Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 66 – 86

doi:10.3906/elk-1203-3

Optimal fuzzy load frequency controller with simultaneous auto-tuned membership functions and fuzzy control rules

Abas Ali ZAMANI*, **Ehsan BIJAMI**, **Farid SHEIKHOLESLAM**, **Bahram JAFRASTEH**
Department of Electrical Engineering, Isfahan University of Technology, Isfahan, Iran

Received: 01.03.2012 • **Accepted:** 04.10.2012 • **Published Online:** 20.12.2013 • **Printed:** 20.01.2014

Abstract: In this paper, an auto-tuned fuzzy load frequency controller (FLFC)-based artificial bee colony (ABC) algorithm is developed to quench the deviations in the frequency and tie-line power due to load disturbances in an interconnected power system. Optimal tuning of membership functions (MFs) and fuzzy control rules is very important to improve the design performance and achieve a satisfactory level of robustness for a particular operation. In this work, to reduce the fuzzy system design effort and take large parametric uncertainties into account, a new systematic and simultaneous tuning method is developed for designing MFs and fuzzy rules. For this, the designing problem is restructured as an optimization problem and the ABC algorithm is employed to solve it. This newly developed method provides some advantages such as a flexible controller with a simple structure and easy algorithm. For the purpose of the proposed method's evaluation, the designed controller is applied to a 2-area power system with considerations regarding governor saturation and the results are compared to the one obtained by a classic proportional-integral controller. Simulation results show better operation and improved system parameters, such as the settling time and step response rise time, using the proposed approach, in the presence of system parameter variations.

Key words: Fuzzy logic controller, artificial bee colony, load frequency control

1. Introduction

With the development of extensive power systems, especially with increasing size, the changing structure and complexity of these interconnected systems, load frequency control (LFC) has become an important criterion in electric power system design and operation and has received a great deal of attention [1]. An interconnected modern power system with commercial and industrial loads requires operation at a constant frequency with stable and reliable power. The fundamental goals of LFC in an interconnected power system are to hold a reasonably uniform frequency at each area and to maintain the tie-line power interchanges in a predefined tolerance in the presence of modeling uncertainties, system nonlinearities, area load disturbances, and sudden changes in load demands [2].

In the modern power system, as the power load demand changes randomly, the tie-line power interchange and area frequency also change. Therefore, a load frequency controller design is necessary to maintain the reliability of the electric power system and to provide better conditions for electricity trading and the power system's safe operation.

During the past decades, several control approaches have been proposed and applied to the LFC design

*Correspondence: a.zamani@ec.iut.ac.ir

problem, including optimal control, adaptive control, model predictive control, sliding mode control, and robust control, which can be found in [2–6], respectively. Each of these techniques has its own advantages and disadvantages.

More recently, there has been growing interest in artificial intelligence techniques, such as fuzzy logic control (FLC) [7–13], artificial neural network (ANN) [14,15], and biologically inspired algorithms [16–18], to design a load frequency controller in a power system by research around the world. Among these techniques, since fuzzy logic controllers provide an effective means to model and control a complex and ill-defined plant, many control strategies based on the FLC approach have been suggested and applied successfully regarding the load-frequency control of power systems. These intelligent control methods are independent of the power system's mathematical model parameters. For instance, a fuzzy logic-based tie-line bias control scheme was introduced in [7]. The authors proposed a fuzzy gain-scheduled proportional-integral controller and its implementation on an Iraqi national super grid power system in [8]. A new optimal fuzzy logic-based proportional-integral-derivative controller is presented in [9] to deal with the LFC problem. Moreover, some authors used fuzzy proportional-integral-derivative methods to solve the LFC problem [10,11].

In designing a fuzzy load frequency controller (FLFC), one will need to determine components such as membership functions (MFs) and fuzzy control rules. In order to improve the performance of the FLFC and to reduce fuzzy system effort cost and time consumption, fuzzy logic controllers combined with a genetic algorithm (GA) based on the hill climbing method and partial swarm optimization (PSO) algorithm were proposed in [12] and [13], respectively. In these strategies, the authors used a GA and PSO for the tuning of fuzzy MFs through multistage procedures. Although these algorithms may be good approaches to solve some optimization problems, the existence of a greater number of local optimum in some problems causes premature convergence and leads these algorithms to be trapped in local optimum easily. Another important factor in such methods is the coding method and optimization problem formulation. This issue may take a long simulation time to obtain the solution.

In this paper, to overcome these drawbacks and improve the performance of the FLC, an efficient artificial bee colony (ABC)-based approach is suggested for the simultaneous auto-tuning of MFs and fuzzy control rules of a FLFC by considering a new optimization formulation. The ABC algorithm is a robust search and optimization technique that has been applied in many practical researches and has proven its superior capabilities, such as faster convergence and better global minimum achievement.

In the proposed method, during the evolutionary process, the parameters to be optimized by the ABC algorithm are the centers and widths of the triangular MFs, and the fuzzy rules corresponding to every combination of the input linguistic variables. This is implemented by a new type of coding and a new way of formulating the optimization problem.

To show the efficiency of the proposed approach, it is used as an alternative approach in solving the LFC problem on a 2-area interconnected power system with considerations regarding the governor saturation and a wide range of parametric uncertainties. The system parametric uncertainties are obtained by changing the parameters by 25% simultaneously from their nominal values. Moreover, to make a comparison, the proposed controller is compared with 2 different controllers: a conventional proportional-integral (PI) controller and a FLFC based on the PSO algorithm. The simulation results show the superiority and capability of the proposed method in improving system parameters, such as the settling time and step response rise time, in the presence of system parameter variations.

The paper is organized as follows: to establish a proper background, the basic concepts of the ABC algorithm are briefly described in Section 2. The study system used in the simulations studies is given in Section 3. In Section 4, the ABC-based FLFC scheme is presented. Simulation results in the study system are provided in Section 5 and some conclusions are drawn in Section 6.

2. Overview of the ABC algorithm

Over the last decades, there has been growing interest in algorithms inspired by the observation of natural phenomena. It has been shown by many researches that these algorithms are good alternative tools to solve complex computational problems.

The ABC algorithm is a population-based stochastic optimization algorithm inspired from the particular intelligent behavior of honeybee swarms when searching for a food source and it has proven its superior capabilities, such as faster convergence and better global minimum achievement [19].

The ABC algorithm, originally developed by Karaboga in 2005 [20], simulates the foraging behavior of a bee colony. Due to the advantages of the ABC algorithm, such as its simple concept, easy implementation, and fewer control parameters, it has been researched and utilized to solve different kinds of optimization problems by researchers around the world since 2005, such as data clustering [21,22], training ANNs [23], the leaf-constrained minimum spanning tree problem [24], designing infinite impulse response filters [25], and designing the optimal parameters of a power system stabilizer [26].

In the ABC algorithm, the colony of artificial honey bees consists of 3 types of bees: employed bees, onlookers, and scouts; half of the colony are employed bees and the remaining are onlooker bees [20]. Each solution of the optimization problem is called a food source in the search space. In other words, the searching process of bees for the food source stands for the finding process for the optimum solution of the problem to be optimized. The fitness of solution corresponds to the profitability of the food source. Moreover, the related profitability (fitness) of a food source (solution) is calculated by the evaluation of the fitness function of the corresponding variables considering the related objective function. The number of solutions is known as SN and is considered to be equal to the number of employed bees or onlookers. The employed bee whose food source is abandoned by the bees becomes a scout [19,20].

The main procedures of the ABC algorithm can be written as follows:

Step 1: Initialize the population.

Step 2: Send the employed bees onto their food sources and evaluate their nectar amounts (fitness).

Step3: Place each onlooker bee on a food source according to its nectar amounts (quality of her solution), based on the information provided by the employed bees

Step4: Determine the source to be abandoned and assign its employed bee as a scout for searching the area to discover new food sources.

Step5: Memorize the best food source (solution) found so far.

Step6: Until the termination criterion is satisfied, repeat steps 2–5.

Similar to other evolutionary algorithms, this algorithm begins with an initial population of SN food source positions created randomly within the feasible space. Each food source corresponds to a solution in the search space. For D-dimensional problems (D variables), the position of the ith food source is represented as $X_i = [x_{i1}, x_{i2}, ..., x_{iD}]^T$. The initial population of artificial bees is generated randomly within the range of the boundaries of the parameters, as follows:

$$X_{ij} = X_j^{\min} + rand(0,1)(X_j^{\max} - X_j^{\min}), \tag{1}$$

where $i \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., D\}$, in which D is the number of optimization parameters, and $X_j^{\min}$ and $X_j^{\max}$ are the lower bound and upper bound of parameter j, respectively. Thus, *rand ()* is a random number in the range of [0, 1].

