Optimizing use of Air Conditioning Systems for Energy Conservation

By

Azaan Mahmood

00000327668

Supervisor

Cdre. Dr. Syed Sajjad Haider Zaidi

A Thesis submitted to Pakistan Navy Engineering College (PNEC), Karachi

National University of Sciences and Technology (NUST), Islamabad

In partial fulfillment of requirements for the degree of

Master of Science (MS) in **Electrical Engineering**

With specialization in Control

March, 2024

Optimizing use of Air Conditioning Systems for Energy Conservation

By

Azaan Mahmood

00000327668

Supervisor

Cdre. Dr. Syed Sajjad Haider Zaidi

Co-Supervisor

Dr. Antash Najib

G.E.C Committee

Dr. Muhammad Farhan Khan

Dr. Ashraf Yahya

A thesis submitted in partial fulfillment of requirements for the the degree of

Master of Science (MS) in **Electrical Engineering**

With specialization in Control

Department of Electronics and Power Engineering

Pakistan Navy Engineering College(PNEC), Karachi

National University of Sciences and Technology (NUST), Islamabad

March, 2024

FORM TH-4
(MUST BE TYPE WRITTEN)

National University of Sciences & Technology

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: (Student Name& Regn No.) Azaan Mahmood 00000327668

Titled: Optimizing use of Air Conditioning Systems for Energy Conservation be accepted in partial fulfillment of the requirements for the award of Masters degree.

Examination Committee Members

1. Name: Dr. Muhammad Farhan Khan Signature:

2. Name: Dr. Ashraf Yahya Signature:

Supervisor's name: Dr. Syed Sajjad Haider Zaidi Signature:

Date: 05-03-24

Co-Supervisor's name: Dr. Antash Najib Signature:

Date: 05-03-24

MUHAMMAD ALI
Captain Pakistan Navy
Dean Electronics & Power Engg
NUST - PNEC

Head of Department

05-03-24

Date

COUNTERSINGED

Date: 12-03-24

MIRFAN NADEEM
Captain Pakistan Navy
Deputy Commandant
PNS JAUHAR

Dean/Principal

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by Mr. Azaan Mahmood (Registration No. 00000327668), of PNEC (College) has been vetted by under signed, found complete in all respects as per NUST Statutes/Regulation, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ____________________

Name of Supervisor: Dr. Syed Sajjad Haider Zaidi

Dated: 05-03-24

Signature: ____________________

Name of Co-Supervisor: Dr. Antash Najib

Dated: 05-03-24

Signature: (HoD) MUHAMMAD ALI
Captain Pakistan Navy
Dean Electronics & Power Engg
NUST - PNEC

Dated: 05-03-24

Signature: (Dean/Principal) M IRFAN NADEEM
Captain Pakistan Navy
Deputy Commandant
PNS JAUHAR

Dated: 12-03-24

Declaration

I, *Azaan Mahmood* declare that thesis titled **"Optimizing use of Air Conditioning Systems for Energy Conservation"**, the work published are my own and created by me according to my own research study.

I endorse that:

1. The work has not been presented elsewhere for assessment.
2. The material that has been used from other sources it has been properly acknowledged / referred.
3. This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Azaan Mahmood,

00000327668

Certificate for Plagiarism

It is certified that PhD/M.Phil./MS Thesis Titled "Optimization Method of Air Conditioner for Minimum Energy consumption while improving user comfort" by Mr. Azaan Mahmood (Regn No. 327668) has been examined by us. We undertake the follows:

a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
c. There is no fabrication of data or results which have been compiled / analyzed.
d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

Name & Signature of Supervisor

Supervisor name: Dr. Syed Sajjad Haider Zaidi

Signature: ______________

Date: 13th February 2024

turnitin

Similarity Report ID. oid:12111:51070571

DR. S. SAJJAD HAIDER ZAIDI
Commodore
Associate Professor
NUST-PNEC

- 13% Overall Similarity

Top sources found in the following databases:

- 6% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	mdpi.com Internet	<1%
2	University College London on 2023-11-06 Submitted works	<1%
3	University of Wales, Lampeter on 2023-09-29 Submitted works	<1%
4	Imperial College of Science, Technology and Medicine on 2021-09-03 Submitted works	<1%
5	Yan Du, Fangxing Li, Kuldeep Kurte, Jeffrey Munk, Helia Zandi. "Demon... Crossref	<1%
6	repository.up.ac.za Internet	<1%
7	Imperial College of Science, Technology and Medicine on 2019-09-12 Submitted works	<1%
8	Roehampton University on 2024-01-02 Submitted works	<1%

Copyright Notice

This thesis is dedicated to *my cherished and supportive parents, as well as my mentoring older siblings. Their unwavering support, encouragement, and esteem have been instrumental in bringing about the completion of this work.*

Abstract

A thorough examination of air conditioning (A/C) systems on the basis of cost reduction and energy efficiency is required. Existing literature lacks a comprehensive analysis of A/C systems, specifically the comparison between traditional control methods with modern approaches based on Reinforcement Learning (RL) algorithms on basis of cost and energy. As such evaluating the performance of A/C systems, employing both traditional and modern control techniques, assessing the effectiveness of modern RL based controllers. For such a task a simulation environment was developed using MATLAB and SIMULINK, which includes a Room Model, A/C Model, and Controller Model. Rigorous testing is conducted on proposed controllers, encompassing traditional PI controller and two modern RL based controllers based on DQN and TD3 algorithms. Additionally a TD3-based PI controller was also developed and over three experiments were conducted for each controller model and a comparative analysis was performed. Experimental outcomes highlight a substantial 140-150% performance differential between traditional and modern control groups. Additionally, negligible differences are observed between DQN and TD3 algorithms in the context of A/C systems. These insights contribute to optimizing A/C systems for enhanced performance and thermal comfort.

Keywords: *A/C Systems, MATLAB, DQN, TD3, Reinforcement Learning, PI, Energy Conservation, Thermal Comfort*

Acknowledgments

All glory to ALLAH for the His countless blessings in completing this thesis in time. All appreciation to my family, who were very understanding throughout this endeavor. I would like to give my sincere gratitude to my supervisor Dr. Syed Sajjad H Zaidi for the thorough supervision and support. I attribute the level of my Master degree to his inspiration and determination because without him, this work would not be possible. His consistency, promptness, regularity helped me in completing my thesis in a well-organized manner.

I extend heartfelt appreciation towards Dr. Antash Najib for his supportive suggestions, clarifications, reinforcement and guidance on the various aspects on research thesis.

I would like to offer sincere regards to Dr. Ashraf Yahya and Dr. Muhammad Farhan Khan for their invaluable contributions as members of my thesis committee.

Additionally, I would also like to extend special thanks to all the teachers that taught me during my coursework.

Contents

1 Introduction **1**

1.1 Description of Thesis . . . 1

1.2 Research Approach . . . 3

1.3 Problem Statement . . . 4

1.3.1 Optimization Algorithms . . . 5

1.4 Thesis Layout . . . 5

1.4.1 Chapter 2 Literature Review . . . 5

1.4.2 Chapter 3 System Model . . . 5

1.4.3 Chapter 4 PI Controller Model . . . 6

1.4.4 Chapter 5 Simulation Environment . . . 6

1.4.5 Chapter 6 Reinforcement Learning based models . . . 6

1.4.6 Chapter 7 Results . . . 6

1.4.7 Chapter 8 Conclusion . . . 6

1.4.8 Chapter 9 Future Recommendation . . . 7

2 Literature Review **8**

3 System Model **16**

3.1 Air Conditioner Overview . . . 16

3.1.1 Indoor Unit . . . 17

3.1.2 Outdoor Unit . . . 18

3.2 Engineering Issues in Air Conditioning Systems 19
3.2.1 Traditional Controller and Control Techniques Limitations: 20
3.3 Simulation Environment of the System: 22
3.3.1 Air Conditioner Model: 22
3.3.2 Room Model: 23
3.3.3 Smart Controller: 25
3.4 Parameters of Simulation Environment: 25

4 PI Controller Model **26**
4.1 PID Algorithm 27
4.1.1 Omission of Derivative Controller: 27
4.2 PI Model 28

5 Reinforcement Learning based models **30**
5.1 RL Controller model: 30
5.1.1 Agent Policy: 33
5.1.2 RL Terminology: 33
5.1.2.1 Episode: 34
5.1.2.2 Average Episode Reward: 35
5.1.2.3 Learning Rate: 35
5.1.2.4 Q-Values: 35
5.1.2.5 Actor-Critic Pair 36
5.2 Controller Models: 37
5.2.0.1 The Reward Function: 38
5.2.0.2 Deep-Q Network Algorithm: 39
5.2.0.3 Twin-Delayed Deep Deterministic Policy Gradient Algorithm: 40
5.2.1 RL-PI Controller: 41

6 Results **42**

Appendices **56**

.1 DQN Code: 56

.2 TD3 Code: 60

.3 Plot Graph: 66

.4 Reward Function Code: 68

List of Figures

1.1 Electricity consumption of different sectors of Pakistan. [1] 2

1.2 Aggregated Electrical Consumption from 2012-2022.[2] 3

1.3 Basic Closed Loop Model of A/C Optimization Model 4

2.1 Optimization Techniques presented in Literature 8

3.1 Air Conditioning Action 18

3.2 A/C Model 23

3.3 Room Model 24

4.1 System Model for Traditional PI controller 29

5.1 RL-Based approach for A/C system. DNN: deep neural network 32

5.2 DQN and TD3 Based Controller Model 41

5.3 RL-PI based Controller Model 41

6.1 First experiment of DQN/TD3 agent. 43

6.2 Second experiment of DQN/TD3 agent 43

6.3 Third experiment of DQN/TD3 agent 44

6.4 Episode Graph of DQN agent 44

6.5 Episode Graph of TD3 agent 45

6.6 Cost graph of traditional control technique. 45

6.7 Temperature response of the PI controller systems using CST and RL methods. 47

6.8 Episode graph of RL-PI controller using TD3 agent 47

List of Tables

1.1 Electricity Consumption of different sectors (July-March) Fiscal Year 2023. [1] . . . 2

3.1 Parameters of the Simulation Environment . . . 25

5.1 DQN Agent Parameters . . . 39

5.2 TD3 Agent Parameters . . . 40

5.3 Observation and Reward Function Space for respective controllers . . . 41

6.1 TD3/DQN Agent Cost out of total 1440 minutes. **Note:** Third row signifies higher cost due to positive 8 temperature added for simulation purposes. . . 42

6.2 Difference between both algorithms. . . . 45

6.3 Cost of PI Controller tuned by RL and inbuilt CST of MATLAB. The experiment is ran for a total of 1440 minutes. . . . 46

6.4 Cost difference of the different controller models. Only the first experiment of each is considered . . . 46

List of Abbreviations and Symbols

Abbreviations

A/C = Air Conditioners or Air Conditioning

HVAC = Heating, Ventilation, Air Conditioning

PI = Proportional Integral

RL = Reinforcement Learning

DL = Deep Learning

NN = Neural Network

RNN = Recurrent Neural Network

DDPG = Deep Deterministic Policy Gradient

TD3 = Twin Delayed DDPG

DQN = Deep Q-Network

RL-PI = Reinforcement Learning based Proportional Integral

Chapter 1

Introduction

A notable amount of energy consumption of a building in the sub-continent comes from cooling systems. According to Pakistan Economic 2023 report, referring to table(1.1) and figure(1.1) about 46% of total generated electrical energy is consumed by households [1].

Additionally total energy consumption consumption over the 10 year period from 2012-2022 is also illustrated in figure(1.2) [2]. Due to higher temperatures in certain regions, people prefer Air Conditioners (referred to as A/C systems moving forward) as a means of higher quality comfort, and in doing so causes increasing usage of such devices. Hence it is of importance to work on efficient energy consumption for A/C systems.

1.1 Description of Thesis

There have been various different methods researched and developed ranging from sensor optimization to applying elaborate control strategies. Among such, control strategies are popular and feasible to implement, as they do not require much physical components and are economical. Even among control methods, optimizing such methods have also been greatly researched upon and provide more cost effective solutions than their original counter parts.

Sector	**Consumption (in GWh)**	**Share(%)**
Household	39,200	46.65%
Commercial	6,576	7.83%
Industry	23,687	28.19%
Agriculture	6,906	8.22%
Others	39,200	9.12%
Total	**84,033**	

Table 1.1: Electricity Consumption of different sectors (July-March) Fiscal Year 2023. [1]

For an A/C system, optimization depends on targeting various processes of the device. Different optimization techniques, whether physical or programmable exist separately for the three main components of an A/C system; (1) Condenser, (2) Evaporator, and (3) Compressor (including the Variable Frequency Drive design implemented in inverter A/C variants).

Additionally, the core parameters of an A/C system can be targeted as well; (1) Air Quality Index (2) Temperature (3) Humidity. The methods for developing new optimization techniques encompasses these systems. We aim to study the different effects of various optimization algorithms on an A/C system model using MATLAB simulations.

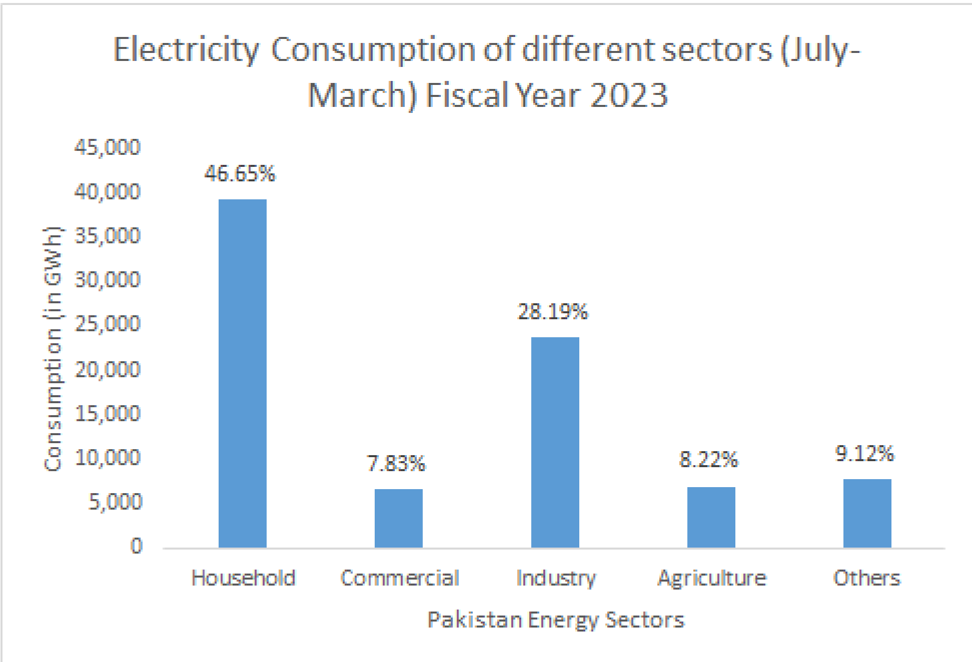

Figure 1.1: Electricity consumption of different sectors of Pakistan. [1]

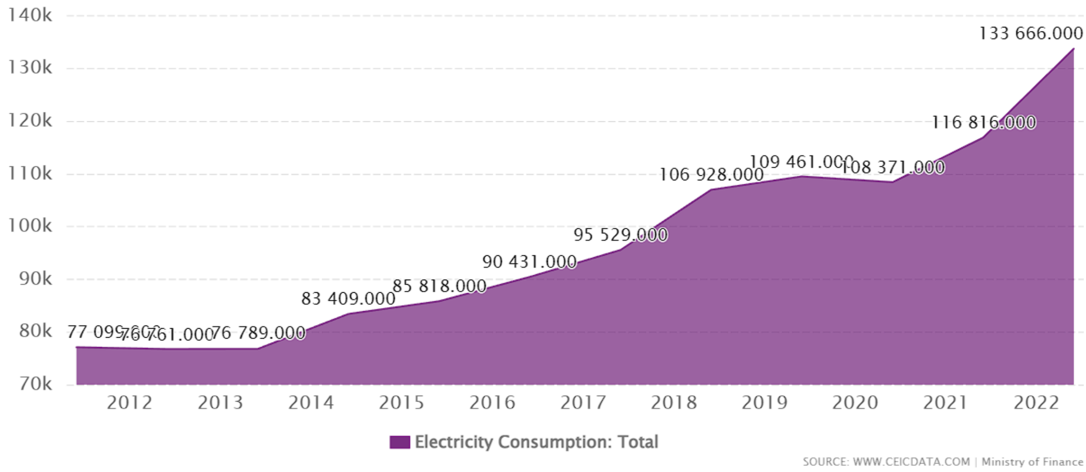

Figure 1.2: Aggregated Electrical Consumption from 2012-2022.[2]

1.2 Research Approach

Optimization methods are based on studying the effects of current methods and utilizing additional mathematical techniques on an existing method or entirely new processes utilizing the data provided by already in use approaches.

