

Research Article

Reinforcement Learning Based Neuro-fuzzy Controller for Coffee Roasting Process

Abiy Amare^{1,*} , **Solomon Seid²** ¹Electromechanical Engineering Department, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia²College of Engineering, Ethiopian Defence University, Bishoftu, Ethiopia

Abstract

Supervised learning is mainly used to optimize Adaptive Neural Fuzzy Inference System (ANFIS) controllers. In order to generate data for supervised learning, a controller is designed and optimized using Particle Swarm Optimization (PSO) or any other algorithms. This paper proposes and compares reinforcement learning based ANFIS and Approximate Reasoning Intelligent controller (ARIC) controllers. Reinforcement learning based ANFIS reduces the work flow required to train it by directly optimizing the membership functions using Proximal Policy Optimization (PPO) algorithm. ANFIS and ARIC neuro fuzzy controllers are designed for nonlinear dynamics of coffee roasting process using Schwartzberg's model. A custom layer is designed for every membership function and fuzzy inference operations using MATLAB's Deep Learning Toolbox. This neural connectionist model of ANFIS and ARIC is used as actor. The critic which evaluates the goodness of action taken is a two-layer neural network with sigmoidal activation function. Simulink environment is also created to represent the dynamics of coffee roasting process. The agent is trained to track roast profile for 50 episodes. The training converged at 50th iteration. After training, the Root Mean Square Error (RMSE) for ARIC architecture reduced from 0.5134 to 0.08122. Similarly, the RMSE of ANFIS improved from 0.2026 to 0.0624.

Keywords

Reinforcement Learning, ANFIS, ARIC, PPO, Schwartzberg Model, Spouted Bed Roasting

1. Introduction

Neural network and fuzzy system are combined due to the need to tune fuzzy systems to obtain optimal performance. According to Ajiht [1], one drawback of neural network is its incapability to accommodate prior knowledge in the learning process. In addition, the black box nature of neural networks is fundamental drawback for control system applications. On the contrary fuzzy systems have capability to accommodate prior knowledge in the form of if-then rules and input-output membership functions. To incorporate prior knowledge in to

Fuzzy Inference System (FIS) it is mandatory to have know-how on how inputs affect output. This know-how can be obtained from an expert, here in this case the control engineer is responsible to generate if-then rules using knowledge from control engineering. By combining neural network and fuzzy systems it is possible to gain advantages of the two paradigms.

Ivan Petrovic [2] designed a reinforcement learning based Generalized Approximate Reasoning Intelligent Controller

*Corresponding author: abiy.amare251@gmail.com (Abiy Amare)



(GARIC) for inverted pendulum on a cart system. Ivan utilized weak reinforcement signals, as described in Equation (1), but these weak reinforcement signal yield unsatisfactory result.

$$r = \begin{cases} -1 & \text{failure state} \\ 0 & \text{any other state} \end{cases} \quad (1)$$

The author also experimented with more informative reinforcement signals as shown Equation (2). This elaborate signal enabled the system to reduce error and controller effort.

$$r = \begin{cases} -1 & \text{failure state} \\ -\frac{|x|}{x_{msx}} & \text{any other state} \end{cases} \quad (2)$$

Miaolei Zhou proposed GARIC neural fuzzy controller for Magnetic Shape Memory Alloy Actuator to control displacement [3]. According to S. Rößler Magnetic Shape Memory (MSM) is a phenomenon where shape or size of magnetic material changes when magnetic field changes [4]. The author used gaussian membership functions in fuzzification layer, and a lookup table to represent crisp defuzzied outputs. Mohammad Hossein Fazel Zarandi and Javid Jouzdani [5] proposed a GARIC architecture in which Action Evaluation Network (AEN) has similar structure with Action Selection Network (ASN) but only consequent labels of both AEN are updated during learning.

HA DUC NGUYEN proposed reinforcement learning based Adaptive Neural Fuzzy Inference System (ANFIS) controller for Maximum Power Point Tracking (MPPT) control of variable speed wind turbine. The author implemented ANFIS structure for actor and critic representation in reinforcement learning algorithm [6]. Mohamed Elsis proposed ANFIS controller for Blade Pitch Controller for Wind Energy Conversion Systems Against Wind Speed Fluctuations. The author utilized supervised learning method to optimize ANFIS architecture. The data set needed to train ANFIS is obtained using Mayfly Optimization Algorithm (MOA) optimized Proportional Integral Derivative (PID) controller. Machrus Ali [7] has also presented such controller for photovoltaic axis tracking. Genetic Algorithm, Particle Swarm Optimization and Ant Colony optimization are also used to optimize ANFIS controllers [7-10]. Proximal Policy Optimization, (PPO) reinforcement learning algorithm is used by Ming Chen to control classical cartpole system. The author specifically proposed adaptable learning policy rate PPO and compared to a fixed learning rate PPO [11].

Wei He used reinforcement learning based controller to control flexible two link robot. Both the actor and critic are represented as neural networks [12].

Jonas Degraeve proposed a neural network controller optimized by using maximum a posteriori policy optimization (MPO) for Magnetic control of tokamak plasmas. The author stated that the control objective is to shape and maintain high temperature plasma in tokamak vessel. Tokamak is a machine

used to produce nuclear fusions. According to Jonas in order to maintain stable plasma position in is mandatory to use feedback control. Jonas Degraeve trained a neural network on a simulated environment that sufficiently represents the dynamics of tokamak machine. To represent the actor the author used four-layer feedforward network. The size of actor neural network is limited to computational capacity of hardware on which the policy will be implemented, but there is no restriction on critic network. The author did not perform real time training of the neural network and directly implemented it on 10kHz control hardware [13].

2. Coffee Roasting Process Model

In spouted bed roasters hot air is used to heat and agitate coffee beans in a roasting chamber. Spouted bed roaster machines have three basic components; a blower, a heater, and a roasting chamber. The blower is used blow air at higher pressure and velocity so that coffee beans are agitated by the upward pressure. Air at higher velocity passes through a heating element picking up heat energy from resistive heating elements. The roasting chamber is where green coffee beans are placed.

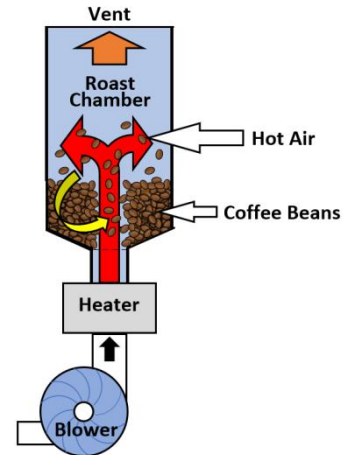


Figure 1. Spouted bed coffee roaster [14].

2.1. Heater Dynamics

Heating system is a mechanism by which temperature of air is raised to value capable of initiating roasting process. Resistive heating elements are considered to be 100% efficient at converting electrical energy to thermal energy.

$$\frac{d\Delta T}{dt} = \frac{VI}{m_m c_{pm}} - \frac{\dot{m} g c_{pg} \Delta T}{m_m c_{pm}} \quad (3)$$

2.2. Schwartzberg Model

Bean Temperature

$$\frac{dTb}{dt} = \frac{GCpg[Tgi - Tgo] + mbs * \left(Qr + Lv \frac{dX}{dt} \right)}{mbs(1+X)Cpb} \quad (4)$$

$$Tgi - Tgo = (Tgi - Tb) \left(1 - \exp \left(\frac{-\alpha * Ab}{GCpg} \right) \right) \quad (5)$$

Moisture content

$$\frac{-dX}{dt} = \frac{4.32 * 10^9 * X^2}{db^2} * \exp \left[\frac{-9889}{Tb + 273.15} \right] \quad (6)$$

Rate of exothermic heat generation

$$\frac{dq}{dt} = Qr = A \exp \left(\frac{-Hg}{RgTb(K)} \right) \left(\frac{Het - He}{Het} \right) \quad (7)$$

3. Methodology

3.1. Fuzzy Controller Design

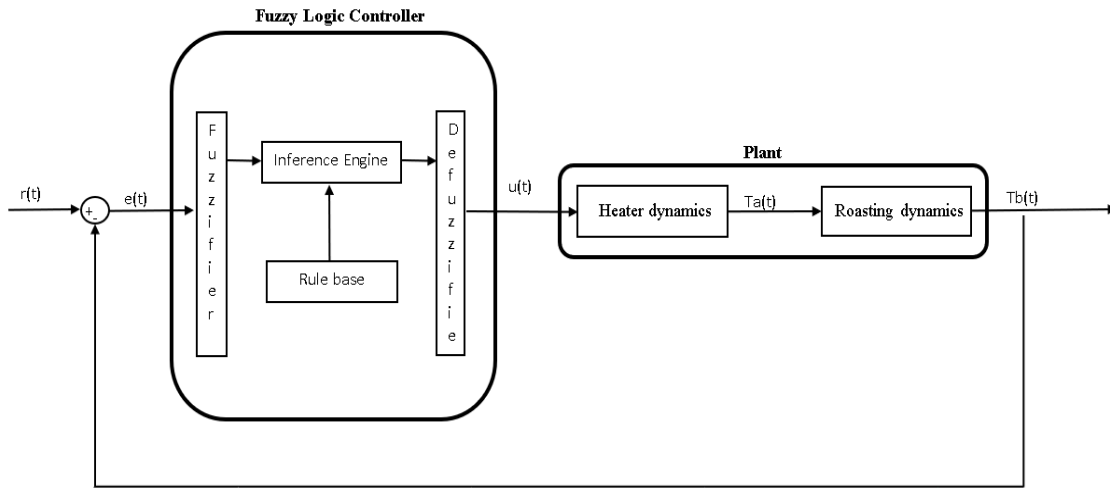


Figure 2. Fuzzy controller structure.

Fuzzy logic provides mathematical tools to represent human knowledge in the form of if-then rules for some set of input and output linguistic variables. Architecture of fuzzy logic controller is shown Figure 2 and it has four elements. The fuzzifier receives crisp input value; in control system these values may be states of the plant, error between reference signal and actual plant states, error derivatives or error integrals, and these crisp values are converted to linguistic values in the fuzzifier. The fuzzy rule base is a way to represent the control policy that will mimic the actions of an expert. The inference engine performs appropriate fuzzy set operations to decide which rules to fire at the current state of the dynamics. The defuzzifier provides a crisp output or crisp control signal using fuzzy values from the inference engine [15].

Fuzzy set

A fuzzy set is an ordered pairs of elements from universe of discourse and its membership value. Membership functions have values from 0 to 1 and they quantify the degree an element in universe of discourse belong to a fuzzy set. For control applications most widely used membership functions are triangular and trapezoidal due to their computational efficiency.

