

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,800

Open access books available

142,000

International authors and editors

180M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Using Genetic Algorithm to Optimize Controllers of Thermal Load System in Thermal Power Plant

Pham Thi Ly and Bui Quoc Khanh

Abstract

This chapter presents the sequence of implementing the genetic algorithms using the programming language in the Mfile application of MATLAB Simulink software to optimize the two controller parameters of the coordinated control system structure of the thermal load system in coal-fired thermal power plants: electric power controller and steam pressure controller. Optimal standards are determined to be fast-tracking and fuel-saving. Operational data at a thermal power plant in Vietnam have been used to simulate the operation of a thermal load control system with a coordinated control structure in a thermal power plant to test controller parameters found from the genetic solution. To clarify the superiority of the genetic algorithm method in control of the thermal load system of a thermal power plant, the authors give an evaluation of the original control system compared to the control system using the parameters found from the genetic algorithm method. The results show that the thermal load control system in the thermal power plant using controller parameters found from the genetic algorithm method is much more optimal in terms of fuel consumption and the ability to follow the set amount.

Keywords: genetic algorithm, coordinated control system, thermal power plant, thermal load control system, optimization

1. Introduction

Genetic algorithms are general algorithms that can successfully solve difficult problems in many fields, which cannot be solved by other methods [1–6]. The application scope of genetic algorithms is confronted with problems that are impossible or ineffective to solve.

In recent years, the applications of genetic algorithms have increased greatly in many fields such as engineering (engine design, aircraft design, etc.), optimization, robot operation, product classification system, machine learning system, pattern recognition, neural network training, fuzzy system tuning, planning, adaptive control, game programming, transportation problems, tourism problems, etc. [1–3].

The control system's design involves many issues, such as, system stability, transient and steady-state quality, etc. Each problem depends a lot on the structure and parameters of the control system. However, this dependence may not be expressed mathematically. In addition, when designing, it is necessary to ensure that conflicts among the quality criteria are adequately resolved. The lack of systematic methods for selecting values for many controller parameters is a major obstacle to satisfying control requirements. To solve these problems using genetic algorithms, we encode the structure and parameters of the controller into a chromosomal sequence and define a fitness function as a function of the quality requirements. Accordingly, we can convert the design problem into the problem of minimizing an objective function according to the controller parameters. Since genetic algorithms only use the fitness function in the optimization process, they can perform this search. The innovative combination of existing control methods with genetic algorithms can create a powerful tool to solve real control problems.

The field of control of coal-fired power plants is complicated and contains many difficult processes to solve, especially the thermal load control system of coal-fired power plants [7–9]. Currently, many authors have used genetic algorithms in their research to solve problems in production and control activities at coal-fired power plants. The studies [10–15] all have a thermal load system model described by a first-order differential equation or a system of nonlinear equations. These control models have ignored the physical effects of the boiler base loops in the thermal load system in the thermal power plant: the fuel control loop, the gas and air control loop, the water supply control loop, superheat steam, and spray control loop. The use of genetic algorithms in these studies is mainly using the GA toolbox in MATLAB Simulink with different goals: [10–12] interested in reducing fuel consumption in the production of each MW of electricity, fast-tracking the setpoint; [16] interested in minimizing the influence of turbine valve adjustment and combustion speed on the power generator; [13, 17] investigated all the objectives which are related to each parameter in the operation of thermal power plants: coal flow, smoke flow, ID fan, FD fan, heat reduction spray flow of the process of superheating and spray, etc. while [14] gives the target of the fastest action time for the control parameters of the fuel valve, steam valve, and water supply valve.

In principle, the parameters of a PID controller in a control system can be easily designed using classical methods such as root locus or bode plots, etc. However, in the industry, most parameters of the PID controller are experimentally adjusted by the errors test method because there are no mathematical models of the object. The adjustment process in many cases is very difficult and time-consuming. A proposed solution to solve the above-mentioned difficulties is to use a genetic algorithm to automatically adjust the parameters of the PID controller so that the control systems reach the minimum value of criterion.

In this chapter, a genetic algorithm is used to optimize the parameters of two main PID controllers in the thermal load control system with a new coordinated structure which is proposed by the authors (G_{CN} power controller and G_{CP} steam pressure controller). This new coordinated control structure built by the authors is based on a method that combines the description of dynamic processes with data collection and operating parameters of a real thermal power plant. This is a complicated model, including many linear and nonlinear processes with mutual and interleaving influence. The goal is used to find solutions for optimal parameters of power controller and pressure controller using genetic algorithms is to optimize the operation: fast-tracking the setpoint and saving the fuel (kg of coal/kWh). Because the model is so large and complicated that no equation can fully describe its characteristics, it is impossible to use the GA toolbox as the other authors do to apply genetic algorithms in finding the optimal parameters for the controllers of the

thermal load system. The present authors build a program on MATLAB Simulink's Mfile with all the properties and characteristics of a genetic algorithm. This program will find out the optimal parameters of the controllers in the thermal load system by affecting the model built on Simulink to find the optimal parameters for the controllers.

Achieve expected results:

- A software program will be written for a genetic algorithm is applied to find optimal parameters for the controllers of a model with a complicated structure.
- New optimal parameters are better for the controllers of the thermal load system in thermal power plants.
- The consumption of fuel is reduced during the production of electricity (at least 1 g coal per kW power).

2. New coordinated control structure

2.1 New coordinated control structure of thermal load control system in thermal power plant

The thermal load control system according to the new coordinated structure proposed by this chapter's authors in **Figure 1**:

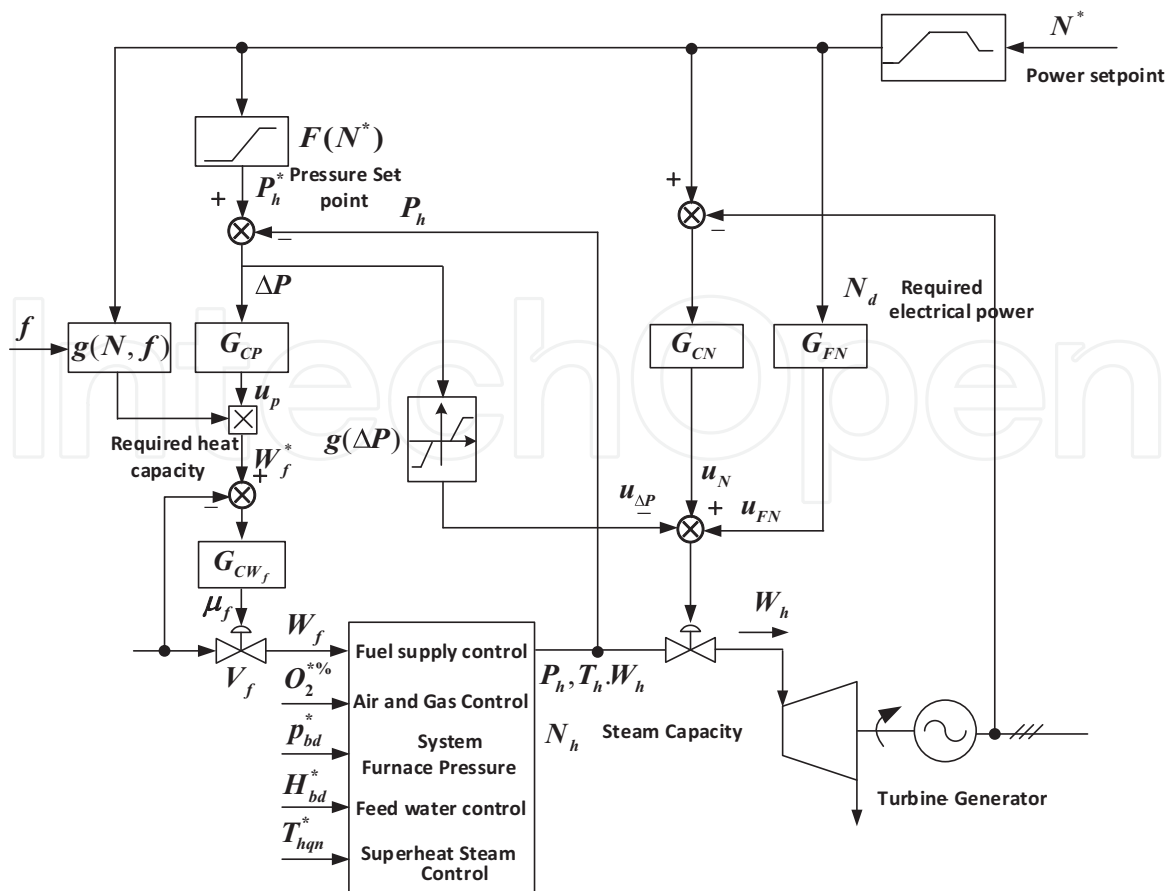


Figure 1.
 The new coordinated control structure.

