarımı. Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 12(1): 164-181.

To Cite: Çimen ME, Garip Z, Emekli M, Boz AF, 2022 Fuzzy Logic PID Design using Genetic Algorithm under Overshoot Constrained Conditions for Heat Exchanger Control. Journal of the Institute of Science and Technology, 12(1): 164-181.

Fuzzy Logic PID Design using Genetic Algorithm under Overshoot Constrained Conditions for Heat Exchanger Control

Murat Erhan ÇİMEN^{*1}, Zeynep GARİP², Mehmet EMEKLİ¹, Ali Fuat BOZ¹

ABSTRACT: In this study, a controller design was carried out for the heat exchanger, which is widely used in the industry. Firstly, Zeigler Nichols step, Zeigler Nichols frequency, AMIGO step and AMIGO frequency methods were used for the PID controller in the control of this system. Then, using the mathematical model of the heat exchanger system, 2%, 5% and 10% overshoot constraints were added to the ISE performance criteria, and controller designs were realized with genetic algorithm. In addition, two different topologies were used for the fuzzy PID controller in the controller design. The results obtained were examined and it was seen that the design realized with fuzzy logic for this study could be improved more. However, topologies designed with fuzzy logic have obtained better results than classical PID controllers and the classical PID designed study in the literature.

Keywords: Heat exchanger, genetic algorithm, fuzzy pid, constrained optimization

¹ Murat Erhan ÇİMEN ([Orcid ID: 0000-0002-1793-485X](https://orcid.org/0000-0002-1793-485X)), Mehmet EMEKLİ ([Orcid ID: 0000-0001-9917-7057](https://orcid.org/0000-0001-9917-7057)), Ali Fuat BOZ ([Orcid ID: 0000-0001-6575-7678](https://orcid.org/0000-0001-6575-7678)), Sakarya University of Applied Sciences, Faculty of Technology, Department of Electrical and Electronic Engineering, Sakarya, Turkey

² Zeynep GARİP ([Orcid ID: 0000-0002-0420-8541](https://orcid.org/0000-0002-0420-8541)), Sakarya University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, Sakarya, Turkey

***Sorumlu Yazar/Corresponding Author:** Murat Erhan ÇİMEN, e-mail: muratcimen@subu.edu.tr

INTRODUCTION

Chemical processes have an important place in the industry as they are used a lot in areas such as medicine, milk, food, air conditioning-refrigerator-cooling and petroleum, which meet the basic needs of human beings. For this reason, it is also important to control the devices used in chemical processes. One of these devices of the chemical processes is heat exchangers. Heat exchangers are devices that transfer the temperature of the fluid given to the inlet to the outlet by decreasing or increasing it. Especially, heat exchangers used in chemical processes are preferred because they can be designed in many different sizes, they are high efficiency and low cost. In heat exchangers, the speed, temperature and pressure of the fluid are controlled in order to achieve the desired quality in the production of chemical products. In addition, it is desired to keep the heat value of the fluid taken from the output of the system constant in case of disturbance to the system or under changing conditions. Many different control techniques have been applied in the control of the heat exchanger, and the most common is PID controllers.

In particular, there is on-off control, which is widely used in the control of refrigerators, which is one of the thermal systems (Hamid et al., 2009; Mraz 2001). This method is based on operating the cooler with the help of a thermostat. If the temperature of the refrigerator is not at the desired value, the refrigerator is tried to be cooled. Otherwise, the system does not apply any check marks. Hamid et al. used a PID controller to control the temperature of the refrigerator (Hamid et al., 2009). Tridianto et al. carried out their study with a cascaded PID controller in the control of Shell and tube heat exchanger system (Tridianto et al., 2017). Subsequently they performed the Heat Exchanger control with PID in their study and examined its dynamic effects (Tridianto et al., 2017). Khare et al. carried out the design of PID and Internal Mode Control under the control of the heat exchanger and compared their results (Khare et al., 2010). Padhee et. al. used the autotuning technique for the recognition of the Heat Exchanger System and designed a controller with the Zeigler Nichols method. In addition to PID, they added feedforward and designed it in feedforward-PID. In addition to these, they also designed the Internal Model Controller and compared the results of all of them (Padhee et al., 2011). Isa et al. carried out a PID controller design using the M-Constrained Integral Gain Optimization (MIGO) method for the heat exchanger. They examined the effect of sampling time (Isa et al., 2014). In addition, Vasickaninová et al. performed the control of the heat exchanger system with PID as well as artificial neural networks predictive control (Vasičkaninová et al., 2011). Apart from these, Khames et al. carried out the control of the heat exchanger by using the sliding mode control method, which is one of the non-linear control methods (Khames et al., 2020). Soesanti et al., in their study, directly controlled the heat exchanger system with a takagi sugeno-based fuzzy-based control (Soesanti et al., 2019). Kishore et al., on the other hand, carried out the control of the system by using the PI fuzzy structure in the control of the heat exchanger. As a result, they compared fuzzy logic controller and PI based fuzzy logic control of this structure (Kishore et al., 2013). Kesavan et al. also performed a PID-based control under the control of the heat exchanger. They are designed with the Particle Swarm optimization algorithm of the PID controller. They compared their results with internal Model Controller (Kesavan et al., 2016). Reddy, on the other hand, performed the control of the heat exchanger with a PID-based fuzzy logic controller in their study. They used Genetic algorithm to determine the parameters of the controller (Reddy et al., 2020).

In this study, the control of the heat exchanger is designed classically and fuzzy logic based controllers. Classically designed PID controllers are designed according to Zeigler Nichols step, Zeigler Nichols frequency, AMIGO PID step and frequency methods. Then, using genetic algorithm, PID

controller design was carried out according to ISE performance criteria. After that, fuzzy logic was performed with PID controllers according to ISE performance criteria by using genetic algorithm. A similar study exists for fuzzy logic based PID. However, in the study of Reddy and his friend (Reddy et al., 2020), information such as fuzzy logic controller topology and fuzzy logic membership functions and rule table have not been expressed clearly. For this reason, the simulation study on fuzzy PID that Reddy et al. have done could not be applied in this study. For this reason, only the results with the classical PID could be compared. In addition, in this study, the topologies for the fuzzy logic-based controller are given clearly to contribute to the studies in the literature. In addition, Zeigler Nichols and AMIGO methods are compared for classical PID controller design in controlling the heat exchanger. In addition, the originality of this study; The overshoot is added to the objective criterion with a penalty function as a constraint at certain desired values, and a fuzzy-based PID controller is designed for different topologies using genetic algorithms. This is applied to the heat exchanger, which is a very common system in the industry.

In the 2nd Section of this study, the modeling of the heat exchanger is given in the materials and methods section. PID and Fuzzy PID controller, purpose criterion, Genetic Algorithm and Controller design are given as the basis. In 3rd Section, the results obtained in the control of the heat exchanger are given and the results are analyzed and discussed in detail. In the 4th Section, conclusion is given.

MATERIALS AND METHODS

Heat Exchanger Model

Heat exchangers are probably used in many places in industry such as food, cleaning, medicine, dairy, food, air conditioning-refrigerator-refrigeration and petroleum. In the chemical process industry, the shell and tube heat exchanger is usually encountered (Kesavan et al., 2016; Kishore et al., 2013; Soesanti et al., 2019). A shell and a number of tubes make up this sort of heat exchanger. Heat is transferred from hot to cold fluid using a heat exchanger system. The structure of the heat exchanger system is depicted in Figure 1.

The cold fluid reservoir is connected to the tube inlet by the tube inlet pump, and the hot fluid reservoir is connected to the shell inlet by the shell inlet pump. The cold fluid passes through the tube, while the hot fluid passes through the heat exchanger's shell. The flow rate of hot and cold fluid determines the rate of heat transfer between the shell and tube. The temperature of the tube outlet is controlled by adjusting the flow rate of the shell inlet. The controller sends a command to the shell inlet pump, which regulates the flow rate depending on the tube outlet temperature (Padhee et al., 2011; Tridianto et al., 2017). Generally it was rather difficult to obtain an accurate model with the present method (Reddy et al., 2020). Thus most of the chemical processes are modelled first order system or second order system due to complexity.

