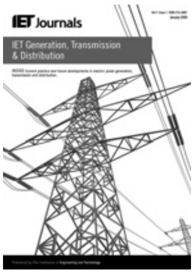
Published in IET Generation, Transmission & Distribution
Received on 30th March 2009
Revised on 28th August 2009
doi: 10.1049/iet-gtd.2009.0168



ISSN 1751-8687

Load–frequency control: a GA-based multi-agent reinforcement learning

F. Daneshfar H. Bevrani
Department of Electrical and Computer Engineering, University of Kurdistan, Sanandaj, PO Box 416, Kurdistan, Iran
E-mail: daneshfar@ieee.org

Abstract: The load–frequency control (LFC) problem has been one of the major subjects in a power system. In practice, LFC systems use proportional–integral (PI) controllers. However since these controllers are designed using a linear model, the non-linearities of the system are not accounted for and they are incapable of gaining good dynamical performance for a wide range of operating conditions in a multi-area power system. A strategy for solving this problem because of the distributed nature of a multi-area power system is presented by using a multi-agent reinforcement learning (MARL) approach. It consists of two agents in each power area; the estimator agent provides the area control error (ACE) signal based on the frequency bias (β) estimation and the controller agent uses reinforcement learning to control the power system in which genetic algorithm optimisation is used to tune its parameters. This method does not depend on any knowledge of the system and it admits considerable flexibility in defining the control objective. Also, by finding the ACE signal based on β estimation the LFC performance is improved and by using the MARL parallel, computation is realised, leading to a high degree of scalability. Here, to illustrate the accuracy of the proposed approach, a three-area power system example is given with two scenarios

1 Introduction

Frequency changes in large-scale power systems are a direct result of the imbalance between the electrical load and the power supplied by system connected generators [1]. Therefore load–frequency control (LFC) is one of the important power system control problems for which there have been considerable research works [2–5]. In [6], a centralised controller is designed for a two-area power system, which requires the knowledge of the whole system. In [7], decentralised controllers for a two-area power system are proposed. These controllers are designed based on modern control theory, and each area requires knowledge of the other area. If the dimensions of the power system increase, then these controllers may become more complex as the number of the state variables increase significantly.

Also, there has been continuing interest in designing load–frequency controllers with better performance using various decentralised robust and optimal control methods during the last two decades [8–18].

In [17], two robust decentralised control design methodologies for the LFC are proposed. The first one is based on control design using the linear matrix inequalities (LMI) technique and the second one is tuned by a robust control design algorithm. However, in [18], a decentralised LFC synthesis is formulated as an HN-control problem and is solved using an iterative LMI algorithm that gains lower order proportional–integral (PI) controller than [17]. Both controllers are tested on a three-control area power system with three scenarios of load disturbances to demonstrate their robust performances.

But all the above-mentioned controllers are designed for a specific disturbance; if the nature of the disturbance varies, they may not perform as expected. Also, they are model-based controllers that are dependent on a specific model, and are not usable for large systems like power systems with non-linearities, not-defined parameters and model. The proposed methods assume that all model parameters are defined and measurable (fixed) too, that in a real power system some parameters like β change with environment conditions and they do not have constant values. Therefore the design of intelligent controllers that are more adaptive than linear and robust controllers has become an appealing approach [19–21].

One of the adaptive and non-linear control techniques that has found applications in the LFC design is reinforcement learning (RL) [22–27]. These controllers learn and they are adjusted to keep the area control error (ACE) small in each sampling time of a discretised LFC cycle. Since they are based on learning methods and are independent of environmental conditions and can learn each kind of environmental disturbances, therefore they are not model based and can easily be scalable for large-scale systems. They can also work well in non-linear conditions and non-linear systems.

In this paper, a multi-agent reinforcement learning (MARL)-based control structure is proposed that has the β estimation as one of its functionalities. It consists of two agents in each control area that communicate with each other to control the whole system. The first agent (i.e. the estimator agent) provides the ACE signal based on the β parameter estimation and the second agent (i.e. the controller agent) provides ΔP_c according to the ACE signal received from the estimator agent, using RL; then it is distributed among the different units under control using fixed participation factors. In a multi-area power system, the learning process is a MARL process and all agents of all areas learn together (not individually).

The above technique has been applied to the LFC problem in a three-control area power system as a case study. In the new environment, the overall power system can also be considered as a collection of control areas interconnected through high-voltage transmission lines or tie-lines. Each control area consists of a number of generating companies (Gencos) and it is responsible for tracking its own load and performing the LFC task.

The organisation of the rest of the paper is as follows. In Section 2, a brief introduction to the single-agent-based and multi-agent-based RL and the LFC problem is given. In Section 3, an explanation on how a load–frequency controller can work within this formulation is provided. In Section 4, a case study of a three-control area power system, for which the above architecture is implemented, is discussed. Simulation results are provided in Section 5 and paper is concluded in Section 6.

2 Background

In this section, a brief background on single-agent RL and MARL [28] is introduced. First, the single-agent RL is defined and its solution is described. Then, the multi-agent task is defined. The discussion is restricted to finite state and action spaces, since the major part of the MARL results are given for finite spaces [29].

2.1 Single-agent RL

RL is learning what to do – how to map situations to actions – so as to maximise a numerical reward signal [26]. In fact, the learner will discover which action should be taken by interacting with the system and trying the different actions that may lead to the highest reward. RL will evaluate the actions taken and gives the learner a feedback on how good the action taken was and whether it should repeat this action in the same situation or not. In another words, RL methods learn to solve a problem by interacting with a system. The learner is called the agent and the system it interacts with is known as the environment. During the learning process, the agent interacts with the environment and takes an action α_t from a set of actions at time t. These actions will affect the system and will take it to a new state $\boldsymbol{x}_{t+1}$. Therefore the agent is provided with the corresponding reward signal (γ_{t+1}). This agent–environment interaction is repeated until the desired goal is achieved. In this paper, what is meant by the state is the required information for making a decision, therefore what we would like, ideally, is a state signal that summarises past perceptions in a way in which all relevant information is retained. A state signal that succeeds in retaining all relevant information is said to be Markov, or as having the Markov property [26] and an RL task that satisfies this property is called a finite Markov decision process (MDP). If an environment has the Markov property, then its dynamics enable us to predict the next state and expected next reward, given the current state and action. In the remaining of the text, it is assumed that the environment has the Markov property, therefore an MDP problem is solved. In each MDP, the objective is to maximise the sum of returned rewards over time, and the expected sum of discounted rewards is defined by

$$R = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

where $0 < \gamma < 1$ is a discount factor, which gives the maximum importance to the recent rewards.

Another term is the value function which is defined as the expected return (reward) when starting at state $\boldsymbol{x}_t$ while following policy $\pi(\boldsymbol{x}, \alpha)$ [see (2)]. Policy is the way in which the agent maps the states to the actions [26]

$$V^{\pi}(\boldsymbol{x}) = E_{\pi}\left\{\sum_{k=0}^{\infty} [\gamma^k r_{t+k+1} | \boldsymbol{x}_t = \boldsymbol{x}]\right\} \tag{2}$$

The optimal policy is the one that maximises the value function. Therefore once the optimal state value is derived, the optimal policy can be found using

$$V^{*}(\boldsymbol{x}) = \max_{\pi} V^{\pi}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in X \tag{3}$$

In most RL methods, instead of calculating the state value, another term known as the action value is calculated (4),

doi: 10.1049/iet-gtd.2009.0168

which is defined as the expected discounted reward while starting at state $\boldsymbol{x}_t$ and taking action $\boldsymbol{a}_t$

$$Q^{\pi}(\boldsymbol{x}, \boldsymbol{a}) = E_{\pi}\left\{\sum_{k=0}^{\infty}[\gamma^k r_{t+k+1}|\boldsymbol{x}_t = \boldsymbol{x},\ \boldsymbol{a}_t = \boldsymbol{a}]\right\} \tag{4}$$

Bellman's equation, as shown below, is used to find the optimal action value. Overall, an optimal policy is one that maximises the Q-function defined in (5)

$$Q^{\uparrow *}(\boldsymbol{x}, \boldsymbol{a}) = \max\ T\boldsymbol{\pi}E_{\downarrow}\pi\{r_{\downarrow}(t+1) + \gamma \max\ T\boldsymbol{a}' \\ [Q^{\uparrow *}(\boldsymbol{x}_{\downarrow}(t+1), \boldsymbol{a})]|\boldsymbol{x}_{\downarrow}t = \boldsymbol{x},\ \boldsymbol{a}_{\downarrow}t = \boldsymbol{a}\} \tag{5}$$

Different RL methods have been proposed to solve the above equations. In some algorithms, the agent first approximates the model of the system in order to calculate the Q-function. The method used in this paper is of the temporal difference type which learns the model of the system under control. The only available information is the reward achieved by each action taken and the next state. The algorithm, called Q-learning, will approximate the Q-function and by the computed function, the optimal policy that maximises this function is derived [26].