After initialization, all of the food sources (solutions) are subjected to repeat cycles of the search processes of the honeybees. The search process is continued until the termination criterion is met. The termination criterion could be the maximum cycle number (MCN) or when an error tolerance (ε) is met [21].

In step 2, the employed bees produce a modification on the position of the food sources (solutions) in their memories, depending on the local information (visual information), and produce new food source positions (new solutions), V_{ij}, in the neighborhood of the old food source positions (old solutions), x_{ij}, using the following equation:

$$V_{ij} = X_{ij} + r_{ij}(X_{ij} - X_{kj}), \tag{2}$$

where $j \in \{1, 2, ..., D\}$ is a random integer in the interval [1, D] and $k \in \{1, 2, ..., SN\}$ is a randomly chosen index, where $k \neq i$. Moreover, r_{ij} is a uniformly distributed real random number in the range of [–1, 1]. It is an adaptively control parameter that controls the production of neighbor food sources around X_{ij} and determines the comparison of 2 food positions visually by a bee. As can be seen from Eq. (2), as the difference between X_{ij} and X_{kj} is reduced, the perturbation on the position X_{ij} is decreased as well. Thus, as the search finds a better solution, the step length is steadily decreased.

If this repositioning process produces sources (solutions) with higher nectar amounts (better fitness) than those of the previous ones, the bees replace the position of the new sources with the previous ones. Otherwise, they keep the position of the previous food sources in their memories.

In step 3, after all of the employed bees complete their search process, they communicate their information related to the nectar amounts (fitness') and the positions of their food sources (solutions) to the onlooker bees. Next, the onlooker bees calculate the nectar information taken from all of the employed bees and select food sources using a selection probability that depends on the fitness values of the solutions in the population. As the fitness of the solution increases, the probability of that solution being chosen also increases [27].

This probabilistic selection scheme might be a roulette wheel, stochastic universal, rank selection, disruptive selection, tournament selection, or another selection method. The basic ABC algorithm uses the roulette wheel selection mechanism, in which the probability value associated with a food source, P_i, can be expressed by the following expression:

$$P_i = \frac{fitness_i}{\sum\limits_{i=1}^{SN} fitness_i}, \tag{3}$$

where *fitness* $_i$ is the fitness value of the ith food source (solution) and is proportional to the nectar amount of the food source in the ith position. Moreover, *SN* denotes the number of food sources, which is equal to the number of employed or onlooker bees.

After selecting the food source, as in the case of the employed bee, onlookers start to carry out the exploitation process and produce some modifications on the positions in their memories using Eq. (2). Here, new positions V_{ij} are produced for the onlookers from the solutions X_{ij}, selected depending on P_i. Once again, if this repositioning process produces food sources (solutions) with higher nectar amounts (better fitness) than those of the previous ones, the bees replace the position of the new sources with the previous ones. Otherwise they keep the position of the previous food sources in their memories [20,28].

In the ABC algorithm, there is a control parameter called the '*limit*' for abandonment. The *limit* is a predetermined number of cycles that controls the update times of a certain solution and is used to determine if there is any exhausted source to be abandoned. For this, after all of the employed and onlooker bees complete their searches, the algorithm checks the counter value, which has been updated during search. If the value of the counter is greater than the *limit* value and no improvement is possible in the food source position, then the source associated with this counter is assumed to be abandoned and the employed bee becomes a scout. The scout then starts to search for a new food source to be replaced with the abandoned one.

This is simulated by generating a site position randomly and replacing it with the abandoned one. If the abandoned source is Z_i, then the scout randomly discovers a new food source to be replaced with Z_i. This operation can be expressed as in Eq. (4). In the basic ABC, it is assumed that only 1 source can be exhausted in each cycle, and only 1 employed bee can be a scout [23].

These steps are repeated through the MCN or until a termination criterion is satisfied.

$$Z_i^j = Z_{\min}^j + rand(0,1)(Z_{\max}^j - Z_{\min}^j) \tag{4}$$

Figure 1 shows the detailed pseudocode for the ABC algorithm.

1) Initialize the population of solutions X_{ij} by using (1)
2) Evaluate the fitness of population
3) Set cycle to 1
4) **Repeat**
5) **FOR** each employed bee {
   Produce new solutions V_{ij} in the neighbourhood of X_{ij}, using (2)
   Evaluate the fitness's value
   Apply the selection process between X_{ij} and V_{ij}
   Calculate the possibility value, P_i, using (3)}
6) **FOR** each onlooker bee {
   Select a solution X_{ij} depending on P_i
   Produce new solutions V_{ij} from the solutions X_{ij}
   Evaluate the fitness's value}
7) Determine the abandoned solution X_{ij}, if exists, replace it with a new randomly produced solution X_{ij} for the scout bee, using (4)
8) Memorize the best solution achieved so far
9) Cycle=cycle + 1
10) **Until** cycle=MCN

Figure 1. Pseudocode of the ABC algorithm.

3. Power system model

A typical power system naturally composed of complex and multivariable structures with different interconnected control areas, where, in each area, generators are assumed to constitute a coherent group. All of the control areas are generally nonlinear, time-variant, and/or nonminimum phase systems with complicated characteristics that connect to each other using tie lines. These tie lines are employed to exchange the power between areas and enhance the fault tolerance of the entire power system in the case of abnormal conditions.

In actual power system operations, the load varies randomly and continuously throughout the day. As a result, both the frequencies in all of the areas and the tie-line power flow between the areas are affected by these load changes at the operating point. These changes create a mismatch between the generations and the demand, where a result in exact forecast of the real power demand cannot be assured. Therefore, for good and stable power system operation, both the frequency and tie-line power flow should be kept constant against the sudden area load perturbations, system parameter uncertainties, and unknown external disturbances. Therefore, to ensure the quality of the power supply, a load frequency controller is needed to restore the system frequency and the net interchanges to their desired values for each control area.

The area frequency deviation (Δf) and tie-line power deviation (ΔP_{tie}) are 2 important parameters of interest. Their linear combination is known as the area control error (ACE). The measurements of the entire generation and load in the system, for the computation of the mismatch between the generation and obligation, in one area are very difficult. The mismatch is measured at the area control center using the ACE. The ACE for the ith area is defined as:

$$\begin{aligned} ACE_i &= P_{tie_i}^{act} - P_{tie_i}^{s} - 10B_i(f_i^{act} - f_i^{s}) \\ &= \Delta P_{tie_i} - 10B_i \Delta f_i \end{aligned}, \tag{5}$$

where $P_{tie_i}^{act}$ and $P_{tie_i}^{s}$ are the actual and scheduled (manually set) interchanges of the ith area with neighboring areas, respectively. Moreover, f_i^{act} and f_i^{s} are the area's actual and scheduled frequencies in the ith area, and B is the frequency bias coefficient of the ith area, which is a negative number measured in MW per 0.1 Hz.

However, the ACE signal is often calculated using the area frequency response characteristic β instead of B, as follows:

$$ACE_i = \Delta P_{tiei} + \beta_i \Delta f_i \tag{6}$$

with

$$\beta_i = \frac{1}{R_i} + D_i. \tag{7}$$

In which D_i is the damping ratio or the frequency sensitivity of the ith area's load and R_i is the regulation due to the governor action in the ith area, or droop characteristic. Moreover, β_i is a frequency bias constant and should be high enough such that each area adequately contributes to the frequency control [10].

The frequency and interchanged power are kept at their desired values by means of the feedback of the ACE, containing the deviation in the frequency and error in the tie-line power, and controlling the prime movers of generators. The main objective of the control system is to damp these variations to 0 as fast and as smoothly as possible following a change in the load demand values.

A 2-area interconnected power system, while considering governor limiters, is investigated in this study. Each area consists of 3 major components, which are the turbine, governor, and generator. The detailed transfer function block diagram of the uncontrolled 2-area system is shown in Figure 2, where Δf_1 and Δf_2 are the frequency deviations in area 1 and area 2, respectively, in hertz; and ΔP_{L1} and ΔP_{L2} are the load demand changes in areas 1 and 2, respectively, in per unit (p.u.). Moreover, T_{gi}, T_{ti}, and M_i are the speed governor time constant(s), turbine time constant(s), and power system time constant(s) of the ith area, respectively. The detailed transfer function models of the speed governors and turbines were discussed in [1]. Typical data for the system parameters and governor limiters, for nominal operation conditions, are adopted from [10] and presented in Table 1.