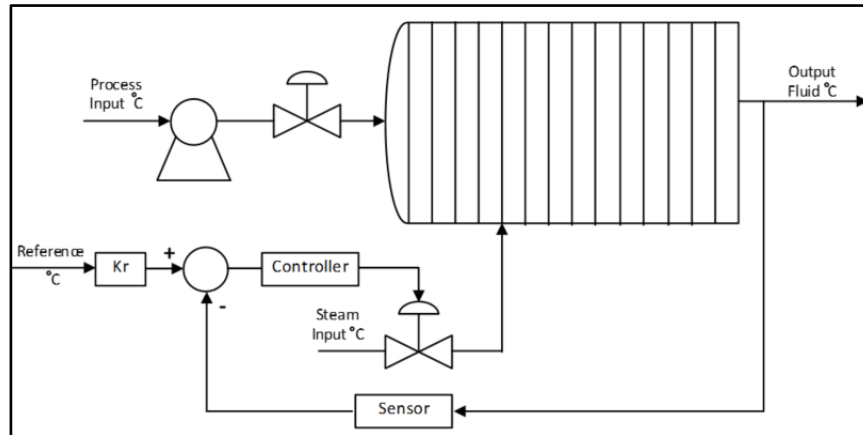


Figure 1. Heat Exchanger System Control Diagram

For simplicity, each structure is modeled as a first-order system. So Plant model is given as in Equation 1. Valve changes system dynamic and its model is given as in Equation 2. Sensor' model is given as in Equation 3 (Reddy et al., 2020). The control block structure of the heat exchanger to be controlled is given in Figure 2. Kr given in the system is an actuator coefficient and its value is taken as 0.16 (Reddy et al., 2020).

$$G_{plant} = \frac{50}{30s+1} \quad (1)$$

$$G_{valve} = 0.13 \times \frac{0.75}{3s+1} \quad (2)$$

$$G_{sensor} = \frac{0.16}{10s+1} \quad (3)$$

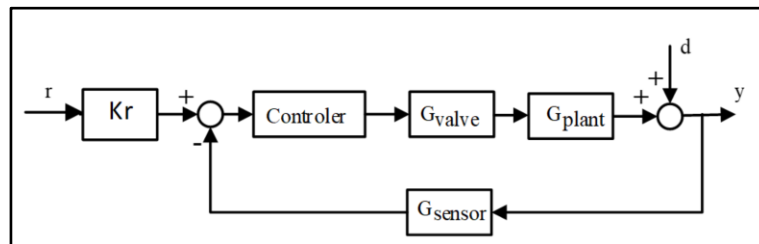


Figure 2. Control block structure of heat exchanger

PID controller

The most common type of controller encountered in controlling systems in industry is the PID controller type. It was proposed by studies by Zeigler and Nichols in the mid-19th century (Ziegler and Nichols, 1942; Ziegler and Nichols, 1943). In the proposed methods, the controller design was carried out from the reaction of the system to the step input. The other method they suggest is the frequency method. In this method, the closed loop gain of the system is increased, allowing the system to oscillate, and a controller design has been carried out according to the oscillation response of the system. These methods are still being developed and recently AMIGO step and AMIGO frequency methods have also been proposed (Åström and Hägglund, 2004; Hägglund and Åström, 2002). In addition to these technique, the PID controller can be designed according to an objective function with metaheuristic algorithms (Boz et al., 2017a; Boz et al., 2017a; Çimen et al., 2019; Çimen et al., 2017; Çimen et al., 2021a; Rajagopal et al., 2021; Yılmaz et al., 2021). Generally this controller has a three parameters that are Proportional constant (K_P), Integral constant (K_I), and Derivative constant (K_D).

$$G_{PID} = K_P + K_I \frac{1}{s} + K_D s \quad (4)$$

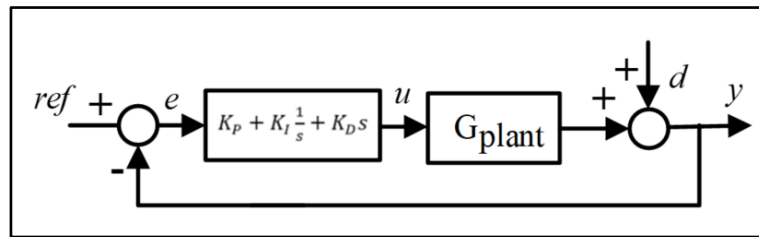


Figure 3. Control block diagram of heat exchanger

Fuzzy PID controller

Fuzzy logic was first introduced in 1964 by Lotfi A. Zadeh, an Azerbaijani scientist. Fuzzy logic inferences work with fuzzy sets (Zadeh, 1996). While classical sets use only bivalent membership values, fuzzy sets expand by using intermediate values as the concept of partial membership (Peçe, et. al., 2020). Fuzzy sets are widely used most effectively to quantify verbal expressions. This feature also makes fuzzy logic inferences more effective. The operation of a simple fuzzy inference mechanism is illustrated in Figure 4 (Cihan et al., 2008; Kaplan et al. 2020).

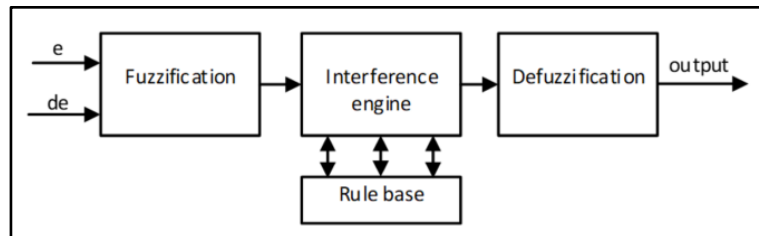


Figure 4. Fuzzy logic inference scheme

In the literature, there are Type 1 and Type 2 fuzzy inference structures using Mamdani and Takagi-Sugeno structures. In this study, Takagi-Sugeno method in Type 1 structure was preferred due to its ease of processing. In order to produce fuzzy inference in the Mamdani and Takagi-Sugeno structure, the inputs must be fuzzied. As shown in Figure 4, the inputs should be fuzzied first. Then, inference should be made according to the rule table and then defuzzification should be done. In Figure 5, an example fuzzy inference structure is given for Takagi-Sugeno. Input membership functions are given in triangular form and there are x and y inputs. The x inputs are blurred with the μ_X fuzzy set, while the y inputs are blurred with the μ_Y fuzzy set. The output membership function is associated with the determined polynomial membership functions. In the 1st rule, the minimum value η_1 of the value produced from the fuzzy sets μ_X and μ_Y is determined. Then the z_1 value is determined. The same operations are performed for all rules as in Equation 5. Then the output is determined by taking the average of these values as in Equation 6.