2.2 Multi-agent reinforcement learning

A multi-agent system [30] can be defined as a group of autonomous, interacting entities (or agents) [31] sharing a common environment, which they perceive with sensors and upon which they act with actuators [32]. Multi-agent systems can be used in a wide variety of domains including robotic teams, distributed control, resource management, collaborative decision support systems and so on. Well-understood algorithms with good convergence and consistency properties are available for solving the single-agent RL task, both when the agent knows the dynamics of the environment and the reward function (the task model), and when it does not. However, the scalability of algorithms to realistic problem sizes is problematic in single-agent RL, and is one of the great reasons for which MARL should be used [29]. In addition to scalability and benefits owing to the distributed nature of the multi-agent solution, such as parallel computation, multiple RL agents may utilise new benefits from sharing experience, for example, by communication, teaching or imitation [29]. These properties make RL suitable for multi-agent learning. However, several new challenges arise for RL in multi-agent systems. In multi-agent systems, other adapting agents make the environment no longer stationary, violating the Markov property that traditional single-agent behaviour learning relies on; this non-stationarity properties decrease the convergence properties of most single-agent RL algorithms [33]. Another problem is the difficulty of defining a good learning goal for the multiple RL agents [29]. Only then it will be able to coordinate its behaviour with other agents. These challenges make the MARL design and learning difficult in big applications; therefore it uses a special learning algorithm as mentioned below. (Violating the Markov property problem – caused by the multi-agent structure – can be solved using following learning algorithm.)

2.2.1 Learning algorithm:

The generalisation of the MDP to the multi-agent case is as follows:

Suppose a tuple $\langle X, A_1, \ldots, A_n, p, r_1, \ldots, r_n\rangle$ where n is the number of agents, X is the discrete set of environment states, $A_i,\ i = 1, \ldots, n$, are the discrete sets of actions available to the agents, yielding the joint action set $A = A_1 \times \cdots \times A_n,\quad p{:}X \times A \times X \rightarrow [0, 1]$ is the state transition probability function, and $r_i = X \times A\ \ \times X \rightarrow R,\quad i = 1, \ldots, n$ are the reward functions of the agents.

In the multi-agent case, the state transitions are the result of the joint action of all the agents, $\boldsymbol{a}_k = [a_{1\,k}^{\mathrm{T}}, \ldots, a_{n,k}^{\mathrm{T}}]$, $\boldsymbol{a}_k \in A$, $\boldsymbol{a}_{i,k} \in A_i$ (where T denotes vector transpose). As a result, the rewards $r_{i,k+1}$ and the returns $R_{i,k}$ also depend on the joint action. The policies $\boldsymbol{h}_i{:}X \times A_i \rightarrow [0, 1]$ form together the joint policy $\boldsymbol{h}$. The Q-function of each agent depends on the joint action and is conditioned on the joint policy, $Q_i^{\boldsymbol{h}}{:}X \times A \rightarrow R$ [29].

2.3 Load–frequency control

A large-scale power system consists of a number of interconnected control areas [34]. Fig. 1 shows the block diagram of control area i, which includes n Gencos, from an N-control area power system. As is usual in the LFC design literature, three first-order transfer functions are used to model generators, turbine and power system (rotating mass and load) units. The parameters are described in the list of symbols in [34]. Following a load disturbance within a control area, the frequency of that area experiences a transient change, the feedback mechanism comes into play and generates appropriate rise/lower signal to the participating Gencos according to their participation factors α_{ji} in order to make the generation follow the load. In the steady state, the generation is matched with the load, driving the tie-line power and frequency deviations to zero. The balance between connected control areas is achieved by detecting the frequency and tie-line power deviations in order to generate the ACE signal which is, in turn, utilised in the PI control strategy, as shown in Fig. 1. The ACE for each control area can be expressed as a linear combination of tie-line power change and frequency deviation [34]

$$\mathrm{ACE}_i = B_i\Delta f_i + \Delta P_{\text{tie}-i} \tag{6}$$

3 Proposed control framework

In practice, the LFC controller structure is traditionally a PI-type controller using the ACE as its input, as shown in Fig. 1. In this section, the intelligent control design algorithm for such a load–frequency controller using the MARL technique is presented. The objective of the proposed design

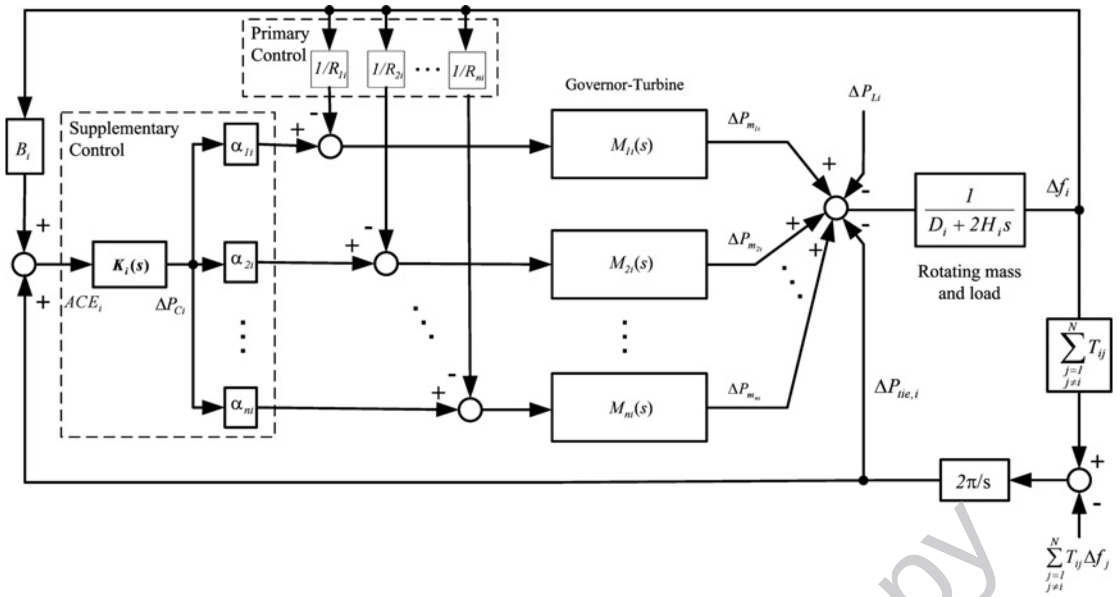


Figure 1 *LFC system with different generation units and participation factors in area i [34]*

is to control the frequency so as to achieve the same performance as the proposed robust control design in [17, 18].

Regarding the LFC problem to be an MDP problem [28], the RL can be applied to its controller.

Fig. 2 shows the proposed model for area *i*; two kinds of intelligent agents have been used in this structure, the controller agent and the estimator agent. The estimator agent is responsible for estimating the frequency bias parameter (β) and calculating the ACE signal, whereas the controller agent is responsible for finding ΔP_c according to the ACE signal using RL and GA.

3.1 Controller agent

The PI controller in LFC can be replaced with an intelligent controller that decides on the set point changes of separate closed-loop control systems. This will allow us to design the LFC algorithm in order to operate on a discrete time scale and the results are more flexible. In this view, the intelligent controller (controller agent) system can be abstracted as follows: At each instant (on a discrete time scale *k*), $k = 1, 2, \ldots$, the controller agent observes the current state of the system, $\boldsymbol{x}_k$, and takes an action, $\boldsymbol{a}_k$. The state vector consists of some quantities, which are normally available to the controller agent. Here, the average of ACE signal instances and ΔP_{L} instances over the LFC execution period are used as the state vector. For the algorithm presented in this paper, it is assumed that the set of all possible states *X*, is finite. Therefore the values of various quantities that constitute the state information should be quantised.