The scope of this study is limited to discussing the various algorithms being used on an A/C system model and providing a comparative study and a cost benefit ratio between the traditional PI controller against newer controllers based on reinforcement learning algorithm. Figure[1.3] shows the basic methodology to be followed. A temperature signal will be provided the system which consists of the controller and the plant; the air conditioner system and the area housing the system.

A closed loop feedback system will be implemented with a temperature sensor which provides error values and the controller will be tuned with a Reinforcement Learning algorithm that will reduce the error given time and accuracy towards the set-point.

The Research work will be divided into the following major categories and are explained individually in their own sections:

1. Study of Air Conditioning systems.

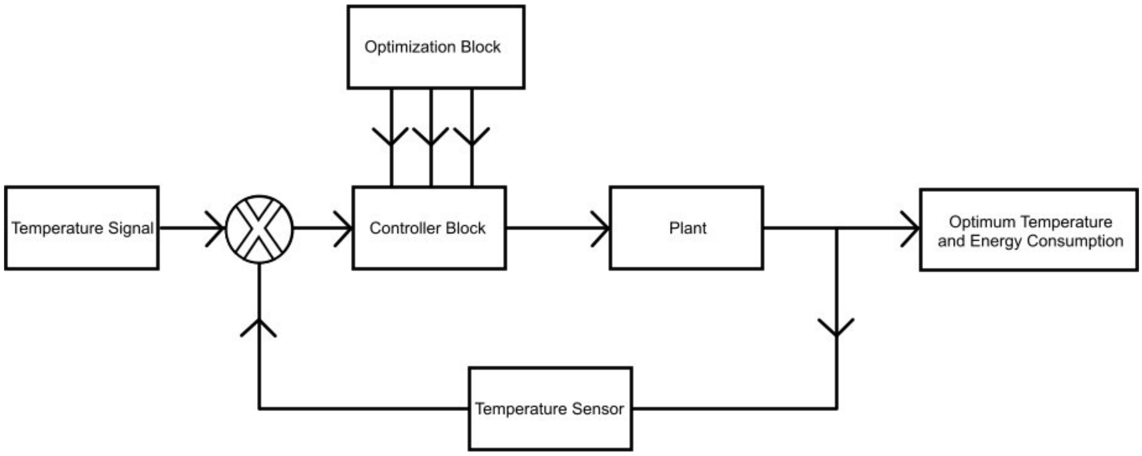

Figure 1.3: Basic Closed Loop Model of A/C Optimization Model

2. Study of worked already produced for the energy efficiency problems.

3. Study, design and implementation of:

    (a) pure PI Controller model.

    (b) Deep Q-Network (DQN).

    (c) Twin Delayed DDPG (TD3).

    (d) RL based PI controller model.

4. Simulation of all aforementioned models.

5. Comparative analysis of the aforementioned models.

All of these are explained in their individual chapters.

1.3 Problem Statement

Research showcases there is an inexhaustible list of optimization techniques available, however there is no comparative analysis and cost analysis between traditional and modern optimization techniques, while also lacking any in modern versus modern analysis and it would be difficult to compare all of them. As such, a select few artificial intelligence based algorithms will be given a comparative analysis on a PI enabled A/C system.

1.3.1 Optimization Algorithms

This research focuses on finding a feasible, sophisticated yet generalized and energy efficient model based on different optimization techniques. The optimization techniques RL based, that are compared and their cost is analyzed. The inspiration for cost analysis approach has been adopted from Yan Du et. al. [3] Following techniques have been studied and analyzed for this air conditioners optimization model:

1. PI Controller Model
2. DQN Model
3. TD3 Model
4. RL Based PI Controller Model (based on Model Predictive Controller)
5. Machine Learning

1.4 Thesis Layout

The remaining chapters will be divided as follows:

1.4.1 Chapter 2 Literature Review

In this chapter we discussed the trend of research in the field of optimization techniques on A/C models. Different research papers have be scrutinized to find out the research gap.

1.4.2 Chapter 3 System Model

In this chapter brief overview of the System model is presented. The system model is a combination of the Air Conditioning system and the housing of the system. The operating principle is described and the MATLAB simulation environment is described. Components of the system are discussed.

1.4.3 Chapter 4 PI Controller Model

In this chapter, a traditional control technique is used on the Air Conditioning model to have a base line of comparison with other models. The efficiencies and deficiencies of the controller are also discussed. The simulations are provided separately for this model in this chapter.

1.4.4 Chapter 5 Simulation Environment

In this chapter, a simulation environment is proposed, that the RL based models are to use in Chapter 6. While in-depth, some parts are omitted and included in Chapter 6 for coherency

1.4.5 Chapter 6 Reinforcement Learning based models

In this chapter, all of the modern reinforcement learning based models are discussed, this chapter is divided into several subsections for in-depth discussion of each model. The simulations for each are provided separately for this model in this chapter

1.4.6 Chapter 7 Results

In this chapter, the results attained from the proposed techniques are mentioned along with comparative analysis of each model.

1.4.7 Chapter 8 Conclusion

In this chapter, on the basis of Chapter 7 a complete holistic overview is provided.

1.4.8 Chapter 9 Future Recommendation

In this chapter, future recommendations and way ahead with the hardware and algorithm are discussed for researchers to look up to.

Chapter 2

Literature Review

Researchers have used many optimization techniques on various different parts of an A/C system. Figure [2.1] shows a brief outline of what optimization techniques that are being utilized on A/C systems.

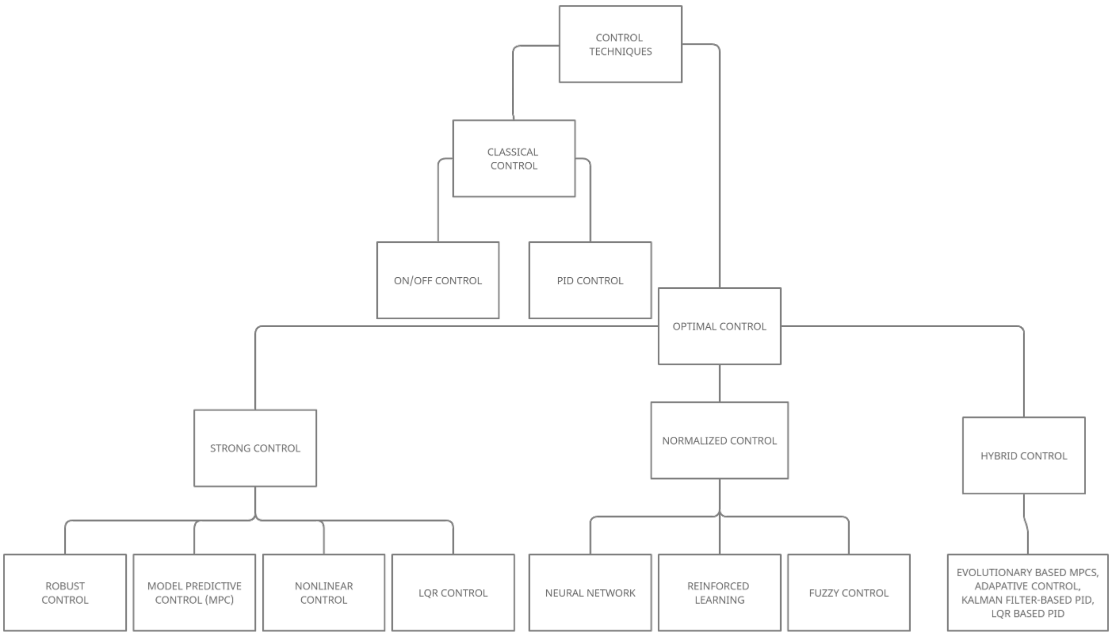

Figure 2.1: Optimization Techniques presented in Literature

A hierarchical distributed MPC scheme is proposed for buildings with multiple connected rooms and zones HVAC systems. The objective of the system is to minimize energy while maintaining acceptable comfortable thermal values. The scheme contains two separate controllers, which provide the temperature and predictive information of every room called the high-level controller, while the second controller optimizes efficiency and thermal comfort. The author states 6.8 second time feasibility given that the operating system is powerful [4].

Forecasting is another technique utilized with MPC. Forecasting is a technique used to minimize power consumption in peak times. The load profile defined by the system is then forecasted. A study uses this algorithm as a two-part forecaster where an exponentially weighted moving average (EWMA) model is used for long trends and a second-order regression model (AR2) is used for smaller variations. Given a thermostat schedule, HVAC power consumption can be reduced [5].

An alternating Direction Method of Multipliers (ADMM) based distributed model predictive control is formulated [6]. The method minimizes the sum of the total energy cost of an HVAC system, in different zones of the university building. The proposed model allows for scalability and provides flexible trade-offs in thermal comfort and energy cost.

In [7], a Stochastic MPC (SMPC) algorithm is proposed. The study strategizes statistical knowledge with predefined comfort levels to minimize energy usage using error forecasting based on the building's ambient conditions. The study promises dynamic learning properties with 90% satisfaction levels.

Linear MPC is also used by linearizing the non-linear model of an HVAC system. As non-linear models are generally now cost-effective and computing-intensive, the paper provides an approximate linear model by using Feedback Linearization and Piecewise Linearization for an analytical model. The study deduces that the given linearization technique structure along with the optimal control algorithm provides results approximate to the nonlinear model [8].

Author Chanthawit et al. proposed a design for a supervisory MPC. To ensure thermal comfort, Predicted Mean Vote (PMV) technique is utilized for acceptable set-point bounds. The technique uses two controllers to implement the design. One for steady-state analysis and set point thresholds, and one for applying the MPC algorithm. This technique allows for a 13.43% reduction in electricity consumption by an HVAC system [9].

Another MPC technique is discrete MPC. Consisting of three major components: the predictive model implemented by radial neural networks using a genetic algorithm; an objective function for optimizing thermal comfort and energy; and an optimization technique, which is discrete MPC. The technique has up to 30% savings in energy usage. [10].

On-Off control can also be optimized using optimal control laws. Using multi-objective optimization, the controller regulates indoor temperature considering outdoor ambient temperatures as well. Acceptable performance but the key feature is a low cost of implementation [11].

This study compares three different on-off optimal control laws; (1) Traditional ON/OFF control; (2) An intelligent ON/OFF control using a prediction algorithm; and (3) designing an optimal controller utilizing the PMV technique to conform to acceptable thermal comfort. For minimal energy consumption cases, PMV is best suited [12].

As Linear Quadratic Regulator is an optimization scheme as well as a well-known control law, feedback linearization was applied to an HVAC system to linearize it around the constraints (thermal comfort and CO2 control). The proposed approximated linear model was then optimized using the LQR control scheme [13].

Optimal Control (OPC) Schemes are control techniques employing generalized control structures with robust algorithms for optimization. Papers have utilized several different algorithms on classic control to generate an idea of feasibility and cost efficiency.

Velimir Congradac et al. [14] in their study put forward a Genetic Algorithm for classical control to solve non-linear problems. The study is simulation-optimization based and the suggested method is employed on a CO2 concentration control for a standard

HVAC system. Along with MATLAB Simulink, the results are co-related and verified with Energy Software.

In 2006, an important study by K.F. Fong et al. [15] was introduced. The study utilizes evolutionary programming for optimal control. Evolutionary algorithms are nature-inspired algorithm that mimics natural evolutionary schemes and are population-based metaheuristic algorithms for optimization. Newer Genetic Algorithms are a sub-set of evolutionary programming. The study implements a scheme to handle constrained, discrete and non-linear problems. The optimization effectiveness was deduced via month-wide simulation, which provided 7% savings with no added cost.

Scheduling Algorithms are techniques used for optimization based on day, weather, and other external factors based on the model of the system the study is going for. Accordingly, a study optimizes general HVAC energy usage in Industrial processes based on weather data. The study proposes a Binary Integer Linear Programming (BILP) approach for scheduling. Results of the study simulate savings of up to 30%, however comments on the impracticality of the approach for shorter periods (an hour at a time) [16].

Another study performs a global optimization model for the whole HVAC model. The schedulable potential evaluation indexes many parameters such as peaks, valleys, cutting, and capacity. These parameters are then used to develop a load profile for the model. A simulated HVAC operation is implemented and the load profile is applied. The results of the processes are found to be adjustments to the pumps and fan load curves [17].

Greedy scheduling is another scheme proposed by researchers. N. Chakraborty et al. propose a multi-object optimal scheduling framework. It is based on Johnson's elementary circuit, having the objective of continuous thermal comfort and minimizing power fluctuations. The study implements a greedy algorithm based on Pareto optimization, which utilizes the survival criterion of parameters. Efficient or chosen parameters are optimized more heavily compared to non-optimal parameters. If chosen correctly, such optimization schemes provide efficient optimization and good computation headroom. However, such algorithms do not provide complete and fair optimizations [18].

As HVAC systems are getting considerably complex, algorithms needed need also to be complex. Using interconnected Neural Networks and using online functionality, the author has used a deterministic search algorithm to optimize the three separate artificial neural network substations (rooms). The results are then averaged and compared to provide better training [19].