On the new coordinated control structure, the setpoint of power N_e^* (MW) is used as the setpoint for both the steam power control system (boiler) and the generating capacity control system (turbine-generator). In the steam power control system, the variables to be controlled are steam temperature and steam pressure. To meet the control requirements of this control system, it is necessary to adjust the basic control loops so that these control loops work well, meeting the operating requirements: fuel control loop, gas and air control loop (control of residual oxygen concentration and control of negative pressure in the combustion), water supply control loop, superheated steam and spray control loop. A pressure regulating limiter is designed to be placed at the outlet of the G_{CP} pressure controller. This setting is to avoid the amount of fuel being too large or zero to cause fluctuations in the amount of fuel set.

On the power generation control system with G_{CN} power controller, the variable that needs to be controlled is the electrical power N_e (MW), and the actuation variable is the turbine regulating valve which is to bring the standard steam flow into the turbine to rotate the machine generate electricity. At the control system side of the turbine-generator cluster, it is designed to feedforward the power setpoint G_{FN} in the turbine control part to increase the ability to fast track the power setpoint. This structure has a signal that compensates for pressure acting on the turbine valve opening signal to eliminate the channeling between the steam capacity and electrical power control systems.

Because fuel often changes the chemical composition and calorific value of coal varies with the type of coal. The determination of variation is usually done by the chemistry lab at the plants that experiment and test these fuels and then set the operating requirements for that coal. In this new coordinated control mechanism, we proposed to add fuel characteristics $g(N, f)$ to actively change the fuel setpoint according to the change of the power setpoint corresponding to each coal with different calorific values, reducing adjustment time and contributing to fuel economy. The characteristic $G(N, f)$ function tested when changing the calorific value of coal affects electrical power and vapor pressure is shown in **Figure 2**.

Due to the large inertia of the boiler, the steam pressure control system responds more slowly to the turbine-generator output steam power requirements to meet the

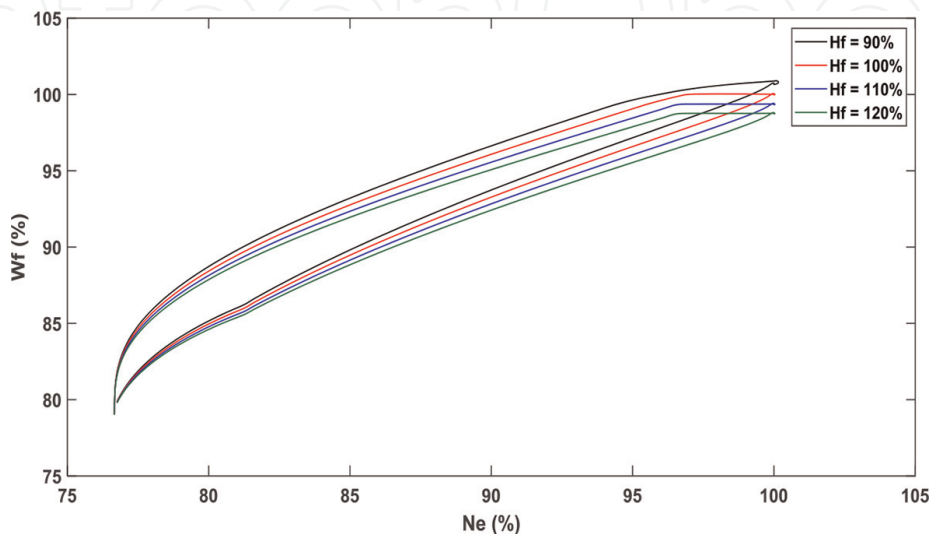


Figure 2.
The $G(N, f)$ characteristic of the new coordination control structure.

capacity electrical requirements. Thus, an inter-channel signal is required between the two vapor pressure control systems and the power control systems. Therefore, in the structure of **Figure 1**, we have introduced a static nonlinear function $g(\Delta P)$ [18] to compensate for the interleaving between the two steam pressure control systems G_{CP} and the power control system G_{CN} to ensure that the pressure is slightly more stable and the transmitter power responses faster. This static nonlinear compensation function $g(\Delta P)$ is developed from the structure of Flynn [18].

2.2 Simulation of the operation of a thermal load control system with a coordinated control structure

MATLAB Simulink software is used to simulate the control operation of the coordinated control structure in **Figure 1**.

2.2.1 Simulation parameters

Simulation parameters are taken at the stable working time (load from 230 to 300 MW) of a unit with a capacity of 300 MW of the Hai Phong thermal power plant [7, 8].

The volume of the furnace: 8485 m^3 .

- Temperature of smoke exiting from the furnace: 1047°C .
- Smoke fan speed: 620 RPM.
- Total flow of air supplied to the boiler: 295.486 kg/s.
- Specific heat capacity of wind: 1005 J/kg.K.
- Coal flow rate: 28.3 kg/s.
- Specific heat of fuel: 1260 J/kg.K.
- Smoke/wind ratio: 1.1.
- Ratio of fuel inlet and slag out: 0.2.
- Coal and wind temperature level 1: 228°C .
- Wind temperature level 2: 341°C .
- Specific heat capacity of water: 4200 J/kg.K.
- Water supply pump speed: 4842 RPM.
- Saturated steam specific heat: 1840 J/kg.K.
- Main steam flow: 191 kg/s.
- Saturated steam pressure: 14.2 MPa.

- Saturated steam temperature: 340°C.
- Main steam temperature: 541°C.
- Water supply flow: 176 kg/s.
- Feedwater temperature: 280°C.
- Water flow to reduce temperature: 18 kg/s.
- Water spray valve opening to reduce heat: 29.42%.
- Water temperature reduced: 25°C.
- Valve opening: 75%.

2.2.2 Simulation model

After building a simulation model of the control loops in the control systems: the boiler control system and the turbine-generator combination control system. The new coordinated control structure of the thermal load control system is shown in **Figure 1**, using experimental data experience and operation collected from Hai Phong thermal power plant [7] to build a simulation model on MATLAB Simulink in **Figure 3**.

Figure 4 shows the coordinated control structure in the thermal load control system:

2.2.3 Simulation results of the control system without changing the coal calorific value

Simulation scenario: The system is operating stably at the balanced point at the generating capacity of 230 MW. There is a request to increase capacity from 230 to 300 MW at 1000 s. At 9000 s, there is a request to reduce the generating capacity from 300 to 230 MW. Load increase and decrease rate is 3 MW/min, steam pressure is set according to sliding pressure characteristic, at 1000 s start to increase from 14.2 to 16.7 MPa (68 ÷ 80%, respectively) and at the time of 9000 s start to decrease from 16.7 to 14.2 MPa.

