A new intelligent LFC design in a deregulated environment

Daneshfar F, Bevrani H

Published (to be published) in: 20th Iranian Conf. on Electrical Engineering ICEE-2012

(Expected) publication date: 2012

A New Intelligent LFC Design in a Deregulated Environment

Fatemeh Daneshfar and Hassan Bevrani
Department of Electrical and Computer Engineering
University of Kurdistan
Sanandaj, Iran
daneshfar@ieee.org

Abstract— **In this paper, an intelligent approach based on XCSR (accuracy-based learning classifier system with continuous-valued inputs) method is proposed for the load-frequency control (LFC) system using a modified traditional frequency response model suitable for a bilateral-based deregulation policy. Model independency and flexibility in specifying the control objectives; cause it as an interesting solution for the LFC design in new power system environment.**

To demonstrate the capability of the proposed solution, a simulation on a 3-area power system with possible contract scenarios is given.

Keywords- Load frequency control; learning classifier systems; XCSR; deregulated environment

I. INTRODUCTION

In a deregulated environment, the load-frequency control (LFC) design has an important role to enable power exchanges and to provide better conditions for the electricity trading. It is treated as an essential auxiliary service to keep the electrical system reliability at a suitable level [1]. However usually the load frequency controllers used in the industry are proportional-integral (PI) type that are designed for a specific operating points, these conventional LFC designs are not usable for large-scale power systems with nonlinearities, undefined and uncertain parameters, also if the nature of the disturbance varies, they may not perform as expected [1]. Then as a result, adaptable and flexible controllers like intelligent controllers [2-8] are more suitable than the classical ones, for the LFC problem in deregulated environments.

XCSR is a "continuous-valued input, Learning Classifier System (LCS)". It is one of the intelligent approaches which have received little attention in the area of power system control. It is a machine learning approach for producing adaptive, flexible systems in an unknown environment using reinforcement learning (RL), evolutionary computing and other heuristics [9].

All evolutionary computing approaches like genetic algorithms (GA) mainly are based on searching a problem space by producing and developing an initially random individuals (of solutions) such that fitter individuals (solutions) are generated over time [10]. The RL is also a machine learning technique which the agent (learner) interacts with the environment and learns through mapping state and action combinations to their utility with the aim of being able to maximize future environment reward [10].

XCSR is a system with a knowledge base of rules, where the rules are usually in the RL traditional production system form of "IF state THEN action". Evolutionary computing techniques are used to generate and search the space of legal rules, whilst RL techniques are used to assign rewards for the existing rules, and therefore guiding the search process to find better and fitter rules [10].

In this paper, an XCSR based control structure design is described and applied to a modified dynamical model for a general control area in the deregulated environment with bilateral contract policy introduced in [1]. It has an intelligent controller that receives the area control error (ACE) and its deviation (ΔACE) signals and provides generator set point signal (ΔP_c), using XCSR method; then it is distributed among the different units under control using fixed participation factors.

The above technique has been applied to the LFC problem in a three-control area power system as a case study. In the performed simulation, the test power system is considered as a collection of control areas interconnected with high-voltage transmission lines (tie-lines).

The organization of the rest of the paper is as follows. In Section 2, a brief discussion on a test system for LFC synthesis problem is given. An explanation on XCSR method and how a load–frequency controller can work within this formulation is provided in Section 3. In Section 4, a case study of a three-control area power system, for which the above architecture is implemented for, is discussed. Simulation results are provided in Section 5, and paper is concluded in Section 6.

II. TEST SYSTEM

Here, to illustrate the effectiveness of the proposed control strategy for LFC design, a generalized dynamical model for a control area in restructured power system is used [1]. The modified LFC block diagram for control area i can be obtained as shown in Fig. 1. A power system in a deregulated environment includes separate generation, transmission and distribution companies with an open access policy.

