COLLABORATIVE REINFORCEMENT LEARNING CONTROL OF PAIRED VERTICAL AXIS WIND TURBINES

A Thesis

Presented to

the Faculty of the Department of Mechanical Engineering

California State University, Los Angeles

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Mechanical Engineering

By

Vladimir Pena

May 2024

© 2024

Vladimir Pena

ALL RIGHTS RESERVED

The thesis of Vladimir Pena is approved.

Dr. Arturo Pacheco-Vega, Committee Chair

Dr. Mathias Brieu, Committee Member

Dr. Salvador Rojas, Committee Member

Dr. Mathias Brieu, Department Chair

California State University, Los Angeles

May 2024

ABSTRACT

Collaborative Reinforcement Learning Control of

Paired Vertical Axis Wind Turbines

By

Vladimir Pena

Capable of operating in omni-wind directions, vertical axis wind turbines are great candidates for ocean and urban applications, where wind directions vary frequently and dramatically. However, there are still many challenges when it comes to the design of control methods for turbines in urban settings, mainly due to the complex flow through the turbine, which makes it challenging to obtain an accurate dynamics model. This thesis proposes the use of a reinforcement learning algorithm to optimize a pitching trajectory for the VAWT to maximize efficiency. A twin collaborative VAWT setup is simulated under an urban-like environment for enhanced performance and faster learning. The application of a Parameter Exploring Policy Gradient (PGPE) is implemented to learn the optimal pitch control of a VAWT under varying wind conditions. The control system uses power measurement only to adjust the control policy. A comprehensive discussion of the parameter selection for policy approximation will be discussed. Three wind conditions are simulated to validate the effectiveness of the proposed reinforcement learning control method along with two policies tested. A comparison of the new wind turbine control to the traditional controls, operation, and energy efficiency will be also provided.

ACKNOWLEDGMENTS

Academic acknowledgments

I would like to thank **California State University, Los Angeles**, the **College of Engineering, Computer Science and Technology** and the **Mechanical Engineering Department** for providing me the opportunity to continue my education and obtain a Master's Degree. I would also like to thank the National Science Foundation and the CREST Center for Advancement toward Sustainable Urban Systems (under Award HRD-2112554), for the support in developing this work.

Personal acknowledgments

I want to express immense gratitude to Dr. Pacheco-Vega for his guidance, support, and time in finishing this work. I would also like to thank Dr. Brieu for his support in continuing this work. A special thanks to Dr. Rojas for helping me complete the last leg of my journey as a master's student. Thank you all for your support and I'm incredibly appreciative to have you all on my committee. Finally, I would like to especially thank Dr. He Shen for his unwavering support, his time invested in me, and his commitment to providing the best education and Master's experience during his tenure at Cal State LA. This work would not have been possible without him and I am deeply grateful for his help and mentorship.

To my family and friends, whom I am grateful for having throughout this journey, thank you for always supporting and encouraging my education and believing in me even when I had doubts. Without their support, this would not have been possible.

TABLE OF CONTENTS

ABSTRACT iv

ACKNOWLEDGMENTS v

1 Introduction 1

1.1 Motivation 3

1.1.1 Renewable Energy and its Importance 3

1.1.2 Overview of Turbine Wind Energy 6

1.1.3 Evolution of Vertical Axis Wind Turbines (VAWT) 9

1.1.4 Historical Developments of VAWT Control Systems 14

1.1.5 Importance of Control Systems in Wind Turbines 15

1.2 Literature Review 17

1.2.1 State-of-the-Art Control Strategies & Recent Advances in Innovations in VAWT 19

1.2.2 Challenges and Opportunities in VAWT Control 24

1.3 Objectives and Potential Contributions 25

1.4 Thesis Overview 26

2 Background 27

2.1 Mathematical Modeling of the VAWT 27

2.1.1 Blade Element Theory 27

2.1.2 Double Multiple Stream Tube Model (DMST) 31

2.1.3 System Description and Dynamic Modeling of VAWT 37

2.2 Control System of VAWT 41
2.2.1 Policy Gradient Policy Exploration (PGPE) for Pitch Optimization 41
2.2.2 PI Speed Control 46

3 Collaborative Dual Wind Turbine Reinforcement Learning 47
3.1 Policy Updates with CoRL - Symmetric Sampling 47
3.1.1 Clip Up 51
3.2 Pitching Policy 52
3.2.1 Sine Policy 53
3.2.2 Spline Policy 56
3.3 Twin VAWT Setup 57
3.3.1 MATLAB Pitch Optimization 58
3.3.2 Environment 61
3.4 Policy Updates for Single VAWT 64
3.4.1 PGPE for Single Turbine Operation 64
3.4.2 Environment 66

4 Numerical Results 68
4.1 Initialization of Dual Turbines 69
4.2 Optimal Pitching Trajectory using Online PGPE Algorithm 70
4.2.1 Constant Wind Condition 70
4.2.2 Variable Wind Condition using a Ramp Wind Function 80
4.2.3 Synthetic Random Wind 88
4.3 Dual Turbine Performance Metric 93

