OPTIMAL TRAJECTORY PLANNING FOR CONTROL OF NUCLEAR RESEARCH REACTORS USING GENETIC ALGORITHMS AND ARTIFICIAL NEURAL NETWORKS

Ramazan COBAN
Cukurova University, Department of Computer Engineering, 01330 Balcali, Saricam, Adana, Turkey.
E-mail: rcoban@cu.edu.tr

ABSTRACT

In this study, an optimal trajectory planning based on artificial neural networks and genetic algorithms was proposed for control of nuclear research reactors. The trajectory being followed by the reactor power is composed of three parts. In order to calculate periods of all parts of the trajectory, a period generator was designed based on a feedforward neural network. Period values of the trajectory used to train the artificial neural network were acquired by utilizing genetic algorithms. The contribution of the proposed trajectory to the reactor control system was investigated. Furthermore, the behavior of the controller with the proposed trajectory was tested for various initial and desired power levels, as well as under disturbance. It was seen that the controller could control the system successfully under all conditions within the acceptable error tolerance.

Keywords: Reactor control, Genetic algorithms, Neural networks, Trajectory planning.

1. INTRODUCTION

Output power, rod position, period and fuel temperature of reactors were used as the controllers’ input variables in control of nuclear research reactors and these variables required fuzzification in previous works [1, 2]. A method including a trajectory has been proposed to eliminate the fuzzification of such variables of the controllers in [3, 4]. This method uses tracking error and rate of change of the error instead of those variables. However, period values of the trajectory were determined only by the user in those studies. For safer and faster control of reactors, optimal trajectory planning is required. For this purpose, the reactors have to follow the desired trajectory within the shortest time, which has been planned considering the minimum time and safety requirements. The first requirement for optimal trajectory planning is the minimum time at which the reactor power reaches its steady-state value. The second one is the safety requirement which is assumed here that the reactor power moves along the desired path with the minimum tracking error. In simulation studies for design of a controller when a trajectory is used, the choice of any trajectory must be appropriate to the physical
system so that simulation studies could be closer to the real system. Because of the physical limitations of the reactor, it can not be controlled with the same period values for all values of the desired and initial powers. The choice of a period value of the trajectory depends on human operator’s experience. Human operator generally selects these period values by trial and error method. This procedure is time consuming and tedious. To overcome this problem, a period generator was designed based on the artificial neural network (ANN) using trade-off between the shortest period information with which the reactor power reaches its steady-state value and minimum tracking error for safety requirement in [5]. However, in that work the neural network based period generator computes only the period of the main part of the trajectory. The remaining two parts are calculated depending on the main part. Furthermore, the period values used to train the neural network based period generator were obtained by trial and error. Probably, these values are not optimal or near optimal. However, the present work deals with all the periods of the three parts of the trajectory. Also, the trade-off between the shortest period information and minimum tracking error was optimized by employing genetic algorithms (GA). Hence, the optimal (or near optimal) period values of the trajectory through this period generator instead of any human operator are generated to insert automatically into the control system. To demonstrate the efficiency and effectiveness of this optimal trajectory planning, Triga Mark-II research reactor (ITU Triga Mark-II) located at Istanbul Technical University in Turkey was considered.

The reminder of the paper is organized as follows: The physical structure of ITU Triga Mark-II research reactor and the available control system is introduced at Section 2. The trajectory used in the study is explained in Section 3. Artificial neural networks and genetic algorithms are briefly explained at Section 4. The optimal trajectory is proposed in Section 5. Section 6 shows the simulation results regarding the control system including the proposed trajectory planning. The last section is devoted to the conclusions, discussions, and suggestions.

2. STRUCTURE OF ITU TRIGA MARK-II RESEARCH REACTOR

ITU Triga\(^1\) Mark-II research reactor is an open-tank-type one with light water coolant and graphite reflector. The reactor operates with solid fuel elements containing a homogeneous mixture of zirconium hydride moderator combined with partially enriched uranium. The reactor can be operated in two different modes named as steady-state mode and pulsed mode. It can reach a nominal power level of 250 kW at the steady-state and 1200 MW at the pulsed mode in a short time. The reactor has three neutron-absorbing control rods. These are transient rod, safety rod, and regulating rod. The safety and regulating rods can be fired with an electro-mechanical mechanism. The transient rod runs with pressurized air and the electro-mechanical mechanism at steady-state and also with pressurized air at pulsing operation [6].

