Reinforcement Learning based Load Frequency Control for Power Systems

Mohaddeseh Sadeghinejad
Department of Electrical Engineering, Imam Khomeini International University, Qazvin, Iran

Mehdi Rahmani
Department of Electrical Engineering, Imam Khomeini International University, Qazvin, Iran

Abstract

Load Frequency Control (LFC) holds significant importance in the operation of power systems. It is a critical system that requires intelligent methods to address its associated challenges. In this article, we employ Reinforcement Learning (RL) as a solution to tackle the LFC problem specifically for two turbines.

RL proves to be a promising approach for optimizing LFC due to its ability to learn from experience and make decisions accordingly. By utilizing RL, we aim to enhance the performance and efficiency of LFC in power systems.

The application of RL in LFC involves training an RL agent to make decisions based on observed states, such as frequency and tie-line power deviations. The agent learns from feedback in the form of rewards and updates its policy accordingly. Through this iterative learning process, the RL agent aims to find an optimal control strategy for maintaining system frequency and power balance.

By employing RL techniques, we strive to improve the effectiveness and reliability of LFC in power systems with two turbines, ultimately contributing to the stability and operational efficiency of the overall power grid.

Keywords: LFC, reinforcement learning, non-reheated and reheated turbines.

I. Introduction

Due to rapid economic growth, the demand for electricity has increased, and there has been a significant increase in the use of distributed energy sources. As a result, power grids have become larger and more complex, leading to an increase in factors that can disrupt the power grid. These disturbances, such as changes in load, can negatively impact the quality of the power frequency and pose a security threat to the grids. [1]

Load Frequency Control (LFC) is a vital system in power system operation. Its purpose is to maintain a consistent frequency, distribute the load among generators, and manage tie-line interchange schedules. LFC ensures a stable power system by adjusting generator output in response to frequency changes. It plays a crucial role in supplying reliable power with good quality. Load-frequency control is necessary for stable power system operation and is employed to meet local load demands and restore the system's steady-state frequency (Δf) to zero. Therefore, it is important to implement load-frequency control in power grids to maintain a stable power frequency, typically at values like 50 and 60 Hz. Load-frequency control plays a crucial role in maintaining the balance between power supply and demand, especially when a large number of distributed energy sources are connected.

As modern power systems have become more complex, various advanced control methods have been proposed for load-frequency control (LFC). These methods include optimal control [2-4], variable structure control [5, 6], adaptive control [7], self-regulation [8, 9], intelligent control [10], robust control [11], [12], as well as the application of machine learning and reinforcement learning (RL). Reinforcement learning is a technique in machine learning that imitates the learning abilities of humans and animals. It provides a model-free approach to solving optimal control problems expressed as Markov decision processes [1]. RL-based controller are applied in various practical applications including robotic systems, buildings energy, Tuning of an Aircraft Pitch PID Controller, etc. However, to the best of the authors' knowledge, it has not been applied to design the load frequency controller for power systems so far. In this paper, we first present the LFC model of a power system with two different turbines including reheated and non-reheated ones. Then, by introducing an appropeiate reward function, the design steps and parameter tuning of the load frequency control based on reinforcement learning is presented. Finally, the proposed contoller is applied to the LFC problem in power systems.

The remainder of this paper is organized as follows: Section II presents, history, basics and applications of Reinforcement learning. In Section III the LFC model of a power system with two different turbines are presented. Section IV proposes parameter tuning and numerical studies and simulations and Section V presents conclusions with future work.

II. History, importance and introduction of Reinforcement Learning

Reinforcement learning is a field within machine learning that focuses on training intelligent agents to make sequential decisions in an environment. It draws inspiration from behavioral psychology and aims to develop algorithms that learn through trial and error interactions. The concept of reinforcement learning stemmed from the term "optimal control" that emerged in the late 1950s. It involved formulating a problem by designing a controller to minimize a measure of a system's behavior over time. Bellman introduced the concept of Markov decision processes (MDPs), which are fundamental to reinforcement learning, to formulate optimal control problems. Recent years have seen significant advancements due to the availability of computational resources and the development of deep learning techniques.