In an open energy market, a distribution company (Disco) has the freedom to contract with any available generation company (Genco) in its own or another control area (there can be various

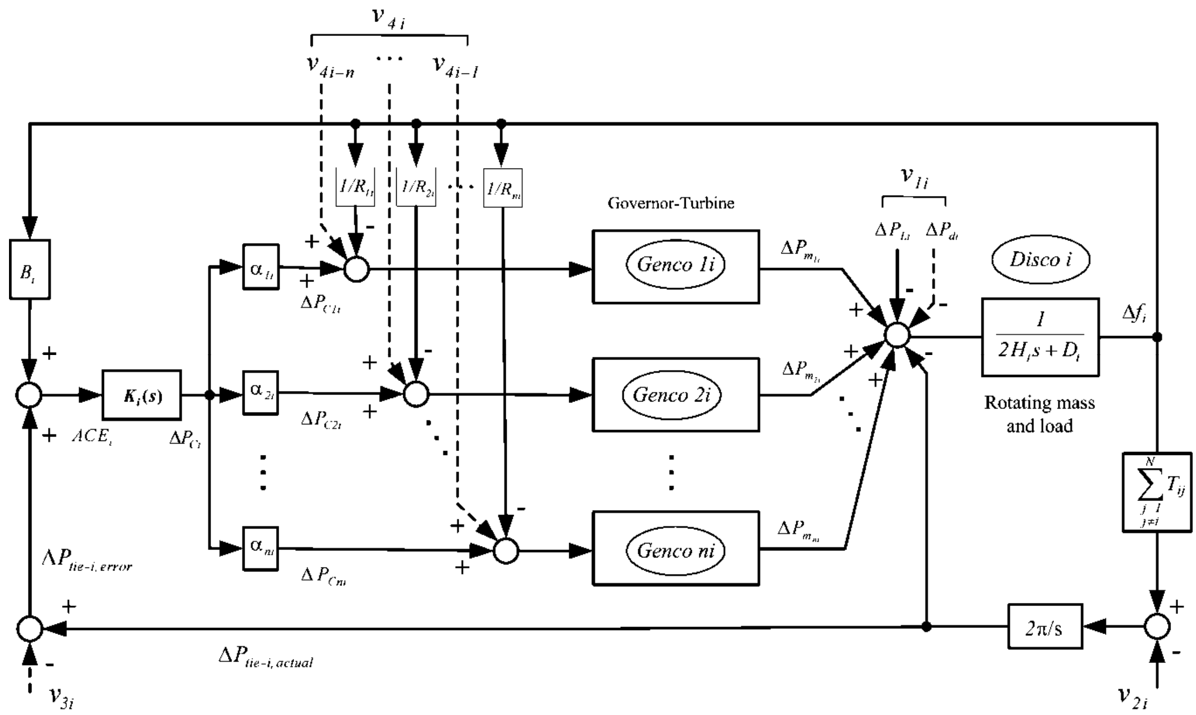

Fig. 1 LFC structure in deregulated environment [1]

combinations of contracts between each Disco and available Gencos) [1]. It is assumed that each Disco is responsible for tracking its own load and honoring tie-line power exchange contracts with its neighbors.

In a restructured environment the bidding process is as follow, the Gencos submit their ramp rates (megawatts per minute) and bids to the market operator. After bidding evaluation and approving the ramp rates by the responsible organization, those Gencos selected to provide regulation services must perform their functions according to their contract [11].

As shown in Fig. 1, each control area has its own AGC synthesis which is responsible for tracking its own load and honoring tie-line power exchange contracts with its neighbors. The signals in the model identify which Genco has to follow a load demanded by a specified Disco and the scheduled flow over the tie lines must be adjusted by demand signals of those distribution control areas having a contract with Gencos out of its boundaries [1].

More explanations on the model and related equations have been described in detail in [1], [11].

III. THE XCSR APPROACH

As it is shown in Fig. 2, XCS system consists of following elements [12]:

- A reinforcement learning component which the XCS is in interaction with environment via its detectors to get the necessary inputs and its effectors to do action (a_i). Also the environment gives the XCS, a scalar reward at different times,
- A box labeled [*P*], including classifiers (rules) population,
- A match set [*M*], which consisting the rules of [*P*] that match the input,
- A genetic algorithm component which acts on [*M*] classifiers,
- A system prediction array $P(a_i)$, which is an array that explain each action prediction, represented in [*M*],
- An action set [*A*], consisting of the [*M*] rules which their actions advocating the chosen action of $P(a_i)$. The selected action is sent to the effectors and a reward is returned by the environment.
- Previous action set $[A]_{-1}$ will be updated, and in each time a new reward signal is provided by the environment.