In this work, the Yavcan Code developed by Can, Yavuz and Akbay [7] is employed for the non-linear behavior of ITU Triga Mark-II research reactor core. Eleven sets of neutronic and thermal-hydraulic differential equations are solved simultaneously by the modified Hansen’s method in the code. Two sets of Xenon and Iodine concentration dynamic equations are calculated by the fourth order Runge-Kutta method. The point reactor model with six delayed neutron groups is given as [7]:

\[
\frac{dn(t)}{dt} = \frac{P(t) - n(t)}{\Lambda} + \sum_{i=1}^{6} \lambda_{C_i}(t),
\]

(1)

\[
\frac{dC_i(t)}{dt} = \frac{\beta_{C_i}}{\Lambda} n(t) - \lambda_{C_i}(t), \quad i = 1, \ldots, 6.
\]

(2)

The thermal-hydraulic equations of the reactor core can be written as:

\[
\frac{dT_r(t)}{dt} = \frac{P(t)}{\rho_f V_f C_f N} - \mu N \left( T_r(t) - T_w(t) \right)
\]

(3)

\(^1\) Triga is an abbreviation for “Training, Research, Isotopes, General Atomics.”

R. COBAN
Optimal Trajectory Planning for Control of Nuclear Research Reactors Using Genetic Algorithms and Artificial Neural Networks

\[
\frac{dT_c(t)}{dt} = \frac{\mu}{\rho_s(t) V_n C_n} T_f(t) - \frac{\mu N + 2 \dot{m}_s(t) C_m}{\rho_s(t) V_n C_n N} T_m(t) + \frac{2 \dot{m}_s(t)}{\rho_s(t) V_n C_n} T_{in}
\]

(4)

\[
\frac{du(t)}{dt} = \zeta u^2(t) + \frac{\rho_u - \rho_s(t)}{\rho_s(t)} g
\]

(5)

\[
\frac{d\rho_s(t)}{dt} = -\frac{\rho_u - \rho_u(t)}{\rho_u H_c} u(t) \rho_s(t)
\]

(6)

where it is assumed that mass flow rate and inlet temperature of the coolant are time-independent.

Iodine and Xenon concentration dynamic equations are given as:

\[
\frac{dl(t)}{dt} = \gamma_s \sum \nu_r n(t) - \lambda_0 I(t)
\]

(7)

\[
\frac{dx(t)}{dt} = \gamma_s \sum \nu_r n(t) + \lambda_0 I(t) - [\lambda_0 + \sigma_r n(t)] x(t)
\]

(8)

Using the neutronic equations, the reactor core power can be represented as follows:

\[
P(t) = \gamma_s \sum \nu_r N V_n n(t)
\]

(9)

Furthermore, depending on the external reactivity, which is introduced through the regulating rod, and the reactivity feedback due to fuel temperature variation, the total reactivity with respect to time in the reactor core can be written as:

\[
\rho(t) = \rho_s(t) - \alpha_f (T_f(t) - T_{in}) - \frac{\sigma}{\varphi \Sigma_f} [x(t) - x_s]
\]

(10)

The reactor power level can be changed manually or automatically by using the control rods. In the manual control mode, position of the control rods is adjusted by an operator. In the automatic control mode, only the regulating rod position is arranged by the analog controller with electro-mechanical chopper. The available analog control system is shown in Fig. 1 [6].

The regulating rod only is handled to automatically control the power level of the reactor. The reactivity is inserted into the reactor core by movement of the regulating rod, so the power changes are provided in this way. The production of neutrons rises with withdrawing the regulating rod and therefore, amount of the reactivity insertion increases; the production of neutron decreases with pushing the rod into the core and so does amount of the reactivity insertion. Thus, control of the reactor power level is achieved. The regulating rod position is adjusted by a servomotor. The regulating rod speed is arranged by the exciting voltage exposed to the servomotor.