Reinforcement learning is a unique category of machine learning where the learner, or agent, learns to associate situations with actions that maximize a delayed reward signal. Unlike supervised or unsupervised learning, RL doesn't rely on a "teacher" to provide instructions for each action. Instead, it makes decisions through trial-and-error search and recognizes the delayed reward it receives from the environment it interacts with. RL is important for several reasons. Firstly, it provides a framework for training autonomous agents to learn and adapt in complex and dynamic environments without explicit programming. This makes it valuable in domains where explicit rules or solutions are difficult to specify, such as game playing, robotics, and autonomous driving. Secondly, reinforcement learning enables agents to learn optimal behavior through trial and error. By receiving feedback in the form of rewards or penalties from the environment, agents can improve their decision-making abilities over time. This is particularly useful in scenarios where the optimal strategy is not known in advance, and the agent must explore different actions to determine the best course of action. Additionally, reinforcement learning has contributed to advancements in our understanding of learning processes in both artificial and biological systems. Studying how artificial agents learn through reinforcement provides insights into the underlying mechanisms of learning and decision-making.

In RL, an agent, which is the decision maker, takes an action (a_t) in a given state (s_t) and receives a reward (r_t). After receiving the reward, the agent uses it to update the parameters of the "Policy Function" and the "Value Function."
The goal is to maximize the total discounted expected reward. In reinforcement learning, developers establish a system of rewarding desired behaviors and punishing negative behaviors. Positive values are assigned to desired actions to encourage the agent to repeat them, while negative values are assigned to discourage undesired behaviors.

In a dynamic sequential decision-making process, the state ($s_t \in \mathcal{S}$) represents a specific condition of the environment at discrete time steps (t=0,1,…). The agent, through its interaction with the environment, observes this state and selects a deterministic or stochastic action ($a_t \in \mathcal{A}$) with the goal of maximizing future returns. In return, the agent receives an instantaneous reward ($r_{t+1} \in \mathcal{R}$) as it transitions to the new state (s_{t+1}). This reward is typically quantitatively measured, and together, a sequence of states, actions, and rewards is generated, forming a Markov Decision Process (MDP). Figure 1 shows how the agent and the environment interact with each other in an MDP (Markov Decision Process).

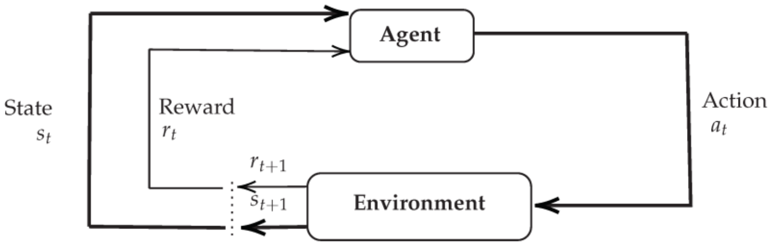

Figure (1) The interaction between agent and environment in an MDP.

Reinforcement learning relies on a reward signal, which is a single value generated by the environment. This signal serves as a measure of how well the agent is performing in achieving its task objectives. To put it simply, the reward quantifies the effectiveness of taking a specific action given a certain observation (state). During the training process, the agent updates its policy based on the rewards it receives for different combinations of states and actions. In general, positive rewards are used to encourage certain actions by the agent, while negative rewards also known as penalties are employed to discourage other actions. A well-designed reward signal serves as a guide for the agent, helping it maximize its long-term reward expectation.

In reinforcement learning, the agent consists of two main components: a policy and a learning algorithm.The policy serves as a guide for the agent, determining which actions to take based on the observations it receives from the environment. Typically, the policy is represented by a function approximator with adjustable parameters, such as a deep neural network. On the other hand, the learning algorithm plays a crucial role in continuously updating the policy parameters. It does so by considering the actions taken by the agent, the observations received, and the associated rewards. The ultimate aim of the learning algorithm is to discover an optimal policy that maximizes the total reward accumulated throughout the task. In essence, reinforcement learning involves the agent learning the most effective behavior through repeated interactions with the environment, without requiring direct human involvement. Table 1 also declares the relation between reinforcement learning and classical control systems and Figure 2 Showes the steps involved in training an agent using reinforcement learning.

