2008

Reinforcement learning-based control design for load frequency control

Sara Eftekharnejad
West Virginia University

Reinforcement Learning-Based Control Design for Load Frequency Control

by

Sara Eftekharnejad

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

Professor Muhammad Choudhry, Ph.D.
Professor Powsiri Klinkhachorn, Ph.D.
Professor Ali Feliachi, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2008

Keywords: automatic generation control, load frequency control, NERC, control performance standards, reinforcement learning

ABSTRACT

Reinforcement Learning-Based Control Design for Load Frequency Control

by

Sara Eftekharnejad

Master of Science in Electrical Engineering

West Virginia University

Professor Ali Feliachi, Ph.D., Chair

Energy balance in electric power systems is continuously disrupted by constant demand changes due to customers' switching in and out, or loss of generating units. Load frequency control (LFC) is very essential for interconnected power systems in order to maintain the energy balance which is assessed through the Area Control Error, a signal that is made up of deviations from their nominal values of the system frequency and power area interchanges. Each balancing authority is responsible for its own energy balance in accordance with North American Electric Reliability Corporation (NERC) standards.

This thesis presents a novel approach to the LFC problem. An adaptive intelligent controller, or agent, changes the gains of a proportional-integral (PI) controller based on the operating conditions. The intelligence and decision making is provided by means of a reinforcement learning (RL) algorithms. This approach keeps the simple design of the PI controllers and in the mean time makes them more adaptive and applicable to different disturbances. Moreover, the developed controller can be applied to different systems with various parameters with almost no change in the controller design due to their ability to learn proper settings through interaction with the environment.

Each control authority should comply with NERC control performance standards CPS1 and CPS2. In order to comply with these standards and decrease the control cost, tight control should be prevented. The second approach in this thesis is to design a reinforcement learning based controller that tunes the gains of the PI controller in a way to achieve this goal. Simulations are performed in MATLAB / Simulink to demonstrate performance of all the proposed controllers.

Acknowledgments

Among all who have contributed to my education at the West Virginia University, my greatest appreciation surely belongs to my advisor, Professor Ali Feliachi who guided me through the Master's program. None of my work could be possible without his help and support. I also want to thank my committee members: Prof. Mohammad Choudhry and Prof. Powsiri Klinkhachorn for their precious time and valuable suggestions for the work done in this thesis.

I would also like to thank my friends and colleagues at the Advanced Power and Electricity Research Center (APERC) for their encouragement and help. Many thanks to my Morgantown friends for the wonderful time.

Finally, my Sincere thanks to my family, whose unconditional love and support has been the greatest motivation for me to keep progressing during these years. Specially, I thank my father and my mother, who were my first teachers and their love and encouragement inspired my passion for learning. It is to commemorate their love that I dedicate this thesis to them.

This work was supported in part by grants from the US DEPSCoR/ONR grant No. N00014-03-1-0660 and the US DoE grant No. DE-FC26-06NT42793.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years the structure of electric power systems has changed due to deregulation and increased number of customers. This change has faced the Generation (Genco), Transmission (Transco), and Distribution (Disco) companies with more complex problems regarding control task and compliance with standards. With these complexities, more sophisticated devices are needed to replace the traditional hydraulic and mechanical components. Electronic devices driven with computers are finding more applications in today's power systems. Therefore numerous research investigating the performance of computer applications in power systems have been carried out previously.

In power systems the active power has to be generated at the same time that it is consumed. Any mismatch between the demanded and generated power leads to a power imbalance. This power imbalance causes the system frequency and the tie-line power to deviate from their nominal and scheduled values. The basic role of load frequency control (LFC) is to maintain the megawatt output of a generator in balance with the demand and therefore control the interconnection frequency [2]. This goal is achieved by automatic control of the steam valves or water gates of speed governors to adjust the amount of the steam or water flowing through the turbines. As a result of this control, the mechanical power and thus the generated electrical power is adjusted.

LFC has been the topic of numerous research in the past decades and numerous control techniques have been proposed in literature. However, proportional integral (PI) controllers are more widely used in industry. The gains of these controllers are tuned once a month [4] by trial and error and are not accurate enough to consider all operating conditions. Therefore many studies have been conducted to design adaptive controllers that can be applied to many systems with a wide range of operating conditions. As it will be discussed later in this thesis, most of these methods are based on the detailed model of the system and thus are complex in design. Furthermore, some of these controllers are centralized and need to have access to the information from the entire power system

which makes them less useful in power system applications. This is one of the main drawbacks of the adaptive controllers when they are applied to power systems, as in many cases all the system information are not measurable and available to the designer. Hence, a control method that is not based on the system model and is adaptive, to be applied to different operating conditions, is desirable.

In order to make controllers more adaptive new control techniques are used in control design. Each method is suitable for a specific problem, depending on the nature of the control problem. Artificial neural network (ANN), Genetic algorithm (GA) and fuzzy logic are among the most widely used methods in the literature. However, due to the fact that in a load frequency control problem each control area can have random load changes, many of these methods may not be useful as they require substantial amount of training based on predicted scenarios and specific system parameters. Also in some cases defining the method's required parameters, such as membership functions in the case of fuzzy logic, is a formidable task. Therefore, a learning method that can learn the proper setting of the controller without need for a considerable knowledge of system parameters is more applicable for the LFC problem. This method can be applied to conventional controllers such as PI controllers to make them more adaptive and in the mean time decrease the human interference for tuning their gains.

The primary objective of the LFC is to balance the generation and demand in a way to respond to the needs of customers. This balancing task should be in compliance with the standards defined by North American Electric Reliability Council (NERC) in order to be acceptable. Any unit violating these standards will be penalized by NERC and has to change its settings to comply with these criteria. In February 1977, NERC adopted new compliance performance standards CPS1 and CPS2 to replace the old standards A1 and A2 [23]. These criteria assess characteristics of a control area's "area control error" (ACE). In order to comply with NERC both CPS1 and CPS2 should be satisfied, however; the statistical data from NERC illustrate that some control areas can be highly compliant with CPS1 while violating CPS2. In order to avoid the penalties which are the results of violating the standards, new control techniques based on these standards should be designed.

Although the standards should be satisfied, too tight control of the ACE signals will be costly and can increase unit maneuvering. An ideal control technique should be able to keep the area's performance within the NERC's standards and in the mean time decrease the fuel cost and the rapid movements of the unit equipments. If each area is controlled with this approach in a decentralized manner, they could both balance the generation and demand locally and keep the interconnection power flows within the limits.

The objective of this research is to propose a new control technique that can be applied to solve the LFC problem in conjunction with the widely used PI controllers in industry. The new technique is capable of learning the proper gain settings of the PI controllers and in the mean time reduces the control costs of the overall system. This controller is based on reinforcement learning (RL) methods and is flexible enough to define different control objectives. The proposed strategy is model free and thus applicable to a wide range of systems with various parameters.

This thesis is organized as follows. A literature survey and the problems associated with some previously designed controllers are discussed in Chapter 2. An introduction to reinforcement learning and the method used in this research along with the fundamentals of load frequency control is presented in Chapter 3. Next, in Chapter 4, a new design strategy for PI controller based on reinforcement learning methods is introduced. In Chapter 5 this technique is followed by a new approach in which NERC standards are taken into account while, at the same time control effort is being minimized. Finally, conclusions are given in Chapter 6.

Chapter 2

Literature Survey

2.1. Introduction

Multi agent (MA) control is an emerging field in power systems and has been reported in many applications and some promising results were obtained in several areas including operation, markets, diagnosis and protection. The focus of this research will be on the application of reinforcement learning agents in load frequency control problem. Since this field is almost new to power system applications, other applications of reinforcement learning in power systems should be explored first.

With the increasing complexities of power systems, there is more need for intelligent and learning controllers that can adapt themselves to different operating conditions and learn the proper control actions in case of unpredicted situations. Therefore, making the conventional controllers more intelligent has been investigated in many research studies. Different methods are used in order to achieve this goal. Reinforcement learning (RL) is one of these methods that has recently gained a considerable attention in many fields requiring control. Power systems are also not apart from these areas and RL methods are applied for different problems such as voltage control and automatic generation control.

In this chapter a literature survey is presented as follows: first, the concepts of reinforcement learning agents and their applications in power system are surveyed. Then, selected published work in the area of load frequency control, along with their advantages and drawbacks, are discussed. Finally, the contribution of this thesis in solving LFC problem is discussed.

2.2. Proportional-Integral (PI) Parameter Tuning and Optimization

Proportional-Integral (PI) controllers have widely been used in industry for the purpose of load frequency control. Numerous research studies have therefore concentrated on different techniques to tune the parameters of these controllers. In case of

the load frequency control problem, the objective is to improve the transient performance. These controllers in fact adjust the control signal with the aid of a proportional (K_P) and integral gain (K_I). In general, the equation for the output in the time domain is [5]:

$$u(t) = K_P e(t) + K_I \int e(t)\ dt \tag{2.1}$$

Depending on the signals selected for control, i.e. frequency, area control error (ACE) or tie line power, different performance indices are considered for optimization purposes.

Optimization techniques and heuristic search methods have been applied in tuning the gains of the PI controllers used for LFC problems. Most of these methods need several simulations of the system in order to optimize the gains and reach the best performance index defined by the control designer. The choice of this index is important on the optimization results and thus on the behavior of the controller.

Abdel majid et al.'s paper [8] deals with GA for optimizing the parameters of automatic generation control (AGC) systems. The controller considered in this study is of an integral type. Two performance indices have been widely used in the literature to find the optimum values of the classical AGC systems. Likewise, authors in this paper have also used these indices in association with genetic algorithm problems. The first performance index is the integral of the square error (ISE) and is defined in (2.2). This criterion penalizes the errors with respect to their weighting factors. The square of the error is derived in order to treat the positive and negative errors equally.

$$S_1 = \int_0^\infty e^2(t)dt \tag{2.2}$$

The second performance index is defined as the integral of time multiplied by the absolute value of the error (ITAE) and is formulated in (2.3). This standard includes time factor to penalize the settling time.

$$S_2 = \int_0^\infty t\left|e(t)\right|dt \tag{2.3}$$

Area control error (ACE) is one of the signals usually used for automatic generation control problems. This signal is a combination of area frequency and net tie-line power interchange. The ACE for each balancing authority or control area is defined as follows:

$$ACE_i = \Delta P_{tie\ i} + B_i \Delta f_i \tag{2.4}$$

where, B_i is the frequency bias factor of each area.

The integral controller will change the generation set point by affecting the ACE signal with an integral gain:

$$u_i = \Delta P_{ci} = -KI_i \int ACE_i dt \tag{2.5}$$

Genetic algorithm or other heuristic search methods can be applied to this problem to find each area's optimum values of the integral gains (KI_i) in order to minimize the defined performance indices. The similar approach is pursued in [9] and [25] in order to find the optimal gain settings of controllers for a two area hydro power system using GA.

2.3. Intelligent Controllers

Intelligent learning methods are applied to different control techniques in order to make the controllers more sophisticated with less need for human interaction. One of the major capabilities of the intelligent controllers is their ability to make decisions on taking proper actions, when there is a change in the system that requires an action from the controller. Agents are a group of these controllers that learn and take actions according to the operating conditions, taking advantage of different learning methods. Various applications of agents in power systems are reported in literature. Heo and Lee [27] have proposed a multi-agent based intelligent heuristic optimal control system for reference governor and optimal feedforward and feedback controls. Particle swarm optimization (PSO) is used as tool by the agents in order to generate optimal setpoints by realizing the reference generator. In the paper it is suggested that with the agent's intelligent and autonomous properties the complexity of large scale systems can be reduced due to a reduction in the coupling between subsystems.

2.3.1. Intelligent Load Frequency Controllers

Classical load frequency controllers are based on fixed-gain PI controllers. Like the methods discussed before, in many other studies the gains of the PI or PID controllers are fixed after they are optimized for a specific operating condition. These controllers may no longer perform satisfactory when the operating conditions of the system deviate from the nominal values. Also, the optimal controllers are functions of all the states of the system and in practice they may not be available. Additionally the control is dependent on the

load demand which requires accurate prediction of this variable [14]. Therefore, along with various areas in power systems, intelligent controllers have also been applied for load frequency control purposes in order to make the LFC scheme more applicable to real systems and compensate for the drawbacks of the conventional controllers.

Fuzzy logic is one of the methods that has been widely applied to this problem. Based on the type of the defined membership functions, the controller will adapt itself to the new operating conditions.

Fuzzy rule based load frequency control is addressed in Rerkpreedapong et al.'s paper [6]. Each area is controlled by an integral type controller. The control gain is adjusted in accordance to compliance with North American Electric Reliability Council (NERC) standards, CPS1 and CPS2. In fact the fuzzy gain will prevent wear and tear of generating units' equipment by preventing tight control. Input and output membership functions are defined so as to give the highest priority to the CPS1 compliance factor. In order to make the test system more realistic, regulation and load following services are considered in this paper.

In [7] the same authors have proposed two robust load frequency control designs. The first method is based on H_∞ design techniques using Linear Matrix Inequalities (LMI). The interface terms associated with the interconnections are treated as disturbances in this formulation, and thus the objective is to minimize the effect of this disturbance on the response of each area with a proper set of gains. Although the performance of the controller is very satisfactory, it has a complex structure and size of the controller is equal to the size of the system which makes its application in power systems unrealistic. The second approach, which is simpler in structure, is a PI controller formulated as an H_∞ problem tuned with genetic algorithm (GA). The proposed method called GALMI shows the same robust performance of the LMI based controller but with a simpler structure. One important drawback of both of the discussed methods is that they are based on the system model and in order to design the controllers the nonlinearities, such as generation rate constraints (GRC) are neglected.

Chang and Fu have also applied fuzzy logic to gain scheduling of area load frequency control [10]. In this paper a modified expression for area control error is used to

guarantee zero steady state time error and inadvertent interchange. This new area control error (ACEN) is the sum of conventional ACE and the integral of the conventional ACE:

$$ACEN_i = ACE_i + \alpha_i \int ACE_i dt \tag{2.6}$$

Generation rate constraint and governor dead-band are included in the system model in order to illustrate applicability of gain scheduling to nonlinear systems. The simulation results illustrate the acceptable performance of the controller when there is a small step change in each area. However, there is not much difference between a fixed PI controller and the proposed controller in order to justify the cost associated with applying this method to power systems.

One major drawback of the fuzzy gain scheduling approach is that the selection of fuzzy if-then rules requires a substantial amount of heuristic observations to achieve a proper strategy. To overcome this problem associated with fuzzy logic; in [11-12] authors have applied GA techniques in order to automatically design the membership functions of fuzzy controllers. Juang et al. [13] have proposed a new GA approach that reduces the fuzzy rule number and achieves a better performance. It should be noted that although the performance of the fuzzy system is improved, the complexity and other problems caused by GA is added to the design method.

Artificial neural networks (ANN) or simply neural networks (NN) have been identified as powerful tools for pattern recognition, functional mapping and generalization. Controllers based on neural networks have shown satisfactory performance in literature. The adaptive nature of ANN and their applicability to non-linear systems makes them more attractive for power system applications. Load frequency controllers are among the most widely used applications of NN in power systems.

Britch et al. [15] investigated the use of neural networks to identify the characteristics of the system and perform the control action that reduces ACE to zero. To train the network a supervised technique is employed that used different examples from the actual system to find the weights of the NN. The fact that a large number of inputs are fed into the network, makes the training process more complex and in some cases less accurate.

Also in the proposed method, the load for the present time step should be forecasted which itself requires a considerable amount of calculations.

Chatuverdi et al. [16] have proposed a new NN, named generalized neural network (GNN), which can compensate some of the drawbacks of the conventional networks. The NN controller regulates the output power and system frequency by controlling the speed of the generator with the help of water or steam flow control. The performance of the conventional neural network and the GNN are very close in response to a step load change. Also, the controller utilizes the rate of change of frequency in order to estimate the load perturbations, which again makes the controller more complex.

The major drawback of neural networks, which comes to mind once its operation is explained, is that it requires a considerable amount of training in order to expect a good performance from the network. The results are also very dependent on the selection of the training data. Therefore, in some cases if the system faces unpredictable conditions, which are not considered in the training phase, the NN might not be able to output a proper action.

In order to deal with the problem of offline training, Kuljaca et al. in [17] have designed a neural network control scheme that does not require training and is capable of online learning of the network parameters. The weight updating is based on lyapunov stability theorem. Therefore, the controller is designed based on the linear system model and there is no guarantee of stability if nonlinearities are included.

2.3.2. Reinforcement Learning Based Control

Most of the methods previously discussed are based on the system model and the controller needs some information from the system in order to decide on the control action. Therefore, designing a controller that can learn the appropriate control action without a need to acquire information from the system is an appealing approach in power systems, as in many cases it is not an easy task to perform measurement and gain an access to states of the system. Reinforcement learning (RL) has been utilized recently in different control applications, including power systems, in order to deal with this problem. Depending on the task performed different variations of RL methods are applied to the problems. Some of these methods are model based and some are non-

model based and can directly estimate the system parameters. As this topic is almost a new research in power systems, the number of publications in this area is limited. In this thesis RL techniques are applied to load frequency problems, but first the application of this method in different control tasks is investigated.