Triangular Membership functions

Three parameters characterize a triangular membership function. These parameters are the three corners on the graph of triangular membership functions.

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x \geq c \end{cases} \quad (8)$$

Trapezoidal membership functions

Trapezoidal membership function is characterized by four parameters,

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \geq d \end{cases} \quad (9)$$

Fuzzy operations

For two fuzzy sets A and B the following fuzzy operations

are defined:

Equality: two fuzzy sets A and B are equal if and only if the membership values $\mu_A(x)$ is equal to $\mu_B(x)$ for all x in the universe of discourse U .

$$\mu_A(x) = \mu_B(x) \text{ for all } x \text{ in } U \quad (10)$$

Union: The union of two fuzzy set is given by

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \text{ for all } x \text{ in } U \quad (11)$$

Intersection: The intersection of two fuzzy sets corresponds to the minimum of the membership values of the two sets for all x in the universe of discourse U .

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \text{ for all } x \text{ in } U \quad (12)$$

Complement: corresponds to the Boolean NOT function and is given by

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \text{ for all } x \text{ in } U \quad (13)$$

On the book Methodology of Fuzzy control (page 41), it is discussed that algebraic product and algebraic sum as in Equation (14) and Equation (15), are possible ways to represent linguistic AND and OR operations. The problem with $\min(x)$ and $\max(x)$ methods is its non-differentiability, but algebraic product and algebraic sum are continuous and differentiable at all points.

$$\mu_{A \cap B}(x) = A * B \quad (14)$$

$$\mu_{A \cup B}(x) = A + B - A * B \quad (15)$$

Inputs and outputs

The first input to FLC block is $e(t)$ which is the difference between a desired roast profile and actual bean temperature. Universe of discourse of input ranges from -20K to 300K. The second input is $\dot{e}(t)$ which is derivative of error.

$$e(t) = r(t) - Tb(t) \quad (16)$$

$$\dot{e}(t) = \frac{de(t)}{dt} \quad (17)$$

Based on the structure of the system to be controlled, the variable that should be manipulated to properly track a desired roast profile is input voltage supplied to heating element (t).

Fuzzification

Fuzzification is a process of mapping crisp inputs to fuzzy set's membership values. The error input has universe of discourse from -20K to 300K. This linguistic values of $e(t)$ are:

1. Negative big error: it is trapezoidal membership function with universe of discourse from -1K to -20K.
2. Negative small error: it is triangular membership function with universe of discourse from -1K to 0.1K.
3. Positive small error: it is triangular membership function with universe of discourse from -0.1K to 1K.
4. Positive big error: it is trapezoidal membership function with universe of discourse from 1K to 300K.

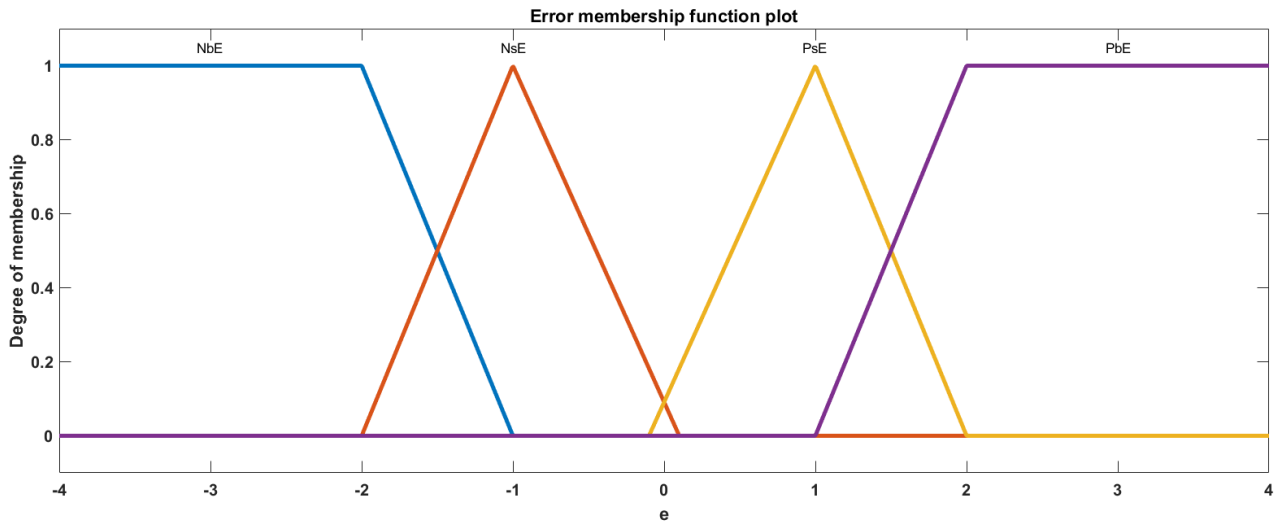


Figure 3. Membership functions of input error (Display range is from -4 to 4).

The error derivative $\dot{e}(t)$ input has three linguistic variables:

1. Negative error derivative: is triangular membership function from -0.01 to 0
2. Zero error derivative: is triangular membership function from -0.002 to 0.002
3. Positive error derivative: is triangular membership function from 0 to 0.01

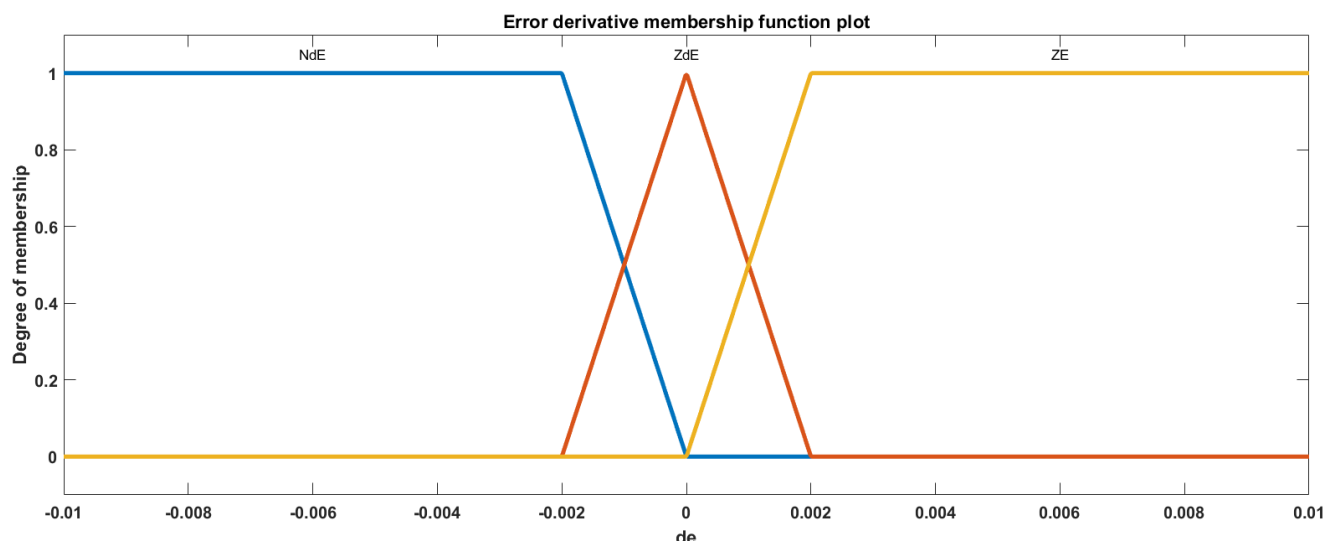


Figure 4. Membership functions of error derivative.

The rule base

Table 1. The rule base for Mamdani type fuzzy logic controller.

de	e			
	NbE	NsE	PsE	PbE
NdE	LV	LV	MLV	HV
ZdE	LV	MLV	MHV	HV
PdE	LV	MHV	HV	HV

error are defined as NbE (Negative big error), NsE (Negative small error), PsE (Positive small error), PbE (Positive big error). The membership functions for error derivatives are NdE (Negative error derivative), ZdE (Zero error derivative), PdE (Positive error derivative). The membership functions for consequent section are LV (Low Volt), MLV (Medium Low Volt), MHV (Medium High Volt), and HV (High Volt).

Fuzzy inference

The output variable is voltage supplied to the heating element. The range for this variable is from 0V to 220V, since the universe of discourse should cover all possible output points the range is extended from -20V to 240V on the fuzzy system. But in the Simulink model a saturation block is used to limit voltage output from 0V to 220V.

The rule base has eight rules. The membership functions for

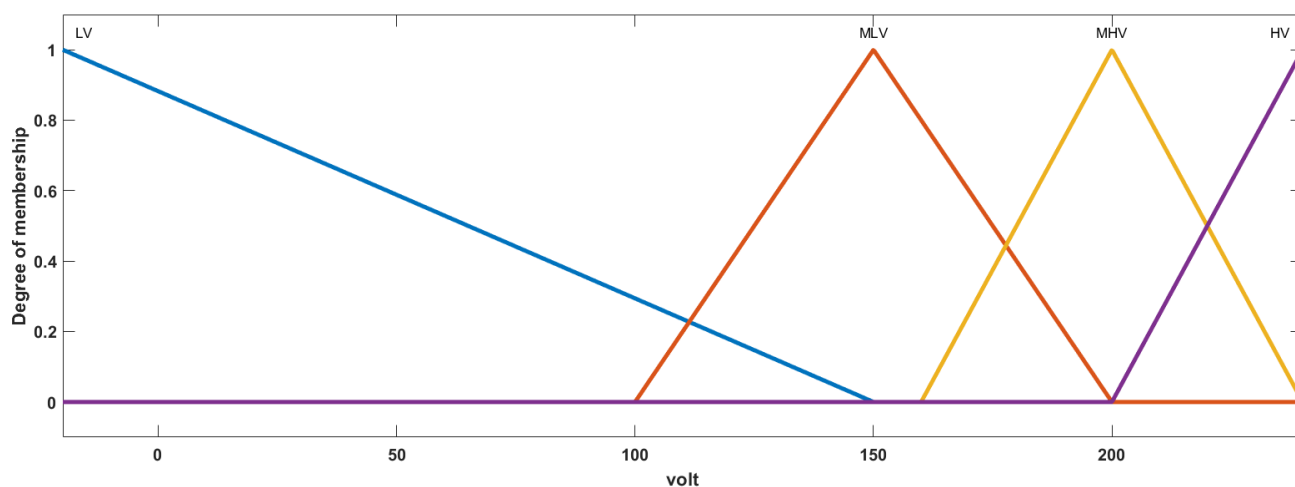


Figure 5. Membership function of consequent.