Figure (2) The steps involved in training an agent through reinforcement learning.

Table 1- Translation of the behavior and policy of reinforcement learning into control system representation.

Reinforcement Learning	**Control Systems**
Policy	Controller
Environment	Everything that is not the controller — In the preceding diagram, the environment includes the plant, the reference signal, and the calculation of the error. In general, the environment can also include additional elements, such as: • Measurement noise

Reinforcement Learning	**Control Systems**
	• Disturbance signals • Filters • Analog-to-digital and digital-to-analog converters
Observation	Any measurable value from the environment that is visible to the agent — In the preceding diagram, the controller can see the error signal from the environment. You can also create agents that observe, for example, the reference signal, measurement signal, and measurement signal rate of change.
Action	Manipulated variables or control actions
Reward	Function of the measurement, error signal, or some other performance metric — For example, you can implement reward functions that minimize the steady-state error while minimizing control effort. When control specifications such as cost and constraint functions are available, you can use generateRewardFunction to generate a reward function from an MPC object or model verification blocks. You can then use the generated reward function as a starting point for reward design, for example by changing the weights or penalty functions.
Learning Algorithm	Adaptation mechanism of an adaptive controller

III.

A) Dynamic model of LFC system

For the load-frequency control problem, the power system primarily experiences small variations in load, allowing it to be effectively represented by a linear model shown in Figure 3.

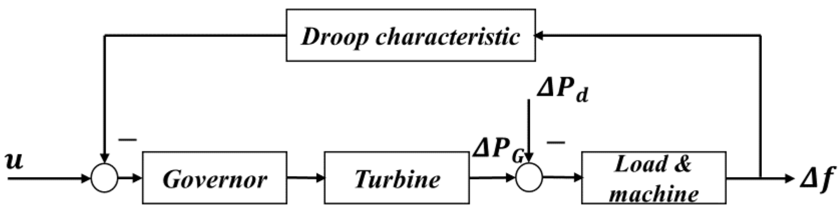

Figure (3) Linear model of a single-area power system.

To enhance the damping characteristics of the power system, the droop characteristic is utilized as a feedback gain, typically set to 1/R prior to load frequency control design. In this case, the plant model for LFC Design with droop characteristic is given by:

$$P(s) = \frac{G_g G_t G_p}{1 + G_g G_t G_p / R} \tag{1}$$

Here, G_g represents the dynamics of the governor (as described in equation 3), G_p represents the dynamics of the load and machine (as described in equation 4), and G_t corresponds to the dynamics of the turbine (equation 4 for non-reheated turbines and equation 5 for reheated turbines).
In this work, two different kinds of turbines are considered for the LFC design.

1) Non-reheated Turbine : In a power system with a non-reheated turbine, the plant can be divided into three main components:

- Governor with dynamics:

$$G_g(s) = \frac{1}{T_G\, s + 1} \tag{2}$$

- Turbine with dynamics:

$$G_t(s) = \frac{1}{T_T s + 1} \tag{3}$$

- Load and machine with dynamics:

$$G_p(s) = \frac{K_p}{T_P s + 1} \tag{4}$$

Now the open-loop transfer function for load frequency control is:

$$P(s) = G_g G_t G_p = \frac{K_p}{(T_P s + 1)(T_T s + 1)(T_G s + 1)} \tag{5}$$

2) Reheated Turbine: For reheated turbine, the dynamics of the turbine becomes:

$$G_t(s) = \frac{cT_r s + 1}{(T_r s + 1)(1 + T_T s)} \tag{6}$$

In this context, T_r is a fixed value, and c represents the proportion (percentage) of power generated by the reheat process in relation to the total generated power. And the open-loop transfer function for this turbine becomes:

$$P(s) = G_g G_t G_p = \frac{K_p(cT_r s + 1)}{(T_P s + 1)(T_T s + 1)(T_G s + 1)} \tag{7}$$

B) Model Parameters

1) Non-Reheated Turbine:

The model parameters for the power system that utilizes a non-reheated turbine are provided as follows:

$$K_p = 120,\ T_p = 20,\ T_T = 0.3,\ T_G = 0.08,\ R = 2.4$$

Consequently, the plant model is:

$$P(s) = \frac{120}{(20s + 1)(0.3s + 1)(0.08s + 1)} \tag{8}$$

2) Reheated Turbine:

The model parameters for a power system with a reheated turbine are given by:

$$K_p = 120,\ T_p = 20,\ T_T = 0.3,\ T_G = 0.08,\ R = 2.4,\ T_r = 4.2,\ c = 0.35$$

And The plant model is:

$$P(s) = \frac{120(1.47s + 1)}{(20s + 1)(0.3s + 1)(0.08s + 1)} \tag{9}$$

IV. Proposed RL-based LFC design

Reinforcement learning (RL) is a framework that doesn't require a model and is used to tackle optimal control problems. Many RL research studies utilize open-source tools like Python and OpenAI Gym environments. In our case, we make use of MATLAB's Reinforcement Learning Toolbox. We develop the Load Frequency Control (LFC) using this toolbox and train it using the Deep Deterministic Policy-Gradient (DDPG) algorithm. To simulate the LFC and evaluate its performance, we employ Simulink. This approach allows industrial engineers to easily adapt and experiment with other systems of their choice.

Equation (8) described the Non-Reheated turbine transfer function and we write one realization of this transfer function for simulation. In simulation the input of the system is a desired constant and the its output is the load frequency controller changes. The results of the turbine's simulation are shown in Figure 4.

The RL Agent is created using MATLAB's RL Toolbox. It takes three inputs: the observation vector, a reward function, and a boolean variable called "isdone".

- A constant input is considered zero.
- The action signal for this environment that is sent to the plant.
- For this environment, there are three observation signals sent to the agent, specified as a vector signal. The observation vector is $[\int edt \quad e \quad h]^T$, where:
  - Δf is the frequency deviation.
  - $e = r - \Delta f$, where r is the reference value for Δf. Showed in figure 5.
  -

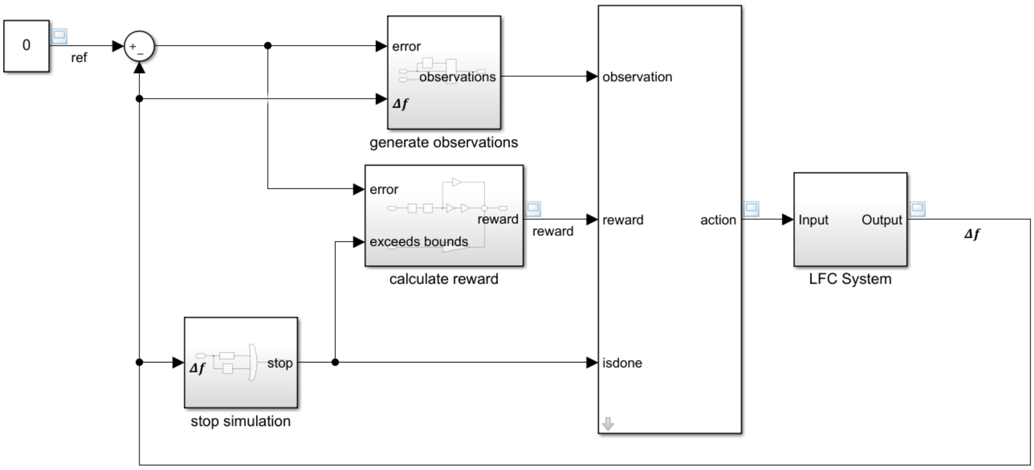

Figure (4)) SIMULINK Model for the proposed RL-based LFC.

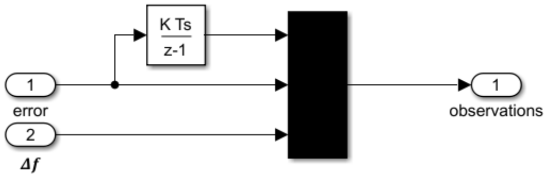

Figure (5) Signals of observation.