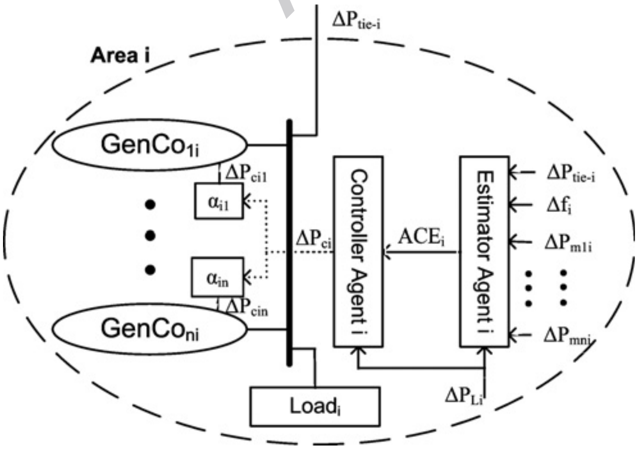


Figure 2 *Proposed model for area i*

The possible actions of the controller agent are the various values of ΔP_{c}, that can be demanded in the generation level within an LFC interval. ΔP_{c} is also discretised to some finite number of levels. Now, since both *X* and *A* are finite sets, a model for this dynamic system can be specified through a set of probabilities.

Here an RL algorithm is used for estimating Q^* and the optimal policy. It is the same as the algorithm used in [25].

Suppose there is a sequence of samples ($\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{a}_k, r$), $k = 1, 2, \ldots$ (*k* is the LFC execution period). Each sample is such that $\boldsymbol{x}_{k+1}$ is the (random) state that resulted when action $\boldsymbol{a}_k$ is performed in state $\boldsymbol{x}_k$ and $r_k = g(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{a}_k)$ is the consequent immediate reinforcement. Such a sequence of samples can be obtained either through a simulation model of the system or by observing the actual system in operation. This sequence of samples (called training set) can be used for estimating Q^*. The specific algorithm that

doi: 10.1049/iet-gtd.2009.0168

is used is as follows. Suppose Q^k is the estimate of Q^* at the kth iteration. Let the next sample be $(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{a}_k, r)$. Then Q^{k+1} is obtained as

$$Q^{k+1}(\boldsymbol{x}_k, \alpha_k) = Q^k(\boldsymbol{x}_k, \alpha_k) + \alpha[g(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \alpha_k) + \gamma \max_{\alpha \in A} Q^k(\boldsymbol{x}_{k+1}, \alpha^*) - Q^k(\boldsymbol{x}_k, \alpha_k)] \quad (7)$$

where $0 < \alpha < 1$ is a constant called the step size of learning algorithm.

At each time step (as determined by the sampling time for the LFC action), the state input vector, $\boldsymbol{x}$, to the LFC is determined, and then an action in that state is selected and applied on the model; the model is integrated for a time interval equal to the sampling time of the LFC to obtain the state vector $\boldsymbol{x}^*$ at the next time step.

Here, the exploration policy for choosing actions in different states is used. It is based on a learning automata algorithm called the pursuit algorithm [35]. This is a stochastic policy where, for each state $\boldsymbol{x}$, actions are chosen based on a probability distribution over the action space. Let $P_{\boldsymbol{x}}^k$ denote the probability distribution over the action set for the state vector $\boldsymbol{x}$ at the kth iteration of learning. That is, $P_{\boldsymbol{x}}^k(\alpha)$ is the probability of choosing action α in state $\boldsymbol{x}$ at iteration k. Initially (i.e. at $k = 0$), a uniform probability distribution is chosen. That is

$$P_{\boldsymbol{x}}^0(\alpha) = \frac{1}{|A|} \quad \forall \alpha \in A \quad \forall \boldsymbol{x} \in X \quad (8)$$

At the kth iteration, let the state $\boldsymbol{x}_k$ be equal to $\boldsymbol{x}$. An action α_k, based on $p_{\boldsymbol{x}}^k(\cdot)$ is chosen at random. That is,

$\mathrm{Prob}(\alpha_k = \alpha) = p_{\boldsymbol{x}}^k(\alpha)$. Using our simulation model, the system goes to the next state $\boldsymbol{x}_{k+1}$ by applying action α in state $\boldsymbol{x}$ and is integrated for the next time interval. Then Q^k is updated to Q^{k+1} using (7) and the probabilities too are updated as follows.

$$\begin{aligned} P_{\boldsymbol{x}}^{k+1}(\alpha_g) &= P_{\boldsymbol{x}}^k(\alpha_g) + \beta(1 - P_{\boldsymbol{x}}^k(\alpha_g)) \\ P_{\boldsymbol{x}}^{k+1}(\alpha) &= P_{\boldsymbol{x}}^k(\alpha)(1 - \beta) \quad \forall \alpha \in A,\ \alpha \neq \alpha_g \\ P_{\boldsymbol{y}}^{k+1}(\alpha) &= P_{\boldsymbol{y}}^k(\alpha) \quad \forall \alpha \in A,\ \forall \boldsymbol{y} \in X,\ \boldsymbol{y} \neq \boldsymbol{x} \end{aligned} \quad (9)$$

where $0 < \beta < 1$ is a constant. Thus at iteration k, the probability of choosing the greedy action α_g in state $\boldsymbol{x}$ is slightly increased and the probabilities of choosing all other actions in state $\boldsymbol{x}$ are proportionally decreased.

3.1.1 *Learning the controller:*

In this algorithm, the aim is to achieve the conventional LFC objective and keep the ACE within a small band around zero. This choice is motivated by the fact that all the existing LFC implementations use this as the control objective and hence, it will be possible for us to compare the proposed RL approach with the designed robust PI based LFC approaches in [17, 18]. As mentioned above in this formulation, each state vector consists of two state variables: the average value of the ACE (the first state variable, x^1) and the ΔP_L (the second state variable, x^2). Since the RL algorithms are considered, which assume a finite number of states, state variables should be discretised to finite levels. Here, the genetic algorithms (GAs) optimisation is used to find good discretised state vector levels.

The control action of the LFC is to change the generation set point, ΔP_c. Since the range of generation change that can be affected in an LFC cycle is known, this range can be discretised to finite levels using GA, too.

The next step is to choose an immediate reinforcement function by defining the function g. The reward matrix initially is full of zero; at each LFC execution period, the average value of ΔP_L and the average value of the ACE signal are obtained, and then according to the discritised values gained from GA, the state of the system is determined. Whenever the state is desirable (i.e.|ACE| is less than ε), reward function $g(\boldsymbol{x}_k, \boldsymbol{x}_{k+1,}\alpha_k)$ is assigned a value zero. When it is undesirable (i.e. $|\mathrm{ACE}| > \varepsilon$), then $g(\boldsymbol{x}_k, \boldsymbol{x}_{k+1,}\alpha_k)$ is assigned a value $-|\mathrm{ACE}|$ (all actions which cause to go to an undesirable state are penalised with a negative value).

3.1.2 *Finding actions and states based on GA:*

GA is used to gain better results and to tune the quantised values of the state vector and the action vector.

To quantise the state range and the action range using GA, each individual that is explanatory quantised values of states and actions, should be a double vector. It is clear that with increasing the number of variables in a double vector, the states (ACE signal quantised values) are found more precisely. Actually here, system states are more important than system actions (ΔP_c quantised values) and has a greater effect on the whole system performance (keeping the ACE within a small band around zero), because systems with more states can learn better (more precisely) than similar systems with less states. Actually, the maximum number of states that could be defined in GA for the mentioned system were 400 (for the values greater than this, there was memory error); however, for actions variable if the number of variables is limited, the speed of learning process is more (because it is not necessary to examine extra actions in each state). The least valuable actions number that could be defined was 6. Then each individual should be a double vector (population type) with 406 variables between [0 1] that consists of 400 variables for the ACE signal and six variables for the ΔP_csignal.

The start population size is equal to 30 individuals and it was run for 100 generations. Fig. 3 shows the result of the running proposed GA for area 1 of the three-control area power system given in [17, 18].