$$z_i = \mu_X(X_1) \cap \mu_Y(Y_1) \quad 5$$

$$z_{output} = \frac{\sum_{i=1}^n z_i \eta_i}{\sum_{i=1}^n \eta_i} \quad 6$$

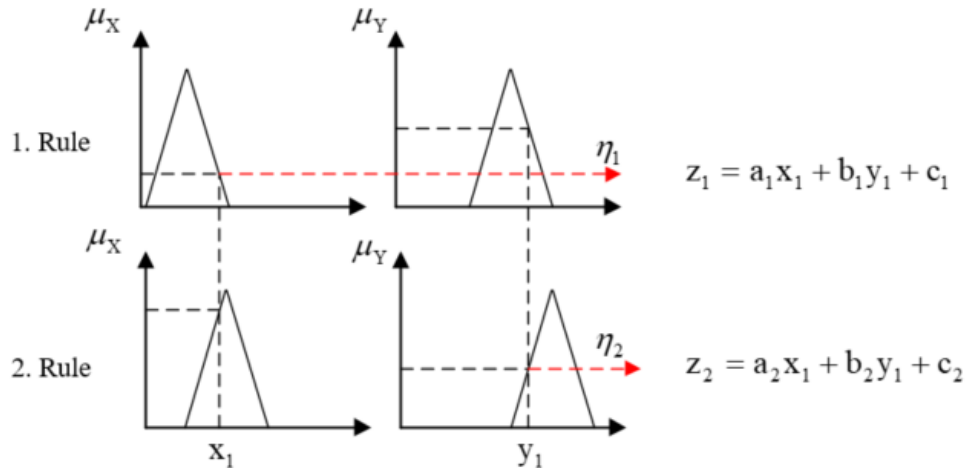


Figure 5. Takagi-Sugeno fuzzy inference system

In particular, fuzzy logic has been successful in overcoming the difficulties in nonlinear system modeling and control. For this reason, it has found a wide variety of uses in the field of control engineering. Fuzzy controllers can have very different inputs and very different outputs used to control the system, depending on the system to be controlled (Gül et al., 2021). A simple fuzzy-PID controller structure is shown in Figure 6. In this controller, both the error and the change of the error are taken as input. Since different situations may occur for different systems, a sufficient number of appropriate control inputs should be determined. Each fuzzy logic produces an output value and a control output is produced for system control by means of these values (Cihan et al., 2008).

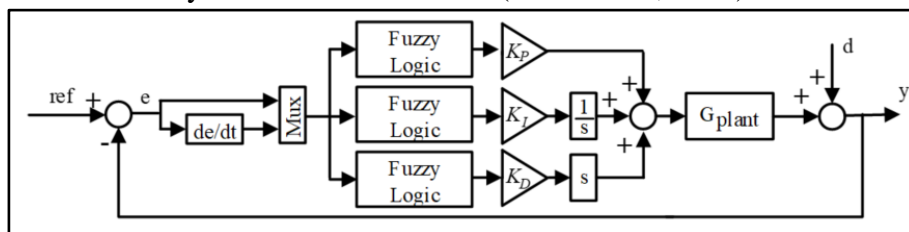


Figure 6. Fuzzy PID control diagram

In the structure given in Figure 7, the signal coming to the PID block is produced directly from fuzzy logic structures. This requires the output of the fuzzy logic to produce values in a very wide range. Instead, in this study, a topology will be used in which the control signal will be generated by adjusting the error signal coming to the inputs of the control parameters. This topology is given in Figure 7.

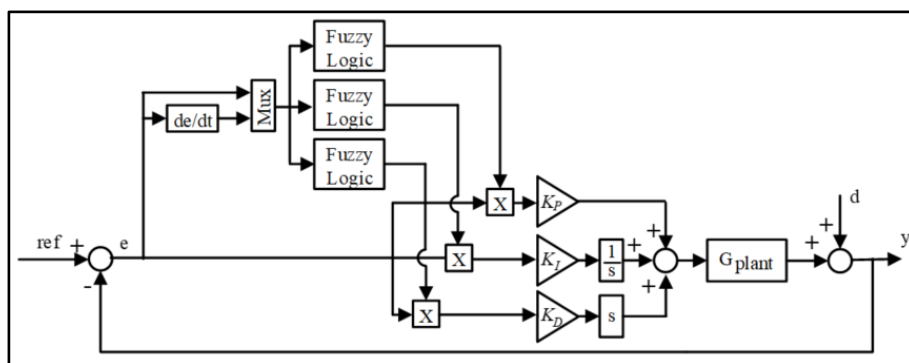


Figure 7. PID control diagram weighted with Fuzzy Logic

Five triangular membership functions are used for the temperature error (e) to be used to control the system. Five triangular memberships were used for the derivative error ($eDot$) of the temperature, which is the other input value to be used. Fixed membership function is used in its output. In this way,

fuzzy logic is made singleton. The membership functions values of temperature error (e) used in fuzzy sets are Negative Large (NL), Negative Small (NS), Zero (ZR), Positive Large (PL), Positive Small (PS). Parameters of (e) membership functions have been determined by genetic algorithm in the range of $[-5, 5]$. The membership functions values of derivative error of temperature ($eDot$) used in fuzzy sets are Negative Large (NL), Negative Small (NS), Zero (ZR), Positive Large (PL), Positive Small (PS). Parameters of membership functions ($eDot$) have been determined by genetic algorithm in the range of $[-1,1]$. Ranges of $eDot$ is smaller than e because time constant of the system is high, that is, system response speed is at low level. On the outputs, Positive Large (PL) is constant 1, Positive Small (PS) is constant 0.5 and Zero (ZR) is constant 0 membership functions are used. The reason why negative constant values are not used in output membership functions; It is possible to prevent the system from entering a possible instability when the controller parameters are weighted with a negative weight. All details about membership function parameters are depicted in Table 6 and Table 8 before each applications. Because these parameters are only valid on the relevant control system. Nevertheless, general fuzzy logic structure of the system used in Matlab is given in Figure 8. In Figure 8, inputs and outputs fuzzy inference structure is given.

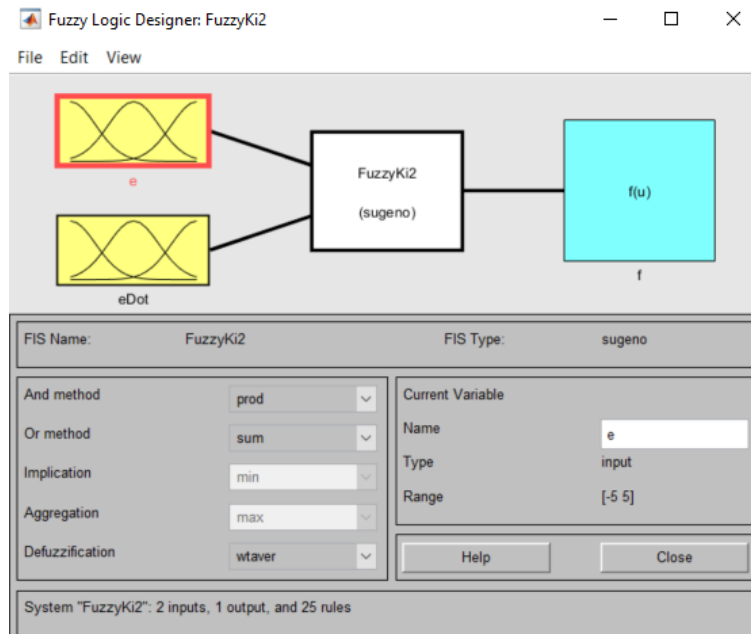


Figure 8. Fuzzy logic interference

A rule table is needed in order to process the fuzzy error and error variation input in fuzzy membership functions and to obtain the appropriate control output from the output membership function in Figure 8. The rule table created for the five-variable case in this study is given in Table 1.

In the created rule table, the “ e ” parameter shows the error and the “ $eDot$ ” parameter shows the change of the error. In this study, that is depicted at Figure 8, the and operator is used as prod in the rule firing method in fuzzy logic control design and the output membership functions are constant values. In Figure 8, the average method was used for defuzzification.

In this study, PID controller with fuzzy logic is used to control the heat exchanger. Since there may be different topologies when fuzzy logic and PID controller are combined, only two different topologies are applied in this study. One of them is the topology where fuzzy logic is applied to the K_p value as shown in Figure 9. The second topology is the fuzzy logic topology of the K_i value given in Figure 10.