- The reward function is designed in such a way that if the frequency control load changes are less than or equal to 1, it will get a reward of 10, and if this value is greater, the reward value will be 1. Otherwise, 100 negative points are considered as a penalty as shown in Figure 6.

  Reward function:

  $10(|e| < 0.1) - 1(|e| \geq 0.1) - 100(\Delta f \leq -60 \,||\, \Delta f \geq 60)$

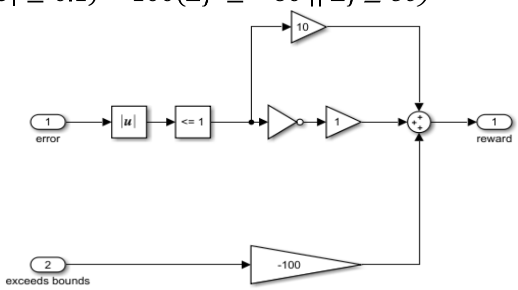

Figure (6) Reward function

- The isdone signal terminates the simulation if it goes out of the specified frequency range. In such cases, the agent does not receive a reward at the end of the episode as shown in Figure 7.
-

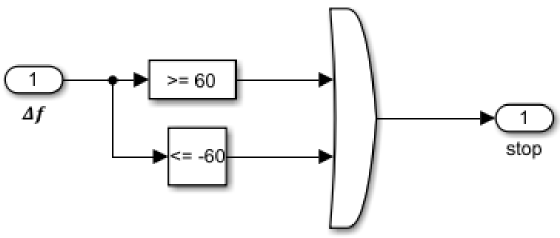

Figure (7) The isdone signal

- The output from the RL Agent contains the load frequency control changes.

The realization of Non-Reheated and Reheated turbines' transfer function are shown in Figures 8 and 9, respectively.

