Comparison of PSO-PID, FLC and PID in a Circulating Fluidized Bed Boiler

Hilmi Aygun[1], Huseyin Demirel[2]

[1] Department of Electrical and Electronics Engineering, Karabuk University, Karabuk, Turkey
hilmiaygun@karabuk.edu.tr
[2] Department of Electrical and Electronics Engineering, Karabuk University, Karabuk, Turkey
hdemirel@karabuk.edu.tr

Abstract

Bed temperature is one of the most important parameters in a circulating fluidized bed boiler. Because combustion efficiency increases with rising of bed temperature and also harmful emissions are affected by bed temperature. Classical PID controller can't give fast response to instant changes in boiler. So modern control methods must be used for bed temperature control of circulating fluidized bed boiler. In this study PSO-PID (Particle Swarm Optimization based PID), FLC which are modern controllers, and classical PID controller are used for controlling bed temperature of a circulating fluidized bed boiler and they are compared. Simulation results show that settling time in PSO-PID controller is lower than the other controllers. PSO-PID decreases overshoot like FLC but in classical PID controller overshoots are the biggest. In FLC there are no overshoot. So if overshoot isn't wanted, a FLC must be used and if short settling time is wanted, a PSO-PID controller must be used.

1. Introduction

In the boilers when the fossil fuels are burned, they spread harmful emissions. So these emissions damage our health. Using fluidized bed boiler is required for protection of health of people. In circulating fluidized bed boiler SO_2 gase is captured by using limestone and combustion efficiency increases with rising of bed temperature by providing an excellent mixture of gase and solid. Because the particles which leave from boiler, are gathered in the cyclone and they return to boiler. NO_x emission increases linearly when bed temperature rises. But CO emission decreases with rising bed temperature. So control of bed temperature is very important for decreasing these harmful emissions and increasing combustion efficiency in circulating fluidized bed boiler.

B. Lixia, Z. Junxia and F. Song made mathematical modeling of bed temperature of a circulating fluidized bed boiler and they showed that dynamic characteristic of bed temperature changes with quantity of coal [1]. Ping Fu et al. controlled bed temperature with a fuzzy logic based PID controller and classical PID controller by using this model. They showed that fuzzy logic based PID controller is more effective than classical PID controller [2]. Ali Akbar Jalali and Aboozar Hadavand controlled bed temperature with H∞ algorithm by using the same model. They showed that H∞ controller decreases settling time of the system and big overshoots which aren't wanted, happen [3].

In this study modern control methods are used for controlling bed temperature of this circulating fluidized bed boiler. One of the modern control methods is particle swarm optimization based PID controller and the other one is fuzzy logic controller. Also bed temperature is controlled by classical PID controller. Particle swarm optimization is developed by inspiring from behaviours of bird swarms by J. Kennedy and R. C. Eberhart [4]. In this method system starts with a population which includes random solutions. In PSO particles wander in search space for finding best solution by watching optimum particle. Fuzzy logic is brought out by L. A. Zadeh and he showed that machines can control the systems by using uncertain informations [5]. Fuzzy logic controller determines output signal according to membership functions of input signals and fuzzy rules.

2. Dynamic Characteristic of Circulating Fluidized Bed Boiler

Mathematical model obtained by B. Lixia is shown at Equation 1 [1].

$$G_p(s) = \frac{(1-\alpha s)}{(1+T_p s)^2} K_p e^{-\tau s} \tag{1}$$

At Table 1 the numerical values of K_p, T_p and τ are given for different boiler loads. Here K_p is the static gain, T_p is time constant, τ is time delay varying with the condition of the boiler, α is about 12. When boiler load changes between 25% and 100%, K_p changes 5 and 10, T_p changes between 100 and 200, τ changes between 30 and 60.

Table 1. Parameters at different boiler loads

Parameters	Under 25% boiler load	Under 65% boiler load	Under 100% boiler load
K_p	5	7.5	10
T_p	100	150	200
τ	30	45	60

At Figure 1 open loop step response curves are showed for different boiler loads. Curve a shows the open loop step response for 25% boiler load, curve b shows the open loop step response for 65% boiler load and curve c shows the open loop step response for 100% boiler load. The dynamic characteristic of bed temperature of circulating fluidized bed boiler changes at different boiler loads and these changes complicate the control of bed temperature. Delay time and passing time to stability for the system increase in parallel with boiler load.