Following, the above components functionality are described in detail.

A. Action Selection

The population matrix [*P*] contains fixed maximum size of *N* classifiers [13]. For the current application, *N* is equal to 200 rules and the population matrix initialized with no classifiers.

Each classifier presents a general shape of an RL rule: “IF condition THEN action”. In this application, condition parameters assumed to be, ACE and its deviation signal, and action parameter is set to ΔP_c. However, the mentioned parameters’ values are real, then the continuous-valued version of XCS (XCSR) that is provided for real-valued parameters has

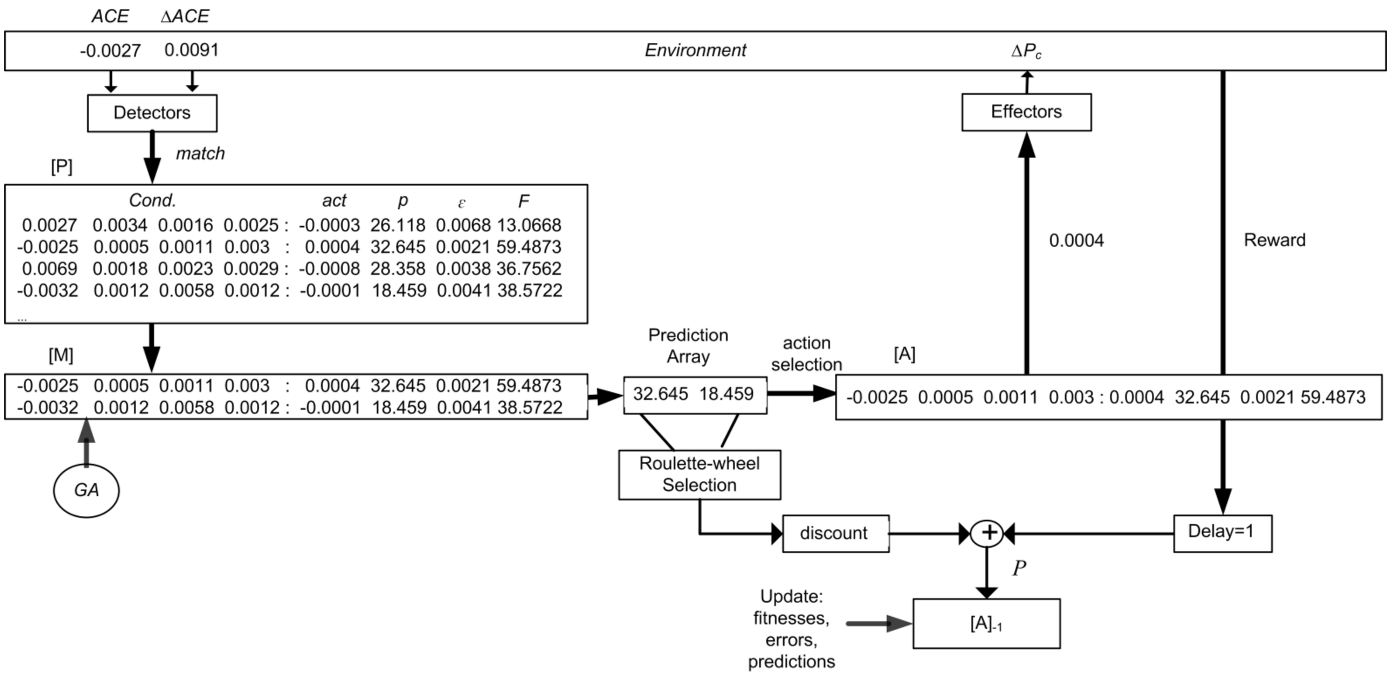

Fig. 2 XCS and its components [13]

been used [14]. In this method, the classifier condition is a concatenation of intervals [14]:

$$Interval_i = (c_i ; s_i) \tag{1}$$

where c_i and s_i are real numbers. The c_i is the center value of $Interval_i$ and the s_i is a delta defined concerning c_i [14]. Table I shows all the detailed classifiers' parameters [13].