5 Conclusion and Future Works 103

5.1 Conclusion 103

5.2 Future Direction 104

REFERENCES 105

A Simulating Environment using DMST Model 113

B PGPE Symmetric Sampling with Clip Up 118

LIST OF TABLES

3.1 Parameter definition. 55

3.2 VAWT geometric parameters. 58

4.1 Hyperparameters for CoRL. 70

4.2 Hyperparameters for single operation WT. 74

4.3 Result summary of constant wind case. 81

4.4 Result summary of varying wind case. 86

4.5 Result summary of synthetic wind case. 91

LIST OF FIGURES

1.1 Vertical axis wind turbine models [19]. 9

1.2 Airfoil aerodynamics depiction after [21]. 11

2.1 Working principle of a lift-driven VAWT. 28

2.2 Aerodynamic forces acting on blades. 30

2.3 Double multiple stream tube model. 35

2.4 Control system block diagram. 37

2.5 Structure of CoRL pitch control. 40

3.1 Illustration of pitching trajectory formulation. 55

3.2 Spline policy formulation using PGPE with symmetric sampling. 57

3.3 VAWT performance curve for collective blade pitching. 59

3.4 Optimal pitching trajectories under various tip speed ratios. 60

3.5 VAWT performance for individual pitch control. 62

3.6 CoRL environment. 63

3.7 Control block diagram for single turbine. 66

4.1 CoRL sine policy - Constant wind. 71

4.2 Optimal sinusoidal pitch trajectory - Constant wind. 73

4.3 CoRL spline policy - Constant wind. 75

4.4 Optimal spline pitch trajectory- Constant wind. 76

4.5 Single WT sinusoidal policy - Constant wind. 77

4.6 Single WT sinusoidal policy - Updated initialization. 79

4.7 Optimal pitch trajectories comparison for constant wind. 80

4.8 Pitch evolution for constant wind environment. 81

4.9 Variable wind profile - Ramp function. 82

4.10 CoRL sinusoidal policy – Varying wind. 84

4.11 CoRL spline policy – Varying wind. 85

4.12 Single WT sinusoidal policy – Varying wind. 87

4.13 Synthetic random wind. 88

4.14 CoRL sine policy – Synthetic wind. 90

4.15 CoRL Spline policy – Synthetic wind. 92

4.16 Sinusoidal policy parameter convergence. 94

4.17 Dual turbine speed comparison. 97

4.18 Large difference between turbine speeds. 98

4.19 Reward comparison of turbines. 99

4.20 Periodic pitching trajectories. 100

4.21 Non-pitched turbine. 101

4.22 Angle of attack reduction during operation for $\lambda = 1.94$. 102

CHAPTER 1

Introduction

It is well known that climate change imposes a huge challenge on many sensitive bio-diverse regions and on the human population as well. With the continual use of fossil fuels, climate change only accelerates and therefore the current global push for alternative clean energy. Recently the demand for renewable energies has increased significantly. Since the 19th century industrial revolution wind energy has been abandoned in favor of fossil-fueled energy. Today, still roughly 60% of the electricity generated in the US comes from fossil fuels compared to 8.5%[1] from renewable energies reports the U.S. Energy Information Administration (EIA), [1]. Although the push for legislative policy changes across the world under the Paris Agreement has implied the need to alter the course of power generation through the use of clean renewable energy.

As of 2023, the EIA shows the trend of consumption of both solar and wind energy increasing rapidly overtaking the consumption of hydroelectric power. Wind turbines contributed roughly around 18.3% of the power consumed in the renewable energy sector in the year 2022, and as of August 2023 the leading clean sustainable energy source in the US, overcoming hydroelectric and solar in terms of energy consumed [2]. From the years 2020 to 2021 a total capacity of wind energy of 13.4 GW was installed bringing the total wind energy capacity to 135.9 GW, reports the EIA. On the other hand, solar energy got an increase of an additional 20.2 GW in the year 2022 bringing the total PV capacity to 140.6 GW. And to add to that, in the first half of 2023 the US installed 34% more PV

[1] As of September 2023 the EIA began reporting the captured renewable energy to its equivalence in fossil fuel and therefore may be reported as 13% in other papers. See reference for more info.

installations over the previous year [3]. In the USA the major leading renewable energy sources are solar, wind, hydroelectric, geothermal, and biomass, biomass is a topic of debate but nevertheless still a renewable source if harvested and sourced carefully, and in total account for 300 GW installed capacity of renewable energy. Yet still renewable energy accounts for 8.5% of the total energy consumed in 2022 in the US while demand for energy keeps growing worsening the problem.

In a report published by the International Renewable Energy Agency (IRENA), [4], the global energy capacity installed per renewable source are as follows, in GW, for the year 2022: hydroelectric at 1255, solar PV at 1055, onshore wind turbines 836, bio-energy at 151, offshore wind 63, geothermal 14.6, concentrated solar power at 6.6, and marine 0.5. Globally the leading country in renewable energy is China with a capacity installed of 510 GW and by the end of 2023, it is expected to have installed a record-breaking additional 200 GW of PV panels and ending the year with an expected total of 230 GW of added renewable energy capacity [3]. The Global Wind Energy Council (GWEC) reports, [5], the added new wind energy global capacity in 2021 was 93.6 GW beating the previous record by 1.8% from the previous year and bringing the total global wind energy capacity to 837 GW. But the GWEC claims we are far from the 1.5°C pathway and a net zero energy production by 2050 set out by the Paris Agreement and the IRENA in which case wind energy installments should quadruple to stay on track. Although renewable energy trends continue to grow and catch traction globally, there is much more work needed to reach sustainable energy production.