The membership functions which are triangular in shape are:

1. Low voltage is triangular membership function with universe of discourse from -20V to 150V.
2. Medium low voltage is triangular membership function with universe of discourse from 100V to 200V.
3. Medium high voltage is triangular membership function

with universe of discourse from 160V to 240V.

4. High voltage is triangular membership function with universe of discourse from 200V to 240V.

Aggregation

In Mamdani type controllers the firing strength of every rule is combined before defuzzification. In this paper *sum()* aggregation method is used.

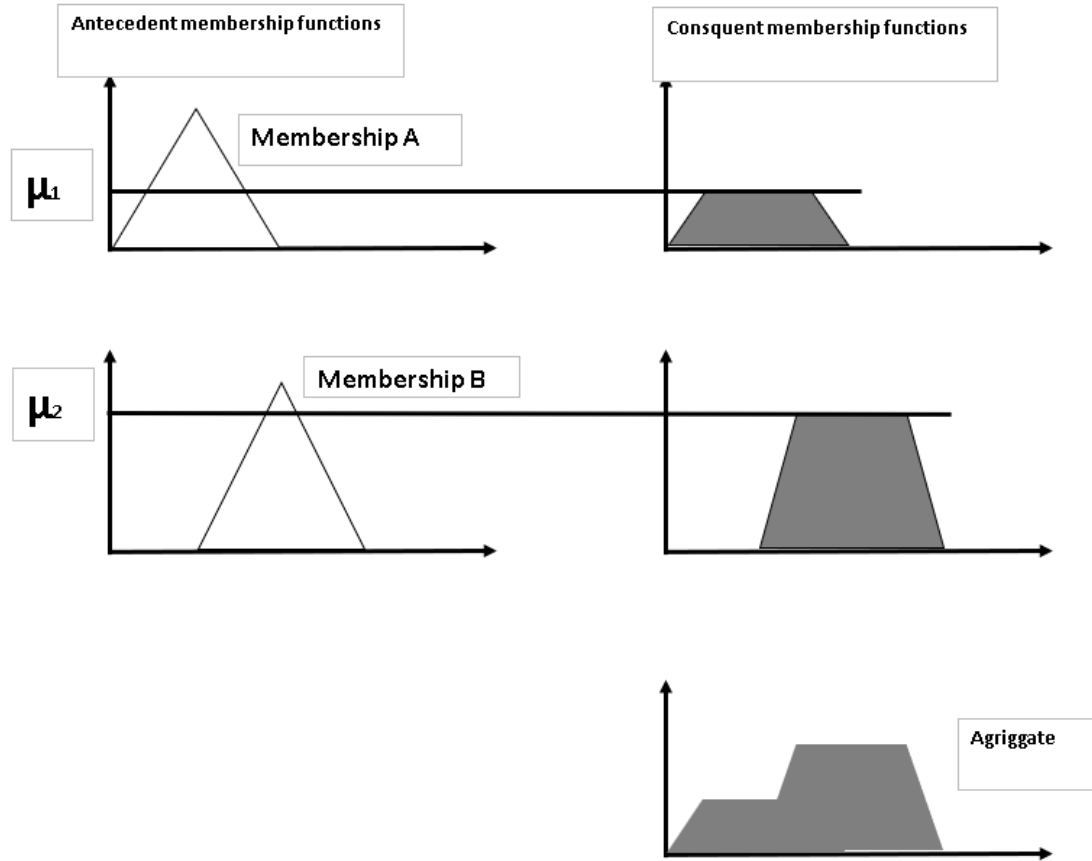


Figure 6. Aggregation in fuzzy inference system.

Defuzzification

Defuzzification process provides a crisp fuzzy control signal. Here center of area method is used and it is expressed as

$$u = \frac{\text{sum first moment of area}}{\text{sum of area}} \quad (18)$$

For continuous system

$$u = \frac{\int u\mu(u)du}{\int \mu(u)du} \quad (19)$$

For discrete systems

$$u = \frac{\sum_{i=1}^n u_i \mu(u_i)}{\sum_{i=1}^n \mu(u_i)} \quad (20)$$

3.2. ARIC Controller Design

Approximate Reasoning based Intelligent control (ARIC) is a neural fuzzy architecture used to tune FIS using reinforcement learning. It has three components, action selection network (ASN), action evaluation network (AEN), and stochastic action modifier (SAM) [16]. The ASN is a fuzzy controller represented in neural network connections model. It receives input from the plant and outputs appropriate control action. The critic network (AEN), according to Brenji, is a two layer network that maps reward signal and current plant state to a scalar number that measures how good or bad the action is. The stochastic action modifier uses signals from ASN and AEN to modify the actions taken from the plant in such a way that if action at previous time step is good, the action at current time step does not deviate much

from the previous action [16, 17]. Brenji also proposed an architecture called Approximate Reasoning Intelligent Controller (ARIC) which does not have stochastic action modifier as shown in Figure 7.

On a review of neural fuzzy systems Nkumbah P. J [18], stated that GARIC architecture utilizes gradient descent learning method to update AEN and reinforcement learning method to update ASN.

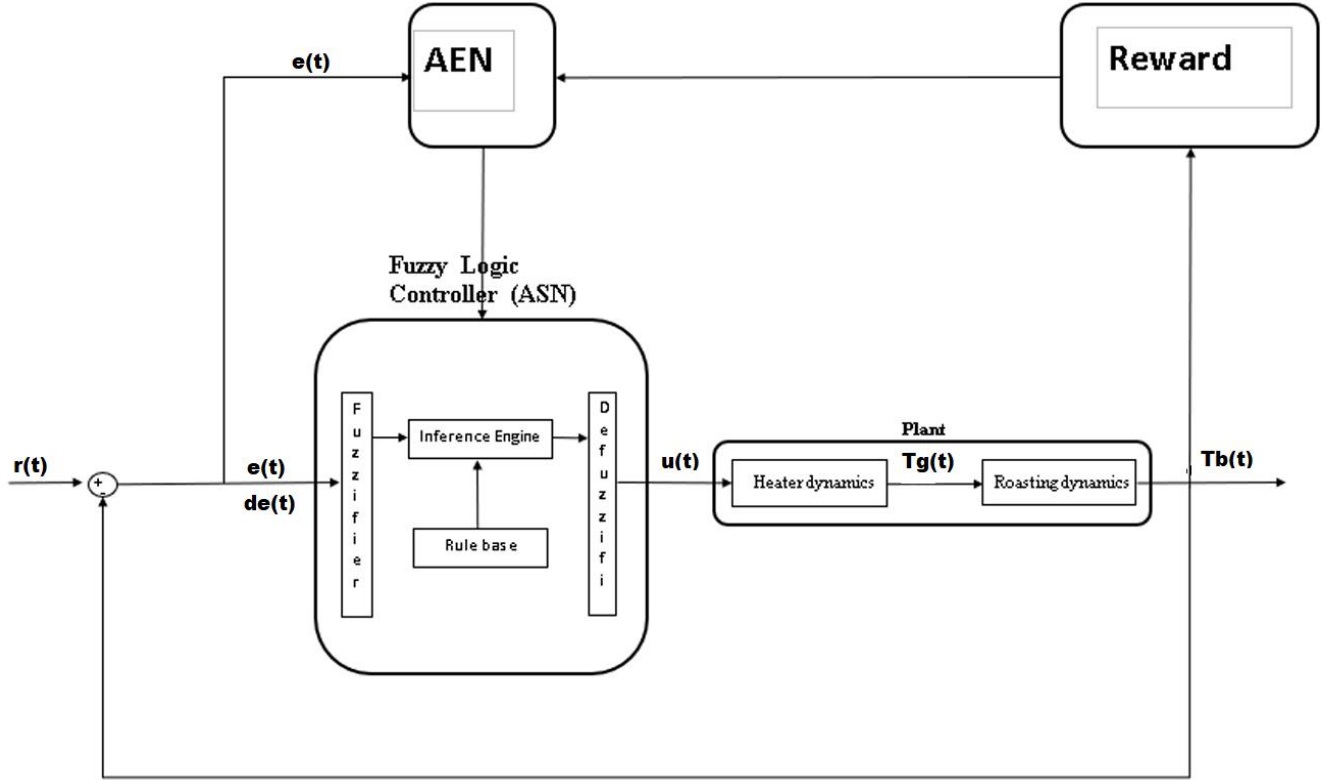


Figure 7. ARIC neuro fuzzy architecture.

Action selection network

The action selection is the neural representation of Mamdani type fuzzy system with slight modification. This network has five layers which represent each step in the fuzzy inference system.

Layer one: This is input layer. In this research error and error derivatives are used as inputs. Each input is represented by a node on the neural connectionist model, hence node 'error' is for error input and node 'error derivative' is for error derivative input. No computation is performed and there is no learnable parameter in this layer.

Layer two: This layer represents fuzzification process in FIS system. There are seven nodes that represent linguistic variables used in fuzzification. Each node outputs firing strength μ_i of the membership function.

Layer three: This layer is equivalent to the rule base that performs min() operation on FIS. To overcome non differentiability of min() function Hamid R. Berenji proposed a soft min operator as described in Equation (21). As the constant in the equation is increased in magnitude the equation yields

equivalent value as the min operator. As can be seen on the Figure 8 each node corresponds to a rule in the rule base [16, 19].

$$w_r = \frac{\sum_i \mu_i e^{-k\mu_i}}{\sum_i e^{-k\mu_i}} \quad (21)$$

Layer four: Layer four is the consequent layer which evaluates the output action that a node would contribute. Layer four represents the consequent part of fuzzy inference system. According to Berenji the output of layer four is the x-coordinate of the centroid of the membership function. For triangular membership functions,

$$\mu_i^{-1}(w_r) = b + 0.5(a - 2b + c)(1 - w_r) \quad (22)$$

Layer five: Layer five is defuzzification layer. Crisp defuzzified output is calculated as follows.

$$u = \frac{\sum w_r \mu_i^{-1}(w_r)}{\sum w_r} \quad (23)$$

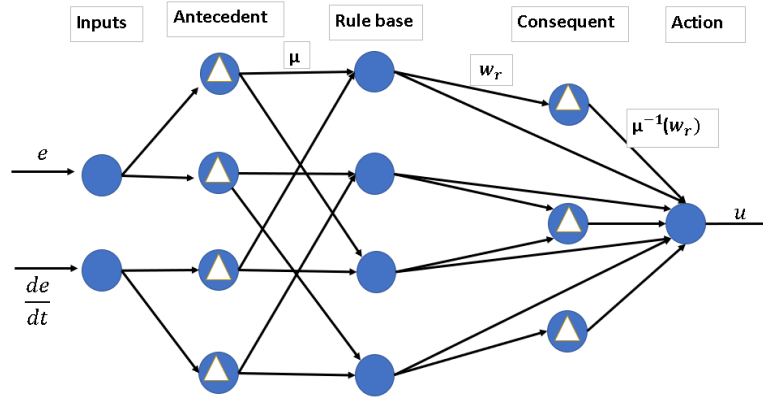


Figure 8. ASN of ARIC neuro fuzzy architecture.