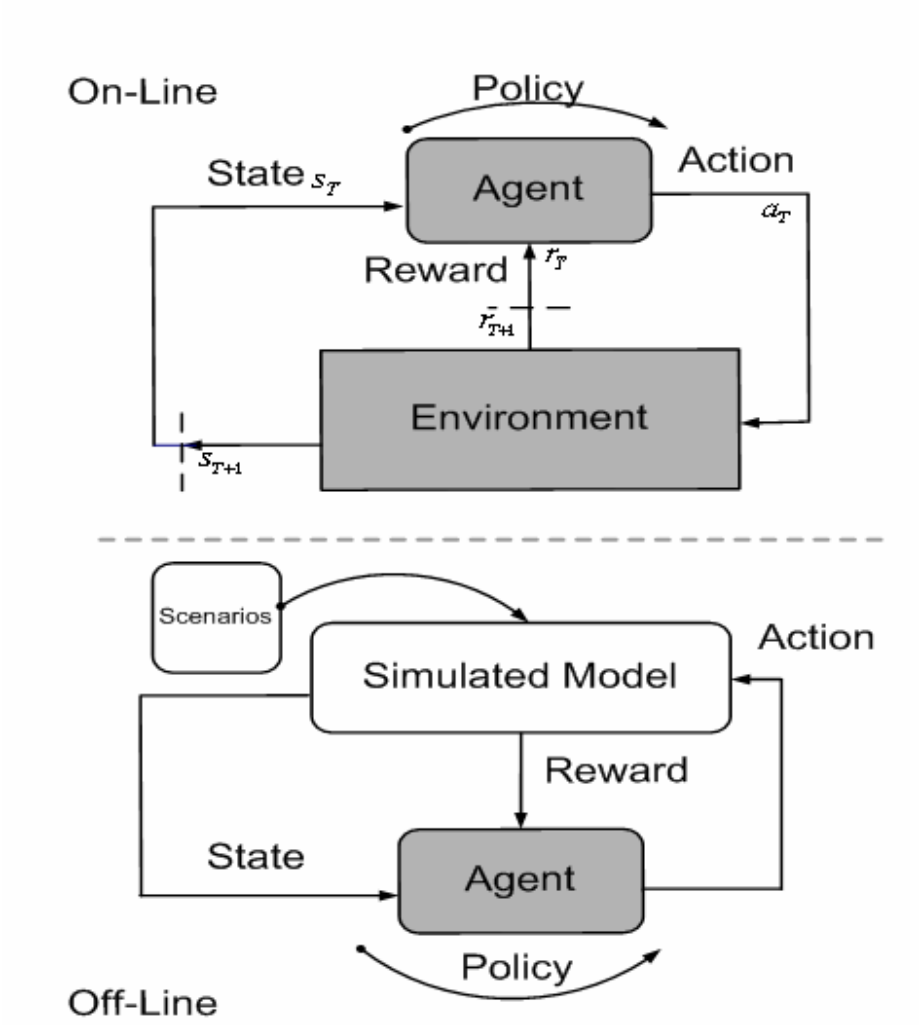


Figure 2.1: Online and Offline modes of control.

Former applications in power system control were applied off-line while the control interacts with the simulation model of the system before being applied to the real system. The learning capability of reinforcement learning methods makes them more applicable to online control applications, while the controller deals with the real system instead of a simulation model and therefore the decisions made by the agent will directly impact the

system. Figure 2.1 illustrates the main differences between the online and offline methods.

Q-learning is one of the RL methods finding applications in online control problems. This method is used in [18] for PID adaptive tuning while there is no prior information of the system available and also the system parameters are uncertain. Steady state error and overshoot are selected as the variables to define the states of the system. The actions are defined as a discrete increase or decrease in the PID gains. The proposed method can be applied for both offline and online applications. However, its online application makes it more attractive than the conventional, offline-tuned, controllers.

Off-line and on-line applications of RL are investigated by Ernst et al. in [19], [20], and [21]. The off-line mode concerns the design by means of RL algorithm for a dynamic brake controller. The objective of the dynamic brake controller is to damp large electromechanical oscillations to avoid loss of synchronism between generators. For the on-line mode, Flexible AC Transmission Systems (FACTS) devices with thyristor controlled series capacitor (TCSC) are considered to damp the power system oscillations. Reinforcement learning is used to determine the reactance reference of the TCSC. The reward function is defined based on the steady state error of the electrical power transmitted through the line. The model-based methods are used in order to design the controllers.

Imthias et al. [22] have applied RL methods to the automatic generation control (AGC) problem to adjust the generation set-points of each control area while they are subject to step load changes. The controller is designed offline, meaning that the agent learns through interaction with model of the system with different training samples. The actions taken by the agent is to increase or decrease the generation set-points. A two area system is simulated in this paper and an independent AGC controller controls each area in a decentralized manner. In this study the agents only decide on two actions and the method of setting the set-points is more appropriate for a linear model of the system. Therefore, when there are more limits on the system, this control method may not perform satisfactory. Although the authors in [23] have tested this method when generation rate constraint (GRC) and governor deadband are included in the system model, still it does not guarantee an acceptable performance when the disturbance on the

system is not a step change or is more than what simulated in the paper. Also, the offline training feature of this design is one of the drawbacks of the controller.

Chapter 3

Background Information

3.1. Reinforcement Learning

Reinforcement learning (RL) has attracted an increasing interest in the field of machine learning in the last decade. The ability of RL methods to provide systems with the intelligence of learning without a previous knowledge makes them even more attractive in current control applications.

In this thesis, the optimization problem is formulated as a Markov Decision problem (MDP). Different methods are studied to solve these optimization problems while RL techniques are one category of these methods. In the rest of this chapter, the basic features of a MDP problem are presented first. Then different methods that solve these problems are briefly introduced.

3.1.1 Markov Decision Problem (MDP)

An optimization task is said to be a markov decision problem if it consists of the following components [3]:

- A set of states $\mathcal{S}$,
- A set of actions $\mathcal{A}$,
- Transition probabilities $P^{a}_{s\to s'}$,
- Transition Rewards $R^{a}_{s\to s'}$.

The definitions of the states and actions will be discussed in the following sections. The state transition probabilities specify the probability of each possible next state s' as a function of state and agent's action:

$$P^{a}_{s\to s'} = P\left\{s_{t+1} = s' \middle| s_t = s, a_t = a\right\} \tag{3.1}$$

The transition reward determines the expected value of the next reward as a function of state and action:

$$R^{a}_{s\to s'} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \tag{3.2}$$

The model is said to be *Markov* if the state transition probabilities are independent of the previous states or actions. The MDP problems have some major components such as state and action value functions that will be discussed in definition of RL problems.

3.1.2 Reinforcement Learning Problem

Having the system parameters, dynamic programming (DP) methods can be used to find optimal solutions to MDP problems. However, obtaining the transition probabilities and transition rewards is often a difficult task and requires considerable amount of complex mathematics and it is sometimes impossible to find these parameters. Therefore, methods that can solve the problems without a need for the system model are required. Reinforcement Learning (RL) algorithms can satisfy this requirement and have shown satisfactory performance for optimization of unknown environments. It can be said that most of the RL algorithms are derivations of dynamic programming methods that do not require constructing the model of the system.

Reinforcement learning (RL) is learning to take actions by observing the current state of the system in order to maximize a long-term reward (Sutton 1998). This definition is a general expression for a series of methods trying to find the actions that result in the best reward. The agent will discover which action should be taken by interacting with its environment and trying different actions which may lead to the highest reward. In other words, the idea is to reward good actions and penalize bad actions and learn from trial and error. The term “reward”, which is perhaps the most important element in an RL problem, will be explained later in this chapter. Figure 3.1 is a block diagram representation of the reinforcement learning problem. The agent interacts with the environment and takes an action a_t from a set of actions $\mathcal{A}$, at time t. These actions will affect the system and will take it to a new state s_{t+1} from the set of states $\mathcal{S}$. The agent is then rewarded for this action, gaining the reward r_{t+1}. This agent-environment interaction is repeated until the desired goal is achieved.

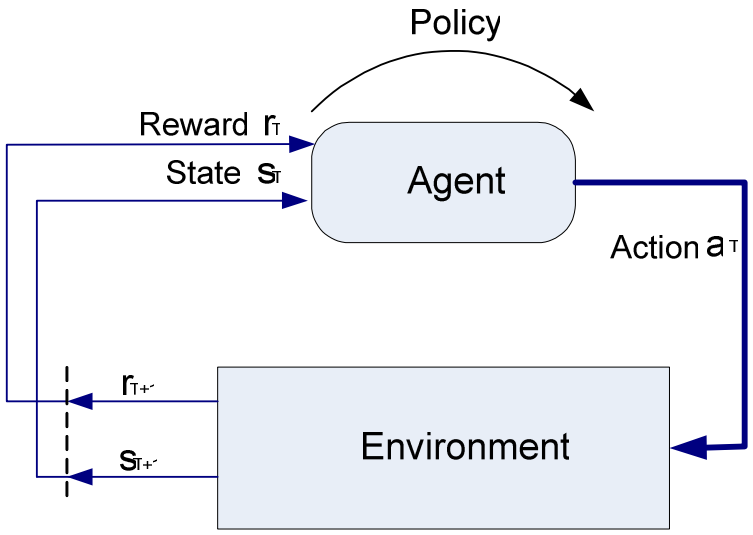


Figure 3.1: Block diagram representation of an agent-environment interaction.

In this text what is meant by the state is the system parameters that affect the reward function and are required for the agent to learn the value of taking a specific action in a specific situation. Conceptually each RL problem has the following important components:

• **State:** Series of information from the system that determines the degree of closeness to the objective. In other words, the state of a RL problem determines the current situation of the system based on the observations from the states of the system.

• **Action:** Decision made by controller that will affect the environment or system under control. This action varies depending on the application of the agent. In a control problem, for example it can be to set the gains of a controller or a change in setpoints.

• **Policy:** The set of actions an agent will take in specific states of the system are called the *Policy* of that agent. Policy is a mapping from the states to actions and is denoted by $\pi(s,a)$ .The role of the RL methods is to find the policy resulting in the maximum long term reward.

• **Reward:** The goal of an agent is to maximize its long term *reward*. Reward is in fact a scalar signal that determines how good (in getting closer to achieving its objective) is a taken immediate action. The reward function plays an important role in determining the performance of an agent because the agent decides on the action based on the received reward signal. The reward function is an external signal, assigned to the agent based on

its functionality. The better the definition of the reward function is, the better the performance of the agent would be. Also, the reward may be delayed as only several sequential actions may lead to the desired state. RL allows delayed rewards in its update process.

• **Return:** The sum of the expected rewards in the future is defined as *return* of the system. It is given by:

$$R(t) = \sum_{k=0}^{\infty} r_{t+k+1} \tag{3.3}$$

In general, the role of the agent is to maximize its return in the long run. From its definition it is understood that the future effect of an action is included in the definition of the return. However in many applications, a discount factor $0 \leq \gamma \leq 1$ is introduced and the return is modified so that the agent will maximize a discounted return defined by:

$$R_d(t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3.4}$$

The discount factor is included in the equation to determine the current value of future rewards [1]. Also, it can be thought of a way to bound the return in the long term. If the objective is to just maximize the immediate reward achieved by taking action a_t then $\gamma=0$. When $\gamma=1$ then the equation will be the classical definition of returns. In general, this definition means that a reward obtained k time steps in the future is discounted by a factor of γ^{k-1} of what it would be if it were received immediately.

• **State Value Function:** Different reinforcement learning algorithms are based on estimating the value functions. The value of each state *s* is called the state-value function and determines the value of being in a specific state in terms of the future expected rewards. This term is defined as the expected return when starting at state s_t using policy $\pi(s,a)$ and is given by [1]:

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1}\middle| s = s_t\right\}$$
$$= \sum_{a} prob(s,a)\sum_{s'} P^{a}_{s\to s'}\left(R^{a}_{s\to s'} + \gamma V^{\pi}(s')\right) \tag{3.5}$$

Where *prob(s,a)* is the probability of taking action a in state s under policy π and $P^{a}_{s\to s'}$ and $R^{a}_{s\to s'}$ are the probabilities of meeting next state s' and the expected value of next reward, respectively.

• **Action Value Function:** The action value function of each state s and action a, is defined as the expected return, or expected discounted reward, when starting at state s_t, taking action a_t, using policy $\pi(s,a)$. This term shows the value of a taken action in a specific state. It is known as a Q-function and it is given by:

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1}\middle| a = a_t,\ s = s_t\right\} \tag{3.6}$$

The reinforcement learning task is to find the optimal policy that maximizes the value function, V^*, for all states in the state space, i.e.

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \tag{3.7}$$

The optimal policy will also maximize the optimal action value function for all states and actions.

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{3.8}$$

One of the properties of the value functions is that they satisfy a number of recursive equations. With these equations the optimality conditions for these functions are found and represented by the *Bellman optimality equations* [1].

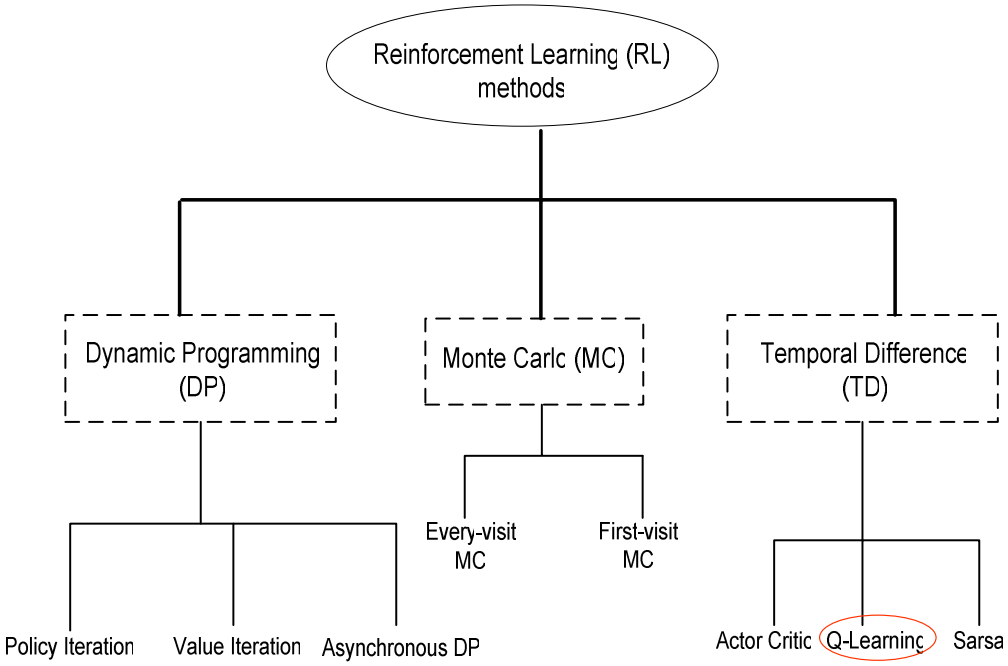


Figure 3.2: Classification of reinforcement learning methods.

$$\begin{aligned} V^*(s) &= \max_a E_{\pi^*}\left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| a = a_t, s = s_t \right\} \\ &= \max_a E\left\{ r_{t+1} + \gamma V^*(s_{t+1}) \middle| a = a_t, s = s_t \right\} \\ &= \max_a \sum_{x'} P^a_{s \to s'} \left(R^a_{s \to s'} + \gamma V^*(s') \right) \end{aligned} \tag{3.9}$$

Bellman's equation can also be written for the Q-function;

$$\begin{aligned} Q^*(s,a) &= E\left\{ r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \middle| a = a_t, s = s_t \right\} \\ &= \sum_{s'} P^a_{s \to s'} \left(R^a_{s \to s'} + \gamma \max_{a'} Q^*(s', a') \right) \end{aligned} \tag{3.10}$$

Different RL methods are suggested for solving the mentioned optimization problem. One can classify the methods that solve MDP problems in three major groups: Dynamic Programming (DP), Monte Carlo (MC) and Temporal Difference (TD) methods. As discussed before, DP methods require a complete model of the system and are

mathematically complex. Monte Carlo methods do not require a system model and are simple. However, these methods are not appropriate for an incremental computation. TD methods are model free and suitable for incremental computations. Due to the characteristics of this group of RL algorithms, there are found to be more applicable to power systems. Figure 3.2 illustrates this classification. Next, a brief overview of these methods is given along with a method that is used in this thesis. More comprehensive analysis of RL methods can be found in [1].

Dynamic Programming (DP) - DP methods are collections of algorithms that are guaranteed to find optimal policies for the MDP problems. Although, theoretically important, these methods require great computational expenses because they need a perfect model of the environment to be able to solve a problem. Therefore DP methods are not a good choice to be applied to complex systems. However, the rest of RL methods are in fact variations of dynamic programming with less computation and without assuming a perfect model of the system.

In all the RL algorithms it is tried to find the optimal policies by calculating or somehow estimating the value functions. As explained before, once the optimal value functions, V^* and Q^*, are found the optimal policies are derived. DP methods use Bellman's optimality equations to update the approximations of the value functions.

Before, describing the way policies are found, first the computation of state-value function under policy π, V^π, is considered. From (3.5), the state value functions for each state are defined as the functions of the transitions probabilities and immediate rewards. Therefore, if the dynamics of the environment are completely known, then (3.5) is a set of *n* linear equation with *n* unknowns, where *n* is the number of the available states. Iterative methods can be applied to this problem to find the solution to these equations. One variation of these methods is called *iterative policy evaluation*. It starts with arbitrary value assumptions for the values of each state and continues by updating these values, in each iteration, from equation (3.11).

$$V_{k+1}(s) = \sum_a prob(s,a) \sum_{s'} P^a_{s \to s'} \left(R^a_{s \to s'} + \gamma V_k(s') \right) \tag{3.11}$$

It is proven that the sequence $\{V_k\}$ converges to V^π when $k \to \infty$. Different variations of this method are proposed to increase the speed of convergence.