1.1 Motivation

1.1.1 Renewable Energy and its Importance

An article by the International Energy Agency (IEA) gives a good overview of the current global situation on access to energy which is important to summarize here [6]. Over 760 million people of the population live without access to electricity dominated by the sub-Saharan African desert. This region accounts for over 80% of the population without access to electricity. COVID-19 and the energy crisis have slowed the progress toward the continued expansion of access to electricity, and less than a fifth of African countries have a target date of 2030 to provide universal access to electricity for their population. In Asia, the population with access to electricity greatly increased. In 2010 the population with access to electricity was 79% and in the year 2022 increased to 97%. On a brighter note, under the Indian Saubhagya scheme India, Indonesia, and Bangladesh were able to extend access to electricity to more than 99% of their population even under COVID-19 and the energy crisis. But the IEA reports we are far off track from achieving universal access to electricity to the entire population claiming a need for 110 million new connections to be made which at the current pace is not achievable. Instead, a reliance on decentralized energy solutions for the time being can meet the goal of 2030 where 90% of the electricity generation would be made available by renewable energy and connections made via small energy grids and stand-alone systems. This solution is the least costly, but an increase in consumer end products will limit the implementation of these systems [6].

Bio-diverse regions and societies alike face unprecedented rapid changes due to climate

change. According to the United States Environmental Protection Agency (EPA), the retention of greenhouse gases not only affects climate change but also has direct adverse effects on human health and society, large ecosystems, oceans, and Arctic regions covered in snow and ice. On human health the warmer climate leads to health concerns for the vulnerable population such as the elderly, poor health, the disabled, and the indigenous. An increase in average temperature means longer heat waves that can cause heat-related illness, an increase in mosquitoes that can carry potentially deadly diseases, and an increase in Lyme disease that is influenced partially by climate among other things. Society must adapt to a rapidly changing environment that experiences extreme weather patterns such as new record temperature highs, flooding, eroding shore fronts due to rapid ocean rise, extreme tropical storms that cause vast devastation on property, and agriculture changes. Wildfire season is peaking much earlier and the top 10 worst wildfires to have occurred all happened since the year 2004 and not only affect humans but have an equal if not worse effect on wildlife displacing many native wildlife and damaging entire ecosystems reports EPA. In water bodies streams and lakes are rising in temperature affecting the water evaporation rate and marine life has also shifted along the US coast in response to the rise in temperature. Finally, EPA reports Arctic regions are rapidly losing their once abundance of glaciers and snow caps that have an impact on wildlife, the rise of water levels, and the ability to reflect radiation. More information on the effects of climate change can be found in reference [7].

Developed nations have a moral obligation to keep the environment clean for everyone. For many years developed nations have enjoyed the benefits of having access to electricity accelerating their economy at a rapid pace increasing their infrastructure

and hence requiring more and more energy. This hunger for an increase in energy led many countries to exploit the natural resources within their countries and in many cases expand to weaker underdeveloped nations leaving them with the burden of an eroding landscape. While no one country is immune from the effects of greenhouse gases and the pollution caused by the production of electricity, underdeveloped nations experience the consequences without the benefits of access to electricity leaving a vulnerable population to the extremes of climate change. Therefore the moral obligation to produce energy in a sustainable clean form and most importantly aid these underdeveloped countries in ensuring their access to universal access to electricity.

Finally, an impactful aspect of renewable energy is ensuring a sustainable world for generations. It is well known the globe climate is dynamic, changing over the course of many years, and in the process, so do its inhabitants, including humans. From the ice age to the natural global warming of our planet that exposed the modern landscape we have now. Yet what many fail to recognize is the time frame in which these events occur. For example, over the span of 100,000 years, the planet slowly increased in temperature from 5° C to 8° C (9° to 14.4° F) [8]. Currently, the rate at which the global temperature is increasing is occurring at a much faster rate with the highest increase beginning since 1975, increasing 0.10° to 0.15° C per decade and continuing since [9]. Since 1880, the global average temperature has increased by 1.1° C. In modern times the production of electricity contributes to 3/4 of greenhouse gases globally. The greenhouse gases are carbon dioxide, nitrite oxide, methane, and fluorinated gases and compose up to 40%, 20%, 2.5 times, 100% more of the atmosphere chemical composition than pre-industrial times [10]. Fluorinated gas is 100% man-made. Limiting the GHG and reducing the

global temperature by becoming net zero energy producers will limit and preserve the globe we currently inhabit for many generations. Therefore the need to continue to invest in renewable energy technologies and hence the reason for this paper.

1.1.2 Overview of Turbine Wind Energy

Wind energy has been around for many centuries and used for a wide variety of tasks ranging from sailing to food production. By the 11th century, wind turbines were used extensively by the Middle East and were eventually introduced to the Europeans by merchants and crusaders which eventually made its way to the Western world [11]. Wind energy has proven to be an invaluable form of energy for many such as in early pre-industrial Europe, rural areas, and early settlers in the Western world. The Dutch windmill is an image familiar to most when one thinks of a windmill, which is common for a picturesque backdrop of the Netherlands and is widely used as a water pump and food processing plant. A predecessor to the modern utility-scale wind turbines currently used today was introduced by two engineers from Austria and Scotland in the 1880s along with other prominent engineers from that time era. Josef Friedländer from Austria and Prof James Blyth from Scotland are said to be the first to install wind turbines for electricity generation [12]. These systems were used to charge batteries to then power single factories and a small number of other loads. The electrification of rural areas powered by oil and the expensive costs of battery systems paused the continuing development of wind turbines.