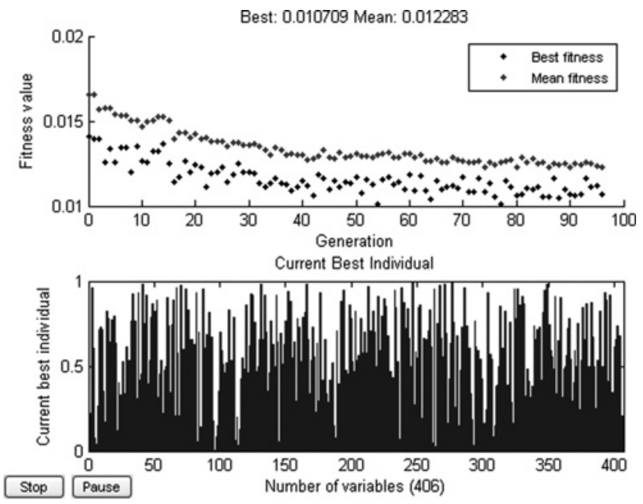


Figure 3 *Result of running GA for area 1 of the three-control area power system given in [17, 18]*

Finding individual's fitness: To find the eligibility (fitness) of individuals, six variables are randomly chosen as discretised values of actions from each individual (which contains 406 variables); then, these values should be scaled according to the action range (variable's range is between 0 and 1; however, the variation of the ΔP_c action signal is between $[\Delta P_{\downarrow} c\,\mathrm{Min}\ \Delta P_{\downarrow} c\,\mathrm{Max}]$) and the remaining other 400 variables are discretised values of the ACE signal which should be scaled to the valid range ($[(\mathrm{ACE})_{\downarrow}\mathrm{Min}(\mathrm{ACE})_{\downarrow}\mathrm{Max}]$). After scaling and finding the corresponding quantised state and action vector, the model is run with these properties, and the individual's fitness is obtained from

$$\text{Individual fitness} = \frac{\sum |\mathrm{ACE}|}{(\text{simulation time})} \tag{10}$$

Each individual that has the smallest fitness is the best one.

Finding ACE and ΔP_c variation range: Hence, the AGC's role is limited to correcting the observed ACE in a limited range; if the ACE goes beyond this range, other emergency control steps may have to be taken by the operator. Let the valuable range of the ACE for which the AGC is expected to act properly is $[(\mathrm{ACE})_{\downarrow}\mathrm{Min}(\mathrm{ACE})_{\downarrow}\mathrm{Max}]$. In fact, $\mathrm{ACE}_{\mathrm{Min}}$ and $\mathrm{ACE}_{\mathrm{Max}}$ get automatically determined by the operating policy of the area. $\mathrm{ACE}_{\mathrm{Max}}$ is the maximum ACE deviation that is expected to be corrected by the AGC (in practice, ACE deviations beyond this value are corrected only through operator intervention) and $\mathrm{ACE}_{\mathrm{Min}}$ is the amount of ACE deviation below which we do not want the AGC to respond. This value must be necessarily non-zero.

The other variable to be quantised is the control action ΔP_c. This also requires that design choice be made for the range $[\Delta P_{\downarrow} c\,\mathrm{Min}\ \Delta P_{\downarrow} c\,\mathrm{Max}]$. $\Delta P_{c\,\mathrm{Max}}$ is automatically determined by the equipment constraints of the system. It is the maximum power change that can be effected within one AGC execution period. $\Delta P_{c\,\mathrm{Min}}$ is the minimum change that can be demanded in the generation.

3.2 Estimator agent

To find the ACE from (6), it is necessary to know the frequency bias factor (β). The conventional approaches in tie-line bias control use the frequency bias coefficient $-10B$ to offset the area's frequency response characteristic, β. But it is related to many factors and with $-10B = \beta$, the ACE would only react to internal disturbances. Therefore many works have used the β approximation (which is not easily obtained on a real time basis) instead of a constant value [36–39]. Getting a good estimate of the area's β to improve the LFC performance is a motivation for the estimator agent to estimate the β parameter and find the ACE signal based on it.

Each time, the estimator agent obtains ΔP_{tie}, Δf, ΔP_{m}, ΔP_{L} (ΔP_{L} is not measurable directly but it can be estimated by classical and conventional solutions [40]) as inputs, then calculates the β parameter and finds the ACE signal according to that. The proposed estimation algorithm used in this agent is based on the ACE model given in Fig. 1.

The per unit equation of the electromechanical power balance for the local control area (Fig. 1) can be expressed as

$$\begin{aligned}\sum_{j=1}^{n} \Delta P_{\mathrm{m}ji}(t) - \Delta P_{\mathrm{L}i}(t) - \Delta P_{\mathrm{tie},i}(t) \\ = 2H_i \Delta f_i(t) + D_i \Delta f_i(t)\mathrm{pu}\end{aligned} \tag{11}$$

Also, the equation given below is concluded from (6)

$$\Delta P_{\mathrm{tie},i}(t) = \mathrm{ACE}_i(t) - \beta_i \Delta f_i(t)\ \mathrm{pu} \tag{12}$$

Using the results of the two above equations

$$\begin{aligned}\sum_{j=1}^{n} \Delta P_{\mathrm{m}ji}(t) - \Delta P_{\mathrm{L}i}(t) + \beta_i \Delta f_i(t) - \mathrm{ACE}_i(t) \\ = 2H_i \Delta f_i(t) + D_i \Delta f_i(t)\end{aligned} \tag{13}$$

Then

$$\begin{aligned}\mathrm{ACE}_i(t) = \sum_{j=1}^{n} \Delta P_{\mathrm{m}ji}(t) - \Delta P_{\mathrm{L}i}(t) + (\beta_i - D_i)\Delta f_i(t) \\ - 2H_i \Delta f_i(t)\ \mathrm{pu}\end{aligned} \tag{14}$$

Applying the following definition for a moving average over a T-second interval to (14)

$$\overline{X_T} = \frac{1}{T}\int_{t_{\mathrm{i}}}^{t_{\mathrm{f}}} X(t)\mathrm{d}t, \quad t_{\mathrm{f}} - t_{\mathrm{i}} = T\ \mathrm{s} \tag{15}$$

doi: 10.1049/iet-gtd.2009.0168

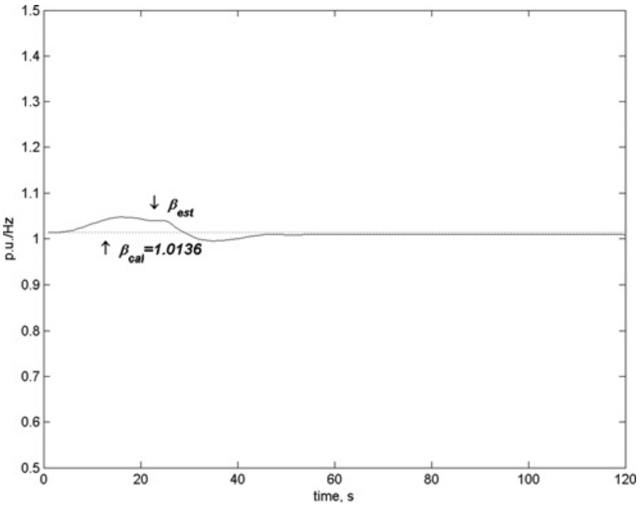


Figure 4 *β_{est} and β_{cal} over 120 s*

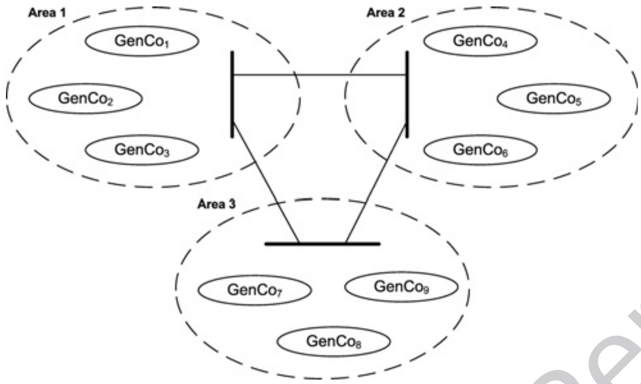


Figure 5 *Three-control area power system*

the following equation is obtained

$$\overline{\mathrm{ACE}}_T = \frac{1}{T}\left\{\sum_T \sum_{j=1}^{n} \Delta P_{\mathrm{m}ji} - \sum_T \Delta P_{\mathrm{L}i} + (\beta_i - D_i) \times \sum_T \Delta f_i - 2H_i(\Delta f(t_\mathrm{f}) - \Delta f(t_\mathrm{i}))\right\} \qquad (16)$$

By applying the measured values of ACE and other variables in the above equation over a time interval, the values of β can be estimated for the corresponding period. Since the values of β vary with system conditions, these model parameters would have to be updated regularly using a recursive least square (RLS) algorithm [41].

Suitable values of the duration T depend on the system's dynamic behaviour. Although a larger T would yield smoother β values, it would also slow the convergence to the proper value. For the example at hand, a T equal to 60 s gave good results.

Fig. 4 shows the estimated and calculated β of Fig. 1 for area 1 of the three-control area power system given in [17, 18] over a 120 s simulation. For this test, the $-10B$ of the target control area was set equal to β_{cal}. As it is clear, β_{est} converged rapidly to the β_{cal} and remained there about over the rest of the run.