Once the value functions for each policy are calculated, better policies are searched by comparing the value of each action in each state and selecting the *greedy* action that results in maximum action value function, $Q^{\pi}(s,a)$. This will improve the current policy and create a new policy π'. It is proven that $V^{\pi'}(s) \geq V^{\pi}(s)$, therefore, the new policy will be closer to the optimal than the previous one. This process of improving the policy by creating new policies based on selection of greedy actions is called policy improvement. Once the policy is improved it can be improved even further until the optimal policy is achieved. This repeating process of evaluation and improvement is called *policy iteration*, which is one of the dynamic programming methods. Other DP methods such as *value iteration* and *asynchronous dynamic programming* try to decrease the amount of calculations and value evaluations in order to reduce the time of convergence. However, all these methods utilize two processes: value evaluation and policy iteration. As it will be discussed later, all other RL methods are based on these two principle theories.

Monte Carlo (MC) - Monte carlo methods are based on experience and they do not need a complete knowledge of the environment. These methods solve RL problems by averaging sample returns. MC methods are only defined for episodic tasks, meaning that experience is divided into episodes. It should be noted that value function estimates and policies are only changed upon completion of an episode.

Based on the averaging technique, different MC algorithms are developed. In *every-visit* MC method, $V^{\pi}(s)$ is estimated as the average of the returns following all the visits to state *s* in a set of episodes. The most widely studied MC method is the *first-visit* method in which just the returns following the first visit to *s* are averaged. For the policy evaluation purpose, the action value functions should also be estimated when there is no model of the system available. The same approach is used in order to average the returns followed by the visit to a state when the action was selected.

Policy improvement is done by selecting the greedy actions with respect to the current estimate of value function, i.e. selecting the actions that maximize the action value function in each state. Again, the value evaluation, policy improvement loop is repeated until the optimal policy is achieved. However, in order to guarantee the convergence of

this problem, all the state-action pairs should be visited so that an accurate approximation of the action-value function is achieved.

Temporal Difference (TD) Learning – These methods combine the two features of MC and DP and are one of the most applicable methods to control problems. They learn from experience in order to estimate the value functions and the update procedure depends on the previous values of the functions. Unlike MC methods that have to wait until the end of each episodic task, the TD methods can update the value functions after each time step. This is an advantage over the MC methods mainly because sometimes waiting until the end of an episode can be a long time which will considerably slow down the process of learning.

One of the simples TD methods known as *TD(0)* takes advantage of the following equation for the value function estimation:

$$V(s_t) \leftarrow V(s_t) + \alpha\left[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\right] \tag{3.12}$$

Where α is a constant step-size parameter and γ is the discount factor. From (3.12) it is observed that TD methods involve looking ahead a sample successor state to update the value of the original state. It is proven that for any policy π the TD algorithm described above will finally converge to V^{π} if the constant step size parameter (α) is sufficiently small [1].

Now that the method for estimating the value functions are described these estimate should be applied for control, i.e. to approximate the optimal policies. The same approach of policy improvement is followed, but we should make sure that all the state action pairs are visited during the experience. There are two approaches to meet this criterion: *on-policy* and *off-policy* methods. On-policy methods improve the policy used to make decisions. In fact, these methods estimate the value of each policy while using it for searching for the optimal policy. In off-policy methods however the policy that is used to generate the behavior is separate from the policy which is evaluated.

Sarsa is one of the on-policy TD methods for control purposes. In this method the current action value functions $Q^{\pi}(s,a)$ should be essentially estimated for the current behavior policy π. The same theories of *TD(0)* can be used in this case as well:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right] \tag{3.13}$$

Like all on-policy methods the Q^{π} is continuously estimated for the behavior policy π and simultaneously the policy is changed towards the greediness.

Q-learning is an off-policy TD control algorithm which in its simplest from it is defined by the following update equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right] \quad (3.14)$$

In this method the learned action value function directly approximates Q^* independent of the policy being followed. The convergence is guaranteed if all state action pairs are visited and their corresponding action value function Q(s,a) is updated. Figure 3.3 shows the procedural form of the Q-learning algorithm. In order to select an action the *ε-greedy* policy is used. This approach selects the action with the currently highest action-value (the greedy action) as experienced through interaction with the environment with the probability of (1-ε) and a random action with probability ε. With this policy the agent has the chance of trying non-greedy actions to explore the state-action space. The algorithm will repeat the procedure until the optimal policy is achieved or a certain number of states are visited.

Initialize Q(s,a) for all states and actions
Repeat for each run of the algorithm
    Initialize *s*
    Repeat for each step
    Take action *a* based on the policy determined by Q. (e.g. ε-greedy policy)
    Observe s_{t+1} and *r*

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$$

$$s \leftarrow s_{t+1}$$

until the desired goal is achieved or the terminal state is reached

Figure 3.3: Q-learning

3.2. Load Frequency Control (LFC)

The objective of load frequency control or LFC is to maintain the frequency in the scheduled value by balancing the generation and demand and to control the tie-line interchange schedules. Figure 3.4 represents the block diagram of the LFC loop and its basic operation [24]. A change in frequency and the real tie-line power are sensed through a change in the rotor angle, $\Delta\delta$. The frequency deviation Δf and tie-line power deviation ΔP_{tie} are amplified and transformed into a real power command signal ΔP_V which is sent to the prime mover which changes the torque by adjusting the amount of steam flowing through the valve. The prime mover then changes the generator output by an amount of ΔP_g changing the values of Δf and ΔP_{tie} accordingly.

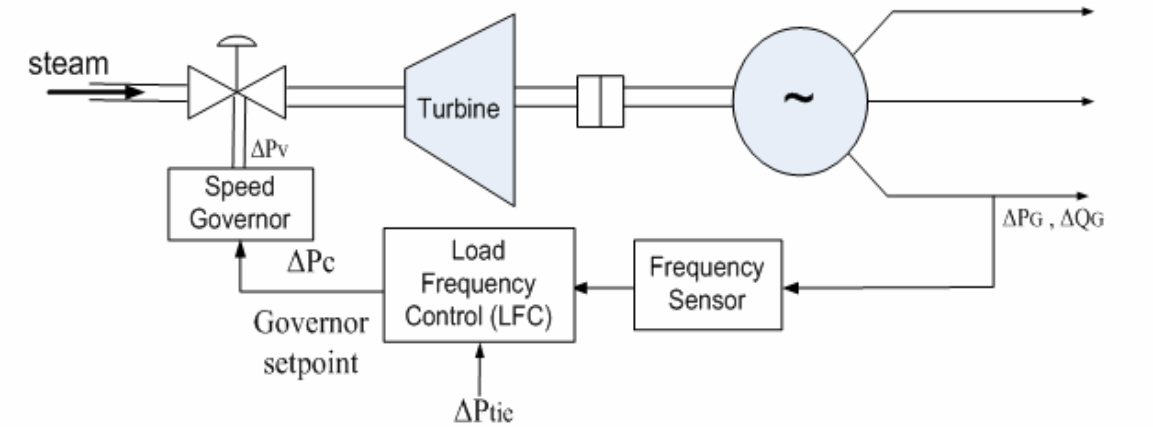


Figure 3.4: Block diagram representation of load frequency control loop [24].

From Figure 3.4 it is observed that the LFC will adjust the governor setpoint in order to compensate for the power imbalance. Figure 3.5 shows the schematic diagram of a conventional governor which consists of the following major parts. The *speed governor* which is essentially constructed of centrifugal flyballs driven by turbine shaft. Upward and downward movements are produced proportional to the speed change. The flyball movements are transformed to the turbine valve by *linkage mechanism* through *hydraulic amplifiers.* The hydraulic amplifier is needed to transform the movements of the governor into mechanical forces that control the steam valve. Finally, the *speed changer* schedules the load at nominal frequency with the aid of a servomotor which is operated manually or automatically.

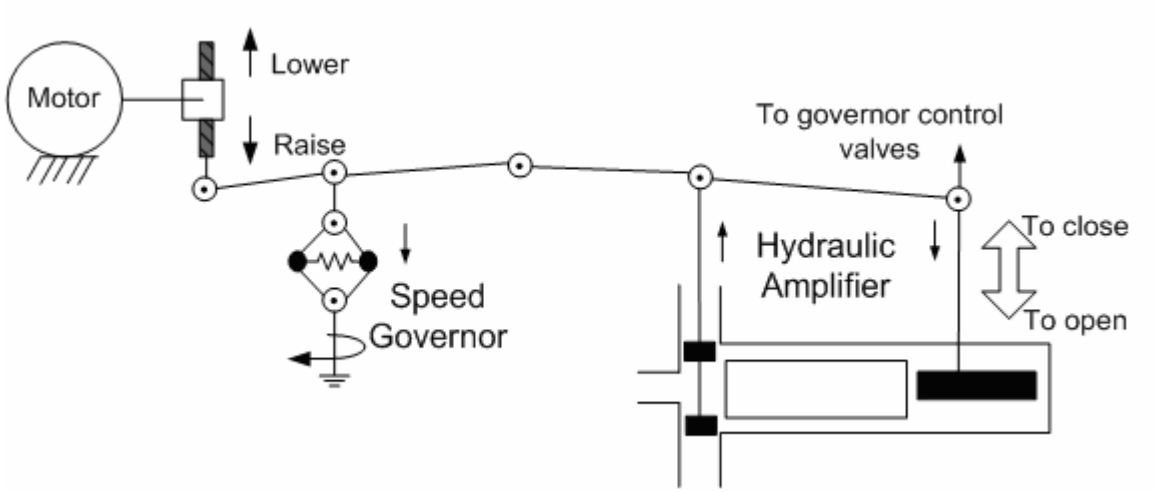


Figure 3.5: Speed governor system [25].

Various types of LFC yield different performances depending on the objective function chosen for the control design. In this thesis the objective is to regulate the area control error or (ACE) signal which is a combination of area frequency deviation and net power interchange error and is depicted in equation (2.3). The performance of the controller is assessed by the control performance standards. Two different approaches are used in order to reach this goal. In the first approach it is tried to regulate this signal and bring its variations as close as possible to zero when load changes are applied to each control area. In the second approach the controller is modified to reduce the unit maneuvering and wear and tear during operation.

In order to analysis the behavior of a system and design a control for that the mathematical model of the system is required. Consequently, the first step is to derive a model of the system. Proper approximations are made and the components of the system are represented in the form of transfer functions.

Figure 3.6 illustrates the equivalent model of the control area *i* of the power system studied in this thesis. This model is inspired from [4]. The model is a general representation of a control area with more than one speed governor and generating unit. In order to find the equivalent transfer function of the ith area's generator, all the generators in that area are lumped and they are represented by a single transfer function whose output is the area frequency deviation. Each control area is connected to the other

areas though tie lines. As it will be discussed later in this text, a conventional PI controller is used in each area to regulate the ACE signal.

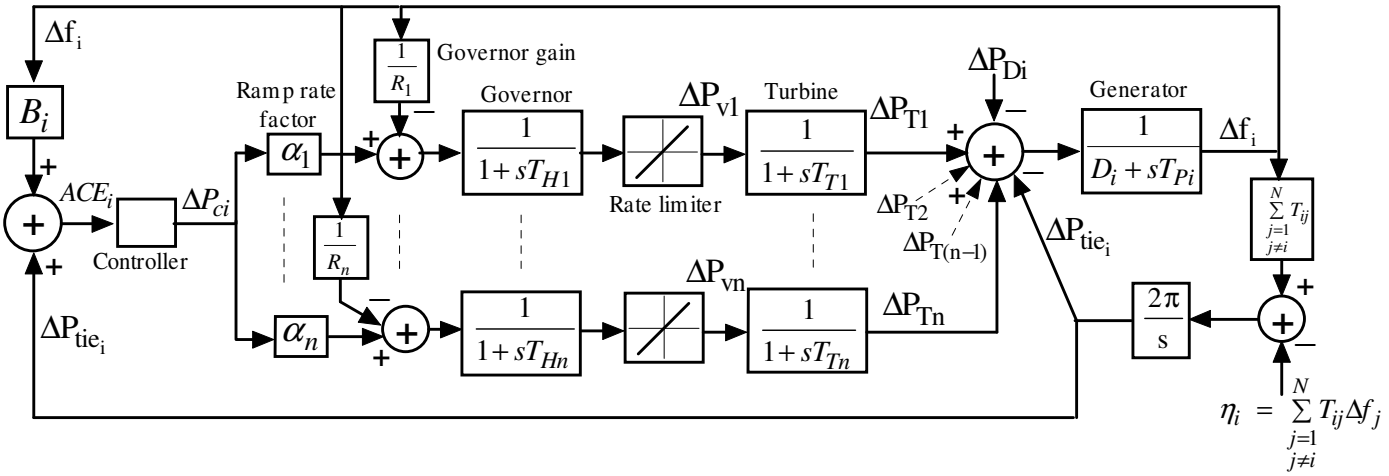


Figure 3.6: Dynamic model of control area i for the LFC problem [4].

P_T: turbine power — P_C: governor load setpoint

f: area frequency — Δ: deviation from nominal values

T_{ij}: tie-line synchronizing coefficient between area i and j

T_T: turbine time constant — α: ramp rate factor

P_V: governor valve — T_P: area aggregate inertia

T_H: governor time constant — B: frequency bias

P_{tie}: net tie line power — P_D: power demand

N: number of control areas — η: interface

D: damping coefficient — R: Droop characteristics

Chapter 4

Decentralized Reinforcement Learning-Based Load Frequency Control

4.1. Introduction

Most of the methods used previously for load frequency controls are model based and are designed for a specific operating condition. Although these controllers may demonstrate a satisfactory performance in normal situations, they might not be able to control the system while there is a sudden change in system parameters and operating conditions which is not considered in the controller design. Adaptive controllers surveyed in chapter 2 can serve as good alternatives in this case in order to adapt their parameters depending on the type of disturbance imposed to the system. However, these controllers are complex in their design and are still designed for specific system parameters. Including nonlinearities and limits in the model is also a hard task that should be accounted for in a new type of design.

This chapter will start by describing the power system model used for the simulation purposes during the entire thesis. Thereafter, the issues of fixed load frequency controllers are discussed and compared with the adaptive controllers. The two area power system is simulated for the two types of controllers when both areas are subject to load changes. Then a new adaptive controller is proposed that will learn the proper gains of the controllers without any knowledge of the system. Reinforcement learning is the main tool used in the design of this controller. Simulation results compare the performance of the proposed controller with the conventional adaptive controllers. In the end, the advantages and disadvantages of using these types of controllers in power systems are discussed.

4.2. Adaptive Versus Fixed Controllers

Before discussing the advantages of the adaptive controllers designed for LFC problems, over the controllers with fixed parameters the power system model described in the previous part is simulated with both types of the controllers. For simplicity the two-area power system model is selected for simulation. The type of disturbance applied to each area is a constant and random load change in addition to sudden step changes in the loads of each area. The parameters of the system are presented in Table 4.1. These parameters are inspired from [4].

TABLE 4.1
TWO AREA SYSTEM PARAMETERS

Parameters	Genco				
MVA base(1000MW)	1	2	3	4	5
Rate (MW)	1000	800	1000	1000	800
D(pu/Hz)	0.015	0.014	0.015	0.015	0.014
T_p (pu.sec)	0.1667	0.12	0.2	0.1667	0.12
T_T (sec)	0.4	0.36	0.42	0.4	0.36
T_H (sec)	0.08	0.06	0.07	0.08	0.06
R (Hz/pu)	3	3	3.3	3	3
B (Hz/pu)	0.3483	0.3473	0.318	0.3483	0.3473
α	0.4	0.4	0.2	0.4	0.4

The fixed controller in this case is a conventional proportional integral (PI) controller which is widely used in industry. The gains of this controller are tuned by optimization techniques introduced in Chapter 2 [7] and are presented in Table 4.2. Each area is equipped with a PI controller and therefore different areas are controlled in a decentralized manner.

TABLE 4.2
PI CONTROLLER PARAMETERS

	Area 1	Area 2
Proportional Gain	-3.27×10^{-4}	-7×10^{-4}
Integral Gain	-0.333	-0.343

Different kinds of adaptive controllers are surveyed in Chapter 2. H_∞ controllers are one of these various controllers that have demonstrated a good performance when applied to different control tasks. The parameters of the designed H_∞ controller are presented in [7]. The simulation results of area control error (ACE) and governor mechanical power deviation are presented in Figure 4.1 and 4.2., respectively.

By comparing the results it is clearly seen that the H_∞ controller outperforms the fixed PI controller when there is a sudden change in the operating conditions. The PI controller only performs satisfactory when the load changes are close to the scenarios that their design was based upon.

Next it is assumed that system parameters are changed by 20% and the same scenario is simulated to observe the behavior of these model-based controllers when model deviates from the original one. The simulation results are shown in Figure 4.3 and 4.4. The results for the adaptive controller illustrate that the controller is highly dependent on the system model. Although the H_∞ controller was acting properly in the previous scenario, after a change in system parameters it couldn't control the system.

Therefore a need for a more sophisticated adaptive controller is justified. This controller should be able to learn the necessary changes in the control settings according to the changes in the system parameters. With these characteristics, the above mentioned controller can be applied to any system without a need for pre-adjustments. In the next section the basic features of this controller are described and the load frequency problem is solved with the new proposed controller and compared to the previous adaptive controller.