Modern-day utility turbines were developed following the 1970 oil crisis, but the first utility-scale turbine in the MW scale dates back to 1941, with the installation of a

HAWT developed by Palmer Cosslett Putnam and Clinton S. Smith, which was installed in Vermont, USA [13]. The turbine had a capacity of 1.25 MW with a rotor diameter of 175 feet and was used to power the local utility grid. Following the 1970 oil crisis US Department of Energy invested heavily to modernize wind technology for both the VAWT and the HAWT by employing NASA, Sandia National Laboratories (SNL), and many other contractors. One of the main objectives of these studies was to transfer research and technology of VAWT development from the government to the commercial sector [14], [15]. SNL and NASA, alongside other companies, prototyped and employed many turbines to study and improve the aerodynamics, structural tower design, electrical motors, blades, modeling and control strategies, and the scalability of turbines. Although none of these turbines would become commercially available, the goal of pushing turbine technology forward was successful as many of the companies involved in the study were, and to this day, wind turbine manufacturers. Danish development of wind turbines was also fundamental given they had a vast majority of decentralized HAWT turbines providing 5 to 25 kW of power. In 1978 the Danish deployed the first multi MW capacity wind turbine in the world with modern technology such as pitch control and modern blade design [16]. Nowadays the push for wind turbine technologies is driven by the preservation of the environment more so than that of the scarcity of oil [11].

To this day there are many types of wind turbines and many more intricate novel designs being developed that fall under a class of their own. Arguably the most prominent and widely used are the Horizontal Axis Wind Turbine followed by the Vertical Axis Wind Turbine. New turbine designs include bladeless turbines that turn kinetic wind energy into vibrations designed by the company Vortex Bladeless, turbine designs inspired by the

Archimedean spiral such as the Danish Archimedes windmill (AWM), turbines that are inspired by hummingbirds considered for residential areas by Tyer Wind, and turbines that have cylindrical tubes for blades for harsh typhoon environments implemented by a Japanese startup Challenergy. In addition, novel turbine implementations such as floating turbines that are taken up by scientific balloons, hybrid turbines that use two types of turbines in one, and wind turbine implementation on infrastructure. All wind turbines have pros and cons and have specific environmental implementation and therefore there is not one that fits all applications. Such differentiating factors can be decentralized vs centralized grid solutions, urban vs open field, onshore vs offshore, limitation on land usage, moderate wind to extreme wind, aesthetics, and noise pollution among other factors. These are some of the factors considered when choosing the correct wind turbine for any application.

Economically speaking wind turbines have an Energy Return of Investment (ERI) of 18:1 but can vary depending on the size and type of the turbine [17]. Costs associated with turbines are capital costs, operation and maintenance, capacity factors, lifetime and depreciation, grid connectivity, and government programs such as incentives and subsidies, which also impact the economics of wind turbines. With the increase of technology and the general availability of wind energy, the price of electricity is beginning to become increasingly affordable and is predicted to be a quarter of the price of fossil fuels in the very near future as many indicators over the past decades show renewable energies price fall below fossil fuels [18].



Figure 1.1: Vertical axis wind turbine models [19].

1.1.3 Evolution of Vertical Axis Wind Turbines (VAWT)

The Vertical Axis Wind Turbine has been around for many years but the most notable contribution in recent history is the development by French engineer Georges Darrieus in 1925. The original Darrieus wind turbine has two blade designs, the Giromill and the troposkein design, or the "egg-beater" design, which rotates about the axis perpendicular to the incoming stream of air or a vertical axis as the name implies. This turbine is a lift-based device that utilizes the aerodynamic forces created using airfoils such as those found in airplanes as opposed to drag-based devices that rely solely on drag forces to spin. Shown in Figure 1.1 are the many models developed over the years, from left to right the Darrieus, H-blade, Helical, a variant of the Savonius drag type turbine, and the Hybrid VAWT that incorporated both Savonius and Darrieus turbines. The focus of this writing is on lift-type devices that utilize the aerodynamic forces created using airfoils. In summary, an airfoil experiences aerodynamic forces by the creation of a pressure difference acting on the airfoil surfaces which when integrated gives the aerodynamic

forces. The summation of these forces gives the resultant force which is taken to act at the aerodynamic center of the airfoil at $\frac{1}{4}$ length of the chord. The resultant force is broken down into two components, lift and drag. These two forces are a function of the airfoil's angle of attack which is defined as the angle between the chord of the airfoil and the relative wind velocity, refer to Figure 1.2. Unfortunately, the relation between the aerodynamic forces and the angle of attack gets much more complicated as the airfoil reaches higher angles of attack, or more specifically once the airfoil reaches and passes its stall angle [20]. Once the airfoil reaches its stall angle drag is the dominant force caused by the boundary layer separation. In addition, the forces are no longer steady but rather unsteady, and transient analysis is required among other challenges. Consequently, the VAWT suffers from a phenomenon known as dynamic stall due to the high angles of attack during normal operation, which greatly reduces the performance of the VAWT in power generation. Not to mention the turbine's inability to self-start and lower efficiency when compared to the HAWT. On the other hand, the turbine's ability to receive wind from any direction, omnidirectional, ease of manufacturing and installation, and load reduction on the tower due to the location of the generator at the base have all made the Darrieus VAWT a formidable candidate for urban-based applications and more recently in offshore applications [15]. This renewed interest has brought a wave of research in recent years and the development of a newer advanced VAWT.