4 Case study: a three-control area power system

To illustrate the effectiveness of the proposed control strategy, and to compare the results with linear robust

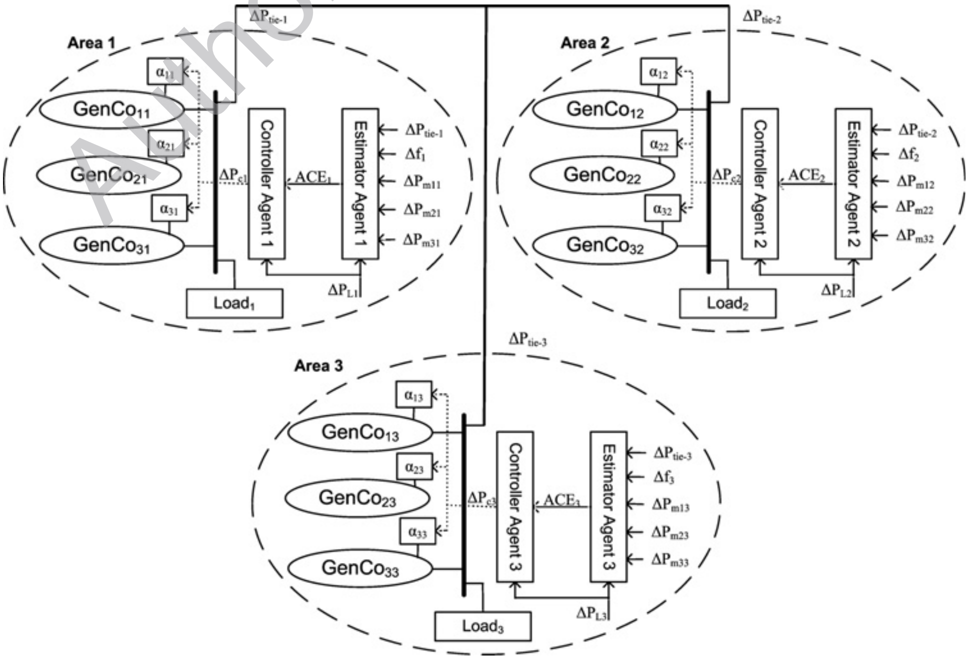


Figure 6 *Proposed multi-agent structure for the three-control area power system*

doi: 10.1049/iet-gtd.2009.0168

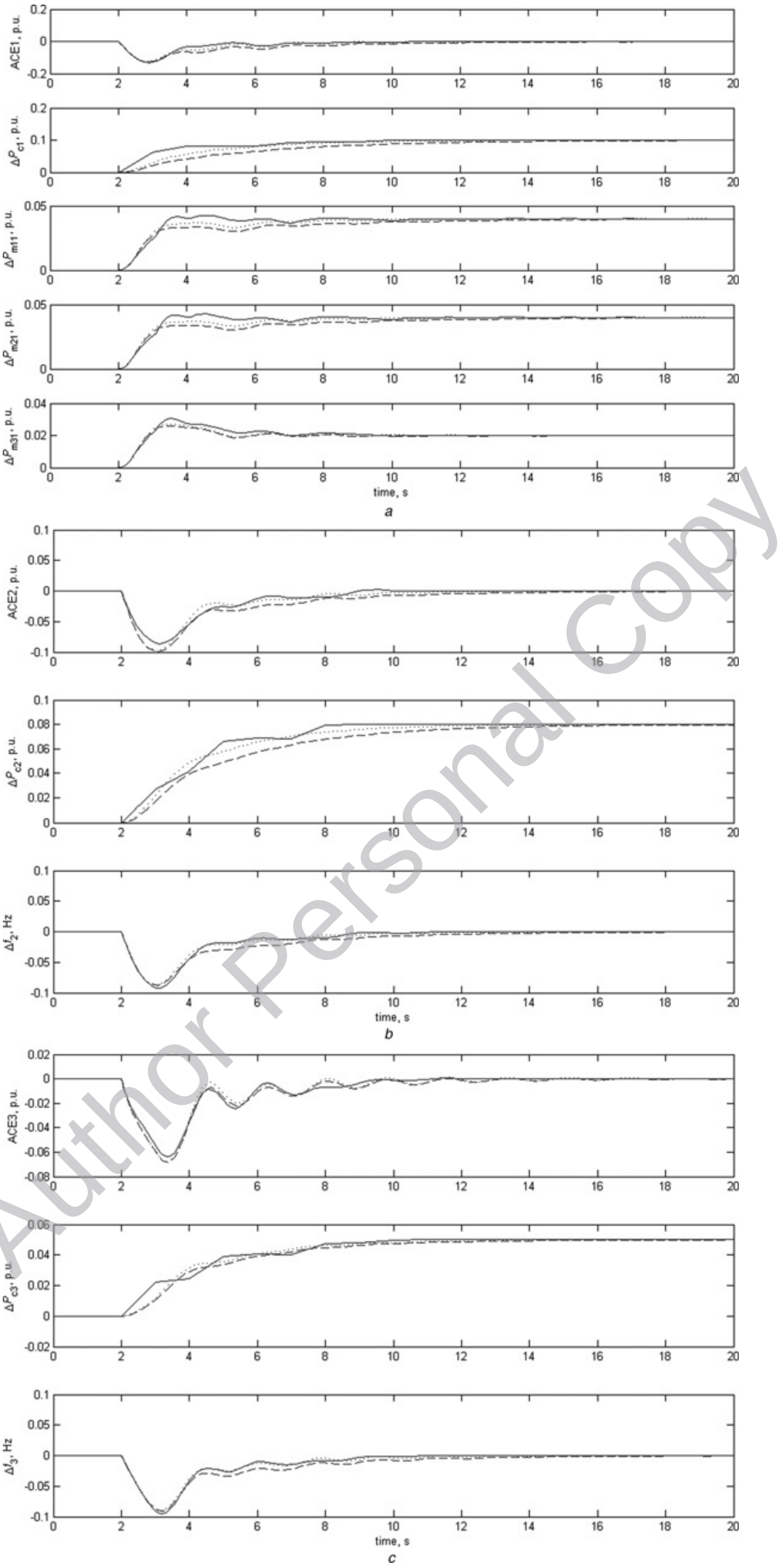


Figure 7 *System responses in Case 1*

a Area 1
b Area 2
c Area 3
Solid line: proposed method; dotted line: robust PI controller [17]; and dashed line: robust PI controller [18]