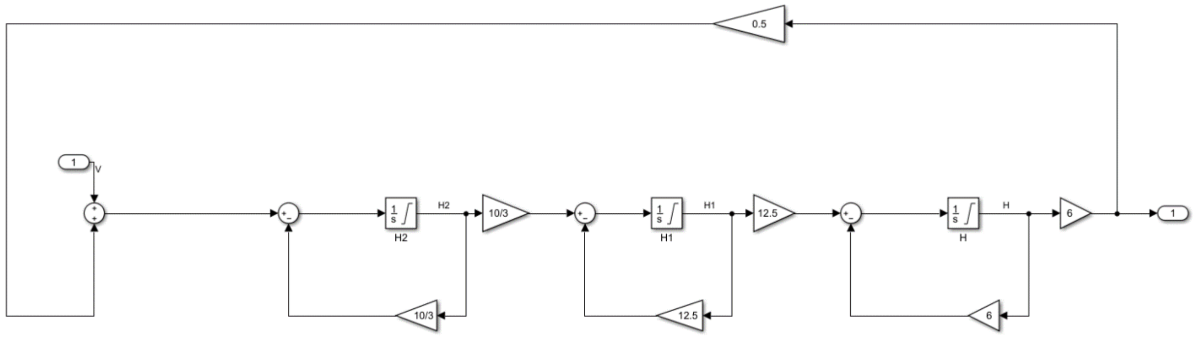

Figure (8) The realization of Non-Reheated turbine transfer function

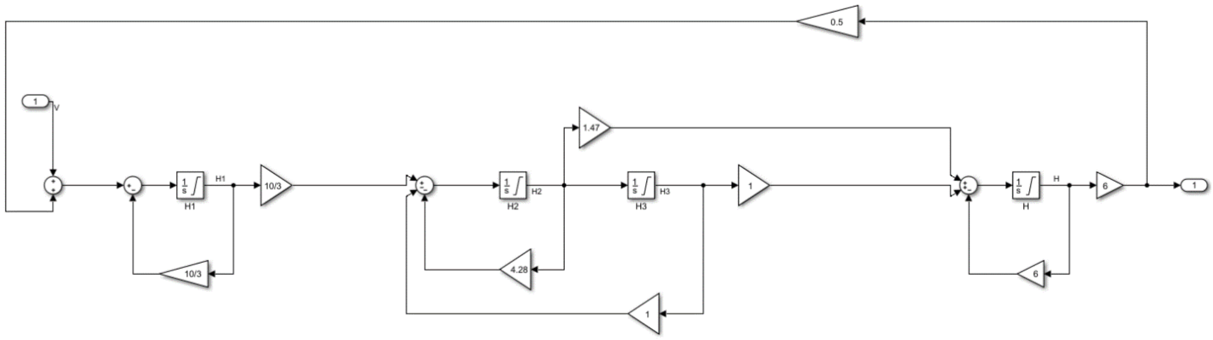

Figure (9) The realization of Reheated turbine transfer function.

DDPG is an algorithm that operates without a model and focuses on learning policies in complex environments with high-dimensional and continuous action spaces. It achieves this by utilizing deep function approximators.

The DDPG agent is comprised of two main components: the actor and the critic. Both the actor and the critic have two fully connected layers, with a hidden layer size of 500. Additionally, they incorporate a Rectified Linear Unit (ReLU) activation layer. Please note that in Figure 10.

Only the critic's architecture is displayed. By utilizing deep neural networks and the DDPG algorithm, the actor and critic work together to optimize the agent's decision-making process in continuous action spaces. This architecture allows the agent to effectively learn and approximate the optimal policy for the given task.

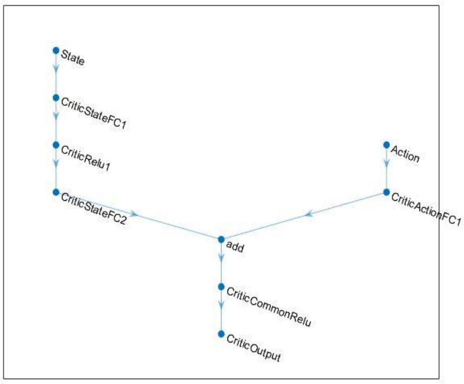

Figure (10) Critic Architecture.

IV. Results Discussion

Now we present the results of experiments conducted. We checked all permutations of the integrator blocks and choose the training that was faster and had better output. Every training curve was stopped at an Average Reward value of 1900 as at this value, the agent could successfully tune the controller. Figure 11 and 12 show the training for Non-Reheated and Reheated turbine.

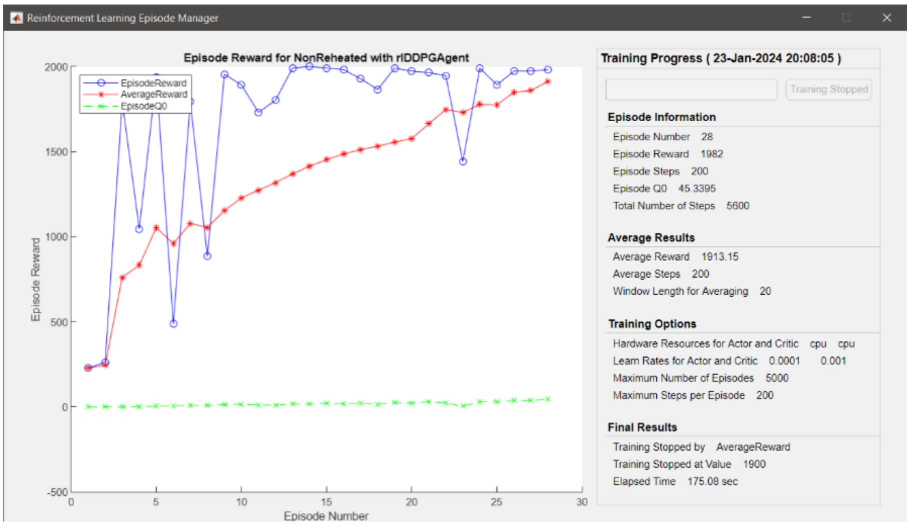

Figure (11) Training result for Non-Reheated turbine.