Beginning in the 1970 the Sandia National Laboratories began research on the Darrieus VAWT in an effort to bring the VAWT up to date with modern wind technology and share gathered knowledge with the commercial industry. These studies resulted in a better understanding of the VAWT and all its challenges along with the development

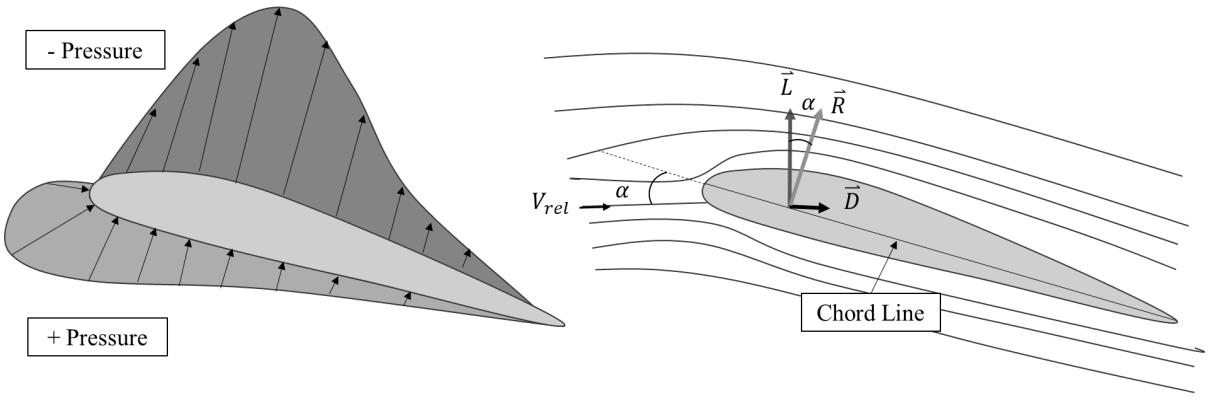


Figure 1.2: Airfoil aerodynamics depiction after [21].

of 4 prototype turbines. The biggest challenge was the structural integrity and the safe operating speeds of the turbines as 1 of the four turbines collapsed due to buckling and another failed due to braking malfunctions. With the gained knowledge from their last culminating prototype turbine, the 34 m 'Test Bed' VAWT turbine, the Sandia National Laboratories partnered with FloWind Corporation in attempt to commercialize the VAWT using the 17 m VAWT platform developed by SNL due to its high success. Although the VAWT showed great success the project was never realized after FloWind Corporation went bankrupt in 1997. After this, the VAWT lost all funding and interest from the US government until 2012. Most importantly from these studies, the SNL found that VAWT at 500kW and higher capacity ratings are direct competitors to the HAWT at the time of this research [15].

Aside from the large-scale VAWT implementation, there has been a great push for small-scale VAWT implementation for urban use. Many researchers have studied the influence of geometric design parameters on wind turbine performance. Some of these parameters are the aspect ratio of the turbine, the solidity of the turbine, the number of blades, blade supports, the turbine tower, and blade design. Ultimately these parameters

affect the aerodynamics of the turbine and vary greatly depending on the operating environment the turbine is to be installed. In many studies, solidity is considered to be one of the most important design parameters [22], [23], [24]. The solidity of a turbine is a key parameter in determining the optimal operating speed of the turbine relative to the wind speed or the tip speed ratio, TSR, and is defined as the ratio of the blade area to the frontal swept area of the turbine. This is one parameter that defines the optimal operating speed of the turbine. In an urban setting [22] reports a high solidity turbine with a variable speed setup is optimal. In general, the higher the TSR the lower the pitching amplitude and vice-versa. Research cited has investigated the effects of solidity by increasing the number of blades and using larger chord length blades [22], [23]. Increasing the number of blades also helps with the reduction of torque peaks that cause early failure on the bearings and gear train. Blade design and blade profile design have also been researched extensively. Most notably in the SNL research campaign blade profiles were designed specifically for the VAWT. In addition to blade profiles, helical blades designed to combat cyclic stress the VAWT experiences have been implemented. The helical blade design ensured the blades were always in contact with the incoming wind ensuring a smooth torque transition as opposed to the two peak torque per revolution, or how many number of blades the turbine has, experienced by traditional straight blades [25]. The aspect ratio of a turbine is another key parameter and is defined as the height of the turbine to its radius. The aspect ratio is found to influence the Reynolds number where increasing the aspect ratio decreases the Reynolds number and vice versa. [26] found that increasing the Reynolds number, by having a low aspect ratio, increases the power coefficient of the VAWT by increasing the lift coefficient of the blade and reducing

the drag coefficient. Both the solidity of a turbine and the aspect ratio determine the optimal operational speed of a turbine. Generally, the lower the aspect ratio and the higher the solidity of the turbine, the slower the rotational speed. These are just some of the important parameters that have influenced the VAWT design for urban applications over the years, but many more are currently being investigated.