Table 1. Fuzzy logic rule table

e/de	NL	NS	ZR	PS	PL
NL	PL	PL	PL	PL	PL
NS	PS	PS	PS	PS	PL
ZR	PS	PS	ZR	PS	PS
PS	PS	PS	PS	PS	PL
PL	PL	PL	PL	PL	PL

Objective Function and Penalty Function

The objective function or fitness function in an optimization problem is a real-valued function whose value is to be either maximized or minimized with regard to a set of possible solution. The objective function is an indispensable part of all optimization problems, starting from our daily life and involving inextricably complex problems. For example, in an intercity journey, it is an objective function to adjust the speed with minimum risk and minimum fuel consumption. On the other hand, completing the journey of the same vehicle in the shortest time is also a different objective function. Optimization is made over the objective function by using variables such as adjusting the speed and taking a break to fulfill this purpose. When the objective function is considered, it is definitely found in places such as Corporate Finance, Investments Production Distribution Purchasing Human Resources, Airlines and Trucking Oil and Gas, Agriculture, Electric Power, Financial Services, where optimization is available. There are many constraints in achieving these goals. An exemplary unimodal objective function to be minimized in the simplest way is given in Figure 11a. An example of a multimodal objective function with more than one local point is given in Figure 11b. However, since everything can be a constraint in real life, Figure 11c and Figure 11d are given to express the constraints in these functions. The minimum values of these two functions did not change under the constraints, but the problems became constrained problems (Çimen et al., 2021b; Garip et al., 2021). Therefore, when constraints are added to the objective function, it becomes a constrained optimization problem.

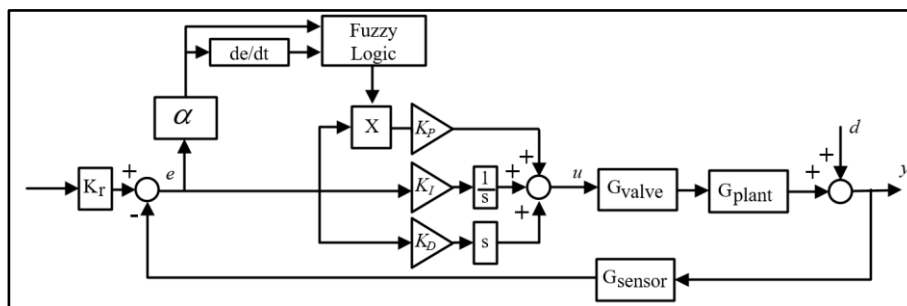


Figure 9. PID controller topology with fuzzed K_p

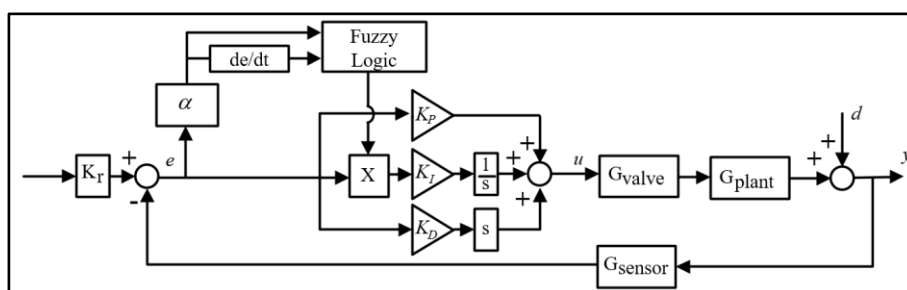


Figure 10. PID controller topology with fuzzed K_i

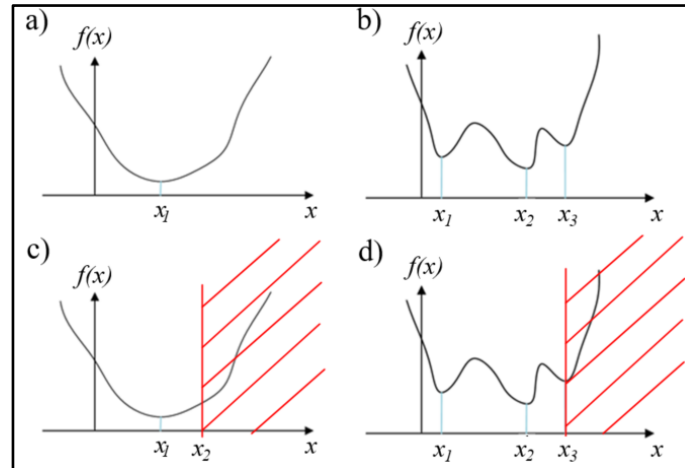


Figure 11. a) unimodal, b) multimodal c) constrained unimodal d) constrained multimodal objective functions

In this study, it is desired to minimize the Integral Square Error (ISE) that is an error between the system response and the reference signal. This equation is given in Equation 7. In addition, an overshoot constraint will be added to the system as it is desired that the overflow remains at a certain level in the system. The overshoot will be accomplished by adding a penalty function to the objective function. This process is given in Equation 8. Therefore, the block diagram for a system to be designed with a PID controller is given in Figure 12. According to the value of the error signal (e), the controller generates a control signal (u) is applied to the system and the system responds (y). Depending on the system response and the reference, a value is generated from Equation 8 with an objective function. The objective is to minimize the value obtained from the objective function for the controller design problem.

$$ISE = \int (ref - y)^2 dt \quad (7)$$

$$f = \begin{cases} ISE & \text{Overshoot is acceptable} \\ ISE + 10^5 & \text{other} \end{cases} \quad (8)$$

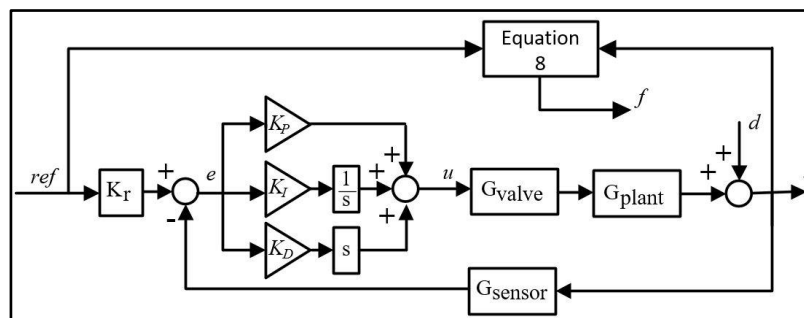


Figure 12. Obtaining the objective function value obtained according to the parameters of the PID controller used of the heat exchanger according to equation 8

Genetic Algorithm

Genetic algorithm is an algorithm designed by inspired by the sequence of genes and chromosomes and developed for solving difficult problems (Çimen et al., 2017). Chromosomes are created by coding the parameters of the problem studied in the first stage of genetic algorithms. An initial population of these chromosomes is then created. Each chromosome in the population is a candidate for solving the problem. Then, some chromosomes are selected from the population and better chromosomes are tried to be obtained by applying "crossover" and "mutation" genetic operators based on genetic processes in nature. With successive iterations, new generations are created and the fitness values of these new

generations are compared with those in the population. Chromosomes with better fitness values among the chromosomes in the new generations are included in the population. In this way, the chromosomes in the population renew themselves with better chromosomes in an evolutionary process. This process continues until the number of new generations reaches a certain number or until there is a certain amount of improvement in fitness values in the population (Çolak, 2010; Mirjalili, 2019; Sourabh et al., 2021). Pseudocode of this algorithm is given in Figure 13.

