Nuclear Engineering and Technology 50 (2018) 877–885

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Control of the pressurized water nuclear reactors power using optimized proportional–integral–derivative controller with particle swarm optimization algorithm

Seyed Mohammad Hossein Mousakazemi, Navid Ayoobian*, Gholam Reza Ansarifar

Department of Nuclear Engineering, Faculty of Advanced Sciences and Technologies, University of Isfahan, Hezarjarib Avenue, 81746-73441 Isfahan, Iran

ARTICLE INFO

Article history:
Received 27 August 2017
Received in revised form
14 April 2018
Accepted 27 April 2018
Available online 28 May 2018

Keywords:
Gain Tuning
Lyapunov Stability Approach
Particle Swarm Optimization
Pressurized Water Reactor
Proportional–Integral–Derivative Controller
Optimization

ABSTRACT

Various controllers such as proportional–integral–derivative (PID) controllers have been designed and optimized for load-following issues in nuclear reactors. To achieve high performance, gain tuning is of great importance in PID controllers. In this work, gains of a PID controller are optimized for power-level control of a typical pressurized water reactor using particle swarm optimization (PSO) algorithm. The point kinetic is used as a reactor power model. In PSO, the objective (cost) function defined by decision variables including overshoot, settling time, and stabilization time (stability condition) must be minimized (optimized). Stability condition is guaranteed by Lyapunov synthesis. The simulation results demonstrated good stability and high performance of the closed-loop PSO–PID controller to response power demand.

1. Introduction

The development of load-following issues in nuclear reactors has always been of interest to researchers because of their nonlinear nature and the dependence of some dynamic parameters to the output power level. Accordingly, various controllers have been designed and optimized [1–4]. For example, Upadhyaya et al.[5] used a T-average controller on the primary side of integral pressurized water reactor (PWR) [5]. Proportional–integral–derivative (PID) controllers are widely used in various industries including nuclear facilities [6]. Therefore, various methods of PID gains tuning have been developed [7–9], and several methods have been used to optimize these gains for load-following in the nuclear power plants. Intelligent methods, such as fuzzy logic, have been at the forefront of these efforts. A comparative study of fuzzy, PID, and advanced fuzzy controls to simulate a nuclear reactor operation based on the experimental data was done by Li and Ruan [10]. Liu et al. [11] designed and optimized fuzzy-PID controller to control the nuclear reactor power and used the genetic algorithm to improve the "extending" precision. Their simulation results demonstrated good performance of the fuzzy-PID controller. Ye et al. [12] investigated water level control of a PWR based on radial basis function neural network and PID controller. The results showed remarkable robustness, adaptive ability, and higher control accuracy of this method. Dong [13] has used a physical approach to design proportional–derivative (PD) power-level control for a PWR. The globally asymptotic stability was established for the reactor state variables. This method has been shown to be suitable for the cases in which the state-space model is used.

Particle swarm optimization (PSO) is a metaheuristic and real-coded algorithm. PSO is originally credited to Kennedy and Eberhart [14]. Primarily, it was intended by Shi and Eberhart [15] to simulate social behavior. de Moura Meneses et al. [16] have applied PSO to the nuclear reload problem of a PWR. Also, Pereira et al. [17] have used PSO for nonperiodic preventive maintenance scheduling programming for a high-pressure injection system of a typical 4-loop PWR. The power-level control is popular in comparison with other control methods such as coolant temperature. The numerous studies have been conducted on the reactor power-level control [18–20]. For example, Ansarifar and Akhavan [21] have employed sliding mode control design for a PWR during load-following

* Corresponding author.
E-mail addresses: smh_mousakazemi@pnu.ac.ir (S.M.H. Mousakazemi), n.ayoobian@ast.ui.ac.ir (N. Ayoobian), ghr.ansarifar@ast.ui.ac.ir (G.R. Ansarifar).

https://doi.org/10.1016/j.net.2018.04.016

operation. The present study is trying to optimize (tune) and schedule PID gains using the PSO algorithm. This controller is tuned to control a PWR-type nuclear reactor based on point kinetic model with any power demand (set point). The tuned PID is used to control relative power level changes which are equivalent to the relative neutron density/flux. It is shown that the coolant temperature is controlled along with the power level. The optimization is performed by minimizing an objective function of decision variables including overshoot, settling time, and stabilization time. Therefore, the tracking error between the output of the system and desired set point is minimized in each time interval.

2. Materials and methods

2.1. Nuclear reactor model

In this work, the point kinetic model of a nonlinear PWR core has been used with three groups of delayed neutrons (Skinner–Cohen's three groups model) and reactivity feedbacks due to changes in xenon concentration, lumped fuel, and coolant temperature (Eqs. (1)–(8)) [22]:

$$\frac{dn_r}{dt} = \frac{\rho_t - \beta}{\Lambda} n_r + \sum_{i=1}^{3} \frac{\beta_i}{\Lambda} c_{ri} \tag{1}$$

$$\frac{dc_{ri}}{dt} = \lambda_i n_r - \lambda_i c_{ri}, \quad i = 1, 2, 3 \tag{2}$$

$$\frac{dX}{dt} = \left(\gamma_X \Sigma_f - \sigma_X X\right) \frac{P_0}{G \Sigma_f V} n_r - \lambda_X X + \lambda_I I \tag{3}$$

$$\frac{dI}{dt} = \gamma_I \Sigma_f \frac{P_0}{G \Sigma_f V} n_r - \lambda_I I \tag{4}$$

$$\frac{dT_f}{dt} = \frac{f_f P_0}{\mu_f} n_r - \frac{\Omega}{\mu_f} T_f + \frac{\Omega}{2\mu_f} T_{in} + \frac{\Omega}{2\mu_f} T_{out} \tag{5}$$

$$\frac{dT_c}{dt} = \frac{\left(1 - f_f\right) P_0}{\mu_c} n_r - \frac{(2M + \Omega)}{2\mu_c} T_{out} + \frac{(2M - \Omega)}{2\mu_c} T_{in} \tag{6}$$

$$\frac{d\rho_{rod}}{dt} = G_r Z_r \tag{7}$$

$$\begin{aligned} \rho_t &= \rho_{rod} + \rho_T + \rho_X \\ &= \rho_{rod} + \alpha_f \left(T_f - T_{f0}\right) + \alpha_c (T_c - T_{c0}) - \frac{\sigma_X}{\nu \Sigma_f} (X - X_0) \end{aligned} \tag{8}$$

The parameters in Eqs. (1)–(8) are shown in Table 1. Also, the parameter values of a typical PWR at the beginning of fuel cycle in 100% of nominal power are displayed in Table 2.