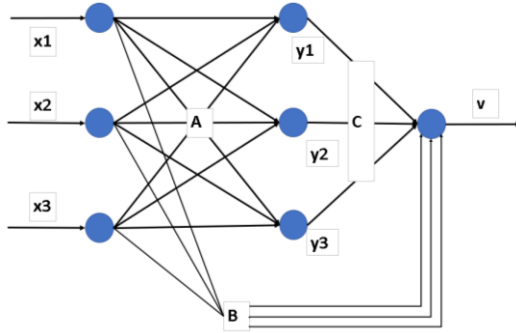


Figure 9. AEN of ARIC neuro fuzzy architecture.

Action evaluation network

This network predicts the reward that would be collected

based on the current state of the environment. AEN has ‘error’ and ‘error derivative’ as its inputs. These inputs are passed to a sigmoidal layer with five nodes. The predicted output reward is given by the Equation (26).

$$y_i = g(\sum a_i x_i) \quad (24)$$

$$g(s) = \frac{1}{1 + e^{-s}} \quad (25)$$

$$v = \sum b_i x_i + \sum c_i y_i \quad (26)$$

Where,

b_i is wight of connection between output layer v and input layer.

y_i is output from sigmoidal layer.

c_i is weight of connection between output and sigmoid layer.

x_i is input layer.

3.3. ANFIS Controller Design

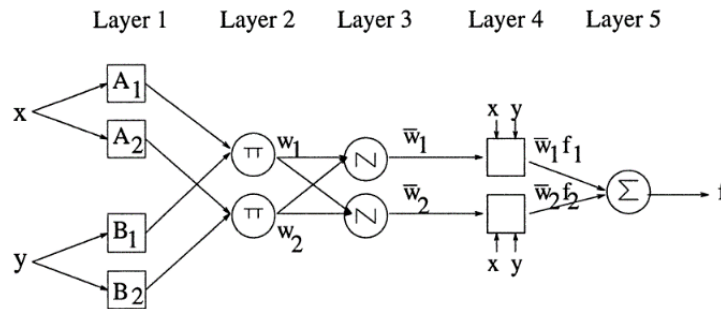


Figure 10. ANFIS controller architecture [8].

Layer one: This layer performs fuzzification on its inputs and identical to layer one in ARIC.

Layer two: This layer is product minimum operator given by Equation (27).

$$w_1 = \mu(A)_i \mu(A)_i \quad (27)$$

Layer three: This layer is normalization layer where its output is given by

$$\bar{w}_1 = \frac{w_1}{w_1 + w_2} \quad (28)$$

Layer four: this layer is the consequent section where its

membership variables are represented by linear equation. The coefficients of error, error derivative, and the constant are all learnable parameters.

$$LV = e + \dot{e} + 27.47 \quad (29)$$

$$MLV = e + \dot{e} + 150 \quad (30)$$

$$MHV = e + \dot{e} + 200 \quad (31)$$

$$HV = e + \dot{e} + 220 \quad (32)$$

3.4. Neural Realization of Fuzzy Inference System

A straight forward method to represent a fuzzy system as a neural network is to make each neuron behave like a fuzzy operator. For example, the membership functions on antecedent and consequent parts of FIS can be represented by an activation function of a neuron. According to Lin [15] $\min()$ activation function can be used to represent AND operation in addition $\max()$ activation function can be used to represent OR operation.

In MATLAB 2021a it is possible to design a custom layer with learnable parameters and activation function. A custom layer is designed for each membership function in antecedent and consequent sections. For triangular membership functions the centers are kept constant and only the bases are learnable parameters.

Figure 11 shows implementation of action evaluation network of ARIC architecture. It has input layer with error

and error derivative, five neurons with sigmoidal activation function and an output layer.

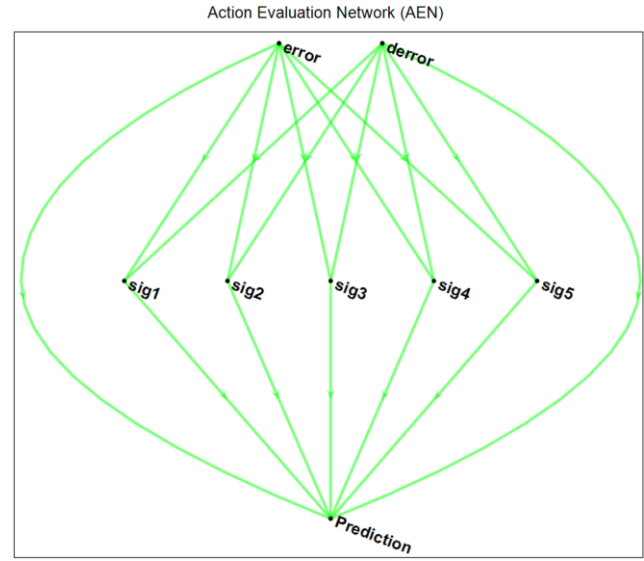


Figure 11. AEN implementation using MATLAB.

Figure 12 shows MATLAB implementation of action evaluation network of ARIC architecture.

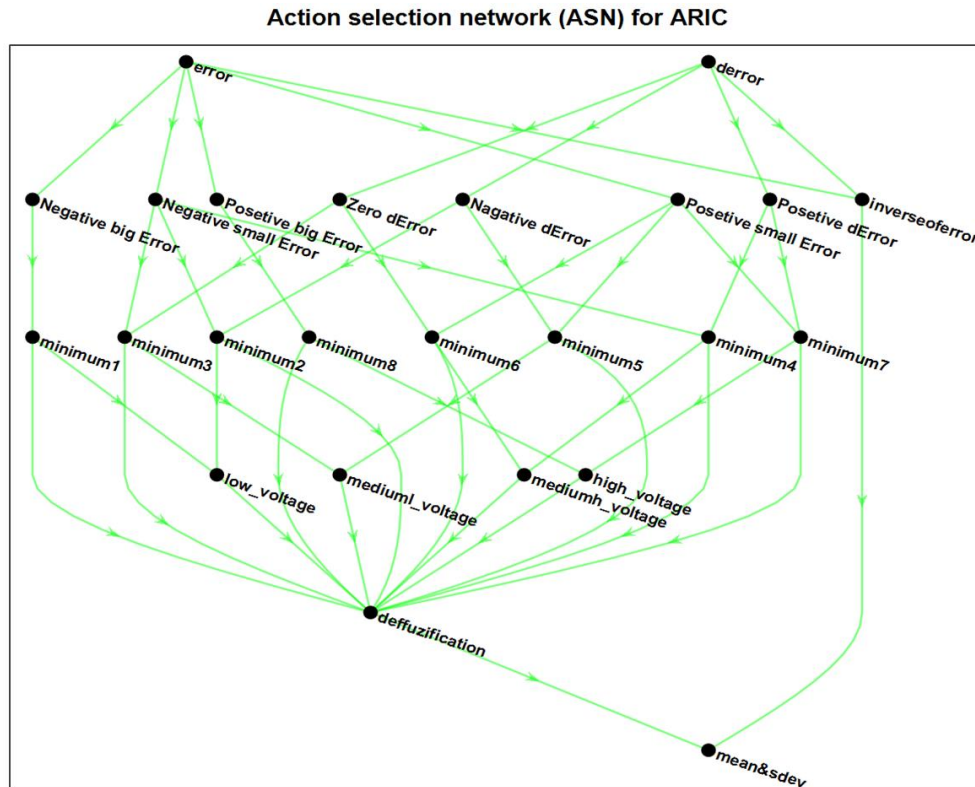


Figure 12. ASN implementation using MATLAB.

ARIC neural fuzzy representation the soft $\min()$ operator discussed in Equation (21) is used as $\min()$ for the second layer. SUGEUNO type neural fuzzy controllers use product to realize $\min()$ operation on the third layer.

Table 2. Learnable parameters of ARIC neural fuzzy architecture.

Membership Function	Before Optimization	Learnable Parameters
Negative big error	[-20 -20 -2 -1]	[-20 -20 c d]
Negative small error	[-2 -1 0.1]	[a b 0.1]
Positive small Error	[-0.1 1 2]	[-0.1 b c]
Positive big Error	[1 2 300 300]	[a b 300 300]
Negative error derivative	[-0.01 -0.01 -0.002 0]	None
Zero error derivative	[-0.002 0 0.002]	None
Positive error derivative	[0 0.02 0.01 0.01]	None
Low Voltage	[-20 -20 125]	[b b c]
Medium Low Voltage	[100 150 200]	[a 150 b]
Medium High Voltage	[160 200 240]	[a 200 b]
High Voltage	[200 240 240]	[a b b]

Figure 13 shows MATLAB implementation of ANFIS architecture.

