TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: M-339-MSREE-2018-2022

Voltage Control of Active Distribution Network Using Reinforcement Learning

by

Kapil Mishra

A THESIS

SUBMITTED TO DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN RENEWABLE ENERGY ENGINEERING

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

LALITPUR, NEPAL

MARCH, 2022

COPYRIGHT

Head

Department of Mechanical and Aerospace Engineering

Pulchowk Campus, Institute of Engineering

Lalitpur, Nepal

TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis entitled **"Voltage Control of Active Distribution Network Using Reinforcement Learning"** submitted by Mr. Kapil Mishra in partial fulfillment of the requirements for the degree of Master of Science in Renewable Energy Engineering.

Supervisor, Dr. Ajay Kumar Jha
Associate Professor, Department of Mechanical and Aerospace Engineering, Pulchowk Campus

External Examiner, Er. Birendra Kumar Jha
Manager, Nepal Electricity Authority

Committee Chairperson, Dr. Surya Prasad Adhikari
Head
Department of Mechanical and Aerospace Engineering, Pulchowk Campus

March 27, 2022

ABSTRACT

The decreasing trend in the Levelized Cost of energy produced from renewable energy resources and their widespread potential has motivated towards the distributed generation (DG), mainly solar photovoltaic (PV). The DG output variation and consumer load variation may cause voltage violation in the distribution feeder. In this research, a model-free control algorithm based on reinforcement learning (RL) is presented for the regulation of distribution feeders by controlling the reactive power output from the smart inverters (SI) with minimum PV generated active power curtailment. The SI serves two purposes, generation of the power to supply load demand and reactive power generation for the voltage control. An RL, deep deterministic policy gradient algorithm is used to train an agent, from training the agent will learn a policy to control the output of smart inverters based on a designed reward function and later the trained agent was used to determine the setpoint of P, Q output from SI keeping the voltage within the desired limit. This algorithm is implemented in the IEEE-33 radial distribution feeder. A Load flow program has been developed to get the voltage magnitude information of each node, using Kirchoff's law, where the voltage angle and magnitude is the function of total power flowing through the node, branch reactance & branch impedance.

Distributed generation is added in the IEEE-33 bus, and load flow analysis is performed. The analysis shows that without DG connection and at a low level of DG penetration, the line loss increased with the load and node voltage decreased and vice versa. However, with the high penetration of the DG, the line losses have increased during light load.

The designed DDPG agent has successfully kept the node voltage within the limit under the varying load. Also, the active power curtailment of the model is compared with the volt-VAR droop model and results show that the active power curtailment by the model is less than the volt-VAR droop control method by 2.41 percent.

ACKNOWLEDGMENT

The master’s thesis on the topic “Voltage Control of Active Distribution Network Using Reinforcement Learning” has been completed successfully with guidance, encouragement, and the constant source of support and inspiration from my supervisor, friends, and family members. I would like to take this opportunity to extend express my sincere gratitude appreciation and indebtedness to all those persons who have helped me during the entire period of research work.

First of all, I would like to thank the Institute of Engineering, Pulchowk Campus, Department of Mechanical and Aerospace Engineering for including thesis work in our master’s degree. I would like to thank my supervisor Dr. Ajay Kumar Jha for his continuous guidance and support during the entire period. I would also like to thank Dr. Surya Prasad Adhikari, Head of Department, Department of Mechanical and Aerospace Engineering, and Dr. Hari Bahadur Darlami, Program Coordinator, Renewable Energy engineering for his constant guidance.

Last but not the least; I would like to thank Srijanshil Puraskar Fellowship Program for supporting this thesis by Manchuka Memorial Fund.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC	Alternating Current
ANSI	American National Standards Institute
DDPG	Deep Deterministic Policy Gradient
DG	Distributed Generation/Dispersed Generation
DERs	Distributed Energy Resources
DQN	Deep Q-Network
GA	Genetic Algorithm
IEEE	Institute of Electrical and Electronics Engineers
kV	Kilo-Volt
kW	Kilo-Watt
MATLAB	MATrix LABoratory
MDP	Markov Decision Process
NEA	Nepal Electricity Authority
OLTC	On-load tap changer
PMU	Phasor measurement unit
PPO	Proximal Policy Optimization
PVs	Photovoltaic System
PU	Per-Unit
RL	Reinforcement Learning
RES	Renewable Energy Source
SCADA	Supervisory Control and Data Acquisition
SG	Synchronous Generator
SIs	Smart Inverters
SVC	static VAR compensator
VAR	Volt-amp reactive

CHAPTER ONE: INTRODUCTION

1.1 Background

An electric power system is expanding rapidly. With the increased connection of the distributed energy resources (DER) such as solar PV, wind, hydropower, etc., the power system architecture becomes more complex which creates difficulty in reliable and quality operation of the system. During lightly loaded conditions, the presence of the local generation sources on the distribution feeder causes overvoltage at the point of the coupling and also causes the reverse power flow in the feeder.

The goal of the distribution system operator is to accommodate the maximum distributed generation without violating the operational constraints such as reliability, frequency, voltage (Klonaria, et al. 2016). The recent development of the data management system in the power system is an opportunity for the distribution system operator to develop a real-time data-driven system to overcome such operational challenges (Hojabri, et al. 2019),. Many researchers have proposed feasible methods to control the feeder's voltage under this condition.

Many researchers have proposed feasible model-based methods to control the feeder's voltage under this condition. However, those methods could not respond to the uncertainties in PV generation and energy consumption. In this thesis, a smart inverter is controlled using the reinforcement learning algorithm. Reinforcement learning is a data-driven control algorithm where the RL agents take action based on the data in an environment and receive a reward. Based on the received reward the agent will improve its further action. There are several types of reinforcement learning agents, for example, Q-learning agents, Deep Q-Network agents (DQN), SARSA agents, PPO agents, DDPG agents. For the continuous observation and continuous action space, a DDPG agent is used (Matlab 2021) where the agent will control the reactive power generation / absorption of the smart PV inverters.

The developed policy, which determines the action of the inverter in the future, will be learned through the past data of the system, which do not require any input of the system parameter. This developed smart PV inverter control algorithum will achieve better voltage regulation, and also reduce the PV generated active power curtailment, on comparison to the existing smart inverter technology.

1.2 Problem Statement

In a conventional power system, in the absence of distributed generation, the power flow from the grid end to the consumer end. The voltage of the system drops monotonously, i.e., the voltage at the grid side will be higher and decreased along the feeder (Jacome, et al. 2019). In such cases, conventional voltage control methods such as on-load tap changers (OLTCs), capacitor banks, etc. can regulate the distribution feeder voltage. Over the past years, the high penetration of distributed generation energy sources, such as photovoltaic, may cause the reverse power flow i.e., towards the grid substation which creates none uniformity in voltage distribution along the line. This has made distribution networks faced with the problem of voltage regulation (Júnior, Waenga and Pinto 2018).

Conventional voltage control algorithms such as on-load tap changers (OLTCs), shunt capacitors, etc., are used for the voltage regulation in distribution feeders whose response to the voltage variation is slow and localized. Fast regulatory devices PV inverters and static var compensators (SVCs)) need to cut off during the severe voltage violation and once the system was restored normal then it will be connected back to the system (Singh, 2017). The cut-off of SI causes the DG-generated active power curtailments. Also, During this switching process, the system will generate the transient voltage and current which results in a decline in the quality of the (Klonaria, et al. 2016). Under the new standards/rules by IEEE std-1547, 2018 (IEEE 2018) , PV inverters are required to contribute to grid regulation via smart functions.

With the development of communication technology and expansion of data acquisition systems such as SCADA, PMU has made the system data readily available to the distribution system operators (Hojabri, et al. 2019). Many researchers have proposed a droop control also called Volt-Var control method which is a model-free and data-driven control algorithm. Researchers have suggested that voltage regulation can be achieved using the Volt-Var control method. During, voltage variation the SI generates/consume the reactive power which reduces the active power generation by PV (Lusis, Andrew and Liebman 2020). The optimization of the Volt-Var droop curve can be achieved by various optimization techniques such as Genetic algorithm (GA), particle swap optimization(PSO) etc. However, still the optimization does not always result the best outcome because of uncertainty in the PV generated load and consumer consumption variation (Li, Jin and Sharma 2019). Hence, a model-free control algorithm based on reinforcement learning (RL) is presented, for the regulation of distribution feeder voltage by controlling the reactive power output from the smart inverters (SI) with minimum PV generated active power curtailment

1.3 Research Objectives

1.3.1 Main Objective:

To design a system for voltage control of active distribution Network using Reinforcement Learning.

1.3.2 Specific objective:

- To develop a load flow program for a radial distribution system in MATLAB and analyze the effect of load variation and DG output variation in system voltage & power loss.
- To develop a reinforcement learning environment and train DDPG agents using MATLAB Simulink.
- To analyze and validate the performance trained agent for voltage control of IEEE-33 radial distribution bus with DG.

1.4 Assumptions and Limitations

- The IEEE 33 node test feeder model is developed and simulated in the MATLAB software instead of a real field/practical radial distribution feeder.
- This thesis is purely technical work, economical analysis shall not be performed.
- The reliability study of the system shall not be performed.

1.5 Outline of the Thesis

This report is categorized into five chapters:

Chapter 2: This chapter gives an overview of the effect of load and generation variation of the distribution system. It also gives an overview of the methods for voltage control and describes reinforcement learning.

Chapter 3: This chapter describes the methodology used to fulfill the objectives of the study.

Chapter 4: Explain the effect of load variation on voltage and also analyze the reinforcement learning-based voltage control.

Chapter 5: Summarizes the thesis's main points and contributions., and proposes future directions for research.

Finally, the thesis ends with a list of papers referred for this study.

CHAPTER TWO: LITERATURE REVIEW

2.1 Distributed generation resources

Conventional power plants such as oil, coal, natural gas, nuclear power plant are located remotely. The electrical energy produced from those large-scale conventional power plants is transmitted to the grid substation near the major load center. The energy from the grid substation shall be distributed to the consumer through a low voltage distribution line. During this process, a significant amount of energy loss occurs during transmission from a power plant to the consumer (Abdel-Ghany, et al. 2015). On the other hand, distributed energy resources are spread over the geographical area. The presence of DG nearby the load center benefits the reduction in line losses (Singh 2017).

Distributed generation is a concept of generating electricity through small-scale technologies close to the load to be served. Small hydropower, biomass, wind power, solar power, wind power, thermal power generated near the customer premises are examples of distributed generation. They are often connected at the low voltage distribution lines. Since, the generation is nearby the load, even in the absence of the grid power, the local generation can serve the load which increases the energy security. Also, the reliability will increase. Moreover, DG sources can use islanding techniques to serve the local distribution network even during the outage of the central grid (National Planning Commission 2018). The distributed generation also increases the economic activity in society. Most of the distributed generation comes to form a clean source of the energy, hence preventing the deterioration of the environment. National Planning Commission, Nepal proposed a plan for sustainable distributed generation and grid access to all by 2022 as shown in table 2 (National Planning Commission 2018).

Table 2-1: Distributed Generation projects

Distributed Generation projects	Number
Hydropower	221
Solar PV	481
Biomass to Electricity	50
Wind Power	1

2.2 Solar Photovoltaic

Sun is the ultimate source of the energy. The energy coming from the Sun can be converted into a useful form using different technologies. Solar energy is renewable energy, a source that is not finite or exhaustible. Solar power contributes more than 2 percent of the current energy demand of the world and is rapidly growing. Nepal got a total potential of 50,000 terawatt-hours per year which is 7000 times higher than the total hydroelectricity potential in Nepal (Andrew Blakers; Sunil prasad Lohani 2020). Solar energy can be converted into electricity using solar PV technology. A solar PV cell converts thermal and radiant energy from sunlight into a direct current. Semi-conductive materials such as silicon, germanium are used to made solar cells. The material used in this solar cell is silicon and in nature. The materials are used in mono-crystalline, poly-crystalline, and amorphous silicon forms.

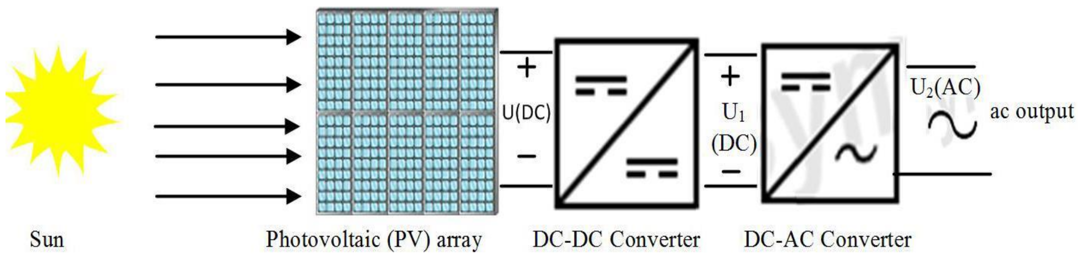


Figure 2-1: Solar PV Power Generation

The dc output current and dc voltage vary with irradiation reaching the PV array module and surrounding temperature. The DC-DC converter ensures that the dc (U1) fed to the dc-ac inverter is stable. The energy from solar PV can be converted into AC for use using a DC-AC converter.

2.2.1 Solar Output Characteristics

Solar power generation is fluctuating, uncontrollable, and unpredictable and depends on resources that are location-dependent. The solar PV output is a function of solar irradiance. The solar PV output also depends on the temperature, with the increase in cell temperature, the energy output decreases. Also, the shadow due to clouds, obstruction of solar radiation by objects sharply decrease solar PV array output. Figure 2.2, provides a graphical example of hourly PV power variability.

Table 2-2: Location data

Host institution	Department of Hydrology and Meteorology (DHM)
Location	Lumle, Nepal
Date	7/1/ 2018
Elevation (m)	1740
Latitude (positive north, decimal degrees)	28.2966
Longitude (positive east, decimal degrees)	83.8179

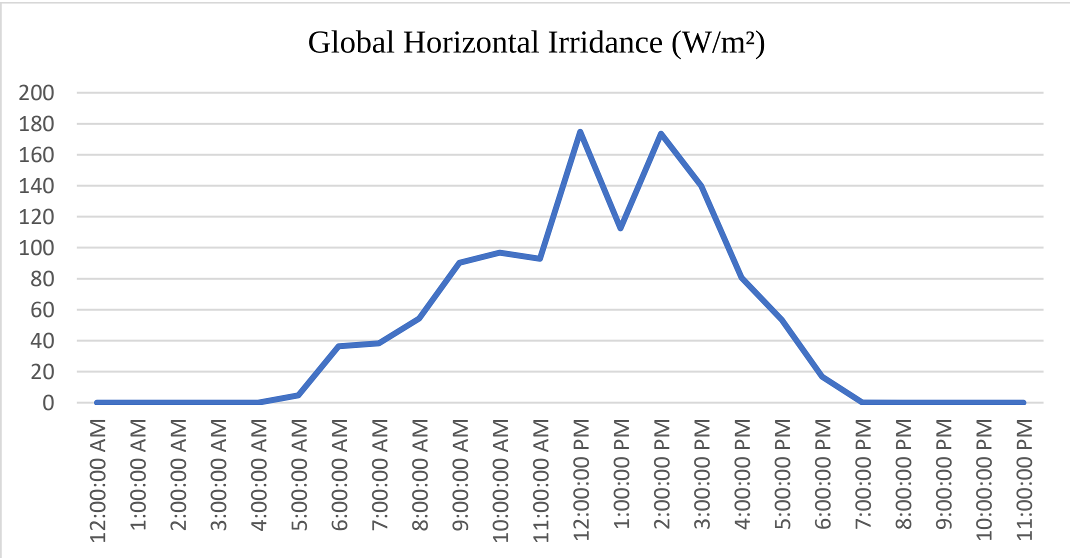


Figure 2-2: Irradiance variation with time

2.3 Voltage regulation

In an electric power system, the voltage magnitude of the sending end voltage and receiving end voltage are different. The voltage must be different for the power flow. Measurement of this voltage difference is voltage regulation.

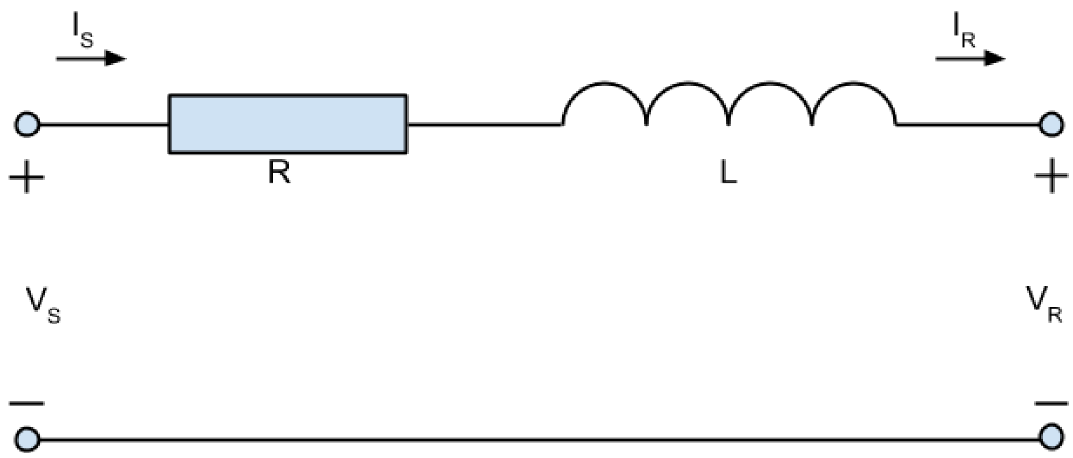


Figure 2-3: Short Line Model

Voltage regulation (%) $= \frac{V_S - V_R}{V_S} X100$ …………………………….. Equation 2-1

Here; V_S= Sending end voltage magnitude

V_R= Receiving end voltage magnitude

The voltage variation occurs due to the line impedance between sending end and receiving end. The voltage regulation for the ideal line, i.e., line with zero resistance and reactance is zero.