Mixed Integer Programming for Neural Networks is also used extensively. An ANN algorithm type is an adaptive neuro-fuzzy inference system (ANFIS). A study provides a systematic approach to optimization using ANFIS via a mix-integer problem with non-linear constraints on a genetic algorithm; by analyzing the model of the HVAC system (consisting of cooling loads and consumption of energy by heat exchangers and other devices) [20].

Similarly, this study optimizes new HVAC systems known as Smart Dual Fuel Switching Systems (SDFSS) via artificial neural networks (ANN). The design requires a rich and diverse feature set of humidity, wind speed, direction, and temperature to produce accurate results (up to 1.3 RMSE) however can be used in IoT applications given powerful computing [21].

In [22], a linear constraint mix-integer method is used with Deep Neural Network (DNN). The method utilizes weather and energy prices during the day (in Detroit, MI, USA) to forecast consecutive days' optimization. This method imitates MPC (which works on forecasting) and is provided as an alternative to it.

Another ANN is Long Short-Term Memory (LSTM). LSTM is an architecture that provides the Short Term of an RNN, however on larger time steps, which RNN cannot do. As such LSTM is good for prediction-based applications where short-term memory is required for larger and more frequent time steps. As such Ming Li. Et al [23] uses LSTM to predict temperatures in their experiments, providing higher prediction accuracy than multi-layer perceptron (MLP) and back propagation (BP) neural networks.

Reinforced Learning (RL) is a technique used when a scenario necessitates trial and error more than immediate reward gains. As disturbance is an issue in any real-life scenario, RL can be used to bias the controller toward said disturbances. As disturbances in HVAC systems can be the outside temperature, heat from occupants, lights, etc. In a study [24], disturbances are assumed to be immeasurable and unpredictable. As such a model is proposed, that biases the controller using the Q-Learning framework. A Q-Learning scheme provides an optimal control solution that allows the convergence of control parameters without the effect of disturbances.

Similarly, another study uses ANN-based MPC to save operating costs between 6% and 53% (for best fit). The study utilizes a new algorithm called the best network after multiple iterations (BNMI). It is a prediction-based ANN algorithm that is utilized with MPCs [25].

Research study [26] models a DNN-based HVAC optimum control scheme. As control settings and energy of HVAC are extremely non-linear, DNN provides robust techniques of data extraction. The author provides a comparative analysis of DNN against several state-of-the-art optimizing techniques (Particle Swarm, SVM, BPNN, and Random Forest) and has the lowest RMSE (of 3.78) among them.

A combination of algorithms that are used to solve the same problem is dubbed under the umbrella term of "Hybrid Algorithms". Hybrid MPCs are considered when traditional MPCs are used in tandem with another algorithm. A study combines an MPC model with Neural Network along with a feedback linearization model. The nonlinearity of the HVAC system is processed by an inverse neural network model. The model implements three different controls, (1) Economizer Control, (2) an optimal start-stop control, and (3) a load-shifting control. This approach provides 13% cost savings [27].

Another MPC-based scheme employs a fuzzy-optimization scheme. The study provides a new framework for optimization schemes involving fuzzy constraints. Fuzzy constraints are values that are real and do not have a fixed value. As real-life works on fuzzy constraints, a particle swarm technique (PSO) is used to solve the non-linear nature of the model. The model provides similar performance as other schemes while employing a completely new technique [28].

Due to HVAC systems having multiple disturbances, researchers have sought to use stochastic models as well. A study utilizes the stochastic nature of the system which incorporates the Markov Occupancy Model (MOM). The model works on the Markov Model, which is a stochastic method for systems having pseudo-randomly changing properties, but does not depend on the past state of the system. An HVAC system has random states that do not depend on the previous state, as such the study utilizes this notion of the model along with building occupancy. The model is used on an online optimizer and an offline optimizer [29].

A recent study in 2019, showcases a multi-objective genetic algorithm (MOGA) along with ANN for optimization between thermal comfort and energy consumption. Objective variables considered by the study are passive solr design, chiller operation, thermostat settings, and person dissatisfaction rate. As objective variables are more than two, a multi-objective genetic algorithm is used [30].

Another study by Rand Talib et al. put forwards a Data-Enabled optimization technique for selective components of an HVAC system. Three different data-driven modeling techniques are employed; (1) Artificial Neural Network (ANN), (2) Support Vector Machine (SVM), and (3) Aggregated Bootstrapping (BSA). The optimization problem is resolved using a Genetic Algorithm to reduce simulated data error from the actual data. The data set so used from the chilled water variable air volume (VAV) from an HVAC system. The proposed model provides low variation between simulation and actual data, with as low as 1.22% for the cooling coil and 9.04% for the fan [31].

This study [32] focuses on compressors of HVA/Cs. The study utilizes a Support

Vector Machine (SVM), as a classification function. SVM is a supervised machine learning method used for the three main Machine Learning schemes; (1) Regression, (2) Classification, and (3) Outlier detection (also called anomaly detection). The design is used to diagnose faults in compressors with more than 80% accuracy.

A study utilizes three different optimization models. (1) Kalman Filter, (2) Genetic Algorithm, and (3) Particle Swarm. The study filters the sensors reading and passes through both Genetic and Particle Swarm algorithms, then they have compared accordingly to the baseline provided by simulation software. The study compares both algorithms to have high optimization rates, 27.32% and 31.42% for PSO and GA respectively [33].

Chapter 3

System Model

In 1902, the very first modern Air Conditioning system was invented by an American Engineer, Willis Haviland Carrier, who was experimenting with humidity control and concepts of mechanical refrigeration. This was later improved using a belt driven condensing unit with blowers for the indoor and outdoor unit and evaporator coils. This current design is the mechanical model of modern air cooling devices [34].

Nowadays, research is held to optimize and increase the efficiency of such systems, ranging from optimizing compressor units which utilizes speed control and frequency control with relatively traditional micro controller technology (PID controllers), to implementation of advance control techniques to optimize the total working of an A/C system on current built physical technology.

3.1 Air Conditioner Overview

A/C System consists of four major components:

- Evaporator
- Condenser

- Expansion Valve
- Compressor

The modern air conditioning systems works on the basis of principles of refrigeration. The cycle is defined as follows:

1. **Evaporation:** As hot air rises and cool air sinks, the warm air is draw in to the evaporator coils(that is housed on the indoor unit) containing the refrigerant coolant, which absorbs the heat from the intake warm air, which cools the air that the blower fan releases from the indoor unit. The refrigerant is in a low-pressure vapor state.
2. **Compression:** The refrigerant which is now vaporized due to the evaporator action, is then compressed by the compressor. The process then raises the temperature and pressure, which makes it into high-pressure/temperature gas.
3. **Condenser:** This unit then releases the heat to the outside from the refrigerant, cooling it down, causing it to condense into high-pressure liquid
4. **Expansion:** The high-pressure liquid, is passed through the expansion valve, where its pressure is dropped, causing it to become partially gaseous, low pressure and low temperature.
5. **Evaporation:** As similar to part 1, this refrigerant cools the intake warm air again, which is then blown outwards, causing the cycle to repeat

Figure [3.1] show cases the A/C action:

3.1.1 Indoor Unit

An air conditioner's indoor unit, often referred to as the evaporator unit consists of components that work together to cool and circulate air within a building. Here are the main components typically found in an indoor unit:

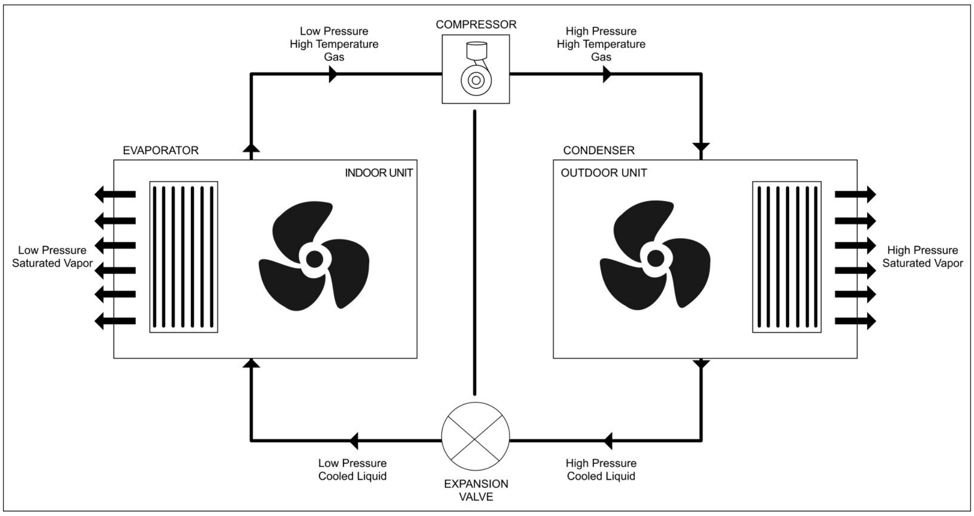

Figure 3.1: Air Conditioning Action

1. **Evaporator Coil:** As the evaporator coil absorbs heat from the indoor air, the refrigerant flowing through the coil warms the indoor air passing over it, which causes the refrigerant to evaporate, cooling the air in the process.

2. **Blower Motor:** The blower motor is responsible for pulling air from the room through the evaporator coil and then pushing the cooled air.

3. **Expansion Valve:** Also known as the thermal expansion valve, controls the flow of refrigerant into the evaporator coil. It regulates the refrigerant's pressure and temperature, allowing for efficient cooling.

4. **Sensors:** Sensors for temperature and humidity are integrated into the indoor unit to monitor and adjust the cooling process for comfort and efficiency.

5. **Main Controller:** The main controller that governs all the processes using various placed sensors in the indoor and outdoor unit.

3.1.2 Outdoor Unit

The Condenser or the outdoor unit, contains components that work together to release heat from the indoor air to the outdoor environment. Here are the main components found in an outdoor unit:

1. **Compressor:** The compressor is responsible for circulating the refrigerant and increasing its pressure and temperature, initiating the heat exchange process.

2. **Condenser Coil:** The condenser coil exchanges heat from the refrigerant flowing from the coils of evaporator allowing the the refrigerant heat to disperse. As the hot, high-pressure refrigerant flows through the coil, it releases heat to the outdoor air, causing the refrigerant to condense from a gas to a liquid state.

3. **Fan Motor:** The fan motor drives the fan blades that draw outdoor air over the condenser coil. This airflow helps dissipate the heat collected from indoor air.

4. **Pressure and Temperature Sensors:** These sensors monitor various parameters within the outdoor unit, aiding in system control and performance optimization.

5. **Suction Line and Liquid Line:** These refrigerant lines connect the indoor and outdoor units, carrying low-pressure, low-temperature refrigerant vapor back to the compressor and the liquid line transporting high-pressure liquid refrigerant from the condenser to the evaporator coil.

3.2 Engineering Issues in Air Conditioning Systems

There are many engineering related issues with air conditioning systems. Some of them are listed as follows:

1. **Energy Efficiency:** Energy efficiency issues are majorly due to old and traditional controlling methods (ON-OFF, or none etc), sensor wear down, fan control issues or lack of maintenance.

2. **IAQ:** IAQ (Indoor Air Quality) refers to the state of the air indoors. It factors parameters such as; temperature, humidity, and pollutants. Good IAQ is vital for health and comfort. Air conditioners help by managing ventilation and maintaining healthier indoor environments.

3. **Refrigerant Management:** Refrigerants are a significant engineering concern in air conditioners due to their environmental impact. Many traditional refrigerants contribute to ozone depletion and global warming when released into the atmosphere. Developing and implementing environmentally friendly refrigerants is crucial to mitigate these effects and ensure sustainable cooling solutions.

4. **Proper Sizing and Selection:** Proper sizing is very important due to the nature of the system. If its too big for the room, the excess moisture and humidity would frost the coils. If its too small for the room, then it would ramp up constantly and never reach the desire set point, causing motor wear, fan wear, high power draw which would cause energy issues.

5. **Thermal Comfort and Load Calculations:** Incorrect thermal comfort and load calculations in an A/C system can lead to discomfort, energy inefficiency, uneven temperature distribution, excessive wear on components, humidity problems, noise issues, and higher costs for adjustments or replacements.

6. **Redundant Control Technologies:** Redundant and traditional control technologies cause limitations and challenges with decrease in energy efficiency. This is discussed in the next subsection.

3.2.1 Traditional Controller and Control Techniques Limitations:

Traditional control techniques in an A/C systems have limitations compared to modern control methods. The traditional control techniques which largely are based on speed drives, frequency drive control of compressors or PID and ON/OFF thermostatic control of the A/C systems are not robust enough to account for the issues of a dynamic and non-linear system.

1. **Limited Adaptability:** Traditional Control techniques rely on fixed temperature set-points and parameters which do not account for variable loads, change in weather

conditions, humidity and temperature variations. An A/C system has to be resistant to such factors.

2. **Energy Inefficiency:** Traditional A/C control techniques are not optimized towards energy efficiency and do not account for load occupancy, as they do not work on predictive algorithms. This causes inefficiency and higher bills.

3. **Linearity:** A/C systems are complex and nonlinear systems based on dynamic parameters such as various sensors, fans, compressors and thermostats. Traditional control techniques such as the PID control require extensive fine tuning which does not equate to the cost of implementation.

4. **Fault Tolerance:** Traditional control techniques lack fault detection capabilities which cause them to be less robust against disturbances.

5. **Limited Integration, Up-gradation Cost and Dependencies:** As IoT is becoming increasingly popular, every device must have internet integration according to protocols. As traditional control techniques have no such capabilities built in, it requires additional overhead cost for their integration up-gradation. Additionally the dependencies on old protocols make them a hurdle to work with and require expert engineering tuning.

It can be concluded that while traditional control techniques have been used for a long time in A/C systems. The factors mentioned above showcases the lack of precision in traditional control techniques and are not well-suited to address the complexities and needs of modern smart building. Energy costs increase every year and efficient machinery is required for comfort while keeping cost minimized. Additionally adaptability is needed in today's A/C systems so they can be up to date with modern protocols. As such modern control methods are required for A/C systems as they leverage advanced algorithms, real-time data, and automation to provide more efficient and comfortable environments.

3.3 Simulation Environment of the System:

The simulation environment is made on MATLAB Simulink platform. The environment consists of the A/C system, and is further subdivided into 3 distinct systems called the sub-systems that simulate the air conditioner workflow, which are namely :

1. Room Model
2. Smart Controller Model
3. A/C Model

Additionally each of these systems have various parameters built in which are imported via the code directory to make them synchronize together with the whole system. The room and air conditioning systems are constant through out the experiments and the smart controller model varies depending on the control technique. Further explanation are pertaining to each of the systems are in the following sub-sections.

3.3.1 Air Conditioner Model:

The Air conditioning model is based on the heat equations (3.3.1,3.3.2, 3.3.3), and the model is inspired by MATLAB's heater model [35] which is transformed to an air conditioner model by using the following heat loss equations. As it can be seen in figure [3.2], the Smart Controller provides a state of ON and OFF which passes through multiplication or AND gate on one input, where-as the other input contains the heat loss equation blocks. The purpose of this methodology is to provide a switching function to our A/C system. This signal is then passed through a low-pass filter, which allows the signal to remove any high frequency noise. This is especially wanted for Reinforcement Learning algorithms as the switching of an RL controller is more frequent than a PI controller. The Energy signal is divided which is then converted to a kWh/Joule signal. This signal is only used for the reinforcement learning model.