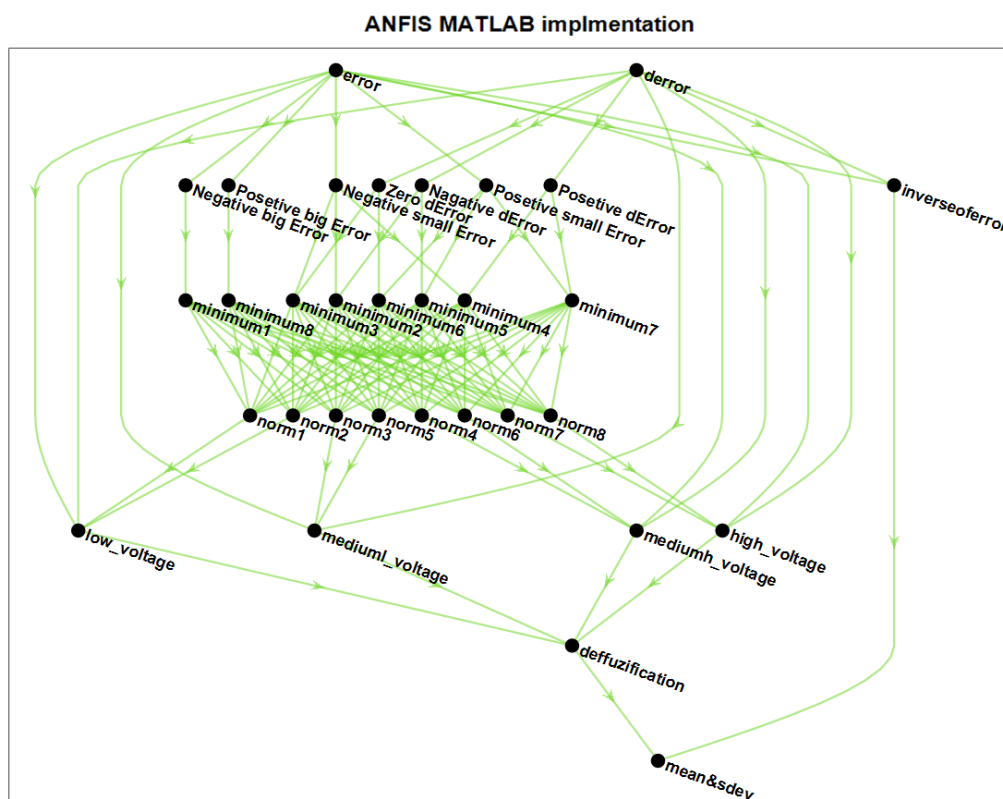


Figure 13. MATLAB implementation of ANFIS.

Table 2 and Table 3 show lists of membership functions with corresponding learnable parameters.

Table 3. Learnable parameters of ANFIS neural fuzzy architecture.

Membership Function	Before Optimization	Learnable Parameters
Negative big error	[-20 -20 -2 -1]	[-20 -20 c d]
Negative small error	[-2 -1 0.1]	[a b 0.1]
Positive small Error	[-0.1 1 2]	[-0.1 b c]
Positive big Error	[1 2 300 300]	[a b 300 300]
Negative error derivative	[-0.01 -0.01 -0.002 0]	None
Zero error derivative	[-0.002 0 0.002]	None
Positive error derivative	[0 0.02 0.01 0.01]	None
Low Voltage	[1 1 22.75]	[a b c]
Medium Low Voltage	[1 1 150]	[a b c]
Medium High Voltage	[1 1 200]	[a b c]
High Voltage	[1 1 227.5]	[a b c]

3.5. Reinforcement Learning

Reinforcement learning is a learning mechanism by which an agent learns to behave properly based on the reward signal from a critic. Fuzzy controllers of ANFIS and ARIC architectures are designed and represented as a neural network. The goal is to train this network to obtain better performance from our controllers. The plant that is being controlled is the environment. Environment is everything outside of the agent.

Reward

Reward is a function that judges the effect of an action taken by the policy in RL agent. Reward equations used for the system is stated in Equation (33).

$$r = \begin{cases} -0.01|u|, & \text{if } e \neq 0 \\ 0.01, & \text{if } e = 0 \end{cases} \quad (33)$$

Proximal Policy Reinforcement Learning Agent

Proximal policy optimization (PPO) is a model-free, online, on-policy, policy gradient reinforcement learning method. Model free means the agent does not learn about the environment, it only learns to find a policy that maximize the reward. Online means new experiences are generated by interacting with the environment for each episode. On-policy means experiences are generated using the latest policy. PPO algorithm uses a clipping function so that newly generated policy does not move very far away from the previous policy. This method improves stability of learning process.

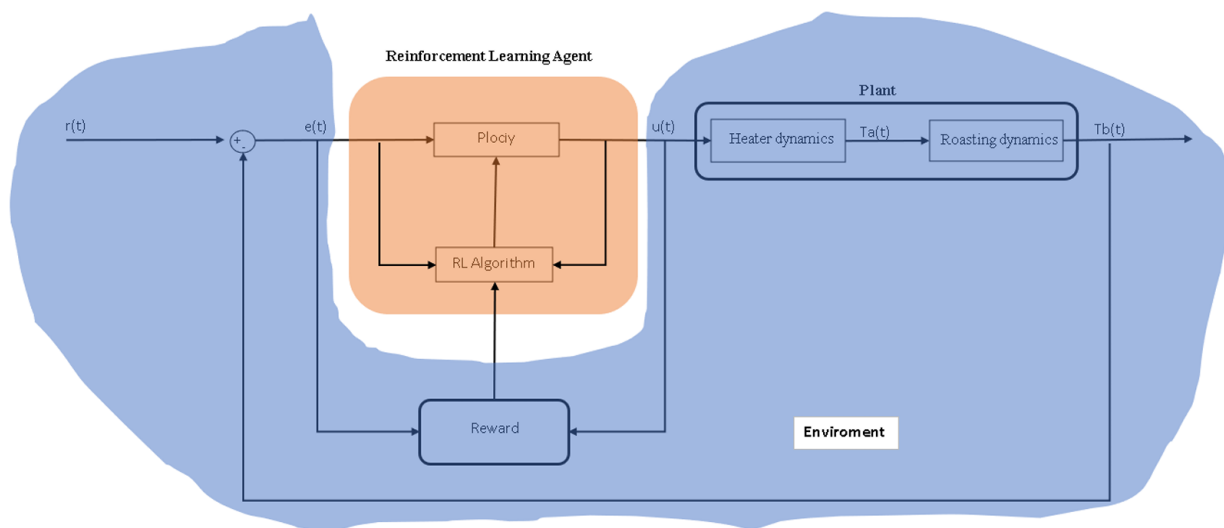
**Figure 14.** Environment (blue) and Agent (orange).

Figure 14 shows how the agent is incorporated in the control loop. Everything that is not the agent is considered environment. The agent receives reward and current states from the environment, and then the agent generates an action based on the current state.

4. Results and Discussions

Open loop response of roasting process is plotted Figure 15 and then tracking performance of fuzzy and neuro fuzzy

controllers are discussed. In addition, robustness of ANFIS controller is checked by simulation for half kilogram of coffee and 50% reduction in mass flow rate of roasting air.

4.1. Open Loop Response

Here open loop response of roasting and heating system is plotted. Open loop response is response of the system without feedback loop.

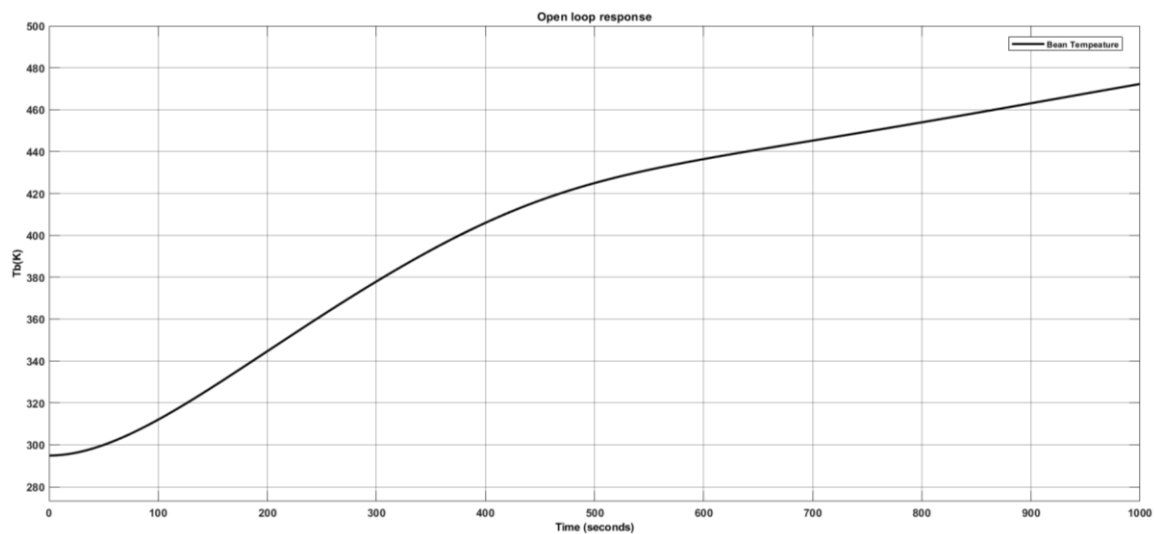


Figure 15. Open loop response of coffee roasting process.

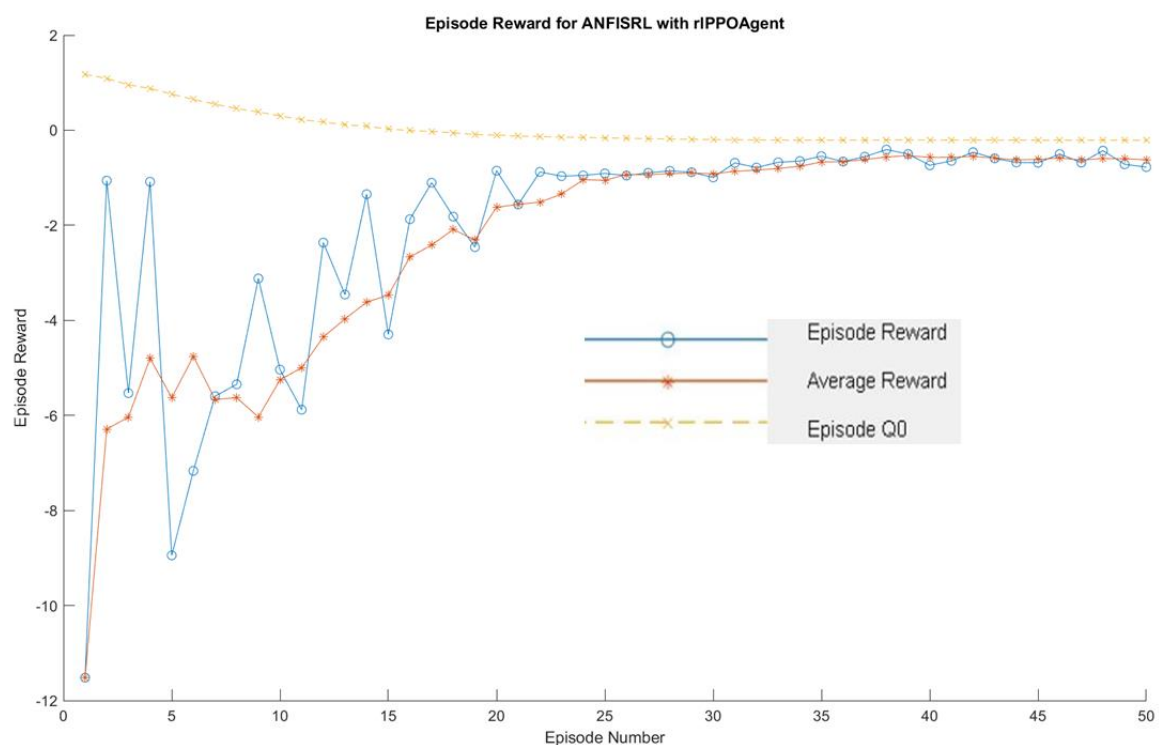


Figure 16. Training Progress of ANFIS neuro fuzzy controller.