The voltage regulation for the distribution line is considered as ±5 percentage or ±10 percentage. For example, in ANSI C84.1 has set distribution voltage tolerance as ±5 percentage.

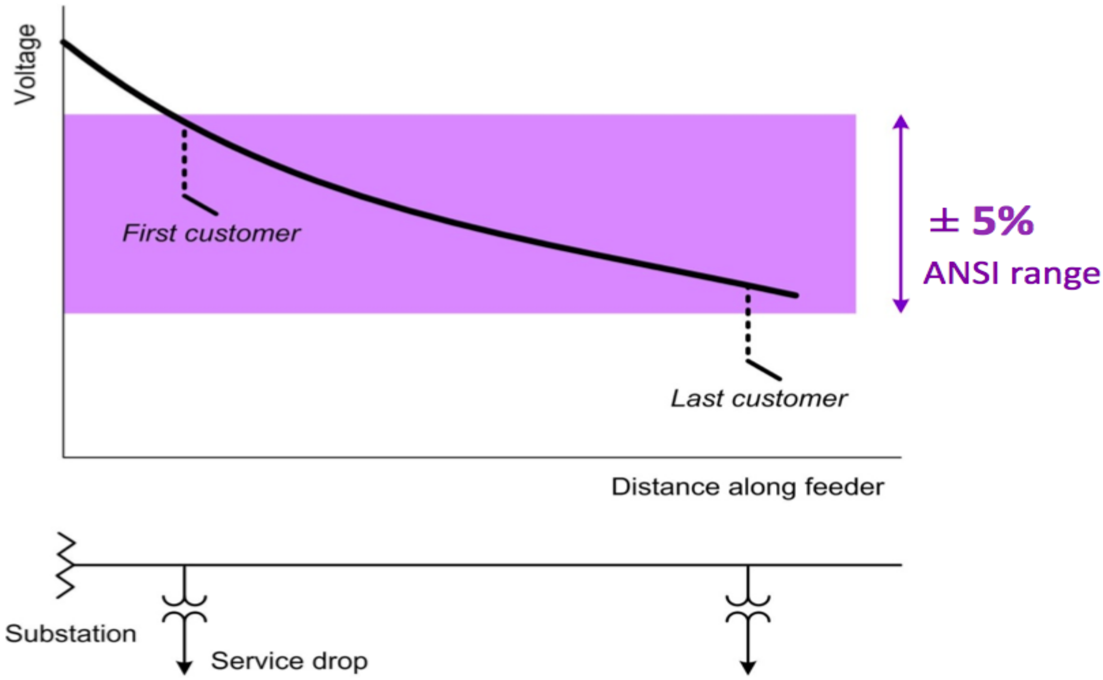


Figure 2-4: Voltage Profile in Distribution line

2.4 IEEE-33 Bus

The IEEE-33 bus system is a radial distribution feeder system is a generic model developed by IEEE to facilitate customization for more specific studies. This radial feeder consists of thirty-two line branches and thirty-three buses. The base voltage is 12.66kV, the total connected load is 3.715MW, and 2.3MVar (Vita 2017) .

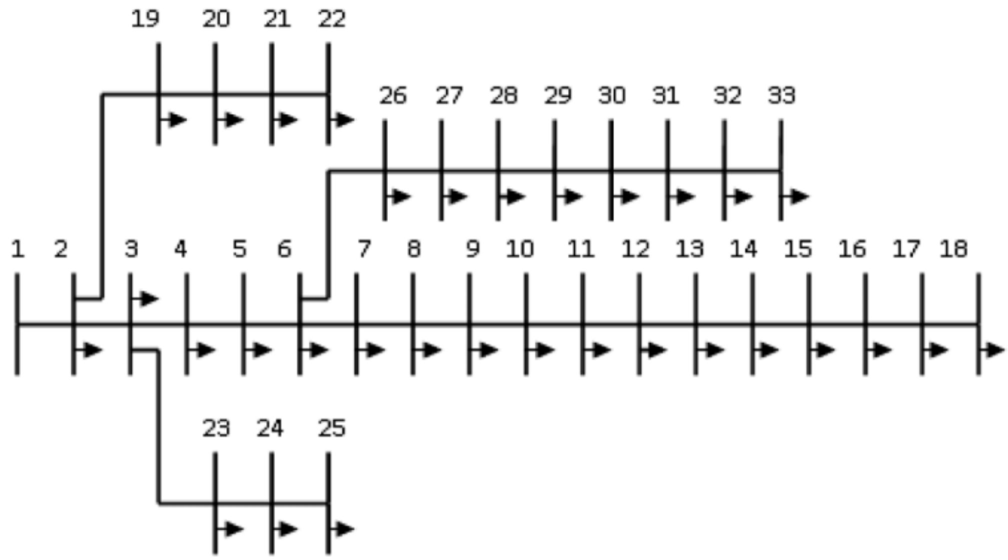


Figure 2-5: Single line diagram of the IEEE 33-bus radial distribution feeder

Table 2-3: Load data of the IEEE 33-bus radial distribution system

Load	Location (Bus Bar)	Real Load (kW)	Reactive Load (kVAR)
L2	2	100	60
L3	3	90	40
L4	4	120	80
L5	5	60	30
L6	6	60	20
L7	7	200	100
L8	8	200	100
L9	9	60	20
L10	10	60	20
L11	11	45	30
L12	12	60	35
L13	13	60	35
L14	14	120	80
L15	15	60	10
L16	16	60	20
L17	17	60	20
L18	18	90	40
L19	19	90	40
L20	20	90	40
L21	21	90	40
L22	22	90	40
L23	23	90	50
L24	24	420	200
L25	25	420	200
L26	26	60	25
L27	27	60	25
L28	28	60	20
L29	29	120	70
L30	30	200	600
L31	31	150	70
L32	32	210	100
L33	33	60	40
	Total load	3715	2300

2.5 Voltage Regulation in Power System

The voltage of the power system varies with the change in the generation and load. The voltage of the system will increase when the load decreases and vice versa. To keep the power system voltage within the limit additional equipment is required to add to the system (Kowsalya 2018). Some of the devices used for the regulation of the power system are as follows.

i) Transformer Tap Changing – The transformer turn ratio is changed to achieve the voltage in the desired limit. The transformer tap can be changed with or without disconnecting the transformer from the supply. The tap changing can be achieved manually or automatically.

ii) Shunt Reactor – The shunt reactor is the inductors that are connected in between the line and neutral. The role of the shunt reactor is to compensate for the capacitive current generated from the high voltage transmission line and power cables. Shunt reactors are connected at the sending and receiving end of the substation.

iii) Shunt Capacitors – The shunt capacitors are connected in parallel with the line. Its role is to compensate the reactive power absorbed by the line. Like the shunt reactor, it is installed at the switching substations, distribution substations, receiving end substations.

iv) Synchronous Phase Modifier – Synchronous phase modifier is a over-excited synchronous motor connected with the load at receiving the end of the line, running without a mechanical load. Synchronous Phase Modifier not only keeps the voltage constant but also improves the power factor.

v) Smart Inverter – IEEE has released a new standard, IEEE std-1547, 2018 (IEEE, 2018) for inverters that are to be used in PV connection to the grid. Under this rule PV inverters are required to contribute to grid regulation via smart functions. To allow the smart inverter for the reactive power generation for injection/ absorption on the grid, the inverter capacity (S) must be greater than PV produced real power (P) of the PV arrays. The smart inverter output curve is shown below.

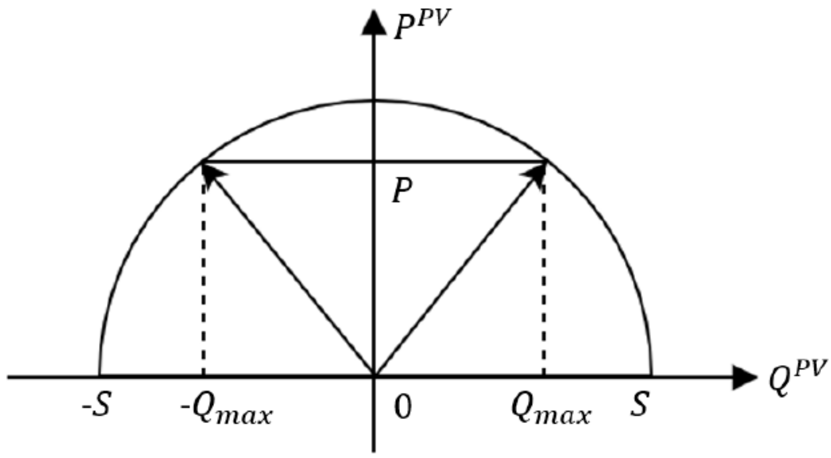


Figure 2-6 : Smart PV Inverter Output Curve

2.6 Voltage Control using Volt–Var Droop of Smart Inverter

Figure 2.7 shows the concept of the voltage regulation by smart inverter using Volt-Var Droop control method. This method outputs the reactive power depending upon the voltage magnitude at the point of the connection. The voltage between V_3 and V_4 is in an acceptable range. During this interval, the reactive power (VAR) generation by SI is zero. If the system voltage drops below V3 then the SI will generate the reactive power and increase above the V4, the SI will absorb the reactive power. The magnitude of the reactive power generation/absorption is determined by the slope of the curve. Similarly, if the voltage is less than the V2 it will generate the maximum reactive power and inject it into the feeder and if the voltage is greater than V5, it will absorb the maximum reactive power from the grid.

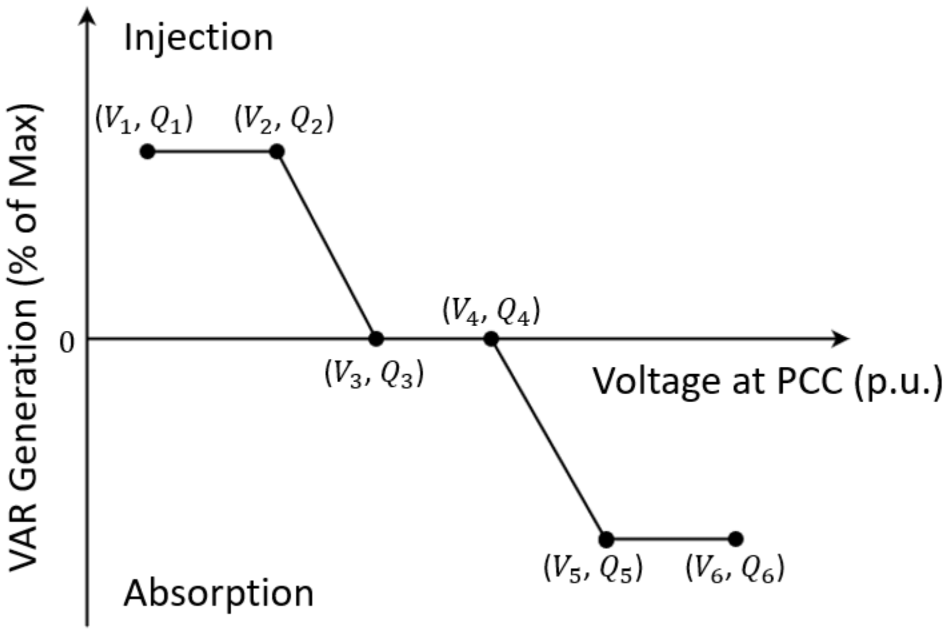


Figure 2-7: Smart PV Inverter Output Curve

2.7 Reinforcement Learning

Reinforcement learning is a branch of the machine learning algorithm that learns through reward function. A reward function is a judgment of how good or bad is the action taken by the agent. The reinforcement learning algorithm will train an agent to achieve the maximum reward. An agent observes the environment and take an action. If the action of the agent is good, then it will get a positive reward else it will get a negative reward. The agent stores the information of the observation, action, and reward it got. During the training, the agent will take an action that has previously provided it a positive reward, it is very unlikely that the agent will take the bad action again. Doing this in a large divergent environment, the agent will explore itself and finally learns a policy that will yield large possible rewards. (Sutton and Barto 2015). This is summarized in the figure below.

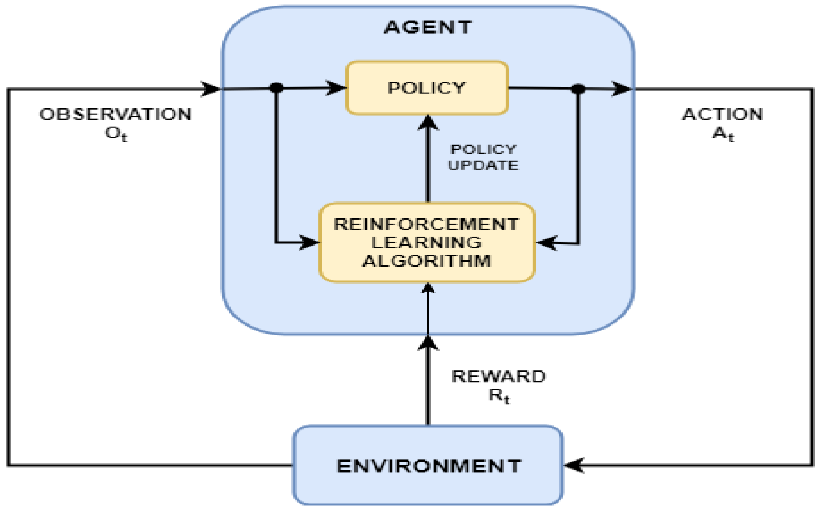


Figure 2-8: Reinforcement Learning Flowchart

2.4.1 Reinforcement Learning Workflow

To develop a reinforcement learning algorithm for control of the environment parameter, first, we have to formulate the problem. During this, we will learn what are the input and outputs of the environment, how the agent will interact with the environment. The main objective is to know what input will agent take and what goal we have to achieve. Then we will create an environment, where the training of agents will occur. To judge the action of the agent we have to create a reward function. Then we will create an agent, the selection of the agent type depends on the action type and environment type, i.e., continuous or discrete. The agent will then be trained in the environment. After the successful training, the agent will training will be validated in the environment

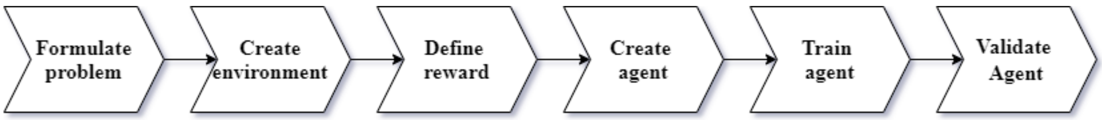


Figure 2-9 Reinforcement Learning Workflow

2.4.2. Deep Deterministic Policy Gradient (DDPG) Algorithm

Deep deterministic policy gradient is a model-free off-policy learning algorithm, which is also called an actor-critic algorithm (Timothy P. Lillicrap 2015). The objective of the DDPG agent is to maximize the expected cumulative long-term reward. DDPG works on continuous action and state space.

Observation Space	Action Space
Continuous or discrete	Continuous

DDPG agent consist of actor and critic of following types.

Observation Space	Action Space
Q-value function critic Q(S,A),	Deterministic policy actor π(S),

Deterministic policy is a policy which maps that maps the state into the actions. It is used in a deterministic environment where the outcome of the action is known.

The figure below shows the working principle of the DDPG agent.

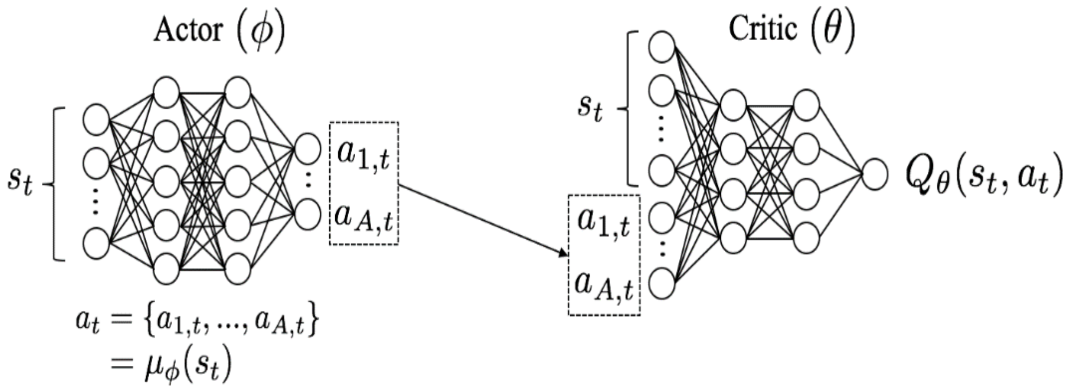


Figure 2-10: Working principle of the DDPG agent, an actor-critic architecture.

Initially, the environment is at the initial state S_t, the actor neural network, which is a neural network with wight parameter, φ choose an action μ_φ (st). The environment will generate a reward r_t for this action and environment transit to next state S_{t+1}. This experience tuple (st, at , rt , st+1) is then stored in the replay memory. The critic parameter is updated by minimizing the loss L across all sampled experiences.

$L = \frac{1}{M}\sum_{t=1}^{M}(yi - Q(St, At; \phi))^2$ …………………………………… Equation 2-2

Where, M is the mini-batch of experience

yi=Ri+γQt(Si′,πt(Si′;θt);ϕt) ……………………………………… Equation 2-3

Where; yi is value function target

Ri is the reward of i^{th} Episode

γ is the discount factor

Q_t is sum of future reward

Similarly, the actor parameter is updated using sampled policy gradient to maximize the expected discounted reward.

$\nabla\theta J \approx \frac{1}{M}\sum_{t=1}^{M} G_{at}G_{\pi t}$ ……………………………………………………….…… *Equation 2-4*

where $G_{at} = \nabla_A Q(S_t, A; \phi)$ and $G_{\pi t} = \nabla\theta\pi(S_{t+1}, \theta)$& $A = \pi(S_{t+1}, \theta)$

Actor and critic networks are updated until convergence using one of the following target update methods.