For smaller turbines, auxiliary systems such as pitching mechanisms have been a hot topic lately. Pitching mechanisms allow turbines to operate in a vast range of wind speed conditions, allow VAWT to self-start, and position blades to reduce or eliminate the production of negative torque [27]. Pitching mechanisms have been implemented in the form of a 4-bar linkage system, the implementation of DC motors for the control of single and multiple-blade pitching, and cam and lobe systems. Although adding these systems generally requires complex assemblies and control, the benefits can outweigh the additional complexity to the design, and [28] finds the control system operation only consumes about 1% of total gross power produced.

While there are many modifications of the VAWT, they are out of the scope of this thesis such as hybrid VAWT that incorporates drag-based turbines with lift-type turbines. As such this is a very brief overview of the development and evolution of the VAWT as there is much more ongoing research to develop the VAWT even further. The previous information was given in the hopes of giving the reader a broad overview of the evolution of the VAWT and where it is today. The focus is mainly on straight-blade VAWT instead of the full Darrieus VAWT.

1.1.4 Historical Developments of VAWT Control Systems

Historical development of the VAWT is narrow, especially for the H- blade VAWT. The early studies of the VAWT done by SNL were mostly focused on those of the Darrieus Phi turbine with curved blades. These turbines had controls for starting/stopping, grid integration, and turbine protection as well as power optimization using variable speed operation. The development of the variable speed control algorithm from the SNL test campaign was a major advancement in the controls of the VAWT and one of the first to implement a true variable speed operation [15]. This variable speed algorithm allowed the turbine to follow its maximum efficiency curve at lower wind speeds, avoid structural damages through speed control, and turbine braking through regenerative braking.

Early control development for the VAWT was a difficult task as there was little known knowledge of the aerodynamics of the turbine. Designing for the optimal operating point of the VAWT encompassed many design considerations. For example, the structural integrity that in turn could affect the aerodynamics of the turbine and on top of that had to be manufacturable and therefore the controls were relied upon to avoid unsafe operational speed. But at the time, and to this day, there is no validated aerodynamic model for the VAWT that can capture the true performance of the turbine as predicted values were often affected by the smoothness of the blade where dirty blades in one occasion led to higher coefficient of performance, SNL 'Test Bed' VAWT, and lowered the performance for another, 19m FloWind turbine. Validated aeroelastic models are currently being researched by SNL for offshore implementation and commercial certification.

Early development and deployment of pitch control was used by the McDonell Aircraft

Company which used both electric DC motors and push rod cam follower mechanisms in their VAWT [15], [29]. The pitch control varied the pitch for one half of the turbine to a constant negative pitch angle and a constant positive pitch for the other half of the turbine. With pitch control, the company claims it can reach a higher maximum coefficient of power than that of the Darrieus VAWT [29]. Through initial research and investigation and early aerodynamic models, the turbine was expected to make a coefficient of power of 0.401 after losses [30]. In the first wind tunnel tests of the Giromill, the power consumption of the pitch control system was only 3% of the total power produced by the turbine, which was deemed cost-effective. The prototype was built in 1980 and successfully deployed, but due to its complexity and higher annual energy cost, the program did not continue [25]. Although the program did not continue, the interest in pitch control mechanisms on smaller VAWT still calls the attention of many researchers and is currently a hot topic for researchers worldwide.

1.1.5 Importance of Control Systems in Wind Turbines

The advancements in technologies translate directly to efficiency gains in the performance of wind turbines through improvements in the control system. Control systems are widely used in modern wind turbine systems, for single turbines but especially for the collection of wind turbines called a wind turbine farm. Technology plays a crucial role in monitoring the performance of individual turbines whether on a farm or stand-alone by using a collection of monitoring equipment that make up the workings of the controls. Advancements in technology such as in the form of physical equipment, sensors, and monitoring equipment, or in nonphysical forms such as advancements in algorithms that

compromise the brains of the control system increase the overall performance of existing turbines. To illustrate the importance of control systems we can look into well-known industries such as automotive. Many automotive manufacturers have been using the same power plant, in this case, an internal combustion engine, for many years with minor mechanical modifications. These cars improve their efficiency in terms of power output and gas consumption. What changed here is the development of new technology in both physical and nonphysical forms. Because of the development of sophisticated sensors, the algorithm has more in-depth knowledge of the workings of the internal combustion engine leading to better informed decision making. Likewise, with the advancements of control algorithms, the control system can take advantage of the additional information provided by the sensors and output the best outcome-provided inputs. The same principle applies to the wind turbine systems in which case the goal of the control system is to provide the best outcome, maximizing power output, given inputs of its environment. Examples of critical control functions in a wind turbine system are

- Turbine speed regulation - different wind turbines have different optimal rotational speeds that are influenced by environmental factors such as wind.
- Power optimization - a high-level goal of the control system for obvious reasons
- Blade angle - depending on the operation can be used to optimize for power, start-up process and shutdown process, blade load control, and more.
- Yaw - position the turbine nacelle in the direction of incoming wind.
- Grid power integration - safely load power production to existing grids.

- Load control - used for optimizing power, turbine speed control and load shedding.
- Start-up and shut down processes - startup and shutdown process uses a conjunction of turbine mechanisms to start or slow turbine.
- Fault detection - monitors the turbine for mechanical issues and operational safety by preventing overloading the structure mechanically and electrically.
- Energy storage - turbine control system with battery storage system integration for decentralized systems.
- Adaptation control - modern algorithms that continuously monitor turbine performance and degradation in addition to external factors to adjust parameters that help the turbine retain or even improve turbine performance.