3. TRAJECTORY

Trajectory is a desired power planning, with which the power of a reactor is supposed to change in accordance. Use of a trajectory reduces some workload such as fuzzification of many variables and use of them as controller inputs encountered in fuzzy control of a reactor and saves computing time. In this work, the power of the reactor was intended to track a reference trajectory. For this purpose, it was presumed that the trajectory should be composed of three separate parts when considering the behavior of the physical system in choice of a trajectory [8]. A three-part trajectory was employed in the present study as seen in Fig. 2.
The first and third parts of the three-part trajectory are represented by a third order polynomial and the other one is represented by the second-order polynomial as [8]:

\[ P_{t_1} = a_3(t - t_0)^3 + a_2(t - t_0)^2 + a_1(t - t_0) + a_0, \]
for \( t_0 \leq t < t_1 \) \hspace{1cm} (11)

\[ P_{t_2} = b_2(t - t_1)^2 + b_1(t - t_1) + b_0, \]
for \( t_1 \leq t < t_2 \) \hspace{1cm} (12)

\[ P_{t_3} = c_3(t - t_2)^3 + c_2(t - t_2)^2 + c_1(t - t_2) + c_0, \]
for \( t_2 \leq t < t_d \) \hspace{1cm} (13)

\[ P_{t_d} = P_d, \text{ for } t \geq t_d \] \hspace{1cm} (14)

The reactor period is dependent to reactor dynamics. So, it is defined as the power level divided by the difference between the actual and desired power. Thus, the period expressions will be as follows [8]:

**R. COBAN**
Optimal Trajectory Planning for Control of Nuclear Research Reactors Using Genetic Algorithms and Artificial Neural Networks

where $P$ denotes the first-order derivative of the power. In Eqn (15), $\tau_1$, $\tau_2$, and $\tau_3$ denote the periods at the midpoints of the first, second, and third parts of the trajectory at $t' = (t_0 + t_1)/2$, $t'' = (t_1 + t_2)/2$, and $t''' = (t_2 + t_3)/2$, respectively, on the time axis in Fig. 2, and also $\tau$ is called the main period.

Given the powers $P_0$, $P_1$, $P_2$, and $P_d$ in Fig. 2 and the periods $\tau_1$, $\tau_2$, and $\tau_3$ in Eqn (15), then $t_0$, $t_1$, $t_2$, and $t_3$ can be calculated [8]. Furthermore, $P_1$ and $P_2$ are assumed as follows [5]:

$$P_1 = P_0 + 2P_d/100 \text{ and } P_2 = 80P_d/100$$

Considering a three-layer ANN as demonstrated in Fig. 3, upon presenting an input signal into the network, the outputs (activation) of each neuron are determined as [11]:

$$O_j = f(\sum w_{ji} O_i + \theta_j)$$

where $f(.)$ is the activation function.

If a momentum term $\alpha$ is added to accelerate the learning by reducing the oscillations at the output response, the changes in the connection weights are obtained as [11]:

$$\Delta w_{kj}(l+1) = \eta \delta_k O_j + \alpha \Delta w_{kj}(l)$$

Furthermore, the learning coefficient (step size) $\eta$ is adjusted to speed up the learning according to the following rules [12]:

$$\eta(l+1) = \eta(l) + \Delta \eta(l)$$

$$\Delta \eta = \begin{cases} +\gamma & \Delta E < 0 \\ -\xi \eta & \Delta E > 0 \\ 0 & \Delta E = 0 \end{cases}$$

where $E$ is a cost function.

4. DESIGN METHODS

Artificial neural networks (ANNs) and genetic algorithms (GA) used to design an optimal trajectory are briefly explained in the following subsections.

4.1. Artificial Neural Networks

An artificial neural network consists of a lot of neurons which is interconnected. In general, multilayer neural networks are used in modeling the physical systems. A multilayer neural network consisting of one input, one output, and one hidden layer is depicted in Fig. 3. The weights of neural networks are determined by the back-propagation algorithm in plenty of ANNs applications [9, 10]. The back-propagation algorithm minimizes a quadratic cost function by using the well-known gradient-descent methods. This algorithm was employed for training the ANN in this study.

4.2. Genetic Algorithms

The underlying principles of Genetic Algorithms (GA) were first proposed by Holland [13]. GA is inspired by the mechanism of natural selection where stronger individuals are likely to be the winners in a competitive environment. GA is a global and stochastic search algorithm based on genetic inheritance and strive for survival. A
Optimal Trajectory Planning for Control of Nuclear Research Reactors Using Genetic Algorithms and Artificial Neural Networks