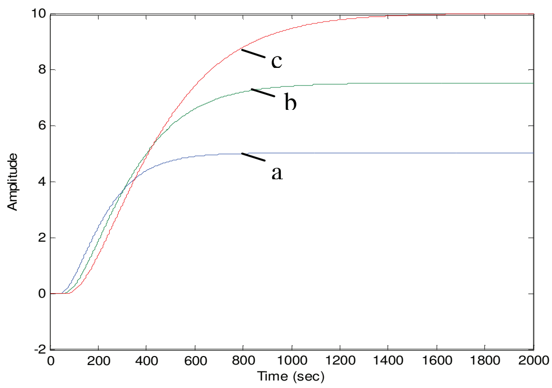


Figure 1. Open Loop Step Responses

3. Classical PID Controller

PID control consists of three base control effects. Control law of PID controller is shown in Equation 2 and Equation 3.

$$m(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de}{dt} \qquad (2)$$

Or

$$m(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de}{dt}\right) \qquad (3)$$

In this study PID parameters are determined by Ziegler Nichols method [12]. At Table 2 the numerical values of PID parameters are given for different boiler loads.

Table 2. PID Parameters

Parameters	Under 25% boiler load	Under 65% boiler load	Under 100% boiler load
P	0.66	0.48	0.378
I	0.0044	0.0022	0.0013
D	24.75	25.2	26.46

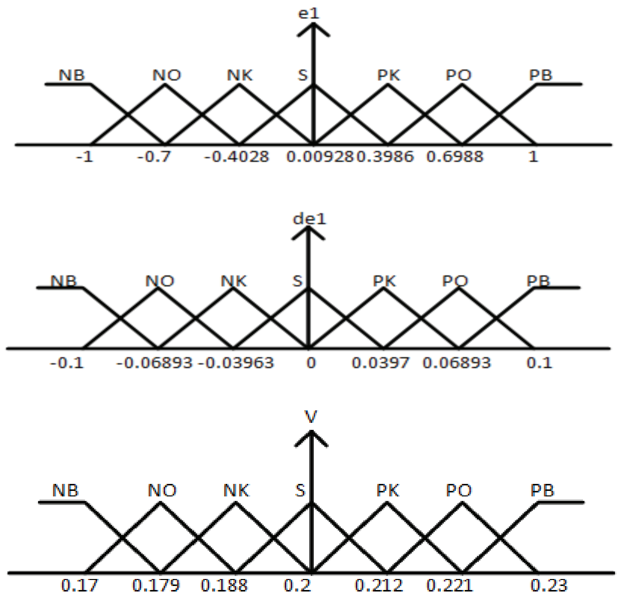
4. Design of Fuzzy Logic Based Controller

Input and output variables are constituted by using 7 membership functions. They are NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium) and PB (Positive Big). NB and PB are selected as trapezoid membership function. Because they are peak values of control ranges. The others are selected as triangle membership function for the purpose of making control ranges more sensitive.