Figure 5 illustrates the responses of power (N_e), steam pressure (Ph), steam flow (W_h), coal fuel flow (W_f), boiler water level (H), superheated steam temperature

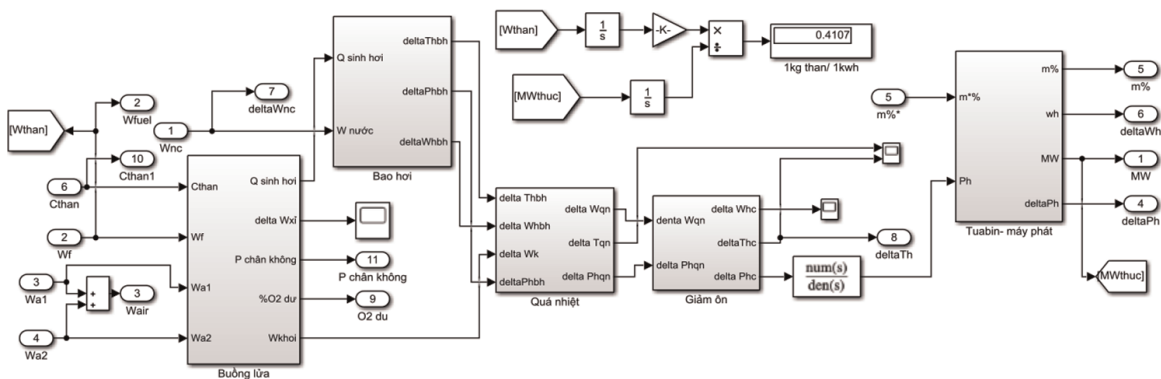


Figure 3.
The model of a thermal load control system in MATLAB Simulink.

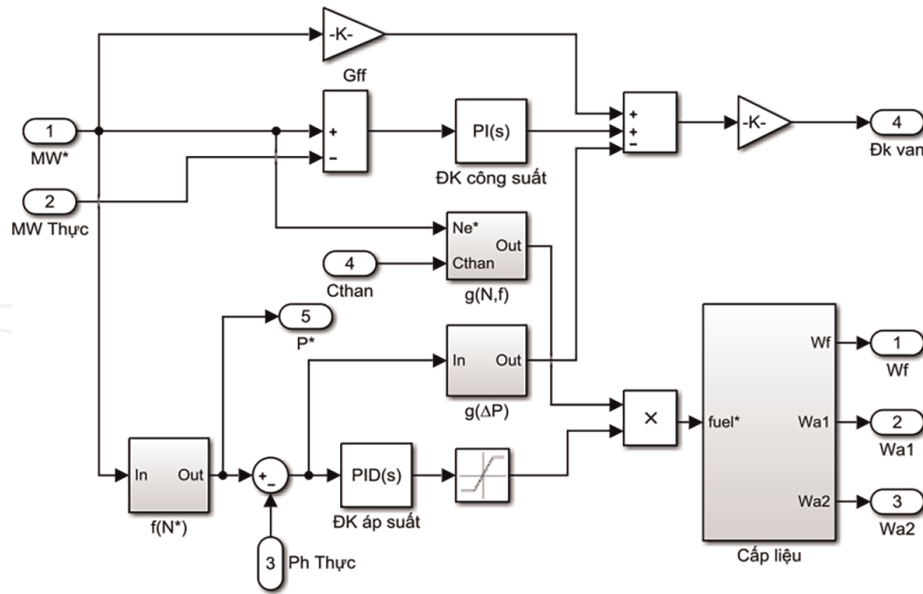


Figure 4.
 Coordinated control structure of thermal load system in MATLAB Simulink.

(T_h), combustion chamber negative pressure (P_{bd}), residual $\%O_2$ concentration (residual O_2) when no change is simulated in the calorific value of coal.

The criteria to evaluate the power and pressure response in **Figure 5** are transient time (T_{qd}), power error, maximum pressure difference (e_{max} (%)) during load increase, decrease time, and static error. We have the following quality rating as shown in **Table 1**.

Thus, it shows that: the power and pressure responses have a fast transition time during both load increase and decrease. The error in the process of increasing and

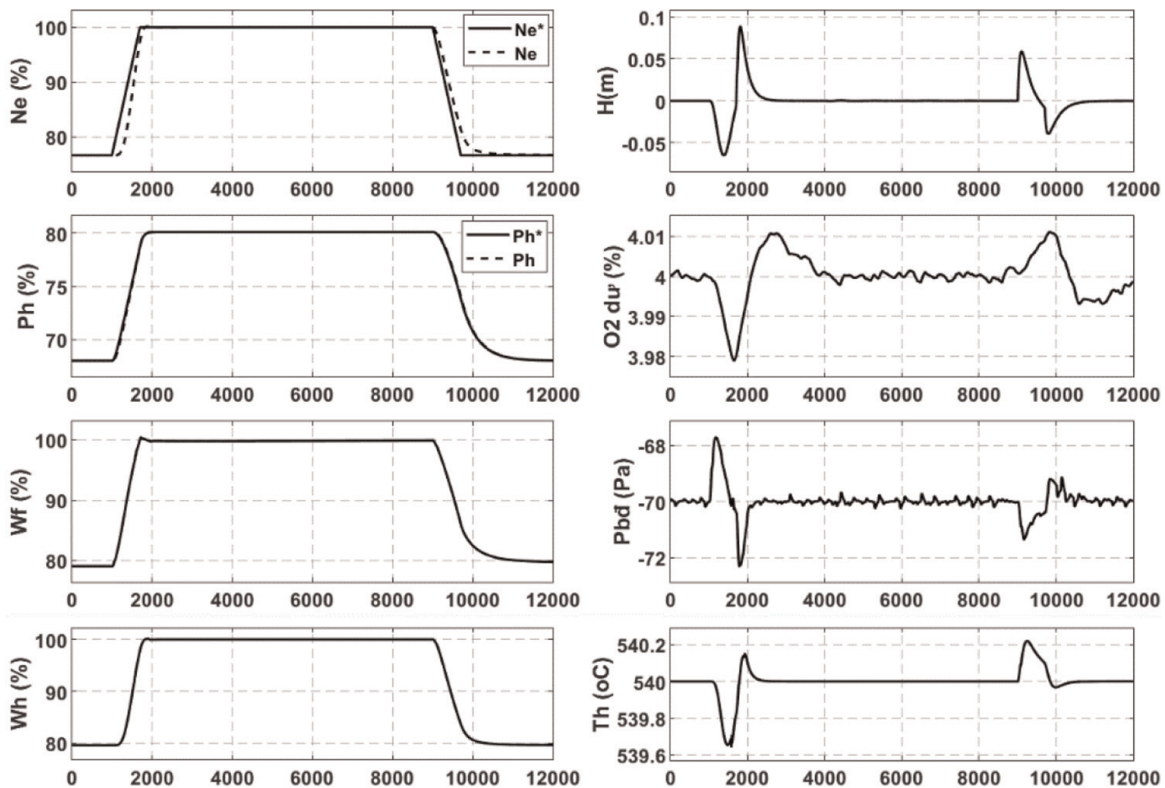


Figure 5.
 Simulated application response mode control combinations.

Control loops	T_{qd} (s)		e_{max} (%)	
	Increase load	Reduce load	Increase load	Reduce load
Power response	822	1155	8.06	4.94
Pressure response	812	2250	0.392	0.18

Table 1.
Power and pressure response control quality assessment.

decreasing the load is small. There is no over-adjustment or oscillation. The real power line (N_e) follows the setpoint power line (N_e^*), the real pressure line (Ph) follows the setpoint pressure line (Ph^*).