standard GA comprises of three operators: Reproduction, crossover, and mutation [14]. Due to simplicity of structure, the standard GA together with a few modifications was used in the current work and henceforth it will be simply denoted by GA. A flow chart of GA is illustrated in Fig. 4. Genetic operations are performed in a pool called population (Pop) consisting of individuals. A fitness value $f_i$ is assigned to each individual in the pool or population by means of the fitness function. The higher the fitness of an individual, the higher the probability of it being selected for reproduction. The parents selected to produce offspring are derived from spinning the roulette wheel. The parents to produce offspring are selected by spinning the roulette wheel [14]. In the current work, the roulette wheel selection together with the elitist model was employed. The elitist model copies the fittest individual in the current generation into the next one. This method can increase the speed of the dominance of a population by means of a superior individual and hence can improve the local search [15]. New offspring which carries genes from the former generations in the search space is produced by crossover operator. A simple crossover process is carried out in two steps. In the first step, the individuals which have been recently copied in the mating pool are randomly matched to each other. In the second, every part of the individuals is exposed to crossover as follows: For crossover position, an integer $k$ is selected uniformly at random within $[1, l-1]$ where $l$ is length of the individual. Two new offspring strings are formed by swapping all characters between positions $k+1$ and $l$ [15]. Although crossover and reproduction give numerous new offspring, they don’t provide any new information into the population at the bit level. In a standard GA, mutation is applied as some new information source at the bit level. Bit positions of the individuals in the population are changed from 0 to 1 or vice versa according to the probability of mutation $p_m$. All bits in the population have equal probability of being subject to mutation [15].

In this work, a GA was used to find the optimal trajectory for a nuclear research reactor power. The period values of the three parts of the trajectory are coded in the individuals of GA by using real-number coding. Standard genetic algorithms used for the optimization problem under consideration include real-number coding (real-coded), roulette wheel selection together with the elitist model, single-point crossover, and uniform mutation. Proportional selection, elitist model, and crossover which have binary coding are also valid for real-number coding. But the situation for mutation is slightly different. While in the binary coding if the chosen bit is 0 it is made 1 and if 1 it is made 0, in the uniform mutation having real-number coding the chosen gene is relocated with a number which has been selected within the working domain of the corresponding parameter at random [15]. The parameters of GA used for optimization of the periods of the trajectory in this research are given in Table 1. These parameters were found by trial and error method and the proper results are given in the table.

R. COBAN
5. OPTIMAL TRAJECTORY DESIGN BASED ON the ANN AND GA

Three period values at the parts of the trajectory have to be suited to the behavior of the physical system. These periods vary with respect to the initial and desired power levels. Furthermore, the maximum value the control signal can take is limited. Accordingly, a trajectory suitable to every desired period value cannot be presumed. Hence, the choice of a period value relating to the trajectory depends on human operator’s experience. These periods had been found by trial and error and manually entered into the control system in the previous works containing a trajectory. This procedure is time consuming and tedious. In place of any human operator, a period generator can generate these period values. Once a period generator is constructed, the period values the trajectory requests are automatically produced. The proposed period generator is based on the ANN whose training data sets are determined by the GA. Its inputs are the initial and desired powers of the reactor, which were randomly selected. The corresponding outputs data were period values of the three parts of the trajectory, which were determined by employing the GA for the minimum tracking error and the fastest response where the reactor power can track the trajectory by running the simulation code developed in [5] at given initial and desired powers.

Before the ANN based period generator was trained, three periods of the trajectory used to train it were determined by the GA. So, there exist a total number of three adjustable parameters. These parameters need to be encoded into the individuals of the GA. The floating-point coding proposed by Michalewicz [15] was used here. An individual constructed in this way is given in Fig. 6. In addition to this, the following issues have to be solved to train a ANN based period generator by means of GA:

- Encoding the adjustable parameters of the trajectory into the individuals of GA,
- Transforming the performance relating to the controlled system developed in [5] into a fitness function which GA will use.

A performance criterion for control systems may contain a weighted combination of various performance characteristics like rise time, settling time, overshoot, and steady-state error. Such a performance criterion is generally employed while the response of a system to a reference input in the form of unit step function is investigated. Here, the output power of the reactor has to trace a reference power value supplied by the trajectory at every time step. So, the main objective of the control system is to make the tracking error zero or minimum. It is also important to take the response time into account.
account. Otherwise, it takes the controlled system so much time to reach its steady-state value while the tracking error can be minimum. The performance criterion for the control system consisting of response time and tracking error is written in the following equations:

\[
\phi_r = 100 \times \frac{1}{D} \sum_{i=1}^{n} \frac{|P_i - P_t|}{P_t} \quad (21)
\]

\[
\phi_f = \frac{100 * t_f}{t_{max}} \quad (22)
\]