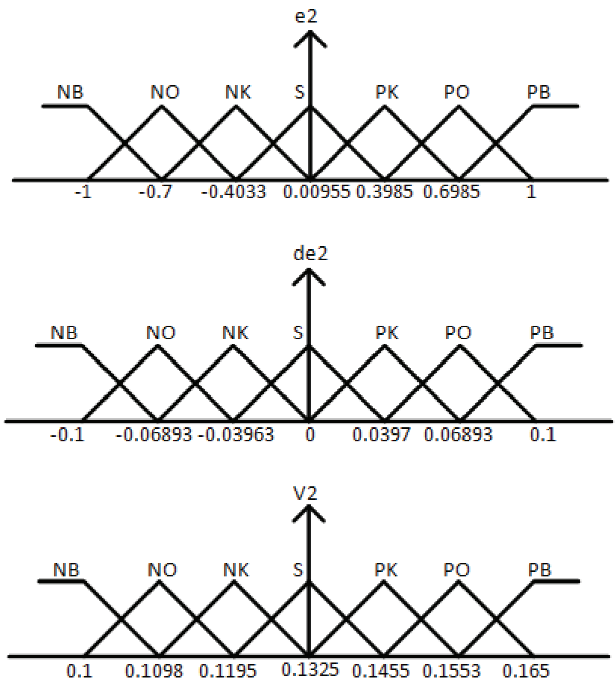
4.1. Intervals of Inputs and Outputs

Error and change of error are two important parameters in fuzzy logic control algorithm. Also output of controller must be determined sensitively. Determined input and output intervals for different boiler loads are shown in Figure 2.

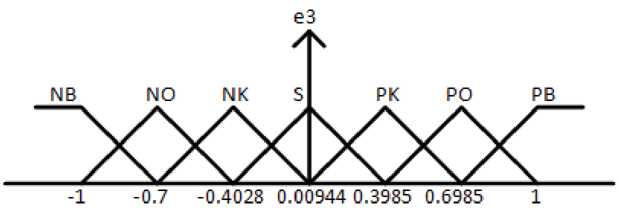
Here e is error value, de is change of error and v is control signal.

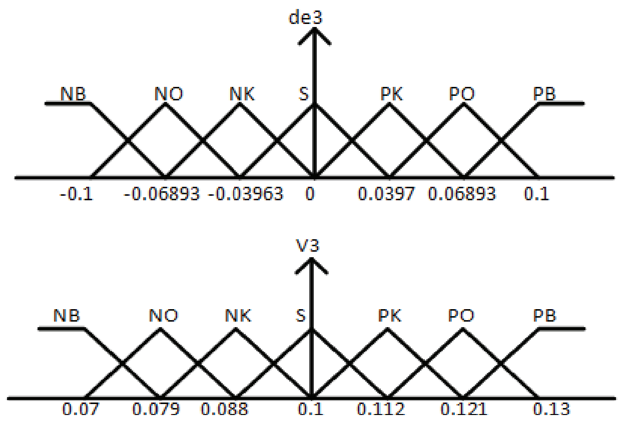


(a)



(b)





(c)

Figure 2. Intervals of Inputs and Outputs
a)For 25% Boiler Load b)For 65% Boiler Load
c)For 100% Boiler Load

4.2. Extraction of Fuzzy Rules

We know that error value is difference between desired value and measured value for output. For example if membership function of error is PB, it means that measured value is much smaller than desired value. If membership function of change of error is PB, it means that error value is tend to increase. Then membership function of control signal must be PB. So output value approaches to desired value. When all of the rules are determined like this, desired value is reached. Extraction of fuzzy rules is given in Table 3.

Table 3. Extraction of Fuzzy Rules

de / e	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NB	NM	NM
NM	NM	NM	NM	NM	NM	NS	NS
NS	NS	NS	NS	NS	NS	Z	Z
Z	Z	Z	Z	Z	Z	PS	PS
PS	PS	PS	PS	PS	PS	PM	PM
PM	PM	PM	PM	PM	PM	PM	PB
PB	PB	PB	PB	PB	PB	PB	PB

In Table 3 there are seven membership functions for error and change of error and there are 49 rules. Here we control bed temperature more sensitive with dividing input signals to a lot of membership functions like this.

5. Algorithm of Particle Swarm Optimization

When a particle moves, it sends its coordinates to a function and its fitness value is measured. In search space new position and speed of particle are calculated according to local best position and global best position. Dimension of solution space can change according to number of unknown parameters in the problem. Here unknown parameters are P, I and D and so solution space has three dimension. Each one of particles which has three parameters, is shown in Equation 4.

$$x_i = (x_{i1}\ x_{i2}\ x_{i3}), \qquad i = 1, 2, 3, \ldots, M \tag{4}$$

The speed of i'th particle is showed in Equation 5.

$$v_i = (v_{i1}\ v_{i2}\ v_{i3}), \qquad i = 1, 2, 3, \ldots, M \tag{5}$$

Best position of each particle is called local best position and best position for each parameters is called global best position. They are shown in Equation 6 and Equation 7.