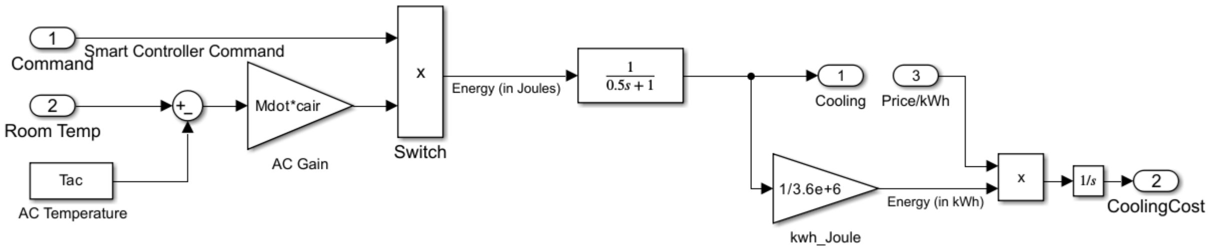

Figure 3.2: A/C Model

$$\dot{Q} = (T_{room} - T_{ac}) \cdot M \cdot c_{air} \tag{3.3.1}$$

where,

$\dot{Q}$ = Heat flow from the A/C

T_{room} = Current Temperature inside the room

T_{ac} = Temperature of cool air from the A/C

M = Air Mass flow rate through the A/C (kg/hr)

c_{air} = Heat capacity of air (at constant pressure)

3.3.2 Room Model:

The Room model used for this study uses the parameters based on NUST-PNEC's power lab. The model of the system is developed using MATLAB's house heating model and the heat loss equations (3.3.1,3.3.2, 3.3.3)[35]. It can be noted from the Heat loss equations and figure [3.3] that the model consists of two differential equations being solved simultaneously. In a MATLAB simulink environment, it is impossible to solve ordinary differential equations without having any prior info placed at run time and during the process if there is an absence of storage, also to note that figure shows the cyclic nature of the two differential equations, as such, a form storage is imperative. For this reason, an "Integrator" block has been placed after the implementation of equation (3.3.2) to allow integration of time in the block environment. Additionally, it is also used to integrate the "$\dot{T}_{room}$" signal for usage in the implementation of equation (3.3.3).

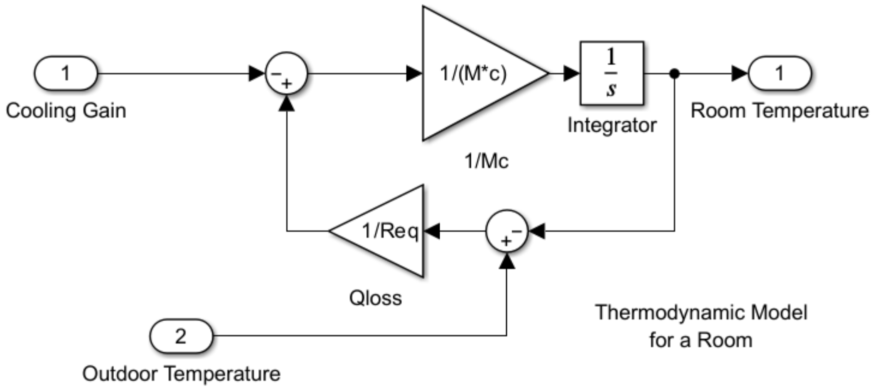

Figure 3.3: Room Model

$$\dot{Q}_{losses} = \frac{T_{out} - T_{room}}{R_{eq}} \tag{3.3.2}$$

$$\dot{T}_{room} = \frac{1}{M_{room} \cdot c_{air}} \cdot (\dot{Q}_{loss} - \dot{Q}_{ac}) \tag{3.3.3}$$

where,

$\dot{Q}$ = Heat flow from the A/C

T_{room} = Current Temperature inside the room

M_{room} = Mass of Air inside the room (kg/hr)

c_{air} = Heat capacity of air (at constant pressure)

R_{eq} = Equivalent thermal resistance of the room

$$C_{Energy} = CurrentCost - PreviousCost \tag{3.3.4}$$

Parameter	**Symbol**	**Value**	**Unit**
Specific Heat Capacity of Air	C_{air}	1005.4	J/kgC
Minimum Comfort Temperature	T_{min}	18	$^{\circ}C$
Maximum Comfort Temperature	T_{max}	25	$^{\circ}C$
Reference Temperature	T_{ref}	24	$^{\circ}C$
Expected A/C Output	T_{ac}	24	$^{\circ}C$
Air Density	ρ	1.2293	kgm^{-3}
Air Mass flow rate through the A/C	$\dot{M}$	1496.0387	kg/hr
Room Length	L_R	3.9624	m
Room Width	W_R	10.8204	m
Room Height	H_R	6.09	m
Window Amount	N_W	4	-
Window Height	H_W	1	m
Window Width	W_W	1	m
Equivalent Room Thermal Resistance	R_{eq}	0.0076	$\frac{^{\circ}C}{W}$
Wall Resistance	R_{Wa}	0.0188	$\frac{^{\circ}C}{W}$
Window Resistance	R_W	0.0128	$\frac{^{\circ}C}{W}$
Mass of Air inside the room	M_{Room}	363.9272	m

Table 3.1: Parameters of the Simulation Environment

3.3.3 Smart Controller:

The smart controller sub-system contains the controller model. This sub-section is the variable part of our study as this changes consistently through-out because of the discussions, simulations and results of various different controllers. As such the smart controllers will be discussed in detail in their sections.

3.4 Parameters of Simulation Environment:

Parameters are values that define the characteristics, behavior, and conditions of the A/C system. They influence how the air-conditioner system behaves and allows customization, altercation to the study for various scenarios without altering the underlying models and subsystems. Table (3.1) showcases the parameters used for our simulation environment.

Chapter 4

PI Controller Model

Proportional-Integral-Derivative (PID) controller is a widely used feedback control mechanism in engineering and industrial applications. It aims to regulate a system's output by adjusting its control inputs based on the error between the desired set-point and the actual process variable. P,I and D are defined as follows.

1. **Proportional (P):** The controller's output is directly proportional to the current error. This term provides a response to changes in the error and helps reduce steady-state error.

2. **Integral (I):** The integral term accumulates the past errors and generates a corrective action that eliminates any residual steady-state error. It is particularly effective in addressing long-term deviations from the set-point.

3. **Derivative (D):** The derivative term is the rate of change of the error. It provides a dampening effect, reducing the controller's sensitivity to rapid changes in the error and improving stability.

4.1 PID Algorithm

The PID algorithm is a control algorithm used in PID controllers to regulate a system's output based on the error between a desired set-point and the actual process variable. It consists of three main terms: Proportional (P), Integral (I), and Derivative (D), which are combined to calculate the controller's output signal. The PID controller is a "error heavy" algorithm as it works on the basis of error reduction from feedback loops. The algorithm's mathematical expression can be represented as follows:

$$u(t) = K_p e_t(t) + K_i \int_0^t e_t(t)\, dt + K_d \frac{de(t)}{dt} \tag{4.1.1}$$

where,

$u(t)$ = The Output Signal.

$e(t)$ = The Error Signal.

K_d = The Derivative Gain.

K_i = The Integral Gain.

K_p = The Proportional Gain.

4.1.1 Omission of Derivative Controller:

In real-life scenarios the Derivative controller is often omitted due to its highly sensitive nature and difficult tuning parameters. In an A/C system, a slight change of sensor erroneous reading can deviate in a large overshoot of derivative controller action. Additionally, for thermal control of an A/C system, the high speed characteristic of Derivative controller is not needed.

1. **Sensitivity to Noise:** The derivative term amplifies high-frequency noise in the error signal, which can lead to erratic and unstable control actions. As real time sen-

sors are subject to noise and disturbances, the derivative term can cause undesirable output especially in mechanical and electrical systems like an A/C system

2. **Difficult Tuning:** The derivative term's effect on the control action is based on the rate of change of the error, tuning gain K_d accurately can be challenging, as it requires an understanding of the system dynamics and error fluctuations. Incorrect tuning of the derivative term can lead to oscillations and instability.

3. **Limited Applicability:** Derivative control is particularly effective in systems with rapid changes in error. While an A/C system has rapid switching, we want to reduce the error output caused by switching as much as possible while reducing switching itself. As the derivative controller has abrupt changes, it can prove to lack smoothness for mechanical and inconsistent for the electrical systems. Therefore the derivative controller is not suitable.

Due to these challenges and considerations, A/C systems opt to use proportional-integral (PI) controllers instead of PID controllers. This allows for less difficulty in tuning, less abrupt switching behavior which in turn allows for smooth and consistent fan control and more resistant to noise.

4.2 PI Model

The algorithm for a PI model is a simplified version of the full PID algorithm that omits the derivative term. This is due to stability, efficiency and relatively high accuracy of the PI controller due to the inherent linear properties, allowing for balanced changed in error [36] and proves to have easier tuning than the derivative controller as mentioned in previous section. The following is the mathematical expression for a PI controller, with the K_p and K_i gains along with the error term $e_t(t)$ which provides a linear solution.

$$u(t) = K_p e_t(t) + K_i \int_0^t e_t(t)\, dt \tag{4.2.1}$$

The PI Controller model tested for our system is shown in figure(4.1). The system model discussed in Chapter 4, will be further discussed in Chapter 5. Additionally the results of our finding will be explained in the Results Chapter.

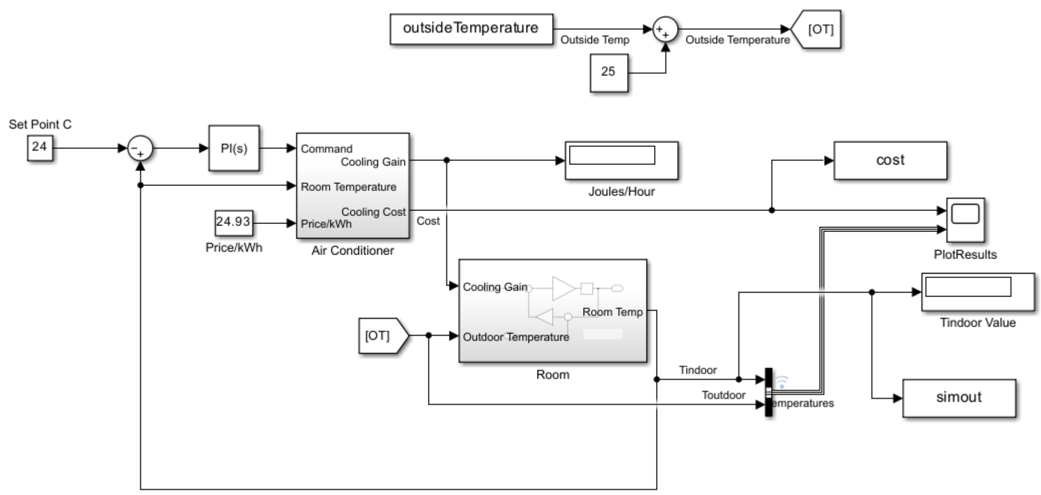

Figure 4.1: System Model for Traditional PI controller

Chapter 5

Reinforcement Learning based models

Reinforcement Learning (RL) is part of machine learning where the concern is with how an agent can learn to make sequential decisions in an environment to maximize a cumulative reward. It is inspired by human behavior and is designed to mimic the way humans and animals learn by interacting with their surroundings.

In RL, an agent learns by trial and error. The agent takes actions in an environment, receives feedback in the form of rewards or penalties, and uses this feedback to improve its decision-making over time. The ultimate goal of the agent is to find a strategy, called a policy, that guides its actions in a way that maximizes the expected long-term cumulative reward.

5.1 RL Controller model:

For the application of a RL controller model, an important assertion is made the problem to be considered as sequential time decision making process. For every time step, the current state is studied in relation to the previous state. In the case of an A/C system, the current indoor temperatures relation to previous time interval only and does not consider earlier intervals. By optimizing the RL controller reward function with respect to designated set point, thermal comfort can be maximized and energy cost be minimized.

The RL controller scheme for an A/C system, as seen in figure(5.1) needs to have the following elements defined:

1. **State (S):** A representation of the current situation or condition of the environment. It provides the necessary information for the agent to make decisions. In some cases, a history of previous states might be necessary for decision-making.

2. **Action (a):** The learner or decision-maker that interacts with the environment. The agent's goal is to learn an optimal policy that helps it take actions to maximize its cumulative reward.

3. **State transition probability ($P(s'|s,a)$):** The state transition function describes the probability distribution over next states given the current state and action.

4. **Reward (R):** The scalar feedback the agent receives from the environment after taking an action. Rewards indicate the immediate benefit or cost associated with an action and are used by the agent to learn which actions are favorable.

5. **Policy (π):** A strategy that maps states to actions. The policy defines how the agent should behave in different states to achieve its goal of maximizing cumulative reward.

6. **Reward Function:** A function that estimates the expected cumulative reward that can be obtained from a given state under a certain policy. It helps the agent evaluate the desirability of different states.

Since the state transition probability calculation is a non-trivial task to accomplish, this has been ignored in the formulation of this controller as the model of an A/C

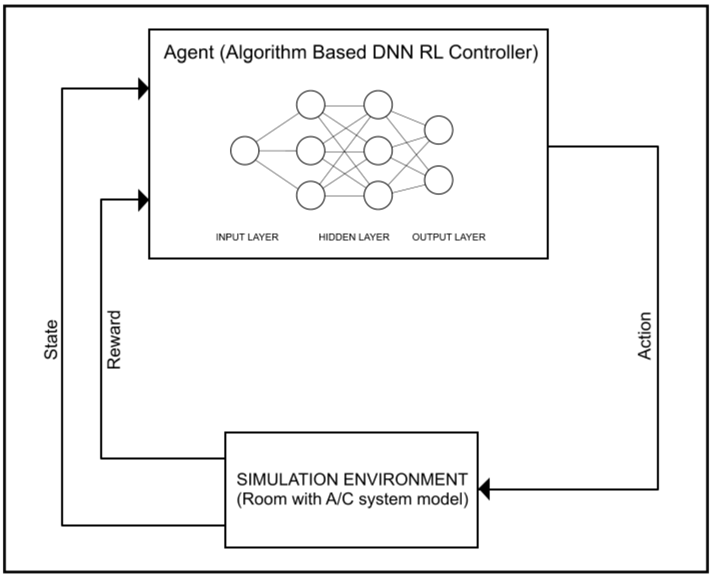

Figure 5.1: RL-Based approach for A/C system. DNN: deep neural network

system is dynamic and sensitive to slight changes in any external or internal parameters such as sensor sensitivity, resistances of a given room, weather conditions, outdoor temperature, UV radiation, electrical disturbances; and so on.