4.2. Controlling Bean Temperature

To track a desired profile ARIC and ANFIS controllers are designed and learnable parameters initialized as discussed in chapter three. Both controllers are represented as neural network. After representing them on as a neural network the membership function are optimized using PPO learning algorithm.

a. ANFIS Tracking Controller

Tracking performance of a single controller for the whole

system is simulated. ANFIS controller is trained for fifty episodes and the agent converges immediately after thirtieth episodes.

By performing a number of training sessions best performance is obtained if membership functions of error derivative term have no learnable parameters. The stopping criterion used to finish training is episode number.

The optimized parameters of membership function are replaced in the fuzzy inference system as shown in Figure 17.

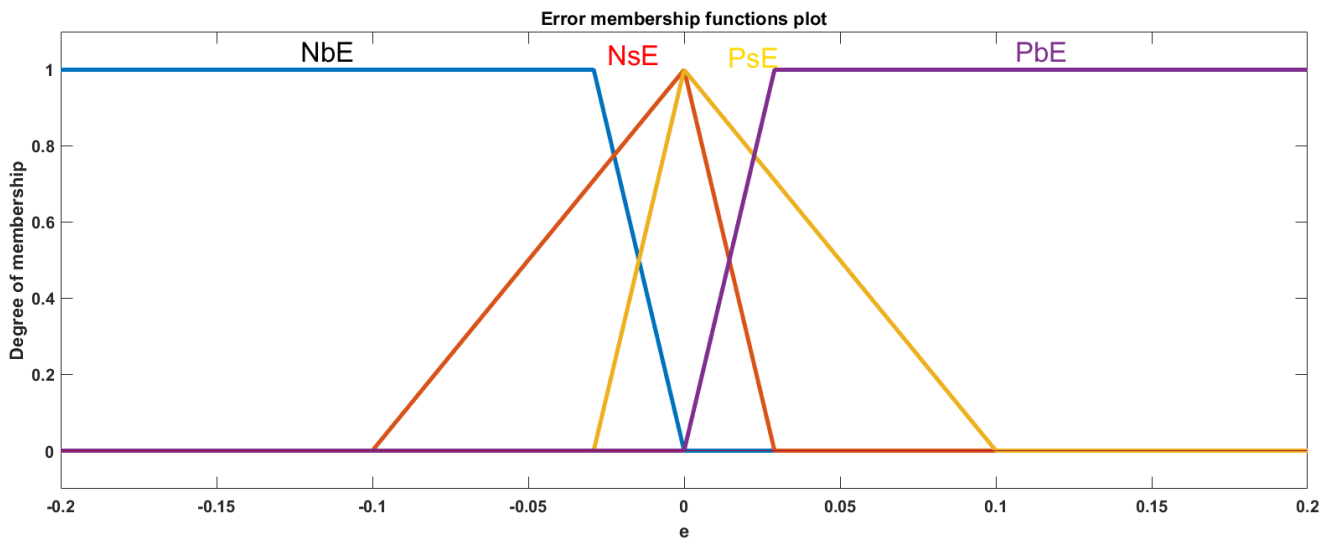


Figure 17. Optimized membership functions plot for error input.

After optimization the membership functions parameters changed as shown in Table 4.

Table 4. Parameters of membership function before and after optimization.

Membership Function	Before Optimization	Learnable Parameters
Negative big error	[-20 -20 -2 -1]	[-20 -20 -0.029004676 -6.4834234e-07]
Negative small error	[-2 -1 0.1]	[-0.029004676 -6.4834234e-07 0.1]
Positive small Error	[-0.1 1 2]	[-0.1 6.4834234e-07 0.029004676]
Positive big Error	[1 2 300 300]	[6.4834234e-07 0.029004676 300 300]
Negative error derivative	[-0.01 -0.01 -0.002 0]	None
Zero error derivative	[-0.002 0 0.002]	None
Positive error derivative	[0 0.02 0.01 0.01]	None
Low Voltage	[1 1 22.75]	[6.4834234e-07 6.4834234e-07 23.780174]
Medium Low Voltage	[1 1 150]	[6.4834234e-07 6.4834234e-07 146.25401]
Medium High Voltage	[1 1 200]	[6.4834234e-07 6.4834234e-07 196.25311]
High Voltage	[1 1 227.5]	[6.4834234e-07 6.4834234e-07 223.75261]

Figure 18 shows tracking performance for Tagako Sugeno fuzzy controller.

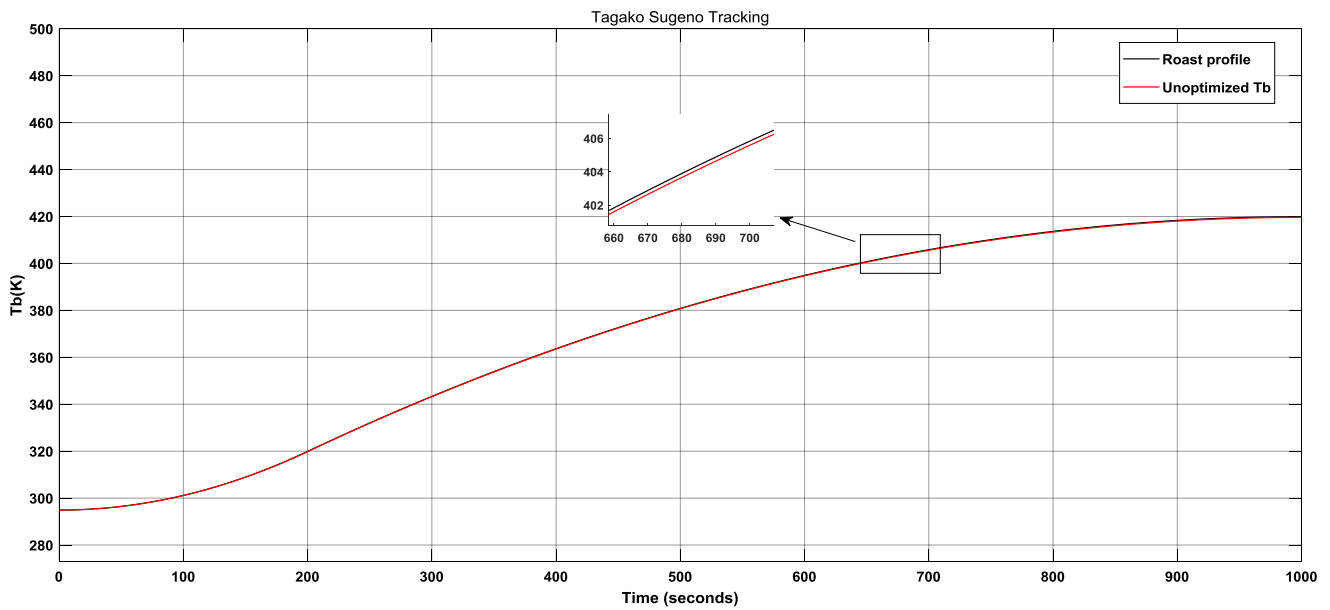


Figure 18. Tracking plot for unoptimized sugeno type controller.

Figure 19 shows tracking performance for ANFIS controller with its membership function parameters optimized using PPO reinforcement learning algorithm.

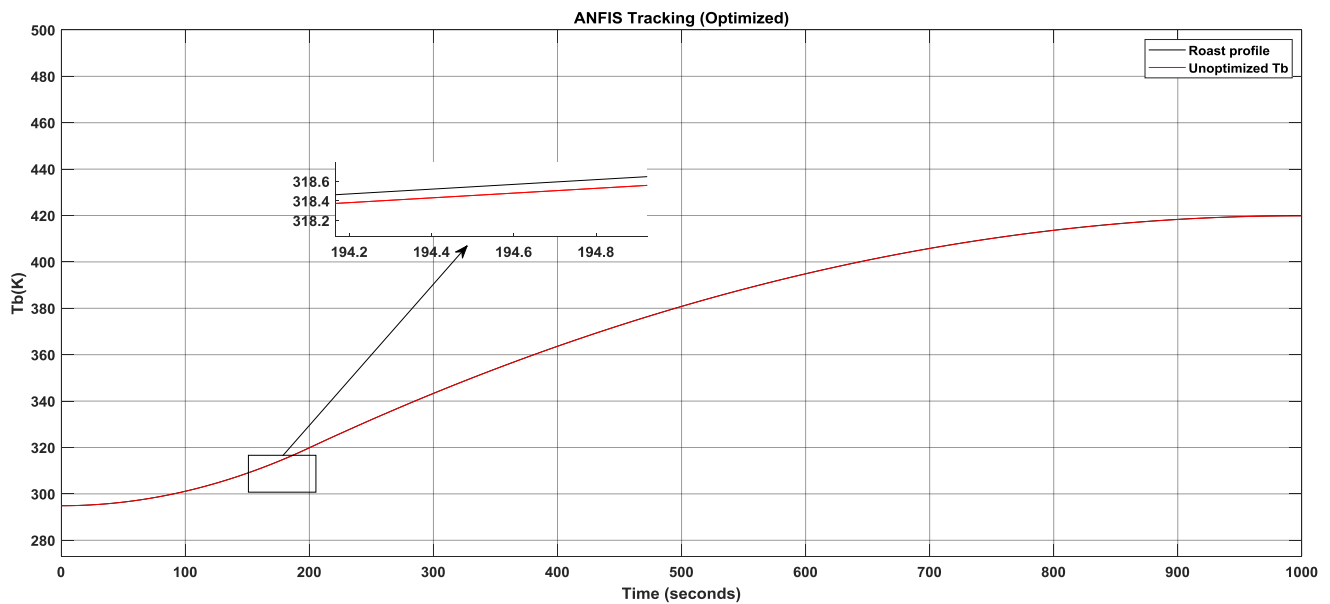


Figure 19. ANFIS neuro fuzzy controller optimized using PPO algorithm.

The RMS of error for Tagako Sugeno controller is 0.2026 where as for ANFIS controller it is 0.0624 as ahown in Figure 20.