The control system of a wind turbine is a crucial part of the turbine which brings it to the modern day era. It ensures the turbine system operates safely while maximizing the power output of a particular turbine or a collection of turbines.

1.2 Literature Review

The focus of this review is on the development of pitch optimization algorithm and their applications through controls. In addition to speed control development and use in wind turbines. Due to the VAWT's inability to self-start, highly fluctuating angle of attack (α) during operation, dynamic stall at low TSR caused by high angles of attack, and fatigue loading due to high torque variation around the azimuthal position of the turbine, the VAWT is regarded as being inefficient when compared to the HAWT. Although inefficient when compared to the HAWT the VAWT has several key design advantages that can

be leveraged to operate in areas deemed inoperable by the HAWT due to highly erratic winds such as in urban areas and can be further enhanced by both pitch control and load control.

Pitch control systems are a crucial component of wind turbines, as they determine the angle of the turbine blades relative to the blade tangential path in a VAWT and maintain steady angles of attack. Pitch control allows wind turbines to optimize energy capture and, in many instances, improve the life expectancy of the turbine by moderating torque loading on the blades and structure overall. More importantly the pitching control mechanism is used to ultimately implement the optimized pitch trajectory. The main goal of the optimized pitch trajectory is to 1) help moderate the angle of attack, 2) help the turbine self-start, and 3) help limit speed by load shedding and thus help avoid structural damage. The predefined or defined pitch trajectory is one calculated off line or developed in real time by the optimization algorithm and thus the job of the controller to implement said trajectory. The development of the pitch trajectory involves the input of multiple variables such as the turbine speed, wind speed, local angle of attack of each blade, velocity inflow, and turbine geometric parameters, and its main goal to reduce the high angles of attack experienced by the nominal operation of the VAWT, first function. These inputs are then fed into a complex formulation that predicts the optimal trajectory based on the current environmental state. The second function allows the turbine to start by positioning the blades in the optimal position to allow for maximum tangential forces and therefore self-start. The third function of pitch control is to use pitching to increase the drag instead of lift forces to limit the overspeeding of the turbine. The second and third points will not be discussed further in this thesis as they are out of the scope.

Recently, there has also been an expanded use case for the pitching mechanism to improve not only the individual turbine performance but the performance of a collective wind turbine farm or in our case a set of turbines. While the fundamental concept of pitch control remains the same, ongoing developments and innovations in this area will be briefly mentioned. The next section will review some of the pitching optimization and control implementations being studied. With an emphasis on Individual Blade Pitch Control and Predictive Pitch Control & Machine Learning and AI.

1.2.1 State-of-the-Art Control Strategies & Recent Advances in Innovations in VAWT

Pitch Optimization

There have been numerous investigations regarding the optimization of the VAWT blade pitch angle for individual blade pitching. Much research has gone into the implementation of a sine-cosine pitch trajectory due to the cyclic nature of the turbine. Given the TSR of the turbine is directly linked to the angles of attack experienced during operation, pitch is a function of TSR. Zhang et al [31],[32], and [33] used a sine function with a scaling factor to increase or decrease the amplitude of pitching trajectory based on turbine speed. Further [31] enhanced the scaling by separating the scaling factor in two for the windward and leeward, upwind and downwind in our definition, sides of turbines, and by applying a fitted curve over the original pitch trajectory to smoothen the blade pitching velocity to reduce torque fluctuation. In the lower tip speed ratio, the author of [31] reported a performance increase of 146%. Li et al [34] developed a genetic optimization algorithm with minimal optimization parameters to sufficiently cover the optimal solution space for

a pitch trajectory formulation for a wide range of TSRs. The algorithm integrated the use of CFD to evaluate 805 random curves generated by the algorithm. After optimizing the turbine, the author reported an impressive 64% efficiency, although it came at a simulation time of 145 hours. Similarly, Ma et al [35] and Paraschivoiu [36] used a genetic algorithm to optimize a cosine-like pitch trajectory using a small number of parameters, 2 and 3 respectively, to find the optimal pitch trajectory across a variety of TSRs. To go through the evolution of pitch trajectory optimization the authors used the single disk multiple stream tube method to validate the fitness of the curves. The authors reported an efficiency increase of 8 to 20% under the variety of different TSRs by [35] and a 30% increase by [36]. Jain et al [27] conducted a parametric study on a small-scale VAWT with a 4 bar drag linkage to implement variable amplitude blade pitching. Their study showed a sinusoidal pitch trajectory with a high amplitude of 35 degrees is needed for TSRs lower than 0.5 and less than 8 degrees after a TSR of 2.5. They concluded that with a pitching mechanism and high solidity turbine, they can extract more energy and increase efficiency from 10% to 36%, depending on the TSR. In addition to the sinusoidal pitching curve, [37] has proposed an adjustment law for blade pitching. Their goal was to maximize the turbine's coefficient of torque and therefore assess the optimal blade angle to always give the highest value. They used a high-fidelity model to improve the flow field around the blades and within the turbine to reflect an accurate local angle of attack. This ensured a better adjustment law given their pitching was highly dependent on the flow field around the blade. Given the highly non-linearity of the optimal trajectory, a curve was fitted over the original trajectory and improved the turbine's output by 14.56%.