In addition, μ_c, M, Ω, α_f, and α_c are not constant but rather a function of the initial equilibrium power level (n_{r0}) as follows [23]:

$$\mu_c = \left(\frac{16}{9}\right) n_{r0} + 54.022 \tag{9}$$

$$M = 28 n_{r0} + 74 \tag{10}$$

$$\Omega = \left(\frac{5}{3}\right) n_{r0} + 4.93333 \tag{11}$$

$$\alpha_f = (n_{r0} - 4.24) \times 10^{-5} \tag{12}$$

$$\alpha_c = (-4 n_{r0} - 17.3) \times 10^{-5} \tag{13}$$

2.2. PID controller

The PID controller is the simplest controller to design and use in about 90% of industries as real-time controllers. It alone indicates the importance of this controller [9]. The proportional–integral (PI) controller can also be used as regards it is less responsive to real and relatively rapid changes in state, and the system will be slower to meet the desired signal. This can be important in controlling of accidents and highly rapid changes in power. In addition, PID controller has less overshoot and settling time compared to PI controller [24]. PID equation specifies as follows:

$$C(t) = K_P + \frac{1}{s} K_I + \frac{s}{1 + \tau s} K_D, \tag{14}$$

Table 1
Model parameters.

P_0	Full core power, MW	Λ	Neutron generation time, s
n_r	Normalized neutron density (relative to neutron density at rated power—P_0)	λ_i	ith Delayed neutron group decay constant, s^{-1}
c_{ri}	ith Group normalized precursor density (relative to density at rated power)	γ_X	Xenon yield per fission
X	Xenon concentration, cm^{-3}	λ_X	Xenon decay constant, s^{-1}
I	Iodine concentration, cm^{-3}	γ_I	Iodine yield per fission
T_f	Fuel average temperature, °C	λ_I	Iodine decay constant, s^{-1}
T_{f0}	Fuel average temperature at the initial condition, °C	Σ_f	Macroscopic thermal neutron fission cross-section, cm^{-1}
T_c	Coolant average temperature, °C	ν	Average number of neutrons produced per fission of ^{235}U
T_{c0}	Coolant average temperature at the initial condition, °C	σ_X	Microscopic thermal neutron absorption cross-section of xenon, cm^{-2}
T_{in}	Coolant inlet temperature, °C	G	Useful thermal energy liberated per fission of ^{235}U, MW·s
T_{out}	Coolant outlet temperature, °C	V	Core volume, cm^3
ρ_t	Total reactivity, δK/K	f_f	Fraction of reactor power deposited in the fuel
ρ_{rod}	Reactivity due to control rod movement, δK/K	μ_f	Fuel total heat capacity, MW·s/°C
ρ_T	Temperature reactivity feedback, δK/K	μ_c	Coolant total heat capacity, MW·s/°C
ρ_X	Xenon reactivity feedback, δK/K	M	Mass flow rate time heat capacity of water, MW/°C .
Z_r	Control rod speed, fraction of core length/s	Ω	Coefficient of heat transfer between fuel and coolant,MW/°C
G_r	Control rod total reactivity, δK/K	α_f	Fuel temperature coefficient, (δK/K)/°C
β	Effective delayed neutron fraction, $\beta = \sum_{i=1}^{3} \beta_i$	α_c	Coolant temperature coefficient, (δK/K)/°C
β_i	ith Group effective delayed neutron fraction		

Table 2
A typical PWR parameter values at BOC, in 100% of nominal power.

Parameter	Value	Parameter	Value
Thermal power	3000 MW	β	0.0065
Core height	400 cm	β_1	0.00021
Core radius	200 cm	β_2	0.00225
σ_X	3.5×10^{-18} cm^2	β_3	0.00404
Σ_f	0.3358 s^{-1}	λ_1	0.0124 s^{-1}
G	3.2×10^{-11} MW·s	λ_2	0.0369 s^{-1}
γ_X	0.003	λ_3	0.632 s^{-1}
γ_I	0.059	G_r	14.5×10^{-3} δK/K
λ_X	2.1×10^{-5} s^{-1}	T_{in}	290 °C
λ_I	2.9×10^{-5} s^{-1}	μ_f	26.3 MW·s/°C
Λ	10^{-4} s	f_f	0.92

BOC, beginning of fuel cycle; PWR, pressurized water reactor.

where $K_P, K_I, K_D \in R$ are the proportional, integral, and derivative gains, respectively, and are set by an optimization method. Also, the derivative action time constant ($\tau > 0$) is assumed to be fixed. To make a pure differentiation, $1/\tau$ is assumed to be large enough. The control system is shown in Fig. 1 and is expressed by the following equation:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\mathrm{d}\tau + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t) \tag{15}$$

where r: desired signal; e: error signal; u: plant input signal; y: plant output signal.

Methods like Ziegler–Nichols, trial-and-error, D-partitioning, and pole placement have been proposed to tune PID gains [25]. These methods are used for linear time invariant systems that are required for extensive knowledge and frequency response of the system. PID gains tuning is implemented for nonlinear systems by various metaheuristic optimization methods. In this work, to avoid reducing accuracy in the linearization of the state-space equations, real equations were used and simulated in MATLAB Simulink environment.

2.3. Stability condition

Lyapunov synthesis is used to analyze the stability condition of the designed PSO–PID code. The Lyapunov-like function is defined as below [26]:

$$V = \frac{1}{2} e^2 \tag{16}$$

where e is tracking error of the desired relative power; $e = P_r - P_{rd}$; $P_r = n_r = \phi_r$.