doi: 10.1049/iet-gtd.2009.0168

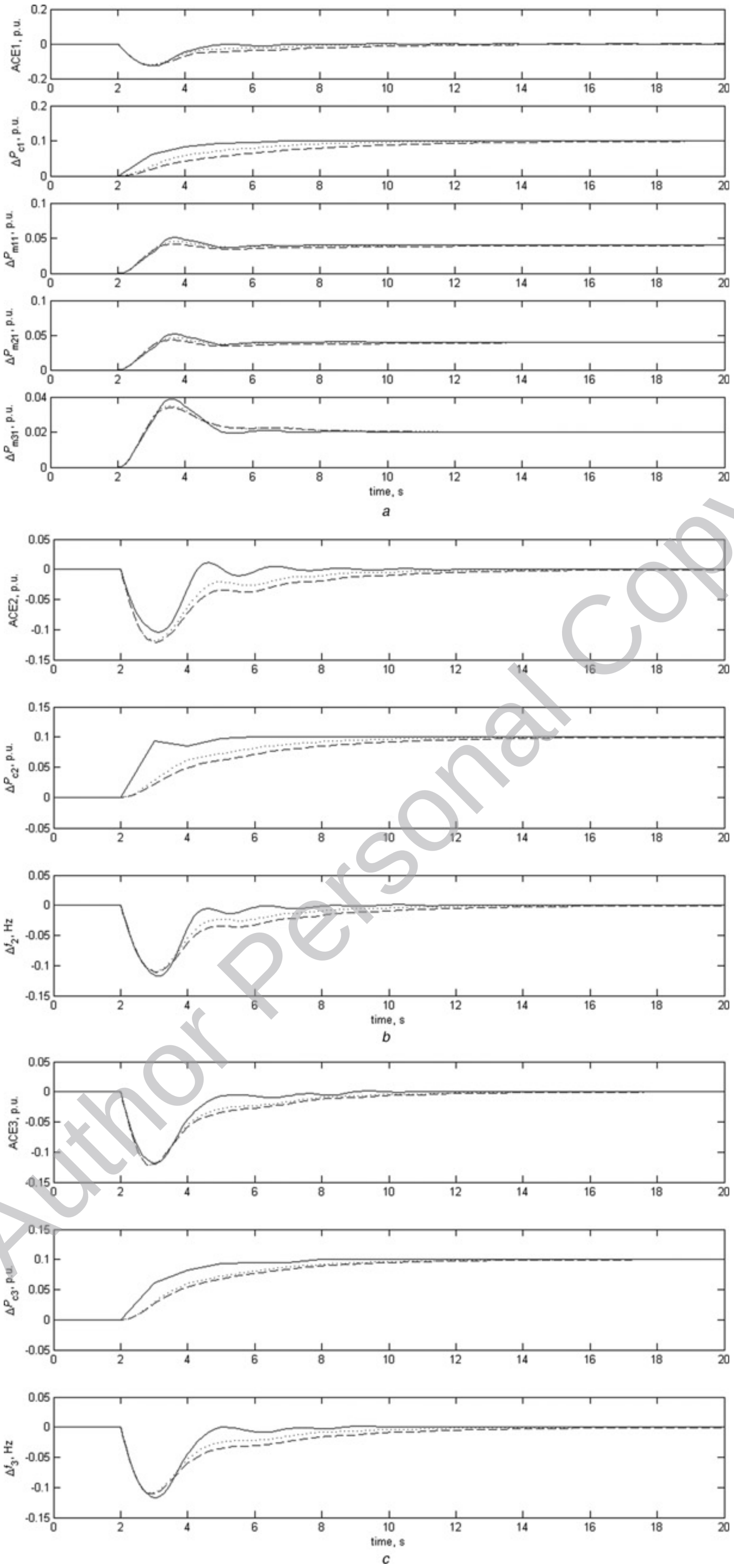


Figure 8 *System responses in Case 2*

a Area 1
b Area 2
c Area 3
Solid line: proposed method; dotted line: robust PI controller [17]; and dashed line: robust PI controller [18]

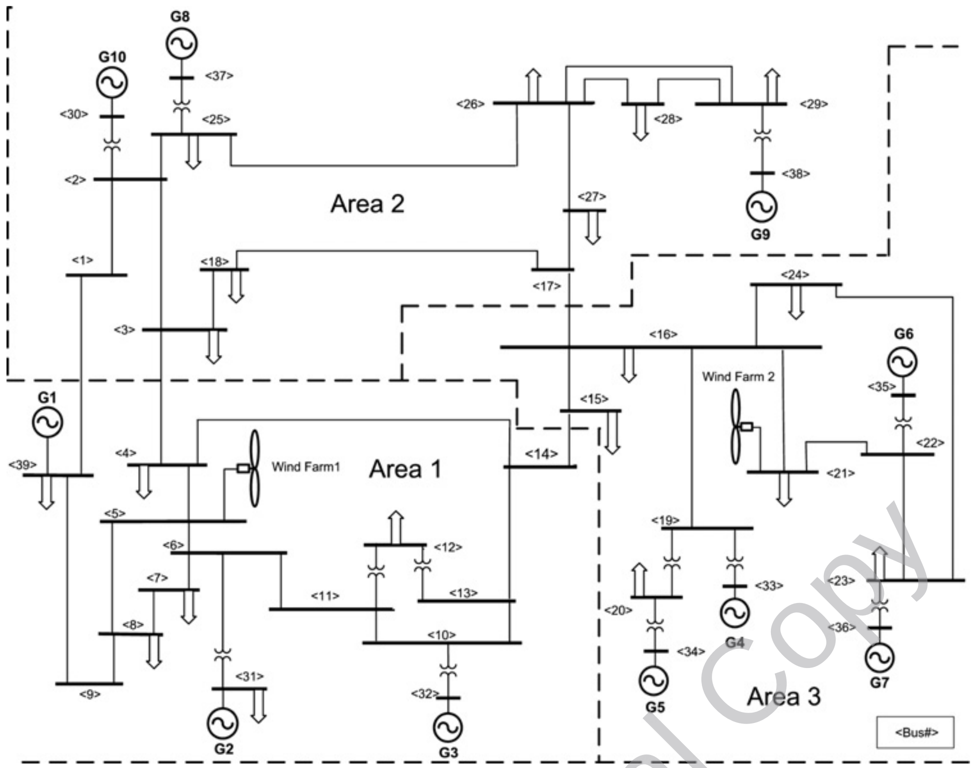


Figure 9 *Non-linear test case study topology*

control techniques, a three-control area power system (same as example used in [17, 18]) is considered as a test system. It is assumed that each control area includes three Gencos and its parameters are given in [17, 18].

A block schematic diagram of the model used for simulation studies is shown in Fig. 5 and the proposed multi-agent structure is shown in Fig. 6.

Our purpose here is essentially to clearly show the various steps in the implementation and to illustrate the method. After the design steps of the application of the algorithm is finished, the controller must be trained by running the simulation in the learning mode as explained in Section 3. The performance results presented here correspond to the performance of the controllers after the learning phase is completed and the controller's actions at various states have converged to their optimal values. The simulation is run as follows: during the simulation, the estimator agent estimates the β parameter based on the input parameters (Δf, ΔP_{tie}, ΔP_{m}, ΔP_{L}) every 60 s (according to Section 3.2); also, the ACE signal is calculated by the estimator agent at each simulation sample time based on the estimated β parameter until that time, then at each LFC execution period (that is greater than the simulation sample time and is equal to 2 s), the controller agent of each area, averages all corresponding ACE signal instances calculated by the estimator agent (based on the estimated β parameter) and averages all load changes instances obtained during the LFC execution period. Three average values of ACE signal instances (each related to one area) plus three average values of load changes instances form the current joint state vector, $\boldsymbol{x}_k$ (that is obtained according to the quantised states gained from GA); then the controller agents choose an action $\boldsymbol{a}_k$ according to the quantised actions gained from GA and the exploration policy mentioned above. Each joint action $\boldsymbol{a}_k$ consists of three actions (ΔP_{c1}, ΔP_{c2}, ΔP_{c3}) to change the set points of the governors. Using this change to the governors setting, the power system model is integrated for the next LFC execution period. During the next cycle (i.e. till the next instant of LFC gained), three values of average ACE instances and average load changes instances in each area are formed in the next joint state ($\boldsymbol{x}_{k+1}$).

In the simulation studies presented here, the input variable is obtained as follows: At each LFC execution period, average values of ACE signal instances corresponding to each area are calculated, they are the first state variables. $(x^1_{\text{avg}1}, x^1_{\text{avg}2}, x^3_{\text{avg}3})$ The average value of load changes instances of three areas at that LFC execution period are the second state variables $(x^2_{\text{avg}1}, x^2_{\text{avg}2}, x^2_{\text{avg}3})$. Because the MARL process is used and agents of all areas are learning together, the joint state vector consists of all state vectors of three areas, the joint action vector consists of all action vectors of three areas and, as shown in triple $\langle (X_1, X_2, X_3), (A_1, A_2, A_3), p, \ r\rangle$ or $\langle X, A, p, r\rangle$, where $X_i = (x^1_{\text{avg}\,i}, x^2_{\text{avg}\,i})$ is the discrete set of each area states and X is the joint state, A_i is the discrete sets of each area actions available to the area i and A is the joint action.