$$p_i = (p_{i1}\ p_{i2}\ p_{i3}), \qquad i = 1, 2, 3, \ldots, M \tag{6}$$

$$g_i = (g_{i1}\ g_{i2}\ g_{i3}), \qquad i = 1, 2, 3, \ldots, M \tag{7}$$

Speeds of particles are weighted with different random terms for reaching to the best position. The speed and particle of each particle are updated in Equation 8 and Equation 9 [6,7].

$$v_{iD}^{t+1} = K.[v_{iD}^t + c_1 r_1 (p_{iD}^t - x_{iD}^t) + c_2 r_2 (g_i^t - x_{iD}^t)] \tag{8}$$

$$x_{iD}^{t+1} = x_{iD}^t + v_{iD}^{t+1} \tag{9}$$

Here the first term is speed of particle at that moment. The second and the third terms include experiences of the particle and the swarm. c_1 and c_2 which pull the particle to local and global best positions, are constants. r_1 and r_2 are random numbers between 0 and 1. t is the iteration number. Among all possible position values of i'th particle, p_{iD} is the one which has the biggest fitness value. The biggest p_{iD} (i=1, 2, .., M) is denoted by g_i. K factor which is used for quaranteeing convergence performance of optimization, is shown in Equation 10 [8].

$$K = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|}, \qquad \varphi = c_1 + c_2 > 4 \tag{10}$$

Positions and speeds of particles must be limited for preventing the particles' leaving from search space. Speed value is limited according to determined maximum position value of particles and when the limits of particles are selected as more appropriate, better optimal results are obtained. These limits are shown in Equation 11 and Equation 12 [9].

$$\mathrm{v}^{\mathrm{maks.}} = \mathrm{k}.\mathrm{x}^{\mathrm{maks.}}, \qquad 0.1 \le \mathrm{k} \le 0.5 \tag{11}$$

$$\mathrm{v}^{\mathrm{min.}} = -\mathrm{v}^{\mathrm{maks.}} \tag{12}$$

If $\mathrm{v}^{\mathrm{k+1}}$ is bigger than $\mathrm{v}^{\mathrm{maks.}}$, $\mathrm{v}^{\mathrm{k+1}}$ will be taken equal to $\mathrm{v}^{\mathrm{maks.}}$ and if $\mathrm{v}^{\mathrm{k+1}}$ is smaller than $\mathrm{v}^{\mathrm{min.}}$, $\mathrm{v}^{\mathrm{k+1}}$ will be taken equal to $\mathrm{v}^{\mathrm{min.}}$. Also if $\mathrm{x}^{\mathrm{k+1}}$ is bigger than $\mathrm{x}^{\mathrm{maks.}}$, $\mathrm{x}^{\mathrm{k+1}}$ will be taken equal to $\mathrm{x}^{\mathrm{maks.}}$ and if $\mathrm{x}^{\mathrm{k+1}}$ is smaller than $\mathrm{x}^{\mathrm{min.}}$, $\mathrm{x}^{\mathrm{k+1}}$ will be taken equal to $\mathrm{x}^{\mathrm{min.}}$. Here k shows iteration number.

PSO starts with a random solution and optimal result is found with updates. After initial values are determined, the steps below are performed [13].

Step-1: Speed and position limits are determined for each unknown parameters.
Step-2: Initial speeds and positions of particles are appointed randomly in predetermined ranges.

Step-3: Fitness value of particles are measured according to fitness function and the position of particle which has the best fitness value between local best positions, is appointed to global best position and global fitness value.
Step-4: If fitness value of particle is bigger than its own local best fitness value, position and fitness value of particle is appointed to local best position and local best fitness value.
Step-5: If best fitness value of local best position vector is bigger than global best fitness value, position and fitness value of that particle is appointed to global best position and global best fitness value.
Step-6: Speeds and positions of particles are updated.
Step-7: This operation is repeated from Step-3 until iterations are finished.

5.1. PSO-PID Controller

The study principle of PSO-PID controller is showed in Figure 3. Unit step function is applied to input of the system. Error values are measured and they are recorded to a file. In PSO software these error values are loaded and fitness value of particles are calculated by using error values according to target function.