The derivative of the Lyapunov-like function (16) is identified as follows:

$$\dot{V} = \dot{e} \cdot e \leq 0 \tag{17}$$

where $\dot{e}$ is derivative track error of the desired relative power; $\dot{e} = \dot{P}_r - \dot{P}_{rd}$.

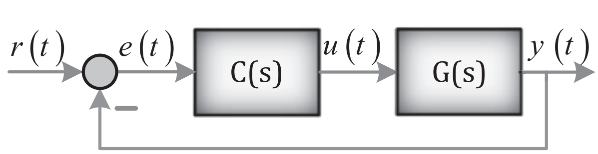

Fig. 1. Classical PID system. $C(s)$, Controller; $G(s)$, plant. PID, proportional–integral–derivative.

When the Lyapunov function is bounded, $\dot{V} \leq 0$, the stability condition is satisfied. The implementation of this criterion is indirectly accomplished by finding the stabilization time and entering it into the objective function of the optimization algorithm. This stability analysis is not theoretically used in determining the feedback gains of PID.

2.4. Particle swarm optimization

PSO is a real-coded algorithm. This algorithm is a metaheuristic and solves problems with the least information. Because of repeated evolution mechanisms, some people classified it in evolutionary algorithms, but in fact, it is in the swarm intelligence category. The main elements of swarm intelligence are shown in Fig. 2.

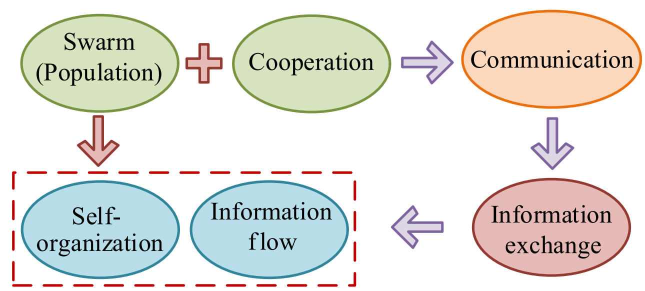

Fig. 2. The main elements of swarm intelligence.

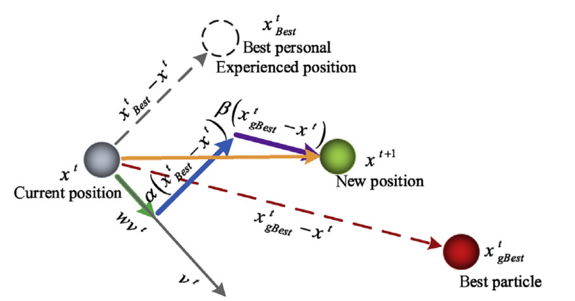

Fig. 3. Particles movement pattern in PSO algorithm ($\alpha = U_1(0,1)$, $\beta = U_2(0,1)$). PSO, particle swarm optimization.

```
Algorithm: PSO (npop, c1, c2)
% Initialize generation 1:
        k = 1;
        Pk = a population (particle swarm) of nPop ...
            randomly-generated positions in each dimension;
% Evaluate Pk:
        Compute cost(i) for each particle i ∈ Pk;
do {
        % 1. Calculate velocity:
        Calculate the velocity of particles;
        % 2. Update position:
        Update the position of particles;
        % 3. Evaluate:
        Compute cost(i) for particles;
        % 3. Select pBest:
        Selecting the best personal cost for particles ...
            compared to the best previous own cost;
        % 4. Select gBest:
        Selecting the best cost for particles compared to ...
            the best previous global cost;
        % Increment:
        k = k + 1;
} while termination criterion is not reached;
return the best cost particle from Pk;
```

Fig. 4. Pseudo-code of PSO.
PSO, particle swarm optimization.

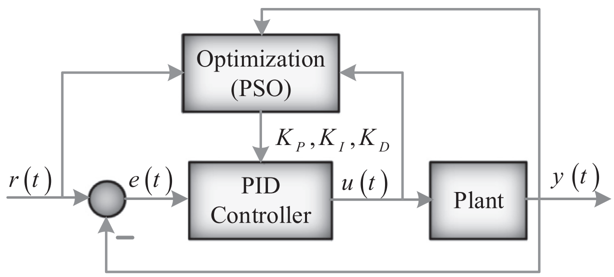

Fig. 5. The diagram of the proposed PSO–PID controller.
PID, proportional–integral–derivative; PSO, particle swarm optimization.

2.4.1. *Algorithm*

According to PSO, each particle in space has a position. In the optimization problem, there is an objective function which is intended to minimize (cost) or maximize (fitness). So because of the location, the objective function is also there. The particle is moving in the direction of the weighted sum of its earlier direction of the vector displacement to the best personal position and the displacement vector to the best global position (Fig. 3).

In terms of mathematics, if the particle is situated at the time t and to decide its speed at the time $t+1$, Eqs. (18) and (19) are used. This process has been named self-organization law in PSO, and all particles are obliged to this law. It means that all particles initially update their speeds, and new velocity vector is added to the current position of each particle; thus, the new position is determined. Also, the best personal position should be updated; that is, the improvement of personal/global records, are checked.

$$v_i^{t+1} = \omega v_i^t + c_1 U_1(0,1) \times \left(x_{i,Best}^t - x_i^t\right) + c_2 U_2(0,1) \times \left(x_{gBest}^t - x_i^t\right) \tag{18}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad i = 1, 2, \ldots, n_{pop} \tag{19}$$

where n_{pop}: population number (particle swarm); t: iteration (generation) index; x_i: the position of the ith particle; $x_{i,Best}$: the best personal position experienced of the ith particle; x_{gBest}: the best global position experienced in all particles up to iteration t; v_i: the speed of the ith particle; ω: inertia weight; c: acceleration coefficient (learning factor); and $U(0,1)$: a uniform random number generator.