The state-space model of a foregoing power system can be modeled as a multivariable system, as the following equation:

$$\dot{x} = Ax(t) + Bu(t) + \Gamma d, \tag{8}$$

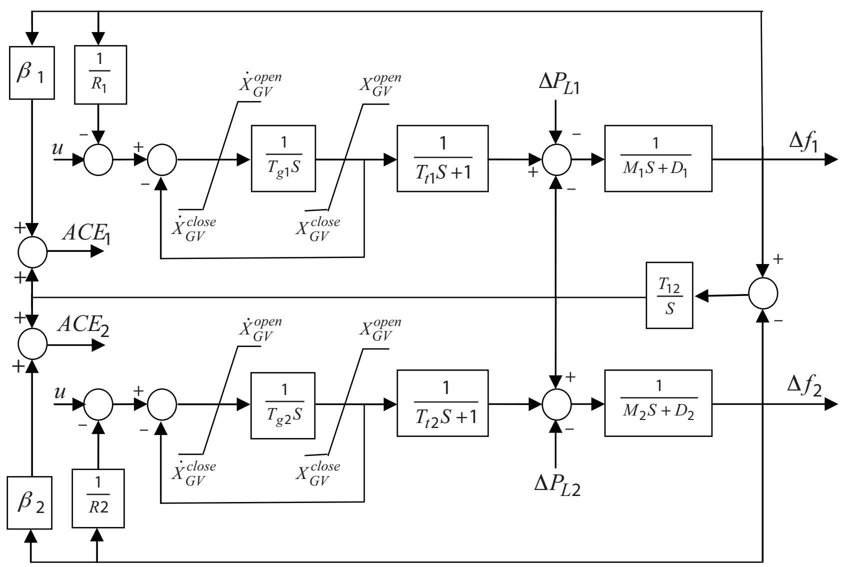

Figure 2. Two-area interconnected power system.

where $x(t)$, $u(t)$, and d are the state, control, and load change disturbance vectors, respectively, and are represented as following form:

$$\begin{aligned} u &= [\Delta P_{ref1} \Delta P_{ref2}] \\ d &= [\Delta P_{L1} \Delta P_{L2}] \\ x &= [\Delta P_{v1} \Delta P_{m1} \Delta\omega_1 \Delta P_{Tie} \Delta P_{v2} \Delta P_{m2} \Delta\omega_2 \Delta E_1 \Delta E_2] \end{aligned} \tag{9}$$

Moreover, A, B, and Γ are given in Eq. (10) and are, respectively, the system state, control input, and disturbance constant matrices of the appropriate dimensions associated with the above vectors. The corresponding coefficient matrices are given in Table 1.

$$A = \begin{bmatrix} \frac{-1}{\tau_{g1}} & 0 & \frac{-1}{R_1\tau_{g1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\tau_{T1}} & \frac{-1}{\tau_{T1}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{M_1} & \frac{-D_1}{M_1} & \frac{-1}{M_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & T_{12} & 0 & 0 & 0 & -T_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau_{g2}} & 0 & \frac{-1}{R_2\tau_{g2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\tau_{T2}} & \frac{-1}{\tau_{T2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{M_2} & 0 & \frac{-1}{M_2} & \frac{-D_2}{M_2} & 0 & 0 \\ 0 & 0 & B_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & B_2 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{\tau_{g1}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\tau_{g2}} & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{10}$$

To provide a reasonable dynamic performance for the system, 2 load frequency controllers are designed using the proposed fuzzy-based ABC algorithm approach. The implementation of the proposed method to design the load frequency controllers is given below.

Table 1. The 2-area interconnected power system parameters.

Area	Parameters
Area 1	M = 10, D_1 = 0.8, T_g = 0.2, T_t = 0.5, R_1 = 0.05, $\dot{X}_{GV}^{open}=0.4$, $\dot{X}_{GV}^{close}=1.5$, $X_{GV}^{open}=1.2$, $X_{GV}^{close}=0.4$, T_{12} = 2
Area 2	M = 8, D_2 = 0.9, T_g = 0.3, T_t = 0.6, R_2 = 0.0625, $\dot{X}_{GV}^{open}=0.4$, $\dot{X}_{GV}^{close}=1.5$, $X_{GV}^{open}=1.2$, $X_{GV}^{close}=0.4$, T_{12} = 2

4. FLFC based on the ABC algorithm

Because of the nonlinear and complex characteristic and multivariable conditions of modern power systems, classical control approaches may not give satisfactory solutions. Their robustness and reliability make fuzzy-based controllers useful for solving a wide range of control problems in power systems. In this paper, a new auto-tuned fuzzy controller-based ABC optimization algorithm is suggested for the solution of the LFC problem. In the following, the architecture of the proposed FLFC and its optimization method based on the ABC algorithm are described.

4.1. FLFC structure

The basic structure of the fuzzy logic controller includes 4 principal components: a fuzzifier, inference system, knowledge base, and defuzzifier. First, the fuzzifier translates its input signals (real values) to fuzzy numbers (fuzzy values). These numbers are the input of the inference system, which applies a fuzzy reasoning mechanism to calculate the fuzzy number of the controlled output signal by taking the appropriate decisions. The knowledge base comprises the fuzzy rule and MF sets known as the rule table. Finally, the defuzzifier converts a set of modified control outputs into a nonfuzzy control action. The basic structure of the FLFC is shown in Figure 3.

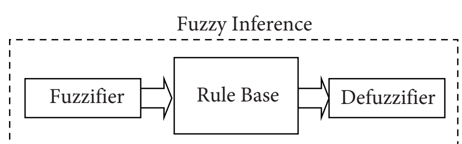

Figure 3. Basic structure of the fuzzy controller.

As stated before, the extraction of an appropriate set of MFs and proper fuzzy rules from a human expert is computationally expensive and time-consuming. Moreover, a proper fuzzy rule base design and a good tuning of FLFC's parameters (the centers and the widths of the triangular MFs in the inputs and output) is an important and key factor to achieve a satisfactory level of control performance for a particular operation.

In most of the previous works, the interdependence between MFs and fuzzy control rule sets is not considered. In these methods, the fuzzy control rules are designed for specific MFs only. However, it is possible that other fuzzy control rules and different MFs will be more appropriate for the specific process. Hence, it is required that the MFs and fuzzy rules must be tuned simultaneously. Recently, the application of metaheuristic algorithms to fuzzy logic controllers has held a great deal of attention in overcoming 2 of the major problems in fuzzy controller design, the design time and design optimality.

In this paper, an auto-tuned method-based ABC algorithm is developed for the optimal tuning of MFs and fuzzy control rules, simultaneously. For this, the designing problem is restructured as an optimization problem and the ABC algorithm is employed to solve it.

To incorporate the ABC algorithm into the problem of designing a FLFC, the MFs' parameters and fuzzy rule sets must be coded to form the solution array and a cost function must be defined in such a manner that the design criteria are satisfied through minimizing it. Here we developed a new encoding and decoding scheme for rule extraction, MF adjustment, and training the fuzzy rule base. This coding approach significantly minimizes the length of the solution array and makes possible simultaneous tuning parameters.

4.2. The encoding and decoding procedures

In the proposed encoding approach, only the rules that participate in the fuzzy model are encoded. This method has 2 significant advantages; first, to decrease the size of the solution to be found by the ABC, and, second, to increase the interpretability due to the reduction in the number of rules.

In the proposed coding method, an index (a positive integer number) is allocated to every possible rule. In the first step, each linguistic term is converted into a T-based number, where T is defined as the following form:

$$T = \max(m_i)|_{i=1}^{n+1}, \tag{11}$$

where T and m_i are the maximum number of linguistic terms and the number of linguistic terms of the ith linguistic variable, respectively. In this translation, 0 is assigned to the lower linguistic term, 1 is assigned to the next one, and so on. The second step transforms the T-based numbers (which represent the linguistic term's codes) to positively signed decimal integers representing the index of the fuzzy rule as follows:

$$I = \sum_{i=1}^{n+1} L_i T^{(n+1-i)}, \tag{12}$$

where I is the index of the respective fuzzy rule (in base 10) and the fuzzy linguistic term index is given by $L_i \in \{0, 1, ..., m_i - 1\}$.

Decoding the rule is the reverse of this process, in which I is converted back into (n + 1) and the T-based numbers are calculated from Eq. (13):

$$L_{(n+2-i)} = [I.T^{-(i-1)}] - T[\frac{[I.T^{-(i-1)}]}{T}]|_{i=1}^{n+1}, \tag{13}$$

where [.] denotes that only the integer part of the operation result is taken. To have better clarity of the presented coding and decoding procedures, a numerical example is presented by considering a simple fuzzy rule.

Consider a system with 3 inputs (linguistic variable, $n = 3$), X_1, X_2, X_3 and 1 output U, where X_1 consists of 3 linguistic terms (i.e. $m_1 = 3$): {N, ZE, P} , which are described by the indices {0, 1, 2} , respectively. Moreover, X_2 and X_3 are composed of 5 linguistic terms ($m_2 = m_3 = 5$): {NM, NS, ZE, PS, PM} that are represented by the indices {0, 1, 2, 3, 4} , respectively, and finally, U has 7 linguistic terms {NB, NM, NS, ZE, PS, PM, PB} that are described by the indices {0, 1, 2, 3, 4, 5, 6} , respectively. Now, consider the following simple fuzzy rule:

$$IF X_1 is N and X_2 is NS and X_3 is PM THEN U is PM.$$

First, the codes of the employed linguistic terms are calculated, describing a specific rule. For the above considered example, the rule is described by the following indices: L = {0, 1, 4, 5} . Note that the '0' describing the fuzzy linguistic term associated with the 1st linguistic variable refers to 'N', and the '1' associated with the 2nd linguistic variable refers to 'NS', and so on.