- Smoothing, periodic & Periodic smoothing

2.8 Load Flow Analysis

Load flow analysis also called power flow analysis of a power system is a steady-state analysis to determine the operating condition of the system. The objective of load flow analysis is to determine node voltage magnitude and phase angle in given load (active and reactive) cases and to calculate the power losses etc. Widely used load flow analysis methods are the Newton-Raphson method, Gauss-Seidel method, and fast decoupled method (Sivkumar and Das 2008). The distribution system has a radial structure and high line resistance to reactance ratios as compared to a transmission line. So these load flow methods take a longer duration for convergence, take more computational memory and can not provide good results in the distribution system.

CHAPTER THREE: METHODOLOGY

3.1 Basic

This section describes the method followed during the research process method. The IEEE 33 Bus radial distribution system is used as a test network for obtaining the objective of this thesis. The standard IEEE 33 Bus system's data was obtained from the standard document of IEEE and load flow was carried out using MATLAB under the standard condition. The optimal location and capacity of the Solar units are considered to form the published paper, which uses the genetic algorithm to find the optimal size and capacity. According to (Minh Quan Duong 2018), the optimal location for the DG connection is Bus: 09 – Size: 1.0625 MW; Bus: 16 – Size: 1.005 MW; Bus: 24 – Size: 1.0447 MW; Bus: 30 – Size: 0.9518 MW; Bus. Since the observation space and action space are continuous in nature, the DDPG agent was created. The next step is to design an environment to train the DDPG agent. The DDPG agent takes the voltage at each node as an observation vector and outputs reactive power output of SI’s. The trained agent is then used to carry out the simulation under a continuously varying environment.

The flow chat shown in the figure below expresses the basic methodology of this thesis.

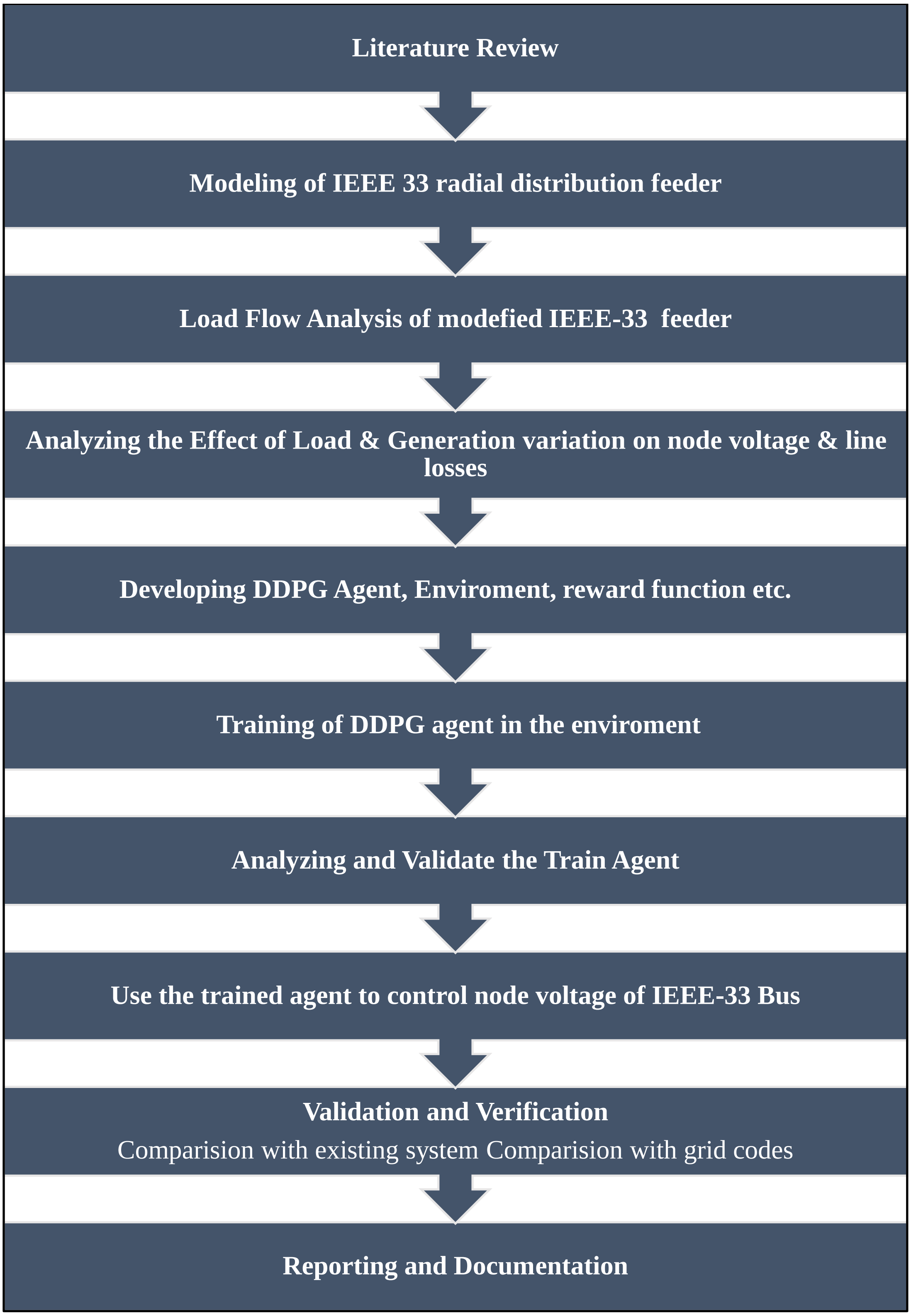


Figure 3-1 : Flow Chart of RL based voltage control

3.2 Modeling of a distribution system

Most of the distribution feeder are radial type. In this thesis, the IEEE-33 Bus radial distribution feeder is used as a test bus for implementing the concept of the voltage control using RL algorithum. The standard IEEE 33 Bus system's data can be get form the standard document (The Institute of Electrical and Electronics Engineers, Inc. n.d.). The load flow program is developed using the methods stated in chapter 2.8.

3.3 Optimal Location and Sizing of Solar PV

The optimal size and location for the PV placement in IEEE 33 radial distribution system is suggested by (Minh Quan Duong 2018).

- Bus: 09 – Size: 1.0625 MW
- Bus: 16 – Size: 1.023 MW
- Bus: 24 – Size: 1.0447 MW
- Bus: 30 – Size: 0.9518 MW

3.4 Load Flow

A load flow program has been developed based on the Kirchhoff's voltage law and Kirchhoff's Current law.

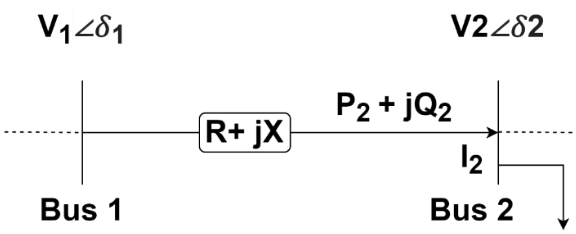


Figure 3-2 : PQ Bus

$P_2 + jQ_2 = V_2 I_2^*$ ……………………………………………….Equation 2-1

$I_2 = (P_2 - jQ_2)/V_2^*$ ………………………………………..…….Equation 2-2

$I_2 = (V_1\angle\delta_1 - V_2\angle\delta_2)/(R + jX)$ …...……………………….….Equation 2-3

Here;

$V_1\angle\delta_1$ is voltage at Bus-1

$V_2\angle\delta_2$ is voltage at Bus-2

$P_2 + jQ_2$ is power flowing through Bus-2

I_2 is current flowing through Bus-1 to Bus-2

Equating equation 1 & 2 and solving we get;

$$V_2 = \sqrt{\left[\{(P_2R + Q_2X - 0.5|V_1|^2)^2 - (R^2 + X^2)(P^2 + Q^2)\}^{\frac{1}{2}} - (P_2R + Q_2X - 0.5|V_1|^2)^2\right]}$$

……...…………………….…….Equation 2-4

$$\delta_2 = \delta_1 - \tan^{-1}\{(P_2X - Q_2 R)/ (|V_1|^2 + P_2R + Q_2X)\}$$

……...…………………….…….Equation 2-5

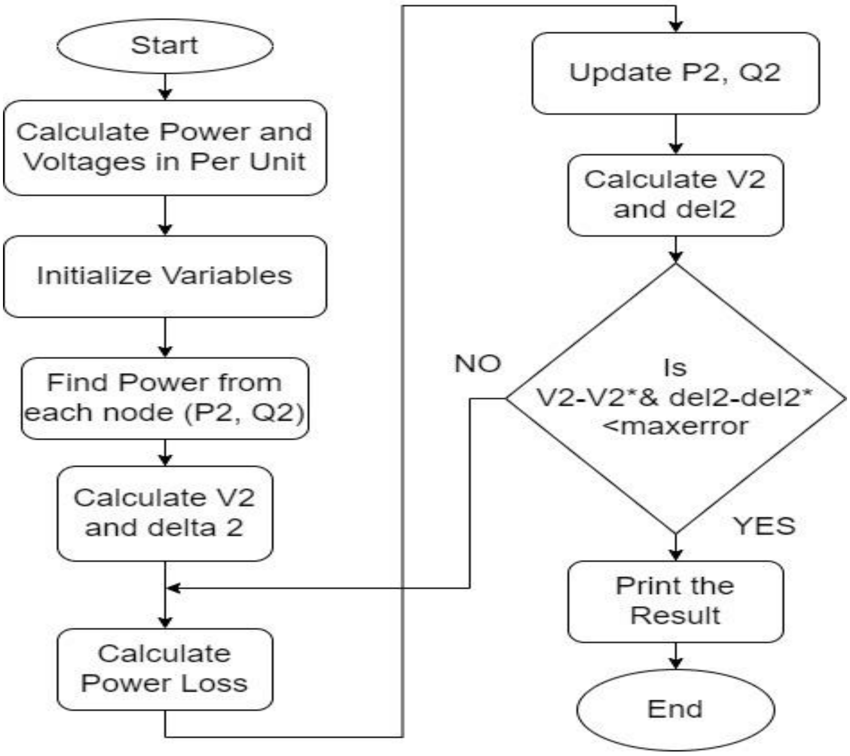


Figure 3-3 : Load Flow flowchart

3.5 Voltage Control using Volt–Var Droop of Smart Inverter

The result of (Lee, et al. 2020) optimization curve has been taken for the Volt-Var control. The weight for the figure 2-7 is , Q1=Q2=0.82 Pu; V2=0.9 Pu; V3=0.95Pu, V4=1.05 Pu, V5=1.18 PU and Q5=Q6=-0.82Pu.

3.6 Solar PV Output variation

The solar output variation is modeled by using a random number generator. It output a normally (Gaussian) distributed random signal.

3.7 RL Controller setup

A RL controller was designed in MATLAB simulink using the agent in MATLAB reinforcement learning section. A DDPG agent is used because of the need of contineous action space. DDPG agent consist of actor and critic model. The actor and critic network is created using MATLAB programming function. DDPG Agent Simulink block has an observation, reward signal, termination (isdone) input and action as output.

3.7.1 Observation Vector

The voltage output of each node is an observation vector to the agent. In the real field application, this data can be obtained form SCADA, PMU. Here, the observation vector is obtained from the load flow, which is carried out in the Power_Flow block.

3.7.2 Reward Function

In this research, reward function shall be designed using a user-defined MATLAB Simulink function. The objective of the reinforcement learning agent is to maintain the voltage within the desirable limit and to reduce the PV-generated active power curtailment. Based on these objectives the reward function is divided into two parts

(i) For node voltage limits violation: a negative reward (Penalty) proportional to the voltage deviation. The voltage is categorized into different zones, based on the severity of the voltage violation.

(ii) For active power curtailment: A penalty for proportional to SI generated reactive power.

The reward for active power curtailment is defined as:

$R_Q(j) = \sum_{i=1}^{n} C(1 - q_i); Where\ q_i = (|Qi|\ /Si)$, also called the utilization factor

Where , Qi = Reactive Power output of i^{th} Unit, Si= Capacity of the Inverter

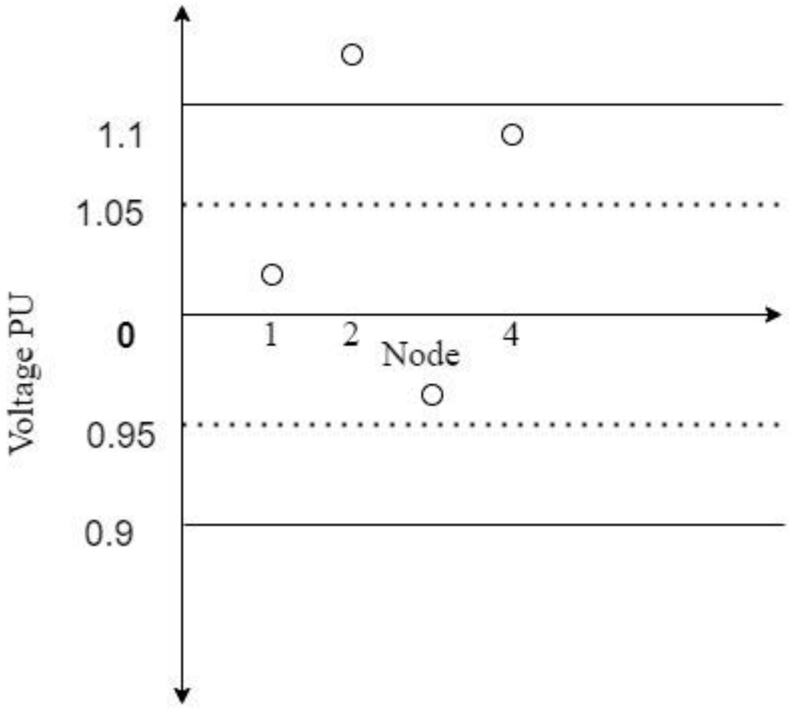


Figure 3-4 : :Voltage Zone categorization for reward

3.7.3 Action

The RL agent will take an action to achive the desired objective. In this case, the reactive power generaiton by SI’s are the action signal. Action vector consist of four action signal, which is reactive power output of each SI.

3.7.4 DDPG Training Algorithm

Figure 3-3, is the flowchart of DDPG training algorithm, which describes the process of DDPG training. Initially, the load flow is solved to get the initial observation and this is fed into the DDPG agent. The DDPG agent will take an action and generate the new reactive power output. With this new generation value, the load flow is performed again at the meantime the reward is evaluated. DDPG will update it action until the reward converges or the training reached for maximum episode.

factor.

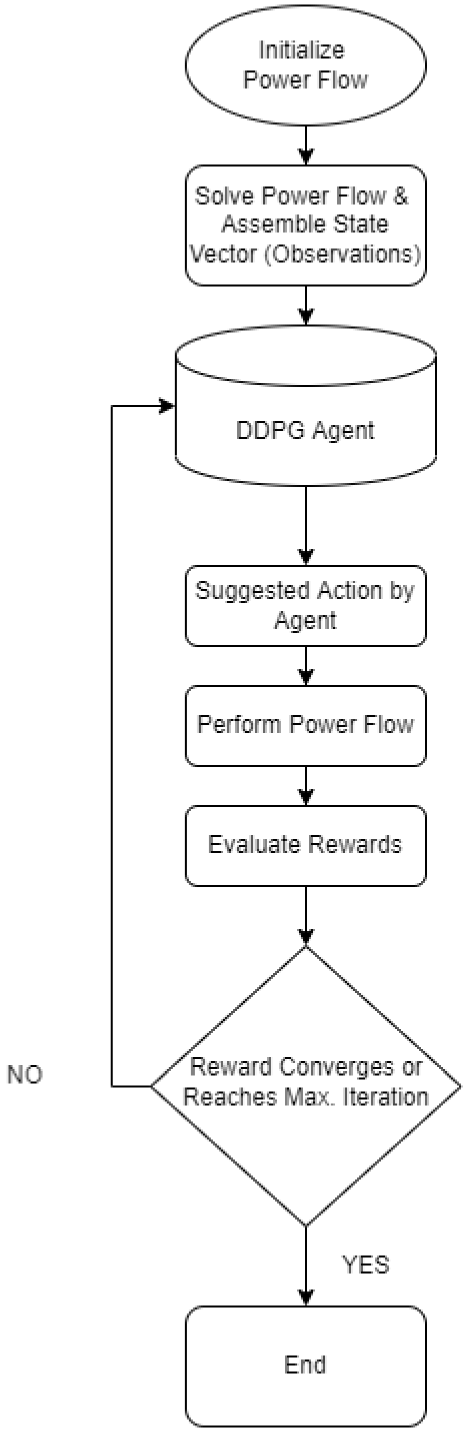


Figure 3-5 : DDPG Algorithm

The critic and actor target network is updated using the smoothing technique. The target will be updated using the target update parameter, also called target smooth factor.

The actor network is updated by:

$\theta_t = \tau\theta + (1 - \tau)\theta_t$

The actor network is updated by:

$\emptyset_t = \tau\emptyset + (1 - \tau)\emptyset_t$

Here,

τ *is target smooth factor,* $\emptyset$ *is critic parameter and* θ *is actor parameter*

3.8 Software and Tools

Some of the software that will be used for the implementation of this thesis are:

3.8.1 MATLAB 2021a

MATLAB is a high-level programming language for programming, mathematical computation, visualization. It is used in many fields of scientific and engineering applications. MATLAB use includes embedded system, control system, digital signal processing, wireless communication, image processing, and computer vision, internet of things, FPGA design and Codesign, Mechatronics, test and measurement, computation; biology and computational finance, robotics, data analytics, predictive maintenance, motor and power control, deep learning, reinforcement learning etc. (MathWorks 2018).