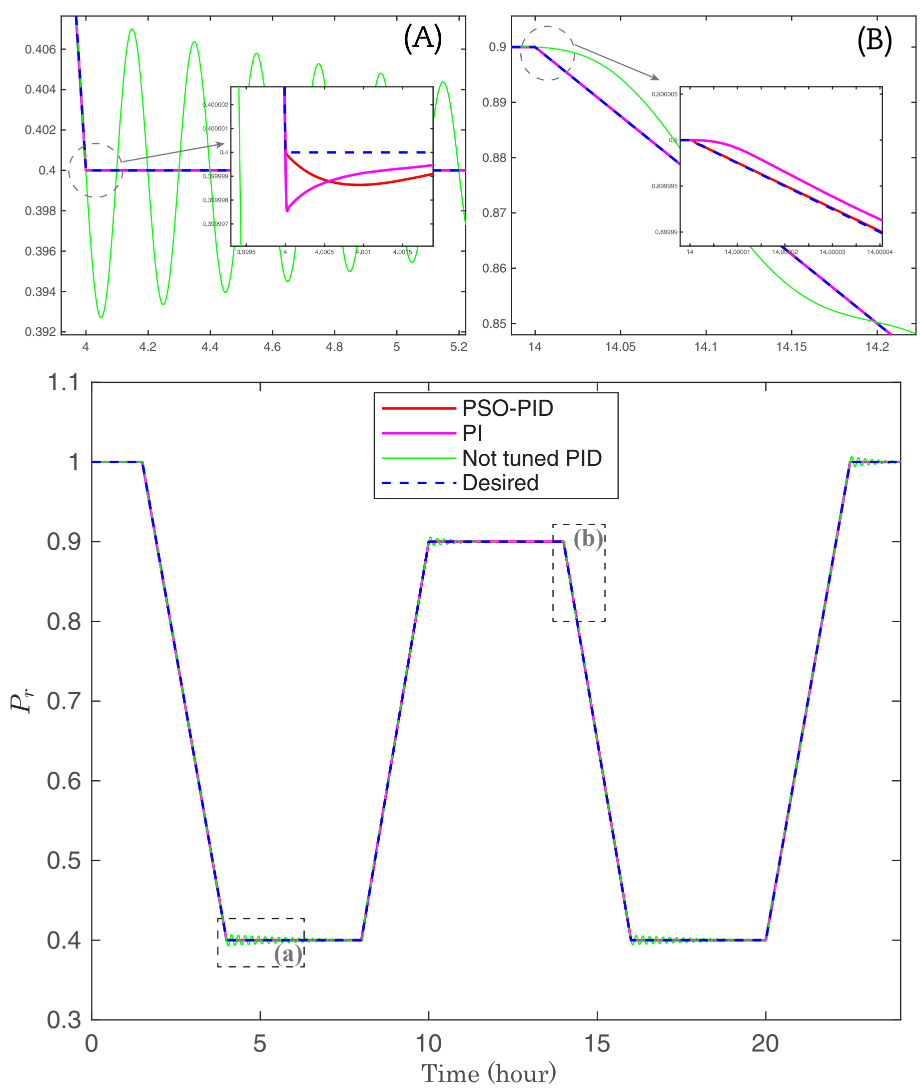

Fig. 6. Desired relative power (relative neutron density) with tuned/not tuned PID and PI controllers.
PID, proportional–integral–derivative; PSO, particle swarm optimization.

Experience shows that with increasing ω exploration increases, and its reduction leads to increase exploitation. If c_1 and c_2 are too large, it leads to increase exploration, and if they become too small, it leads to increase exploitation of the current responses. However, intermediate values help the exploitation of the best personal and global responses [27]. Confidence coefficients (ω, c_1, and c_2) are calculated according to only one parameter, ϕ, as below [28]:

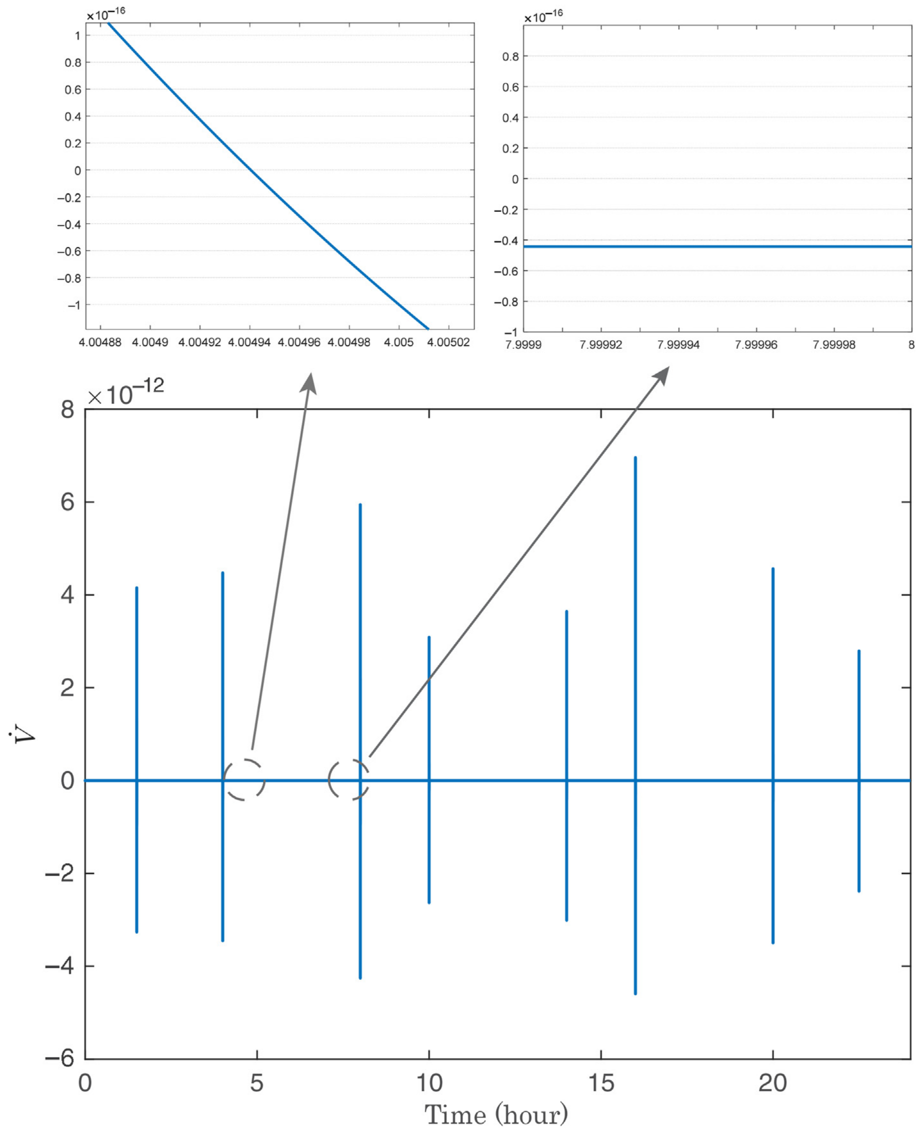

Fig. 7. The changes of $\dot{V}$ based on the Lyapunov analysis.

Table 3
The results of the simulation.

Region[a]	Region 1	Region 2	Region 3	Region 4	Region 5	Region 6	Region 7	Region 8
Time intervals (h)	1.5 – 4	4 – 8	8 – 10	10 – 14	14 – 16	16 – 20	20 – 22.5	22.5 – 24
K_P	14.82896	31.0973	21.38479	33.80283	24.42162	16.0208	30.07828	31.89282905
K_I	31.61285	68.3604	46.23613	73.25326	52.76062	34.53516	65.69102	68.34741336
K_D	4.325523	9.796275	6.825423	9.691284	7.061317	5.131498	9.524116	9.047942153
Overshoot/undershoot[b]	3.21×10^{-6}	1.37×10^{-7}	2.98×10^{-6}	1.51×10^{-6}	2.08×10^{-6}	3.97×10^{-6}	2.04×10^{-6}	1.52×10^{-6}
Settling time (s)	–	0	–	0	–	0	–	0
Rise time (s)	–	0	–	0	–	0	–	0
Stabilization time (s) (Lyapunov synthesis)	21.119	17.784	17.673	21.071	20.988	17.63	17.742	21.444
Final error (end of each time interval)	1.58×10^{-10}	8.57×10^{-11}	2.05×10^{-10}	6.54×10^{-11}	1.66×10^{-10}	9.77×10^{-11}	5.05×10^{-11}	1.39×10^{-9}
Best cost	21.11900321	17.78400014	17.67300298	21.07300151	20.98800208	17.63000397	17.74200204	21.44400152

In all regions: $\tau = 1/N = 0.01$.
[a] In the initial condition of the simulation (the first 1.5 h): $K_P = 39.63496$, $K_I = 1.092397$, $K_D = 0.035706$.
[b] Maximum output derivation from the desired signal in ramp mode.

$$\forall\ \phi_1, \phi_2 > 0 : \phi \triangleq \phi_1 + \phi_2 > 4 \tag{20}$$

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \tag{21}$$

χ is defined as a construction coefficient. When Clerc's constriction method is used, $\phi_1 = \phi_2 = 2.05$, and χ is about 0.7298. Accordingly, confidence coefficients are $\omega = \phi,\ c_1 = \chi\phi_1,\ c_2 = \chi\phi_2$ [29].

The pseudo-code is implemented in MATLAB script as displayed in Fig. 4. After passing a certain number of iterations, based on trial-and-error of the best cost convergence, the loop terminates.