However for any controlled environment, an analytical solution can be configured if the state transition probability function $P(s'|s,a)$ based on system dynamics is known or calculated before hand. However, this is not a generalized solution and we require a robust controller for efficient A/C control. The subsequent dynamics of the air conditioner system are discussed in Chapter 3.

Consider the action space a vector for the air conditioning agent.

$$A = a, a = [0, 1] \tag{5.1.1}$$

Where, 0 is OFF and 1 is ON for switching. Additionally the observation or state space is given as:

$$S = [T_{max}, T_{min}, T_{room}, T_{out}, C_{Energy}, A_{pa}] \tag{5.1.2}$$

where,

1. Maximum Comfort Temperature (T_{max})
2. Minimum Comfort Temperature (T_{min})
3. Current Indoor Temperature (T_{room})
4. Current Outdoor Temperature (T_{out})
5. The retail per unit cost (C_{Energy}) in Pakistani Rupees.
6. Agent Output (Previous Action)(A_{pa})

5.1.1 Agent Policy:

The policy of an agent is defined as the behavior of the agent. It tells how the agent is mapping its output from the given reward function and observation states. The policy of an agent can either be deterministic (for quicker output) or stochastic (for larger exploration), in the case of air conditioning system, the action state has only two plausible outcomes, either ON or OFF, which causes it to be deterministic. This can be expressed as:

$$\pi = max(f_\theta(s)) \tag{5.1.3}$$

5.1.2 RL Terminology:

Reinforcement Learning has a lot of terminologies that need explanation, it is to be noted that not all RL agents make use of all the stated terminologies, but in the case of this study, the important terminologies are explained.

5.1.2.1 Episode:

In RL, episode refers to the sequence of actions that the agent takes on the environment, that starts from an initial state and ends when the termination condition is reached, whether if the Episode Reward converges or max specified episode count is achieved. The steps of an Episode are as follows:

1. **Initialization:**
   - The episode begins with the agent receiving an initial state from the environment.
   - The agent uses its policy to select an action based on the current state.
2. **Action and Transition:**
   - The agent takes the selected action and interacts with the environment.
   - The environment transitions to a new state based on the current state and action taken. This transition is governed by the state transition function $P(s'|s, a)$.
   - The agent receives a reward from the environment based on the reward function $R(s, a)$.
3. **Iteration:**
   - The process of selecting actions, transitioning between states, and receiving rewards is repeated in subsequent time steps. This continues till the termination condition is met
4. **Termination:**
   - The episode terminates when a predefined condition is reached. This condition can vary based on the problem setting and goals. This can be if the Episode Reward converges or max specified episode count is achieved.

5. **Policy Update:**

   - Based on the reward provided, the agent updates the policy for future states, actions for further episodes.

This process is then repeated till the termination condition is achieved.

5.1.2.2 Average Episode Reward:

Average Episode Reward is a simple and robust criterion for Episode termination. It is used as a means to achieve precise measure of performance. Additionally easier condition to achieve because of convergence. The A/C while having dynamic states, the end action is the temperature controller based on switching principle and provides an early stopping of the algorithm when a suitable condition is met or if its highly stable.

5.1.2.3 Learning Rate:

The learning rate in machine learning algorithms, including reinforcement learning, is a hyper-parameter that determines the step size at which the algorithm adjusts its model parameters based on the gradient of the loss function. A small learning rate is of 1×10^{-3} is used for convergence, robustness to noisy data, steadiness to local minima's and avoidance of oscillations.

5.1.2.4 Q-Values:

Q-Value or Q-Learning is the initialization of the learning agent at the beginning of the Episodes. They help balance between exploration and exploitation. The agent explores different actions in various states to gather information about the environment and it provides a benchmark to the agent to allow exploration around a given range and maximizes learning from the environment.

5.1.2.5 Actor-Critic Pair

An Actor-Critic pair is a fundamental strategy used in some reinforcement learning (RL) algorithms. They are defined as follows.

1. **Actor:**
   - The actor is responsible for learning and updating the policy. It directly interacts with the environment by selecting actions based on the current state.
   - The actor's goal is to learn a policy that maximizes the expected cumulative reward over time.
   - The actor is the agent's decision-maker, determining what actions to take in different states.
2. **Critic:**
   - The critic evaluates the actions chosen by the actor. It estimates the value or expected cumulative reward that the agent can achieve from a particular state while following the actor's policy.
   - The critic's role is to provide feedback to the actor by providing information on how good the chosen actions are in different states.
   - The critic helps guide the learning process by indicating which actions are more favorable in terms of expected reward.

The interaction between the actor and the critic forms a loop where the actor learns to make better decisions based on the critic's evaluations, and the critic's evaluations become more accurate as the actor's policy improves over time. Some algorithms do not use actor-critic pair and one of these is the DQN algorithm.

5.2 Controller Models:

Controller models using the RL algorithms are "off-policy" or "no-policy" algorithms. The decision of such algorithms is based on already having a data-set to train the RL agent. It is to be noted that off-policy and no-policy are two distinct algorithm choices, they have clear differences, but the ability of utilizing data-set on both algorithms is considered a key-point in this discussion. The feature similarities of such algorithms based on the basis of preexisting data-set are as listed.

1. **Utilizing Historical Data:** Both off-policy and no-policy algorithms capitalize on preexisting data-sets, leveraging prior interactions with the environment to guide decision-making.

2. **Optimal Reward Maximization:** Both approaches aim to maximize cumulative rewards over time by making well-informed choices based on the provided data. Refer to section **RL Terminology**.

3. **Framework of Reinforcement Learning:** Both strategies operate within the framework of reinforcement learning, which entails learning from interactions in an environment.

Based on these characteristics, strictly one of each, no-policy and off-policy algorithms for our controllers are chosen based on extensive research. There are (1)RL Controller Models testing both DQN and TD3 algorithms and (2)Reinforcement Learning based Proportional Integral (RL-PI) Controller models using only TD3 algorithm.

The objective of our control strategy is to employ a model free RL-based control system and evaluate the performance. However before the implementation, a few things must be considered.

1. RL is a trial and error based method.

2. These methods must be pre-trained before employment, as they required to be trained.
3. Due to the local maxima and minima nature of RL problems, the RL system can take random actions at its initial learning stage

5.2.0.1 The Reward Function:

The reward function of the subsequent models are based on the observation space and the action space. This is given by $R(s, a)$. The inputs of the reward function takes some of the observation space, the current action, the previous action and outputs a reward. Hence the reward function takes on the form of:

$$R = [T_{max}, T_{min}, T_{room}, T_{out}, C_{Energy}, A, A_{pa}] \tag{5.2.1}$$

$$w = -0.01 \tag{5.2.2}$$

The reward function takes the above inputs and performs the following mathematical instructions. This is the reward function algorithm. The weight "w" assumes the value of -0.01 which are negative to showcase penalization. Small weights allow stability and are scale-able, which allows the algorithm to explore more, which in turn allows convergence. The reward function is inspired by [37] and [3].

$$R = R_c + C_{Energy} - A_p \tag{5.2.3}$$

$$C_{Energy} = UnitPrice \cdot ElectricityConsumed \tag{5.2.4}$$

$$R_c = \begin{cases} 0.1 & T_{min} \leq T_{room} \leq T_{max} \\ w|T_{room} - T_{min}| & T_{room} < T_{min} \\ w|T_{room} - T_{max}| & T_{room} > T_{max} \end{cases} \tag{5.2.5}$$

No.	Parameter	Value
1	Max Episodes	150
2	Training Termination Criterion (Episode Value)	200
3	Learning Rate	0.001
4	Epsilon Greedy Decay Rate	0.0001
5	Sample Time	120
6	Max Steps Per Episode	1000
7	Use RNN	True
8	Experience Buffer	100000

Table 5.1: DQN Agent Parameters

$$A_p = \begin{cases} -0.1 & a_t \neq a_{t-1} \\ 0 & otherwise \end{cases} \tag{5.2.6}$$

The next chapter discusses the results of our findings based on the explained models.

5.2.0.2 Deep-Q Network Algorithm:

Deep-Q Network (DQN) is a no-policy RL algorithm and an amalgamation of DNN and Q-Learning. Q-Learning agent is a value-based RL agent that trains a critic and estimates future returns or rewards. For any given number of observations, the agent gives an output action which estimates the greatest return. The difference between Q-Learning and DQN is the agents usage of DNN for DQN instead of a lookup table called the Q-Table. This generalizes the problem statement and allows robust solutions. It has a discrete action space and is suitable for replacing on-off control scenarios.

The idea behind DQN is to build a neural network table and estimate greatest returns and rewards on the basis of Q-Value. Higher Q-Values dictates higher returns. The selection criteria of the Q-Values is conditioned by the neural network.

No.	Parameter	Value
1	Max Episodes	150
2	Training Termination Criterion (Episode Value)	85
3	Learning Rate	0.001
4	Sample Time	120
5	Max Steps Per Episode	1000
6	L2 Regularization Factor	0.00001

Table 5.2: TD3 Agent Parameters

5.2.0.3 Twin-Delayed Deep Deterministic Policy Gradient Algorithm:

The Reinforcement Learning Twin-delayed deep deterministic policy gradient (TD3) is a model free, off-policy algorithm which consists of actor-critic combinations to search for optimal solution policy that maximizes an expected long-term cumulative reward.

The TD3 agent is a modification to the DDPG (Deep Deterministic Policy Gradient) agent. The issues with DDPG agent is the overestimation of value functions which provides sub-optimal solutions. As such the modifications are:

1. Consists of two critic and one actor pair. This allows for reduction of Q-Value bias and chooses the minimum value estimate to update its policy.
2. Noise is considered to allow for unbiased policy updates.
3. Policy updates are less frequent than its critics.
4. Addresses over-estimation.

In DQN, the action space is strictly discrete. This only allows for a discrete range of policy updates or control actions for a given set-point. Example, a range of set-points between 24 and 28 is given as {24,25,26,27,28} and agent will only learn from those action spaces. The TD3 agent provides a continuous action space, this allows for a fuzzy generalized action space that can help stabilize a given strategy.

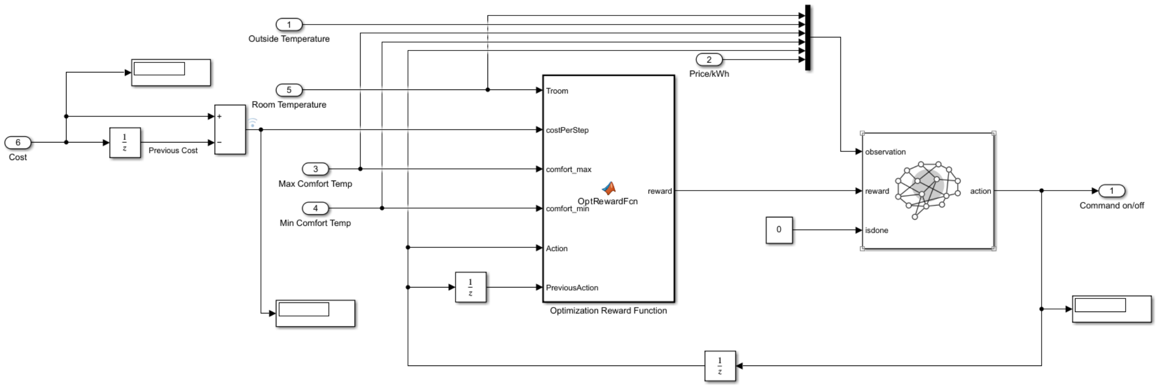

Figure 5.2: DQN and TD3 Based Controller Model

Controllers	Observation Space S	Reward Function Inputs
DQN	$[T_{max}, T_{min}, T_{room}, T_{out}, C_{Energy}, A_{pa}]$, 6	$[T_{max}, T_{min}, T_{room}, C_{PerStep}, A, A_{pa}]$, 6
TD3	$[T_{max}, T_{min}, T_{room}, T_{out}, C_{Energy}, A_{pa}]$, 6	$[T_{max}, T_{min}, T_{room}, C_{PerStep}, A, A_{pa}]$, 6
PI	$[Kp, Ki]$,2	$[T_{max}, T_{min}, T_{room}, C_{PerStep}, A, A_{pa}]$, 6

Table 5.3: Observation and Reward Function Space for respective controllers

5.2.1 RL-PI Controller:

The RL PI controller utilizes the observation space $S = [Kp, Ki]$ and the reward function being shown in equation (5.2.3 - 5.2.6). The model can be seen in figure(5.3)

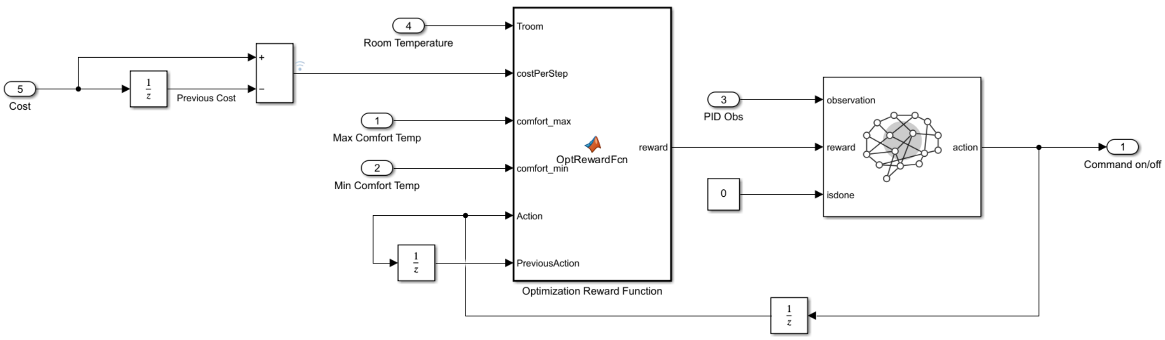

Figure 5.3: RL-PI based Controller Model

Chapter 6

Results

In figures(6.1, 6.2, 6.3) and Table(6.1) for the TD3 and DQN agent showcase the temperature curve and the cost of each of the experiments using TD3 agent per 1440 minutes for a total of 4320 minutes. As it can be seen, the temperature response remained inside the temperature boundaries for the entirety of the experiment. The cost of the third experiment is only higher because of 8 more temperature added for simulation purpose. The stopping criterion of the TD3/DQN algorithm is when the average reward reaches 85/200 respectively. The training for both algorithms stop very early at an average of 6 to 8 episodes, the rest of the experiments showcase robustness of the system as shown in figures (6.5, 6.4).

Table (6.2) showcases the mean cost, and total difference along with percentile difference between both algorithms. It can be seen that with a 1.06903% there is no difference between the two algorithms and any of them can be used for air conditioner

Exp No.	**TD3**	**DQN**
1	824.6003	821.2656
2	777.9522	775.6900
3	1633.05505	1633.3272
Total	3,265.60755	3230.8828

Table 6.1: TD3/DQN Agent Cost out of total 1440 minutes. **Note:** Third row signifies higher cost due to positive 8 temperature added for simulation purposes.