\[
\phi_s = \frac{100 * t_s}{t_{max}} \quad (23)
\]

\[
\phi_t = \frac{100 * t_t}{t_{max}} \quad (24)
\]

where \( D \) and \( t_{max} \) denote size of data in a transient and duration of a transient, respectively. Durations of the first, second and third parts of the trajectory in Eqns (22)-(24) were regarded as response time. They are written in the following:

\[
t_f = t_1 - t_0 \quad (25)
\]

\[
t_s = t_2 - t_1 \quad (26)
\]

\[
t_t = t_d - t_2 \quad (27)
\]

where \( t_0, t_1, t_2, \) and \( t_d \) denote initial time of the first part, ending time of first part (or initial time of second part), ending time of second part (or initial time of third part), and ending time of third part of the trajectory, respectively, as stated previously.

When the GA is implemented for an optimization problem it must be decided about how to assess the evaluation of the individuals in the population. For this purpose, a fitness value is assigned to each individual. The fitness function responsible for assigning a fitness value changes from problem to problem. The task of GA here is to try to do the tracking error to be zero or minimum. Whichever individual in the population makes this error minimum will take the maximum fitness value. Obviously, the smaller the error is, the bigger the fitness value will be.

Using Eqns (21)-(24), the fitness function to assess the fitness values of the individuals can be written in terms of performance criterion as:

\[
f_r = \frac{100}{\phi_r} \quad (28)
\]

\[
f_f = \frac{100}{\phi_f} \quad (29)
\]

\[
f_s = \frac{100}{\phi_s} \quad (30)
\]

\[
f_t = \frac{100}{\phi_t} \quad (31)
\]

where \( f_r, f_f, f_s, \) and \( f_t \) denote fitness functions of each component of the performance criterion.

As for many real-world problems the problem at hand is a nonlinear, multi-objective and constrained one. There are some classical methods for multi-objective optimization. These include a method of objective weighting, where multiple objective functions are combined into one overall objective function. For the problem under consideration total fitness function is the weighted summation of the fitness functions \( f_r, f_f, f_s, \) and \( f_t \):

\[
f_{\text{sum}} = K_1 * f_r + K_2 * f_f + K_3 * f_s + K_4 * f_t \quad (32)
\]

where \( K_1, K_2, K_3, \) and \( K_4 \) are weight coefficients of the fitness functions. Their values were found by trial and error and are given in Table 2.

Table 2 Weight coefficients of the fitness functions.

<table>
<thead>
<tr>
<th>Weight coefficients</th>
<th>Their values</th>
</tr>
</thead>
<tbody>
<tr>
<td>K_1</td>
<td>0.02</td>
</tr>
<tr>
<td>K_2</td>
<td>0.38</td>
</tr>
<tr>
<td>K_3</td>
<td>0.57</td>
</tr>
<tr>
<td>K_4</td>
<td>0.03</td>
</tr>
</tbody>
</table>

To incorporate constraints into a GA search, the penalty method was used. In the penalty method, a constrained problem in optimization is transformed to an unconstrained problem by associating a penalty with all constraint violations. For this purpose, the mean absolute percentage error (MAPE) of each part of the trajectory was calculated as follows:
Optimal Trajectory Planning for Control of Nuclear Research Reactors Using Genetic Algorithms and Artificial Neural Networks

\[ e_T = 100 \cdot \frac{1}{T_{\text{tr}}} \int_{t_{\text{tr}}}^{T_{\text{tr}}} \left| \frac{P_T - P_i}{P_T} \right| dt \]  \hspace{1cm} (33)

\[ e_e = 100 \cdot \frac{1}{T_{\text{te}}} \int_{t_{\text{te}}}^{T_{\text{te}}} \left| \frac{P_T - P_i}{P_T} \right| dt \]  \hspace{1cm} (34)

\[ e_s = 100 \cdot \frac{1}{T_{\text{ts}}} \int_{t_{\text{ts}}}^{T_{\text{ts}}} \left| \frac{P_T - P_i}{P_T} \right| dt \]  \hspace{1cm} (35)

\[ e = 100 \cdot \frac{1}{T_{\text{max}}} \int_{t_{\text{max}}}^{T_{\text{max}}} \left| \frac{P_T - P_i}{P_T} \right| dt \]  \hspace{1cm} (36)

where \( e_T \), \( e_e \), and \( e_s \) denote the mean absolute percentage error of first, second, and third part of the trajectory, respectively. \( e \) indicates the mean absolute percentage error of a transient. The penalty was included in the objective function evaluation in the following:

If \( e > E_e \) then \( f_e = 0 \), \( f_i = 0 \), \( f_s = 0 \),
If \( e_T > E_{e_T} \) then \( f_i = 0 \),
If \( e_e > E_{e_e} \) then \( f_s = 0 \),
If \( e_s > E_{e_s} \) then \( f_i = 0 \),

(37)

where \( E_T \), \( E_e \), \( E_s \), and \( E_{e_T} \) refer the permissible mean absolute percentage errors of a transient, first, second, and third part of the trajectory, respectively.

Before the period parameters of the trajectory are determined by GA, their ranges need to be determined. The regulation ranges of the parameters used in optimization process are presented in Table 3.

Table 3 Regulation ranges of the period parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Regulation ranges (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau )</td>
<td>1 - 150</td>
</tr>
<tr>
<td>( \tau_1 )</td>
<td>1 - 600</td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>1 - 300</td>
</tr>
</tbody>
</table>

With many scenarios describing different operating conditions of the reactor at given initial and desired powers, GA integrated with the control system developed in [5] was run ten times for each training data set of three parameters. The best result of ten runs was recorded. In doing so, 600 data set were obtained to use for training the ANN based generator.

ANN based generator includes hyperbolic tangent function type activation functions, and the normalization range for data was \([-0.9; +0.9]\). All weights and biases were adapted by the back propagation algorithm in the learning process once each data pair was presented into the ANN. The learning rate was adjusted by Eqns (19) and (20). The initial values of weights, biases, momentum, learning rate, and number of neurons in hidden layer were found by trial and error and the best values are given in Table 4.

After the training process, the network matrix composed of weights and biases was saved into a file such that it could be used for test process. The test results from the control system with the proposed trajectory [see Fig. 7] are compared with those from the controlled system with the trajectory developed previously by Coban [5]. The obtained test results will be discussed in Section 6.

Table 4 The parameters of the ANN.

<table>
<thead>
<tr>
<th>Networks size</th>
<th>2-20-20-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial values of weights and biases</td>
<td>[-0.05; +0.05]</td>
</tr>
<tr>
<td>Initial values of learning coefficient</td>
<td>0.02</td>
</tr>
<tr>
<td>Initial values of momentum coefficient</td>
<td>0.8</td>
</tr>
<tr>
<td>Increasing coefficient of step size</td>
<td>0.0001</td>
</tr>
<tr>
<td>Decreasing coefficient of step size</td>
<td>0.2</td>
</tr>
<tr>
<td>Number of data used in learning</td>
<td>600</td>
</tr>
<tr>
<td>Number of Iteration</td>
<td>20000</td>
</tr>
<tr>
<td>Training error</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

6. SIMULATION RESULTS

This section presents simulation results from the proposed trajectory and overall control system’s performance. As a performance criterion a mean absolute percentage error (MAPE) was selected and can be written in the following form:
1124

Optimal Trajectory Planning for Control of Nuclear Research Reactors Using Genetic Algorithms and Artificial Neural Networks

Fig. 7. The control system.

\[ MAPE = 100 \times \frac{1}{D_s} \sum_i \left| \frac{P_1 - P_2}{P_1} \right| \]  

(38)

where \( D_s \) denotes the size of data.

6.1. Simulation Results from The ANN and GA Based Period Generator

The period values obtained by using the period generator developed in this research and in the reference [5] are given in Table 5. As seen in the table, for the initial and desired power pairs (15, 6450), (487, 12900), (1650, 28750), (3250, 198540), (6980, 172500), (8120, 14750), (9370, 249500) Watts except the power pairs (75, 120000), (100, 22500), and (198, 45000) Watts the main periods \( \tau \) predicted by the period generator developed in this research are smaller than in the reference [5]. For the periods \( \tau_1 \) the period generator developed in this research performs better than the other for almost most of the power pairs. The periods \( \tau_2 \) are smaller for all the power pairs in the table. As a result, for the almost same error the period generator proposed here produces smaller period values than the one proposed in the reference [5] for most of the power pairs. Smaller period values in turn enable faster response time.

Table 5 The test results for the ANN and GA based-period generator.