The characteristics of fuel flow and steam flow have been met according to control requirements. The four basic control loops in the boiler were working true. The oscillations of the responses around the preset point are within the allowable range. The level of water in the boiler fluctuates in the range of $-65 \div 88$ (mm), the concentration of residual $\%O_2$ fluctuates in the range of $-0.02 \div 0.01$ (%), the negative pressure of the combustion chamber fluctuates in the range of $-67.7 \div 72.3$ (Pa), superheated steam temperature ranges $-0.34 \div 0.22$ ($^{\circ}C$).

In summary, the four control loops in the boiler control are working true. The coordinated control structure has also shown good quality when ensuring fast response of pressure control and power control to load requirements, eliminating the inter-channel effect between boiler and turbine-generator.

2.2.4 Simulation results of the control system when the coal calorific value changed

Figure 6 shows the responses of the calorific value of coal (H_{than}), fuel flow control loop (W_f), power error ($ErrN_e$), pressure error ($ErrPh$), residual $\%O_2$ concentration (residual O_2) control loop, and the combustion chamber negative pressure control loop (Pbd).

Coal calorific value is changed when the system is stable at 100% load capacity. At the time 4000 s start to change the calorific value as shown in **Figure 6**. There are two stages of heating value increase: the heating phase increases from 100% up to 109.5% compared to the average calorific value, and the period of reducing the calorific value to 95.2% compared with the average calorific value.

It can be seen that the fuel response changes immediately as soon as the calorific value of the coal changes. As the calorific value of coal increases, the fuel flow is reduced by 1.15% of the maximum required fuel flow. As the calorific value of coal decreases, the fuel flow increases to 0.66% of the maximum fuel flow.

Power and pressure errors are small when the coal calorific value changes. This error value is so small that it can be ignored. The response of the control loops the residual $\%O_2$ concentration, and the combustion chamber negative pressure also fluctuates slightly around the preset value. It shows that these control loops operate stably, not affected by the interference of temperature change.

Summarizing this section, the combined control structure gives good control quality and works well when there is a change in coal calorific value, which affects the new coordinated control system proposed by the authors. It has eliminated the inter-channel effect between the boiler and the turbine generator.

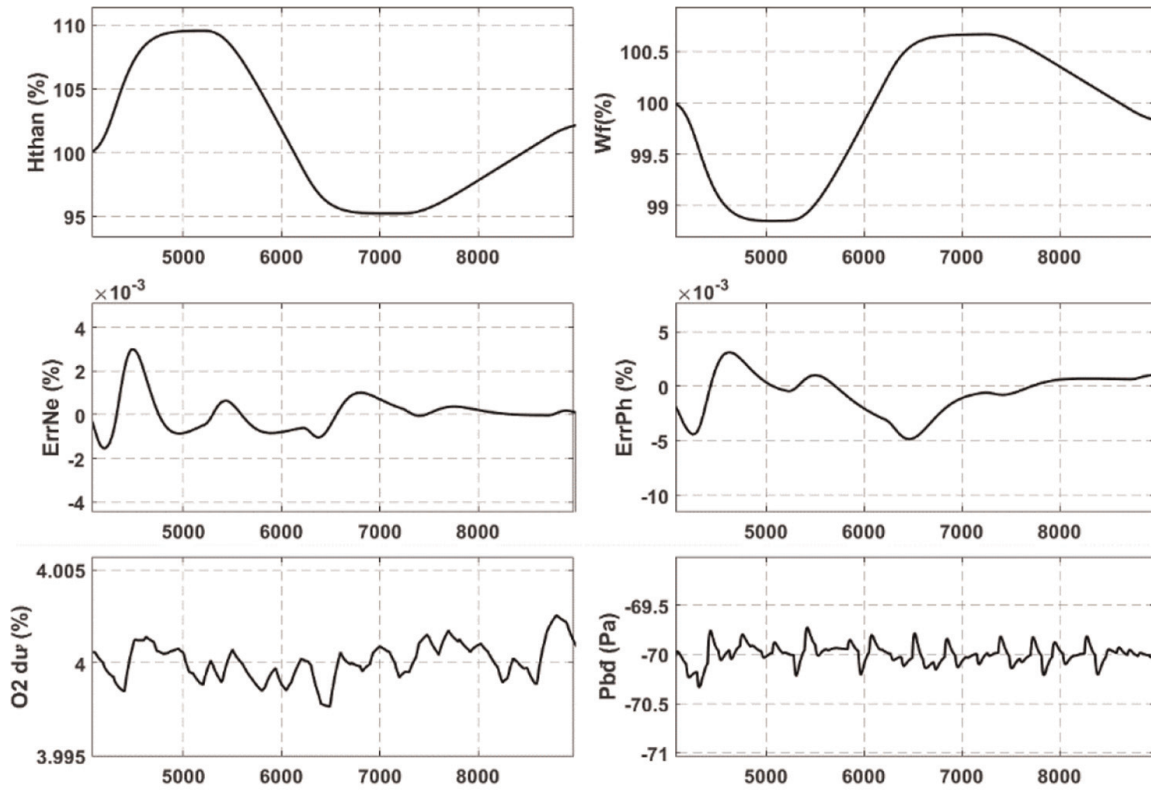


Figure 6.
 Simulation response when the coal calorific value changed.

3. Using genetic algorithm to optimize parameters of two controllers of the new coordinated control structure

Although the pressure controller G_{CP} and the power controller G_{CN} have met the control requirements for the thermal load control system of the thermal power plant. However, those controllers only use the IMC-PID method and PID self-tuner tool on MATLAB Simulink. The parameters of the pressure controller and power controller in the new coordinated control structure of the thermal load control system are not the most optimal. Therefore, to improve production efficiency and optimize operation (fast-tracking the setpoints and saving fuel), the chapter's authors continue to research and design a software program using genetic algorithms to find out the optimized parameters of PI and PID controllers of steam pressure control system G_{CP} and power control system G_{CN} .

For the objective of power tracking control, absolute value integral standard is applied with minimal power error as shown in Eq. (1):

$$J_N = \frac{1}{t - t_0} \int_{t_0}^t |e_N^{\%}| dt \rightarrow \min \quad (1)$$

For the target of reducing fuel costs for electricity production (kg coal/kWh), it is calculated as in Eq. (2):

$$J_f = \frac{\int_{t_0}^t W_f(t) dt}{\int_{t_0}^t N_e(t) dt} \rightarrow \min \quad (2)$$

The objective function of refining G_{CN} and G_{CP} controllers using genetic algorithms in this contents are:

$$J = J_N + J_f \quad (3)$$

The purpose of the applied GA algorithm is to find the optimal values $\{Kp1_out, KI1_out, Kp2_out, KI2_out, Kd2_out\}$ of the controller G_{CN} and G_{CP} of the new coordinate system, where the functions J_N and J_f reaches the minimum value. In other words, the objective function of the GA algorithm is $\min (J)$.

To limit the search space of the genetic algorithm, it is assumed that the optimal values $\{Kp1_out, KI1_out, Kp2_out, KI2_out, Kd2_out\}$ are located around the value $\{Kp1, Ki1, Kp2, Ki2, Kd2\}$ of two controllers G_{CN} and G_{CP} . We know that these parameters $\{Kp1, Ki1, Kp2, Ki2, Kd2\}$ are obtained from using the PID Turner tool of MATLAB Simulink and the IMC_PID method. The search limits for those five parameters of the PID controller are as shown in Eq. (4) as follows:

$$\begin{aligned} \alpha K_{p1} &\leq K_{p1-out} \leq \beta K_{p1} \\ \alpha K_{i1} &\leq K_{i1-out} \leq \beta K_{i1} \\ \alpha K_{p2} &\leq K_{p2-out} \leq \beta K_{p2} \\ \alpha K_{i2} &\leq K_{i2-out} \leq \beta K_{i2} \\ \alpha K_{d2} &\leq K_{d2-out} \leq \beta K_{d2} \end{aligned} \quad (4)$$

In which, the coefficients α, β are chosen so that the search space is large enough to accommodate the desired optimal value. Simulation results on the thermal load system with the new coordinated control structure, $\alpha = 0$ and $\beta = 300$ are satisfied.