doi: 10.1049/iet-gtd.2009.0168

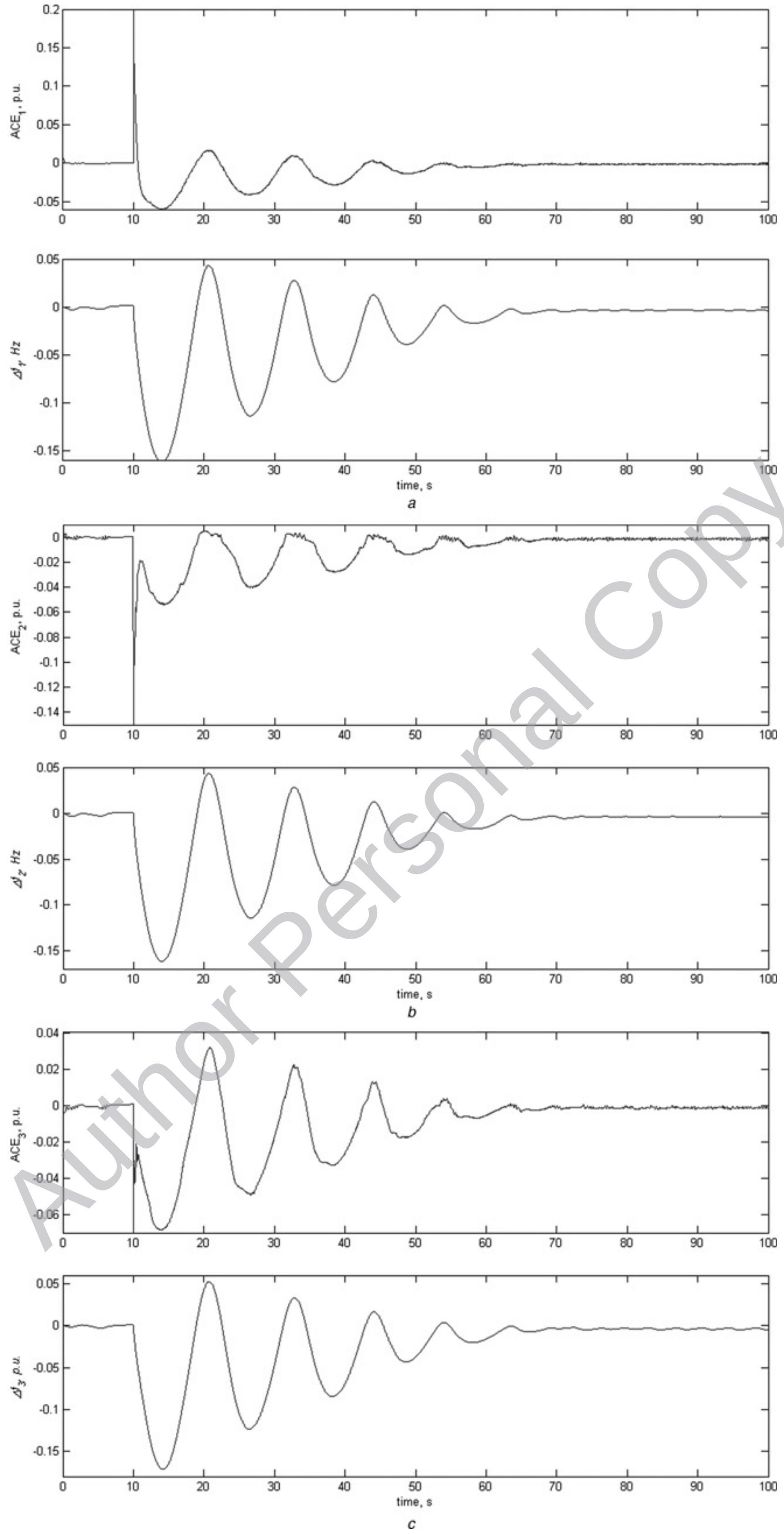


Figure 10 *System responses to a network with the same topology as IEEE 10 Generators 39 Bus with the proposed method*

a Area 1
b Area 2
c Area 3

In each LFC execution period after averaging of ACE_i and $\Delta P_{\mathrm{L}i}$ of all areas (over instances obtained in that period), depending on the current joint state (X_1, X_2, X_3). the joint action $(\Delta P_{c1}, \Delta P_{c2}, \Delta P_{c3})$ is chosen according to the exploration policy. Consequently, the reward r also depends on the joint action whenever the next state (X) is desirable (i.e. when all $|\mathrm{ACE}_i|$ are less than ε, where ε is the smallest ACE signal value that the LFC can operate) and then reward function r is assigned a zero value. When the next state is undesirable (i.e. when least one $|\mathrm{ACE}_i|$ is greater than ε), then r is assigned an average value of all $-|\mathrm{ACE}_i|$

Here, since all power system areas are connected through a tie-line power, and according to (6), the tie-line power changes because of ACE signal changes, therefore agents of all areas can't operate independently; thus, single-agent RL is not capable of solving this problem.

The MARL speeds up the learning process in this problem. Also, this RL algorithm is more scalable than the single-agent RL.

5 Simulation results

To demonstrate the effectiveness of the proposed control design, some simulations were carried out. In these simulations, the proposed controllers were applied to the three-control area power system described in Fig. 6 with the assumptions that the parameters D and H are known, time invariant and fix parameters. Also, D and H's area values are a linear combination of all generator's D and H values in that area.

In this section, the performance of the closed-loop system using the linear robust PI controllers [17, 18] compared to the designed MARL controller will be tested for the various possible load disturbances.

Case 1: As the first test case, the following large load disturbances (step increase in demand) are applied to three areas

$$\Delta Pd_1 = 100\,\mathrm{MW},\ \Delta Pd_2 = 80\,\mathrm{MW},\ \Delta Pd_2 = 50\,\mathrm{MW}$$

The frequency deviation (Δf), ACE and control action (ΔP_{c}) signals of the closed-loop system are shown in Fig. 7 (because the frequency is the same in all areas, it is not necessary to show Δf_1).

Case 2: Consider larger demands by areas 2 and 3, that is

$$\Delta Pd_1 = 100\,\mathrm{MW},\ \Delta Pd_2 = 100\,\mathrm{MW},\ \Delta Pd_2 = 100\,\mathrm{MW}$$

The closed-loop responses for each control area are shown in Fig. 8 (because the frequency is the same in all areas, it is not necessary to show Δf_1).

Using the proposed method with the β estimation, the ACE and frequency deviation of all areas are properly driven back to zero, as well as robust controllers. Also, the convergence speed of the frequency deviation and the ACE signal to its final values are good; they attain to the steady state as rapidly as the signals in [17, 18]. However, the maximum frequency deviation occurs at 2 s in which load disturbances occur.

As shown in the above figures, the generation control signal deviation (ΔP_{c}) change is low and it smoothly goes to the steady state and satisfies the system physical conditions well. Also, it is clear that the ΔP_{m} (mechanical power deviation) is proportional to the participation factor of each generator precisely.

Case 3: As another test case, the proposed method was applied to a network with the same topology as IEEE 10 Generators 39 Bus System [42] as a non-linear test case study (Fig. 9) and was tested with the following load change scenario (more explanations on the network are given in Appendix): In area 1, 3.8% of total area load at bus 8, 4.3% of total area load at bus 3 in area 2, and 6.4% of total area load at bus 16 in area 3 have been added.

The closed-loop responses for each control area are shown in Fig. 10.

As shown in the simulation results, using the proposed method, the ACE and frequency deviation of all areas are properly driven close to zero. Furthermore, assuming that the proposed algorithm is an adaptive algorithm and is based on the learning methods – in each state, it finds the local optimum solution so as to gain the system objectives (the ACE signal near zero) – therefore the intelligent controllers provide smoother control action signals.

6 Conclusion

A new method for the LFC, using an MARL based on GA optimisation and with β estimation functionality, has been proposed for a large-scale power system. The proposed method was applied to a three-control area power system and was tested with different load change scenarios. The results show that the new algorithm performs very well, compares well with the performance of recently designed linear robust controllers, and, finding the ACE signal based on the β estimation, improves its performance. The two important features of the new approach: model independence and flexibility in specifying the control objective, make the approach very suitable for this kind of applications. However, the scalability of the MARL to realistic problem sizes is one of the main reasons to use it. In addition to the scalability and the benefits owing to the distributed nature of the multi-agent solution, such as parallel computation, multiple RL agents may utilise new benefits from sharing experience, for example by communication, teaching or imitation.

doi: 10.1049/iet-gtd.2009.0168

7 References

[1] JALEELI N., EWART D.N., FINK L.H.: 'Understanding automatic generation control', *IEEE Trans. Power Syst.*, 1992, **7**, (3), pp. 1106–1112

[2] IBRAHEEM K.P., KOTHARI P.: 'Recent philosophies of automatic generation control strategies in power systems', *IEEE Trans. Power Syst.*, 2005, **20**, (1), pp. 346–357

[3] ATHAY T.M.: 'Generation scheduling and control', *Proc. IEEE*, 1987, **75**, (12), pp. 1592–1606

[4] KOTHARI M.L., NANDA J., KOTHARI D.P., DAS D.: 'Discrete mode AGC of a two area reheat thermal system with new ACE', *IEEE Trans. Power Syst.*, 1989, **4**, pp. 730–738

[5] SHOULTS R.R., JATIVA J.A.: 'Multi area adaptive LFC developed for a comprehensive AGC simulator', *IEEE Trans. Power Syst.*, 1991, **8**, pp. 541–547

[6] MOON Y.H., RYU H.S.: 'Optimal tracking approach to load frequency control in power systems'. IEEE Power Engineering Society Winter Meeting, 2000

[7] ALRIFAI M., ZRIBI M.: 'A robust decentralized controller for power system load frequency control'. 39th Int. Universities Power Engineering Conf., 2004, vol. 1, pp. 794–799