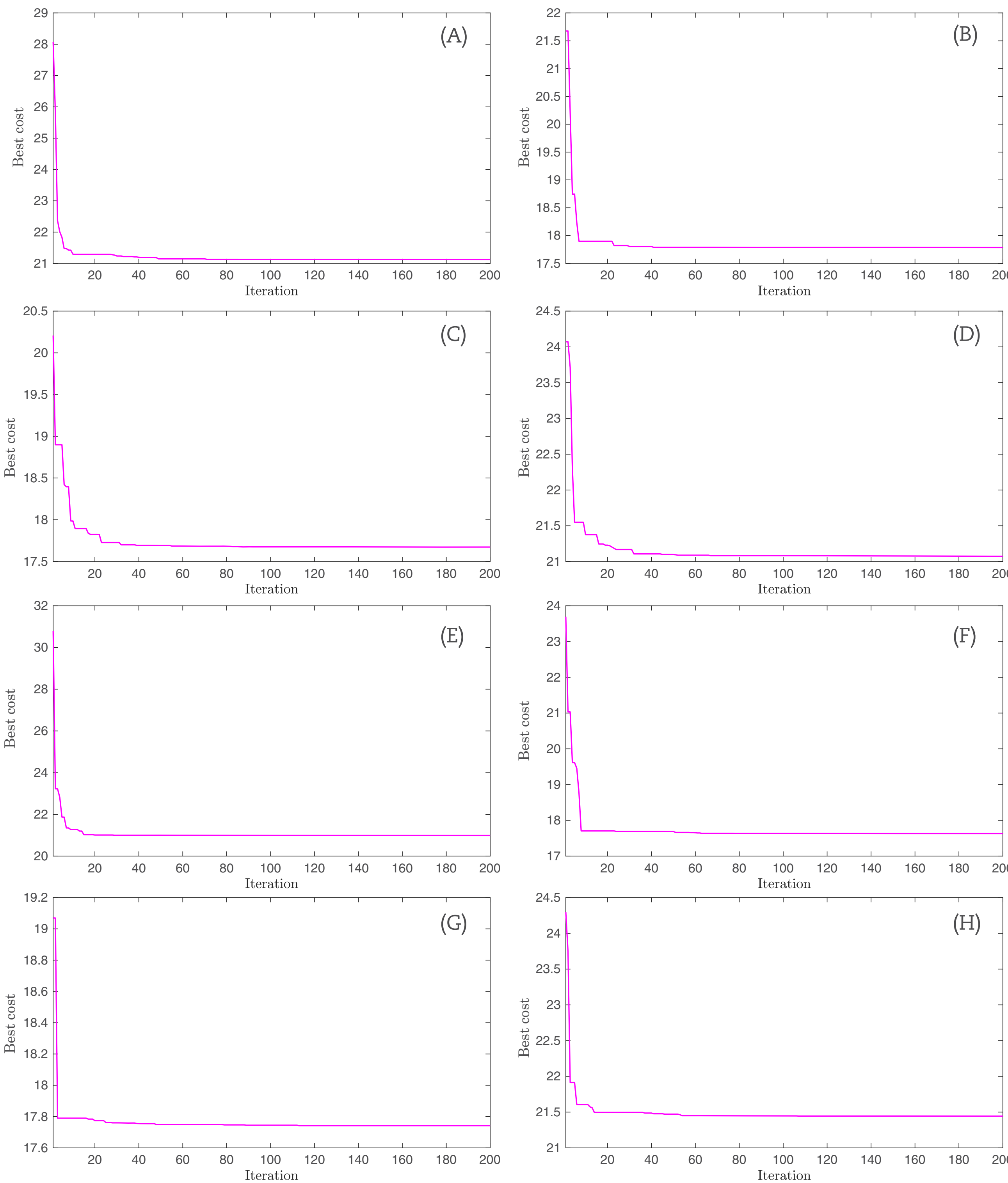

Fig. 8. The changes of the best costs versus the iterations. (A) Region 1. (B) Region 2. (C) Region 3. (D) Region 4. (E) Region 5. (F) Region 6. (G) Region 7. (H) Region 8.

2.4.2. Objective function definition

In PSO, an objective (cost) function needs to be defined to check the quality of particles in the population. Traditionally, the cost function is a weighted sum of the parameters of the decision variables, such as overshoot and settling time, which provided the steady-state stability. In this work, the stability condition has been added to the cost function, too. To this end, the stabilization time of the system in each time intervals has been considered based on Lyapunov synthesis (Eq. (17)). The objective (cost) function is defined as follows:

$$F = w_1 \times MP + w_2 \times TS + w_3 \times TSt \quad (22)$$

where w_i is weight of each factor; MP is overshoot/undershoot from last steady-state (maximum output derivation from the desired signal in ramp mode); TS is settling time; and TSt is stabilization time.

In the ground state, the coefficients of the weighted function are equal to one. In this work, the weight coefficients are considered the same.

2.4.3. PSO parameters

The regulated PSO parameters that have been used in the PSO scripts: members of each particle are K_P, K_I, and K_D; maximum iteration is 200; and population size (n_{pop}) is 30.

2.5. PSO–PID controller

The written PSO code is added to the PID controller system in accordance with Fig. 5, where $r(t)$ is desired signal (relative power); $y(t)$ is output signal; $e(t)$ is error signal between the input and the output signals; and $u(t)$ is control signal. First, PSO gets the instantaneous values of the intended inputs/outputs of the system, according to the parameters used in the objective (cost) function. Then, the control system delivers the best gains (K_P, K_I, K_D) in each iteration (generation). This process is repeated until the termination criterion of the PSO loop is satisfied.

The Lyapunov approach has no direct influence on the determination of feedback gains. The generated gains for each individual of the swarm are sent to the controller and the dynamic model (in the Simulink) in each time interval. The $\dot{V}$ signal (Eq. (17)) is delivered to the PSO script. The Lyapunov stability condition is searched from the end to the beginning of that time interval. The stabilization time is when the scalar value of the signal $\dot{V}$ is in threshold positive. The obtained time (TSt) is used in the objective function of the PSO (Eq. (22)). Therefore, theoretical stability analysis and linearization of state space equations are not required. So the real equations and outputs of the system are used. The linearization approximation of the equations leads to the reduction of the output accuracy. Actually, Lyapunov stability condition is checked by PSO script as a criterion.

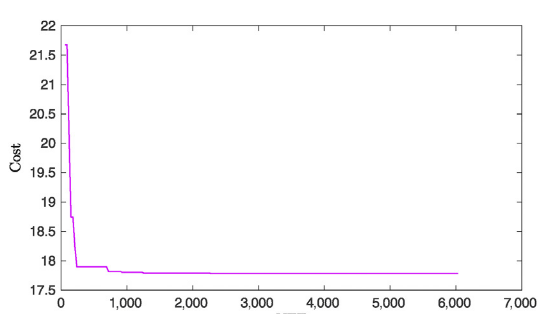

Fig. 9. The change of the best cost versus the NFE in the region 2. NFE, number of function evaluation.

3. Results and discussion

The proposed optimization system (PSO–PID) is applied to the load-following operation problem of a typical PWR. The trajectory

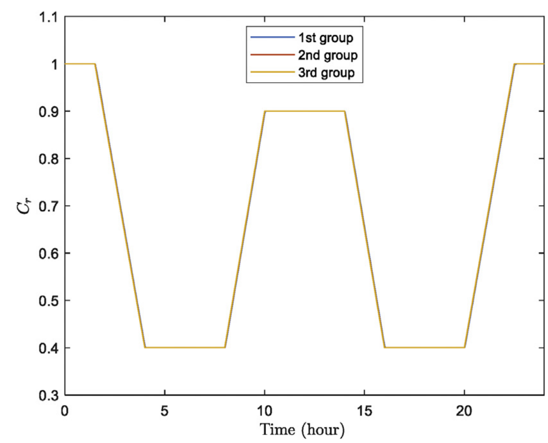

Fig. 10. Relative precursor density.

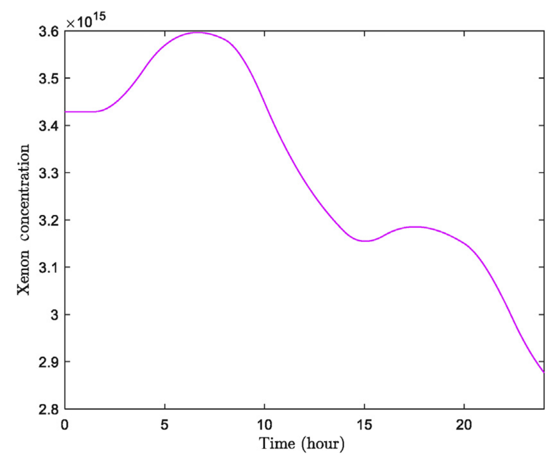

Fig. 11. Xenon concentration.