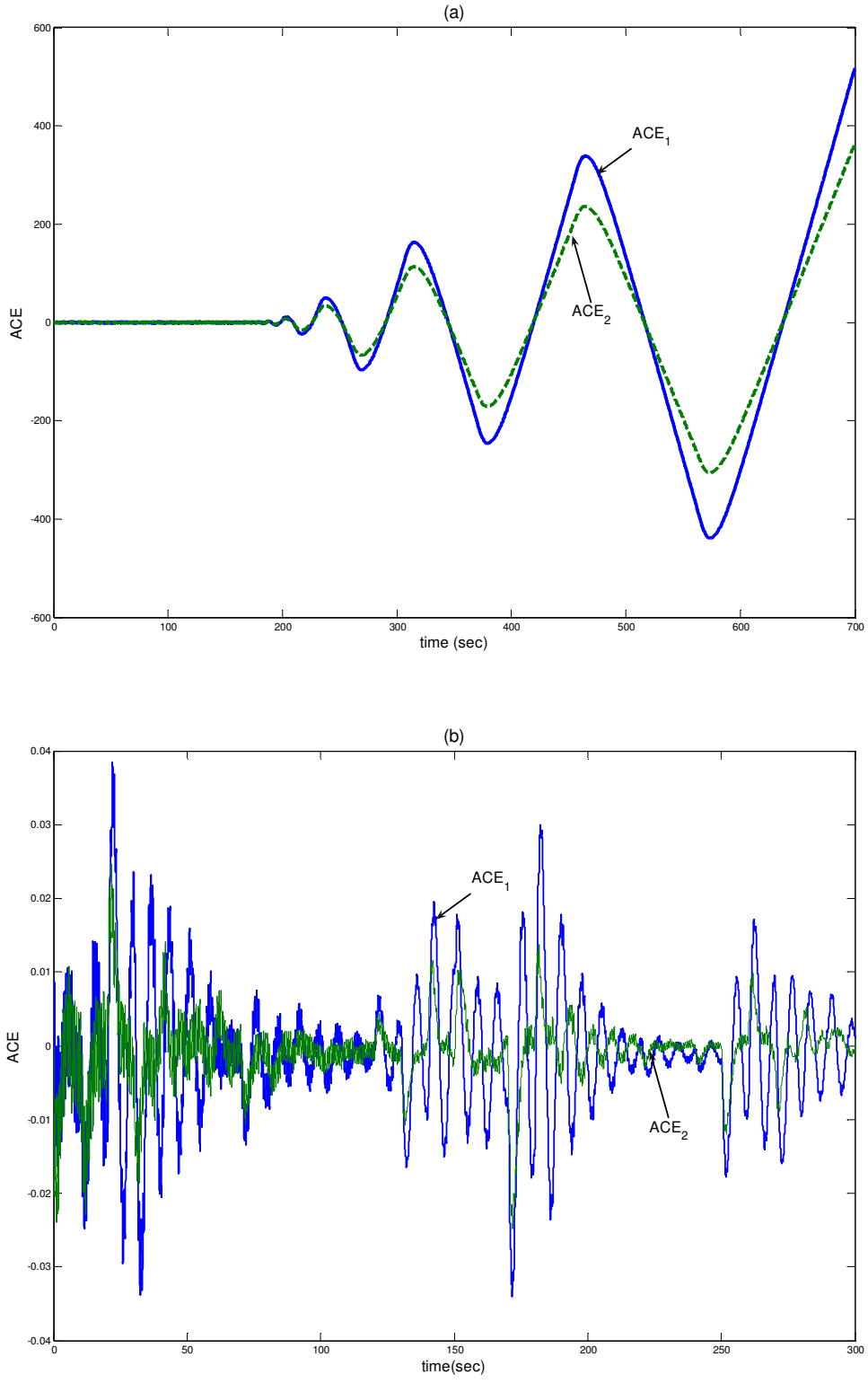


Figure 4.1: Area control error (ACE) for a two-area system: (a) fixed PI controller, (b) H_∞ controller.

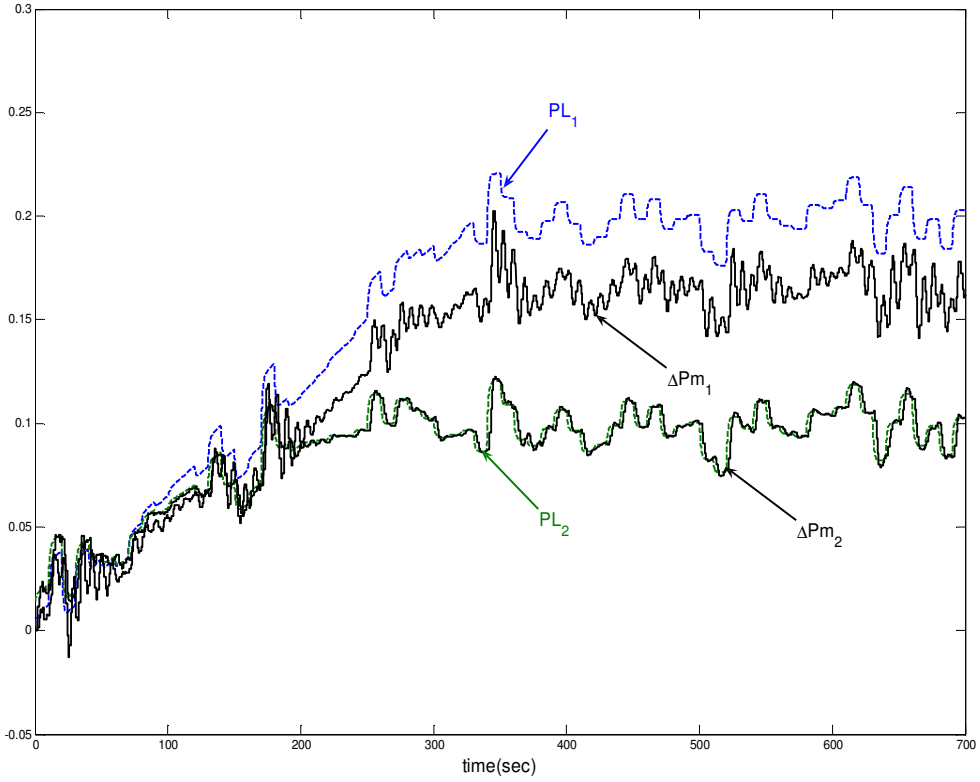


Figure 4.2: Generated governor mechanical power for H_∞ controller.

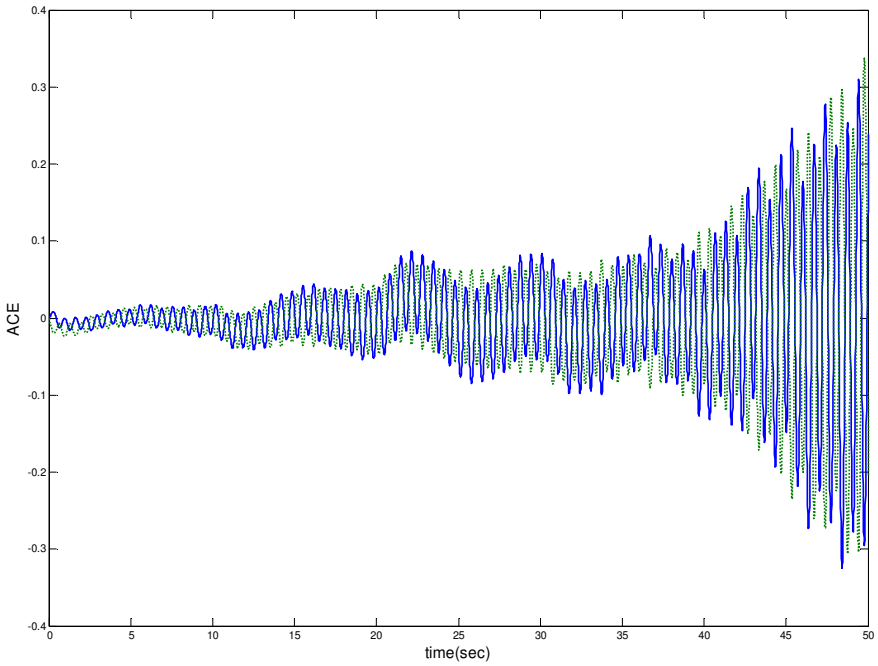


Figure 4.3: Area control error (ACE) signal for a two area system with H_∞ controller when system parameters are changed by 20%.

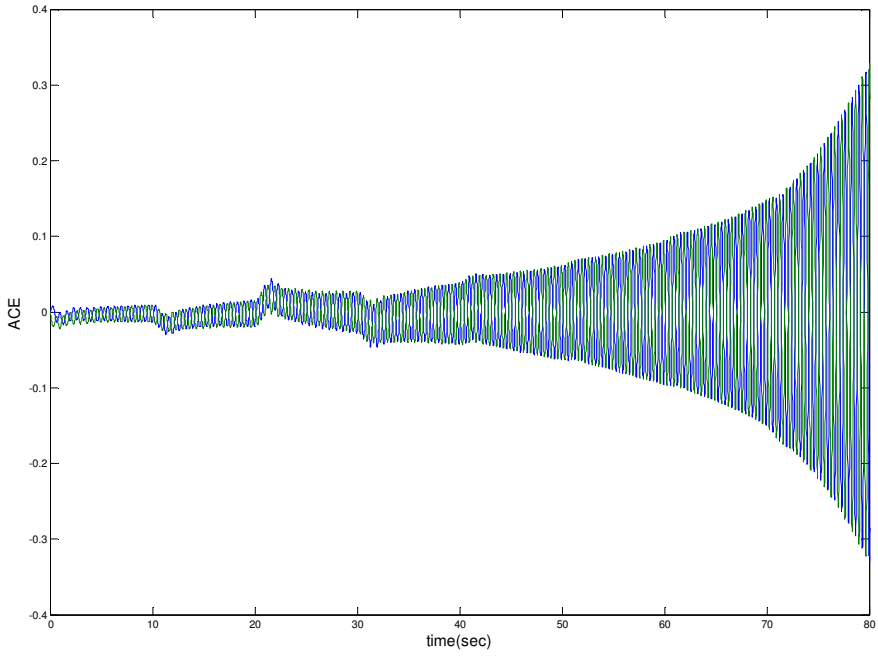


Figure 4.4: Area control error (ACE) signal for a two area system with fixed PI controller when system parameters are changed by 20%.

4.3. Reinforcement Learning-Based PI Controller

Ability to learn from experience can compensate for many problems associated with the model based controllers. Conventional PI controllers have shown good performance in many normal conditions and their design is considerably simpler than most of the adaptive controllers. Therefore if the learning capability is combined with the simple design of PID controllers the performance of these controllers could be enhanced in many cases. Also, once designed, the controllers can be applied to various systems with different system parameters.

Reinforcement learning methods are therefore used in this thesis in order to design the controller with the mentioned characteristics. From the desired features of these controllers, non-model based methods become more attractive in solving such problems than the methods based on the system model. Q-learning is one of these methods that have widely been used for power system applications. As explained in Chapter 3 in this method the agent does not require any prior knowledge of the system in order to make a decision on the action that should be taken. However, the experience gained by interacting with the environment will gradually improve the performance of the controller. Next, the LFC problem is formulated as an RL problem.

The controller proposed in this thesis is the conventional PI/PID controller and its gains are tuned by means of reinforcement learning algorithms. The proportional, integral and derivative gains are changed each time a disturbance is applied to the system. Consequently, these controllers will adjust themselves to the new operating conditions. The main advantage of the new PID controllers is their simple design and ability to learn the proper gains without any prior knowledge of the system and its parameters. Also in contrast to many adaptive controllers applied to LFC problem there is no need to estimate the load changes on the system. However, in order for the agent to make decisions some of the system variables such as frequency should be measured and fed back into the agent as the inputs. Figure 4.5 presents the basic structure of the RL based controller.

Before applying reinforcement learning for the control problem, the elements of the RL problem should be defined. Among these elements, states, actions and reward are of more importance and in fact define the task and objective of the learning process. These elements are defined next.

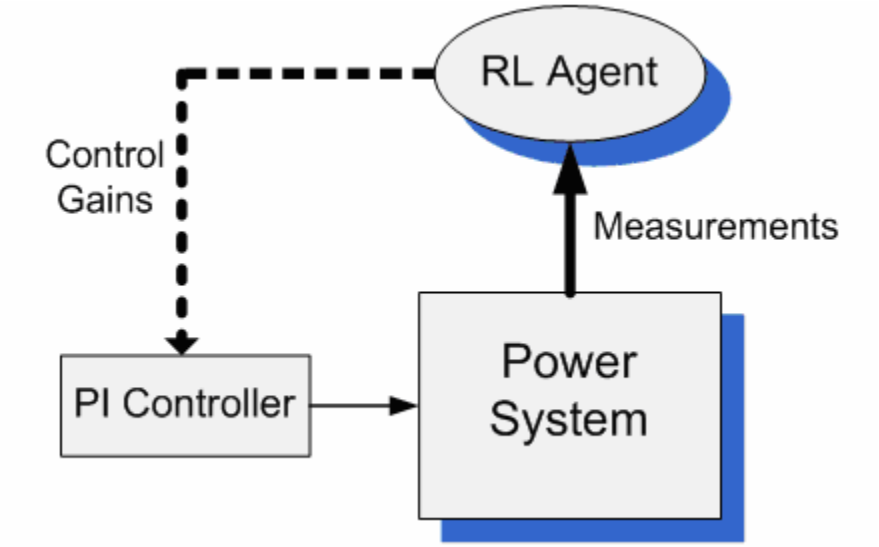


Figure 4.5: Block diagram of proposed reinforcement learning based PI controller.

Sate: State in this case should be a signal that determines the performance of the controller. In load frequency problem the ultimate goal of the controller is to regulate the ACE signal and maintain its variations within a limit acceptable by the standards. Therefore, the ACE signals can be a good representer of the controller's behavior. Based on what was explained, the state is defined as the discrete levels of the ACE signal within an interval [ACE_{min}, ACE_{max}] considered for AGC. The |ACE| in this interval is quantized into finite levels and each level is considered as a state of the system. A controller that reaches ACE_{max} , where |ACE| > ACE_{max}, assumes to not act properly and starts learning better control settings. Also, if 0<=|ACE|< ACE_{min} then there is no need for control action by the agent. The reason the signals are discretized is that the RL problem considered in this thesis is assumed to be a Markov Decision Process (MDP) and as explained in Chapter 3 they require a discrete and finite state space. The average value of the ACE signal can also be considered as the state signal. However, the simulation results show that the instantaneous value of the ACE signal could be a better choice rather than its average value. Also, it should be noted that based on the system considered for control and its parameters one can change the state levels in order to find the optimal performance. However, with an accurate enough definition of states and with an

acceptable number of state levels a satisfactory performance can be achieved from the RL agent.

Action: When RL techniques are applied to a control problem the action of the RL agent should directly affect the controller. In the case of this problem the action would be to increase or decrease the proportional or integral gains of the controller, if the controller used is of a PI type. It should be noted that in some cases the agent might choose not to change any of the gains which is also considered as an action. Therefore the agent will have at least three actions to choose for each gain which leads to a total of 6 actions for a PI controller.

Similar to states, the agent should be able to choose between a finite set of actions. As the changes in the gains of the controllers could be continuous, these changes should somehow transform to discrete variations. In order to achieve this, the increment between the changes of the gains is defined so that the agent should exactly know how much increase or decrease in the gains is applied. Depending on the system this increment can be changed and adjusted and a good selection of this parameter can effectively improve speed of the learning process.

In order to further improve the performance of the controller, one may define actions in a way that two sets of increase or decrease of the gains are defined. One is to change the gains a relatively large amount and the other would be to change it less. Although this would make the decision process more complex for the agent, mostly because the number of actions will increase, this will make the controller more applicable to various system parameters.

Reward: As explained in Chapter 3, reward function plays an important role in the learning process. Thus this function should be carefully defined. Area control error can be used as a variable to define this function because its variations determine if the controller is learning in a correct direction or another action should be taken to get closer to the objective. The perfect ACE signal is the one that has been driven to zero, therefore if a taken action drives this signal closer to zero, it should expect more reward than an action who has increased the fluctuations of this signal. From this definition one can understand that a linear function of the ACE signal can be a good candidate for the

reward function. Later in this chapter the effect of different reward functions on the controller performance will be investigated.

Power system is divided to several control areas and each area should have its own control structure in order to deal with the load changes of the local areas without affecting other areas. Therefore the proposed controller is applied to each control area, in a decentralized manner. Each area is equipped with an RL agent which decides on the proper controller gains for that area. The measurements are done locally and the only information available to each area from the rest of the system is the tie-line power. These measurements include the frequency and tie-line power which are needed to calculate the ACE signals in each sample time. Figure 4.6 presents the basic structure of multi agent control for multi area load frequency control.

It should be noted that each control area can consist of several generating units and in order to model the system all these units are lumped together and an equivalent model is derived. Also, what is meant by frequency of each area is actually the equivalent frequency of that control area.