The genetic algorithm is used as a tool to solve the optimization problem, to obtain the values $\{Kp1_out, KI1_out, Kp2_out, KI2_out, Kd2_out\}$ satisfying the objective functions on Eq. (3) with space search is limited by Eq. (4). Algorithm flowchart of genetic algorithm to determine PID parameters of two G_{CN} power controllers and G_{CP} pressure controllers in the new coordinated control system in **Figure 7**:

The steps to search for optimal parameters for two controllers G_{CP} and G_{CN} according to the algorithm flowchart of the genetic algorithm method in **Figure 7** are as follows.

3.1 Startup

Individuals in the origin natural population $Kp1, Ki1, Kp2, Ki2, Kd2$ of the two controllers G_{CP} and G_{CN} are randomly startup so that at the time of initialization they can exist for a long time in the environment. The initial parameters for the startup process are as follows:

- Number of individuals in the initial population: 25

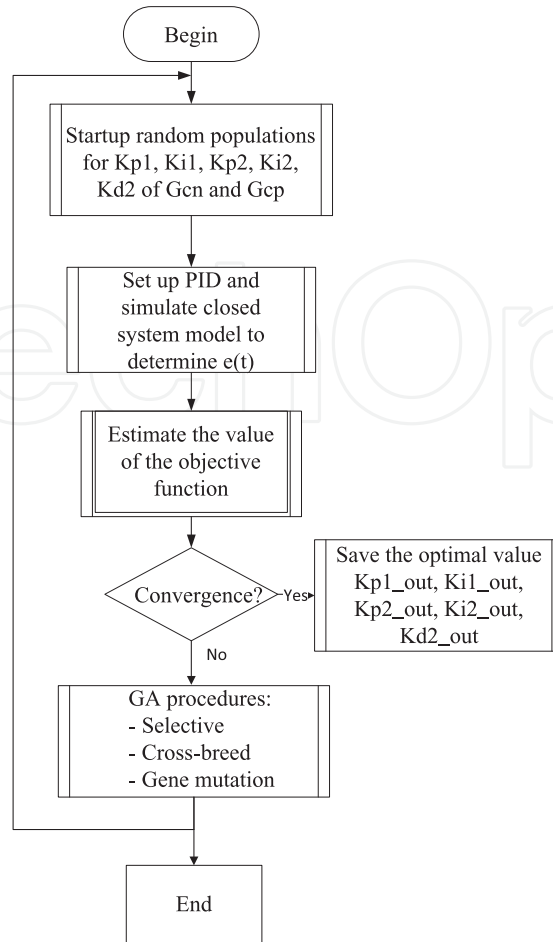


Figure 7. Process flow chart of genetic algorithm determining PID parameters of two power and pressure controllers in a new coordinated control system.

- Each individual has five chromosomes. Each of these individuals is equivalent to a power controller parameter (Kp1, Ki1) and pressure controller parameters (Kp2, Ki2, Kd2) with a limit of parameters Kp1, Ki1, Kp2, Ki2 from 0 to 10 and Kd2 parameter from 0 to 300. The subroutine of the startup process is as follows:

```

function par = Init(N,d,range)
% input variables:
% N: population size
% d: Number of parameters of the function to find the extreme
% range: two-dimensional array [2xd] stores the value domain of the parameters
% output variables:
% par: two-dimensional array [Nxd]store randomly initialized population
for pop_index = 1:N
    for par_index = 1:d,
        par(pop_index,par_index) = ...
            (rand-0.5)*(range(2,par_index)-
range(1,par_index))+
            0.5*(range(2,par_index) + range(1,par_index));
    end
end
end
  
```

3.2 Evaluate

To evaluate individuals, a fitness function must be defined. A fitness function is usually a function that needs to find an extreme or an equivalent transformation of the function. The fitness function is calculated from the objective function (Eq. (5)) as follows:

$$J = (Ne^* - Ne)^2 + (Ph^* - Ph)^2 + \left(\frac{w_f}{u_T}\right)^2 \rightarrow \min \quad (5)$$

The fitness function is the inverse of the objective function (J):

$$fitness = \frac{1}{J + C} \quad (6)$$

Where C is added to ensure that the fitness function is always positive.

To find the solution of the objective function, we take the parameters Kp1, Ki1, Kp2, Ki2, Kd2 of the two controllers G_{CP} and G_{CN} through the initial start-up process, and put them into the simulation on Simulink, which has built this condition control of the new coordinated control structure. If the PID parameter satisfies the stable system, then GA calculates J. Otherwise, if the system is unstable, stop the simulation and find another PID parameter. Because we use the order of fitness of chromosomes ascending (GA is looking for the max value), we must inverse the objective function to find the max value (if J finds the min). Then the best instance (with the smallest J) is stored in the bestchrom memory cell. The program implemented for the evaluation process is as follows:

```

% Calculate the fitness of the initial population
for pop_index = 1:N,
    Kp1 = par(pop_index,1);
    Ki1 = par(pop_index,2);
    Kp2 = par(pop_index,3);
    Ki2 = par(pop_index,4);
    Kd2 = par(pop_index,5);
    sim('newcoordinated.slx');% Simulation of the
                            % coordinated control
                            % system in figures 3 and 4
    if length(e1) > 79500
        Kp1;
        Ki1;
        Kp2;
        Ki2;
        Kd2;
        J = (e1'*e1) + (e2'*e2) + (e3'*e3);
        fitness(pop_index) = 1/(J + eps);
    else
        J = 10^100;
        fitness(pop_index) = 1/(J + eps); % GA to find the
                                        % maximum of the

```

```
                                % adaptive function
end
end;
[bestfit,bestchrom] = max(fitness)
```

Thus, after the startup and evaluation of the genetic algorithm, 25 individuals were initially randomly selected to exist in the habitat. These individuals are taken to the next evolutionary process.

3.3 Encode

To apply the genetic algorithm to the optimization problem, it is necessary to encode the solution of the problem into a chromosome sequence. In this problem, we need to find five parameters Kp1, Ki1, Kp2, Ki2, and Kd2. Each solution consisting of five parameters is called an individual containing a sequence of chromosomes. Each parameter is a gene segment on the chromosome. There are three ways commonly used to encode: binary, decimal, and real number encoding. In this problem, the encoding is a decimal encoding.

The genome used to encode the solution in this decimal encoding method consists of 10 symbols {0,1,2,3,4,5,6,7,8,9}. This decimal encoding has the advantage that the NST string length is significantly shortened compared to the binary encoding, thus making the algorithm run faster. It is possible to directly apply the familiar genetic operations given to binary encoding. The implementation program for the decimal encoding process is as follows:

```
% Endcode_Decimal.m: Decimal encoding subroutine
% Programmer: Pham Thi Lý, UTC, HaNoi, VietNam
function pop = Encode_Decimal(par,sig,dec)
% input variables:
% par: two-dimensional array [Nx d] stores the parameters to be encoded
% N: Population size, d is the number of parameters
% sig: row vector [1x d] stores the number of significant digits equivalent to
each parameter
% dec: row vector [1x d] saves decimal point position
% output variables:
% pop: two-dimensional array [Nx L] save population after decimal encoding
% each row of pop is an instance (L is the length of the chromosomal string)
function pop = Encode_Decimal(par,sig,dec)
if (nargin < 3)
    error(['missing arguments. syntax: pop = Encode_Decimal(par,sig,dec)']);
end;
if size(sig) ~ =size(dec),
    error(['The sig and dec arguments don't match']);
end;
[N,d] = size(par); % Determine population size
for pop_index = 1:N, % Scan from the beginning to the end of the
```

```

                                % population
gene_index = 1;
for par_index = 1:d, % scan parameters
% The first gene encodes the sign of the parameter
if par(pop_index,par_index) < 0,
    pop(pop_index,gene_index) = 0;
else
    pop(pop_index,gene_index) = 9;
end
    gene_index = gene_index+1;
% Subsequent Genes are significant digits temp(par_index) = abs(par
(pop_index,par_index))/10^dec(par_index);
for count = 1:sig(par_index),
    temp(par_index) = temp(par_index)*10;
    pop(pop_index,gene_index) = temp(par_index)-
        rem(temp(par_index),1);
    temp(par_index) = temp(par_index)-
        pop(pop_index,gene_index);
    gene_index = gene_index+1;
end
end
end

```

3.4 Selective

The basic principle of selection is that the more adaptive the chromosome, the greater the probability of selection. The selection not only determines which individuals are allowed to exist but also determines the number of possible offspring.