After selecting the base for the coding procedure, employing Eq. (11) yields: T = max {3, 5, 5, 7} = 7. Using Eq. (12), we generate the decimal integer associated with the considered fuzzy rule, I = $[(0 \times 7^3) + (1 \times 7^2) + (4 \times 7^1) + (5 \times 7^0)] = 82$.

Now we want to change the obtained integer, 82, to the corresponding rule. By applying Eq. (13), the decoding procedure can be shown to create the following outputs, which gives $L = \{0\ ,\ 1\ ,\ 4\ ,\ 5\}$:

$$i = 1 : L_4 = [82 * 7^0] - 7 * [\frac{[82 * 7^0]}{7}] = 5,$$

$$i = 2 : L_3 = [82 * 7^{-1}] - 7 * [\frac{[82 * 7^{-1}]}{7}] = 4,$$

$$i = 3 : L_2 = [82 * 7^{-2}] - 7 * [\frac{[82 * 7^{-2}]}{7}] = 1,$$

$$i = 4 : L_1 = [82 * 7^{-3}] - 7 * [\frac{[82 * 7^{-3}]}{7}] = 0.$$

Now we use the suggested coding and decoding methods to find the optimal parameters of the FLFC for a 2-area power system, which is shown in Figure 2.

There are various MFs used in FLC, such as Gaussian, trapezoidal, or triangular functions, which are defined by 2, 4 and 3 parameters, respectively. However, piecewise linear MFs are preferred because of their simplicity and efficiency with respect to computability.

In this paper, for having variations in the centers and widths of the MF while keeping the dimension of the problem low, 2 trapezoidal memberships and 3 triangular memberships with 5 variables of $X_1, X_2,\ X_3,\ X_4$, and X_5 are considered as the MFs, as depicted in Figure 4, where $X_i,\ I = 1, \ldots, 5$ are the MFs' parameters that must be tuned. On the other hand, the MFs of the inputs and outputs of the proposed fuzzy controller are tuned by 5 parameters ($X_1, X_2,\ X_3,\ X_4$, and X_5). These parameters are obtained using the ABC optimization method.

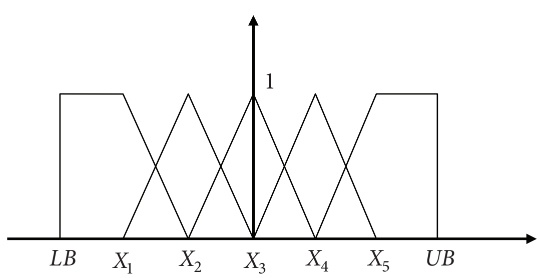

Figure 4. Considered MF for the coding process.

Now, for us to be able to use the ABC algorithm to optimize the MF, both the boundaries and limitations of each parameter should be defined. According to the considered variables in Figure 4, the limitations of the parameters can be expressed as the following equations:

$$\begin{aligned} & X_2 - X_1 \geq 0 \\ & X_3 - X_2 \geq 0 \\ & X_4 - X_3 \geq 0 \\ & X_5 - X_4 \geq 0 \\ & LB < X_1, X_2, X_3, X_4, X_5 < UB \end{aligned} \quad . \tag{14}$$

In Eq. (14), UB and LB are the upper and lower boundaries of each variable and are set as 1 and –1, respectively. In the case of the equations not being in a true condition (e.g., $X_{k+1} - X_k < 0$), then X_{k+1} is obtained using the following equation:

$$X_{k+1} = rand(X_k, UB). \tag{15}$$

Each FLFC designed here has 2 inputs, the ACE and ACE deviation $\left({}^{d}/_{dt}(ACE) \right)$. The output of the fuzzy controller (U) is the control input to each area. In this study, each of the inputs and the output of fuzzy controller consist of 2 trapezoidal memberships and 3 triangular memberships, as illustrated in Figure 4. The FLC inputs are composed of 5 linguistic terms: negative big (*NB*), negative small (*NS*), zero (*ZE*), positive big (*PB*), and positive small (*PS*). Furthermore, the FLC output is partitioned into the same 5 fuzzy sets. The fuzzy inference is carried out using Mamdani's method and the defuzzification employs the center of gravity to calculate the output of this FLFC.

The parameters to be optimized by the ABC algorithm are the centers and widths of the triangular MFs, and the fuzzy rules corresponding to every combination of the input linguistic variables. For the simultaneous adjustments to the MFs and rules table in each fuzzy controller design, each of the solutions in each area should contain the rules and the MFs of the 2 fuzzy controllers.

A basic code structure for the ABC is shown in Figure 5. Since we have 1 FLFC for each of the 2 areas and each FLFC has 15 MFs (each of the FLFC's inputs and output have 5 triangular MFs, as shown in Figure 4) and 25 rules, there are a total of 80 parameters to be optimized in this study. In Figure 5, $I_{iz}(i = 1,..., 25)$ and $X_{kz}^{j}(j = 1, 2, 3$ and $k = 1,.., 5)$ are the fuzzy rule and MF parameters, respectively. Moreover, $z = \{a,\ b\}$, in which indices a and b stand for the fuzzy controller parameters for areas 1 and 2, respectively. For example, X_{1a}^{1} means the 1st parameters of the MF of the 1st FLFC's input (i.e. the ACE) in area 1 and X_{3b}^{2} means the 3rd parameter of the MF of the 2nd FLFC's input (i.e. ${}^{d}/_{dt}(ACE)$) in area 2.

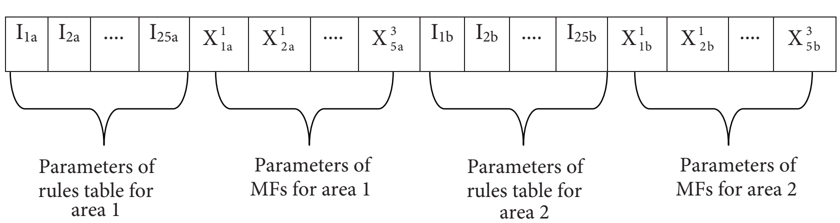

Figure 5. A basic code structure for the ABC.

The ABC searches all of the antecedent and consequent parameters in 80 dimensional spaces using a cost function, as stated in Eq. (16):

$$J_c = \int_{t=0}^{t=t_f} t(C_1 |\Delta P_{tie}| + C_2 |\Delta F_1| + C_3 |\Delta F_2|)dt, \tag{16}$$

where $C_1 = C_2 = C_3 = 10$.

5. Simulation results and time domain analysis

In this study, a 2-area interconnected power system, as shown in Figure 2, is adopted for simulation studies. The simulation studies are implemented in the MATLAB/Simulink 7 software and executed on an Intel Core2Duo 2.32-GHz personal computer with 4-GB RAM. The structure of the control system is given in Figure 6. As can be seen, the FLFC is optimized and tuned using an ABC block optimizer.

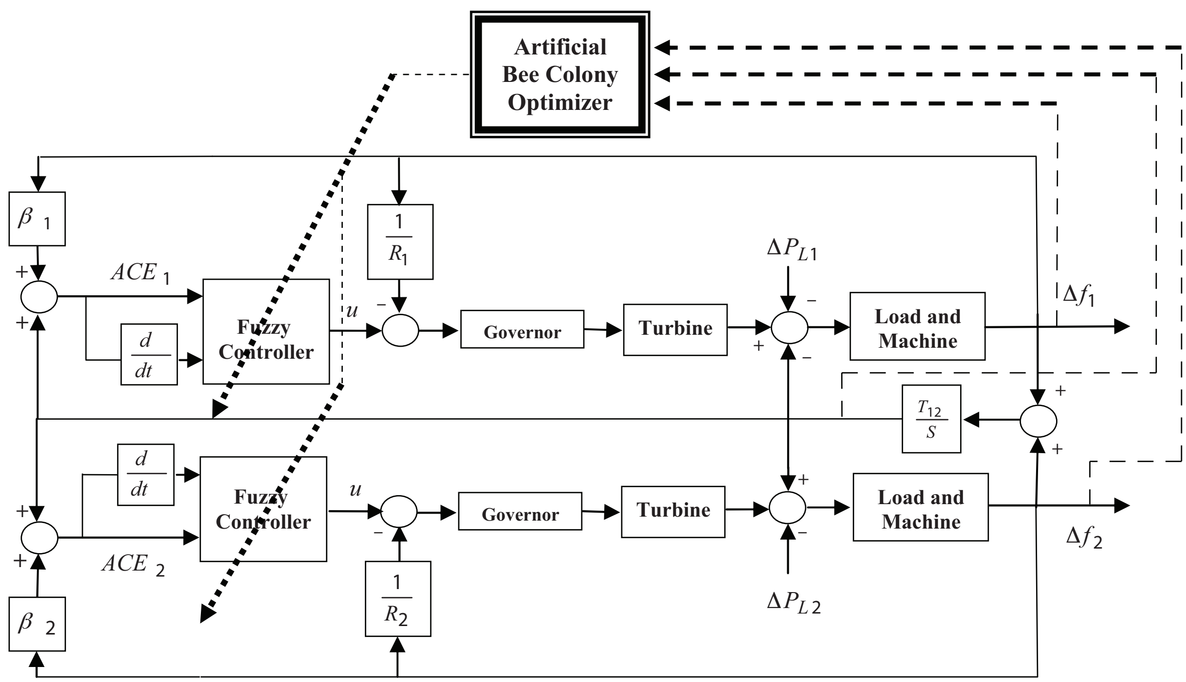

Figure 6. FLFC scheme.