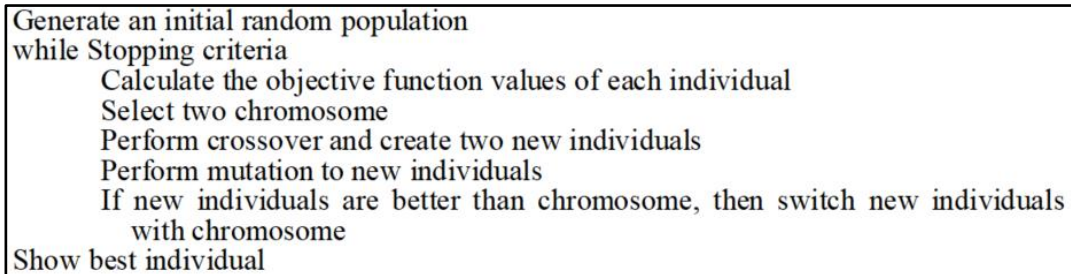


Figure 13. Pseudocode of genetic algorithm

Controller Design

The general structure in Figure 14 is used to design the controller for the heat exchanger. In this structure, the design of the PID controller with Genetic Algorithm is shown, since it is basic. In this method in which the genetic algorithm is run, PID parameters are tried to be determined so that Equation 8 is the minimum for all population values. The fuzzy PID structure is also similar. Controller design was carried out by adding fuzzy logic block structures as in Figure 15 and Figure 16. The error signal (e) that enters the fuzzy logic is multiplied by α to decrease the input value of fuzzy logic. This α value was determined as 0.1 in the study.

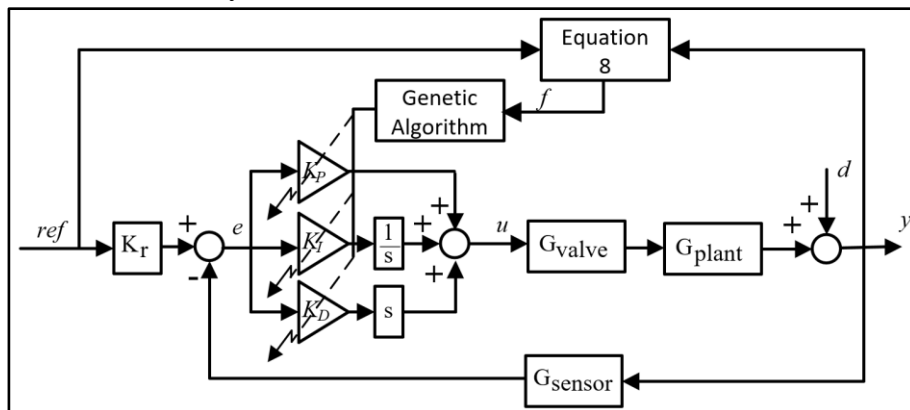


Figure 14. Determination of PID controller parameters used in the control of the heat exchanger by genetic algorithm

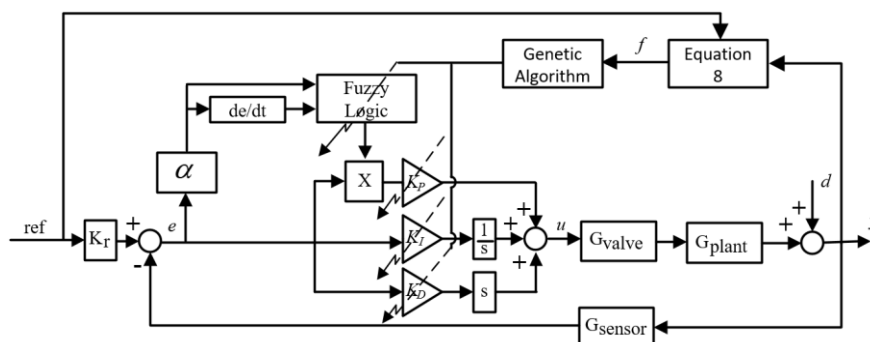


Figure 15. Determination of PID controller with fuzzy K_p parameters used in the control of the heat exchanger by genetic algorithm

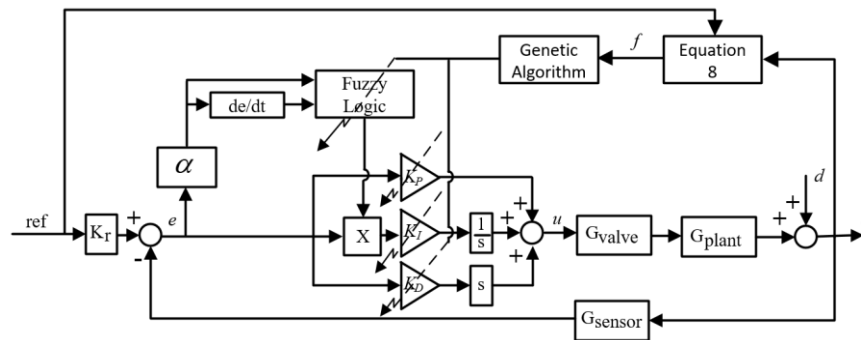


Figure 16. Determination of PID controller with fuzzy Ki parameters used in the control of the heat exchanger by genetic algorithm

RESULTS AND DISCUSSION

All simulation studies were performed on a computer with Intel (R) Core (TM) i5-3470 CPU @ 3.20 Ghz, 64 Bit, 8 GB RAM in Windows 10 operating system. The simulation run time of Heat exchanger is chosen as 400 sec and the set point temperature is kept at 90°C. Firstly PID controller are designs by using Zeigler Nichols Step method, Zeigler Nichols frequency method, AMIGO step method and AMIGO frequency method. Secondly, by using a Genetic Algorithm, PID controller designs are conducted with regard to for each 2%, 5% and 10% overshoots constrained. Then, Fuzzy PID controllers for two topology are designed by using Genetic Algorithm for Heat Exchanger System with regard to for each 2%, 5% and 10% overshoots constrained. Since Genetic Algorithm was used, its parameter values are given in Table 2. In addition, the parameters of the variables to be determined in the Genetic Algorithm are given in Table 3. The results are given in Table 4, 5, 6, 7, 8, 9 and 10. Best results in in Tables in terms of optimization are written bold. While Table 4-9 presents individual results with respect to control, Table 10 summaries all results considered, that enable us to see from a wider perspective.

Table 2. Genetic Algorithm parameters

Constituent	Value or Method
Population size	100
Maximum iteration	500
Crossover	0.8
Mutation	0.01
Selection	Stochastic Uniform

Table 3. Parameters ranges to be optimized in Genetic Algorithm

PID Parameter	Range of PID Parameter	e Membership Function	Range of Triangle Membership Function Parameter	eDot Membership Function	Range of Triangle Membership Function Parameter
Kp	[0 1000]	NL	[-5 5]	NL	[-1 1]
Ki	[0 2]	NS	[-5 5]	NS	[-1 1]
Kd	[0 1000]	ZR	[-5 5]	ZR	[-1 1]
		PS	[-5 5]	PS	[-1 1]
		PL	[-5 5]	PL	[-1 1]

Classical PID Design

First, the results obtained when the controller was designed with classical methods are given in Table 4. As can be seen from the table, the performance values give high results classically. In addition, when Figure 15 is examined, it has given results in a way that supports this result. When we look at the classical methods, it is seen that it is very high in general and the sitting time is high. However, when

the Controller obtained in (Reddy et al., 2020)'s study is used, it has been observed that it gives better results than classical methods.