[8] HIYAMA T.: 'Design of decentralised load–frequency regulators for interconnected power systems', *IEE Proc. C Gener. Transm. Distrib.*, 1982, **129**, pp. 17–23

[9] FELIACHI A.: 'Optimal decentralized load frequency control', *IEEE Trans. Power Syst.*, 1987, **2**, pp. 379–384

[10] LIAW C.M., CHAO K.H.: 'On the design of an optimal automatic generation controller for interconnected power systems', *Int. J. Control*, 1993, **58**, pp. 113–127

[11] WANG Y., ZHOU R., WEN C.: 'Robust load–frequency controller design for power systems', *IEE Proc. C Gener. Transm. Distrib.*, 1993, **140**, (1), pp. 11–16

[12] LIM K.Y., WANG Y., ZHOU R.: 'Robust decentralised load frequency control of multi-area power systems', *IEE Proc. Gener. Transm. Distrib.*, 1996, **5**, (143), pp. 377–386

[13] ISHI T., SHIRAI G., FUJITA G.: 'Decentralized load frequency based on H-inf control', *Electr. Eng. Jpn.*, 2001, **3**, (136), pp. 28–38

[14] KAZEMI M.H., KARRARI M., MENHAJ M.B.: 'Decentralized robust adaptive-output feedback controller for power system load frequency control', *Electr. Eng. J.*, 2002, **84**, pp. 75–83

[15] EL-SHERBINY M.K., EL-SAADY G., YOUSEF A.M.: 'Efficient fuzzy logic load–frequency controller', *Energy Convers. Manage.*, 2002, **43**, pp. 1853–1863

[16] BEVRANI H., HIYAMA T.: 'Robust load–frequency regulation: a real-time laboratory experiment', *Opt. Control Appl. Methods*, 2007, **28**, (6), pp. 419–433

[17] RERKPREEDAPONG D., HASANOVIC A., FELIACHI A.: 'Robust load frequency control using genetic algorithms and linear matrix inequalities', *IEEE Trans. Power Syst.*, 2003, **2**, (18), pp. 855–861

[18] BEVRANI H., MITANI Y., TSUJI K.: 'Robust decentralised load–frequency control using an iterative linear matrix inequalities algorithm', *IEE Proc. Gener. Transm. Distrib.*, 2004, **3**, (151), pp. 347–354

[19] KARNAVAS Y.L., PAPADOPOULOS D.P.: 'AGC for autonomous power system using combined intelligent techniques', *Electr. Power Syst. Res.*, 2002, **62**, pp. 225–239

[20] DEMIROREN A., ZEYNELGIL H.L., SENGOR N.S.: 'Automatic generation control for power system with SMES by using neural network controller', *Electr. Power Comp. Syst.*, 2003, **31**, (1), pp. 1–25

[21] XIUXIA D., PINGKANG L.: 'Fuzzy logic control optimal realization using GA for multi-area AGC systems', *Int. J. Inf. Technol.*, 2006, **12**, (7), pp. 63–72

[22] ATIC N., FELIACHI A., RERKPREEDAPONG D.: 'CPS1 and CPS2 compliant wedge-shaped model predictive load frequency control'. IEEE Power Engineering Society General Meeting, 2004, vol. 1, pp. 855–860

[23] ERNST D., GLAVIC M., WEHENKEL L.: 'Power system stability control: reinforcement learning framework', *IEEE Trans. Power Syst.*, 2004, **19**, (1), pp. 427–436

[24] AHAMED T.P.I., RAO P.S.N., SASTRY P.S.: 'Reinforcement learning controllers for automatic generation control in power systems having reheat units with GRC and dead-band', *Int. J. Power Energy Syst.*, 2006, **26**, (2), pp. 137–146

[25] AHAMED T.P.I., RAO P.S.N., SASTRY P.S.: 'A reinforcement learning approach to automatic generation control', *Electr. Power Syst. Res.*, 2002, **63**, pp. 9–26

[26] EFTEKHARNEJAD S., FELIACHI A.: 'Stability enhancement through reinforcement learning: load frequency control case study', *Bulk Power Syst. Dyn. Control-VII*, Charlston, USA, 2007

[27] AHAMED T.P.I.: 'A neural network based automatic generation controller design through reinforcement learning', *Int. J. Emerging Electr. Power Syst.*, 2006, **6**, (1), pp. 1–31

doi: 10.1049/iet-gtd.2009.0168

[28] SUTTON R.S., BARTO A.G.: 'Reinforcement learning: an introduction' (MIT Press, Cambridge, MA, 1998)

[29] BUSONIU L., BABUSKA R., DE SCHUTTER B.: 'A comprehensive survey of multi-agent reinforcement learning', *IEEE Trans Syst. Man. Cybern. C: Appl. Rev.*, 2008, **38**, (2), pp. 156–172

[30] WEISS G. (ED.):'Multi-agent systems: a modern approach to distributed artificial intelligence' (MIT Press, Cambridge, MA, 1999)

[31] WOOLDRIDGE M., WEISS G. (EDS.):'Intelligent agents, in multi-agent systems' (MIT Press, Cambridge, MA, 1999), pp. 3–51

[32] VLASSIS N.: 'A concise introduction to multi-agent systems and distributed AI'. Technical Report Fac. Sci. Univ. Amsterdam, Amsterdam, The Netherlands, 2003

[33] YANG G.: 'Multi-agent reinforcement learning for multi-robot systems: a survey'. Technical report, CSM-404, 2004

[34] BEVRANI H.: 'Real power compensation and frequency control', 'Robust power system frequency control' (Springer Press, 2009, 1st edn.), pp. 15–41

[35] THATHACHAR M.A.L., HARITA B.R.: 'An estimator algorithm for learning automata with changing number of actions', *Int. J. Gen. Syst.*, 1988, **14**, (2), pp. 169–184

[36] HOONCHAREON N.B., ONG C.M., KRAMER R.A.: 'Feasibility of decomposing ACE to identify the impact of selected loads on CPS1 and CPS2', *IEEE Trans. Power Syst.*, 2002, **22**, (5), pp. 752–756

[37] CHANG-CHIEN L.R., HOONCHAREON N.B., ONG C.M., KRAMER R.A.: 'Estimation of β for adaptive frequency bias setting in load frequency control', *IEEE Trans. Power Syst.*, 2003, **18**, (2), pp. 904–911

[38] CHANG-CHIEN L.R., ONG C.M., KRAMER R.A.: 'Field tests and refinements of an ace model', *IEEE Trans. Power Syst.*, 2002, **18**, (2), pp. 898–903

[39] CHANG-CHIEN L.R., LIN Y.J., WU C.C.: 'An online approach to allocate operating reserve for an isolated power system', *IEEE Trans. Power Syst.*, 2007, **22**, (3), pp. 1314–1321

[40] BEVRANI H.: 'Frequency control in emergency conditions', 'Robust power system frequency control', (Springer Press, 2009, 1st edn.), pp. 165–168

[41] CHONG E.K.P., ZAK S.H.: 'An introduction to optimization' (John Wiley & Sons Press, New York, 1996)

[42] PAI M.A.: 'Energy function analysis for power system stability' (Kluwer Academic Publishers, 1989)

[43] DANESHMAND P.: 'Power system frequency control in the presence of renewable energy sources'. MSc dissertation, Department of Electrical and Computer Engineering, University of Kurdistan Sanandaj, Iran, 2009

8 Appendix

To investigate the capability of the proposed control method on the power system (particularly on the system frequency), a simulation study that has been provided in the Simpower environment of MATLAB software and presented in [43] is used. It is a network with the same topology as the well-known IEEE 10 generators 39-bus test system.

This test system is widely used as a standard system for testing of new power system analysis and control synthesis methodologies. This system has 10 generators, 19 loads, 34 transmission lines and 12 transformers and it is updated by two wind farms in areas 1 and 3, as shown in Fig. 9.

The 39 buses system is organised into three areas. The total system installed capacities are 841.2 MW of conventional generation and 45.34 MW of wind power generation. There are 198.96 MW of conventional generation, 22.67 MW of wind power generation and 265.25 MW of load in area 1. In area 2, there are 232.83 MW of conventional generation, and 232.83 MW of load. In area 3, there are 160.05 MW of conventional generation, 22.67 MW of wind power generation and 124.78 MW of load.

The simulation parameters for the generators, loads, lines and transformers of the test system are given in [43]. All power plants in the power system are equipped with a speed governor and a power system stabiliser (PSS). However, only one generator in each area is responsible for the LFC task; G1 in area 1, G9 in area 2 and G4 in area 3.

doi: 10.1049/iet-gtd.2009.0168