The nonoptimal parameters of the fuzzy variables (initial values before the optimization process) are presented in Table 2. Moreover, the nonoptimal fuzzy rules for areas 1 and 2 are given in Tables 3 and 4, respectively. Next, we use the proposed coding and decoding methods and the ABC algorithm to find the optimal parameters of the FLFCs simultaneously (i.e. MFs' parameters and fuzzy rules).

The first step in applying the ABC algorithm, for the optimal tuning of the MFs and fuzzy rules, is producing the initial population, which shows the food source positions randomly. The population size (SN) is set at 50, i.e. the colony size is equal to 50. The number of employed bees is the same as the number of unemployed bees and is chosen as 25. Since the *limit* parameter has a significant effect on the quality of the solutions, it must be tuned first.

Table 2. The nonoptimal parameters of the fuzzy variables.

MFs' parameters for area 1	X^1_{1a}	X^1_{2a}	X^1_{3a}	X^1_{4a}	X^1_{5a}	X^2_{1a}	X^2_{2a}	X^2_{3a}	X^2_{4a}	X^2_{5a}	X^3_{1a}	X^3_{2a}	X^3_{3a}	X^3_{4a}	X^3_{5a}
	–0.566	–0.376	0.022	0.138	0.409	–0.687	–0.453	–0.106	0.245	0.456	–0.766	–0.380	0.104	0.233	0.598
MFs' parameters for area 2	X^1_{1b}	X^1_{2b}	X^1_{3b}	X^1_{4b}	X^1_{5b}	X^2_{1b}	X^2_{2b}	X^2_{3b}	X^2_{4b}	X^2_{5b}	X^3_{1b}	X^3_{2b}	X^3_{3b}	X^3_{4b}	X^3_{5b}
	–0.675	–0.346	0.023	0.298	0.785	–0.566	–0.238	–0.036	0.410	0.714	–0.709	–0.203	0.105	0.404	0.687

Table 3. The nonoptimal rule base of the fuzzy controller for area 1.

$dACE_1/dt$ \ ACE_1	NM	NS	Z	PS	PM
NM	PM	PM	NS	PS	Z
NS	PM	NS	PS	NM	PS
Z	Z	PS	PS	PM	Z
PS	NS	NM	PS	NM	PS
PM	NS	Z	PM	Z	PM

Table 4. The nonoptimal rule base of the fuzzy controller for area 2.

$dACE_2/dt$ \ AC	NM	NS	Z	PS	PM
NM	PS	PS	PM	PM	NM
NS	PM	PM	Z	NS	PS
Z	NS	PS	NM	PM	PM
PS	Z	Z	NS	Z	PS
PM	PS	Z	PS	NS	NM

Based on the authors' previous works and empirical studies on a number of simulation studies, the best range of variation for the *limit* is 2–10. In this work, a large number of experiments are performed by changing the range of variation for this coefficient and it is found that a value of about 5 for the *limit* results in good a convergence of solutions to the global optimum. The number of iterations, which is called the MCN in the ABC algorithm, is considered to be 50, which is the stopping criterion.

Using Eq. (11), the T-base is calculated as $T=\max\{5, 5, 5\} = 5$. Next, using Eq. (12) gives the decimal integer associated with the considered fuzzy rule, $I = [(4 \times 5^2) + (4 \times 5^1) + (4 \times 5^0)] = 124$; thus, the lower and upper real values of the parameters in the vector solution (Figure 5) (i.e. the MFs' parameters and fuzzy rules) are considered as:

$$\begin{array}{l} 0 < I_{iz} \leq 124, i = 1, ..., 25 \\ -1 < X^j_{kz} < 1, j = 1, 2, 3 k = 1, ..., 5 \end{array} . \tag{17}$$

Table 5 shows the obtained optimal parameters of the MFs using the ABC algorithm. Moreover, Figures 7–12 show the obtained fuzzy MFs for each fuzzy controller in the 2 areas, using the ABC algorithm.

Table 5. The optimal parameters of the fuzzy variables obtained by the ABC algorithm.

MFs' parameters for area 1	X^1_{1a}	X^1_{2a}	X^1_{3a}	X^1_{4a}	X^1_{5a}	X^2_{1a}	X^2_{2a}	X^2_{3a}	X^2_{4a}	X^2_{5a}	X^3_{1a}	X^3_{2a}	X^3_{3a}	X^3_{4a}	X^3_{5a}
	–0.695	–0.201	0.099	0.383	0.724	–0.826	–0.287	–0.051	0.439	0.761	–0.521	–0.359	–0.053	0.485	0.900
MFs' parameters for area 2	X^1_{1b}	X^1_{2b}	X^1_{3b}	X^1_{4b}	X^1_{5b}	X^2_{1b}	X^2_{2b}	X^2_{3b}	X^2_{4b}	X^2_{5b}	X^3_{1b}	X^3_{2b}	X^3_{3b}	X^3_{4b}	X^3_{5b}
	–0.654	–0.284	–0.08	0.289	0.710	–0.54	–0.215	0.022	0.283	0.905	–0.637	–0.446	0.095	0.302	0.698

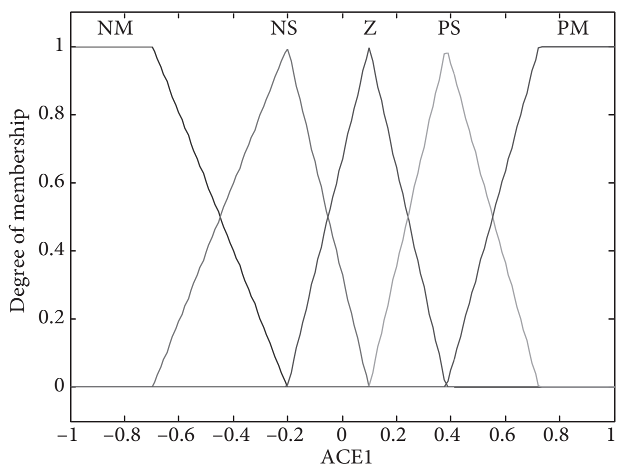

Figure 7. MFs of the ACE in area 1.

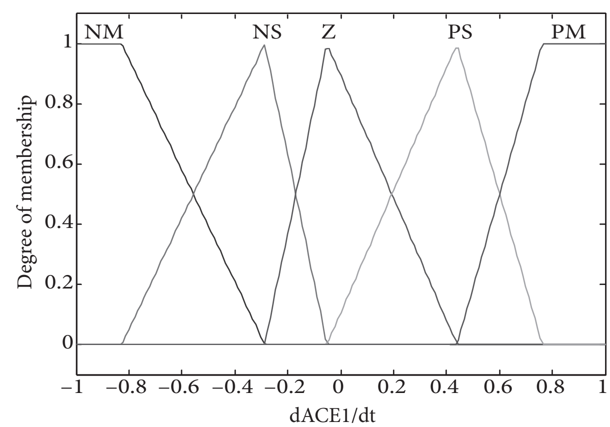

Figure 8. MFs of d/dt (ACE) in area 1.