Table 4. Controller parameters and objective function values designed with a classical PID controller

	Kp	Ki	Kd	Overshoot	Settling Time (sec)	ISE
Zeigler Nichold Step Response	23.1826	1.6559	85.8419	104.67%	137	3 048.6424
Zeigler Nichold Frequency	44.0842	3.0614	158.7030	138.67%	95	2 293.7730
AMIGO Step Response	9.6012	0.4508	32.9506	41.89%	104.5	3 128.7820
AMIGO Frequency	22.5568	22.5568	94.7950	88.89%	111.5	2 667.6512
PID (Reddy et al., 2020)	13.25	0.36	56.25	27.11%	56	2 505.9090

PID design with overshoot constraint by using genetic algorithm

The results obtained when the PID controller designed by using Genetic Algorithm by adding the overshoot constraint, are given in Table 5. As can be seen from the table, it is seen that the objective function value decreases as the overshoot constraint value softens. The obtained settling time for the designed controllers is also given in the table. Since settling time is not considered as a constraint in this study, it is given only as one of the results. Therefore, the comparison was generally made on the objective function. In addition, it was observed that the duration of the simulation decreased with the increase in the overshoot constraint softens. The reason for this is that although the maximum number of iterations is constant since the constraints direct the search space first, it has been seen that making the constraints more difficult increases the time in the solution. However, as the constraints are eased, it is seen that the objective function value, which is the minimum value, decreases even more. Therefore, as the constraints change, a different optimal solution has been found, so the objective function value and the controller parameters have changed. On the other hand, the results obtained are given in Figure 18. When the results were examined, all results gave the expected results as they met the constraints. However, in the constrained condition, 10% exceeding gave the best result in terms of the overshoot constraint. As it can be understood from here, a lower objective function value can be found in the absence of an overshoot. In addition, it has been seen that it produces better results than the classical PID controller result obtained in (Reddy et al., 2020).

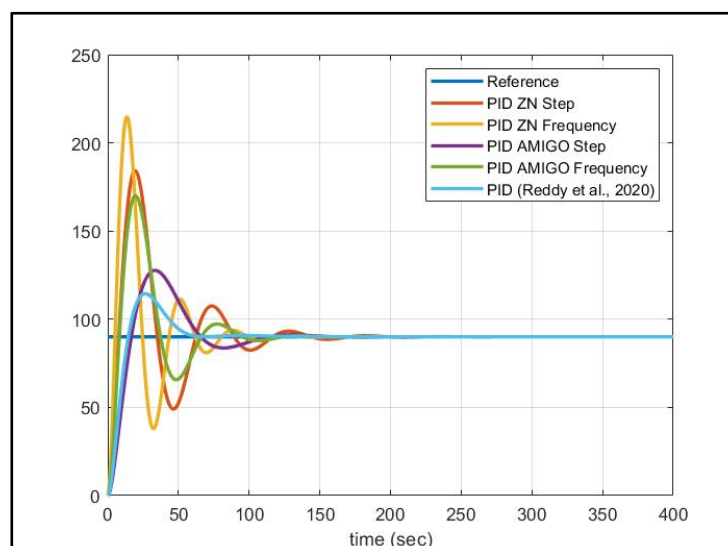
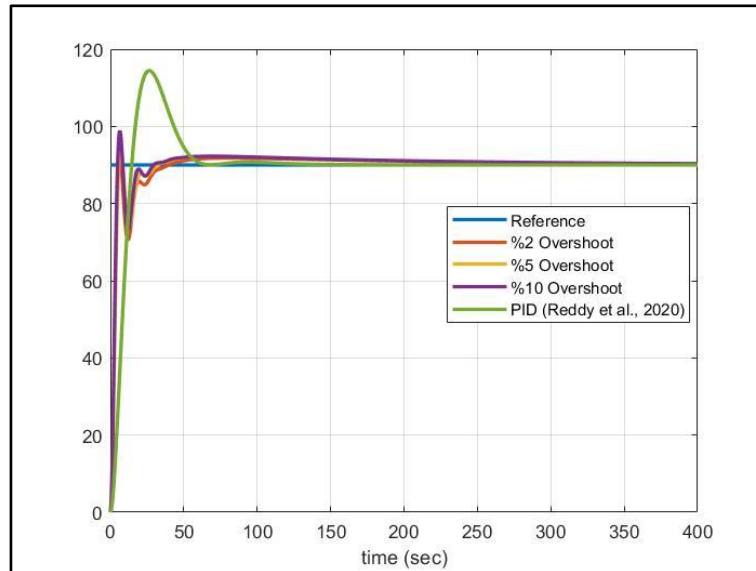


Figure 17. System responses of PID methods designed for heat exchanger control

Table 5. PID controller results with overshoot constrained designed with genetic algorithm

Overshoot Constraint	Kp	Ki	Kd	Overshoot	Settling Time (sec)	Processing Time (sec)	ISE	Objective Function Value
2%	69	0.3711	930.2710	1.98%	83.5	3 119.6	1772.2	1772.2
5%	65	0.4109	830.7716	4.86%	120.5	2 086.2	1730.3	1730.3
10%	75	0.4477	932.5324	9.85%	118.5	678.6	1647	1647

**Figure 18.** System responses of the PID controller designed with GA for the control of the heat exchanger

Design of PID controller that is included overshoot constraint and Fuzzy Kp using genetic algorithm

The results obtained when PID controller with overshoot Fuzzy Kp was designed with genetic algorithm are given in Table 6 and Table 7. As can be seen from the Table 7, it is seen that the objective function value decreases as the overshoot constraint value softens. The obtained settling times for the designed controllers are also given in the table. Since settling time is not considered as a constraint in this study, it is given only as one of the results. Therefore, the comparison was generally made on the objective function. In addition, it was observed that the duration of the simulation decreased with the increase in the overshoot constraint softens. The reason for this is that although the maximum number of iterations is constant since the constraints direct the search space first, it has been seen that making the constraints more difficult increases the time in the solution. However, as the constraints are eased, it is seen that the objective function value, which is the minimum value, decreases even more. Therefore, as the constraints change, a different optimal solution has been found, so the objective function value and the controller parameters have changed. On the other hand, the results obtained are given in Figure 19. When the results were examined, all results gave the expected results as they met the constraints. However, in the constrained condition, 10% exceeding gave the best result in terms of the objective function value. As it can be understood from here, a lower objective function value can be found in the absence of an overshoot. In addition, it was seen that the results with PID controller with Fuzzy Kp containing 5% and 10% overshoot constraint designed by genetic algorithm produced better results than the classical PID controller results obtained in (Reddy et al., 2020).

Table 6. Results of PID controller design that is included overshoot constraint and fuzzy Ki using genetic algorithm

Overshoot Constraint	e Input Membership functions				
	NL	NS	ZR	PS	PL
2%	[-5 -5 -0.3452]	[-5 -0.3452 1.342]	[-0.3452 1.342 2.115]	[1.342 2.115 5]	[2.115 5 5]
5%	[-5 -5 -0.1559]	[-5 -0.1559 2.782]	[-0.1559 2.782 3.549]	[2.782 3.549 5]	[3.549 5 5]
10%	[-5 -5 -3.085]	[-5 -3.085 1.479]	[-3.085 1.479 1.677]	[1.479 1.677 5]	[1.677 5 5]
Overshoot Constraint	eDot Input Membership functions				
	NL	NS	ZR	PS	PL
2%	[-1 -1 -0.702]	[-1 -0.702 -0.5346]	[-0.702 -0.5346 -0.03991]	[-0.5346 -0.03991 1]	[-0.03991 1 1]
5%	[-1 -1 -0.5745]	[-1 -0.5745 -0.1723]	[-0.5745 -0.1723 0.2739]	[-0.1723 0.2739 1]	[0.2739 1 1]
10%	[-1 -1 -0.1958]	[-1 -0.1958 -0.1739]	[-0.1958 -0.1739 0.03285]	[-0.1739 0.03285 1]	[0.03285 1 1]
Overshoot Constraint	Output Membership functions				
	NL	NS	ZR	PS	PL
2%	-	-	0	0.5	1
5%	-	-	0	0.5	1
10%	-	-	0	0.5	1

Design of PID controller that is included overshoot constraint and Fuzzy Ki using genetic algorithm

The results obtained when PID controller with overshoot Fuzzy Ki was designed with genetic algorithm are given in Table 8 and Table 9. As can be seen from the Table 9, it is seen that the objective function value decreases as the overshoot constraint value softens. The obtained settling times for the designed controllers are also given in the table. Since settling time is not considered as a constraint in this study, it is given only as one of the results. Therefore, the comparison was generally made on the objective function. In addition, it was observed that the duration of the simulation decreased with the increase in the overshoot constraint softens. The reason for this is that although the maximum number of iterations is constant since the constraints direct the search space first, it has been seen that making the constraints more difficult increases the time in the solution. However, as the constraints are eased, it is seen that the objective function value, which is the minimum value, decreases even more. Therefore, as the constraints change, a different optimal solution has been found, so the objective function value and the controller parameters have changed. On the other hand, the results obtained are given in Figure 20. When the results were examined, all results gave the expected results as they met the constraints. However, in the constrained condition, 10% exceeding gave the best result in terms of the objective function value. As it can be understood from here, a lower objective function value can be found in the absence of an overshoot. In addition, it was seen that the results with PID controller with Fuzzy Ki containing overshoot constraints designed by genetic algorithm produced better results than the classical PID controller results obtained in (Reddy et al., 2020).