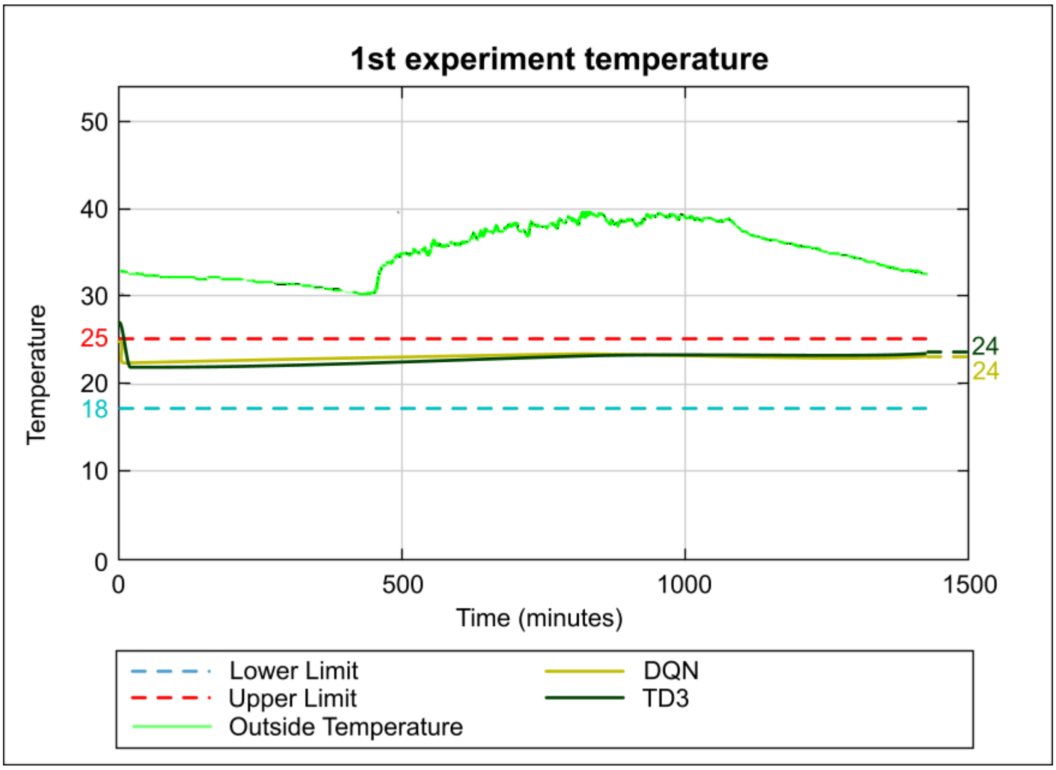

Figure 6.1: First experiment of DQN/TD3 agent.

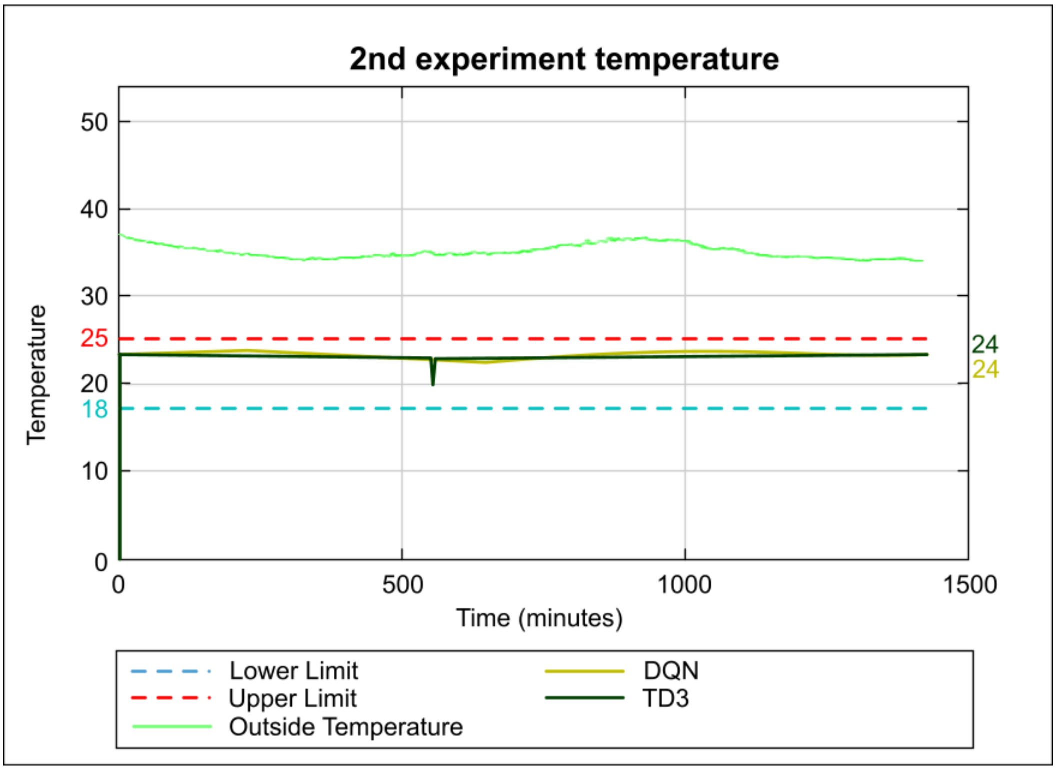

Figure 6.2: Second experiment of DQN/TD3 agent

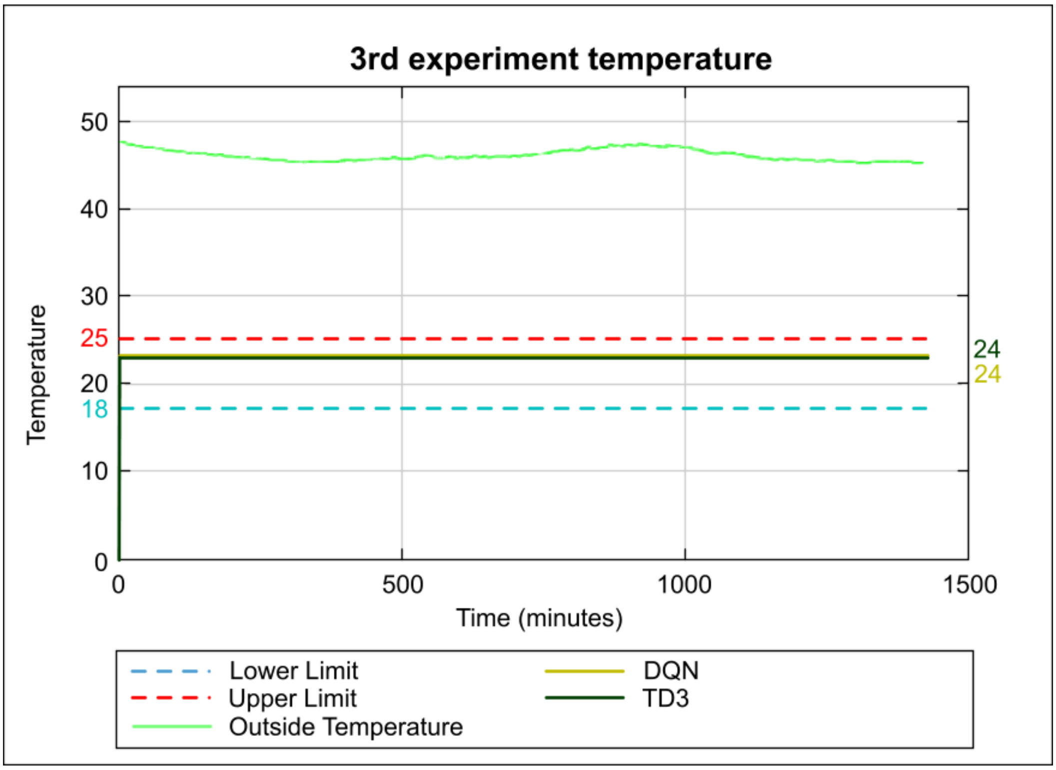

Figure 6.3: Third experiment of DQN/TD3 agent

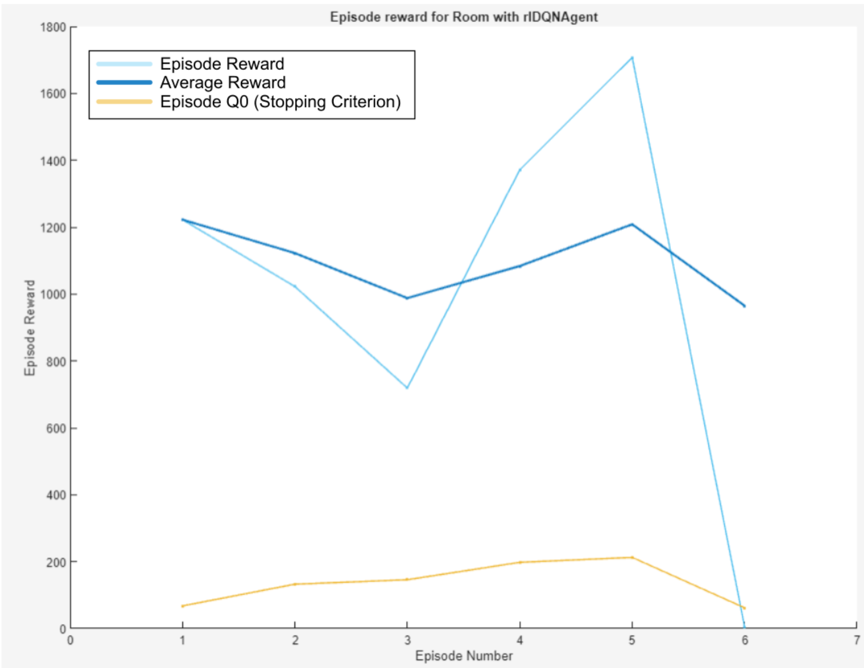

Figure 6.4: Episode Graph of DQN agent

problem.

Since it was concluded that there is minute difference between DQN and TD3

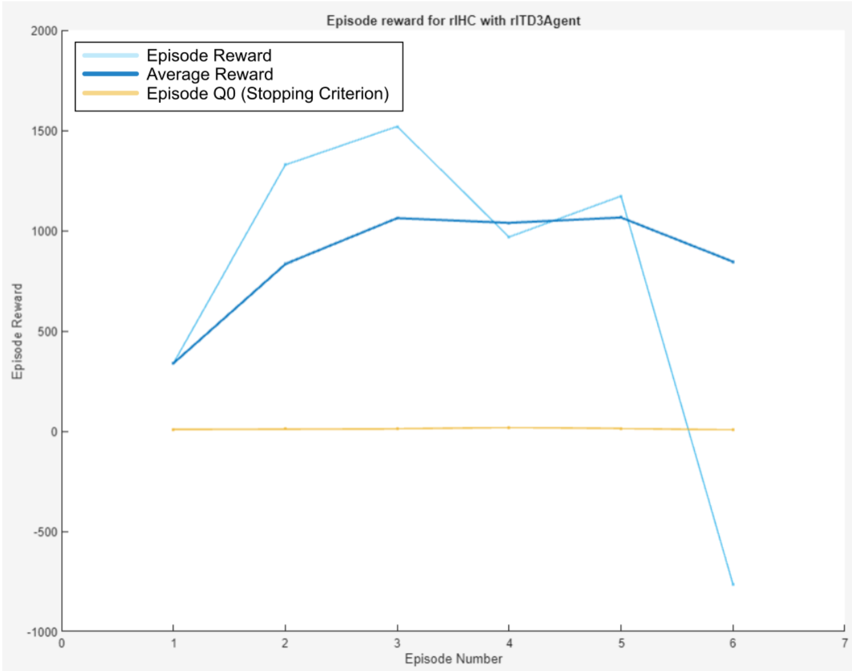

Figure 6.5: Episode Graph of TD3 agent

Algorithm	**Mean Cost**	**Percentile**
DQN	3230.8828	-
TD3	3,265.6075	-
Difference	34.7247	1.06903%

Table 6.2: Difference between both algorithms.

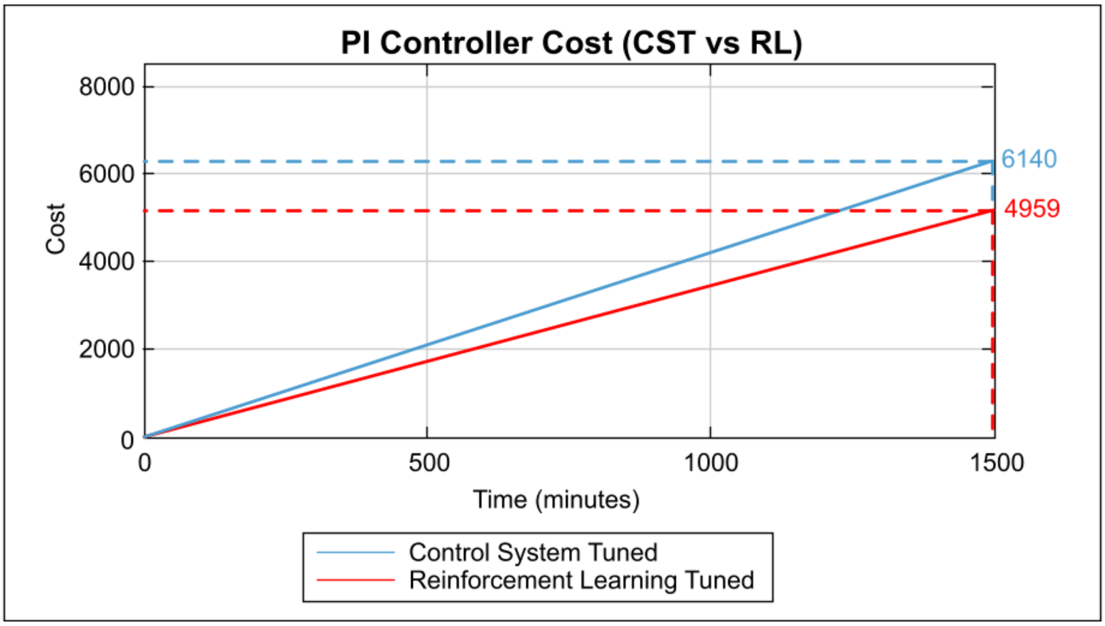

Figure 6.6: Cost graph of traditional control technique.

PI Tuning	Cost
Control System Tuner	6140.8026
Reinforcement Learning	4959.1529

Table 6.3: Cost of PI Controller tuned by RL and inbuilt CST of MATLAB. The experiment is ran for a total of 1440 minutes.

No.	Controller	Cost Difference	Percentile Difference
1	DQN vs RL-PI	4137.8873	143.169%
2	DQN vs CST-PI	5319.537	152.815%
3	TD3 vs RL-PI	4134.5526	142.971%
4	TD3 vs CST-PI	5316.2023	152.646%

Table 6.4: Cost difference of the different controller models. Only the first experiment of each is considered

agent (0.0107%), it was decided to implement TD3 algorithm for the the RL PI experiment. Additionally for simplicity, comfort violation was not considered for this experiment. In table(6.3) and figure(6.6), it can be seen that both cost curves are approximately linear, with an accumulated cost show in table(6.3). However it can be seen from figure(6.7) that both controllers do not achieve the desired 24°C temperature and overshoot the response curve and are always violating the comfort ranges, hence not provided in the tables.