In this research, MATLAB Simulink is used to design an environment, and MATLAB deep learning RL agent is used to perform the voltage control action. The Volt-VAR droop control algorithm, reward function, etc are designed using MATLAB coding.

CHAPTER FOUR: RESULT AND DISCUSSION

This chapter presents the results and discussion of the study.

4.1 Load Flow Analysis

A load flow program for IEEE 33 bus radial distribution system is developed in the MATLAB software and load flow analysis is done to validate the simulation result with the data provided by the IEEE Distribution System Analysis Subcommittee.

4.1.1 Effect of load variation without DG connection.

Table 4-1, shows the effect of the load variation without distributed generation on the node voltage. The node voltage is calculated on a per-unit scale (PU). The load on the feeder is decreased from 150 percent to 25 percent of the load defined by IEEE, which is listed in table 2-3. The results show that the node voltage decreases with the increase in the load. When the load is 25 percent and 50 percentage, the node voltage is within the limit. However, for load greater than 75 percent, the node voltage regulation criteria have been violated and the voltage is minimum in node 33. All the node voltages except in node 1 are less than one, while node voltage in node 1 is 1 PU.

Table 4-1: Effect of load variation without DG on Node Voltage (PU)

Percent Load / Node	Voltage of Node in PU					
	150	125	100	75	50	25
1	1	1	1	1	1	1
2	0.99538	0.99622	0.99703	0.9978	0.99856	0.99929
3	0.97335	0.97822	0.98289	0.98739	0.99172	0.99592
4	0.96154	0.96862	0.97538	0.98187	0.98812	0.99415
5	0.94984	0.95912	0.96796	0.97642	0.98456	0.99241
6	0.9207	0.93546	0.94948	0.96287	0.97571	0.98807
7	0.91511	0.93093	0.94595	0.96029	0.97403	0.98725
8	0.89347	0.91341	0.9323	0.9503	0.96752	0.98406
9	0.8834	0.90527	0.92597	0.94567	0.9645	0.98259
10	0.87406	0.89772	0.92009	0.94137	0.96171	0.98122
11	0.87267	0.8966	0.91922	0.94074	0.9613	0.98102
12	0.87026	0.89465	0.91771	0.93963	0.96058	0.98067
13	0.86041	0.8867	0.91153	0.93513	0.95765	0.97924
14	0.85675	0.88375	0.90924	0.93345	0.95656	0.97871
15	0.85448	0.88192	0.90782	0.93241	0.95589	0.97838
16	0.85227	0.88013	0.90643	0.93141	0.95523	0.97806

Percent Load / Node	Voltage of Node in PU					
	150	125	100	75	50	25
17	0.84899	0.8775	0.90439	0.92991	0.95426	0.97758
18	0.84801	0.8767	0.90377	0.92946	0.95397	0.97744
19	0.99459	0.99556	0.9965	0.99741	0.99829	0.99916
20	0.9892	0.99108	0.99292	0.99473	0.99651	0.99827
21	0.98813	0.99019	0.99222	0.9942	0.99616	0.99809
22	0.98717	0.9894	0.99158	0.99373	0.99584	0.99793
23	0.96788	0.97371	0.97931	0.98472	0.98996	0.99505
24	0.95771	0.9653	0.97264	0.97975	0.98667	0.99342
25	0.95263	0.96111	0.96931	0.97728	0.98504	0.9926
26	0.91765	0.93299	0.94755	0.96146	0.97479	0.98762
27	0.91361	0.9297	0.94499	0.95958	0.97356	0.98702
28	0.89555	0.91505	0.93354	0.95119	0.96809	0.98433
29	0.88257	0.90451	0.92532	0.94516	0.96416	0.98241
30	0.87694	0.89995	0.92177	0.94256	0.96245	0.98157
31	0.87036	0.89462	0.9176	0.93951	0.96047	0.9806
32	0.86891	0.89344	0.91669	0.93884	0.96003	0.98039
33	0.86846	0.89308	0.9164	0.93863	0.9599	0.98032

Table 4-2, shows the effect of the load variation without distributed generation on the line losses. With the increase in the load, the active power losses and reactive power losses both are increased. The load flow data i.e., voltage and line losses for IEEE defined load is the same as stated (Vita 2017).

Table 4-2: Effect of load variation without DG on line losses (kW)

S.N	Branch	Percentage of Load defined in IEEE							
		125		100		75		50	
		Ploss (kW)	Qloss (kVAR)	Ploss (kW)	Qloss (kVAR)	Ploss (kW)	Qloss (kVAR)	Ploss (kW)	Qloss (kVAR)
1	1- 2	19.87	10.13	12.30	6.27	6.71	3.42	2.90	1.48
2	2- 3	84.45	43.01	52.08	26.52	28.30	14.41	12.18	6.20
3	3- 4	32.80	16.70	20.05	10.21	10.81	5.51	4.62	2.35
4	4- 5	30.88	15.73	18.85	9.60	10.15	5.17	4.33	2.20
5	5- 6	63.22	54.58	38.57	33.29	20.75	17.91	8.84	7.63
6	6- 7	3.19	10.55	1.95	6.43	1.05	3.46	0.45	1.48
7	7- 8	19.54	14.10	11.87	8.57	6.37	4.59	2.71	1.95
8	8- 9	7.04	5.06	4.27	3.07	2.28	1.64	0.97	0.69
9	9-10	6.01	4.26	3.63	2.58	1.94	1.38	0.82	0.58
10	10-11	0.93	0.31	0.57	0.19	0.30	0.10	0.13	0.04
11	11-12	1.49	0.49	0.90	0.30	0.48	0.16	0.20	0.07

S.N	Branch	Percentage of Load defined in IEEE							
		125		100		75		50	
		Ploss (kW)	Qloss (kVAR)	Ploss (kW)	Qloss (kVAR)	Ploss (kW)	Qloss (kVAR)	Ploss (kW)	Qloss (kVAR)
12	12-13	4.51	3.55	2.72	2.14	1.45	1.14	0.61	0.48
13	13-14	1.23	1.62	0.74	0.98	0.40	0.52	0.17	0.22
14	14-15	0.60	0.54	0.36	0.32	0.19	0.17	0.08	0.07
15	15-16	0.48	0.35	0.29	0.21	0.15	0.11	0.07	0.05
16	16-17	0.43	0.57	0.26	0.34	0.14	0.18	0.06	0.08
17	17-18	0.09	0.07	0.05	0.04	0.03	0.02	0.01	0.01
18	2-19	0.25	0.24	0.16	0.15	0.09	0.09	0.04	0.04
19	19-20	1.31	1.18	0.83	0.75	0.47	0.42	0.21	0.19
20	20-21	0.16	0.19	0.10	0.12	0.06	0.07	0.03	0.03
21	21-22	0.07	0.09	0.04	0.06	0.02	0.03	0.01	0.01
22	3-23	5.05	3.45	3.18	2.17	1.76	1.21	0.77	0.53
23	23-24	8.17	6.45	5.14	4.06	2.85	2.25	1.25	0.99
24	24-25	2.05	1.60	1.29	1.01	0.71	0.56	0.31	0.24
25	6-26	4.26	2.17	2.60	1.33	1.40	0.71	0.60	0.30
26	26-27	5.46	2.78	3.33	1.70	1.79	0.91	0.76	0.39
27	27-28	18.55	16.35	11.31	9.97	6.08	5.36	2.59	2.28
28	28-29	12.86	11.20	7.84	6.83	4.21	3.67	1.79	1.56
29	29-30	6.40	3.26	3.90	1.99	2.09	1.07	0.89	0.45
30	30-31	2.62	2.59	1.59	1.58	0.86	0.85	0.36	0.36
31	31-32	0.35	0.41	0.21	0.25	0.11	0.13	0.05	0.06
32	32-33	0.02	0.03	0.01	0.02	0.01	0.01	0.00	0.01
Total line loss		344.3	233.61	211.00	143.04	113.99	77.22	48.79	33.03

4.1.2 Effect of load variation with DG connection.

The effect of the load variation with the DG connection has been established here. As suggested by (Abdel-Ghany, et al. 2015) DG of capacity 09 –1.0625 MW; Bus: 16 – 1.005 MW; Bus: 24 – 1.0447 MW; Bus: 30 – 0.9518 MW have been connected in the feeder. With the DG connection of optimal size, the voltage regulation of the line gets improved as shown in figure 4-1.

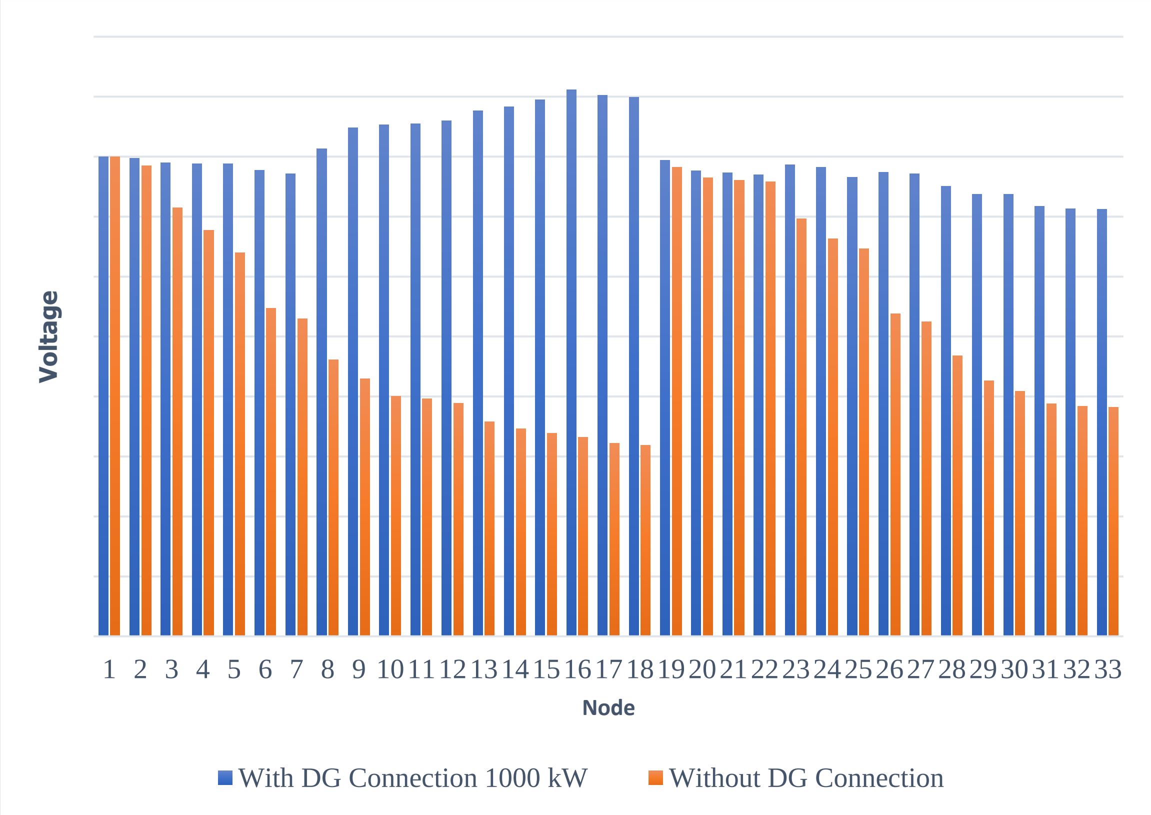


Figure 4-1: Effect of DG Connection on node voltage

Table 4-3 shows the effect of the load variation on node voltage while the DG's of the above-mentioned capacity are connected. The load has been decreased from 150 percent to 25 percent. The node voltage gets increased with the decrease in the load and all the node voltages are within the limit in all load cases.

Table 4-3: Effect of load variation with DG on node voltage

Percent. Load / Node	Percentage of Load defined in IEEE					
	150	125	100	75	50	25
1	1	1	1	1	1	1
2	0.9979	0.9987	0.9994	1.0001	1.0008	1.0015
3	0.9897	0.9939	0.998	1.002	1.0059	1.0097
4	0.9857	0.9918	0.9976	1.0034	1.0089	1.0144
5	0.9821	0.99	0.9977	1.0051	1.0123	1.0193
6	0.9709	0.9834	0.9954	1.007	1.0183	1.0292
7	0.9682	0.9815	0.9943	1.0067	1.0188	1.0304
8	0.9705	0.9869	1.0027	1.018	1.0328	1.0471
9	0.9747	0.9925	1.0096	1.0261	1.0421	1.0577
10	0.9731	0.9922	1.0106	1.0284	1.0456	1.0624

Percent. Load / Node	Percentage of Load defined in IEEE					
	150	125	100	75	50	25
11	0.9731	0.9924	1.011	1.029	1.0464	1.0633
12	0.9734	0.993	1.0119	1.0302	1.048	1.0651
13	0.974	0.9951	1.0153	1.0349	1.0539	1.0722
14	0.9743	0.9958	1.0166	1.0366	1.056	1.0748
15	0.9761	0.9979	1.0189	1.0392	1.0589	1.0779
16	0.9789	1.001	1.0222	1.0428	1.0627	1.082
17	0.976	0.9987	1.0204	1.0414	1.0618	1.0815
18	0.9752	0.998	1.0199	1.041	1.0615	1.0814
19	0.9972	0.998	0.9989	0.9997	1.0005	1.0013
20	0.9918	0.9936	0.9953	0.997	0.9987	1.0004
21	0.9907	0.9927	0.9946	0.9965	0.9984	1.0003
22	0.9898	0.9919	0.994	0.996	0.9981	1.0001
23	0.9872	0.9923	0.9973	1.0022	1.007	1.0116
24	0.983	0.9898	0.9964	1.0029	1.0093	1.0155
25	0.978	0.9857	0.9932	1.0005	1.0077	1.0147
26	0.9694	0.9824	0.9949	1.0069	1.0186	1.03
27	0.9675	0.9811	0.9942	1.0069	1.0192	1.0311
28	0.9576	0.9742	0.9901	1.0055	1.0204	1.0349
29	0.9509	0.9696	0.9875	1.0049	1.0216	1.0379
30	0.949	0.9686	0.9874	1.0056	1.0231	1.0402
31	0.943	0.9636	0.9835	1.0027	1.0213	1.0392
32	0.9416	0.9625	0.9827	1.0021	1.0208	1.039
33	0.9412	0.9622	0.9824	1.0019	1.0207	1.039

Table 4-4, shows the effect of the load variation on line losses with DG connection. With the decrease in the load from 125 percent to 100 percent, the active power losses and reactive power losses have been decreased. However, when the load further decreased from 100 percent to 75 percent and to 50 percent there is an increase in the active power losses and reactive power losses. If we look closer, during this period the branch losses have been increased in between branch 6-16; further, if we check the node voltage, the voltage is higher in node 16 and decreased towards node 7.

Table 4-4: Effect of load variation with DG on line loss

S.N	Branch	Power Loss							
		Active Power (kW)				Reactive Power (kVA)			
		50%	75%	100%	125%	50%	75%	100%	125%
1	1- 2	3.202	2.557	3.249	1.632	1.632	1.304	1.67	2.741
2	2- 3	19.32	14.645	15.458	9.843	9.843	7.459	7.942	11.327
3	3- 4	8.806	7.029	7.273	4.485	4.485	3.580	3.743	4.967
4	4- 5	9.519	7.519	7.388	4.848	4.848	3.829	3.801	4.753
5	5- 6	20.89	16.510	15.920	18.03	18.036	14.252	13.881	16.876
6	6- 7	2.321	1.716	1.272	7.672	7.672	5.673	4.204	3.316
7	7- 8	23.93	18.951	14.833	17.27	17.270	13.677	10.706	8.420
8	8- 9	16.18	13.722	11.516	11.62	11.625	9.858	8.275	6.891
9	9-10	2.885	2.026	1.463	2.045	2.045	1.436	1.038	0.873
10	10-11	0.586	0.431	0.320	0.194	0.194	0.143	0.106	0.085
11	11-12	1.177	0.889	0.667	0.389	0.389	0.294	0.221	0.172
12	12-13	4.953	3.894	3.021	3.897	3.897	3.064	2.377	1.854
13	13-14	1.959	1.605	1.296	2.579	2.579	2.112	1.706	1.367
14	14-15	2.449	2.171	1.907	2.179	2.179	1.932	1.697	1.477
15	15-16	3.309	3.054	2.807	2.416	2.416	2.230	2.051	1.878
16	16-17	0.047	0.109	0.202	0.062	0.062	0.145	0.269	0.439
17	17-18	0.010	0.023	0.043	0.008	0.008	0.018	0.033	0.054
18	2-19	0.040	0.090	0.160	0.038	0.038	0.086	0.153	0.240
19	19-20	0.205	0.464	0.828	0.185	0.185	0.418	0.746	1.171
20	20-21	0.025	0.056	0.100	0.029	0.029	0.066	0.117	0.184
21	21-22	0.011	0.024	0.043	0.014	0.014	0.032	0.057	0.090
22	3-23	0.930	0.575	0.591	0.636	0.636	0.393	0.404	0.678
23	23-24	2.069	1.263	1.049	1.634	1.634	0.997	0.829	1.148
24	24-25	0.298	0.680	1.226	0.233	0.233	0.532	0.96	1.522
25	6-26	0.616	0.743	1.166	0.314	0.314	0.379	0.607	0.978
26	26-27	0.906	1.057	1.586	0.461	0.461	0.538	0.816	1.291
27	27-28	3.552	4.032	5.787	3.131	3.131	3.555	5.103	7.909
28	28-29	2.830	3.148	4.347	2.465	2.465	2.742	3.788	5.693
29	29-30	1.935	2.052	2.604	0.986	0.986	1.045	1.327	1.856
30	30-31	0.321	0.750	1.387	0.318	0.318	0.742	1.371	2.232
31	31-32	0.043	0.100	0.186	0.050	0.050	0.117	0.216	0.352
32	32-33	0.003	0.006	0.011	0.004	0.004	0.010	0.018	0.029
Total line loss		135.3	111.89	109.70	130.89	99.678	135.3	111.89	109.70

When the load on the feeder decreases from 100 percent to the 75 percent, there is an increase in the loss which is the opposite in the case without DG. This is because of an increase in the reverse current flow in the section between 16 to 8. If we see the load profile, we can see that the voltage at these nodes are greater than 1 and also voltage magnitude, as well as angle increases, from node 8 to node 16, which suggests the reverse power flow. Also, the voltage magnitude difference is greater while the load percentage is 75 percent than in 100 percent.