Table 7. Results of PID controller design that is included overshoot constraint and fuzzy Ki using genetic algorithm

Overshoot Constraint	Kp	Ki	Kd	α	Overshoot	Settling Time (sec)	Processing Time (sec)	ISE	Objective Function Value
2%	143	0.2730	911.3536	0.1	0.18%	24.5	6 658.005	1 759. 8742	1 759. 8742
5%	208	0.2508	824.8791	0.1	3.81%	15	22 665.8313	1 629.5293	1 629.5293
10%	276	0.2777	807.8062	0.1	9.94%	18.5	25 417.0475	1 465.7174	1 465.7174

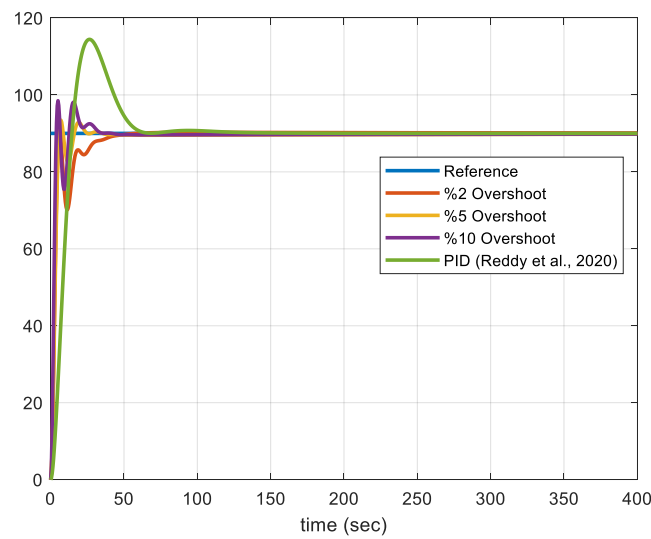


Figure 19. System responses of PID controller design that is included overshoot constraint and fuzzy Kp using genetic algorithm

Table 8. Results of PID controller design that is included overshoot constraint and fuzzy Ki using genetic algorithm

Overshoot Constraint	e Input Membership functions				
	NL	NS	ZR	PS	PL
2%	[-5 -5 -4.109]	[-5 -4.109 -0.09518]	[-4.109 -0.09518 0.4757]	[-0.09518 0.4757 5]	[0.4757 5 5]
5%	[-5 -5 0.2501]	[-5 0.2501 0.7231]	[0.2501 0.7231 1.484]	[0.7231 1.484 5]	[1.484 5 5]
10%	[-5 -5 0.1114]	[-5 0.1114 0.1892]	[0.1114 0.1892 0.7133]	[0.1892 0.7133 5]	[0.7133 5 5]
	eDot Input Membership functions				
	NL	NS	ZR	PS	PL
2%	[-1 -1 -0.425]	[-1 -0.425 -0.3268]	[-0.425 -0.3268 0.05124]	[-0.3268 0.05124 1]	[0.05124 1 1]
5%	[-1 -1 -0.6435]	[-1 -0.6435 -0.6017]	[-0.6435 -0.6017 -0.4445]	[-0.6017 -0.4445 1]	[-0.4445 1 1]
10%	[-1 -1 -0.3325]	[-1 -0.3325 0.1082]	[-0.3325 0.1082 0.1354]	[0.1082 0.1354 1]	[0.1354 1 1]
	Output Membership functions				
	NL	NS	ZR	PS	PL
2%	-	-	0	0.5	1
5%	-	-	0	0.5	1
10%	-	-	0	0.5	1

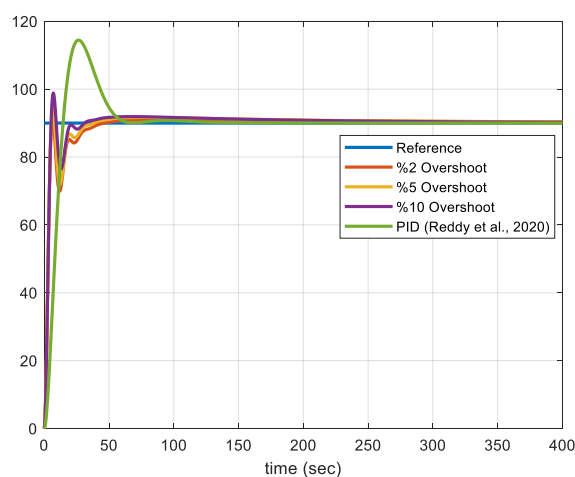


Figure 20. System responses of PID controller design that is included overshoot constraint and fuzzy Ki using genetic algorithm

Table 10 is given in order to evaluate the obtained results more collectively. In this table, the ISE value obtained among the unconstrained results obtained is 2 505.9090 in the study of Reddy. However, PID+Genetic results performed under constrained conditions produced better results than unconstrained results. Because the constrained condition includes the unconstrained condition. When the constrained conditions are examined, it is seen that the PID controller with fuzzy Kp optimized with the genetic algorithm always gives the best results.

Table 9. Results of PID controller design that is included overshoot constraint and fuzzy Ki using genetic algorithm

Overshoot Constraint	Kp	Ki	Kd	α	Overshoot	Settling Time (sec)	Processing Time (sec)	ISE	Objective Function Value
2%	70	0.6003	948.8063	0.1	1.80%	25.5	22 045.4023	1 776.0171	1 776.0171
5%	70	0.5872	910.1589	0.1	4.85%	17.5	24 224.0464	1 725.8108	1 725.8108
10%	67	0.8774	812.3448	0.1	9.84%	17	27 020.5598	1 651.6018	1 651.6018

Table 10. Comparison of Controller Results with respect to objective function (ISE+constraint violation)

Overshoot Constraint	Zeigler Nichold Step Response	Zeigler Nichold Frequency Response	AMIGO Step Response	AMIGO Frequency Response	PID (Reddy et al., 2020)	PID + Genetic	PID with Fuzzy Kp + Genetic	PID with Fuzzy Ki + Genetic
No Constraint	3 048.642	2 293.7730	3 128.7820	2 667.6512	2 505.9090	-	-	-
2%	Violation occurs	Violation occurs	Violation occurs	Violation occurs	Violation occurs	1772.2	1 759. 8742	1 776.0171
5%	Violation occurs	Violation occurs	Violation occurs	Violation occurs	Violation occurs	1730.3	1 629.5293	1 725.8108
10%	Violation occurs	Violation occurs	Violation occurs	Violation occurs	Violation occurs	1647	1 465.7174	1 651.6018

CONCLUSION

In this study, Zeigler Nichols step, Zeigler Nichols frequency, AMIGO step and AMIGO frequency methods are used for the PID controller in the control of the heat exchanger. Then, two different topologies were used for the Fuzzy PID controller to be used in the control of the heat exchanger. As the originality of the study, both PID and Fuzzy PID controllers were designed with genetic algorithm by adding different overshoot constraints to according to the ISE performance. When the results were examined, it is seen that the PID controller with Fuzzy Kp optimized by genetic algorithm gives better results than other methods. In further work, the authors will design controllers using Type 2 fuzzy logic.