At each LFC execution period (that is greater than the simulation time window and is equal to 200 samples), the average values of ACE signal and ΔACE within that period, are separately sent to the XCSR learning core. The new obtained values are compared with all the classifiers conditions of [*P*]. If there is no similar rows, a new classifier with the parameters specified in Table I, will be added to [*P*] and [*M*]. However, if there are similar condition classifiers and input, the matching classifiers will be added to the match set [*M*] (see Fig. 2) and a $P(a_i)$ array will be formed for classifiers' actions represented in [*M*], according to a fitness-weighted average of the predictions of classifiers advocating action *a* [15]. Then an action will be selected from $P(a_i)$ based on the roulette-wheel action selection (the action selection is proportional to $P(a_i)$). The selected action arranges the action set [*A*] including all classifiers in [*M*] has the selected action [13]. The chosen action (ΔP_c value) is then sent to the effectors (LFC system) and when an LFC execution period for the selected action is done, an immediate scalar reward according to (2) is calculated and sent back to the XCSR.

$$reward = -\left(\left|\sum_{i=0}^{200} ACE_i\right| - \left|\Delta\sum_{i=0}^{200} ACE_i\right|\right) \tag{2}$$

The reward function is a measure of the action performance on how it decreases the *ACE* signal distance to zero.

B. Reinforcement Learning

The RL component role is followed after a reward signal received from the power system. It updates the *p*, ε and *F* parameters of classifiers action set $[A]_{-1}$, which was selected in the previous LFC execution period as follows [13]:

The selected previous action, is multiplied by a discounting factor $\gamma = 0.71$, and is added to the received reward from the previous time-step. As it is shown in Fig. 2, the resulting value is called *P* that is used to adjust the classifiers' predictions and prediction error in $[A]_{-1}$ [13].

The *F*, ε and *p* parameters of each classifier are updating respectively according to (3) and (4) [13].

$$p_j = p_j + \beta(P - p_j) \tag{3}$$
$$\varepsilon_j = \varepsilon_j + \beta(\left|P - p_j\right| - \varepsilon_j)$$

$$\begin{cases} k_j = \exp\left[(\ln\alpha)(\varepsilon_j - \varepsilon_0)/\varepsilon_0\right] \quad for \quad \varepsilon_j > \varepsilon_0 \quad otherwise \quad 1 \\ k'_j = k_j \Big/ \sum_{j=1}^{size\ of\ [A]_{-1}} k_j \\ F_j = F_j + \beta\left(k'_j - F_j\right) \end{cases} \tag{4}$$

where α and ε_0 are the classifier's accuracy parameters [13] and are fixed at 0.1, 0.01, respectively. β is the learning rate parameter and its value is 0.2.

More explanation on how the above relations are created can be found in [13].

C. *Genetic Algorithm Component*

When a match set is created, the XCSR computes its classifiers average time-stamp, and runs the GA if the difference between the obtained average and the current CPU time exceeds a threshold (for the current application the threshold is set to 0.25) [15]. Then GA component selects a classifier from [M] with the roulette-wheel selection (according to classifiers' fitness), and performs mutation on its first five elements with probability $\mu = 0.08$ (here for the simplicity, the cross over operator was not implemented). Finally, the selected classifier time-stamp parameter is modified to the current CPU time.

After doing of mutation, the new offspring (classifier) will be added to the [P] if sum of the N parameters of all classifiers is less than 200; otherwise, a classifier is deleted proportional to its inverse fitness probability and then the new classifiers will be added [13].

TABLE I: Classifier Parameters

c_{ACE}	The center value of ACE (is initialized to the input ACE value)
s_{ACE}	The delta defined for ACE (is initialized to a random delta value for ACE signal)
$c_{\Delta ACE}$	The center value of ΔACE (is initialized to the input ΔACE value)
$s_{\Delta ACE}$	The delta value of ΔACE (is initialized to a random delta value for ΔACE signal)
a_i	A real value for ΔPc (is initialized to a random value for the action)
P	Classifier prediction: the environment reward average when that classifier's action controlled the system (is initialized to a random value)
□	Classifier prediction error: an average of a measure of the error in the prediction parameter (is initialized to a random value)
F	Classifier fitness: an inverse function of the prediction error (is initialized to a random value)
N	The number of classifiers has the same conditions to the existing classifier (is initialized to 1)
T	Timestamp: The genetic algorithm (GA) is run in a match set if the average number of classifier time-steps since the last GA in that match set exceeds a threshold (is initialized to the CPU time.)