The episode training graph of RL-PI method can be seen on figure(6.8). It can be deduced that there was sufficient difficulty on the convergence of the system at the first 17 episodes. This can be due to the factors of the $[Ki, Kp]$ parameters of the RL-PI controller that proved difficult.

Table(6.4) showcases the cost difference the percentile difference between newer RL based control techniques compared to traditional control techniques. It is to be noted that only the first experiment cost of each control strategy was used to formulate this conclusion. The differences show that for better and robust controllers, its better to use pure DQN or TD3 based controllers without traditional controller technology.

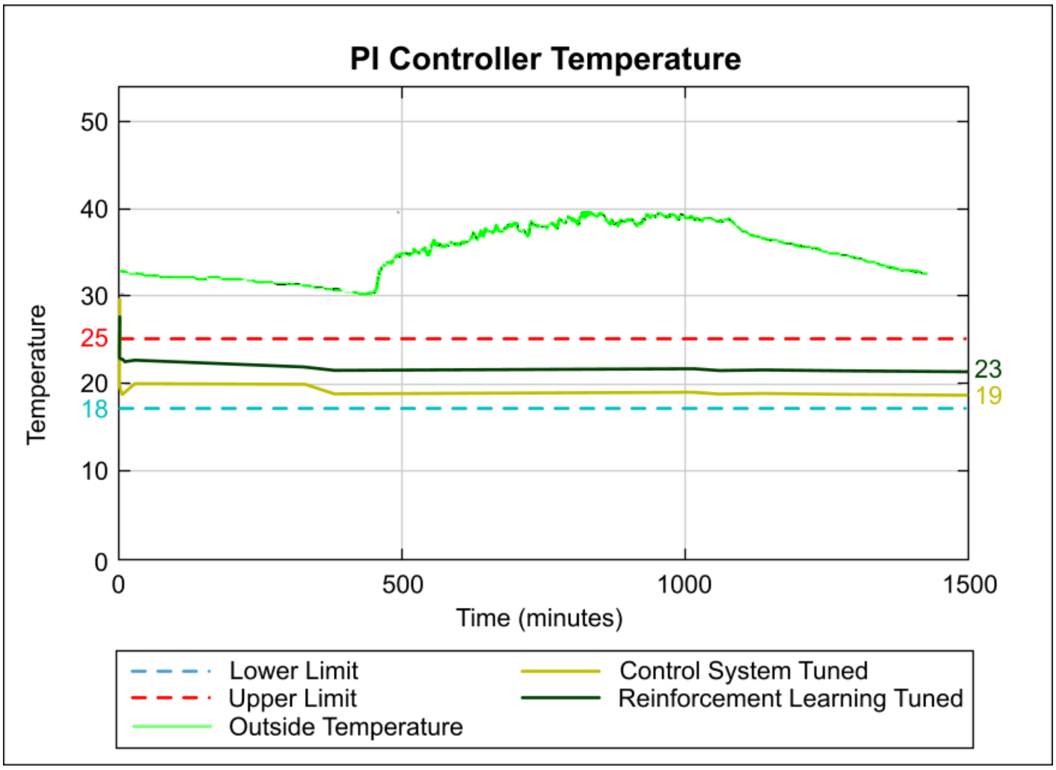

Figure 6.7: Temperature response of the PI controller systems using CST and RL methods.

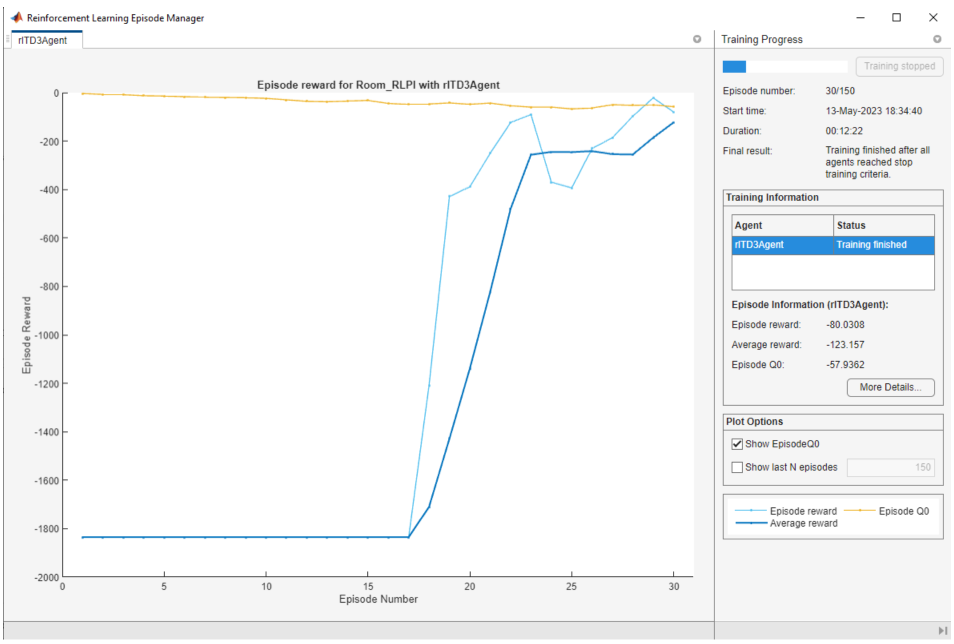

Figure 6.8: Episode graph of RL-PI controller using TD3 agent

Conclusion

In conclusion, this study has examined various methods and procedures of cost-effective thermal comfort management and energy reduction. We initially posed a question on the comparative analysis between modern control algorithms (Reinforcement Learning based) versus traditional control algorithms (PI, PID, ON and OFF) for optimization and through thorough investigation we have found conclusive results.

Our study has shown that modern control techniques provide a more robust solution to the A/C market compared to the traditional techniques. The simulations showcase the efficiency of the algorithms and the efficacy they provide. As seen in this study before, there is an notable difference of about 142-143% for DQN and roughly 152% for TD3 between RL based and traditional control methodologies. Additionally it was discovered that there is a minute difference between TD3 and DQN control techniques, which is about 1.07%. Conversely the training of such systems is not overly costly or time-consuming.

However, it is important to note the limitations of the study as well. Our research was confined to only the temperature variable of the A/C system, additionally our study was only simulation based, and the experiments performed were up to three only. For more rigorous results, more experiments should be performed.

In closing, our study showcases the vital importance modern control techniques of A/C systems. The efficiency of modern control techniques out perform traditional control techniques in many ways, and provide more generalized integration between other technologies that IoT is a part of. As lifestyle changes and as more people pursue the usage of A/C systems, it imperative for A/C manufacturers to start integrating modern control techniques for efficiency and energy saving.

Future Recommendation

As noted in the Conclusion section, there are some limitations to this study that researchers can work and improve upon. These are listed as follows:

- **Experiment Amount:** Future researchers should increase the amount of experiments to highlight even more robustness of the system
- **Practical Implementation:** Practical implementation should be done on A/C systems to see the correlation of the theory to the practicality. This would provide even more conclusive evidence between modern vs traditional control techniques.
- **Reward Function Optimization:** The reward function used in this study is quite simple. An even more robust reward function can be developed to penalize the Reinforcement Learning agent.
- **Extension:** The study can be extended to newer technologies as well. The researcher should try to implement the physical model of the A/C system as opposed to the mathematical model we have done in this study to showcase any difference there might be.

While more work can be done, as the A/C and HVAC subject is very vast, these are one of the few ways according to the author that can be worked upon for future research based on this study.

References

[1] F. M. Pakistan, "Pakistan economic fy economic report - energy - 2023-2024," *[Online]*, p. 227, June 2023.

[2] CEIC, "Pakistan electricity consumption: Total." https://www.ceicdata.com/en/pakistan/electricity-generation-and-consumption/electricity-consumption-total#:~:text=Pakistan%20Electricity%20Consumption%3A%20Total%20data%20is%20updated%20yearly%2C%20averaging%2068%2C987.000,of%2031%2C534.000%20GWh%20in%201991. Accessed: 01, 05, 2024.

[3] Y. Du, F. Li, K. Kurte, J. Munk, and H. Zandi, "Demonstration of intelligent hvac load management with deep reinforcement learning: Real-world experience of machine learning in demand control," *IEEE Power and Energy Magazine*, vol. 20, no. 3, pp. 42–53, 2022.

[4] Y. Long, S. Liu, L. Xie, and K. H. Johansson, "A hierarchical distributed mpc for hvac systems," in *2016 American Control Conference (ACC)*, pp. 2385–2390, 2016.

[5] D. T. Vedullapalli, R. Hadidi, and B. Schroeder, "Optimal demand response in a building by battery and hvac scheduling using model predictive control," in *2019 IEEE/IAS 55th Industrial and Commercial Power Systems Technical Conference (ICPS)*, pp. 1–6, 2019.

[6] D. Xie, L. Yu, T. Jiang, and Y. Zou, "Distributed energy optimization for hvac systems in university campus buildings," *IEEE Access*, vol. 6, pp. 59141–59151, 2018.

[7] A. Parisio, M. Molinari, D. Varagnolo, and K. H. Johansson, "A scenario-based predictive control approach to building hvac management systems," in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 428–435, 2013.

[8] M. Ostadijafari and A. Dubey, "Linear model-predictive controller (lmpc) for building's heating ventilation and air conditioning (hvac) system," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 617–623, 2019.

[9] C. Anuntasethakul and D. Banjerdpongchai, "Design of supervisory model predictive control for building hvac system with consideration of peak-load shaving and thermal comfort," *IEEE Access*, vol. 9, pp. 41066–41081, 2021.

[10] P. M. Ferreira, S. M. Silva, and A. E. Ruano, "Energy savings in hvac systems using discrete model-based predictive control," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2012.

[11] M. Al-Azba, Z. Cen, and S. Ahzi, "Air-conditioner on-off optimization control under variant ambient condition," in *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 1–5, 2018.

[12] K. Chinnakani, A. Krishnamurthy, J. Moyne, A. Arbor, and F. Gu, "Comparison of energy consumption in hvac systems using simple on-off, intelligent on-off and optimal controllers," in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–6, 2011.

[13] C.-S. Kang, J.-I. Park, M. Park, and J. Baek, "Novel modeling and control strategies for a hvac system including carbon dioxide control," *Energies*, vol. 7, no. 6, pp. 3599–3617, 2014.

[14] V. Congradac and F. Kulic, "Hvac system optimization with co2 concentration control using genetic algorithms," *Energy and Buildings*, vol. 41, no. 5, pp. 571–577, 2009.

[15] K. Fong, V. Hanby, and T. Chow, "Hvac system optimization for energy management by evolutionary programming," *Energy and Buildings*, vol. 38, no. 3, pp. 220–231, 2006.

[16] K. H. Khan, C. Ryan, and E. Abebe, "Optimizing hvac energy usage in industrial processes by scheduling based on weather data," *IEEE Access*, vol. 5, pp. 11228–11235, 2017.

[17] C. Zhenyu, Y. Bin, M. Jinjie, P. Yongtao, and L. Yusun, "Global optimization model for hvac systems in public commercial buildings and its schedulable potential evaluation," in *2020 5th Asia Conference on Power and Electrical Engineering (ACPEE)*, pp. 1989–1994, 2020.

[18] N. Chakraborty, A. Mondal, and S. Mondal, “Multiobjective optimal scheduling framework for hvac devices in energy-efficient buildings,” *IEEE Systems Journal*, vol. 13, no. 4, pp. 4398–4409, 2019.

[19] Y.-E. Jang, Y.-J. Kim, and J. P. S. Catalão, “Optimal hvac system operation using online learning of interconnected neural networks,” *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3030–3042, 2021.

[20] L. Lu, W. Cai, L. Xie, S. Li, and Y. C. Soh, “Hvac system optimization—in-building section,” *Energy and Buildings*, vol. 37, no. 1, pp. 11–22, 2005.

[21] G. Demirezen, A. S. Fung, and M. Deprez, “Development and optimization of artificial neural network algorithms for the prediction of building specific local temperature for hvac control,” *International Journal of Energy Research*, vol. 44, no. 11, pp. 8513–8531, 2020.

[22] H. T. Dinh and D. Kim, “Milp-based imitation learning for hvac control,” *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6107–6120, 2022.

[23] M. Li, Y. Li, and X. Min, “Practice and application of lstm in temperature prediction of hvac system,” in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1000–1004, 2020.

[24] S. A. A. Rizvi, A. J. Pertzborn, and Z. Lin, “Reinforcement learning based optimal tracking control under unmeasurable disturbances with application to hvac systems,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.

[25] J. Deng and H. Wang, “Modeling and optimizing building hvac energy systems using deep neural networks,” in *2018 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*, pp. 181–185, 2018.

[26] H. Huang, L. Chen, and E. Hu, “A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study,” *Building and Environment*, vol. 89, pp. 203–216, 2015.

[27] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, “Artificial neural network (ann) based model predictive control (mpc) and optimization of hvac systems: A state of the art review and case study of a residential hvac system,” *Energy and Buildings*, vol. 141, pp. 96–113, 2017.

[28] A. Endo, O. Cartagena, D. Sáez, and D. Muñoz-Carpintero, "Predictive control based on fuzzy optimization for multi-room hvac systems," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, 2020.

[29] J. R. Dobbs and B. M. Hencey, "Predictive hvac control using a markov occupancy model," in *2014 American Control Conference*, pp. 1057–1062, 2014.

[30] Nasruddin, Sholahudin, P. Satrio, T. M. I. Mahlia, N. Giannetti, and K. Saito, "Optimization of hvac system energy consumption in a building using artificial neural network and multi-objective genetic algorithm," *Sustainable Energy Technologies and Assessments*, vol. 35, pp. 48–57, 2019.

[31] R. Talib, N. Nabil, and W. Choi, "Optimization-based data-enabled modeling technique for hvac systems components," *Buildings*, vol. 10, no. 9, 2020.

[32] G. Qi, Z. Zhu, K. Erqinhu, Y. Chen, Y. Chai, and J. Sun, "Fault-diagnosis for reciprocating compressors using big data and machine learning," *Simulation Modelling Practice and Theory*, vol. 80, pp. 104–127, 2018.

[33] I. Ullah and D. Kim, "An improved optimization function for maximizing user comfort with minimum energy consumption in smart homes," *Energies*, vol. 10, no. 11, 2017.

[34] J. Varrasi, "Global cooling: The history of air conditioning," Jun 2011.

[35] MATLAB, "Matlab version: 9.13.0 (r2023b)," 2023.

[36] R. Haines and M. Myers, *HVAC Systems Design Handbook, Fifth Edition.* McGraw Hill LLC, 2009.

[37] MATLAB, "Matlab version: 9.13.0 (r2023b)," 2023.

Appendices

Reset Functions and Validation functions are taken from MATLAB examples and can be found there.