Conflict of Interest

The article authors declare that there is no conflict of interest between them.

Author's Contributions

The authors declare that they have contributed equally to the article.

REFERENCES

- Åström KJ, Hägglund T, 2004. Revisiting The Ziegler-Nichols Tuning Rules For PI Control—Part II The Frequency Response Method. *Asian Journal of Control*, 6(4): 469–82.

- Boz AF, Çimen ME, 2017a. An Interface Design for Controlling Dead Time Systems Using PSO, CS and FA Algorithms. International Advanced Technologies Symposium (IATS'17), 19-22 October, Elazığ, Turkey.
- Boz AF, Çimen ME, 2017b. PID Controller Design Using Improved FireFly Algorithm. International Advanced Technologies Symposium (IATS'17), 19-22 October, Elazığ, Turkey.
- Cihan A, Karakuzu C, 2002. Bulanık-PID Kontrolör Parametrelerinin Diferansiyel Gelişim Algoritması İle En Uygunlaması. ELECO 2008 Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu, 26-30 Kasım, Bursa, Türkiye.
- Çimen ME, Boyraz ÖF, Pala MA, Boz AF, Yıldız MZ, 2019. Ölü Zamanlı Sistemlerde Kullanılan Smith Predictor İçin Balina Sürüsü Optimizasyonu İle PID Tasarımı. Academic Perspective Procedia 2(3): 583–92.
- Çimen ME, Boz AF, 2017. İkinci Dereceden Ölü Zamanlı ve Geri Tepmeli Sistem Parametrelerinin, Röle Testi ve PSO, CS, FA Algoritmaları İle Belirlenmesi. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 34(1): 461–78.
- Çimen ME, Garip Z, Boz AF, 2021a. Chaotic Flower Pollination Algorithm Based Optimal PID Controller Design for a Buck Converter. Analog Integrated Circuits and Signal Processing, 107(2): 281–98.
- Çimen ME, Garip Z, Boz AF, 2021b. Comparison of Metaheuristic Optimization Algorithms with a New Modified Deb Feasibility Constraint Handling Technique. Turkish Journal of Electrical Engineering and Computer Science.
- Çolak S, 2010. Genetik Algoritmalar Yardımı İle Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama. Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi, 19(3): 423–38.
- Garip Z, Çimen ME, Boz AF, 2021. Application of Harris Hawks and Whale Optimization Algorithm with Constraint Handling Techniques: A Comparative Study. Journal of Intelligent Systems: Theory and Applications, 4(2): 76–85.
- Gül V, Şahin S, 2021. Kesintisiz Güç Kaynağı Çıkış Gücü Düzenlemesi İçin Bulanık Mantık ve Kazanç Çizelgesi Uyarlanı Tabanlı PI Kontrolörlerin Performans Karşılaştırılması. Avrupa Bilim ve Teknoloji Dergisi, 24: 416–20.
- Hägglund T, Åström KJ, 2002. Revisiting the Ziegler-Nichols Tuning Rules for PI Control. Asian Journal of Control, 4(4): 364–80.
- Hamid AHN, Mahanijah K, Yahaya FH, 2009. Application of PID Controller in Controlling Refrigerator Temperature. International Colloquium on Signal Processing & Its Applications, 378–84, Malaysia.
- Isa AI, Hamza MF, 2014. Effect of Sampling Time on PID Controller Design for a Heat Exchanger System. IEEE 6th International Conference on Adaptive Science & Technology (ICAST), Ota, Nigeria.
- Kaplan K, Kuncan M, Polat H, Tepe H, Ertunç HM, 2020. PID ve Bulanık Mantık Tabanlı DC Motorun Gerçek Zamanlı Konum Kontrolü. Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 10(2): 900–916.
- Katoch S, Chauhan SS, Kumar V, 2021. A Review on Genetic Algorithm: Past, Present, and Future. Multimedia Tools and Applications, 80(5): 8091–8126.
- Kesavan E, Gowrhaman N, Tharani S, Manoharan S, Arunkumar E, 2016. A Publication of IIETA Design and Implementation of Internal Model Control and Particle Swarm Optimization Based PID for Heat Exchanger System, 34(3): 386–90.

- Khames A, Lesewed AA, Al-mathnani AO, 2020. Synthesis of Sliding Mode Control for Heat Exchanger. Third International Conference on Technical Sciences (ICST2020), 28 – 30 November 2020, Tripoli, Libya.
- Khare BK, Singh Y, 2010. PID Control of Heat Exchanger System. *International Journal of Computer Applications*, 8(6): 22–27.
- Kishore K, Jalalu G, Sumalatha, A, Prasanti K, 2013. Control of Heat Exchanger Using Hybrid Fuzzy-PI. *International Journal of Engineering Research and Applications (IJERA)*, 3(4): 1396–1400.
- Mirjalili S, 2019. Genetic Algorithm. *Evolutionary algorithms and neural networks*: 43–55.
- Mraz M, 2001. The Design of Intelligent Control of a Kitchen Refrigerator. *Mathematics and computers in simulation*, 56(3): 259–67.
- Padhee S, Khare YB, Singh Y, 2011. Internal Model Based PID Control of Shell and Tube Heat Exchanger System. *IEEE Technology Students' Symposium*, Kharagpur, India, 14-16 January, 297–302.
- Peçe F, Yazar E, Karabay S. 2020. PID ve Bulanık Mantık Kontrol Sistemleri İle İki Tekerlekli Kendini Dengeleyebilen Robotik Sistem Tasarımı. *Kocaeli Üniversitesi Fen Bilimleri Dergisi*, 3(1): 99–108.
- Rajagopal K, Çimen ME, Jafari S, Singh JP, Roy BK, Akmeser OM, Akgül A, 2021. A Family of Circulant Megastable Chaotic Oscillators, Its Application for the Detection of a Feeble Signal and PID Controller for Time-Delay Systems by Using Chaotic SCA Algorithm. *Chaos, Solitons and Fractals* 148 (May): 110992. <https://doi.org/10.1016/j.chaos.2021.110992>.
- Reddy CS, Balaji K, 2020. A Genetic Algorithm (GA)-PID Controller for Temperature Control in Shell and Tube Heat Exchanger. *1st International Conference on Computational Engineering and Material Science (ICCEMS - 2020)* 17-18 July, Karnataka, India.
- Soesanti I, Syahputra R, 2019. A Fuzzy Logic Controller Approach for Controlling Heat Exchanger Temperature. *Journal of Electrical Technology UMY*, 3(4): 117–24.
- Trafczynski M, Markowski M, Alabrudzinski S, Urbaniec K, 2016. The Influence of Fouling on the Dynamic Behavior of PID-Controlled Heat Exchangers. *Applied Thermal Engineering*, 109: 727–38.
- Tridianto, Erik et al. 2017. Cascaded PID Temperature Controller for FOPDT Model of Shell-and-Tube Heat Exchanger Based on Matlab / Simulink. *International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 26-27 September, Surabaya, Indonesia 185–91.
- Vasičkaninová A, Bakošová M, Mészáros A, Klemeš, JK Vasi, Anna, Monika B, 2011. Neural Network Predictive Control of a Heat Exchanger. *Applied Thermal Engineering*, 31(13): 2094–2100.
- Yılmaz M, Can K, Başçı A, 2021. PI+Feed Forward Controller Tuning Based on Genetic Algorithm for Liquid Level Control of Coupled-Tank System. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi* 11(2): 1014–26.
- Zadeh LA, 1996. Fuzzy Sets. *Fuzzy sets, fuzzy logic, and fuzzy systems*: 394–432.
- Ziegler JG, Nichols NB, 1942. Optimum Settings for Automatic Controllers. *ASME* 64(11).
- Ziegler JG, Nichols NB, 1943. Process Lags in Automatic Control Circuits. *ASME* 65(5): 433–43.