The best individual has a higher probability of selection than the less fit individual. This selection is so strong that the genes of the highly-adapted individual can prevail. In the opposite case, this selection also occurs, the genes of the low-adapted individual will be suppressed or low-dominance. This causes a local solution or premature convergence. If the selection is weaker, the less well-adapted individuals have a chance to reproduce. This will increase the probability of finding a global solution but slow down the evolution.

There are many selection methods such as proportional selection, round selection, linear ranking selection, and exponential ranking selection. In the content of this chapter, we use linear ranking selectivity. Linear rank selection is to arrange individuals in ascending order of fitness and assign the best individual class N , the worst individual class 1. The probability of selection of each individual is linearly proportional to its rank:

$$p_k = \frac{1}{N} \left[\eta + 2(1 - \eta) \frac{k - 1}{N - 1} \right] \quad (7)$$

where $0 < \eta < 1$ the selection probabilities of the best and worst individuals are η/N and $(2 - \eta)/N$. The program of the linear ranking selection process is as follows:

```
function parent = Select_Linear_Ranking(pop,fitness,eta,elitism,bestchrom)
% Linear selection
% input variables:
% pop: two-dimensional array [NxL], population before selection
% N is the population size, L is the length of the chromosomal chain
% fitness: column vector [Nx1], the fitness of individuals in a population
% elitism: The flag holds the superior individual in the process of evolution
% bestchrom: variable used to save the best individual in evolution
% eta: Parameters of linear rank selection
% Biên ra:
% Parent: two-dimensional array [NxL], population after selection
if (nargin < 5), error('missing arguments');
end;
N = length(fitness); % population size
[fitness,order] = sort(fitness); % Arrange chromosomes in
    % ascending order of
    % fitness
% Calculate the probability of selection of chromosomes after sorting
according to formula (7)
for k = 1:N,
    p(k) = (eta+(2-eta)*(k-1)/(N-1))/N;
end
s = zeros(1,N + 1);
for k = 1:N,
    s(k + 1) = s(k) + p(k);
end;
for k = 1:N,
    if elitism == 1 & order(k) == bestchrom,
% If elitism = 1 and the chromosome in use is the best chromosome, then that
chromosome is selected
        parent(order(k),:) = pop(order(k),:);
    else
% if elitism = 0 or the chromosome in use is not the best one
% then that chromosome will be selected with a probability
% proportional to its rank
        r = rand*s(N + 1);
        index = find(s < r);j = index(end);
        parent(order(k),:) = pop(order(j),:);
    end
end
```

3.5 Crossbreed

The more adaptive an individual is, the more likely it is to survive. Through the process of crossbreeding, the good traits of the previous generation are passed on to the next generation. Hybridization is a method of sharing information between chromosomes. This operation combines the characteristics of two parental chromosomes to produce two offspring with the prospect that a good parent will produce better

offspring. Inbreeding usually does not affect all chromosomes but on the contrary, only occurs between two-parent chromosomes selected at random with a probability of hybridization. The principle of crossbreeding is to randomly pair two chromosomes in a population after having passed the selection step to create two daughter chromosomes, each child chromosome inherits a part of the parent's genes.

There are many different breeding methods such as single point crossbreeding, multiple point crossbreeding, and regular crossbreeding. Two-point crossbreeding means dividing each parent's chromosome into four random parts. When creating two son chromosomes, each son chromosome contains four parts including two parts of the father's chromosome and two parts of the mother's chromosome. The program of the breeding process is as follows:

```
function child = Cross_Twopoint(parent,Pc,elitism,bestchrom)
% Input Variables:
% parent: Two-dimensional array [NxL], population before
% hybridization
% N is the population size, L is chromosome length
% Pc: Probability of hybridization
% elitism: Flag that holds the superior individual in evolution
% bestchrom: The variable used to save the best individual in
% evolution
% Output Variables:
% child: two-dimensional array [NxL], population after
% hybridization
if(nargin < 4),
    error(['have not enough arguments']);
end;
[N,L] = size(parent);
for p1 = 1:N, % parent individual 1
% if elitism = 1 and p1 is a superior individual, do not cross
% (to preserve the best genes)
    if (elitism==1)&(p1==bestchrom)
        child(p1,:) = parent(p1,:);
    else
        if Pc > rand % hybridization occurs with
            % probability Pc
                p2 = p1; % Randomly select individual
                % parent 2 that is different
                % from individual parent 1
                while p2==p1,
                    p2 = rand*N;
                    p2 = p2-rem(p2,1) + 1;
                end
                k1 = rand*(L-1); % random selection of
                % hybrid point 1
                k1 = k1-rem(k1,1) + 1;
                k2 = k1;
                while k2==k1,
                    % randomly select
```

```
                                % hybrid point 2
                                % defferent from
                                % hybrid point 1
                                k2 = rand*(L-1);
                                k2 = k2-rem(k2,1) + 1;
end;
if k1 > k2, t = k2;k2 = k1;k1 = t; end;
% if k2 < k1 then convert k1 to k2 to ensure that k1 < k2
% Children inherit genes from their parents
    child(p1,1:k1) = parent(p1,1:k1);
    child(p1,k1 + 1:k2) = parent(p2,k1 + 1:k2);
    child(p1,k2 + 1:L) = parent(p1,k2 + 1:L);
else
    child(p1,:) = parent(p1,:);
end
end
end
```

3.6 Gene mutation

The offspring born through the process of natural selection and crossbreeding will carry the good qualities of the parent's generation. However, because the initial initialization generation is not rich and not suitable, the individuals cannot evenly spread the entire solution space. This makes it difficult to find the optimal solution. The mutation operation changes one or more genes of an individual to increase the diversity of the population structure. This helps to prevent the genetic algorithm from prematurely converging and the locally optimal solution. However, mutations should occur with low probability, otherwise, they can cause disturbance and loss of selected and highly adaptive individuals.