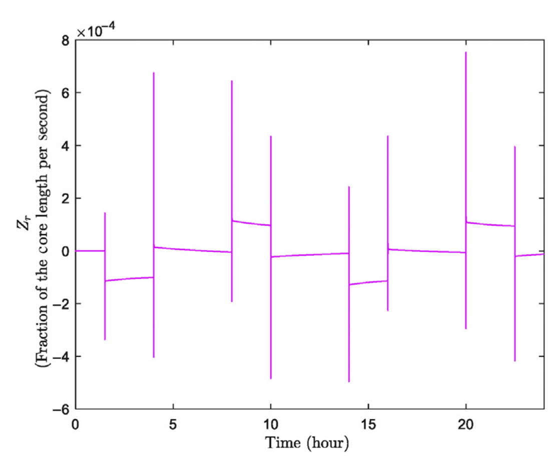

Fig. 12. The changes of the control rod speed (control signal).

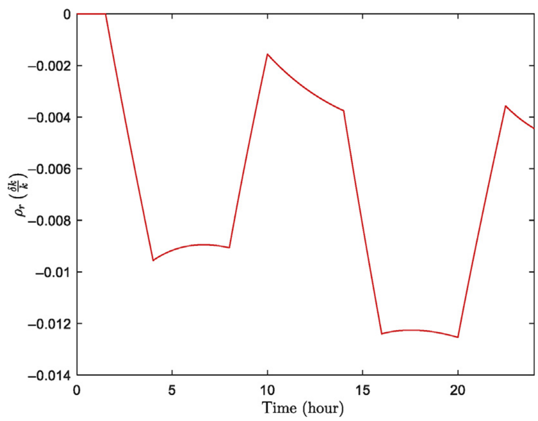

Fig. 13. Control rod reactivity.

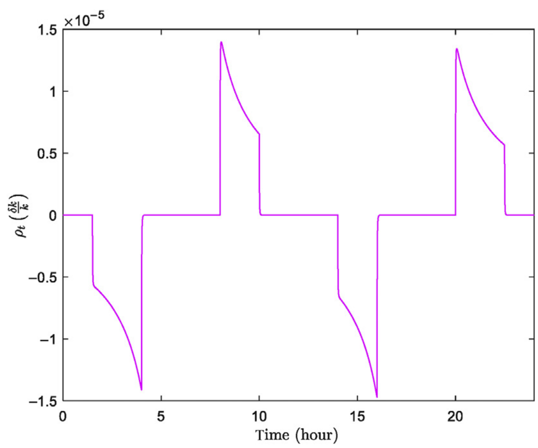

Fig. 14. Total reactivity of the core.

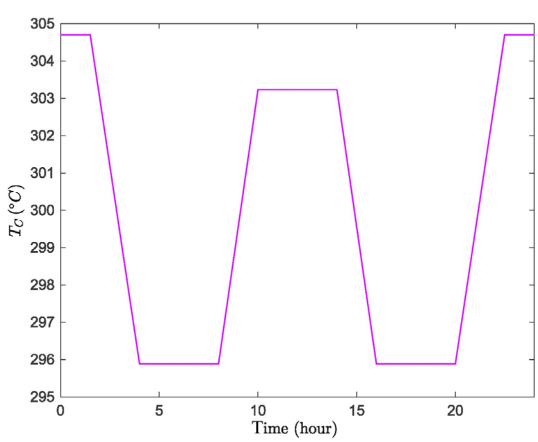

Fig. 15. Coolant temperature.

changes of the relative power following a ramp load is shown in Fig. 6 ($100\% \rightarrow 40\% \rightarrow 90\% \rightarrow 40\% \rightarrow 100\%$). The ramps are $-0.4\%/\text{min}$, $+\frac{5}{12}\%/\text{min}$, $-\frac{5}{12}\%/\text{min}$, and $+0.4\%/\text{min}$, respectively. This trajectory is divided into eight regions. For each region, a separate set of the PID controller gains is scheduled by the PSO–PID code, based on the PSO parameters considered in Section 2. Fig. 6 also shows the result of the tuned/not tuned PSO–PID and PI controllers. As it is seen, the closed-loop tuned controller output has a good agreement with the desired signal. Also, the PSO–PID controller has a better performance than PI controller. It should be noted that the relative power is equivalent to the relative neutron density/flux (Eq. (1)). Indeed, the neutron flux is controlled along with power control.

The changes of $\dot{V}$ over the time and the plant output stability, according to the Lyapunov stability analysis, are shown in Fig. 7. As it is seen, the stability of the PID controller is well established in a short time at the beginning of each region (in accordance with the Table 3 results).

According to the power demand (Fig. 6), the changes of the best costs in all regions over iterations are illustrated in Fig. 8. Curves of this figure show the convergence of the best cost to a constant value in each region.

One of the best criteria for comparing the performance of population-based and nature-inspired metaheuristics is to evaluate the best cost over the number of function evaluation (NFE) as shown in Fig. 9. In the region 2, for example, it is seen that the best cost is converged to a constant value. Figures of type 8 and 9 are criteria of the optimized response accuracy.

Fig. 10 shows the change of relative precursor density. As expected, their behavior is similar to relative power in the long-time transient.

The change of xenon concentration is shown in Fig. 11. The xenon build-up is well illustrated by the power downfall.

The control rod speed (control signal) is shown in Fig. 12. The range of changes is low. So, there is no practical problem in the hardware actuators.

The reactivity of control rod movement and total induced reactivity are shown in Figs. 13 and 14, respectively. These curves have predictable behaviors, and the control rod reactivity is within the proper ranges.

Fig. 15 shows the changes of coolant temperature. It illustrates that the coolant temperature is as well controlled as the power level.