4.1.3 Effect of DG output variation with constant load

Table 4-5, shows the effect of the DG generation on the node voltage. The DG generation is varied from 100 percent (full capacity) to zero (no generation). With the DG generation the node voltage becomes greater than one in node 8-11, which shows that in that section, the load demand is less than the DG generation. With the decrease in the DG generation, the node voltages have been decreased. When the generation is at 25 percent, the node voltage is violated in node 9-18 and 28-33, and the node voltage further decrease with a decrease in a generation.

Table 4-5 : Effect of DG output variation on node voltage.

Percent Capacity / Node	Node Voltage in PU				
	100	75	50	25	0
1	1	1	1	1	1
2	0.9994	0.99885	0.99828	0.99767	0.99703
3	0.99802	0.99451	0.99084	0.98697	0.98289
4	0.99765	0.99251	0.98711	0.98142	0.97538
5	0.99767	0.99082	0.98363	0.97603	0.96796
6	0.9954	0.98487	0.97378	0.96202	0.94948
7	0.99434	0.9833	0.97163	0.95923	0.94595
8	1.0027	0.98665	0.96967	0.95162	0.9323
9	1.00957	0.99048	0.97031	0.94889	0.92597
10	1.01061	0.98996	0.96814	0.94494	0.92009
11	1.01103	0.99009	0.96795	0.94442	0.91922
12	1.01195	0.99045	0.96773	0.94358	0.91771
13	1.01534	0.99168	0.96666	0.94004	0.91153
14	1.01658	0.99212	0.96625	0.93873	0.90924
15	1.01891	0.99359	0.96682	0.93834	0.90782
16	1.02224	0.99584	0.96793	0.93824	0.90643
17	1.02042	0.99397	0.96601	0.93626	0.90439
18	1.01988	0.99342	0.96543	0.93567	0.90377
19	0.99888	0.99833	0.99775	0.99714	0.9965
20	0.99531	0.99475	0.99417	0.99356	0.99292
21	0.9946	0.99405	0.99347	0.99286	0.99221
22	0.99397	0.99341	0.99283	0.99222	0.99158
23	0.99734	0.99312	0.98872	0.98413	0.97931

Node \ Percent Capacity	Node Voltage in PU				
	100	75	50	25	0
24	0.99644	0.99081	0.98498	0.97893	0.97264
25	0.9932	0.98755	0.9817	0.97563	0.96931
26	0.99485	0.984	0.97257	0.96045	0.94755
27	0.99423	0.98292	0.97101	0.9584	0.94499
28	0.99014	0.9771	0.9634	0.94892	0.93354
29	0.98752	0.97316	0.95809	0.94219	0.92532
30	0.9874	0.97223	0.95631	0.93954	0.92177
31	0.98352	0.96828	0.9523	0.93546	0.9176
32	0.98266	0.96742	0.95142	0.93456	0.91669
33	0.9824	0.96715	0.95114	0.93428	0.9164

Table 4-6, shows the effect of the DG generation on the line losses. The DG generation is varied from 100 percent (full capacity) to zero (no generation). The line losses has been an increase with the decrease in the DG generation.

Table 4-6 : Effect of DG output variation on line loss

S.N	Branch	DG Output (Percentage of Full Capacity)							
		100		75		50		25	
		P (kW)	Q (kVAR)	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)
1	1- 2	3.28	1.67	3.56	1.81	5.05	2.58	7.91	4.03
2	2- 3	15.59	7.94	14.37	7.32	19.51	9.94	31.76	16.18
3	3- 4	7.35	3.74	6.17	3.14	7.62	3.88	12.09	6.16
4	4- 5	7.46	3.80	5.82	2.96	6.89	3.51	11.07	5.64
5	5- 6	16.08	13.88	12.10	10.45	13.94	12.03	22.40	19.34
6	6- 7	1.27	4.20	0.53	1.76	0.34	1.11	0.77	2.54
7	7- 8	14.84	10.71	6.06	4.37	2.10	1.52	3.71	2.68
8	8- 9	11.52	8.28	5.05	3.63	1.37	0.99	0.93	0.67
9	9-10	1.46	1.04	0.69	0.49	0.69	0.49	1.60	1.14
10	10-11	0.32	0.11	0.14	0.05	0.10	0.03	0.23	0.08
11	11-12	0.67	0.22	0.27	0.09	0.15	0.05	0.33	0.11
12	12-13	3.02	2.38	1.22	0.96	0.44	0.35	0.86	0.67
13	13-14	1.30	1.71	0.54	0.71	0.15	0.20	0.19	0.25
14	14-15	1.91	1.70	0.89	0.79	0.24	0.21	0.04	0.03
15	15-16	2.81	2.05	1.40	1.02	0.45	0.33	0.04	0.03
16	16-17	0.20	0.27	0.21	0.28	0.23	0.30	0.24	0.32
17	17-18	0.04	0.03	0.05	0.04	0.05	0.04	0.05	0.04
18	2-19	0.16	0.15	0.16	0.15	0.16	0.15	0.16	0.15
19	19-20	0.83	0.75	0.83	0.75	0.83	0.75	0.83	0.75
20	20-21	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12

		DG Output (Percentage of Full Capacity)							
		100		75		50		25	
S.N	Branch	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)
21	21-22	0.04	0.06	0.04	0.06	0.04	0.06	0.04	0.06
22	3-23	0.59	0.40	0.68	0.46	1.13	0.77	1.96	1.34
23	23-24	1.05	0.83	0.97	0.76	1.60	1.26	2.98	2.36
24	24-25	1.23	0.96	1.24	0.97	1.26	0.98	1.27	0.99
25	6-26	1.19	0.61	1.26	0.64	1.50	0.76	1.94	0.99
26	26-27	1.60	0.82	1.64	0.83	1.92	0.98	2.47	1.26
27	27-28	5.79	5.10	5.71	5.03	6.53	5.76	8.36	7.37
28	28-29	4.35	3.79	4.13	3.60	4.59	4.00	5.79	5.05
29	29-30	2.60	1.33	2.26	1.15	2.33	1.19	2.86	1.46
30	30-31	1.39	1.37	1.43	1.42	1.48	1.46	1.53	1.52
31	31-32	0.19	0.22	0.19	0.22	0.20	0.23	0.21	0.24
32	32-33	0.01	0.02	0.01	0.02	0.01	0.02	0.01	0.02
Total		110.23	80.23	79.70	56.05	82.99	56.03	124.74	83.55

4.2 Training of DDPG Agent

A DDPG based agent is made and simulated in a MATLAB environment. The actor/critic network was designed in MATLAB. A suitable reward function is designed using MATLAB Simulink user-defined function. The MATLAB code for the training of DDPG agent in user-defined environment, reward function are attached in the appendix below.

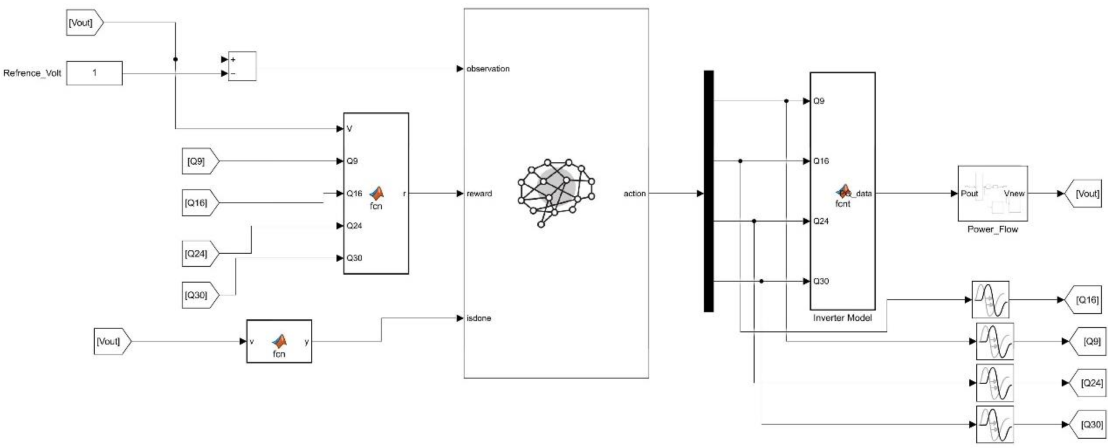


Figure 4-2 : A Model for Voltage control using RL

4.2.1 Actor Network

The actor network consists of an input layer of thirty-three nodal voltage, two fully connected layers and two reluLayer layers, which is an activation function. The reluLayer, also called Rectifier, outputs the positive value and the negative value will be returned as zero. The output of the actor network is action, i.e., reactive power output of the smart inverter.

4.2.2 Critic Network

The critic network consists of a two paths, one for the action output from the actor-network and another layer for the observation input. The action path consists of a input layer and a fully connected layer. The observation path consists of an input layer, fully connected layer, relu layer and a fully connected layer. The common path consists of a common relu layer and an output layer

4.2.3 Training Data

- Sample time (Ts) = 1.0
- Simulation Length (Tf) = 200
- Critic LearningRate = 1e-02
- Actor Learning Rate = 1e-03
- Discount factor (Gamma) = 0.90
- Batch Size = 128
- Target smooth factor τ = 1e-03

4.2.3.1 Exploration Parameters

In DDPG algorithm, the policy is deterministic. The agent might not try a wide variety of actions to determine the useful learning signal. To achieve a good policy for DDPG agents, we add noise to the action during the training process. Following are the noise parameter

- DDPG Variance Decay Rate = 1e-5; To get a better quality for training, the noise variance should be reduced as the process goes. It is the general practice to take value which reduce the variance to half percentage if 1e-5 Half-life of 1,000 episodes. Half life = log(0.5)/log(1-StandardDeviationDecayRate); For training of the maximum 200 episode; decay rate will be 3.5e-5.

- Noise Variance = [100;100;100;100]; The variance should be selected such that, Variance *sqrt(sample time) must be between 1 & 10 % of action range.

4.2.4 Training process

The DDPG object is trained for the total 200 episode. During the start of the each episode, the critic network will calculate Q0, which is the measure of the how well the object was trained. If the agent is well trained then the value of Q0 should overlap with the episode reward. As shown in figure below, towards the end of the training episode, the Q0 value nearly coincide with the episode reward, which shows that the object is well trained. Since, the agent has used the pretrained agent data, the training converges in few episode. The training process is shown in figure 4-3.

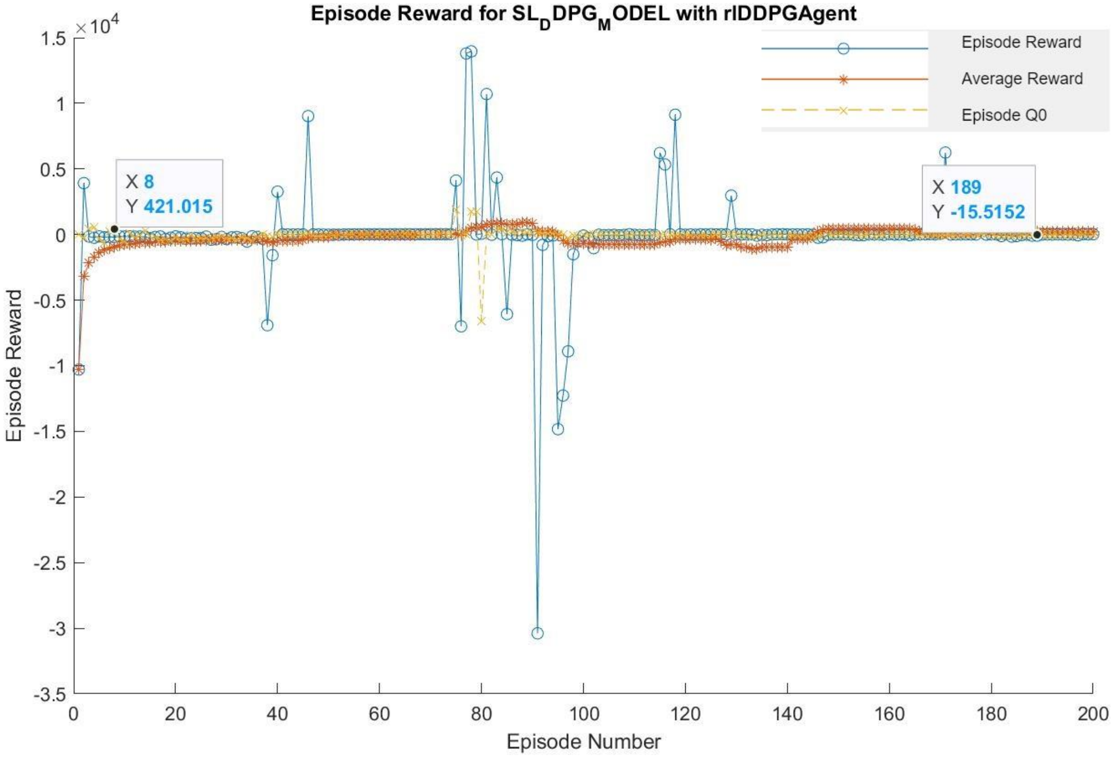


Figure 4-3 : A Model for Voltage control using RL

4.3 Simulation for voltage control

4.3.1 Reward Design

The reward signal consist of two part. First part is penalty for voltage limit violation (r1) is assigned as shown in figure below and the

Table 4-7 : Penalty for voltage violation

Node Voltage	Reward (r1)	Node Voltage	Reward (r1)	Node Voltage	Reward (r1)
0.1	-200	0.85	-6.04475	1.6	-78.2143
0.15	-166.667	0.9	-0.48919	1.65	-90.7143

Node Voltage	Reward (r1)	Node Voltage	Reward (r1)	Node Voltage	Reward (r1)
0.2	-141.667	0.95	0	1.7	-105
0.25	-121.667	1	0	1.75	-121.667
0.3	-105	1.05	0	1.8	-141.667
0.35	-90.7143	1.1	-0.48919	1.85	-166.667
0.4	-78.2143	1.15	-6.04475	1.9	-200
0.45	-67.1032	1.2	-11.9271		
0.5	-57.1032	1.25	-18.1771		
0.55	-48.0123	1.3	-24.8438		
0.6	-39.6789	1.35	-31.9866		
0.65	-31.9866	1.4	-39.6789		
0.7	-24.8438	1.45	-48.0123		
0.75	-18.1771	1.5	-57.1032		
0.8	-11.9271	1.55	-67.1032		

A polynomial regression is used to represent those data. Higher the degree of the polynomial it represents the more data which is shown by R-square value. Regression analysis has been done to derive the polynomial equation that best fits the penalty for the voltage violation. The curve was fit in MATLAB using and the result of the fitting is as follows:

Polynomial of 2nd Degree

Coefficients (with 95% confidence bounds):

p1 = -227.9 (-234.8, -221)

p2 = 455.8 (441.6, 470)

p3 = -228.3 (-234.5, -222.1)

Goodness of fit: SSE: 937.4; R-square: 0.9925; Adjusted R-square: 0.9921; RMSE: 5.251; DFE=34

These data show that the line was a general fit, so that it is not required to go for the higher degree of a polynomial. Fitting a higher degree of the polynomial case the overfitting (ie, fitting of noise).

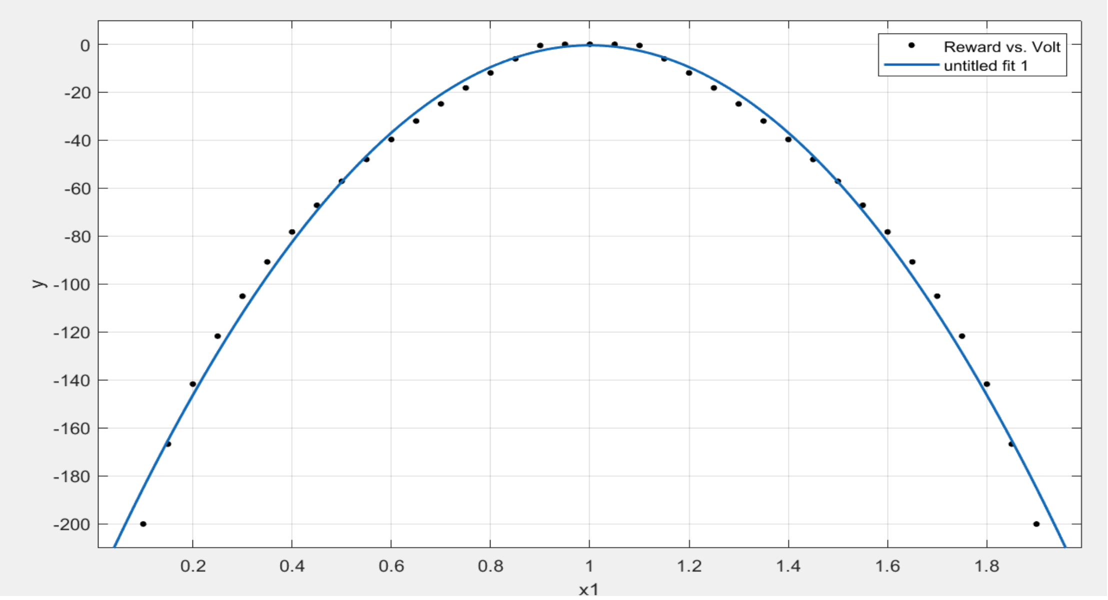


Figure 4-4 : DDPG Algorithm

In order to prevent the overfitting of the cure, here second degree of polynomial is chosen. The equation is $y = -227.9x^2 + 455.8x - 228.3$

The second part of the reward (r2) is for the reduction of active power curtailment.
r2= (1-abs(Q/S)); Q is the reactive power generation and S is the inverter capacity.
The total reward (r) = r1+r2

4.3.2 Voltage Control

To perform the simulation for the control of voltage of modified IEEE-33 bus, the SI of capacity 2000 kVA, with AC/DC ratio of 1.2 is considered. That means the PV active power output capacity is 1667.67 kW. The reactive power generation capacity of the inverter is also supposed 1666.67 kVAR. For this inverter, if the reactive power 1666.67 kVAR is generated by SI, the active power will be curtailed to 1105 kW.