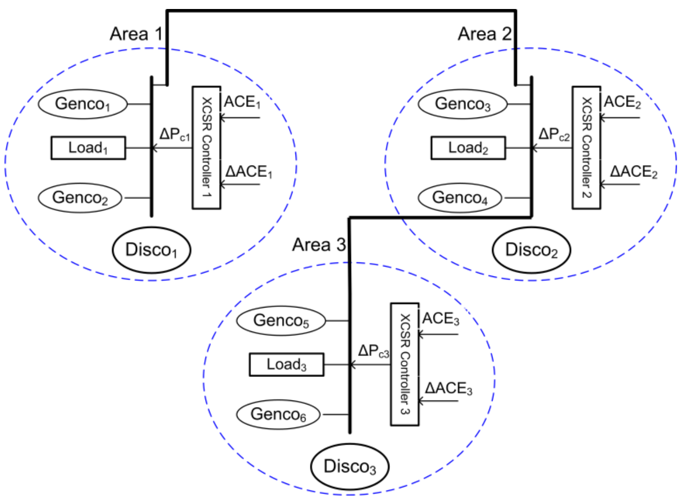

Fig. 3 Three-control area power system with intelligent controllers

IV. Application To A 3-Control Area Power System

To demonstrate the effectiveness of the proposed control design, a three-control area power system based on the model described in Section 2 is assumed as a test system (see Fig. 3). In this simulation, each control area includes two Gencos and one Disco which its parameters are given in [11].

V. Simulation Results

To demonstrate the effectiveness of the proposed approach, some simulations for the various possible scenarios of bilateral contracts and load disturbances were carried out to the 3-control area power system with XCSR based controllers described in Section 4.

In this section, the performance of the closed-loop system using the XCSR method in comparison of robust PI controller (ILMI method) [11] is tested for various possible scenarios of bilateral contracts and load disturbances.

A. *Test Case 1*

For the first test case, a large load disturbance (a step increase in demand) is applied to each area as follow:

$$\Delta P_{L1} = 100\ MW,\ \Delta P_{L2} = 70\ MW,\ \Delta P_{L3} = 60\ MW$$

For simplicity, assume the following generation participation matrix (GPM) [1], is used for the bilateral contracts between three areas.

$$GPM = \begin{bmatrix} 0 & 0 & 0.5 \\ 0 & 0 & 0.5 \\ 0 & 0.5 & 0 \\ 0 & 0.5 & 0 \\ 0.5 & 0 & 0 \\ 0.5 & 0 & 0 \end{bmatrix}$$

The above matrix shows the participation factor of each Genco in the considered control areas (each control area is determined by a Disco). The rows and columns of the GPM are corresponded to Gencos and Discos (control areas), respectively [1]. Therefore, according to the above GPM each Disco demand is only sent to its local Gencos.

The simulation results are shown in Figs. 4 to 6. Solid line is used for XCSR based, and dashed line is used for ILMI method [11]. The notations are defined as follows,

- Δf: Frequency deviation,
- ΔP_{tie}: Actual (current) tie-line power flow,
- ACE: Area control error,
- ΔP_m: Actual power changes,
- ΔP_c: The set point to the governor,

As it is seen from Figs. 4 to 6, the XCSR controller increases/decreases the governor set point monotonically until the frequency deviation and area control error signals of all areas driven back to zero, quickly.

According to explanation of the proposed method in Section 3, the XCSR controller averages input ACE and ΔACE signal values during an LFC execution period, then creates a new value for ΔP_c signal. This function causes the step changes to be appeared in the governor set point. Also, the generated power and tie-line power changes (Fig. 4) are properly converge to specified values.

In this case, since there are no contracts between areas, the scheduled steady state power flows over the tie lines are zero.

B. Test Case 2

As second test case, consider larger demand by Disco 2 and Disco 3 as follow:

$$\Delta P_{L1} = 100\ MW,\ \Delta P_{L2} = 100\ MW,\ \Delta P_{L3} = 100\ MW$$

And assume the following GPM which shows Discos contract with the available Gencos in other areas,

$$GPM = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0.25 & 0.75 \\ 0.25 & 0.25 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

System responses are shown in Figs. 7 to 9.