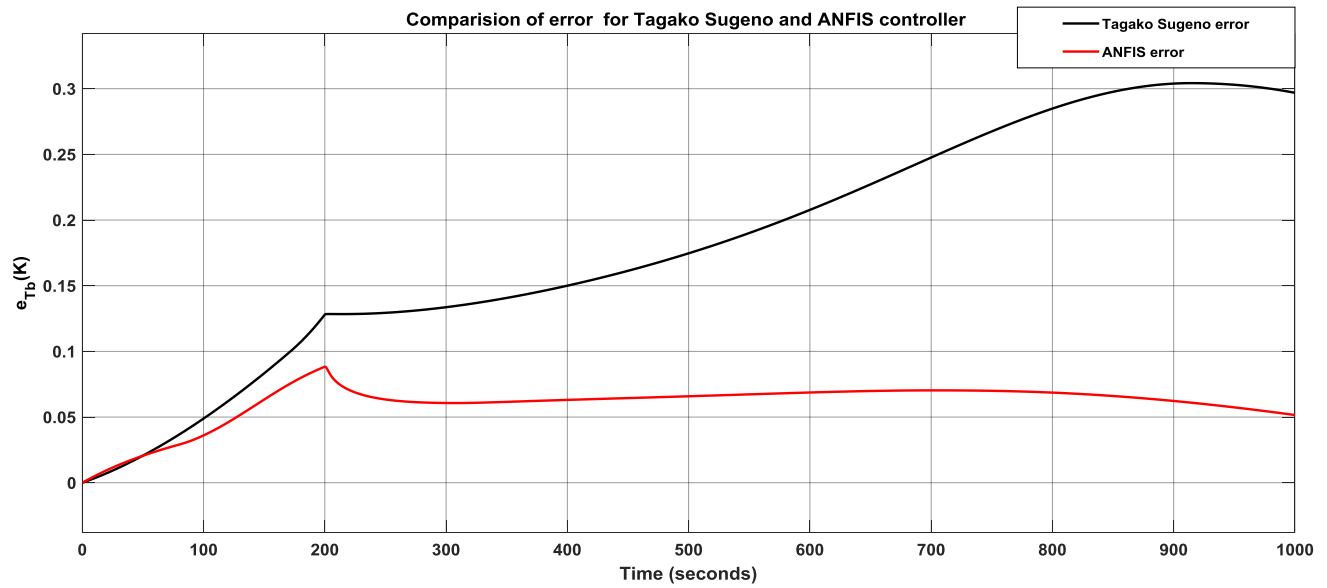


Figure 20. Comparison of Tagako Sugeno and ANFIS controller.

a. ARIC tracking controller

ARIC neuro fuzzy controller is trained for 50 episodes. As can be seen on Figure 21 the training converges at 30th episode.

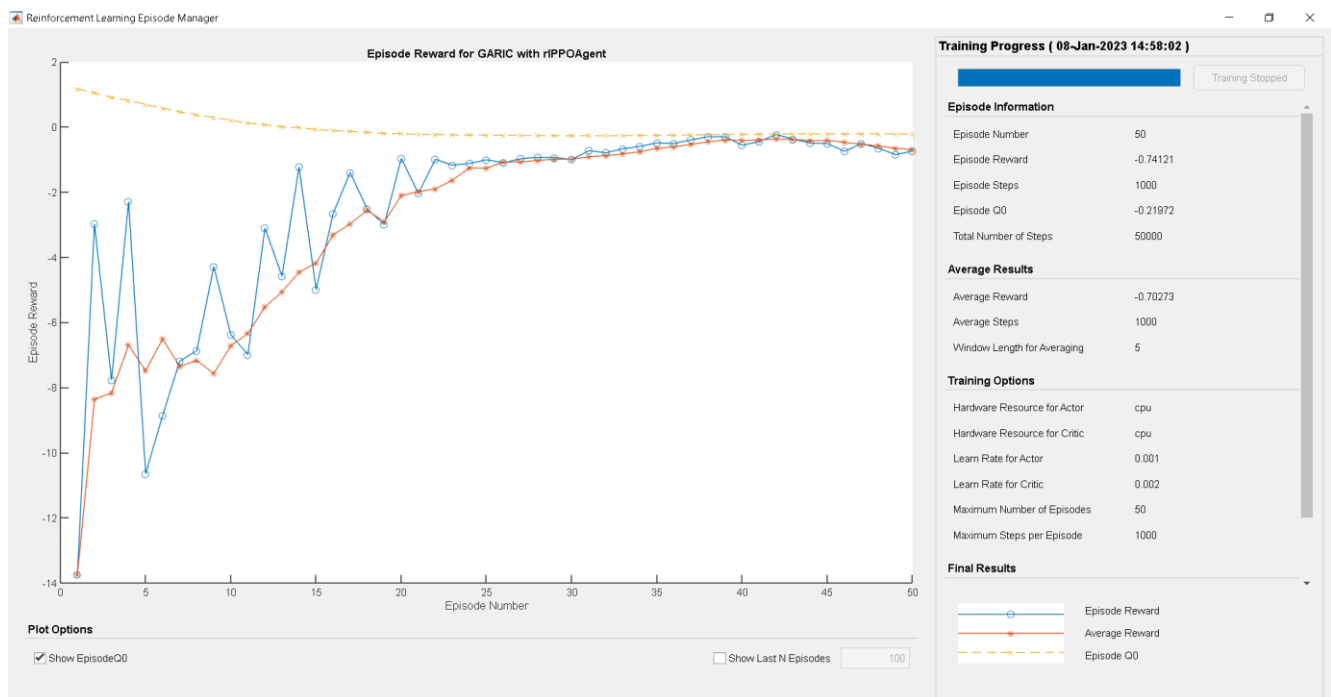


Figure 21. Training progress for ARIC neuro fuzzy architecture.

Figure 22 shows comparison of optimized and unoptimized ARIC controller. RMS of error for unoptimized ARIC agent is 0.5134 and for optimized ARIC controller it is 0.08122.

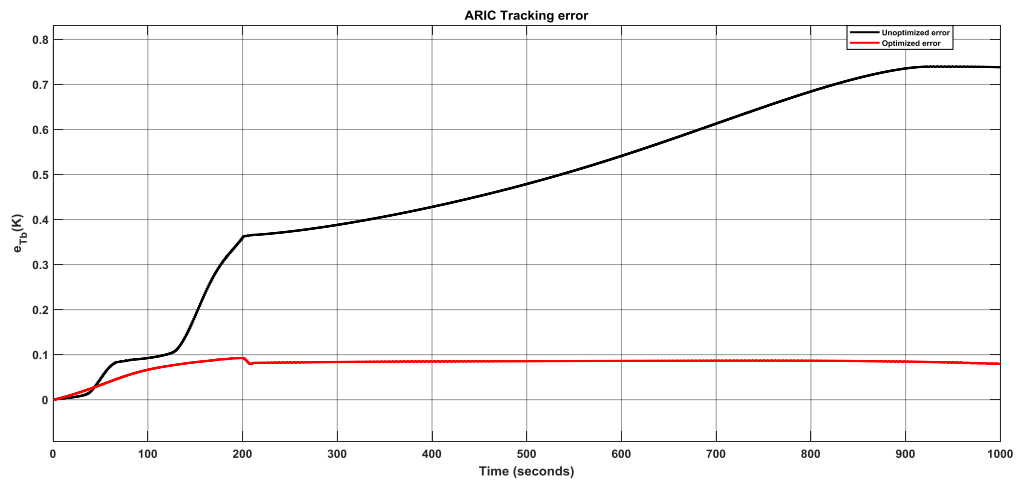


Figure 22. Comparison of error for ARIC unoptimized and optimized controller.

Performance of unoptimized ARIC controller is shown on [Figure 23](#).

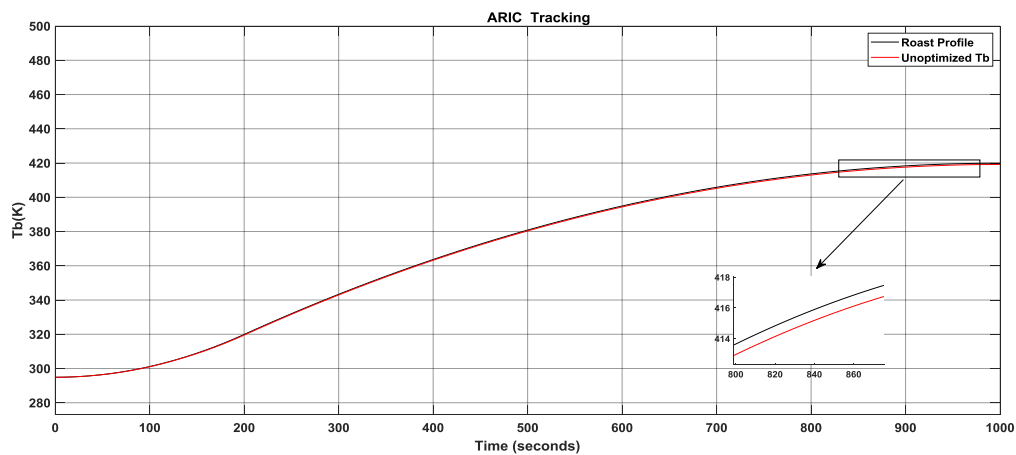


Figure 23. Unoptimized ARIC controller response.

[Figure 24](#) shows ARIC controller performance after optimization.

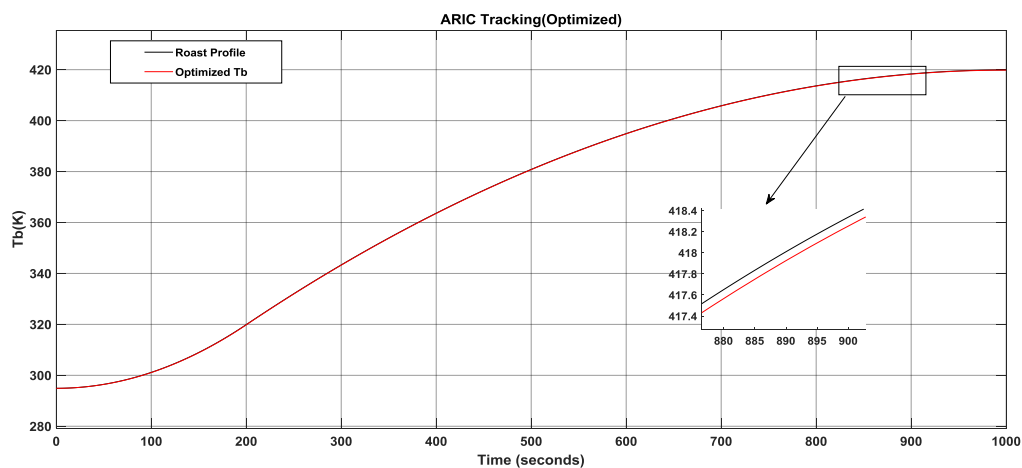


Figure 24. Tracking performance of optimized ARIC controller.