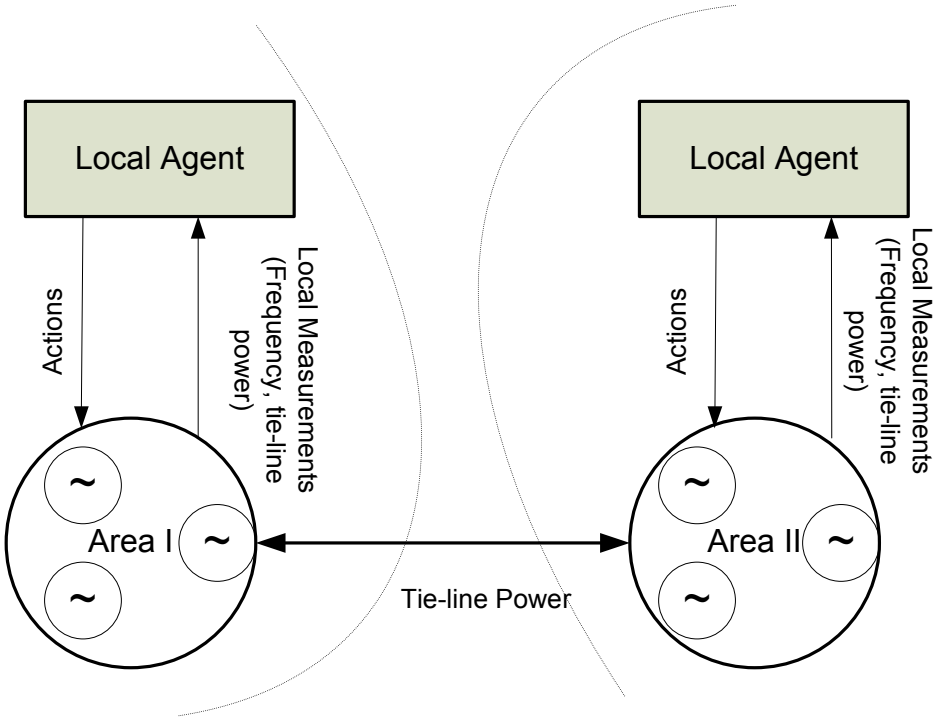


Figure 4.6: Multi-agent LFC controller structure.

4.4. Case Studies

4.4.1. Effect of the Reward Function

In order to show the importance of the reward function, two different functions are defined in this section and their performance is compared through the simulation results. The first reward function is defined as a linear function of the ACE signal as follows:

$$r_1(t) = -100 \times |ACE(t)| \tag{4.1}$$

In this function the instantaneous value of ACE is multiplied by 100 in order to increase the effect of small differences, especially if the simulation is in the per-unit scale, while a small difference can in fact be a huge number in a real scale. The second reward function is defined as bellow which is close to the reward function defined in [23]:

$$r_2(t) = \begin{cases} -1 & if\ ACE_{\min} \le ACE(t) \\ 0 & if\ ACE_{\min} > ACE(t) \end{cases} \tag{4.2}$$

In this definition, when the ACE variation in more than the lower bound the action will receive a penalty (or a negative reward) of -1, independent of the value of the ACE. Next, the two area system shown in Figure 4.7 is simulated while nonlinearities such as generation rate constraint (GRC) and governor dead-band are also considered in the system model.

The simulation results compare the variations of the ACE signals when a series of disturbances are applied to the loads in both areas. These results are presented in Figure 4.8. It is observed that the first reward function serves better in determining the behavior of the RL agent. The reason for this superior performance is that in this definition there is a difference between actions that result in different ACE variations. For example, the action that leads to some ACE oscillations receives less reward than an action that further increases the ACE deviation.

By observing these results, it is confirmed that the definition of the reward function can affect the performance of the RL agent. An improper reward function not only will not improve the performance of the controller, but also its performance can be even worse than the conventional controllers. Figures 4.9 and 4.10 summarize the variation of the proportional and integral gains for both control areas, in both cases.

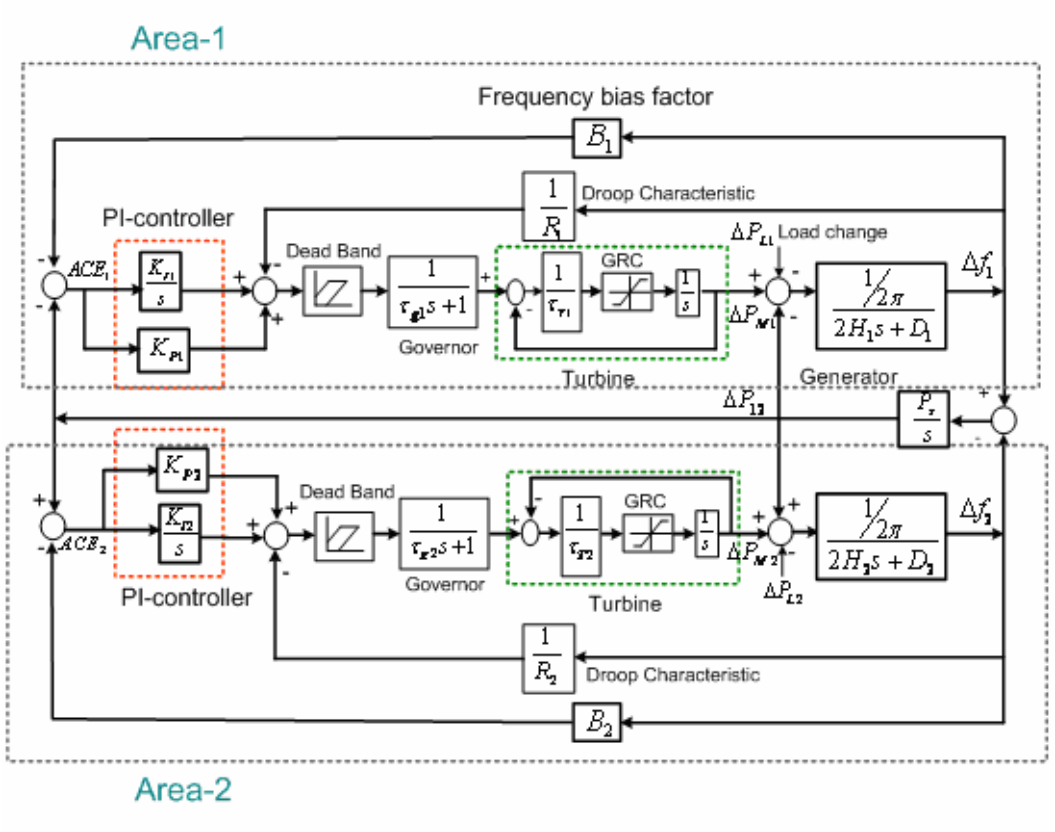


Figure 4.7: Block diagram of two-area power system model with PI controllers for each area.

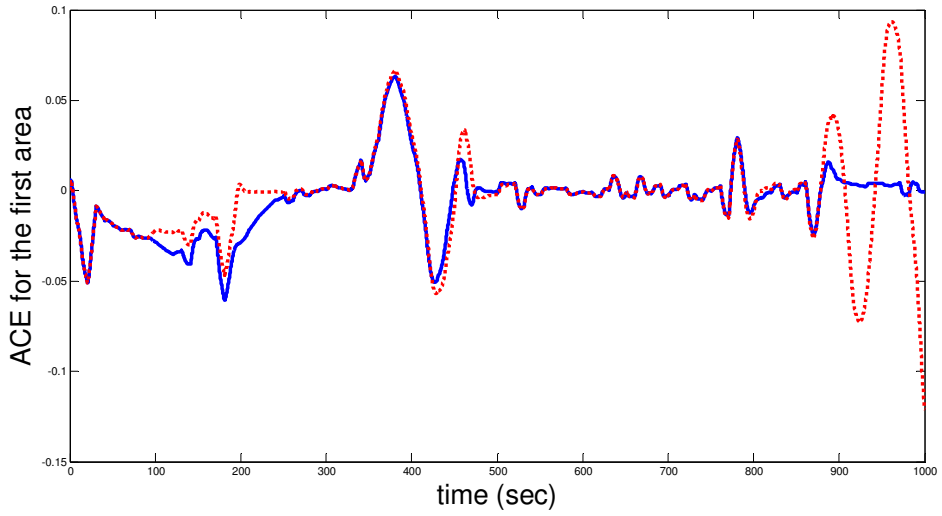


Figure 4.8: ACE signal variations using the first reward function (solid line) and the second reward function (dashed line).

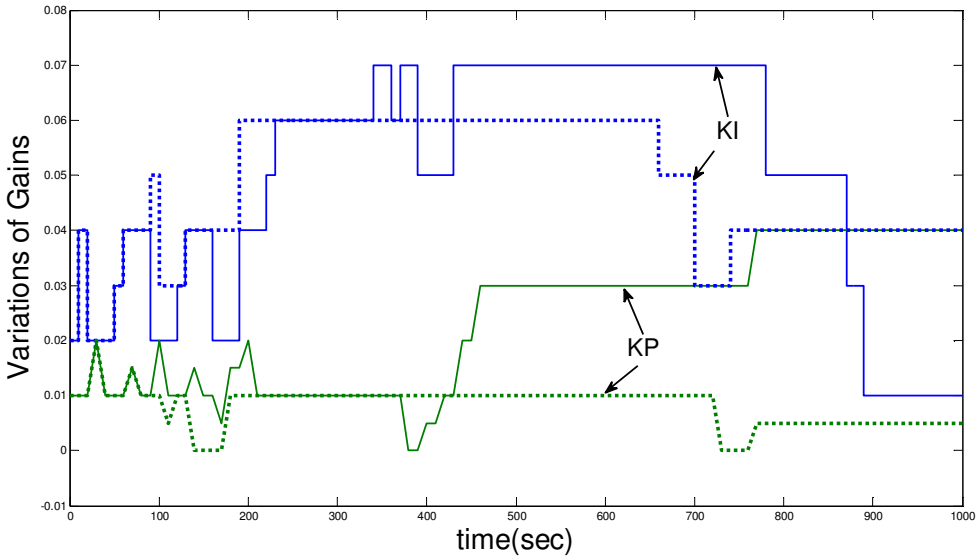


Figure 4.9: Proportional and integral gain variations using the first reward function: first area (solid line) and second area (dotted line)

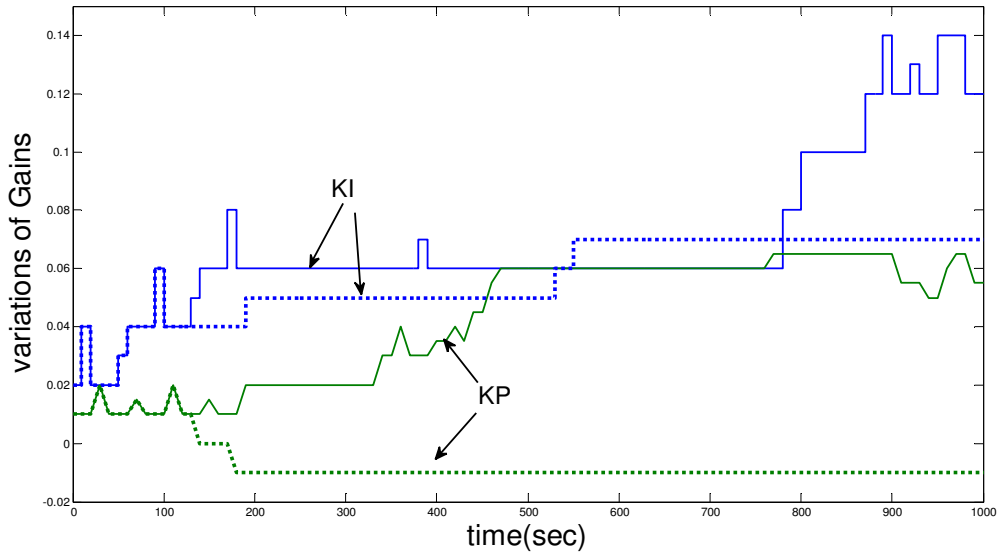


Figure 4.10: Proportional and integral gain variations while using the second reward function: first area (solid line) and second area (dotted line)

4.4.2. Three Area Power System

Intelligent controllers were originally designed to be applied to large systems were the dimensions and complexities make the application of many classic controllers unrealistic and costly. The case studies studied before in this text and many other research papers where implemented on a two area system. However, in order to show the applicability of the designed controller to large systems a three area power system is simulated. Each area has three generation companies (Genco) providing the generated power and one distribution company (Disco). All parameters of the Gencos are presented in Table 4.3 [4]. The tie-line synchronizing coefficients between areas are $T_{12} = 200\ MW/rad$, $T_{23} = 120\ MW/rad$, $T_{23} = 250\ MW/rad$. Each area is equipped with a decentralized RL based PI controller and the areas are connected to each other through the tie-lines. Figure 4.11 shows the structure of the mentioned three area system.

TABLE 4.3
THREE AREA SYSTEM PARAMETERS

Parameters	**Genco**								
MVA base(1000MW)	1	2	3	4	5	6	7	8	9
Rate (MW)	1000	800	1000	110	900	1200	850	1000	1020
D(pu/Hz)	0.015	0.014	0.015	0.016	0.0140	0.0140	0.0150	0.0160	0.0150
T_p (pu.sec)	0.1667	0.1200	0.200	0.2017	0.1500	0.1960	0.1247	0.1667	0.1870
T_T (sec)	0.4	0.36	0.42	0.44	0.32	0.40	0.30	0.40	0.41
T_H (sec)	0.08	0.06	0.07	0.06	0.06	0.08	0.07	0.07	0.08
R (Hz/pu)	3	3	3.3	2.7273	2.6667	2.50	2.8235	3.00	2.9412
B (Hz/pu)	0.3483	0.3473	0.318	0.3827	0.3890	0.4140	0.3692	0.3493	0.3550
α	0.4	0.4	0.2	0.6	0	0.4	0	0.5	0.5
Ramp rate (MW/min)	8	8	4	12	0	8	0	10	10

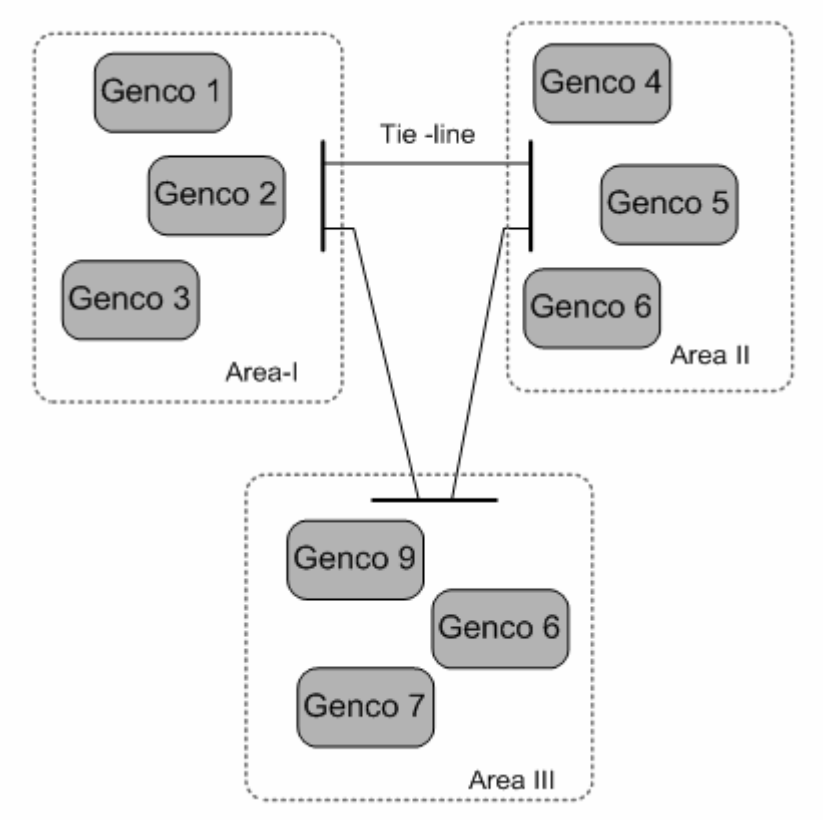


Figure 4.11: A three-area power system.

Two different scenarios are simulated for this system to illustrate the effectiveness of the proposed control technique. The first scenario is when random load changes are applied to all the areas as shown in Figure 4.12(a). The area control errors (ACE) and governor load setpoint (ΔP_C) are presented in Figure 4.12 (b) and (c) respectively. Figure 4.13 illustrates variations of the control gains for three control areas. As it is seen, the controller will keep the variations of the ACE in an acceptable range and these constant, random variations do not lead to system instability.

For the second scenario a large disturbance which is a step increase in demand is applied to each area: ΔP_{D1}= 120 MW, ΔP_{D2}= 100 MW, ΔP_{D3}= 80 MW. Figure 4.14 (a) and 4.14(b) show the performance of the controllers when they are subjected to large disturbances. From the results it is observed that the controllers are able to smoothly increase the governor setpoints to the new value in order to match the generation of each area with their demands. These types of large disturbances rarely occur, mostly because a party that causes a large imbalance between the actual and forecast load is penalized in power market [4].