C. Test Case 3

In this test case the performance of the proposed intelligent LFC system is examined against large and random load changes. For this test, consider the GPM of scenario 2 and assume a bounded random load changes (Fig. 10) as an uncontracted local demand, is applied to each control area.

$$-50MW \le \Delta P_{di} \le +50MW$$

As it is shown from Fig. 11 the XCSR controllers track the load fluctuations, effectively.

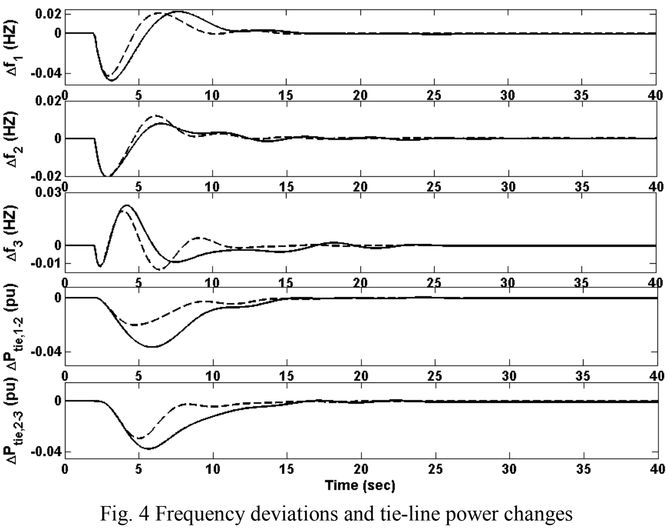

Fig. 4 Frequency deviations and tie-line power changes

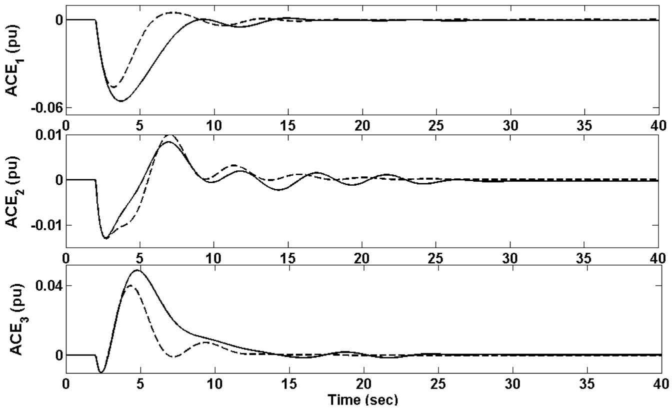

Fig. 5 ACE signals

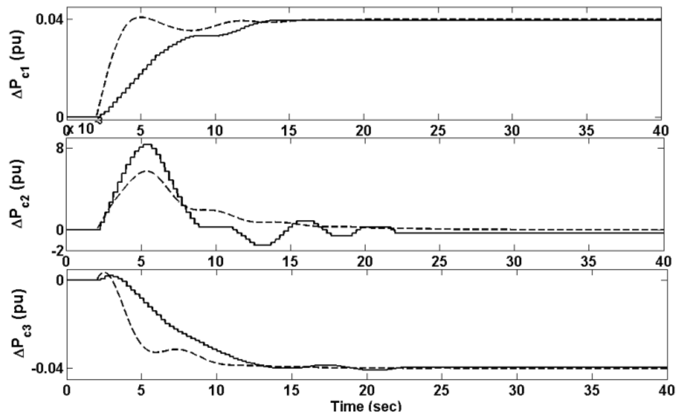

Fig. 6 Area control action signals

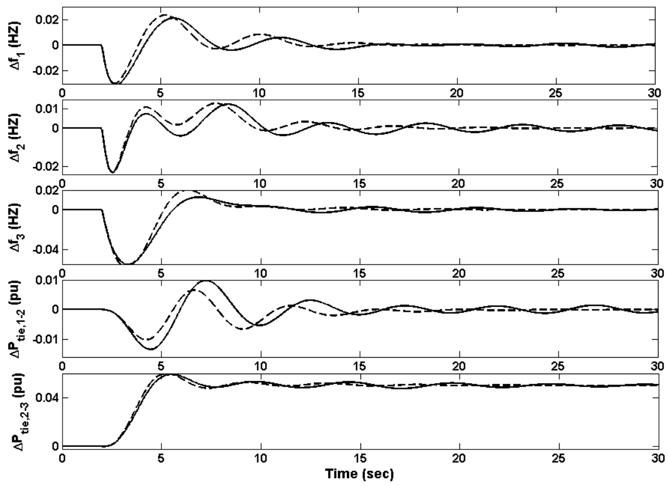

Fig. 7 Frequency deviations and tie-line power changes