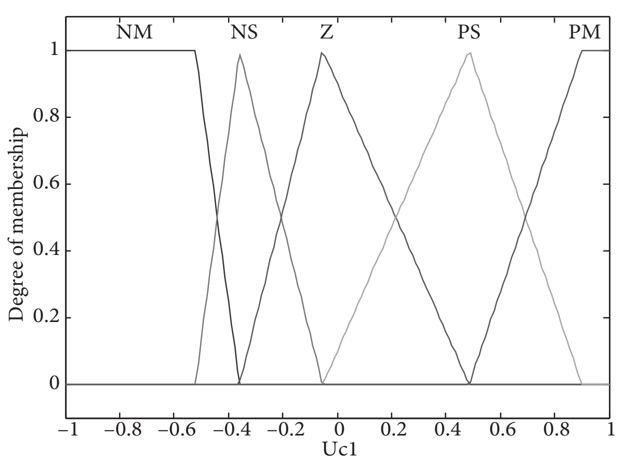

Figure 9. MFs of the output variable in area 1.

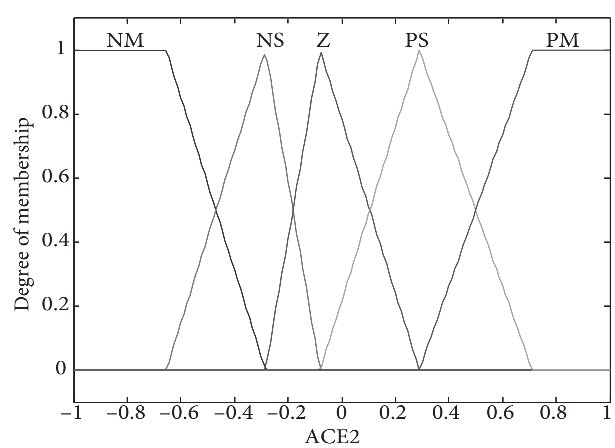

Figure 10. MFs of the ACE in area 2.

The optimal fuzzy rules obtained for areas 1 and 2 are given in Tables 6 and 7, respectively. The fuzzy rules are built from the statement: if input 1 and input 2, then output 1. For instance, consider the 2nd row and 3rd column in Table 6, which means that: if ACE_1 is SN and $\frac{d}{dt}ACE_1$ is Z, then U (the output of the fuzzy controller) is Z.

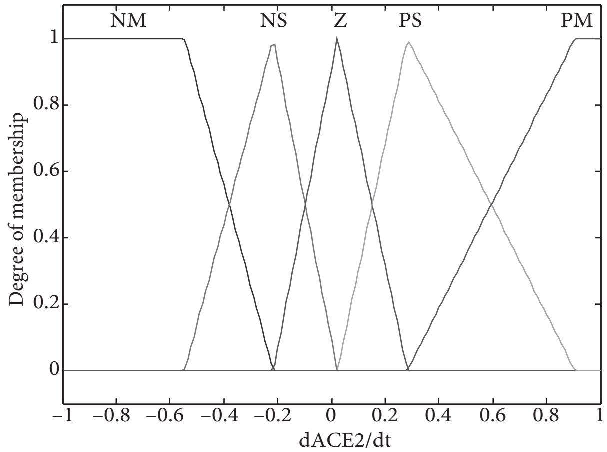

Figure 11. MFs of d/dt (ACE) in area 2.

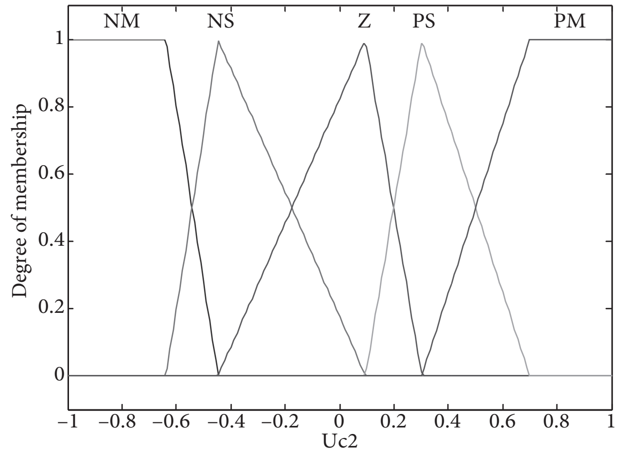

Figure 12. MFs of the output variable in area 2.

Table 6. The optimal rule base of the fuzzy controller obtained by the ABC algorithm for area 1.

ACE_1 \ $dACE_1/dt$	NM	NS	Z	PS	PM
NM	Z	Z	PS	Z	Z
NS	NS	NS	Z	NS	Z
Z	Z	Z	Z	NS	PS
PS	Z	NS	NS	Z	PS
PM	Z	PM	NS	PS	PM

Table 7. The optimal rule base of the fuzzy controller obtained by the ABC algorithm for area 2.

ACE_2 \ $dACE_2/dt$	NM	NS	Z	PS	PM
NM	PS	Z	PS	PM	PS
NS	NM	Z	Z	PM	Z
Z	PS	NS	Z	NS	Z
PS	PS	PS	Z	NS	Z
PM	PS	PM	NS	PS	NS

In order to show the ability and effectiveness of the proposed method, a conventional PI controller, using the approach adopted from [1], is applied for comparison. It is found that $K_{I1} = K_{I2} = 0.3$ are the best selections for having the best performance.

In order to evaluate the performance of the ABC algorithm, a PSO algorithm is applied through the proposed approach. For the PSO algorithm, the population size and number of iterations are considered to be the same as those for the ABC algorithm. The control parameters of the PSO algorithm are chosen as the inertia weight (W), in the considered range of 0 and 1, and the acceleration constants are set as $C_1 = C_2 = 2$.

The designed controllers, as 2 load frequency controllers for 2 areas, and those obtained by the PI controllers and PSO-based method are placed in the case study (Figure 6). To show the effectiveness of the designed controllers, a time domain analysis is performed for the case study. To test the proposed method, a sudden small load perturbation, which continuously disturbs the normal operation of the power system, is applied to the system. Here, we use a step load change of 0.01 p.u. (i.e. $\Delta P_{L1} = \Delta P_{L2} = 0.01$). The frequency deviation of both areas and the tie-line power variation under the nominal conditions of the closed loop system are obtained and shown in Figures 13, 14, and 15, respectively.

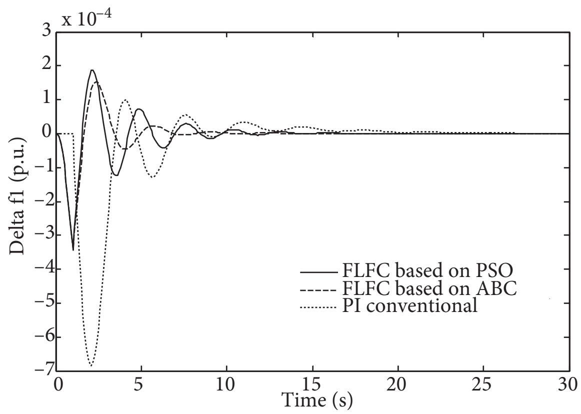

Figure 13. Frequency deviation of area 1.

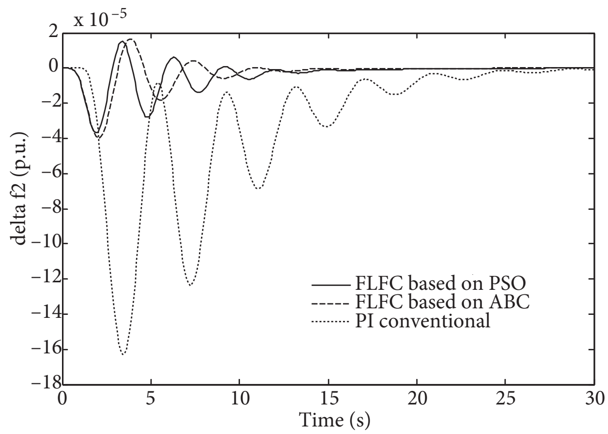

Figure 14. Frequency deviation of area 2.

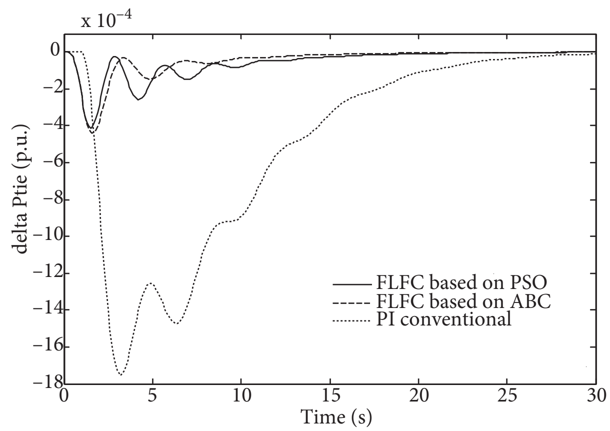

Figure 15. Tie-line power deviation.

From the comparing curves, it can be seen, using the proposed method, that the frequency deviation and tie-line power variation of the 2 areas following the load change and are quickly driven back to 0. It should be mentioned that although the overshoot of the frequency response of the classical PI controller shown in Figure 13 is better than that of the proposed approach, the settling time of the latter is better than that of the former.