Mutation has many methods such as single point mutation, two-point mutation, and regular mutation. In this problem used uniform mutation. The program of the mutation process is as follows:

```
function newpop = Mutate_Uniform(pop,Pm,elitism,bestchrom)
% Input Variables:
% pop: Two-dimensional array [NxL], population before mutation
% N is population size, L is chromosome length
% Pm: Probability of mutation
% elitism: Flag used to preserve the superior individual in the process of
evolution
% bestchrom: The variable used to save the best instance in evolution
% Output variable:
% newpop: two-dimensional array [NxL], population after mutation
if(nargin<4),
    error(['missing arguments']);
end;
[N,L] = size(pop);
```

```

newpop = pop;
for pop_index = 1:N,
if(elitism==0)|| (elitism==1&& pop_index~ = bestchrom),
for gene_index = 1:L,
    if Pm > rand,% The mutation occurs with the
        % probability that Pm generates
        % a random gene different from
        % the gene under consideration
        rand_gene = rand*10;
        while(pop(pop_index,gene_index)==
            rand_gene-rem(rand_gene,1)||rand_gene==10),
            rand_gene = rand*10;
        end
        newpop(pop_index,gene_index)=
            rand_gene-rem(rand_gene,1);
    end
end
end
end
end

```

3.7 Convergence

If the preset number of generations is reached, stop breeding new generations and output an individual with the best chromosome sequence stored in the bestchrom memory cell. In this content, it is necessary to create five chromosomes, Kp1, Ki1, Kp2, Ki2, and Kd2. In case the previous generation objective function value is equal to the next generation objective function value in a preset number of generations, then also stop breeding and output the individual with the best chromosome sequence. The program of the convergence process is as follows:

```

% generation: variable that counts the number of generations
% terminal = 1: flag indicating the end of this algorithm
% stall_generation: variable that counts the number of generations of adap-
tive functions
% check stop condition
if generation == max_generation
    Terminal = 1;
elseif generation >1,
    if abs(bestfit(generation)-bestfit(generation-1)) < epsilon,
        stall_generation = stall_generation+1;
    if stall_generation == max_stall_generation, Terminal =1;
    end
else
    stall_generation =0; % variable count the
                        % number of generations

```

% of fitness function
% unchanged

end;
end;
end;

4. The results of running the genetic algorithm to find the optimal solution

4.1 Parameters of the genetic algorithm

- The number of individuals in the initial population is 25.
- The maximum number of generations is 30.
- The number of generations with the same objective function result is 20.
- The number of chromosomes contained in each individual is 5 corresponding to two power controller parameters (G_{CN}) and using PI structure KP1, KI1, and three pressure controller parameters (G_{CP}) using PID structure are KP2, KI2, KD2.
- Initialization limit for parameters KP1, KI1, KP2, KI2, KD2 from 0 to 300.
- The objective function (J) includes the optimal parameters of the squared deviation of the signal of the setpoint transmit power and the actual transmitted power. Plus, the coal saving criterion is the squared ratio of the fuel flow to the actual generating capacity.

$$J = (Ne^* - Ne)^2 + \left(\frac{w_f}{Ne}\right)^2 \rightarrow \min \quad (8)$$

- Probability of hybridization: 90%.
- Probability of mutation: 10%.

4.2 Results of the genetic algorithm after 30 generations of searching

The program that follows the algorithm of the genetic algorithm on MATLAB Simulink has been presented in Section 3. The value of the objective function in 30 generations is shown in **Figure 8**.

The value of the objective function reduce from 91.226 to 7911 after 30 generations, and the parameters of the two power and pressure controllers are found as follows:

- The G_{CN} power controller's parameter is $K_{p1_out} = 4.3$; $K_{i1_out} = 0.017$
- The G_{CP} pressure controller parameter is $K_{p2_out} = 41$; $K_{i2_out} = 0.05$; $K_{d2_out} = 169$

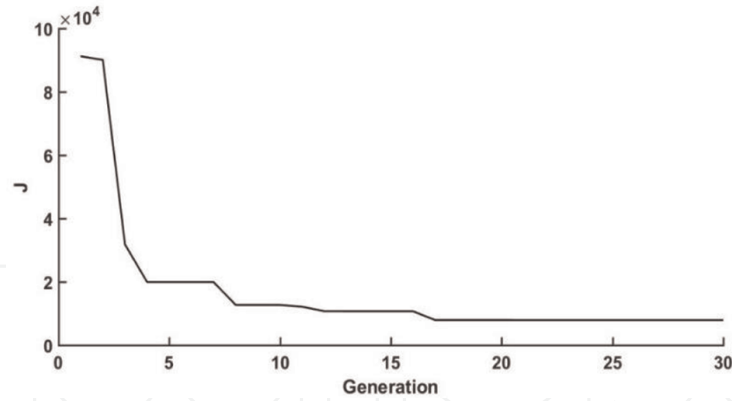


Figure 8.
Target J function decreases over 30 generations of the genetic algorithm.

While the old parameters of the controllers are as follows:

- The G_{CN} power controller's parameter is $K_{p1} = 0.093$; $K_{i1} = 0.0053$
- The G_{CP} pressure controller parameter is $K_{p2} = 1.18$; $K_{i2} = 0.085$; $K_{d2} = 98.1$

We use new parameters that are found by genetic algorithm for two controllers G_{CN} and G_{CP} into the thermal load control system model with a new coordinated control structure on MATLAB Simulink. Then evaluate and compare the model with these new parameters with the model when the controllers have not been optimized parameters by genetic algorithm. The results are made as in Section 5 of this chapter.

5. Simulation and evaluation

To demonstrate that the parameters of the steam pressure controllers G_{CP} and power controller G_{CN} are optimal. In this content, the chapter's authors study, test, and compare a new coordinated control structure that has parameters optimized by genetic algorithms with a new coordinated control structure using IMC_PID and PID self-tuner methods on MATLAB Simulink to find controller parameters. The simulation scenario is kept the same as when simulating the new coordinated control structure designed in Section 2 of this chapter and considering the case of the working system without interference. **Figure 9** is the simulation result of the combined control mode when using GA, including power response N_e (%), pressure response P_h (%), fuel flow response W_f (%), meet steam flow rate W_h (%), response to steam envelope water level H (m), respond to residual O_2 concentration (%), respond to combustion chamber pressure P_{bd} (Pa), respond to superheated steam temperature Th ($^{\circ}C$).

5.1 Simulation results

The simulation scenario here is kept the same as when the simulation for the new coordinated control structure is designed in Section 2 of this chapter and considers the case of the working system without interference. **Figure 9** shows the simulation results of the combined control system when using GA, including power response N_e (%), pressure response P_h (%), fuel flow response W_f (%), response steam flow

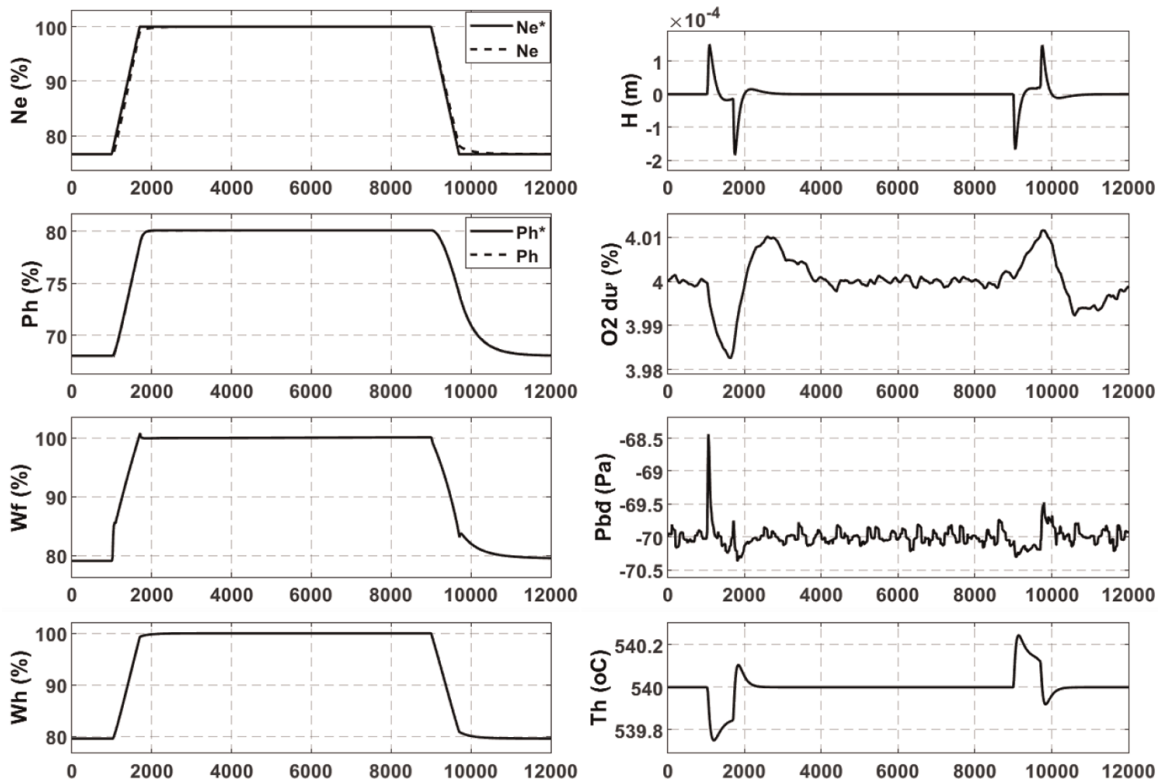


Figure 9.
 Simulation responses of control mode using GA.

response W_h (%), response to steam envelope water level H (m), response to residual O_2 concentration (%), response to combustion chamber pressure Pbd (Pa), response to superheated steam temperature Th ($^{\circ}C$).