<table>
<thead>
<tr>
<th>( P_0 ) (W)</th>
<th>( P_d ) (W)</th>
<th>with the previous work</th>
<th>with the present work</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau ) (sec.)</td>
<td>( \tau_1 ) (sec.)</td>
<td>( \tau_2 ) (sec.)</td>
<td>( \tau ) (sec.)</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>15 6450</td>
<td>5.44</td>
<td>16.32</td>
<td>16.32</td>
</tr>
<tr>
<td>75 120000</td>
<td>5.73</td>
<td>17.19</td>
<td>17.19</td>
</tr>
<tr>
<td>100 22500</td>
<td>5.87</td>
<td>17.61</td>
<td>17.61</td>
</tr>
<tr>
<td>198 45000</td>
<td>5.90</td>
<td>17.70</td>
<td>17.70</td>
</tr>
<tr>
<td>487 12900</td>
<td>8.55</td>
<td>25.65</td>
<td>25.65</td>
</tr>
<tr>
<td>1650 28750</td>
<td>12.46</td>
<td>37.38</td>
<td>37.38</td>
</tr>
<tr>
<td>3250 198540</td>
<td>5.95</td>
<td>17.85</td>
<td>17.85</td>
</tr>
<tr>
<td>6980 172500</td>
<td>10.93</td>
<td>32.79</td>
<td>32.79</td>
</tr>
<tr>
<td>8120 14750</td>
<td>144.31</td>
<td>432.93</td>
<td>432.93</td>
</tr>
<tr>
<td>9370 249500</td>
<td>10.27</td>
<td>30.81</td>
<td>30.81</td>
</tr>
</tbody>
</table>

6.2. Comparison with a Previous Work

The results obtained from the overall control system including the optimal trajectory planning proposed in the current work were compared with those from the control system including the trajectory suggested in [5]. First, it was asked the controller to control the reactor at an initial power of 100 W for a 150-kW desired power. The optimal period generator produced the period values (5.82, 16.01, 12.97) seconds whereas the trajectory developed in [5] generated them (5.58, 16.75, 16.75) sec. In such a case, errors computed depending on Eqn ((38)) for the optimal trajectory and for the trajectory proposed in [5] were 3.6%. The control system with the optimal trajectory performed faster than the other
with the almost same tracking error. Time difference between them to reach steady-state power levels is about 5 sec. as seen in Fig. 8.

Fig. 8. Comparison of the control system including the optimal trajectory with the one including the previous trajectory for $P_0 = 100$ W and $P_d = 150$ kW.

Taking the initial and desired power levels $P_0 = 100$ W and $P_d = 50$ kW, a simulation experiment was carried out for the control system with the optimal trajectory. In this case, a good performance with an error of 3.6\% was achieved. The period values $\tau$, $\tau_1$, and $\tau_2$ were 6.07, 15.34, and 13.44 seconds, respectively. The simulation was repeated with the same initial and desired power levels for inserting the period values by the period generator proposed in [5]. The power tracking error for the previous trajectory was almost the same as the one for the optimal trajectory. The period values $\tau$, $\tau_1$, and $\tau_2$ for the previous trajectory were 5.93, 17.78, and 17.78 seconds, respectively. In this numerical experiment, response of the control system with the proposed trajectory was 10 sec. faster than that of the control system with the previous trajectory. The results are illustrated in Fig. 9.

Fig. 9. Comparison of the control system including the optimal trajectory with the one including the previous trajectory for $P_0 = 100$ W and $P_d = 50$ kW.

The control system was tested taking the initial and desired powers 5 kW and 50 kW, respectively. The tracking errors for the optimal trajectory and for the one proposed in [5] were 0.52\% and 0.53\%, respectively. The period generator proposed in this research predicted the periods $\tau = 15.48$, $\tau_1 = 73.87$, and $\tau_2 = 46.56$ seconds. The period values for the trajectory proposed in [5] were $\tau = 23.54$, $\tau_1 = 70.62$, and $\tau_2 = 70.62$ seconds. The control system including the optimal trajectory reached its steady-state value 20 sec. earlier than the other. Fig. 10 shows the obtained results.
6.3. Disturbance effect

Some simulation experiments were done in order to investigate the behavior of the control system under noisy conditions. The reactor was started up at 100 W for 150 kW. A reactivity of (0.4) $S$ was applied at $120^{th}$ seconds during 80 seconds. The power increased up to 252 kW, so the controller inserted the rod to the position 400 in order to get the power down again to 150 kW (Fig. 11). It was seen that it could control the reactor successfully in the range of (-0.7) $S$ and (+0.4) $S$. It should be noted that the disturbing reactivities worth not to cause the reactor power exceed 275 kW were considered due to some safety regulations.