To have better clarity, the performance of each controller is investigated by considering the settling time for a 5% band of the step load change, maximum oscillation (peak-to-peak), and the integral square error (ISE) performance index as defined in Eq. (18).

$$ISE = 10^4 \int_0^{30} (ACE_1(t)^2 + ACE_2(t)^2)dt \tag{18}$$

Table 8 shows the obtained considered indices for each controller, where it can be concluded that the proposed FLC-based ABC method gives a better dynamic performance than the classical LFC- and FLC-based PSO algorithms in terms of a fast response, lower settling time and maximum oscillation, and a better ISE index (minimum ISE).

Table 8. Comparison of the performance indices' values for the 3 applied controllers under normal conditions.

Controller	Curve	Max oscillation (p-p)	Settling time	ISE
FLFC-ABC	ΔF_1	5.1031e−004	10.403	3.7506
	ΔF_2	5.2459e−005	14.415	
	ΔP_{tie}	4.1381e−004	17.043	
FLFC-PSO	ΔF_1	5.3818e−004	12.654	4.2560
	ΔF_2	5.2591e−005	15.113	
	ΔP_{tie}	4.3263e−004	18.242	
PI	ΔF_1	7.8369e−004	19.934	30.1388
	ΔF_2	1.6272e−004	28.132	
	ΔP_{tie}	0.0018	28.284	

To investigate the robustness of the proposed approach and the effect of changing the system parameters on the system performance, 2 perturbations in the system parameters are considered. The 1st is a 25% increase for all of the system parameters (upper bound) and the 2nd is a 25% decrease for all of the system parameters (lower bound). The dynamic behavior of the system is evaluated for 30 s and illustrated in Figures 16–21. Again, these responses are similar to the responses in Figures 13–15 for nominal conditions, showing the robustness of the designed controller.

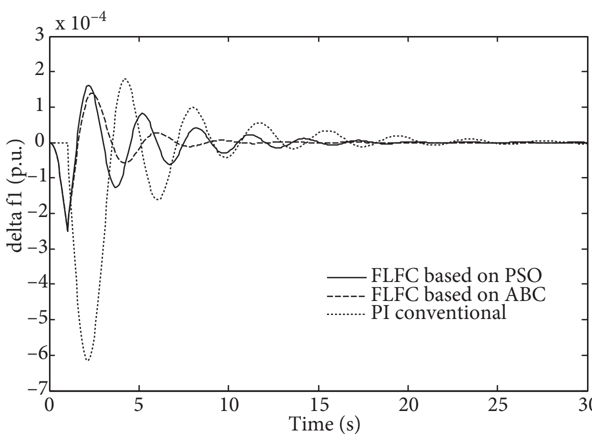

Figure 16. Frequency deviation of area 1 for the upper bound of the parameters.

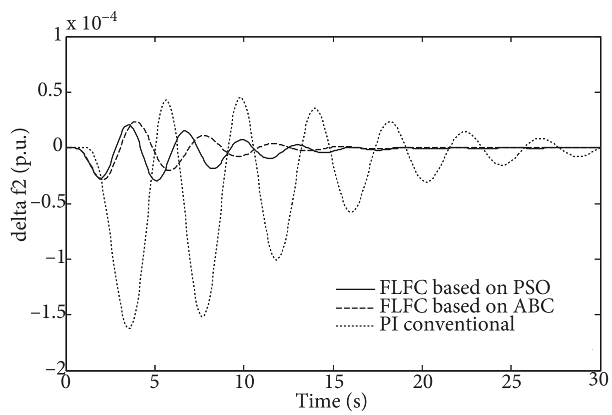

Figure 17. Frequency deviation of area 2 for the upper bound of the parameters.

Once again, the settling time, maximum oscillation, and ISE indices are used to investigate the robustness of the applied controllers against 2 perturbations in the system parameters. Tables 9 and 10 show the obtained indices for each controller under the 25% increase and 25% decrease for all of the system parameters, respectively. A comparison the obtained results reveals that the setting time and maximum oscillation of the optimal FLFC designed using the ABC algorithm are smaller than the obtained results of the other controllers. Moreover, the performance index ISE for the proposed approach based on the ABC and PSO algorithms rarely varies in the presence of system parameter changes. On the other hand, the ISE of the PI controller is larger and changes significantly. These results confirm the robustness of the proposed load frequency controller.

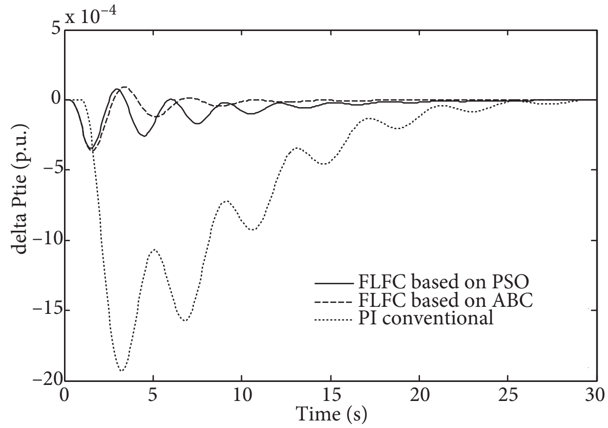

Figure 18. Tie-line power deviation for the upper bound of the parameters.

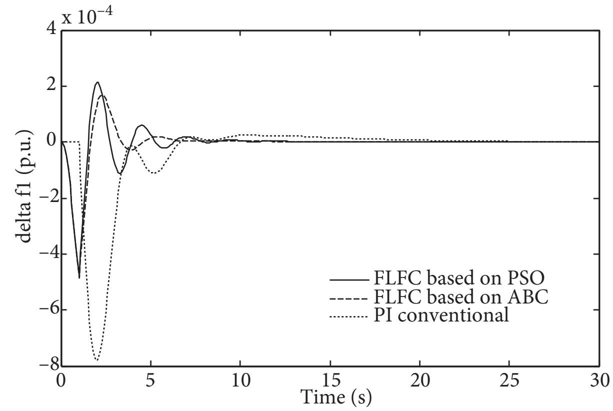

Figure 19. Frequency deviation of area 1 for the lower bound of the parameters.

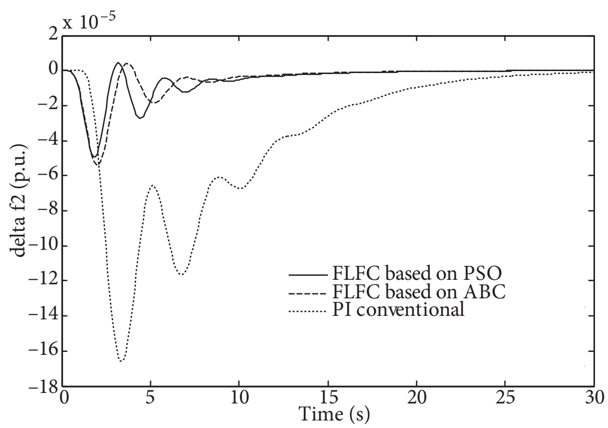

Figure 20. Frequency deviation of area 2 for the lower bound of the parameters.

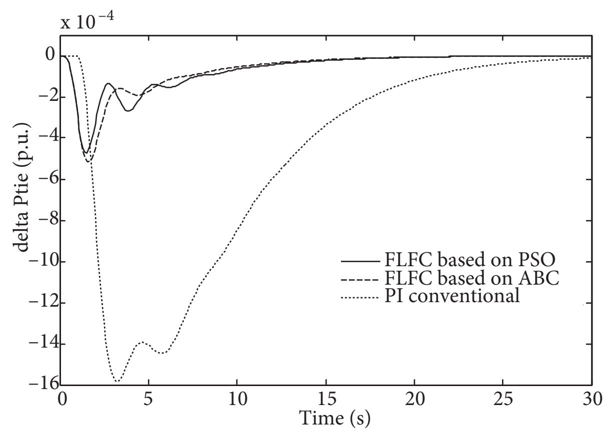

Figure 21. Tie-line power deviation for the lower bound of the parameters.

Table 9. Comparison of the performance indices' values for the 3 applied controllers for the upper bound of the parameters.

Controller	Curve	Max oscillation (p-p)	Settling time	ISE
FLFC-ABC	ΔF_1	5.6721e–004	10.705	3.8429
	ΔF_2	5.6453 –005	13.824	
	ΔP_{tie}	4.1393e–004	9.842	
FLFC-PSO	ΔF_1	5.7919e–004	14.828	4.6536
	ΔF_2	5.6442e–005	15.187	
	ΔP_{tie}	4.2474e–004	17.844	
PI	ΔF_1	7.9646e–004	24.175	39.7199
	ΔF_2	2.0808e–004	>30	
	ΔP_{tie}	0.0019	28.543	

Table 10. Comparison of the performance indices' values for the 3 applied controllers for the lower bound of the parameters.