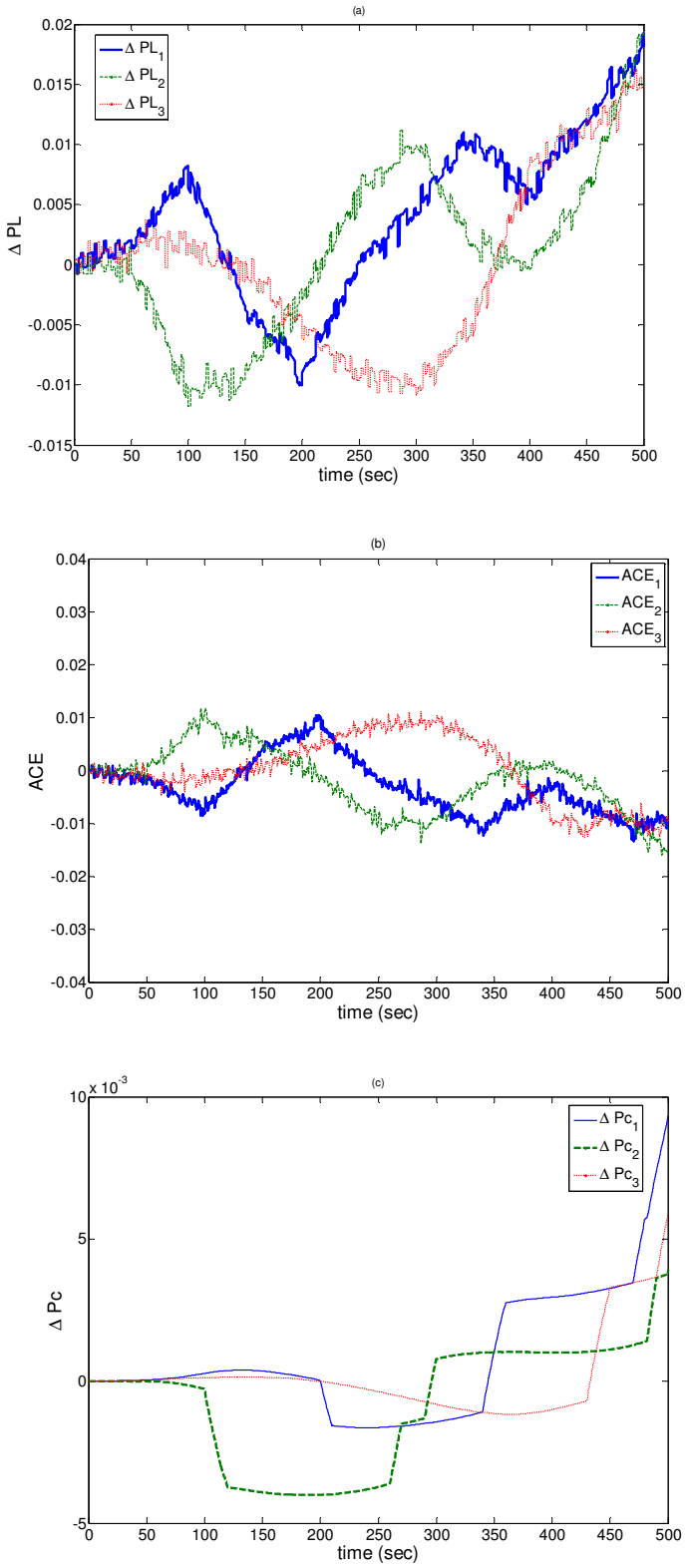
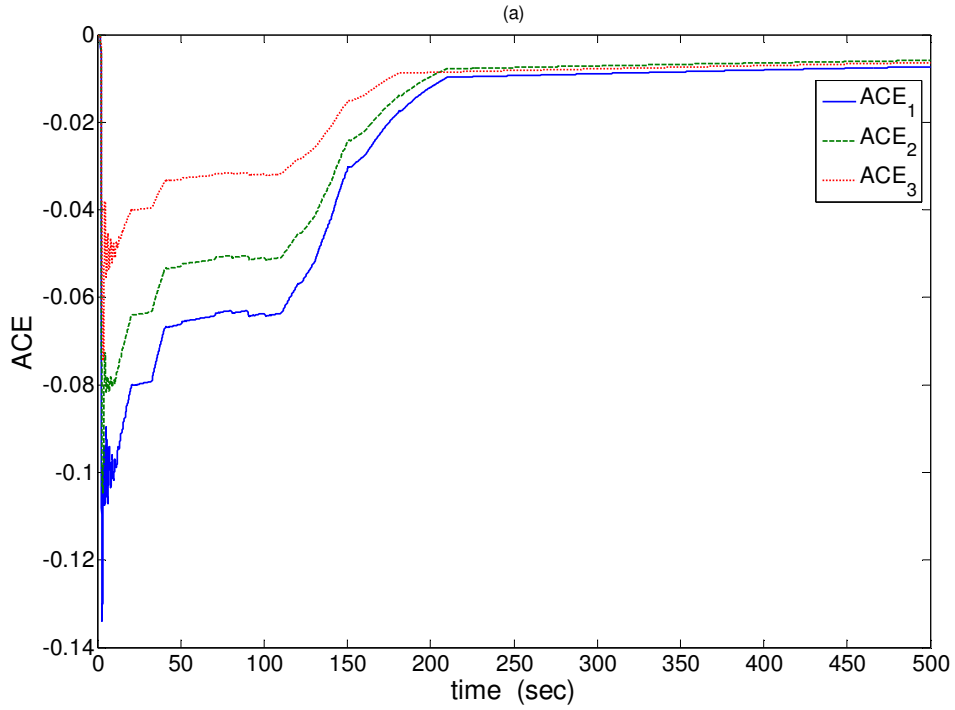
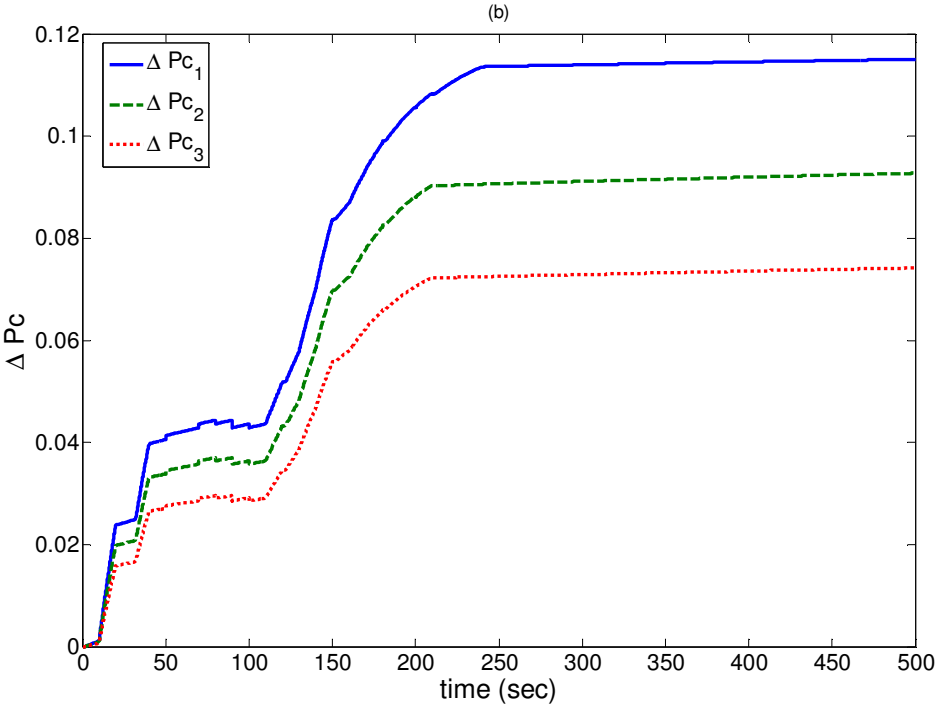


Figure 4.12: (a) Loads, (b) ACE and (c) governor setpoint variations of the three areas for scenario 1.


(a)
ACE
time (sec)
ACE1
ACE2
ACE3
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
0
50
100
150
200
250
300
350
400
450
500


(b)
Δ Pc
time (sec)
Δ Pc1
Δ Pc2
Δ Pc3
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
250
300
350
400
450
500

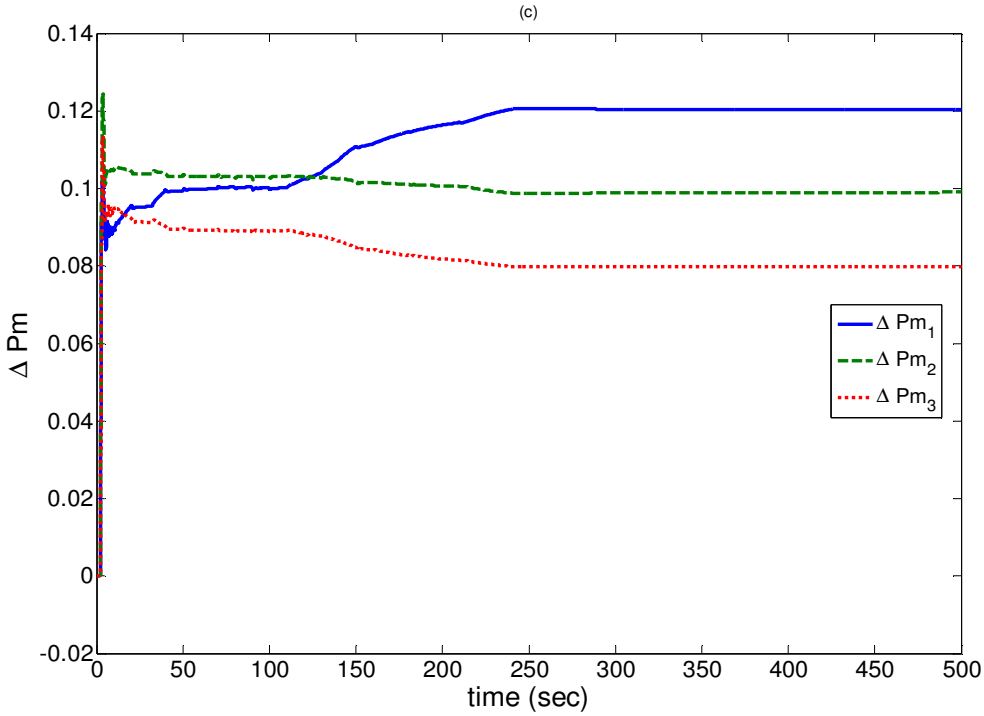


Figure 4.13: (a) ACE ,(b) governor setpoint variations and (c)generated mechanical power of the three areas for scenario 2.

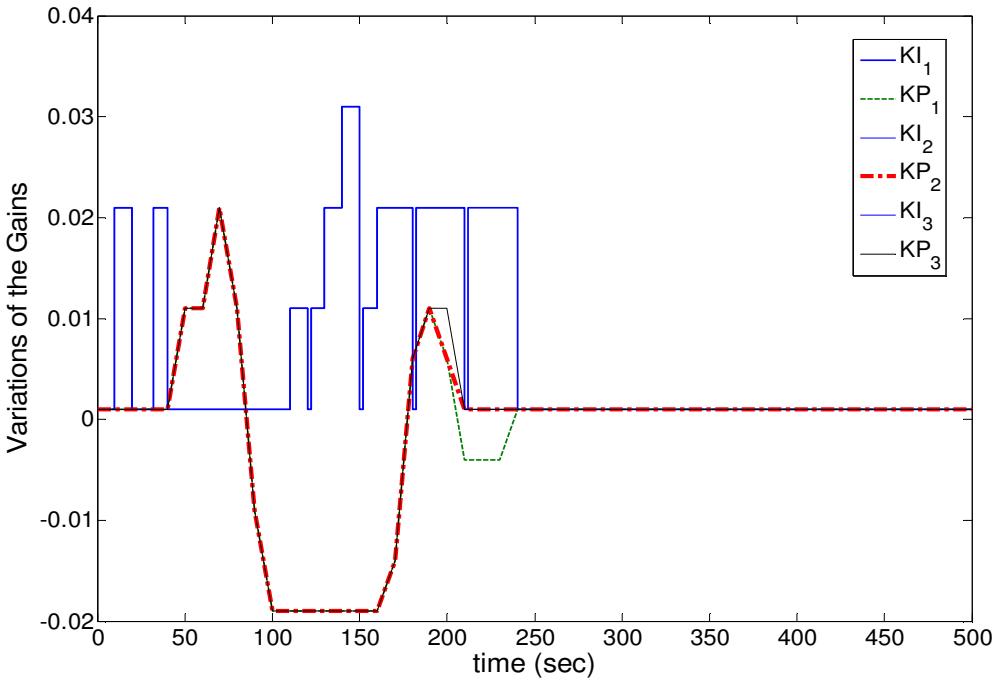


Figure 4.14: Variations of PI controller gains of the three areas for scenario 2.

Chapter 5

Load frequency control based on NERC standards

5.1 Introduction

The North American Electric Reliability Council (NERC) has released new control performance standards CPS1 and CPS2 in 1977 to determine the effectiveness of automatic generation control (AGC) [26]. Each area is required to report its compliance with these standards to NERC at the end of each month. These new standards replaced the old control performance criteria (CPC). CPS1 and CPS2 are based on statistical theories and are mathematically more powerful. Violating these standards may cause serious problems in power systems operation and the units responsible for these violations will be penalized by NERC. The following chapter will discuss a proposed strategy in order to comply with NERC's standards and reduce additional control effort and wear and tear on the system equipment. These standards are explained in detail before describing the controller structure.

5.1.1 CPS1

CPS1 is the first control performance standard which deals with the behavior of the control areas in the long term. This standard is defined on a 12 month period as: the average over 12 month of the "one minute averages" of a control area's ACE divided by "ten times its frequency bias factor" multiplied by the "one minute average of the interconnection frequency error". This value should be less than the square of a given constant ε_1, the target frequency bound. This definition is better expressed by the following equation:

$$AVG_{12-month}\left[\left(\frac{ACE_i}{-10B_i}\right)_1 \times \Delta F_1\right] \le \varepsilon_1^2 \tag{5.1}$$

where

B_i frequency bias of the i^{th} control area in MW/0.1Hz

ε_1 targeted frequency bound for CPS1

ΔF interconnection frequency error

$(.)_1$ one-minute average

In order to simplify the above definition and have an expression for CPS1, two terms i.e. a compliance factor (CF) and a 1-minute average compliance factor (CF1) are defined [4]:

$$CF = AVG_{12-month}\left[CF_1\right] \tag{5.2}$$

$$CF_1 = \left[\left(\frac{ACE}{-10B_i}\right)_1 \times \left(\frac{\Delta F}{\varepsilon_1^2}\right)_1\right] \tag{5.3}$$

From these definitions the CPS1 is defined

$$CPS1 = \left(2 - CF\right)\times 100\% \tag{5.4}$$

According to NERC, CPS1, obtained from equation (5.4) should not be less than 100% at any time in order to comply with standards.

5.1.2 CPS2

The second performance standard CPS2 is defined for 10 minute intervals and requires that the 10-minute average of the area control error for each area be less than or equal to a constant L_{10} given by equation (5.6).

$$AVG_{10-\min ute}\left(ACE_i\right) \le L_{10} \tag{5.5}$$

$$L_{10} = 1.65\varepsilon_{10}\sqrt{\left(-10B_i\right)\left(-10B_s\right)} \tag{5.6}$$

In the above equations B_s is the summation of the frequency bias settings for all control areas in the studied power system. In order for a control area to comply with NERC's standards, the level of its compliance should be more than 90%. The compliance percentage is calculated form the following equation

$$CPS2 = \left[1 - \frac{Violations_{month}}{Total\ \ periods - Unavaiable\ \ periods}\right]\times 100\% \tag{5.7}$$

The term $Violations_{month}$ indicates the number of times the 10-minute average of ACE is greater than L_{10} in one month.

5.2 LFC Control Design Based on NERC's Standards

5.2.1 Application of RL in control design

The controllers designed previously presented acceptable performance when applied to load frequency control. Driving the ACE as close as possible to zero was the main objective for these controllers. However, often these controllers result in tight control which is sometimes unnecessary and will only increase the costs of control due to unnecessary fuel consumption. Therefore a controller that maintains the system in compliance with the standards and in the mean time prevents tight control is of a great interest in today's power systems. In addition, the mentioned control will also reduce the wear and tear of the system equipment by decreasing the control effort and excessive excursions.

The desired control technique should have the ability to adjust the parameters of the controller according to the level of compliance with the standards, which in this case are CPS1 and CPS2. Intelligent techniques are therefore more capable of providing these requirements for the controllers with these characteristics. Different learning methods can be applied to solve this problem. Fuzzy logic is one of these methods that has been applied to this problem in [4] and has shown satisfactory performance when applied to a three area power system. However, as discussed before fuzzy logic may not be as powerful as a method that has the ability to learn with experience and adjust the control parameters according to the new operating conditions.

The nature of the load frequency control requires a trade off between the cost and the performance and consequently the terms rewards and penalties are among the first terms that come to mind. From what explained one may conclude that reinforcement learning can be more applicable to this problem due to the way these methods reward or penalize the actions.

5.2.2 RL-based load frequency control considering CPS1 and CPS2

Reinforcement learning can be applied to change a variety of control parameters depending on the type of the controller. In the problem in hand the RL techniques are applied to tune the proportional and integral gains of a PI controller by constantly observing the level of the compliance with NERC's standards. The RL agent decides

whether it is required to change the gains and how much these changes should be. By preventing the unnecessary gain changes the governor setpoint or raise/lower signal ΔP_c will be modified with less frequency especially when the control area is in high compliance with the standards. This will considerably reduce the wear and tear on the mechanical equipment. Figure 5.1 illustrates the structure of the RL-based load frequency control based on NERC standards.

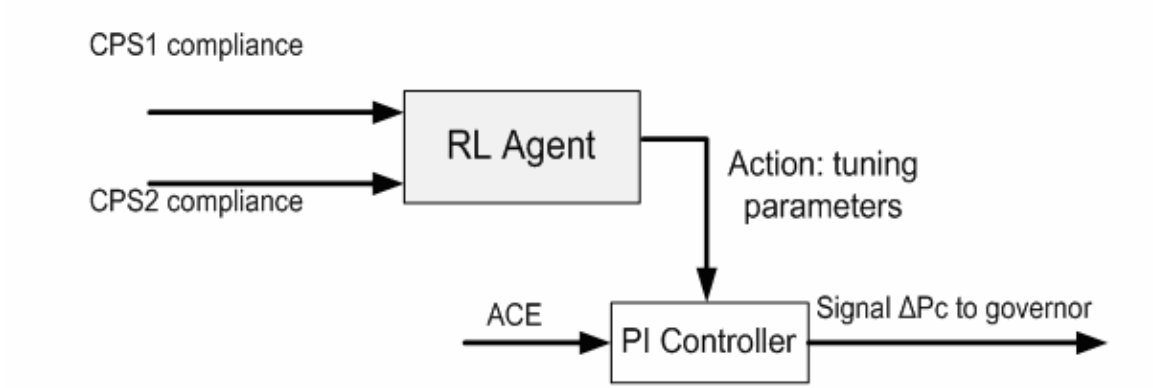


Figure 5.1: Reinforcement learning based load frequency control considering NERC's standards.