6. CONCLUSIONS

In this paper, an optimal trajectory planning has been developed for control of research reactors. The trajectory which is being followed by the reactor power is composed of three parts. A period generator based on ANN and GA has been developed to predict the period values on the midpoint of each part of the trajectory. The initial and desired powers as inputs and the period values as outputs to ANN have been considered. Three period values contrary to those in the previous work, which considered only one period value (the main period), that have been generated by the period generator can be automatically inserted into the control system. The period values used to train the ANN based period generator were obtained by using GA whereas in the previous work they were obtained by trial and error. In this way, the optimal trajectory planning suitable to the behavior of the
physical system has been accomplished.

The control system with the proposed optimal trajectory has been tested for various initial and desired powers and it has been demonstrated that it can control the reactor successfully for every situation, within the acceptable error limits. Response of the control system with the optimal trajectory is faster than that of the control system with the previous trajectory. The control system has been also tested against external disturbances. It has demonstrated that the controller is robust against the disturbances in the range of (-0.7) $ and (+0.4) $.

NOMENCLATURE

ANN Artificial neural networks
$n(t)$ Neutron density
$\rho(t)$ Total reactivity
$\beta$ Total delayed neutron fraction
$A$ Neutron generation time
$\lambda_i$ i$^{th}$ group delayed neutron decay constant
$C_i(t)$ i$^{th}$ group precursor concentration
$\beta_i$ i$^{th}$ group delayed neutron fraction
$T_f(t)$ Fuel temperature
$P(t)$ Reactor power
$\rho_f$ Density of fuel
$V_f$ Volume of the fuel
$C_f$ Heat capacity of the fuel
$N$ Number of the fuel elements
$a$ Total heat transfer coefficient of a fuel element
$T_o(t)$ Coolant temperature
$\rho_m(t)$ Density of coolant
$V_m$ Volume of the coolant
$C_m$ Heat capacity of the coolant
$m_r(t)$ Total mass flow rate of coolant
$C$ Specific heat
$T_{min}$ Inlet temperature of coolant
$u(t)$ Coolant velocity
$g$ Gravity
$H_f$ Total core height
$I(t)$ Iodine concentration
$\gamma_I$ Yield constant for Iodine
$\lambda_I$ Decay constant for Iodine
$x(t)$ Xenon concentration
$\gamma_x$ Yield constant for Xenon
$\lambda_x$ Decay constant for Xenon
$\sigma_x$ Thermal microscopic absorption cross section for Xenon
$x_0$ Initial value of Xenon concentration
$\gamma_f$ Energy released in a nuclear fission reaction
$\Sigma_f$ Thermal group macroscopic cross section
$v_n$ Neutron velocity
$\rho_m(t)$ External reactivity inserted into reactor
$\alpha_j(T_f)$ Temperature coefficient of reactivity
$T_f0$ Initial temperature of a fuel
$O_j$ Activation of unit $j$
$f()$ Activation function
$w_{ij}$ Weight from unit $i$ to $j$
$O_i$ Activation of unit $i$
$\theta_j$ Bias for unit $j$
$\Delta w_{ij}(l+1)$ Change of weight
$l$ Iteration number
$\eta$ Learning rate
$\delta_k$ Error signal for unit $k$
$\alpha$ Momentum factor
$U$ Control action
$\dot{P}$ First order derivative of the power $P(t)$
$\tau_1$ Period at first part of the trajectory
$\tau$ Period at second (main) part of the trajectory
$\tau_2$ Period at third part of the trajectory
$P_0$ Initial steady-state power level
$P_d$ Desired power
$e$ Error
$P_T$ Power predicted by the trajectory
$\Delta e$ Change of error
$D$ Size of data
$MAPE$ Mean absolute percentage error.
7. REFERENCES


Ramazan COBAN received the B.Sc. degree from the Yildiz Technical University, Istanbul, Turkey in electrical engineering in 1994, the M.Sc. and Ph.D. degrees from the Gebze Institute of Technology, Gebze, Turkey, both in electronics engineering in 1999 and 2006, respectively. He has been with the Department of Computer Engineering, Cukurova University since 2008 where he currently holds an assistant professor position. His current research interests are control engineering, fuzzy logic, neural networks.

R. COBAN