.1 DQN Code:

```
disp("Loading Model...")
mdl = "A/C_LAB_MODEL";
open_system(mdl)
sample_time = 120; % seconds
PerEpisodeStep_max = 1000;
agentBlk = mdl + "/Smart Controller/RL Agent";

disp("Logging Data for Temperature...")
data = load('temperatureMar21toApr15_2022.mat');
temperatureData = data.temperatureData;
%Validation Data
temp_march21 = temperatureData(1:60*24,:);
temp_april15 = temperatureData(end-60*24+1:end,:);
%Training Data
temperatureData = temperatureData(60*24+1:end-60*24,:);

%% Room parameters
disp("Loading House Parameters...")
outsideTemperature = temperatureData;
max_comfort = 18;
min_comfort = 25;
Tac = 24;
rad2deg = 180/pi;
% Define the house geometry
% House length = 30 m, my room = 13 ft, lab = 16.9672-6.1976-3.81 (m)
L_House = 6.9596; %in meters
% House width = 10 m , my room = 10 ft, lab = 3.5814 + 7.239 (m)
```

```
W_House = 10.8204; %in meters
% House height = 4 m , ,my room = 10 ft, lab = 6.09m
H_House = 6.09; %in meters
% Roof pitch = 5 deg
roof_pitch = 5/rad2deg;
% Number of windows = 6, lab = 4, og = 1
n_win = 4;
% Height of windows = 1 m
H_win = 1.3;
% Width of windows = 1 m
W_win = 1.3;
windowArea = n_win*H_win*W_win;
wallArea = 2*L_House*H_House + 2*W_House*H_House + ...
           2*(1/cos(roof_pitch/2))*W_House*L_House + ...
           tan(roof_pitch)*W_House - windowArea;
wall_coefficient = 0.038;   % hour is the time unit
wall_density = .2;
wall_resistance = wall_density/(wall_coefficient*wallArea);
kWindow = 0.78;  % hour is the time unit
LWindow = .01;
RWindow = LWindow/(kWindow*windowArea);
R_wall_eq = wall_resistance*RWindow/(wall_resistance + RWindow);
% c = cp of air (273 K) = 1005.4 J/kg-K
%c_air = 1000;
%Mdot = 2655;  % hour is the time unit
Mdot = 1496.0387502928907;
c_air = 1005.4;
densAir = 1.2250;
M = (L_House*W_House*H_House+tan(roof_pitch)*W_House*L_House)*densAir;

%% Setting up specifications
obsInfo = rlNumericSpec([6,1]);
actInfo = rlFiniteSetSpec([0,1]); % (0=off,1=on)
```

```
rng(0)

%% Create DQN Agent with LSTM Network
disp("Making Critic Agent...")
critic_options = rlOptimizerOptions( ...
    LearnRate=0.001, ...
    GradientThreshold=1);

agent_options = rlDQNagent_options(...
    UseDoubleDQN = false, ...
    TargetSmoothFactor = 1, ...
    TargetUpdateFR_wall_equency = 4, ...
    expBufferLength = 1e6, ...
    CriticOptimizerOptions = critic_options, ...
    MiniBatchSize = 64);

agent_options.EpsilonGreedyExploration.EpsilonDecay = 0.0001; %0.0001 og value

useRNN = true;
initialize_options = rlAgentInitializationOptions( ...
    UseRNN=useRNN, ...
    NumHiddenUnit=64);

if useRNN
    agent_options.SequenceLength = 20;
end
agent = rlDQNAgent(obsInfo, actInfo, initialize_options, agent_options);
agent.sample_time = sample_time;

%% Simulink Env
disp("Simulating Env...")
env = rlSimulinkEnv(mdl,agentBlk,obsInfo,actInfo);
env.ResetFcn = @(in) hRLHeatingSystemResetFcn(in);
```

```
validateEnvironment(env)

%% Train Agent
disp("Start Training...")
trainOpts = rlTrainingOptions(...
    MaxEpisodes = 150, ...
    PerEpisodeStep_max = PerEpisodeStep_max, ...
    ScoreAveragingWindowLength = 5,...
    Verbose = false, ...
    Plots = "training-progress",...
    StopTrainingCriteria = "AverageReward",...
    StopTrainingValue = 200); %StopTrainingValue = 85

trainingStats = train(agent,env,trainOpts);

%% Simulate DQN Agent
disp("Showing Graphs...")
max_step= 720;
temp_validation = temp_march21;
env.ResetFcn = @(in) hRLHeatingSystemValidateResetFcn(in);
simulation_options = rlSimulationOptions(max_step = max_step);
exp1 = sim(env,agent,simulation_options);

localPlotResults(exp1, max_step, ...
    max_comfort, min_comfort, sample_time,1)



% Validate agent using the data from April 15
temp_validation = temp_april15;
exp2 = sim(env,agent,simulation_options);
localPlotResults( ...
    exp2, ...
    max_step, ...
```

```
    max_comfort, ...
    min_comfort, ...
    sample_time,2)


% Validate agent using the data from April 15 - 8 degrees
temp_validation = temp_april15;
temp_validation(:,2) = temp_validation(:,2) - 8;
exp3 = sim(env,agent,simulation_options);
localPlotResults(exp3, ...
    max_step, ...
    max_comfort, ...
    min_comfort, ...
    sample_time, ...
    3)
```

.2 TD3 Code:

```
disp("Loading Model...")
mdl = "rlHC";
open_system(mdl)
sample_time = 120; % seconds
PerEpisodeStep_max = 1000;
agentBlk = mdl + "/Smart Controller/RL Agent";

disp("Logging Data for Temperature...")
data = load('temperatureMar21toApr15_2022.mat');
temperatureData = data.temperatureData;
%Validation Data
temp_march21 = temperatureData(1:60*24,:);
temp_april15 = temperatureData(end-60*24+1:end,:);
%Training Data
temperatureData = temperatureData(60*24+1:end-60*24,:);
```

```
%% Parameter Settings for Environment and House
disp("Loading House Parameters...")
outsideTemperature = temperatureData;
comfortMax = 18;
comfortMin = 25;
Mdot = 1496.0387502928907;
cair = 1005.4;
Tac = 24;
wallArea = 320;
wallDensity = 1920;
wallMass = 122880;
wallVolume = 64;
widHouse = 10;
widWindows = 1;
windowArea = 6;
windowDensity = 2700;
windowMass = 162;
windowVolume = 0.0600;
TinIC = 20;
pitRoof = 0.0707;
roofArea = 601.0840;
roofDensity = 32;
roofMass = 3.8469e+03;
roofVolume = 120.2168;
kRoof = 0.0380;
kWall = 0.0380;
kWindow = 0.7800;
LenHouse = 30;
LRoof = 0.2000;
LWall = LRoof;
LWindow = 0.0100;
M_air = Mdot;
```

```
c_air = cair;
c_roof = 835;
c_wall = 835;
c_window = 840;
h_A_R = 12;
h_A_W = 24;
h_A_Wnd = 25;
h_R_Atm = 38;
h_W_Atm = 34;
h_Wnd_Atm = 32;
htHouse = 4;
htWindows = 1;

%% Create Observation Info and Action Info (Output Info)
obsInfo = rlNumericSpec([6,1]); %Each Observation value can have infinite many values
actInfo = rlNumericSpec([1 1],'LowerLimit',[0],'UpperLimit',[1]); %Can only be 0 or 1
rng(0) %Make seed of random number generator 0

%% Create Paths
% Observation Neural Path
obsPath = [
    featureInputLayer(prod(obsInfo.Dimension),Name="obsPathIn")
    fullyConnectedLayer(32)
    reluLayer
    fullyConnectedLayer(16,Name="obsPathOut")
    ];

% Output Neural path
actPath = [
    featureInputLayer(prod(actInfo.Dimension),Name="actPathIn")
    fullyConnectedLayer(32)
    reluLayer
    fullyConnectedLayer(16,Name="actPathOut")
```

```
    ];

% Common Path both can take
commonPath = [
    concatenationLayer(1,2,Name="concat")
    reluLayer
    fullyConnectedLayer(1)
    ];
%% Create Critic
% Add path to Critic. TD3 has two critic for Q Values
criticNet = layerGraph;
criticNet = addLayers(criticNet,obsPath);
criticNet = addLayers(criticNet,actPath);
criticNet = addLayers(criticNet,commonPath);

%Connect individual Observation and Action path to each other
criticNet = connectLayers(criticNet,"obsPathOut","concat/in1");
criticNet = connectLayers(criticNet,"actPathOut","concat/in2");

%Create Deep Learning separate weights for the two separate critics
criticNet1 = dlnetwork(criticNet);
criticNet2 = dlnetwork(criticNet);
summary(criticNet1);
summary(criticNet2);

% Using Critic Net, create individual Q Learning Functions
critic1 = rlQValueFunction(criticNet1,obsInfo,actInfo);
critic2 = rlQValueFunction(criticNet2,obsInfo,actInfo);
getValue(critic1,{rand(obsInfo.Dimension)},{rand(actInfo.Dimension)});
getValue(critic2,{rand(obsInfo.Dimension)},{rand(actInfo.Dimension)});


%% Create Actor
```

```
% TD3 only has single actor
actor_network = [
    featureInputLayer(prod(obsInfo.Dimension))
    fullyConnectedLayer(400)
    reluLayer
    fullyConnectedLayer(300)
    reluLayer
    fullyConnectedLayer(prod(actInfo.Dimension))
    tanhLayer
];

% Single Deep Learning Actor
actor_network = dlnetwork(actor_network);
summary(actor_network);

% Use ActorNet to create TD3 actor
actor  = rlContinuousDeterministicActor(actorNet,obsInfo,actInfo);
getAction(actor,{rand(obsInfo.Dimension)});

%% Making TD3 Agent

%Options for Critic
critic_options = rlOptimizerOptions( ...
    LearnRate=0.001,...
    GradientThreshold=1);
%Options for Actor
actor_options = rlOptimizerOptions( ...
    LearnRate=1e-3,...
    GradientThreshold=1);
    %L2RegularizationFactor=1e-5 ...

%Options for the TD3 Agent
agent_options = rlTD3agent_options;
```

```
agent_options.CriticOptimizerOptions = critic_options;
agent_options.ActorOptimizerOptions = actor_options;
agent_options.SampleTime = sampleTime;

%Create Agent
agent = rlTD3Agent(actor,[critic1 critic2],agent_options);

%% Creating RL Simulink Environment and validating it. validateEnvironment
% is a MATLAB function

disp("Simulating Env...")
env = rlSimulinkEnv(mdl,agentBlk,obsInfo,actInfo);
env.ResetFcn = @(in) hRLHeatingSystemResetFcn(in);
validateEnvironment(env)

%% Train
disp("Start Training...")
trainOpts = rlTrainingOptions(...
    MaxEpisodes = 150, ...
    MaxStepsPerEpisode = 1000, ...
    ScoreAveragingWindowLength = 5,...
    Verbose = false, ...
    Plots = "training-progress",...
    StopTrainingCriteria = "AverageReward",...
    StopTrainingValue = 85);

trainingStats = train(agent,env,trainOpts);

%% Get Results
disp("Showing Graphs...")
max_step= 720;
temp_validation = temp_march21;
env.ResetFcn = @(in) hRLHeatingSystemValidateResetFcn(in);
```

```
simulation_options = rlSimulationOptions(max_step = max_step);
exp1 = sim(env,agent,simulation_options);

localPlotResults(exp1, max_step, ...
    max_comfort, min_comfort, sample_time,1)



% Validate agent using the data from April 15
temp_validation = temp_april15;
exp2 = sim(env,agent,simulation_options);
localPlotResults( ...
    exp2, ...
    max_step, ...
    max_comfort, ...
    min_comfort, ...
    sample_time,2)



% Validate agent using the data from April 15 - 8 degrees
temp_validation = temp_april15;
temp_validation(:,2) = temp_validation(:,2) - 8;
exp3 = sim(env,agent,simulation_options);
localPlotResults(exp3, ...
    max_step, ...
    max_comfort, ...
    min_comfort, ...
    sample_time, ...
    3)
```

.3 Plot Graph:

```
    function localPlotResults(exp, maxSteps, comfortMax, comfortMin, sampleTime, figNum)
    % localPlotResults plots results of validation
```

```
% Compute comfort temperature violation
Violated_Comfort_Minutes = ...
sum(exp.Observation.obs1.Data(1,:,1:maxSteps) > comfortMin) ...
+ sum(exp.Observation.obs1.Data(1,:,1:maxSteps) < comfortMax);

% Cost of energy
total_cost = exp.SimulationInfo(1).logsout{1}.Values;
total_cost.Time = total_cost.Time/60;
total_cost.TimeInfo.Units="minutes";
total_cost.Name = "Total Energy Cost";
final_cost = exp.SimulationInfo(1).logsout{1}.Values.Data(end);
final_cost = abs(final_cost);

% Cost of energy per step
perStepCost = exp.SimulationInfo(1).logsout{2}.Values;
perStepCost.Time = perStepCost.Time/60;
perStepCost.TimeInfo.Units="minutes";
perStepCost.Name = "Energy Cost per Step";
minutes = (0:maxSteps)*sampleTime/60;

% Plot results

fig = figure(figNum);
% Change the size of the figure
fig.Position = fig.Position + [0, 0, 0, 200];

% Temperatures
layoutResult = tiledlayout(3,1);
nexttile
plot(minutes, ...
    reshape(exp.Observation.obs1.Data(1,:,:), ...
    [1,length(exp.Observation.obs1.Data)]),"k")
```

```
    hold on
    plot(minutes, ...
        reshape(exp.Observation.obs1.Data(2,:,:), ...
        [1,length(exp.Observation.obs1.Data)]),"g")
    yline(comfortMin,'b')
    yline(comfortMax,'r')
    lgd = legend("T_{room}", "T_{outside}","T_{Minium Comfort}", ...
        "T_{Maximum Comfort}","location","northoutside");
    lgd.NumColumns = 4;
    title("Temperatures")
    ylabel("Temperature")
    xlabel("Time (minutes)")
    hold off

    % Total cost
    nexttile
    plot(total_cost)
    title("Total Cost")
    ylabel("Energy cost")

    % Cost per step
    nexttile
    plot(perStepCost)
    title("Per Step Cost")
    ylabel("Energy Cost")
    fprintf("Comfort Temperature violation:" + ...
        " %d/1440 minutes, cost: %f PKR\n", ...
        Violated_Comfort_Minutes, final_cost);
end
```

.4 Reward Function Code:

```
    function reward = OptRewardFcn(room_temp, perStepCost,...
```

```
    comfort_max, comfort_min, Action, prev_action)

% energy cost
energy_reward  = perStepCost;

% comfort level reward
if (room_temp <= comfort_min) && (room_temp >=comfort_max)
    comfort_reward = 0.1;
elseif room_temp > comfort_min
    comfort_reward = -0.1*abs(room_temp - comfort_min);
elseif  room_temp < comfort_max
    comfort_reward = -0.1*abs(room_temp - comfort_max);
else
    comfort_reward = 0;
end

% switch penalty
if prev_action ~= Action
    switching_penalty = -0.01;
else
    switching_penalty = 0;
end

reward = energy_reward + comfort_reward + switching_penalty;
```