In the above RL-based design the RL agent uses the information gained from the environment, which in the case reflect the level of compliance with CPS1 and CPS2, and by observing the states and rewards gained adjust the gains by changing the tuning parameters, α_I and α_P. These parameters in fact change the proportional and integral gains in accordance to the level of compliance. The PI controller will then act on the ACE signal and change the governor setpoint.

The most important factor in design of the above mentioned controller is to define the reward function in a way that it penalizes the action that reduces the compliance with the standards and rewards those who increase this compliance. However, in order to prevent the unnecessary control effort, an action that keeps the compliance level within the ranges which are acceptable by the standards with less control effort should be rewarded more than an action that increases the compliance further but will cause more costs and changes on the governor setpoint. Therefore the reward function in this case should necessarily be more complex than the one defined in Chapter 4.

The same step by step procedure of the previous chapter is followed to design the controller, i.e. to start from the definition of the reinforcement learning elements, which are state, reward and the actions.

As explained in the previous case, the definition of the state is dependent on the signals that are serving as inputs to the learner. Signals that determine the level of compliance with CPS1 and CPS2 are considered as inputs in this case and therefore the states are defined based on variations of these signals, but fist they should be defined.

The first signal that is considered as an input to the RL agent is a measure for compliance with CPS1 standard. Since CPS1 is a standard associated with the performance of the controller within a 12-month period, therefore, a type of a signal that can represent the accumulated average compliance [4] since the start of calculation until the current time should be used in order to be able to prevent any possible violations of this standard before it is occurred. This signal is defined as follows.

$$CF_{accumulated} = Average_{t_0 \to t}\left[CF_1\right] \tag{5.8}$$

where t_0 is the start time and t is the current time of the simulations. Also, the same approach is used to define the CF over an entire year, where T is the end of a 12-month period.

$$CF = Average_{t_0 \to T}\left[CF_1\right] \tag{5.9}$$

From these equations it is understood that if the accumulated average is not violating the limits it is guaranteed that the controller will comply with CPS1. Therefore, one of the tasks of the controller is to constantly monitor variations of this signal and make sure that it is not violating the limits defined by the CPS1 standard.

The second input signal shows the compliance of the each control area with CPS2 and is defined directly based on the definition presented in equations (5-5) through (5-7). In this case the signal is fed into the reinforcement learner every 10 minutes due to the fact that the CPS2 standard is based on the 10-minute averages of the ACE signal. Therefore if the controller keeps the 10-minutes averages of the ACE less than the parameter L_{10} then is guaranteed that the CPS2 standard is not violated over a month.

Now that the input signals are defined, the states of the system that the RL learner will identify are described. Based on the variations of these two signals different state levels

can be identified. Table 5.1 summarizes these levels while seven state levels are defined for the RL problem. In this Table the possible values of CPS1 and CPS2 compliance factors are divided into three main levels of low, medium and high. These levels are defined based on the design preferences and also based on the control effort that is desired to be devoted for the load frequency control purposes.

TABLE 5.1
STATE LEVELS FOR THE REINFORCEMENT LEARNING BASED CONTROLLER

State Level	CPS1 compliance factor	CPS2 compliance factor
1	High	-
2	Medium	High
3	Medium	Medium
4	Medium	Low
5	Low	High
6	Low	Medium
7	Low	Very Low

As explained before the role of the RL agent in this case is to change the tuning parameter α in order to satisfy the standards and in the mean time reduce the fuel cost and the wear and tear of the equipment. Therefore, it is expected that the learner will learn to decrease this parameter when the compliance factors of CPS1 and CPS2 are low, which means that the control area is in high compliance with the standards. On the contrary, when any of the compliance factors are high, which shows a poor compliance, the tuning parameter should be increased based on the values of these factors.

Based on the characteristics of the controller described above, the actions of the controller are defined. The RL agent will increase or decrease the tuning parameter a certain discrete amount, which is defined by the designer. These increments are important as they have a substantial effect on the performance of the controller. Generally one can define two sets of increments Δ_1 and Δ_2 while $|\Delta_1| > |\Delta_2|$. This definition gives the freedom of action to the agent and defines four actions for each RL agent: Increase or decrease the tuning parameter by Δ_1 or Δ_2. The difference between these two values should be a reasonable amount. For instance a small difference does not justify having

more actions as then there would not be a considerable difference between the two actions taken. Also, a huge difference can lead to selection of a big PI controller gains which in some cases can take the system to a point where the agent will no longer be able to learn the correct action as the system has reached the instable region.

The last important component of this controller that should be defined by the designer is the reward function. The ideal reward function should penalize those actions that violate the standards and in the mean time reward the actions that keep the control area's performance within CPS1 and CPS2 standards with less control effort and fuel consumption. Based on these characteristics different functions could be defined. However, these functions are expected to differ in their behavior and the one with the best performance should be selected. Next a function that has demonstrated the best performance is introduced.

The selected function is composed of three terms. The first term (r_1) is associated with the CPS1 standard and equals to -1 if the accumulated average factor is more than one (which means the standard is being violated). On the contrary, if the compliance factor is within the limits this term will have a value of zero. The value of -1 is in fact a negative reward, or a penalty applied to the selected action. The second term (r_2) is related to the level of compliance with CPS2 and is defined in the same way as the previous case. The last term (r_3) is perhaps the most important part of this function as it should reduce the unnecessary control effort by penalizing these actions. The more the tuning parameter, the more the penalty would be. Therefore it would be a linear function of the tuning parameter α. These definitions are summarized in the following equations.

$$R(t) = r_1 + r_2 + r_3 \tag{5.10}$$

where,

$$r_1 = \begin{cases} -1 & if\ CF_{accumulated} \geq 1 \\ 0 & otherwise \end{cases}$$

$$r_2 = \begin{cases} -1 & if\ AVG[ACE]_{10-\min ute} \geq L_{10} \\ 0 & otherwise \end{cases}$$

$$r3 = -100 \times \alpha \tag{5.11}$$

5.3 Simulation Results

In this section the two area system described before is tested with the mentioned controller and the integral gains of the PI controllers are tuned with this technique. Each area is controlled by an RL controller that changes the gains of the P controller. Each area is subjected to a relatively large disturbance in the second hour of simulation, while the loads of both areas have constant random changes. The simulation results after three hours of simulation are presented in the following Figures.

Figure 5.2 illustrates the behavior of the controllers by presenting the variations of the ACE signals and governor setpoints for both areas. Figure 5.3 shows the learned tuning parameter for both areas while the areas are subjected to disturbance. In order to be able to judge the behavior of the learners in tuning the gains of the PI controllers, the variations of the compliance factors are presented in Figure 5.4.

From the Figures it is observed that the controllers learn to decrease the gains whenever control areas are in high compliance with NERC's standards and they increase the gains when these standards are violated. After the major disturbance the gains are increased for a while until the time the ACE has reduced its values and is reached more close to zero.

It should be noted that the model is simulated with nonlinearities such as generation rate constraint (GRC) of 10%/min and a governor dead-band of 0.01 and therefore, the controller is proven to be effective for non linear systems. The proposed method provides the conventional PI controllers with an ability to adapt themselves to various operating conditions without necessarily knowing the model of the system. Thus this controller can be applied to many systems with different parameters and yet can learn the proper settings of the gains. However, as mentioned before, the selections of design parameters such as the actions and the rewards play an important role in the behavior of the controller and the learning procedure.

Another observation which is carried out from the Figures is that the governor set points of both control areas are smoothly changed and therefore the unit maneuvering is considerably reduced although the loads are constantly changing.

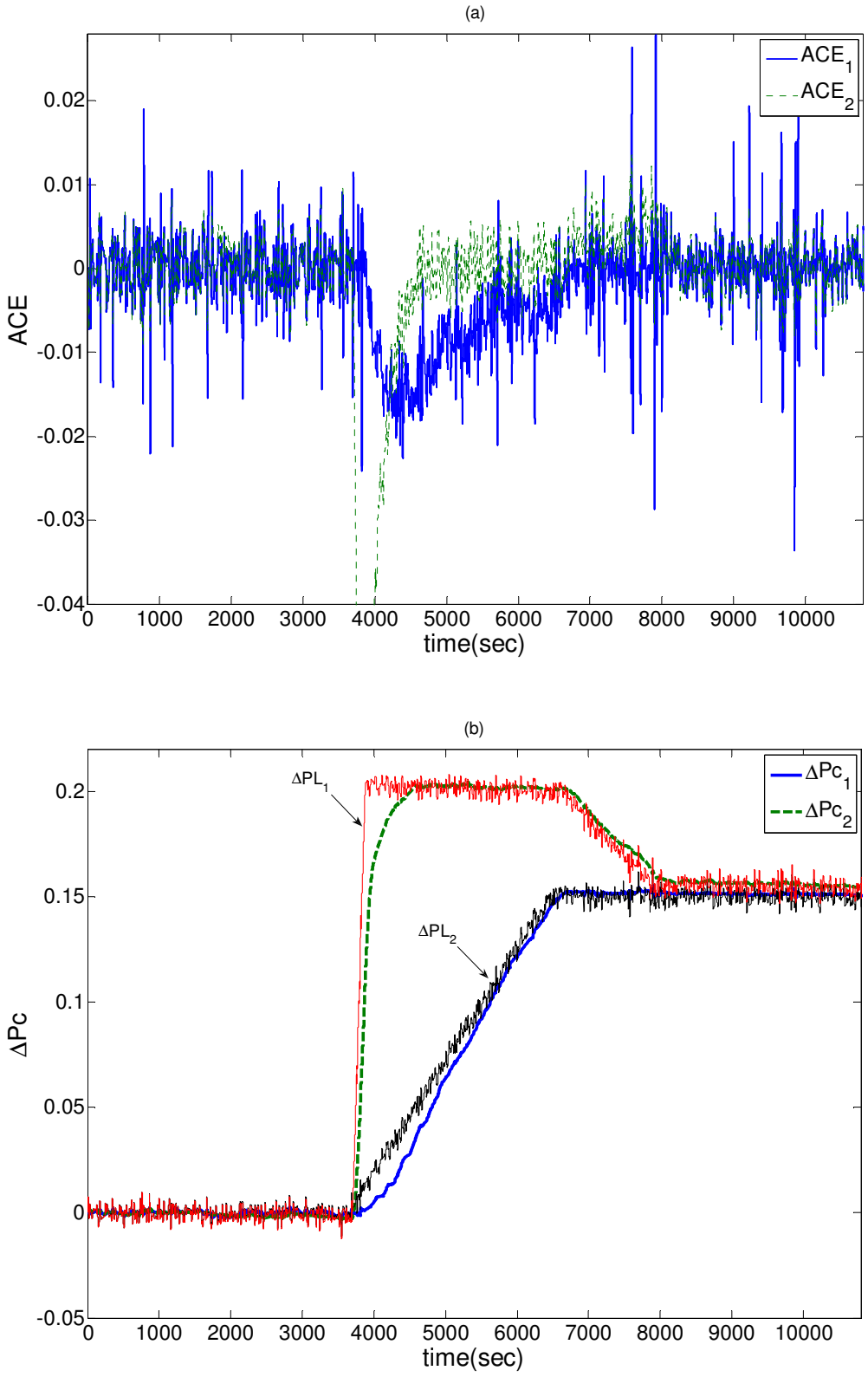


Figure 5.2: (a) Area control error and (b) governor setpoints for two area system taking into account NERC's standards.

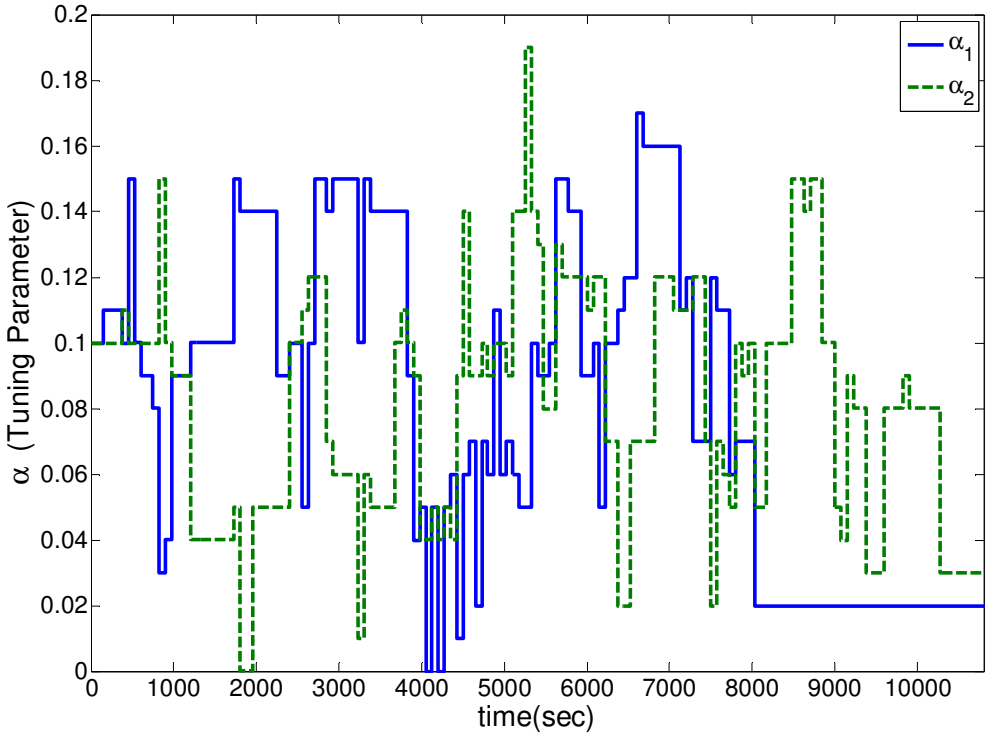


Figure 5.3: Variations of tuning parameter for both control areas when loads are constantly changing.

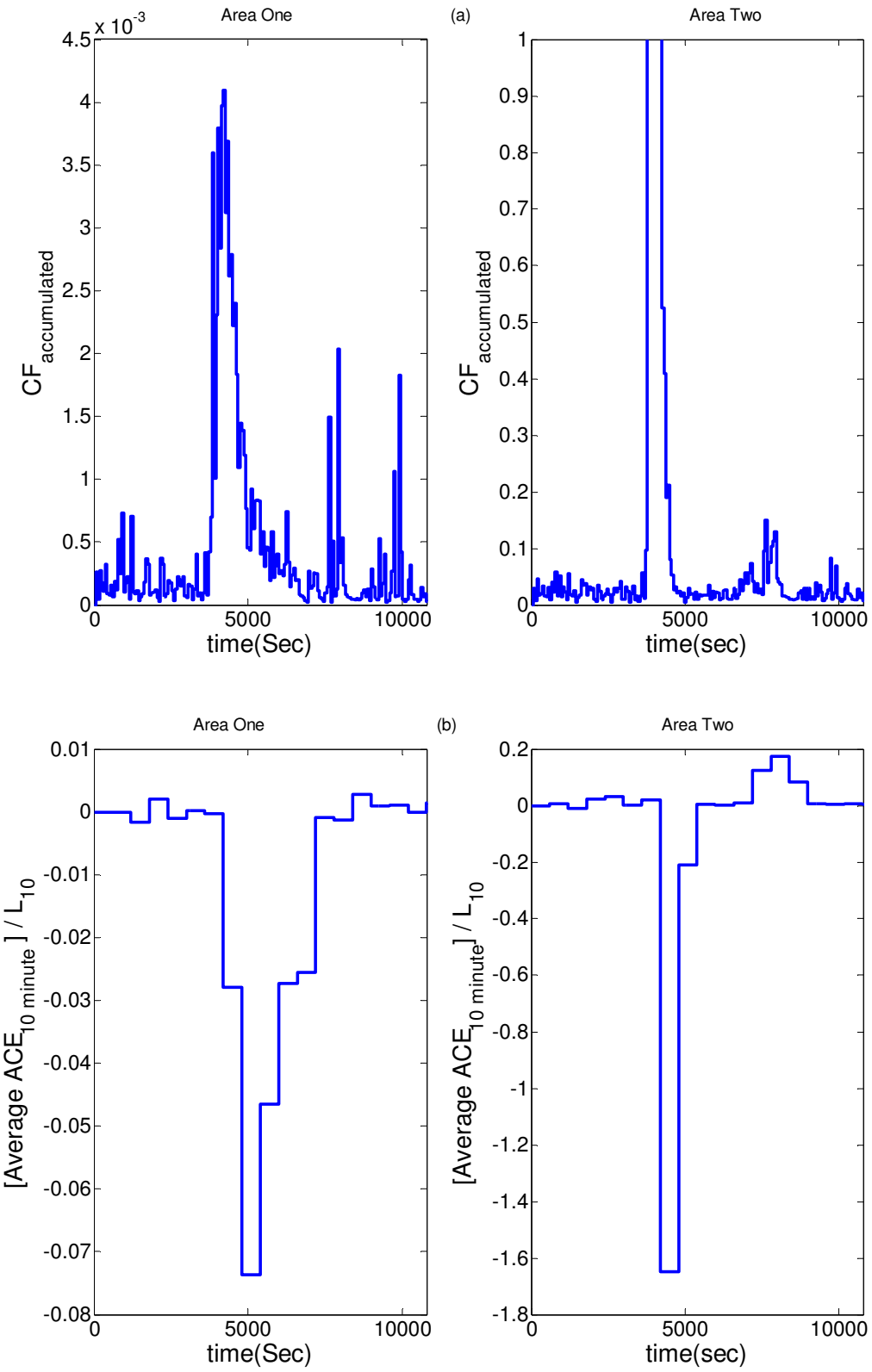


Figure 5.4: (a) CPS1 and (b) CPS2 compliance factors for both areas.