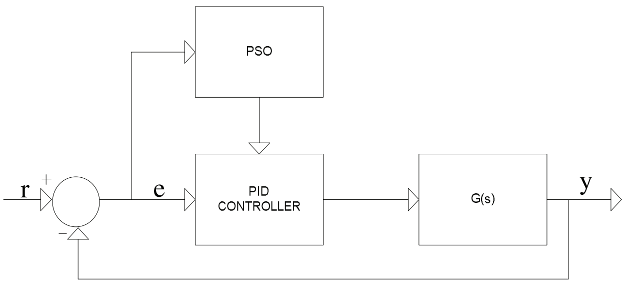


Figure 3. PSO-PID Controller

In this software c_1 and c_2 are 2.05, particle number is 20 and iteration number is 50. Rosenbrock function is used as target function in Equation 13 and fitness value is calculated in Equation 14 [10,11].

$$f(x) = \sum_{i=1}^{n}\left(100.(X_{i+1} - X_i^2)^2 + (X_i - 1)^2\right) \quad (13)$$

$$f = \frac{1}{f(x)} \quad (14)$$

Optimized PID parameters are given in Table 4 for different boiler loads.

Table 4. Optimized PID Parameters

Parameters	Under 25% boiler load	Under 65% boiler load	Under 100% boiler load
P	0.37933	0.4	0.3614
I	0.0013206	0.00078248	0.0005
D	18.729	30.546	36.7762

6. Results

For applying controllers to the system used block diagram is shown in Figure 4. During simulation PID, FLC and PSO-PID controllers are used instead of the controller which is shown in the block diagram. All simulations are performed by using Matlab Simulink software.

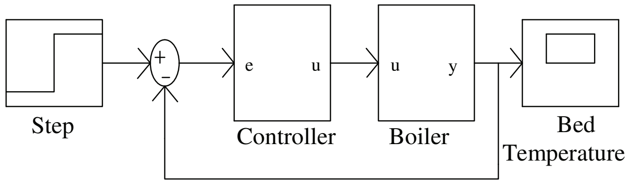
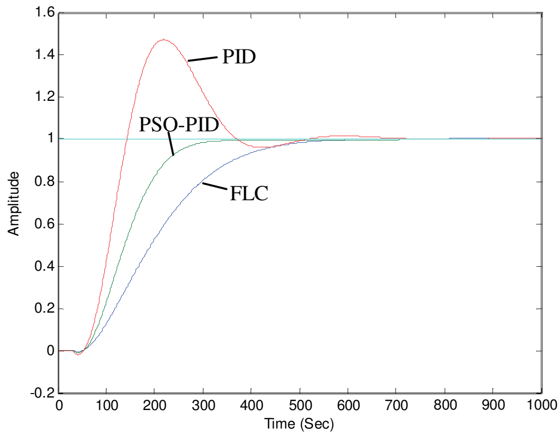
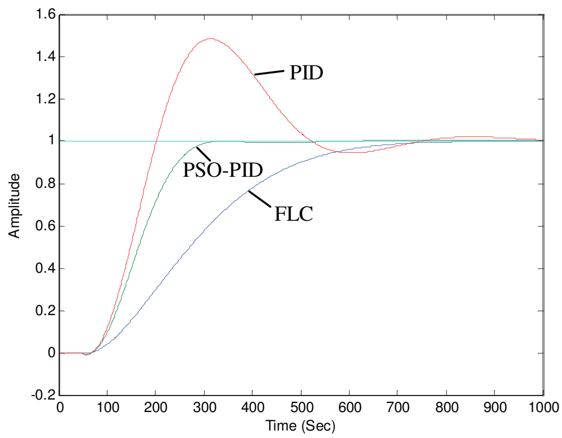