4. Conclusion

In this paper, the popular PID controller was used for designing a PWR control system and tuned by PSO algorithm. The process is simple, and the nonlinear control system has been optimized without complex theory calculations. The objective (cost) function was considered as the weighted linear summation of decision variables including overshoot, settling time, and stabilization time (based on Lyapunov synthesis). The proposed PSO–PID system was tuned and scheduled the PID controller gains, with power demand. The simulation results reveal a good agreement between the desired signal and the closed-loop PID controller.

Conflict of interest

There is no conflict of interest.

References

[1] H.L. Akin, V. Altin, Rule-based fuzzy logic controller for a PWR-type nuclear power plant, IEEE Trans. Nucl. Sci. 38 (1991) 883–890.
[2] M.N. Khajavi, M.B. Menhaj, A.A. Suratgar, A neural network controller for load following operation of nuclear reactors, Ann. Nucl. Energy 29 (2002) 751–760.
[3] M.G. Na, D.W. Jung, S.H. Shin, J.W. Jang, K.B. Lee, Y.J. Lee, A model predictive controller for load-following operation of PWR reactors, IEEE Trans. Nucl. Sci. 52 (2005) 1009–1020.
[4] S.M.H. Mousakazemi, N. Ayoobian, G.R. Ansarifar, Control of the reactor core power in PWR using optimized PID controller with the real-coded GA, Ann. Nucl. Energy 118 (2018) 107–121.
[5] B.R. Upadhyaya, M.R. Lish, J.W. Hines, R.A. Tarver, Instrumentation and control strategies for an integral pressurized water reactor, Nucl. Eng. Technol. 47 (2015) 148–156.

[6] S. Bennett, The past of PID controllers, Annu. Rev. Control 25 (2001) 43–53.
[7] W.K. Ho, C.C. Hang, L.S. Cao, Tuning of PID controllers based on gain and phase margin specifications, Automatica 31 (1995) 497–502.
[8] K.H. Ang, G. Chong, Y. Li, PID control system analysis, design, and technology, IEEE Trans. Control Syst. Technol. 13 (2005) 559–576.
[9] K.J. Åström, T. Hägglund, Advanced PID Control, ISA-The Instrumentation, Systems and Automation Society, 2006.
[10] X. Li, D. Ruan, Comparative study of fuzzy control, PID control, and advanced fuzzy control for simulating a nuclear reactor operation, Int. J. Gen. Syst. 29 (2000) 263–279.
[11] C. Liu, J.-F. Peng, F.-Y. Zhao, C. Li, Design and optimization of fuzzy-PID controller for the nuclear reactor power control, Nucl. Eng. Des. 239 (2009) 2311–2316.
[12] J.-H. Ye, J.-M. Yi, H.-Y. Ji, Research on pressurizer water level control of nuclear reactor based on RBF neural network and PID controller, in: Mach. Learn. Cybern. (ICMLC), 2010 Int. Conf, IEEE, 2010, pp. 1486–1489.
[13] Z. Dong, PD power-level control design for PWRs: a physically-based approach, IEEE Trans. Nucl. Sci. 60 (2013) 3889–3898.
[14] J. Kennedy, R. Eberhart, Particle swarm optimization (PSO), in: Proc. IEEE Int. Conf. Neural Networks, Perth, Aust., 1995, pp. 1942–1948.
[15] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Evol. Comput. Proceedings, 1998. IEEE World Congr. Comput. Intell. 1998 IEEE Int. Conf, IEEE, 1998, pp. 69–73.
[16] A.A. de Moura Meneses, M.D. Machado, R. Schirru, Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor, Prog. Nucl. Energy 51 (2009) 319–326.
[17] C.M.N.A. Pereira, C.M.F. Lapa, A.C.A. Mol, A.F. Da Luz, A particle swarm optimization (PSO) approach for non-periodic preventive maintenance scheduling programming, Prog. Nucl. Energy 52 (2010) 710–714.
[18] S.S. Khorramabadi, M. Boroushaki, C. Lucas, Emotional learning based intelligent controller for a PWR nuclear reactor core during load following operation, Ann. Nucl. Energy 35 (2008) 2051–2058.
[19] D.L. Zhang, S.Z. Qiu, G.H. Su, C.L. Liu, L.B. Qian, Analysis on the neutron kinetics for a molten salt reactor, Prog. Nucl. Energy 51 (2009) 624–636.
[20] K. Torabi, O. Safarzadeh, A. Rahimi-Moghaddam, Robust control of the PWR core power using quantitative feedback theory, IEEE Trans. Nucl. Sci. 58 (2011) 258–266.
[21] G.R. Ansarifar, H.R. Akhavan, Sliding mode control design for a PWR nuclear reactor using sliding mode observer during load following operation, Ann. Nucl. Energy 75 (2015) 611–619.
[22] D.L. Hetrick, Dynamics of Nuclear Reactors, American Nuclear Society, 1993.
[23] P. Ramaswamy, M. Riese, R.M. Edwards, K.Y. Lee, Two approaches for automating the tuning process of fuzzy logic controllers [pwr application], in: Decis. Control. 1993., Proc. 32nd IEEE Conf, IEEE, 1993, pp. 1753–1758.
[24] T. Abhilash, A.P. Pavani, Multi area load frequency control of power system involving renewable and non-renewable energy sources, in: Power Adv. Comput. Technol. (I-PACT), 2017 Innov, IEEE, 2017, pp. 1–5.
[25] P. Cominos, N. Munro, PID controllers: recent tuning methods and design to specification, IEE Proceedings-Control Theory Appl. 149 (2002) 46–53.
[26] G.R. Ansarifar, M. Rafiei, Higher order sliding mode controller design for a research nuclear reactor considering the effect of xenon concentration during load following operation, Ann. Nucl. Energy 75 (2015) 728–735.
[27] J. Kennedy, Particle swarm optimization, in: Encycl. Mach. Learn., Springer, 2011, pp. 760–766.
[28] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (2002) 58–73.
[29] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intell. 1 (2007) 33–57.