Controller	Curve	Max oscillation (p-p)	Settling time	ISE
FLFC-ABC	ΔF_1	5.8741e–004	8.292	3.8294
	ΔF_2	5.2959e–005	15.426	
	ΔP_{tie}	4.2284e–004	17.613	
FLFC-PSO	ΔF_1	6.0112e–004	8.627	4.6891
	ΔF_2	5.2838e–005	15.432	
	ΔP_{tie}	4.2163e–004	17.625	
PI	ΔF_1	8.0711e–004	16.113	24.7704
	ΔF_2	1.6599e–004	28.243	
	ΔP_{tie}	0.0016	28.633	

The computation time of each optimization algorithm is an important criterion to evaluate the efficiency of the algorithm. To demonstrate the computational effectiveness of the ABC algorithm, the central processing unit (CPU) time and the value of the cost function (given in Eq. (14)) for before and after the optimization of the results obtained by the ABC and PSO methods are presented in Table 11, where it is seen that the ABC algorithm is computationally efficient as a time requirement and shows better results when compared to the PSO algorithm. Moreover, the convergence characteristics of the ABC and PSO methods in finding the minimum cost are illustrated in Figure 22, which shows the quickness of the ABC algorithm compared to the PSO approach.

Table 11. The CPU time and cost function's values for before and after optimization of the results obtained by the ABC and PSO methods.

Controller	Cost function value after optimization	Cost function value before optimization	CPU time (s)
FLFC-ABC	0.019	4.8853	4.2
FLFC-PSO	0.035	4.8853	8.9

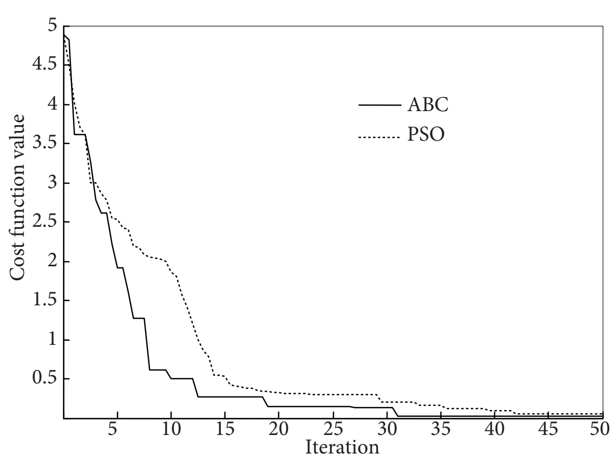

Figure 22. Convergence characteristics of the ABC and PSO methods in finding the minimum cost.

6. Conclusions

In this paper, a new control system incorporating the fuzzy logic controller and ABC algorithm is presented for the control of frequency and damping interarea tie-line power variation in a 2-area interconnected power system. To improve the performance of the FLFC, the MFs and fuzzy control rules of the FLFC are tuned simultaneously using an efficient ABC algorithm. The performance of the designed controller is tested on a 2-area power system while considering governor limiters, and the results obtained are compared with the classical PI controller and PSO-based fuzzy logic controller. The robustness of the proposed method is tested against changes in the parameters. The simulation studies show that the designed controllers by the FLFC-based ABC algorithm have a very desirable dynamic performance, even when the system parameters change. Compared to the conventional PI controller, the fuzzy load frequency-based ABC controller shows much better performances and robustness. It improves both the response time in the transitional-state and reduces considerably the fluctuations in the steady-state. Moreover, the obtained results reveal that the ABC algorithm improves the optimization synthesis and shows better performance than the PSO algorithm in the success rate and solution quality.

References

[1] P. Kundur, Power System Stability and Control, New York, McGraw Hill, 1994.

[2] N.N. Bengiamin, W.C. Chan, “Variable structure control of electric power generation”, IEEE Transactions on Power Apparatus and Systems, Vol. 101, pp. 376–380, 1982.

[3] C.T. Pan, C.M. Liaw, “An adaptive controller for power system load-frequency control”, IEEE Transactions on Power Systems, Vol. 4, pp. 122–128, 1989.

[4] A.M. Kassem, “Neural predictive controller of a two-area load frequency control for interconnected power system”, Ain Shams Engineering Journal, Vol. 1, pp. 49–58, 2010.

[5] K. Vrdoljak, N. Peric, I. Petrovic, “Sliding mode based load-frequency control in power systems”, Electric Power Systems Research, Vol. 80, pp. 514–527, 2010.

[6] H. Shayeghi, “A robust decentralized power system load frequency control”, Journal of Electrical Engineering, Vol. 59, pp. 281–293, 2008.

[7] T. Hiyama, S. Koga, Y. Yoshimuta, “Fuzzy logic based multi-functional load frequency control”, IEEE Power Engineering Society Winter Meeting, Vol. 2, pp. 921–926, 2000.

[8] H. Mohamed, L. Hassan, M. Moghavvemi, S. Yang, “Load frequency controller design for Iraqi national super grid system using fuzzy logic controller”, SICE Annual Conference, pp. 227–232, 2008.

[9] S. Pothiya, I. Ngamroo, “Optimal fuzzy logic-based PID controller for load–frequency control including superconducting magnetic energy storage units”, Energy Conversion and Management, Vol. 49, pp. 2833–2838, 2008.

[10] A. Khodabakhshian, M. Edrisi, “A new robust PID load frequency controller”, Control Engineering Practice, Vol. 16, pp. 1069–1080, 2008.

[11] W. Tan, “Unified tuning of PID load frequency controller for power systems via IMC”, IEEE Transactions on Power Systems, Vol. 25, pp. 341–350, 2010.

[12] H. Shayeghi, A. Jalili, H.A. Shayanfar, “Robust modified GA based multi-stage fuzzy LFC”, Energy Conversion and Management, Vol. 48, pp. 1656–1670, 2007.

[13] H. Shayeghi, A. Jalili, H.A. Shayanfar, “Multi-stage fuzzy load frequency control using PSO”, Energy Conversion and Management, Vol. 49, pp. 2570–2580, 2008.

[14] Q.H. Wu, B.W. Hogg, G.W. Irwin, “A neural network regulator for turbo generator”, IEEE Transactions Neural Networks, Vol. 3, pp. 95–100, 1992.

[15] H. Shayeghi, H.A. Shyanfar, "Application of ANN technique based on μ-synthesis to load frequency control of interconnected power system", Electrical Power and Energy Systems, Vol. 28, pp. 503–511. 2006

[16] Y.L. Abdel-Magid, M.M. Dawoud, "Genetic algorithms applications in load frequency control", Genetic Algorithms in Engineering Systems: Innovations and Applications, Vol. 1, 1995.

[17] P. Li, H. Zhu, Y. Li, "Genetic algorithm optimization for AGC of multi-area power systems", Proceedings of the IEEE TENCON Conference, Vol. 3, pp. 1818–1821, 2002.

[18] E. Bijami, R. Abshari, J. Askari, S. M. Saghaiannejad, "Load frequency control of interconnected power system using brain emotional learning based intelligent controller", Proceedings of the 19th Iranian Conference on Electrical Engineering, pp. 1–6, 2011.

[19] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", Journal of Global Optimization, Vol. 39, pp. 459–471, 2007.

[20] D. Karaboğa, "An idea based on honey bee swarm for numerical optimization", Technical Report-TR06 of Erciyes University, 2005.

[21] D. Karaboga, C. Ozturk, "A novel clustering approach: artificial bee colony (ABC) algorithm", Applied Soft Computing, Vol. 11, pp. 652–657, 2011.

[22] C. Zhang, D. Ouyang, J. Ning, "An artificial bee colony approach for clustering", Expert Systems with Applications, Vol. 37, pp. 4761–4767, 2010.

[23] D. Karaboga, C. Ozturk, B. Akay, "Hybrid artificial bee colony algorithm for neural network training", Proceedings of the Evolutionary Computation Conference (CEC 2011), pp. 84–88, 2011.

[24] A. Singh, "An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem", Applied Soft Computing, Vol. 9, pp. 625–631, 2008.

[25] D. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters", Journal of the Franklin Institute, Vol. 4, pp. 328–348, 2009.

[26] E. Bijami, M. Shahriari-kahkeshi, H. Zamzam, "Simultaneous coordinated tuning of power system stabilizers using artificial bee colony algorithm", Proceedings of the 26th International Power System Conference, pp. 1–8, 2011.

[27] D. Karaboga, B. Basturk, "On the performance of artificial bee colony algorithm", Applied Soft Computing, Vol. 8, pp. 687–697, 2008.

[28] D. Karaboga, B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems", Proceedings of the 12th international Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft Computing, pp. 789–798, 2007.