Chapter 6

Conclusion

PI controllers have widely been used in industry for the LFC purposes; however, the gains of these controllers are fixed and are tuned once a month by trial and error. To compensate this disadvantage, adaptive controllers are designed, but as described in literature survey, these controllers are usually complicated and high order which makes their application to real systems impractical. This research presents a novel control architecture based on a reinforcement learning methods in order to enhance load frequency control. RL methods are applied to conventional PI controllers to keep the simple structure of the controller and n the mean time provide the controllers with the ability to adapt themselves to different operating conditions with less human interaction.

In Chapter 3 the basic fundamentals of the reinforcement learning methods are presented. All the problems in this research are formulated as markov decision processes (MDP) which have certain characteristics such as a discrete state and action space. RL methods propose different procedures to solve MDP problems by gaining experience through interaction with environment. Some of these methods are model based which essentially require some information related to the model of the system in order to make proper decisions on the actions that should be taken in the next step. In contrary the other major groups of RL methods are not based on the system model and directly estimates the Q-function by experienced gained. Action-value function or simply Q-function indicates the value of an action in a specific state in terms of satisfying the objective. Q-learning is one of the model free methods that is extensively used in the literature for control applications. This thesis also takes advantage of Q-learning mostly because it requires no prior knowledge from the system. The proposed controllers learn the proper settings of the PI controllers each time there is a disturbance on the system. Two different approaches are studied in this text in order to observe the effectiveness of this method.

In Chapter 4 a new decentralized, RL based control design was developed for load frequency control applications. Each area is equipped with an RL based control that changes the proportional and integral gains of the PI controllers according to the operating conditions and the disturbances applied to the system. The objective in this case was to drive the area control error (ACE) signal back to zero with proper control actions. Hence, the reward function was defined in a way that the more an action drives ACE to zero the more would be the reward and vice-versa. The disturbances considered for the case studies were large changes in demands accompanied by a white noise signal. The two-area system, with nonlinearities such as GRC and governor deadband, was studied in this chapter while considering two different reward functions. The results determined the importance of the definition of the reward function in performance of the RL agent. Three-area system was also simulated to illustrate the applicability of the proposed decentralized control structure to large systems, while two different disturbance scenarios were simulated separately.

The most important advantage of this controller is that it has a very simple structure and does not need any information from the system and its states to set the gains. Simple measurements such as the frequency and tie-line flows serve as the inputs to the RL agent in each time step the agent makes the decisions on the proper actions. Additionally, the ability of the controllers to learn and adapt themselves to different operating conditions and disturbances in the system makes them more appealing in power system applications.

The North American Electric Reliability Council (NERC) released new control performance standards CPS1 and CPS2 in 1977 and each control area is required to comply with these new standards. As long as a control area is able to keep the ACE variations within the limits defined by these criteria it will not be penalized by the NERC. Therefore, the control areas should reach this goal by any control means they can. However, a tight control and a control that allows small variations of ACE within the limits are both treated the same by NERC but apparently the former will cost more for the control area by increasing the fuel consumption and wear of the mechanical equipment. Consequently, the ideal controller should comply with NERC's CPS1 and CPS2 standards and in the mean time try to decrease the control cost by preventing the unnecessary tight control actions.

An important characteristic of the RL methods is that depending on the desired objective, the agent learns the proper policy towards satisfying the goal. This is achieved through the proper definition of the reward function, because usually the actions that maximize this function are selected. Therefore in order to change the objective of the control, only the reward function should be changed and the main structure of the controller would remain the same as before. This feature provides the controller with the flexibility to be applied to different applications with just a slight modification in the RL definitions.

Chapter 5 proposed a new method for tuning the gains of the PI controllers based on reinforcement learning methods. The main objective of the controller was to keep the variations of the ACE signals in the range to comply with the NERC's standards but, at the same time decrease the unnecessary control effort and thus the control costs. Based on what explained in the previous paragraph this objective is achieved through modifying the reward function. The desired reward function penalizes each action violating CPS1 and CPS2. Also, this function gives more reward to the action that satisfies the NERC's criteria with less control and unit maneuvering. The simulation results for three hours showed that the tuning parameters increase when there is a large increase in the load of each area to provide the extra generation by changing the valve opening in order to decrease the compliance factor. When these factors are decreased (which means the area is in more compliance with standards) the tuning parameters are decreased.

In conclusion, reinforcement learning methods have shown satisfactory performance when applied to power systems. The learning ability of these methods makes them more adaptive and applicable to different problems with changing operating conditions. Two different approaches were presented in this thesis and the simulation results show that the controllers were able to handle both problems with satisfactory performance. Also, due to the fact that by modifying the reward function we can apply the same technique to different control problems, the RL methods are not confined to PI controllers and they can be applied to other, more complex ,such as non-linear, controllers as well. Although these methods are expected to be flexible because they are based on learning and improving through interaction with the environment, they are not expected to perform very fast in the initial stages of learning.

Appendix A

RL: Simulink Block and MDL Files

In order to simulate the power system and observe the results Simulink is used as the main tool for modeling and simulation. However, to use the reinforcement learning algorithm and communicate with the system model, MATLAB S-functions are used and incorporated in Simulink file. Finally this S-function is used with extra links to the outside *mdl* file and is represented in a form of a Simulink block which is shown in Figure A.1.

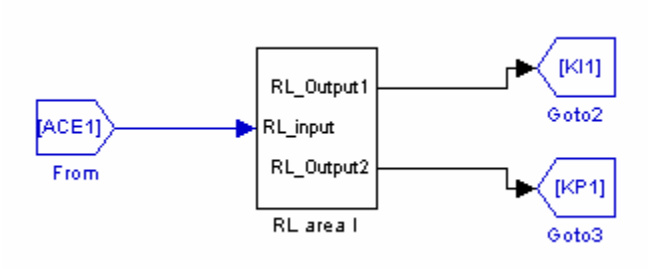


Figure A.1: RL block in Simulink

An advantage of having this block is that it can be used in any control application by adjusting the number of inputs and outputs and the reward functions defined in the S-functions. Next different parts of this block are described in detail.

Figure A.2 represents the main structure of this block. As it is observed, the RL controller consists of two major parts. The first block (rl_1) determines the greedy action and takes the action which will be the ultimate output of the system. This decision making is based on the inputs from the second block (rl_2) which calculates the state and reward gained by taking an action. Also, this block updates the values of Q-function which is later used by the first block to find the proper action. Figure A.3 and A.4 present the details of these block diagrams.

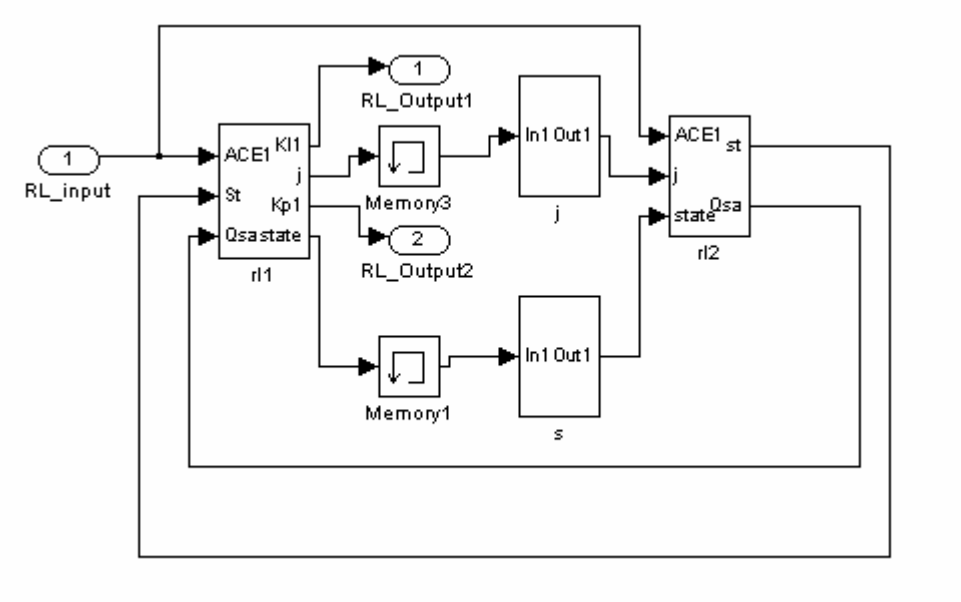


Figure A.2: The main structure of the RL block

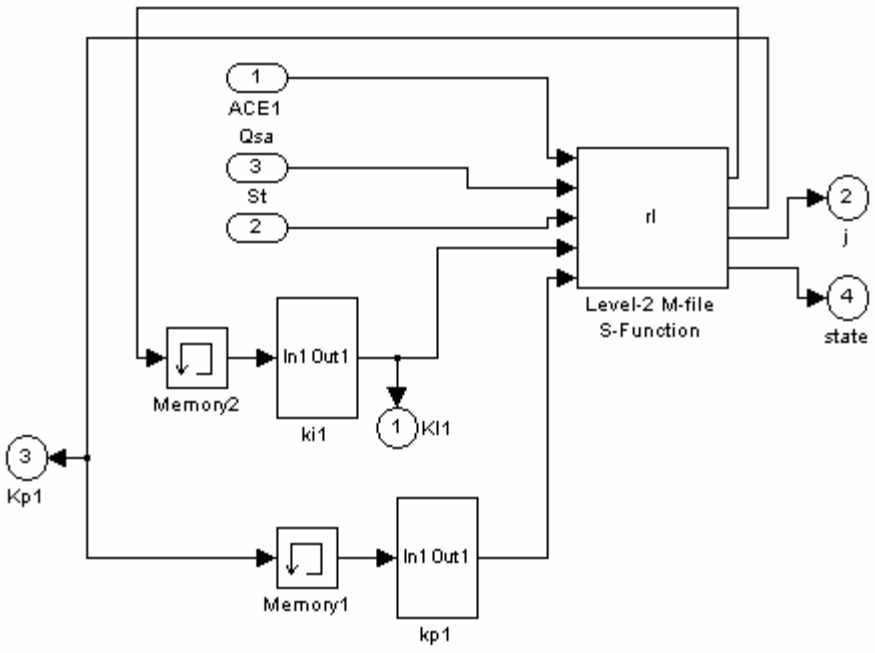


Figure A.3: The interior of the rl_1 block

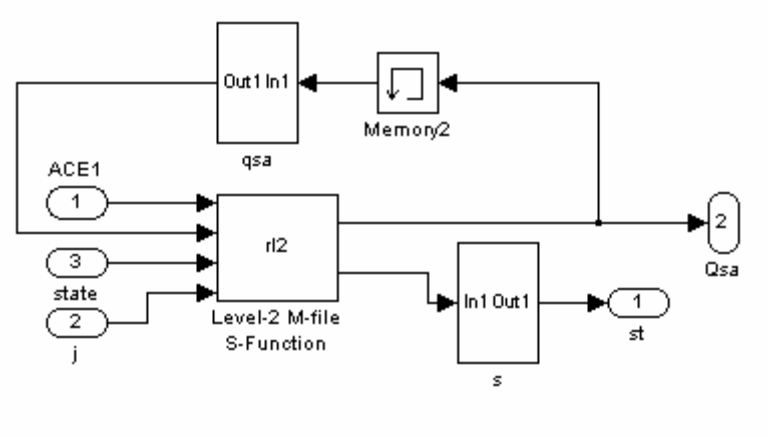


Figure A.4: The interior of the rl_2 block

The reason the reinforcement learning algorithm is divided into two different blocks is that there should be different timing in order to see the effect of a taken action on the system and observe the reward. One of the advantages of using S-functions in the Simulink block is the freedom of having different timings in the simulation. Therefore, by assigning different timings for the mentioned two blocks the Q-functions can be updated more effectively.

References

[1]. Richard S. Sutton, and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[2]. O.I. Elgerd, *Electric Energy System Theory: An Introduction,* Tata Mc-Graw hill, 1982.

[3]. L.P. Kaelbling, M.L. Littman, and A.W. Moore, "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research*, Vol. 4, 1996, pp. 237-285.

[4]. D. Rerkpreedapong, "Novel Control Design and Strategy for Load Frequency Control in Restructured Power Systems", *PhD. Dissertation,* Advanced Power and Electricity Research Center, West Virginia University, 2003.

[5]. Richard C. Dorf and Robert H. Bishop, *"Modern Control Systems",* Addison-Wesley Publishing Company, 1995.

[6]. D. Rerkpreedapong and A. Feliachi. "Fuzzy Rule Based Load Frequency Control in Compliance with NERC's standards", *Proc. of IEEE Power Engineering Society General Meeting, Vol. 3, Issue 25-25 July 2002*

[7]. D. Rerkpreedapong and A. Feliachi., "Robust Load Frequency Control Using Genetic Algorithms and Linear Matrix Inequalities", *IEEE Transactions on Power Systems*, vol. 18, no. 2, May 2003.

[8]. Y.L. Abdel-Majid and M.M. Dawoud, "Genetic Algorithms Applications in Load Frequency Control", *Proc. of first international conference on GALESIA,* pp. 207-213, Sep. 1995.

[9]. Y.L. Karnavas, "On the Optimal Load Frequency Control of an Interconnected Hydro Electric Power System Using Genetic Algorithms", *Journal of series on energy and power systems,* pp. 134-148, 2006.

[10]. C.S. Chang and W. Fu, "Area Load Frequency Control Using Fuzzy Gain Scheduling of PI Controllers", *Electric Power Systems Research,* Vol. 43, No. 2, pp. 145-152, Aug. 1997.

[11]. C.L. Karr, "Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm", *Proc. Of 4th Int. Conference of Genetic Algorithms*, pp. 129-139, 1995.

[12]. A. Homaifar and E. McCormick, "Simultaneous design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms ", *IEEE Trans. on Fuzzy Systems*, Vol. 3, No. 2, pp. 129-139,1995.

[13]. C.F. Juang and A.F. Lu, "Power System Load Frequency Control with Fuzzy Gain Scheduling Designed by New Genetic Algorithms", *Proc. of IEEE International Conference on Fuzzy Systems,* Vol. 1, pp. 64-68, 2002.

[14]. H.D. Mathur and S. Ghosh, "A Comprehensive Analysis of Intelligent Controllers for Load Frequency Control", *Proc. of 2009 IEEE Power India Conference*, April 2006.

[15]. A.P. Birch, A.T. Sapeluk and C.S. Özveren, "An Enhanced Neural Network Load Frequency Control Technique", *Proc. of International Conference on Control*, Vol. 1, pp. 409-415,March 1994.

[16]. D.K.Chaturvedi, P.S. Satsangi, and P.K. Kalra, "Load Frequency Control: a Generalized Neural Network Approach", *Electric Power Systems Research,* Vol. 21, pp. 405-415, 1999.

[17]. O. Kuljaca, F.L. Lewis, and S. Tesnjak, "Neural Network Frequency Control for Thermal Power Systems", *Proc. of 43rd IEEE Conference on Decision and Control*, December 2004.

[18]. L. Koszalka, R. Rudek, and I.Pozniak-Koszalka, "An Idea of Using Reinforcement Learning in Adaptive Control Systems", *Proc. of International Conference on Systems and International Conference on Mobile Communications and Learning Technologies,* April 2006.

[19]. D. Ernst, M. Glavic, and L. Wehenkel, "Power Systems Stability Control: Reinforcement Learning Framework", *IEEE Trans. on Power Systems,* Vol. 19, No. 1, February 2004.

[20]. M. Glavic, D. Ernst, and L. Wehenkel, "Combining Stability and Performance-Oriented Control in Power Systems", *IEEE Trans. on Power Systems,* Vol. 20, No. 1, February 2005.

[21]. M. Glavic, D. Ernst, and L. Wehenkel, "Damping Control by Fusion of Reinforcement Learning and Control Lyapunov Functions", *Proc. of 38th North American Power Symposium"*, pp. 361-367, September 2006.

[22]. T.P. Imthias Ahamed, P.S. Nagendra Rao, and P.S. Sastry, "A reinforcement Learning Approach to Automatic Generation Control", *Electric Power Systems Research,* Vol. 63, pp. 9-26, 2002.

[23]. N. Jaleeli and L.S. VanSlyck, "NERC's New Control Performance Standards", *IEEE Trans. on Power Systems,* Vol.14, No.3, August 1999.

[24]. Haadi saadat, *Power System Analysis*, Mc-Graw hill, 2002.

[25]. S.K. Aditya and D. Das, "Design of Load Frequency Controllers Using Genetic Algorithm for Two Area Interconnected Hydro Power System", *Electrical Power Components and Systems,* Vol. 31, pp. 81-94, 2003.

[26]. North American Electric Reliability Council (NERC), "Performance Standard Training Document", *in operating manual,* pp. ps1-20, Nov. 1996.

[27]. J.S. Heo and K.Y. Lee, "A Multi-Agent System-Based Intelligent Heuristic Optimal Control System for A Large-Scale Power Plant", *Proc. of IEEE Conference on Evolutionary Computation,* July 2006.