To check the performance of the DDPG agent for node voltage control, the load on the system was generated as shown in figure 4-5. A similar load curve as NEA load curve was generated in MATLAB and fed into the RL simulation block.

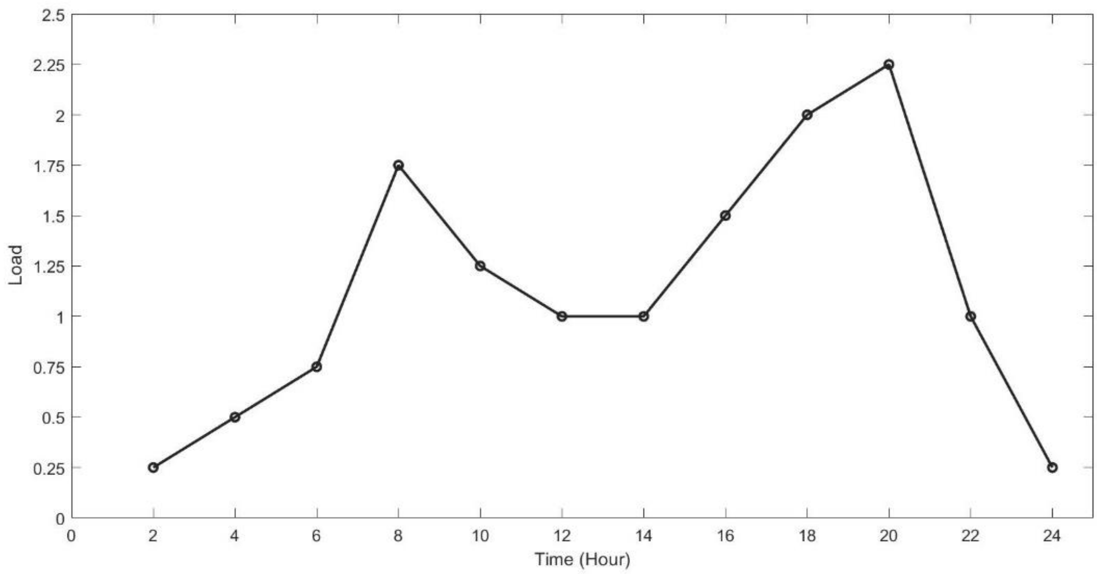


Figure 4-5 : Load Variation

When the load is at 225 percent i.e., at hour 20, the node voltage was below 0.95 PU in node 28, 29, 30, 31, 32, 33. With the reinforcement learning control, the voltage of those nodes are within the limit. When the load is 175 percent, without RL controller, the node voltage is within the limit, however, the use of the RL controller enhanced the node voltage.

Table 4-8 : Node Voltage

	Node Voltage (PU)									
	Without RL	With RL	Without RL	Without RL	Without RL	With RL	Without RL	With RL	Without RL	With RL
Node	225.00%		175.00%		125.00%		75.00%	75.00%	25.00%	25.00%
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	0.997	0.998	0.999	0.999	1.000	1.000	0.999	1.000	1.003	1.001
3	0.984	0.990	0.995	0.995	1.003	0.999	0.995	1.004	1.018	1.005
4	0.978	0.986	0.993	0.993	1.005	0.999	0.993	1.005	1.026	1.007
5	0.974	0.984	0.992	0.992	1.007	1.000	0.992	1.007	1.035	1.009
6	0.959	0.979	0.987	0.986	1.010	0.997	0.987	1.006	1.053	1.006
7	0.955	0.981	0.984	0.984	1.009	0.990	0.984	1.000	1.055	0.997
8	0.961	0.994	0.997	0.998	1.027	0.998	0.997	1.011	1.083	1.001
9	0.970	1.006	1.008	1.009	1.041	1.005	1.008	1.019	1.101	1.005
10	0.968	1.005	1.009	1.011	1.044	1.004	1.009	1.018	1.109	1.004
11	0.969	1.005	1.009	1.011	1.045	1.004	1.009	1.019	1.110	1.005
12	0.969	1.006	1.011	1.013	1.047	1.005	1.011	1.020	1.113	1.006
13	0.972	1.008	1.016	1.019	1.055	1.006	1.016	1.020	1.125	1.005
14	0.973	1.008	1.018	1.021	1.057	1.004	1.018	1.018	1.130	1.003
15	0.976	1.011	1.022	1.025	1.062	1.005	1.022	1.019	1.135	1.003
16	0.981	1.016	1.027	1.031	1.067	1.008	1.027	1.021	1.141	1.004
17	0.977	1.012	1.024	1.028	1.065	1.006	1.024	1.019	1.141	1.003
18	0.975	1.011	1.023	1.027	1.065	1.005	1.023	1.019	1.141	1.003
19	0.996	0.997	0.998	0.998	0.999	0.999	0.998	1.000	1.003	1.000
20	0.987	0.988	0.991	0.991	0.995	0.994	0.991	0.997	1.002	0.999
21	0.986	0.987	0.990	0.990	0.994	0.994	0.990	0.997	1.002	0.998
22	0.984	0.985	0.989	0.989	0.993	0.993	0.989	0.996	1.001	0.998
23	0.978	0.987	0.993	0.994	1.003	0.999	0.993	1.006	1.022	1.008
24	0.969	0.981	0.991	0.992	1.004	1.000	0.991	1.010	1.029	1.013
25	0.962	0.974	0.986	0.986	1.000	0.995	0.986	1.007	1.028	1.011
26	0.957	0.978	0.985	0.985	1.010	0.996	0.985	1.006	1.054	1.007
27	0.954	0.976	0.984	0.983	1.010	0.996	0.984	1.006	1.057	1.008
28	0.940	0.969	0.976	0.974	1.007	0.995	0.976	1.003	1.064	1.009
29	0.930	0.964	0.971	0.968	1.006	0.994	0.971	1.002	1.070	1.010
30	0.928	0.963	0.970	0.967	1.007	0.995	0.970	1.003	1.074	1.013
31	0.919	0.954	0.964	0.960	1.003	0.990	0.964	1.000	1.074	1.011
32	0.917	0.952	0.962	0.959	1.002	0.989	0.962	1.000	1.073	1.010
33	0.916	0.951	0.962	0.958	1.001	0.989	0.962	0.999	1.073	1.010

In order to regulate the voltage in the distribution feeder, the SI’s has generated/absorved the reactive power, which is shown in Table 4-9. When the load is high, the SI’s has generated reactive power. At 225 percent of the load, the total reactive power generated from four SI’s was 3029.25 kVAR. This value get decreased as the load decrese. When the load is 125 perecent, the SI start to absorb the reactive power from network, at 125 percent the total load absorbed is -1377.49 kVAR and this value further increses as the load decresed. At the 25 percent of the load the total reactive power absorbed is -3752 kVAR.

Table 4-9 : Reactive Power Output from SI

SI at Node	Percentage of Load				
	225%	175%	125%	75%	25%
9	1667.67	-21.09	-269.32	-204.12	-1211.51
16	-247.63	127.21	-999.01	-1234.7	-1405.61
24	-60.71	94.11	-207.41	-110	-255.31
30	1667.67	-178.54	97	-925.93	-880.53
Total Reactive Power generation	3029.25	23.44	-1377.49	-2473.97	-3752.71

The reactive power generation / absorption by the SI is shown in figure 4-6.

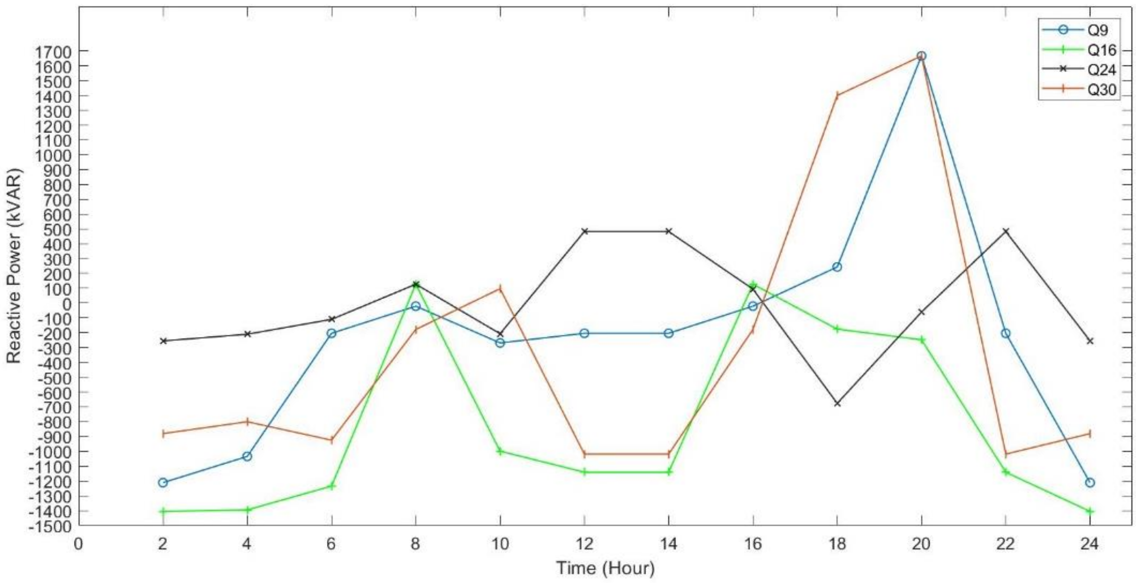


Figure 4-6 : Reactive Power output of SI's

4.3.3 Reduction of active power curtailment

If the system voltage increases, the traditional PV source reduces the power generation to bring the voltage within the limit. Since the AC-DC ratio of the traditional converter is fixed, the SI will reduce the active power generation (Tonkoski and Lopes 2011). Instead of reducing the active power, the SI can absorb the reactive power resulting in the active power curtailment. To verify the reduction in active power curtailment by the SI's, the active and reactive power generation by SI using RL controller, under different load conditions was compared with the active and reactive power output using Volt-VAR droop control method. The following case is considered

The load in the feeder is considered 50 percent and the SI maximum rating is considered as 2000 kVA and the AC-DC ratio of the inverter is considered as 1. Without the control algorithm, the system voltage exceeds the upper limit. Under this criterion, the active power, and reactive power form SI's using Volt-VAR control algorithm and reinforcement learning control algorithm is as follows.

Table 4-10: Effect of DG output variation on node voltage & loss

	SI	9	16	24	30
Volt-VAR	Reactive Power (kVAR)	-1066.49	-1519.09	-295.146	-748.662
	Active Power (kW)	1691.922	1300.913	1978.102	1854.59
RL Controller	Reactive Power (kVAR)	-1050.13	-1391.98	-186.239	-733.531
	Active Power (kW)	1702.124	1436.105	1991.31	1860.627

Table 4-10 shows that the active curtailment is less in the reinforcement learning controller than in the Volt-VAR droop control method. The total active power generation in Volt-Var control method is 6825.527 Watt and 6990.166 Watt in reinforcement learning-based control algorithm, which is 2.412 percentage higher.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

5.1 Conclusion

- The designed load flow program has been tested in the IEEE 33-bus radial feeder system defined by IEEE. The node voltage and line losses calculated from the designed load flow without DG connection at the defined load by IEEE are, active power loss 210 kW and reactive power loss 143 kVAR which is same as mentioned in the IEEE feeder datasheet. The load flow result shows that, when the load is high as compared to the generation, the system line losses increased with the increase in load. With the DG connection, the active power and reactive power loss at 75 percent load is 111.9 kW and 79.9 kVAR where as when the load is reducred to 50 percent the active and reactive losses are increased to 135.3 kW and 82 kW, which shows that with high penetration of DG, during the lightly loaded condition, the line loss in the feeder may increase with the decrease in the load due to the increase in the reverse current flow. With the DG The node voltage gets increased during light loading of line and high penetration of the DG and vice-versa.

- The DDPG agent gets trained in the created environment with feeder node voltage as observation signal and SI reactive power generation as an action of the DDPG agent. The training of the DDPG agent using the pre-trained model shows that the use of the pre-trained data makes the training process fast and effective. During the training the agent performs, the coordination between the SI's resulted in better simulation output.

- The trained DDPG agent was used for simulation for voltage control of the IEEE-33 radial distribution bus. During the load variation, the DDPG agent keeps the voltage within the preset limit. The active power curtailment using reinforcement learning controller is 2.41 percent less than as compared to Volt-Var droop control method.

5.2 Recommendation

- This study has considered a single reinforcement learning agent for the voltage controller design. The use of multiple agents may make the coordination between DG more effective.

- Transient analysis and stability analysis are not included in this study therefore, it is recommended to perform those studies.

- This study is not carried out on real distribution feeders. So, it is suggested to carry out the study in real distribution feeder and also recommended to perform the financial analysis.

REFERENCES

[1] Abdel-Ghany, Hossam A., Ahmed M. Azmy, Nagy I. Elkalashy, and Essam M. Rashad. "Optimizing DG penetration in distribution networks concerning protection schemes and technical impact." *Electric Power Systems Research* 128 (2015): 113-112.

[2] Bhimarasetti, Ravi Teja, Kumar, and Ashwani. "A New Contribution to Distribution Load Flow Analysis for Radial and Mesh Distribution Systems." *IEEE.* 2014.

[3] Choudhary, Niraj Kumar, Soumya Ranjan Mohanty, and Ravindra Kumar Singh. "Coordination of Overcurrent Relay in Distributed System for Different Network Configuration." *Journal of Power and Energy Engineering*, 2015: 1-9.

[4] Duong, Minh Quan, Thai Dinh Pham, Thang Trung Nguyen, and Anh Tuan Doan. "Determination of Optimal Location and Sizing of Solar Photovoltaic Distribution Generation Units in Radial Distribution Systems." *energies* (energies), 2018.

[5] Duong, Minh Quan, Thai Dinh Pham, Thang Trung Nguyen, and Anh Tuan Doan. "Determination of Optimal Location and Sizing of Solar Photovoltaic Distribution Generation Units in Radial Distribution Systems." *Energies*, 2018.

[6] Hojabri, M., U. Dersch, A. Papaemmanouil, and P. Bosshart. "A Comprehensive Survey on Phasor Measurement Unit Applications in." *Energies* (Energies), 2019.

[7] IEEE. "Ieee standard for interconnection and interoperability of distributed energy resources." 2018.

[8] Jacome, Veronica, Noah Klugman, Catherine Wolfram, Belinda Grunfeld, Duncan Callaway, and Isha Raya. "Power quality and modern energy for all." (PNAS) 2019.

[9] Júnior, Celso Fabrício Melo, Aline F. Cordeiro Waenga, and Dayana A. Ferreira Pinto. "Effects of the Photovoltaic Distributed Generation on Electricity Distribution System Voltage – Updated Review." (BRAZILIAN ARCHIVES OF BIOLOGY AND TECHNOLOGY) 61 (2018).

[10] Klonaria, Vasiliki, Jean-François Toubeaua, ZacharieDe Grèvea, and Olgan Durieuxb. "Probabilistic simulation framework for balanced and unbalanced low voltage networks." In *International Journal of Electrical Power & Energy Systems*, 439-451. Elsevier, 2016.

[11] Kowsalya. *CIRCUIT HUB.* 2018. https://circuitglobe.com/methods-of-voltage-control-in-power-system.html (accessed January 12, 2022).

[12] Lee, Hyeong-Jin, Kwang-Hoon Yoon, Joong-Woo Shin, Jae-Chul Kim, and Sung-Min Cho. "Optimal Parameters of Volt–Var Function in Smart Inverters for Improving System Performance." *Energies*, 2020.

[13] Li, Changfu, Chenrui Jin, and Ratnesh Sharma. "Coordination of PV Smart Inverters Using Deep Reinforcement Learning for Grid Voltage Regulation." *IEEE International Conference on Machine Learning and Applications.* Florida, USA: IEEE, 2019.

[14] Lusis, Peter, Lachlan L. H. Andrew, and Ariel Liebman. "Interaction Between Coordinated and Droop Control PV Inverters." (Arxiv) 2020.

[15] MathWorks. *MATLAB.* 2018. http://www.mathworks.com (accessed August 7, 2018).

[16] Matlab. *Reinforcement learning user guide.* 2021.

[17] Minh Quan Duong, Thai Dinh Pham, Thang Trung Nguyen , Anh Tuan Doan ,. "Determination of Optimal Location and Sizing of Solar Photovoltaic Distribution Generation Units in Radial Distribution Systems." *energies*, 2018.

[18] National Planning Commission; NEA Engineering Company. *Universalizing Clean Energy in Nepal.* Kathmandu: GoN, National Planning Commission, 2018.

[19] Singh, Manohar. "Protection coordination in distribution systems with and without distributed energy resources- a review." *Protection and Control of Modern Power Systems* 2, no. 27 (2017): 17.

[20] Sivkumar, Mishra, and Debapriya Das. "Distribution System Load Flow Methods: A Review." (ResearchGate) 2008.

[21] Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction.* Vol. 2. The MIT Press, 2015.

[22] The Institute of Electrical and Electronics Engineers, Inc. *IEEE 13 Node Test Feeder.* IEEE Power Engineering Society, n.d.