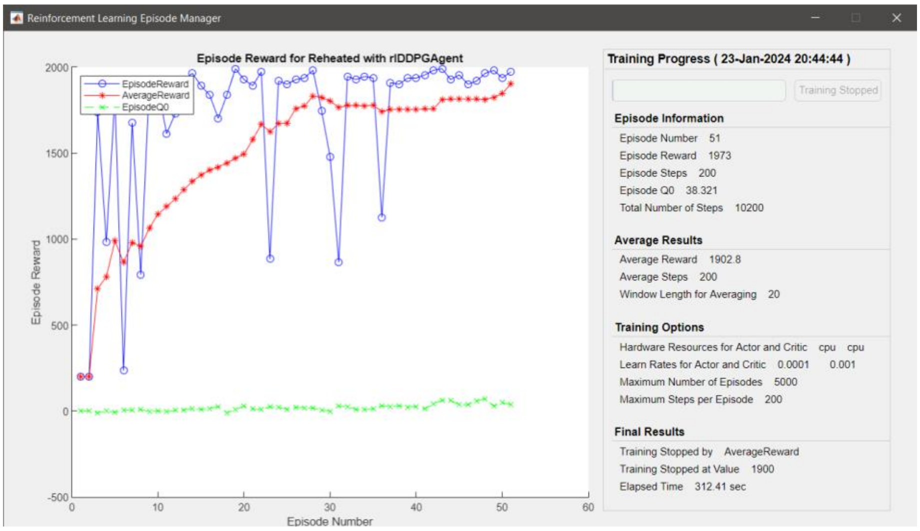

Figure (12) Training result for Reheated turbine.

For comparison, we employed the well-known PID controler for the LFC problem. The frequency deviation (Δf) and control input for the proposed RL-based load frequency controller and PID controller for a power system with non-reheated and reheated turbines are plotted in Figures 15 and 16, respectively. As shown, the time responses of the frequency deviaion and the corresponding control input are much better in our proposed LFC approach.

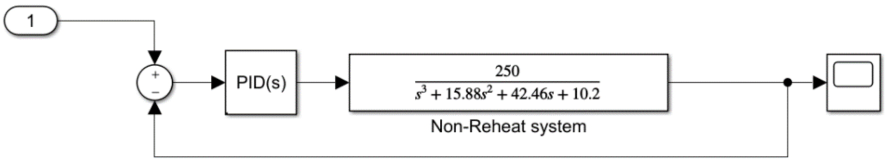

Figure (13) PID controller for Non-Reheated tubine.

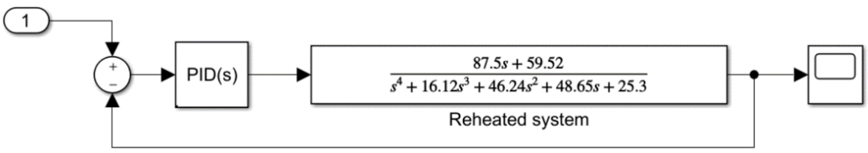

Figure (14) PID controller for Reheated tubine.

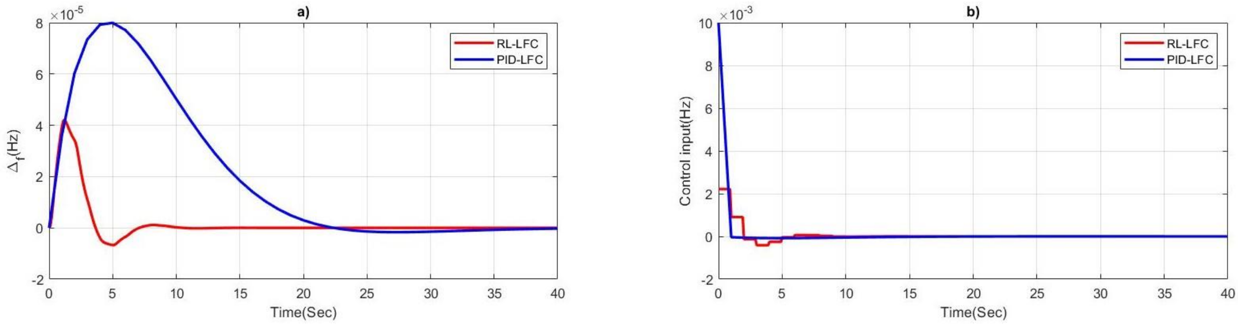

Figure (15) a) Responses of power systems with Non-Reheat turbine in two modes of reinforcement learning and PID controller and b) Control input of Non-Reheated turbine during training.