The above simulation results show that the proposed intelligent LFC methodology operates as well as powerful Robust ILMI Control technique. Furthermore the higher flexibility, higher degree of intelligence and model independency of the proposed solution for a wide range of load disturbances and possible bilateral contract scenarios are investigated.

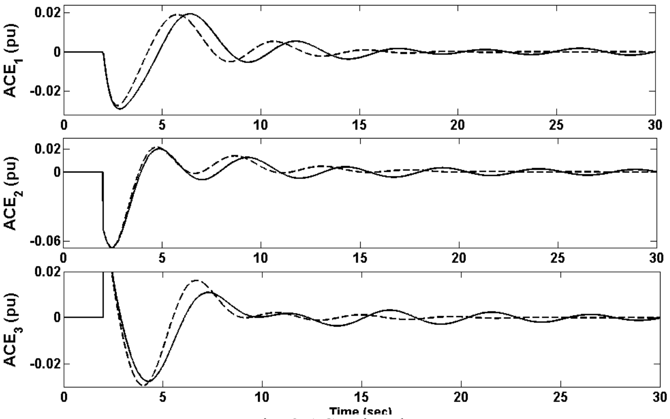

Fig. 8 ACE signals

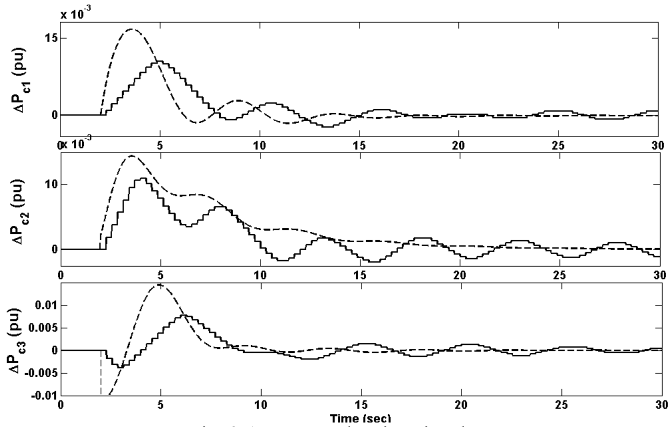

Fig. 9 Area control action signals

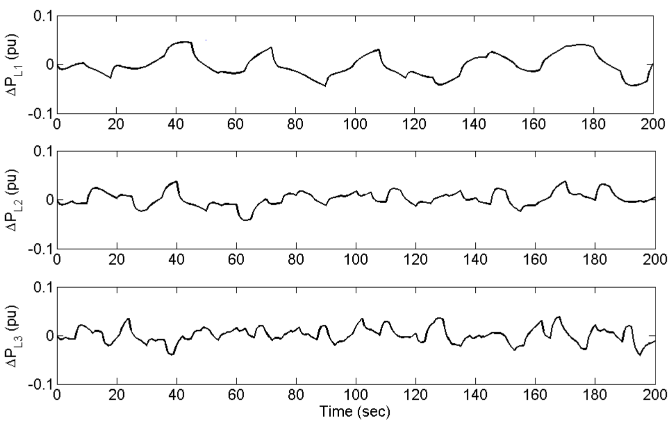

Fig. 10 Random load changes

VI. Conclusion

A new flexible and model independent intelligent method for LFC design in a bilateral based restructured power system is proposed. The proposed method is based on XCSR technique and is applied to a 3-control area power system with different possible scenarios.

The simulation results present a desirable performance under a wide range of load changes and possible contracted scenarios.