[23] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra. "Continuous control with deep reinforcement learning." *arXiv:1509.02971*, 2015.

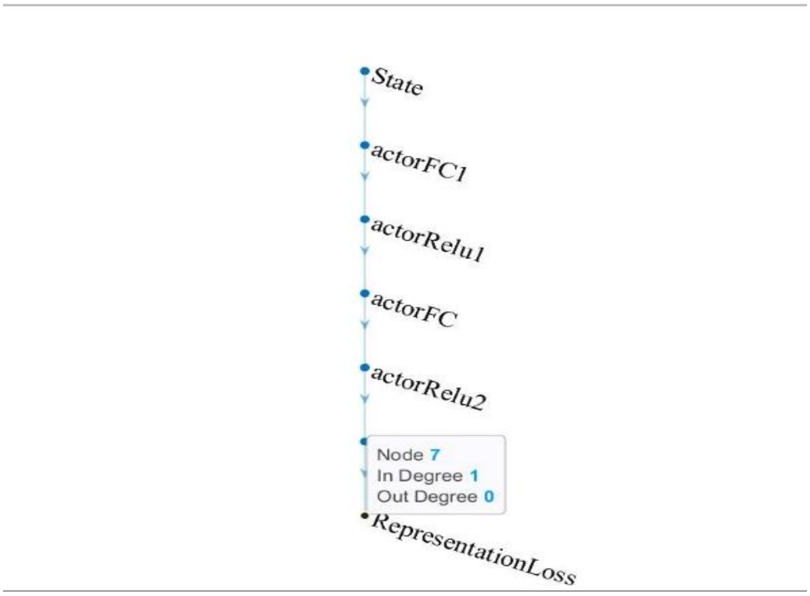
[24] Tonkoski, Reinaldo, and Luiz A.C. Lopes. "Impact of active power curtailment on overvoltage prevention and energy production of PV inverters connected to low voltage residential feeders." *Elsevier*, 2011: 3566-3574.

[25] Vita, Vasiliki. "Development of a Decision-Making Algorithm for the Optimum Size and Placement of Distributed Generation Units in Distribution Networks." *Energies*, 2017.

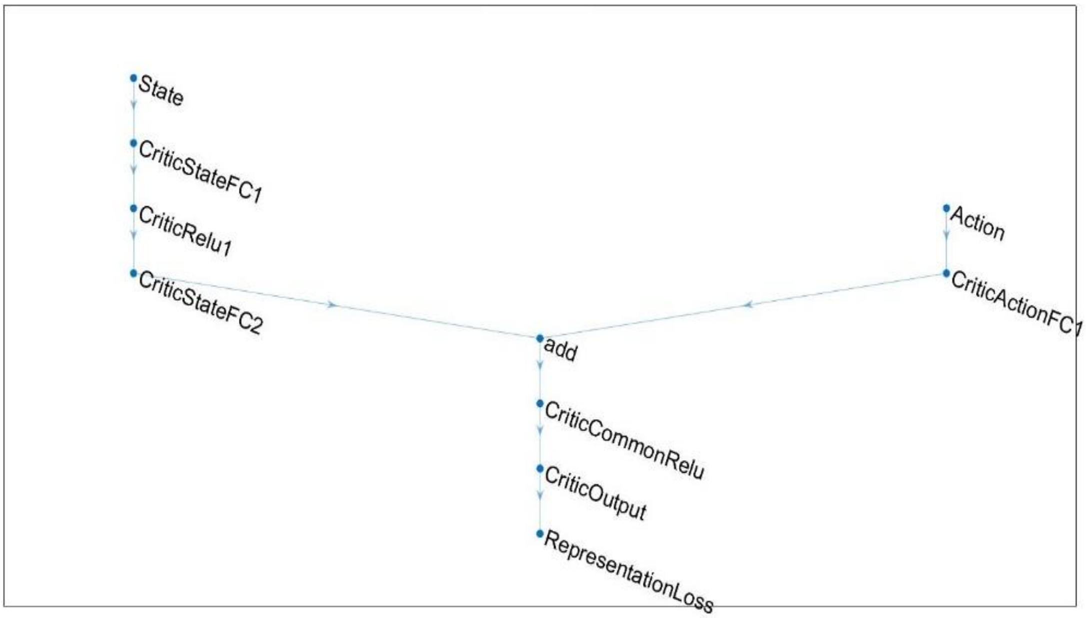
APPENDICES

Appendix-A.: Simulation Output

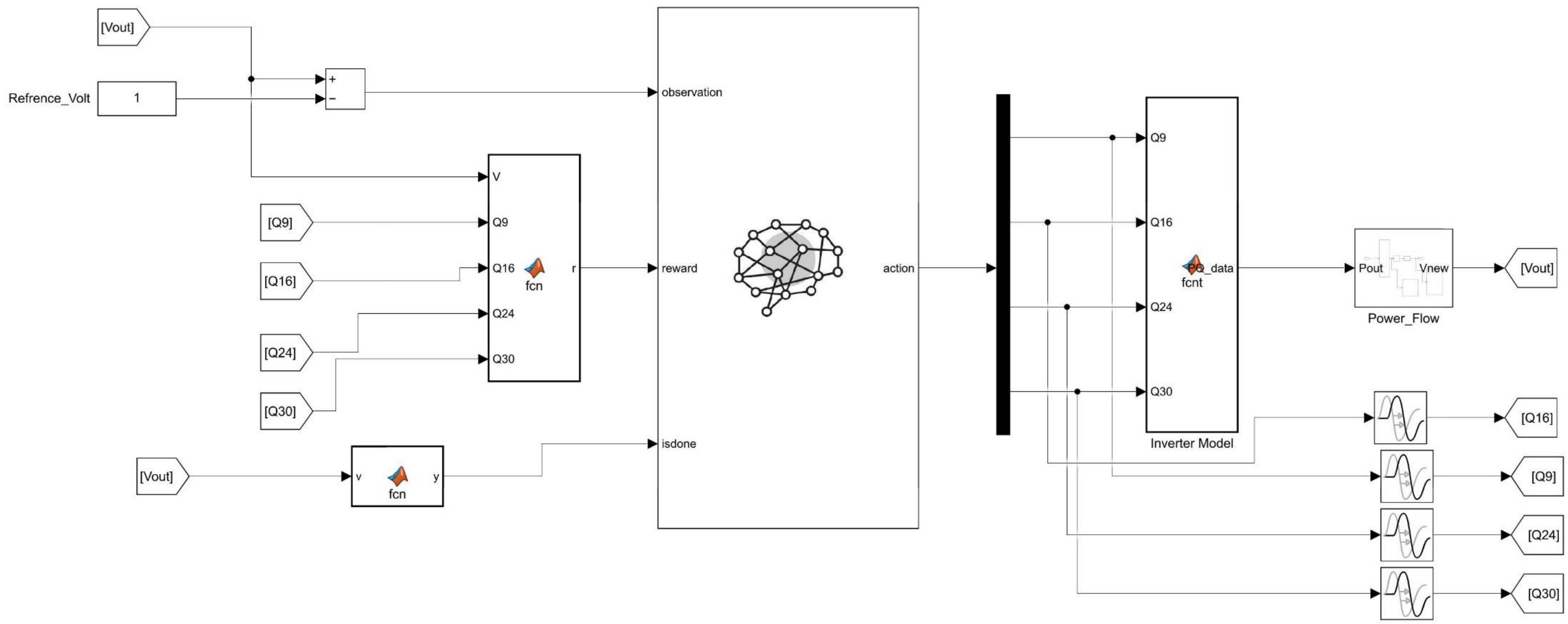
Appendix-A.1: Actor-Network



Appendix-A.2: Actor-Network



Appendix-A.3: Simulation Environment



Appendix-B. MATLAB Source Code

Appendix-B.1: DDPG Training

```
%%DDPG training
%% Voltage Control using Reinforcement Learning
clc;
clear;
 %%Set initial Variable
Pout=0;
Qout=0;
Vin= fcn(Pout,Qout);
%% Model Path and Version
PATH_MODEL = 'results/';
VOLTAGE_CONTROL_MODEL = 'SL_DDPG_MODEL'; % Simulink DDPG
Training Circuit
open_system(VOLTAGE_CONTROL_MODEL);
%%%% Training start
start_time= datetime();
today_date = datetime();
%% Set training parameters
AGENT_SAVE_THRESHOLD = 0;     %% Save an agent if the reward of the
episode is greater than 0
TRAINING_STOP = 70000;          %% Stop training at maximum reward of given
value here
REWARD_MAX = TRAINING_STOP;     % Stop model training at this avg. reward
 %% MODEL parameters
% Epsiode and time related
MAX_EP = 1000;
Ts = 1.0;   % Ts: Sample time (secs)
Tf = 200;   % Tf: Simulation length (secs)
 AVERAG_WINDOW = 50;       % Average over 50 time-steps
% DDPG trainng parameter for actor and ciritic
Critic_Learn_Rate = 1e-03; %%%  less than the actor
```

```
Actor_Learn_Rate  = 1e-04; %%% because of large number of parameter in critic network
GAMMA = 0.90; %%% Discount rate
BATCH = 128; %%% Batch size from buffer
 RL_AGENT = [VOLTAGE_CONTROL_MODEL '/RL Agent'];
%%% RL_MODEL_FILE = strcat(PATH_MODEL, 'RL_Model_', VERSION, '.mat');
 % Observation
%  (1) V(k), node voltage of 33 nos of node
 obsInfo = rlNumericSpec([33 1],'LowerLimit', zeros(33,1),'UpperLimit',2*ones(33,1));
obsInfo.Name = 'observations'; %% Name
obsInfo.Description = 'controlled_volt';
numObservations = obsInfo.Dimension(1);
 actInfo = rlNumericSpec([4 1],'LowerLimit',[-1667,- 1667,- 1667,- 1667]','UpperLimit',[ 1667, 1667, 1667, 1667]'); %%% Define the RL agent action limits
actInfo.Name = 'RectPower'; %%% Action Name
numActions = actInfo.Dimension(1); %%% Action Dimension
 %% Creating Enviroment
 env = rlSimulinkEnv(VOLTAGE_CONTROL_MODEL,RL_AGENT, obsInfo, actInfo);
 %% Reset function, To inisilize the observation value at the start of each episode
env.ResetFcn = @(in)localResetFcn(in, VOLTAGE_CONTROL_MODEL);
rng(0);
% Critic network
 % State Path is also called observation path
% Neurons of Critic network: OP-Observation Path, AP-ACtion Path
statePath = [
    featureInputLayer(33, 'Normalization', 'none', 'Name', 'Voltage') %% Observation input
    fullyConnectedLayer(256, 'Name', 'Critic_StateFC1') %% Hidden layer
    reluLayer('Name', 'Critic_Relu1') %%Activation function
```

```
    fullyConnectedLayer(128, 'Name', 'Critic_StateFC2')]; %% Hidden layer
actionPath = [
    featureInputLayer(4, 'Normalization', 'none', 'Name', 'Action') %%% Action path,
input to critic
    fullyConnectedLayer(128, 'Name', 'Critic_ActionFC1')]; %%% Hidden layer
commonPath = [
    additionLayer(2,'Name', 'addition')     %%% Add two layer
    reluLayer('Name','Critic_CommonRelu') %%% activation layer
    fullyConnectedLayer(1, 'Name', 'Critic_Output')]; %%% Critic output
%%% Critic Option
Critic_Option = rlRepresentationOptions('LearnRate', Critic_Learn_Rate,
'GradientThreshold',1);
 critic =
rlQValueRepresentation(criticNetwork,obsInfo,actInfo,'Observation',{Voltage},'Actio
n',{'Action'},critic_Option); %% Create critic network
%% Actor Network
actorNetwork = [featureInputLayer(33, 'Normalization', 'none', 'Name', 'Voltage')
    fullyConnectedLayer(400,"Name","actor_FC1") %% Hiden layer1
    reluLayer("Name","actor_Relu1") %% Activation function layer 1
    fullyConnectedLayer(300,"Name","actor_FC2") %% Hiden layer2
    reluLayer("Name","actor_Relu2") %% Activation function layer 2
    fullyConnectedLayer(4,"Name","Action_actor") %% Output layer
    ];
%% Actor Option
 actorOptions =
rlRepresentationOptions('LearnRate',Actor_Learn_Rate,'GradientThreshold',1); %%
actor option
 actor =
rlDeterministicActorRepresentation(actorNetwork,obsInfo,actInfo,'Observation',{'Vol
tage'},'Action',{'Action'},actorOptions); %% Create actor
 agentOpts = rlDDPGAgentOptions(...
    'SampleTime', Ts,...
    'TargetSmoothFactor', 1e-3,...
```

```
    'DiscountFactor', GAMMA, ...
    'MiniBatchSize', BATCH ...
    'ExperienceBufferLength', 1e7, ...
    'ResetExperienceBufferBeforeTraining', true, ...
    'SaveExperienceBufferWithAgent', false);
 %% Exploration Parameters
% ----------------------
% Default: 1e-5
 DDPG_VarianceDecayRate = 1e-5; % if 1e-5 Half-life of 1,000 episodes
agentOpts.NoiseOptions.Variance = [55;55;55;55];
agentOpts.NoiseOptions.VarianceDecayRate = DDPG_VarianceDecayRate;
 agent = rlDDPGAgent(actor, critic, agentOpts); % Create agent
 MAX_ST = ceil(Tf/Ts);
 %%% For parallel computing: 'UseParallel',true, ...
%%%% criticOptions.UseDevice = 'gpu';
 % To enable save
trainOpts = rlTrainingOptions(...
    'MaxEpisodes', MAX_EP, ...
    'MaxStepsPerEpisode', MAX_ST, ...
    'ScoreAveragingWindowLength', AVERAG_WINDOW, ...
    'Verbose', false, ...
    'Plots','training-progress',...
    'SaveAgentDirectory', PATH_MODEL, ...
    'SaveAgentValue', AGENT_SAVE_THRESHOLD, ...
    'SaveAgentCriteria','AverageReward', ...
    'StopTrainingCriteria','AverageReward',...
    'StopTrainingValue', TRAINING_STOP);
 % To plot network architectures
% Show neural network architecture
actorNet = getModel(getActor(agent));
criticNet = getModel(getCritic(agent));
%plot(actorNet); title("Actor Network");
%plot(criticNet); title("Critic Network");
```

```
% % Print parameters
actorNet; actorNet.Layers;
% % Print parameters
criticNet; criticNet.Layers;
 %if USE_PRE_TRAINED_MODEL
    % Load experiences from pre-trained agent
    %sprintf('- Voltage control model: Loading pre-trained model: %s',
%PRE_TRAINED_MODEL_FILE)
   % RL_MODEL_FILE = strcat(PATH_MODEL, PRE_TRAINED_MODEL_FILE);
   % load(RL_MODEL_FILE,'agent');
%else
   % agent = rlDDPGAgent(actor, critic, agentOpts);
%end
 % Train the agent or Load a pre-trained model and run in Suimulink
sprintf ('\n\n ==== RL for Voltage Control ====================')
 %if (USE_PRE_TRAINED_MODEL)
 %   sprintf (' ---- Graded Learning in progress. Using pre-trained model: %s',
PRE_TRAINED_MODEL_FILE)
%end
%% Train the agent
doTraining = true;  %% true or false
if doTraining
    % Train the agent.
    trainingStats = train(agent, env, trainOpts);
    %% Save agent
    nEpisodes = length(trainingStats.EpisodeIndex);
    %fname = strcat(PATH_MODEL, VERSION, '.mat');
    %sprintf ('Saving file: %s', fname)
    save("VOltControlagent", "agent");
    display("End training: ")
    tend = datetime();
    display("Elapsed time:")
    telapsed = tend - start_time;
```

```
    telapsed
else
    % Load a pretrained agent for the example.
    load('VOltControlagent_2.mat','agent')
end
 %% Simulate DDPG Agent
%% =
simOptions = rlSimulationOptions('MaxSteps',MAX_ST); %% Simulation option
experience = sim(env,agent,simOptions); %% Simulation
 % ---------------------------------------------------------------------
% Environment Reset function
%%Load Variation for each episode
% ---------------------------------------------------------------------
function in = localResetFcn(in, VOLTAGE_CONTROL_MODEL)
block_Actual_Volt = strcat(VOLTAGE_CONTROL_MODEL,'/Power_Flow/Load');
Load_value = 1+0.4*(1-2*rand(1)); %% Value between 0-2
in = setBlockParameter(in, block_Actual_Volt, 'Value', num2str(Load_value));
end
```

Appendix-B.2: Reset function for DDPG Training

```
%% Termination Function
%% Terminate the episode training if voltage of all node are within the limit
%% If output is 1, RL will start new training episode
function y = fcn(v)
%% Initialize variable
total=0;
a=0;
for i=1:33
    if v(i)<0.95 || v(i)>1.05
      a=0;
      break; %%% If the voltage of any node is not in limit then for loop will terminate
      else
      a=1;
    end
      total=a+total;
end
if total == 33
   y=1;
else
   y=0;
end
```

Appendix-B.3: Reward function for DDPG Training

```
%%% Reward function
%%% Takes node voltage and Reactive Power generation as input

function r = fcn(V,Q9,Q16,Q24,Q30 )
Si=2000; %%% Inverter Capacity
%% Initialize the reward variable
r1=0;
%% For all node voltage
for i=1:33
r1=-58.15*power(V(i),4)+232.6*power(V(i),3)-534.28*power(V(i),2)+603.37*V(i)-
247.57; %% y = -58.15x4 + 232.6x3 - 534.28x2 + 603.37x - 247.57
end

%%%r2=0;r3=0;r4=0;r5=0;
%% Reward function for less active power curtailment
r2=10*(1-(abs(((Q9))/Si)));
r3=10*(1-(abs(((Q16))/Si)));
r4=10*(1-(abs(((Q24))/Si)));
r5=10*(1-(abs(((Q30))/Si)));
r =r1+r2+r3+r4+r5; %%% Total reward of the episode
```