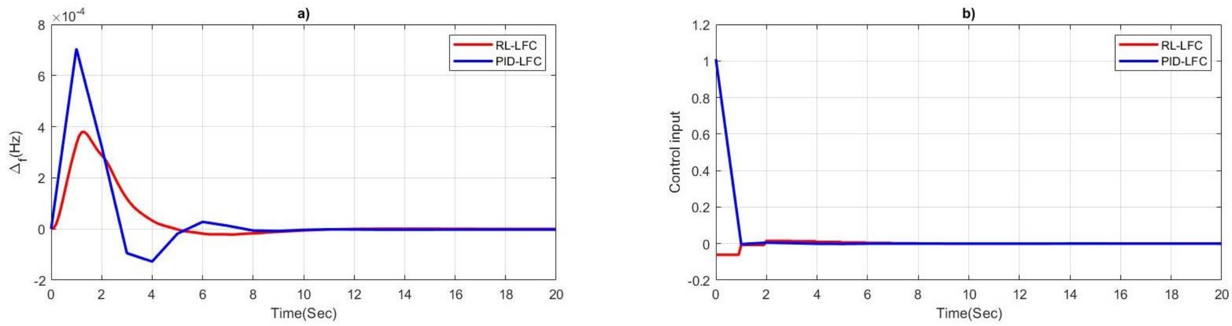

Figure (16) a) Responses of power systems with Reheat turbine in two modes of reinforcement learning and PID controller and b) Control input of Reheated turbine during training.

Conclusions

In order to address the significant Load Frequency Control (LFC) problem in power systems, we employed the MATLAB RL Toolbox. This toolbox operates on the basis of learning policies and algorithms, allowing us to effectively tackle the LFC challenge. The outcomes of our approach are presented in Figures 12 and 13, showcasing the results obtained through RL. It is evident that RL yields superior outputs compared to the PID controller.

The RL methodology offers a powerful and adaptive solution for LFC, enabling the agent to learn optimal control strategies through interactions with the environment. By leveraging RL techniques, we were able to improve the performance and efficiency of the LFC system, resulting in more accurate frequency and power balance. The visual representations in Figures 13 and 14 provide a clear demonstration of the advantages of RL over the conventional PID controller approach. This underscores the effectiveness and potential of RL in addressing complex problems in power systems like LFC.

Refrences

[1] M. Han, X. Zhang, L. Xu, R. May, S. Pan, and J. Wu, "A review of reinforcement learning methodologies on control systems for building energy," 2018.

[2] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004.

[3] R. B. Vinter and R. Vinter, *Optimal control* (no. 1). Springer, 2010.

[4] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2007.

[5] W. Gao, Y. Wang, and A. Homaifa, "Discrete-time variable structure control systems," *IEEE transactions on Industrial Electronics,* vol. 42, no. 2, pp. 117-122, 1995.

[6] J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control: A survey," *IEEE transactions on industrial electronics,* vol. 40, no. 1, pp. 2-22, 1993.

[7] K. J. Åström, "Theory and applications of adaptive control—a survey," *automatica,* vol. 19, no. 5, pp. 471-486, 1983.

[8] M. McClelland *et al.*, "Self-regulation," *Handbook of life course health development,* pp. 275-298, 2018.

[9] M. Inzlicht, K. M. Werner, J. L. Briskin, and B. W. Roberts, "Integrating models of self-regulation," *Annual review of psychology,* vol. 72, pp. 319-345, 2021.

[10] Z. Cai, *Intelligent control: principles, techniques and applications*. World Scientific, 1997.

[11] P. J. Antsaklis, "Intelligent control," *Encyclopedia of Electrical and Electronics Engineering,* vol. 10, pp. 493-503, 1997.

[12] K. J. Åström and T. J. McAvoy, "Intelligent control," *Journal of Process control,* vol. 2, no. 3, pp. 115-127, 1992.