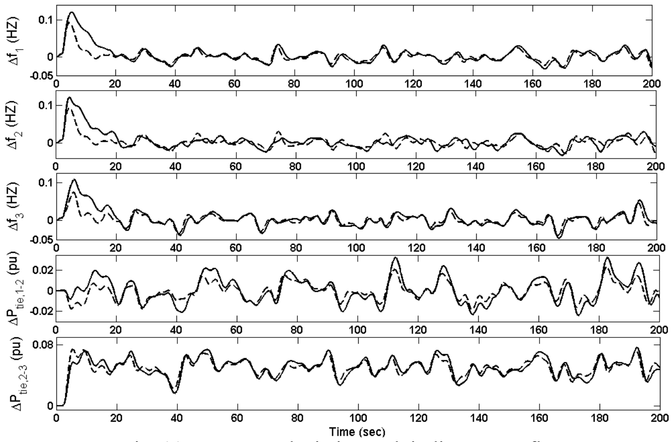

Fig. 11 Frequency deviation and tie-line power flow

References

[1] H. Bevrani, “Frequency Response Characteristics and Dynamic Performance,” In: Robust power system frequency control, Springer Press; 2009, 1st edn. p. 49-59.

[2] F. Daneshfar, and H. Bevrani, ''Load-Frequency Control: A GA-based Multi-agent Reinforcement Learning'', IET Generation, Transmission & Distribution, vol. 4, no. 1, pp. 13-26, 2010.

[3] M. K. El-Sherbiny, G. El-Saady and A.M. Yousef, "Efficient fuzzy logic load-frequency controller”, Energy Convers. Manage., vol. 43, no. 14, pp. 1853-1863, 2002.

[4] F. Daneshfar, H. Bevrani, and F. Mansoori, “Load-Frequency Control: a GA based Bayesian Networks Multi-agent System”, Iranian Journal of Electrical & Electronic Engineering, vol. 7, No. 2, June 2011

[5] D. Rerkpreedapong, A. Hasanovic and A. Feliachi, “Robust load frequency control using genetic algorithms and linear matrix inequalities”, IEEE Trans. Power Syst. vol. 2, no. 18, pp. 855-861, 2003.

[6] H. Bevrani., F. Daneshfar, and P. Daneshmand, ‘Intelligent Power System Frequency Regulations Concerning the Integration of Wind Power Units’ Chapter Book of “Wind Power Systems: Applications of Computational Intelligence” L. Wang et al. (Eds): Wind Power Systems, Green Energy and Technology, pp. 407–437, 2010

[7] H. Bevrani, F. Daneshfar, P. R. Daneshmand, T. Hiyama, “Reinforcement learning based multi-agent LFC design concerning the integration of wind farms”, In: International Conference on Control Applications, CD-ROM, IEEE, Yokohama, 2010

[8] H. Bevrani, F. Daneshfar, P. R. Daneshmand, “Intelligent Automatic Generation Control: Multi-agent Bayesian Networks Approach,” International Conference on Control Applications, CD-ROM, IEEE, Yokohama, 2010

[9] L. Bull, “Learning Classifier Systems: A Brief Introduction” Faculty of Computing Engineering & Mathematical Sciences, University of the West of England, UK, 2003

[10] S. W. Wilson, “ZCS: A zeroth order classifier system. Evolutionary Computation,” Vol. 2, pp. 1-18. 1994

[11] H. Bevrani, Y. Mitani, and K. Tsuji, “Robust AGC: Traditional structure versus restructured scheme,” Trans. of Electrical Engineering in Japan, vol. 124-B, 2004

[12] P_L. Lanzi, W. Stolzmann, and S.W. Wilson, Learning Classifier Systems:From Foundations to Applications. Springer. 2000

[13] S. W. Wilson, “Classifier Fitness Based on Accuracy,” Evolutionary Computation, vol. 3. pp. 149-175, 1995

[14] S. W. Wilson, Get Real! XCS with Continuous-Valued Inputs, Learning Classifier System, Springer, 2000

[15] O. Sigaud, S. W. Wilson, “Learning classifier systems: a survey,” Soft Computing –A Fusion of Foundations, Methodologies and Applications, vol. 11, 2007