We found that when reconfiguring the controllers according to GA, the responses of the four control loops in the boiler still ensure quality and stability.

We compare the response of the system using GA with the new coordinated control model designed by us and not using GA. Evaluation of the standard of transient time (T_{qd}), overshoot (%), maximum power deviation (eN_{max} (%)), maximum pressure difference (eP_{max} (%)) in load rise and fall time. **Table 2** gives a comparison of the power response:

Both cases are compared without overshoot. The coordinated control model using GA has better power response quality than this architecture without using GA. The results show that the response of the structure using GA in terms of transient time when increasing/decreasing the load and the maximum power difference is smaller (**Table 3**).

New coordinated control structure	T_{qd} (s)		e_{max} (%)	
	Increasing load	Decreasing load	Increasing load	Decreasing load
Not using GA	822	1155	8.06	4.94
Using GA	668	692	1.76	1.62

Table 2.
 Evaluate the quality of power response.

New coordinated control structure	T_{qd} (s)		e_{max} (%)	
	Increasing load	Decreasing load	Increasing load	Decreasing load
Not using GA	812	2250	0.392	0.18
Using GA	660	2250	0.24	0.05

Table 3.
Evaluate the quality of pressure response.

Similar to the power response, the pressure response of the two cases to be compared has no overshoot. The combined control mode using GA has better pressure response quality in terms of transient times when the load is increased.

5.2 Evaluation with optimal operating standard

As mentioned above, this chapter only focuses on two objectives: fast-tracking the power setpoint (J_N) and saving fuel (J_f). Therefore, we compared these two criteria for the new coordinated control structure and this new coordinated control structure when using GA to optimize the controller parameters G_{CN} and G_{CP} , the results shown in **Table 4**:

New coordinated control structure	J_N (%)	J_f (kg/kWh)
Not using GA	2.83	0.4119
Using GA	2.63	0.41

Table 4.
Value of targets J_N (%), J_f (kg (coal)/kWh) from the simulation results.

From the results obtained in **Table 4**, it can be seen that the power tracking of the new coordinated control structure when GA is used is much smaller than that structure when GA is not used. This shows that the real power follows the setpoint when GA is used. The coal saving standard when using GA is smaller and saves 1.9 g/kWh. Thus, it is estimated that with a thermal power plant generating a capacity of 300 MW running 6000 hours/year with an output of 1.8 billion kWh, the coal saving amount of 1.9 g/kWh could have great economic value. The obtained results show that these parameters have been optimized by using the genetic algorithm GA.

6. Conclusions

The chapter has shown the advantages of using genetic algorithms in finding the optimal parameters of the controller in the thermal load control system in a coal-fired power plant. In this chapter, a genetic algorithm is designed for a new coordinated control structure with the goal of fast-tracking the set-point and fuel saving. Simulation modeling for these structures has been performed on MATLAB Simulink software. The structures evaluated which is the new coordination control structure proposed by the authors, the other optimized by using GA. The results show that all

the three structures have good control quality, stability, and the new coordination control structure optimized by a genetic algorithm is much better than the other two structures compared to the targets of tracking the set-point quickly and saving fuel (the coal saving amount of 1.9 g/kWh).

IntechOpen

Author details


PhamThi Ly^{1*} and Bui Quoc Khanh^{2*}

1 University of Transport and Communications, Hanoi, Vietnam

2 Institute for Control Engineering and Automation, Hanoi University of Science and Technology, Hanoi, Vietnam

*Address all correspondence to: ptlydk@utc.edu.vn and bqkhanh29@gmail.com

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Ventura S, Luna JM. Pattern Mining with Evolutionary Algorithms. Switzerland: Springer International Publishing; 2016. DOI: 10.1007/978-3-319-33858-3
- [2] Sivanandam SN, Deepa SN. Introduction to Genetic Algorithms. Berlin, Heidelberg: Springer-Verlag; 2008. ISBN 978-3-540-73189-4
- [3] Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-P. New Genetic Local Search Operator for the Traveling Salesman Problem. University of Siegen; 1996. DOI: 10.1007/3-540-61723-X_1052
- [4] Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Berlin, Heidelberg: Springer-Verlag; 1994. DOI: 10.1007/978-3-662-07418-3
- [5] Goldberg DE. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional; 1989
- [6] Garduno-Ramirez R, Lee KY. Multiobjective optimal power plant operation scheduling. IEEE Transactions on Energy Conversion. 2001;16(2):115-122
- [7] Technical Documents of the Hai Phong Thermal Power Plant. Haiphong Thermal Power Plant; 2017
- [8] Technical Documents of the Expanded Uong Bi Thermal Power Plant. Uong Bi Thermal Power Plant; 2001
- [9] Technical Documents of the Duyen Hai Thermal Power Plant. Duyen Hai Thermal Power Plant; 2018
- [10] Jayachitra A, Vinodha R. Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor. Advances in Artificial Intelligence. 2014;2014:1-8. DOI: 10.1155/2014/791230
- [11] Malhotra R, Singh N, Singh Y. Genetic algorithms: Concepts, design for optimization of process controllers. Journal of Computing and Information Science in Engineering. 2011;4(2):39-54. DOI: 10.5539/cis.v4n2p39
- [12] Saragih R. Parameters optimization of flexible structure using genetic algorithm. In: IEEE Int. Conf. Control Autom. ICCA; 2011. pp. 790-793. DOI: 10.1109/ICCA.2011.6137915
- [13] Sanchez-Lopez A, Arroyo-Figueroa A, Villavicencio-Ramirez G. Intelligent control algorithm for steam temperature regulation of thermal power plants. In: Mex. Int. Conf. Artif. Intell. Berlin, Heidelberg: Springer; 2004. pp. 754-763
- [14] Garduno-Ramirez R, Lee KY. A multiobjective-optimal neuro-fuzzy extension to power plant co-ordinated control. Transactions of the Institute of Measurement and Control. 2002;24(2): 129-152. DOI: 10.1191/0142331202tm056oa
- [15] Dhamanda A, Singh Rawat G. GA technique to solve the load frequency and tie-line power problem of thermal generating unit. Advanced Networks. 2019;7(2):51-58. DOI: 10.11648/j.net.20190702.16
- [16] Liu X, Guan P, Chan CW. Nonlinear multivariable power plant coordinate control by the constrained predictive scheme. IEEE Transactions on Control Systems Technology. 2010;18(5): 1116-1125. DOI: 10.1109/TCST.2009.2034640

[17] Wang D, Huang B, Meng L, Han P.
Predictive control for the boiler-turbine
unit using ANFIS. In: Proc. Int. Symp.
Test Meas. Vol. 2. 2009. pp. 1-4. DOI:
10.1109/ICTM.2009.5413102

[18] Flynn D, editor. Thermal Power
Plant Simulation and Control. Stevenage,
UK: IEE Press; 2003

IntechOpen