Appendix-B.3: Load Flow function

```
%% Load Flow for IEEE 33 Bus system
%% Input to this funciton is Load & PV Generation
%%
function Vout = fcn(PQ_data, Load)

busdata_33bus=load('IEEE33busdata.m');

bus_data =  Load*busdata_33bus - PQ_data;

line_data=     [1    1    2   0.0922   0.0470

            2    2    3   0.4930   0.2511

            3    3    4   0.3660   0.1864

            4    4    5   0.3811   0.1941

            5    5    6   0.8190   0.7070

            6    6    7   0.1872   0.6188

            7    7    8   1.7114   1.2351

            8    8    9   1.0300   0.7400

            9    9   10   1.0440   0.7400

           10   10   11   0.1966   0.0650

           11   11   12   0.3744   0.1238

           12   12   13   1.4680   1.1550

           13   13   14   0.5416   0.7129

           14   14   15   0.5910   0.5260

           15   15   16   0.7463   0.5450

           16   16   17   1.2890   1.7210

           17   17   18   0.7320   0.5740

           18    2   19   0.1640   0.1565

           19   19   20   1.5042   1.3554

           20   20   21   0.4095   0.4784

           21   21   22   0.7089   0.9373

           22    3   23   0.4512   0.3083

           23   23   24   0.8980   0.7091

           24   24   25   0.8960   0.7011

           25    6   26   0.2030   0.1034

           26   26   27   0.2842   0.1447

           27   27   28   1.0590   0.9337
```

```
        28   28   29   0.8042   0.7006
        29   29   30   0.5075   0.2585
        30   30   31   0.9744   0.9630
        31   31   32   0.3105   0.3619
        32   32   33   0.3410   0.5302 ];
% Calculation of  per Unit
base_KVA=1000; %1MVA
base_KV=12.66;    %33KV
tolerance=0.0001;  %%% Maximum Error
dimension=size(line_data); %%% Feeder Size
n=dimension(1)+1;
ml_data=line_data(1:dimension(1),2:dimension(2));
%%Sending node
snd_node=ml_data(:,1);
%% Receiving node
rcv_node=ml_data(:,2);
%% Line resistance
line_rest=ml_data(:,3);
%% Line Reactance
line_react=ml_data(:,4);
%% Starting node of the feeder
for i=1:1:n
   if((max(i==rcv_node))==0)
       Snd=i;
       break;
   end
end
%% ending node of the feeder
no_end=4;
end_node=[18,22,25,33];
%% Precedance node of the feeder
pcd=zeros(n,1);
for i=1:1:n
```

```
    if(i~=1)
        pcd(i)=sdn_node(find(i==rcv_node),1);
    end
end
 %Node in decending order
dcu=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18;
    1,2,19,20,21,22,0,0,0,0,0,0,0,0,0,0,0,0;1,2,3,23,24,25,0,0,0,0,0,0,0,0,0,0,0,0;
    1,2,3,4,5,6,26,27,28,29,30,31,32,33,0,0,0,0];
%Brances in Decending order
ubd=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17;
    1,18,19,20,21,0,0,0,0,0,0,0,0,0,0,0,0;
    1,2,22,23,24,0,0,0,0,0,0,0,0,0,0,0,0;1,2,3,4,5,25,26,27,28,29,30,31,32,0,0,0,0];
%Finding n-matrix
nmat=[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1;
    0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1;……
bp=bus_data(:,2);
bq=bus_data(:,3);
%%%%%%%%%%%%%%%%%
max_error=1;
base_Z=bKV*bKV*1000/bKVA;;
S=complex(bp,bq);
S_pu=S/bKVA;
bp_pu=bp/base_KVA;
bq_pu=bq/base_KVA;
real=sum(bp_pu);
Reactive=sum(bq_pu);
lr_pu=lr/(base_Z);
lx_pu=lx/(base_Z);
% initialization
v_n1=1;
V2=0;
del_n1=0;
delta2=0;
```

```
Ploss_pu=0;
Qloss_pu=0;
Ploss_branch=zeros(1,n-1);
Qloss_branch=zeros(1,n-1);
iteration_no=0;
nv_pu0=zeros(1,n)+1;
nv_pu=nv_pu0;
tmpd0=zeros(1,n);
tmpd=zeros(1,n);
mm=0;
flag1=0;
%%  end of initialization
while (max_error>tolerance)
    Ploss_pu=0;
    Qloss_pu_pu=0;
   iteration_no = iteration_no+1;
    v_n1=1;
    del_n1=0;
    n_bu=[17,5,5,13];
    for m=1:1:ne
        for k=1:n_bu(m)   % no of node foe each branch
            cb_n=ubd(m,k);
            cr_n=rn(cb_n);
            csn=sdn_node(cb_n);
            if((nv_pu(cr_n)==nv_pu0(cr_n))&&(tmpd(cr_n)==tmpd0(cr_n)))
                v_n1=nv_pu(csn);
                del_n1=tmpd(csn);
                V2=nv_pu(cr_n);
                delta2=tmpd(cr_n);
                P2=0; Q2=0;
                coder.varsize('P2');
                coder.varsize('Q2');
                row=nmat(cr_n,:);
```

```
for l=1:1:n % also from l= k to n
    if(l==cr_n)
        P2=P2+bppu(l);
        Q2=Q2+bqpu(l);
        else if(row(l)==1)
                %%% Total load flowing through branch
                AddP2=Ploss_branch(find(l==rn));
                coder.varsize('AddP2');
                P2new = AddP2;
                P2=P2+bppu(l)+P2new; % from the current node l>k
               Q2=Q2+bqpu(l)+Qloss_branch(find(l==rn));
            end
    end
end % end of for loop made for summation at a node
A(cr_n)=P2*lr_pu(cb_n)+Q2*lx_pu(cb_n)-0.5*v_n1^2; %%%
%%% Calculation of the voltage
img=(A(cr_n))^2-((lr_pu(cb_n))^2+(lx_pu(cb_n))^2)*(P2^2+Q2^2);
if (img<0)
    fprintf('\n LF: imaginary voltage ');
    flag1=1;
    break;
end

B(cr_n)=sqrt(img);
imgg=B(cr_n)-A(cr_n);
if (imgg<0)
    fprintf('\n LFF: imaginary voltage ');
    flag1=1;
    indlfnc=1;
    break;
end
%% Voltage magnitude
V2=sqrt(imgg);
```

```
            %%% Voltage angle
            dd=(P2*lx_pu(cb_n)-
Q2*lr_pu(cb_n))/(P2*lr_pu(cb_n)+Q2*(lr_pu(cb_n)+V2^2));
            delta2=del_n1-atan(dd);
   % loss  of branchs calculation
            Ploss_branch(cb_n)=lr_pu(cb_n)*(P2^2+Q2^2)/V2^2;
            Qloss_branch(cb_n)=lx_pu(cb_n)*(P2^2+Q2^2)/V2^2
   % Arranging the data
            nv_pu(cr_n)=V2;
            tmpd(cr_n)=delta2;
            Ploss_pu=Ploss_pu+Ploss_branch(cb_n);
            Qloss_pu=Qloss_pu+Qloss_branch(cb_n);
         end  %end of calculation of node
      end     %end of calculation of lateral
      if (flag1==1)
         %%%%fprintf('\n LF1: load flow not converged imaginary voltage ');
         flag1=2;
         indlfnc=1;
         break;
      end
   end        %end of calculation of iterationation
   if (flag1==2)
         fprintf('\n LF2: load flow not converged imaginary voltage ');
         flag1=3;
         indlfnc=1;
         break;
   end
  % end of for loop for each bus load flow
  maxr=zeros(1,1);
  max_error=zeros(1,1);
  v_err=abs(nv_pu0-nv_pu);
  del_err=abs(nv_deg0-nv_deg);
  maxr=max(v_err);
```

```
   max_error=maxr;
   nv_pu0=nv_pu;
   nv_deg0=nv_deg;% Error of this iteration assign

end ; %%% end of while loop
del_deg=nv_deg*180/3.1416;
V_act=nv_pu*bKV;
%% Node voltage to be transferred by the function
Vout=nv_pu;
%% Calculation of actual power loss form the per unit value
Pbranch_act=Ploss_branch*bKVA;
Qbranch_act=Qloss_branch*bKVA;
Ploss_act=Ploss_pu*bKVA; %%% Feeder Active power loss
Qloss_pu_act=Qloss_pu*bKVA; %%% Feeder reactive power loss
%% Displaying result
%% Total number of iteration
fprintf('\n\n\n Maximum error deviation=%8.7f\t\t\t\n\n',max_error);
fprintf('\n\n\t\t total iteration  number=%g\t\t\t\t\n\n', iteration_no);
heading_voltage =['    Result showing voltage profile of the line              '
          '   Bus        Magnitude of Voltage         Voltage Angle'
          '  Number               (pu)                  (rad)      '];
 disp(heading_voltage);
    for m=1:n
        fprintf('\n%5g\t\t\t\t%8.7f\t\t\t\t%8.7f\t\t\t\t%8.7f\t\t\t\t',m,nv_pu(m),nv_deg(m));
    end
fprintf('\n\n');
heading_powerloss =['Active and reactive power losses in the branch'
        '   Branch       SBranch-RBranch          Branch Losses          '
        '   Number           SBN-RBN        RealPL(kW)   ReactivePL(kVAR)'];
 disp(heading_ powerloss);
    for bn=1:n-1
```

```
    fprintf('\n %5g\t\t\t\t %2g-
%2g\t\t\t\t%5.3f\t\t\t\t%5.3f\t\t\t\t',bn,sdn_node(bn),rn(bn),PL(bn)*1000,Qloss_branch
(bn)*1000);
end
fprintf('\n\nTotal Real power Loss of System(kW)    =
%5g\t\t\t\t\t\t\n',Ploss_pu*1000);
fprintf('\nTotal Reactive power Loss of System(kVar)=
%5g\t\t\t\t\t\n',Qloss_pu*1000);
end;
```

Appendix-B.4: Volt-VAR Droop Control

```
%% Volt-Var droop control
%%
clc;
clear;
%% Inverter capacity must be greater than the PV output generation
S_09=2000;
S_16=2000;
S_24=2000;
S_30=2000;
P9=1666.67; P16=1666.67; P24=1666.67; P30=1666.67; %%%% P/S ratio of 1.2
Q9=0; Q16=0; Q24=0; Q30=0;
busdata=    [1    0     0
             2    0     0
             3    0     0
             4    0     0
             5    0     0
             6    0     0
             7    0     0
             8    0     0
             9    P9    Q9 %%% Generation Capacity
             10   0     0
             11   0     0
             12   0     0
             13   0    0
             14   0    0
             15   0    0
             16   P16  Q16
             17   0    0
             18   0     0
             19   0     0
             20   0     0
             21   0     0
             22   0     0
             23   0     0
             24   P24   Q24
             25   0     0
             26   0     5
             27   0     5
             28   0     0
             29   0     0
             30   P30  Q30
             31   0     0
             32   0     0
             33   0     0 ];
busdata_droop=load('IEEE33busdata.m')-busdata;
Vout=loadflow_droop(busdata_droop); %%%% Load flow withour
V_09=Vout(1,09);
V_24=Vout(1,24);
```

```
V_16=Vout(1,16);
V_30=Vout(1,30);
fprintf('\nVotage at node 09 = %g\n',V_09);
fprintf('\nVOtage at node 16 = %g\n',V_16);
fprintf('\nVOtage at node 24 = %g\n',V_24);
fprintf('\nVOtage at node 30 = %g\n',V_30);
%% Setting for volt-VAR loop
Q2=0.44;v2=0.92;v3=0.98;v4=1.02;v5=1.08;
%% Optimal Volt–Var Curve Setting of a Smart Inverter for Improving Its
Performance in aDistribution System
%% For Node 09
if (V_09<=v2)
   Q_09=1*Q2;
   elseif (V_09>v2 && V_09<v3)
        Q_09=Q2*(V_09-v3)/(v2-v3); %%% Q2 in per unit
          else if (V_09>=v3 && V_09<=v4)
           Q_09=0;
             else if (V_09>v4 && V_09<v5)
                 Q_09=-Q2*(V_09-v4)/(v5-v4);
                else
                    Q_09=-1*Q2;
                end
             end
end
%% %% For Node 16
if (V_24<=v2)
   Q_24=1*Q2;
   elseif (V_24>v2 && V_24<v3)
        Q_24=Q2*(V_24-v3)/(v2-v3);
          else if (V_24>=v3 && V_24<=v4)
           Q_24=0;
             else if (V_24>v4 && V_24<v5)
                 Q_24=-Q2*(V_24-v4)/(v5-v4);
                else
                    Q_24=-1*Q2;
                end
             end
end
%% For Node 24
if (V_16<=v2)
   Q_16=1*Q2;
   elseif (V_16>v2 && V_16<v3)
        Q_16=Q2*(V_16-v3)/(v2-v3);
          else if (V_16>=v3 && V_16<=v4)
           Q_16=0;
             else if (V_16>v4 && V_16<v5)
                 Q_16=-Q2*(V_16-v4)/(v5-v4);
                else
                    Q_16=-1*Q2;
```

```
            end
        end
end
%% For Node 30
if (V_30<=v2)
   Q_30=1*Q2;
   elseif (V_30>v2 && V_30<v3)
        Q_30=Q2*(V_30-v3)/(v2-v3);  %%% Reactive power generation
          else if (V_30>=v3 && V_30<=v4)
           Q_30=0;
             else if (V_30>v4 && V_30<v5)
                 Q_30=-Q2*(V_30-v4)/(v5-v4);
                else
                    Q_30=-1*Q2;
                end
             end
end
%% Actual active and reactive power by SI in node 09
Q_9A=Q_09*S_09;
P_9A=sqrt(S_09*S_09-Q_9A*Q_9A);
fprintf('\nActual Active at node 09 is %g\t and Reactive power at 09 =
%g\n',P_9A,Q_9A);
%% Actual active and reactive power by SI in node 16
Q_16A=Q_16*S_16;
P_16A=sqrt(S_16*S_16-Q_16A*Q_16A);
fprintf('\nActual Active at node 16 is %g\t and Reactive power at 16 =
%g\n',P_16A,Q_16A);
%% Actual active and reactive power by SI in node 24
Q_24A=Q_24*S_24;
P_24A=sqrt(S_24*S_24-Q_24A*Q_24A);
fprintf('\nActual Active at node 24 is %g\t and Reactive power at 24 =
%g\n',P_24A,Q_24A);
%% Actual active and reactive power by SI in node 30
Q_30A=Q_30*S_30;
P_30A=sqrt(S_30*S_30-Q_30A*Q_30A);
fprintf('\nActual Active at node 30 is %g\t and Reactive power at Node 30 =
%g\n',P_30A,Q_30A);

%%%%%% Voltage after the load droop control
%% Checking the effect of the volt-VAR droop control

busdata=    [1   0    0
             2   0    0
             3   0    0
             4   0    0
             5   0    0
             6   0    0
             7   0    0
             8   0    0
```

```
 9    P9    Q_9A %%% Generation Capacity
10    0     0
11    0     0
12    0     0
13    0    0
14    0    0
15    0    0
16    P_16A  Q_16A
17    0    0
18    0     0
19    0     0
20    0     0
21    0     0
22    0     0
23    0     0
24    P_24A   Q_24A
25    0     0
26    0     5
27    0     5
28    0     0
29    0     0
30    P_30A  Q_30A
31    0     0
32    0     0
33    0     0 ];
busdata_droop=load('IEEE33busdata.m')-busdata;
Vout=loadflow_droop(busdata_droop); %%%% Load flow withour
V_09=Vout(1,09);
V_24=Vout(1,24);
V_16=Vout(1,16);
V_30=Vout(1,30);
fprintf('\nVotage at node after droop control 09 = %g\n',V_09);
fprintf('\nVOtage at node after droop control 16 = %g\n',V_16);
fprintf('\nVOtage at node after droop control24 = %g\n',V_24);
fprintf('\nVOtage at node after droop control 30 = %g\n',V_30);
```

Appendix-C: IEEE 33 radial feeder data

Line Data

Line Name	From Bus	To Bus	Lime Impedance	
			Resistance (Ohm)	Reactance (Ohm)
Branch-1	1	2	0.0922	0.047
Branch-2	2	3	0.493	0.2511
Branch-3	3	4	0.366	0.1864
Branch-4	4	5	0.3811	0.1941
Branch-5	5	6	0.819	0.707
Branch-6	6	7	0.1872	0.6188
Branch-7	7	8	1.7114	1.2351
Branch-8	8	9	1.03	0.74
Branch-9	9	10	1.044	0.74
Branch-10	10	11	0.1966	0.065
Branch-11	11	12	0.3744	0.1238
Branch-12	12	13	1.468	1.155
Branch-13	13	14	0.5416	0.7129
Branch-14	14	15	0.591	0.526
Branch-15	15	16	0.7463	0.545
Branch-16	16	17	1.289	1.721
Branch-17	17	18	0.732	0.574
Branch-18	2	19	0.164	0.1565
Branch-19	19	20	1.5042	1.3554
Branch-20	20	21	0.4095	0.4784
Branch-21	21	22	0.7089	0.9373
Branch-22	3	23	0.4512	0.3083
Branch-23	23	24	0.898	0.7091
Branch-24	24	25	0.896	0.7011
Branch-25	6	26	0.203	0.1034
Branch-26	26	27	0.2842	0.1447
Branch-27	27	28	1.059	0.9337
Branch-28	28	29	0.8042	0.7006
Branch-29	29	30	0.5075	0.2585
Branch-30	30	31	0.9744	0.963
Branch-31	31	32	0.3105	0.3619
Branch-32	32	33	0.341	0.5302

Appendix-D: Publication

Appendix-E: Plagiarism check report