The RMS of error for optimized ANFIS controller it is 0.0624 whereas RMS of error for optimized ARIC controller it is 0.08122. Figure 25 shows error signal of the two controllers.

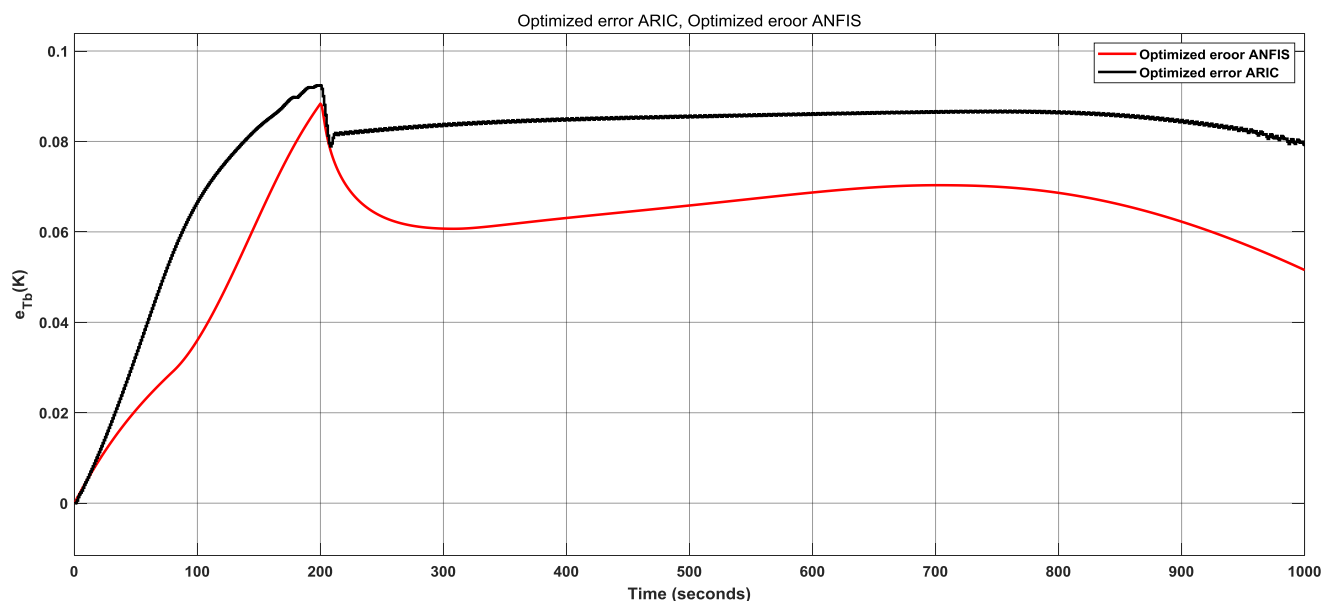


Figure 25. Comparison between ANFIS and ARIC controller.

5. Conclusion

A fuzzy controller can be represented in neural connectionist form and its membership functions can be optimized using PPO agent which is reinforcement learning actor critic agent. RMS of error for unoptimized ARIC agent is 0.5134 and for optimized ARIC controller it is 0.08122. The RMS of error for unoptimized ANFIS controller is 0.2026 whereas for optimized ANFIS controller it is 0.0624. Hence ANFIS has better performance than ARIC.

Reinforcement learning is used to optimize ANFIS controller rather than supervised learning method which requires initial data generated from other controllers. The work flow that is used in supervised learning method of ANFIS optimization is as follows: first some type of controller is designed, second the controller is optimized using PSO or any other algorithms then data is collected from this controller, third this data is used to optimize ANFIS controller. But in this thesis ANFIS controller is directly trained using reinforcement learning which reduced work flow ANFIS optimization.

In addition; ARIC which is reinforcement learning based fuzzy controller architecture is investigated and its performance compared with ANFIS. Even though both have comparable performance it can be inferred from their structure that ANFIS is more computationally efficient than ARIC. This is due to soft minimum layer of ARIC evaluates summation operation, product operation and division operation on its inputs. While ANFIS evaluates only product operation on rules layer which makes it more computationally efficient than ARIC.

The training progress converged at 30th episode this is due to the capability of neuro fuzzy systems in incorporating prior knowledge of the expert. This significantly reduces training time. With MATLAB 2021a custom layers can be designed to represent every fuzzy inference operation and set the desired parameters of membership functions as learnable or fixed.

The only drawback for neuro fuzzy systems with triangular membership functions is that, the designer ensures that while the training progress at least one rule is fired at any instant. This can be achieved by making sure that two consecutive membership functions overlap.

Abbreviations

AEN	Action Evaluation Network
ANFIS	Adaptive Neural Fuzzy Inference System
ARIC	Approximate Reasoning Intelligent Controller
ASN	Action Selection Network
FIS	Fuzzy Inference System
GARIC	Generalized Approximate Reasoning Intelligent Controller
MOA	Mayfly Optimization Algorithm
MPO	Maximum Aposteriori Policy Optimization
MPPT	Maximum Power Point Tracking
MSM	Magnetic Shape Memory
PID	Proportional Integral Derivative
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
RGA	Relative Gain Array
SAM	Stochastic Action Modifier

Author Contributions

Abiy Amare: Conceptualization, Formal Analysis, Writing - original draft, Writing - review & editing

Solomon Seid: Supervision, Writing - review & editing

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] A. Abraham, "BEYOND INTEGRATED NEURO-FUZZY SYSTEMS: REVIEWS, PROSPECTS, PERSPECTIVES AND DIRECTIONS." [Online]. Available: <http://ajith.softcomputing.netPetrovic>
- [2] Petrovic, K. Macek, and N. Peru, "A KNOWLEDGE-BASE GENERATING FUZZY-NEURAL CONTROLLER," 2000.
- [3] M. Zhou, B. Hu, W. Gao, and J. Wang, "Reinforcement Learning Fuzzy Neural Network Control for Magnetic Shape Memory Alloy Actuator," *International Journal of Control and Automation*, vol. 7, no. 6, pp. 109-122, Jun. 2014, <https://doi.org/10.14257/ijca.2014.7.6.11>
- [4] S. Rößler et al., "Two types of magnetic shape-memory effects from twinned microstructure and magneto-structural coupling in Fe_{1+y}Te," *Proc Natl Acad Sci U S A*, vol. 116, no. 34, pp. 16697-16702, Aug. 2019, <https://doi.org/10.1073/pnas.1905271116>
- [5] M. H. F. Zarandi, J. Jouzdani, and I. B. Turksen, "Generalized reinforcement learning fuzzy control with vague states," *Advances in Soft Computing*, vol. 41, pp. 811-820, 2007, https://doi.org/10.1007/978-3-540-72432-2_81
- [6] N. T. T. Vu, H. D. Nguyen, and A. T. Nguyen, "Reinforcement Learning-Based Adaptive Optimal Fuzzy MPPT Control for Variable Speed Wind Turbine," *IEEE Access*, vol. 10, pp. 95771-95780, 2022, <https://doi.org/10.1109/ACCESS.2022.3205124>
- [7] M. Ali, T. Fahmi, H. Nurohmah, H. Suyono, and M. A. Muslim, "Optimization on PID and ANFIS Controller on Dual Axis Tracking for Photovoltaic Based on Firefly Algorithm."
- [8] H. Vinh Nguyen, H. Chi Minh City, H. Nguyen, M. Tien Cao, and K. Hung Le, "Performance Comparison between PSO and GA in Improving Dynamic Voltage Stability in ANFIS Controllers for STATCOM," 2019. [Online]. Available: www.etasr.com
- [9] N. Hamouda, B. Babes, S. Kahla, A. Boutaghane, A. Beddar, and O. Aissa, "ANFIS Controller Design Using PSO Algorithm for MPPT of Solar PV System Powered Brushless DC Motor Based Wire Feeder Unit," in *2020 International Conference on Electrical Engineering, ICEE 2020*, Sep. 2020. <https://doi.org/10.1109/ICEE49691.2020.9249869>
- [10] B. Selma, S. Chouraqui, and U. Artois, "Hybrid ANFIS-ant colony based optimisation for quadrotor trajectory tracking control Hassane Abouaïssa," 2020.
- [11] M. Chen, H. K. Lam, Q. Shi, and B. Xiao, "Reinforcement Learning-Based Control of Nonlinear Systems Using Lyapunov Stability Concept and Fuzzy Reward Scheme," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 10, pp. 2059-2063, Oct. 2020, <https://doi.org/10.1109/TCSII.2019.2947682>
- [12] W. He, H. Gao, C. Zhou, C. Yang, and Z. Li, "Reinforcement Learning Control of a Flexible Two-Link Manipulator: An Experimental Investigation," *IEEE Trans Syst Man Cybern Syst*, vol. 51, no. 12, pp. 7326-7336, Dec. 2021, <https://doi.org/10.1109/TSMC.2020.2975232>
- [13] J. Degraeve et al., "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414-419, Feb. 2022, <https://doi.org/10.1038/s41586-021-04301-9>
- [14] A. Allen, L. Allen, D. Geier, B. Miller, F. Advisor, and R. Diersing, "Fluid-Bed Coffee Roaster." [Online]. Available: <http://creativecommons.org/publicdomain/zero/1.0/>
- [15] C. T. (Ching T. Lin and C. S. G. (C. S. G. Lee, *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. Prentice Hall PTR, 1996.
- [16] H. R. Berenji, "A Reinforcement Learning-Based Architecture for Fuzzy Logic Control," 1992.
- [17] Hung T. Nguyen and Michio Sugeno, *Fuzzy Systems*. Springer US, 1998. <https://doi.org/10.1007/978-1-4615-5505-6>
- [18] Nikam S R, N. P. J. And, and Kulkarni S P, "FUZZY LOGIC AND NEURO-FUZZY MODELING Journal of Artificial Intelligence," vol. 3, no. 2, 2012, [Online]. Available: <http://www.bioinfo.in/contents.php?id=71>
- [19] H. R. Berenji and P. Khedkar, "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements," 1992.