Figure 4. Block Diagram of Controlled System

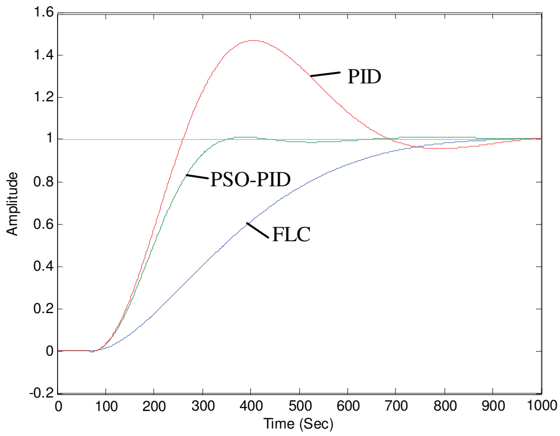
In simulation unit step function is used for desired value and simulation results for all controllers are given in Figure 5.



(a)



(b)



(c)

Figure 5. Comparison of Simulation Results
a) Under 25% Boiler Load b) Under 65% Boiler Load
c)Under 100% Boiler Load

Obtained results with all controllers for bed temperature control are given in Table 5.

Table 5. Comparison of Control Methods

Boiler Load	Parameters	PID	FLC	PSO-PID
25%	Max. Overshoot (%)	46.85	-	1
	Settling Time (sec)	600	600	340
65%	Max. Overshoot (%)	48	-	1
	Settling Time (sec)	860	800	325
100%	Max. Overshoot (%)	46.5	-	1.6
	Settling Time (sec)	1100	900	385

Simulation results and the values in Table 5 show that settling time in PSO-PID controller is lower than the other controllers. PSO-PID decreases overshoot like FLC but in classical PID controller overshoots are the biggest. In FLC there are no overshoot.

7. Conclusions

According to simulation results if overshoot isn't wanted, a FLC must be used and if short settling time is wanted, a PSO-PID controller must be used. PSO-PID has an uncomplicated algorithm and it provides an effective solution. Additionaly, PSO-PID doesn't need experience like FLC. If this theoric study is actually implemented, combustion efficiency will increase at the boiler and harmful emissions will decrease. Also equipments used in power plants will be able to work longer. So maintenance and revision costs will decrease.

8. References

[1] B. Lixia, Z. Junxia And F. Song, "Modeling and simulating of bed temperature control of circulating fluidized boiler", *Journal of North China Electric Power Univ.,* vol. 30, no. 1, pp. 53-56, Jan., 2003.

[2] P. Fu, X. Yu And H. Wang, "Research on fuzzy control algorithm for bed temperature control of circulating fluidized bed boiler", *Proceeding of the Forth International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 825-828.

[3] A. A. Jalali And A. Hadavand, "Bed temperature control of a circulating fluidized bed combustion system using H∞ algorithm", *International Conference on Control , Automation and Systems*, Seoul, 2007, pp. 2658-2662.

[4] J. Kennedy And R. Eberhart, "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[5] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-3, no. 1, pp. 28-44, Jan., 1973.

[6] N. Pillay And P. Govender, "A particle swarm optimization approach for model independent tuning of PID control loops", *The 8th IEEE Africon Conference*, Windhoek, 2007, pp. 1-7.

[7] Y. Shi And R. C. Eberhart, "A modified particle swarm optimizer", *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, 1998, pp. 69-73.

[8] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization", *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, USA, 1999, pp. 1951-1957.

[9] J. Chen, Z. Ren And X. Fan, "Particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters", *1 st International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, 2006, pp. 990-994.

[10] Y. Shi And R. C. Eberhart, "Emprical study of particle swarm optimization", *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, USA, 1999, pp. 1945-1950.

[11] Zwe-Lee Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system", *IEEE Trans. On Energy Conversion*, vol. 19, no. 2, pp. 384-391, June, 2004.

[12] J. G. Ziegler And N. B. Nichols, "Optimum settings for automatic controllers", *Trans. of the A.S.M.E.*, vol. 64, pp. 759-768, Nov., 1942.

[13] H. Aygun, "Akışkan yataklı buhar kazanının yatak sıcaklığının parçacık sürücü optimizasyonu tabanlı PID kontrolör (PSO-PID) ile kontrolü", M.S. thesis, Electrical and Electronics Eng., Karabuk Univ., Karabuk, Turkey, 2011.