Predictive Pitch Control & Machine Learning and Artificial Intelligence:

Predictive algorithms and models use weather forecasts and historical wind data to predict wind conditions in the near future. This information is then used to adjust blade pitch angles to optimize energy production proactively. Artificial intelligence and machine learning algorithms are also being employed in pitch control systems to continuously optimize blade angles based on historical data and real-time conditions as well. Abdalrahman et al [28] used a Multi-Layer Perception Artificial Neural Network (MLP-ANN) algorithm to implement an intelligent pitch controller. The MLP-ANN was used to learn the internal aerodynamics of the VAWT and then used to estimate the power output of the turbine based on three inputs: the turbine TSR, blade angle, and blade pitch. In addition, the MLP-ANN was used as the pitch controller as well and when compared to traditional controls, such as a PID, was much faster in response time. The efficiency of this method improved the turbine performance by as much as 25% in some instances. Shen and Ruiz [38] developed an online fast reinforcement learning algorithm to control a pitching mechanism based on a 4-bar drag bar linkage. The learning algorithm used was a PGPE algorithm to optimize the length of the drag bar link based on the TSR. The pitching trajectory formulated was based on the kinematics of the 4-bar linkage and its correlation to the dynamics of the turbine. Therefore, in controlling the 4-bar mechanism it directly influenced the performance of the turbine. The algorithm was successful in obtaining the optimal length of the drag bar linkage in just under a minute in both constant and variable wind environments. In addition, the algorithm was able to self-start the VAWT. The algorithm developed here increased the turbine performance by 30.5%

efficiency when compared to the turbine without any pitching mechanism, or zero pitch. [39] used a Bayesian reinforcement algorithm with Markov Chain Monte Carlo approach to predict the wind change and learn the VAWT dynamics to keep the turbine speed at an optimal operating speed and therefore maximize power. The inputs to the control, in this case a Radial Basis Neural Network, are the wind speed and acceleration, generator speed and acceleration, and load voltage with load current. Based on the current policy generated, the controller varies the speed by controlling the current load, reducing or increasing the speed. By using the Bayesian RL MCMC controller the efficiency of the controller was 89% as opposed to the traditional Maximum Power Point Tracker of 78% vastly used in the renewable sector and proven controller.

Dual turbine Operation:

With multiple turbines, coordinated pitch control systems can be employed to reduce wake effects. Adjusting the pitch of specific turbines to minimize wake interference can increase overall energy production across the wind farm. [40] used a dynamic pitch control on a combination of both the upwind and downwind turbines based on the turbine array. The upwind turbine with dynamic pitch control had a positive influence on the downwind turbine when their relative angle was less than 30 degrees. In contrast, the downwind turbine with dynamic pitch control benefited from the upwind accelerated turbine wake when their relative angle was greater than 30 degrees. This improved overall efficiency for both turbines by up to 20% when compared with a single turbine. [40] concluded that the dynamic pitching mechanism along with turbine array placement should be done in such a way as to take advantage of the turbine wake and dynamic pitching. [41] applied

machine learning techniques along with optimization techniques to solve for a staggered VAWT configuration dynamic. [41] used the ANN and Kriging methods to estimate the performance of two closely staggered twin-VAWTS. Both ANN and Kriging were trained off of 22.45% of high-fidelity CFD simulations taking the turbines' pitch angle as input and average torque produced as output. They concluded that both the Kriging and ANN had an output R^2 of 99% and 98% respectively when determining the torque based on pitch input. Overall, this method saved design time of a little over 75%. [42] also conducted a study on a twin VAWT setup to determine the effect of pitching on the performance of both the upwind turbine and the downwind turbine. The pitching range of both turbines ranged from 0 to -6 degrees. Chen concluded the downwind turbine benefited from the upwind turbine pitching by an increase of performance by 7.1% whereas the upwind turbine under the influence of the downwind turbine pitching gained a performance increase of 4.7%. [43] Hassanpour et al conducted a study on the effects of 3 design parameters for the formation of twin VAWT with a 3D analysis approach. The author used the Taguchi method to investigate three parameters: the horizontal separation of the turbines h, the angle relative to each other β, and the mid-vertical distance relative to each other S. The author concluded that the most impactful variables according to the Taguchi method are the height difference between the turbines, the angle relative to each other and finally the horizontal distance, $h > \beta > S$ The optimal values the author concluded with is a height difference of $h = 0$, angle difference of $\beta = 90$, and lastly a horizontal distance of 1.5D or 1.5 turbine diameters. The power output increased by 26.60% compared to that of a single turbine. In this study, there was no pitch control. On the other hand, [44] conducted a 2D analysis with 5 parameters

using the same optimization approach. In this analysis, the impact of each parameter is $\lambda > \beta >$ RD $>$ S/d $> \phi$. Where λ is the TSR, β inflow angle, RD rotational direction, S/d horizontal distance, and lastly ϕ blade pitch.

1.2.2 Challenges and Opportunities in VAWT Control

For many years, the Vertical Axis Wind Turbine (VAWT) has been overshadowed by the more efficient Horizontal Axis Wind Turbines (HAWT). As a result, the emphasis on implementing modern control techniques has been directed towards Horizontal Axis Wind Turbines. A major issue with the VAWT, apart from its lower efficiency, is the inability to self-start requiring external power in addition to any auxiliary mechanisms for the external motor or blade pitching mechanisms. The starting phase of the VAWT can be a complex task and even more so the comprehension of the aerodynamics and is easy to see why many would opt to go with the HAWT. Although recently the VAWT's self-starting issue has been addressed by many authors the advantage the VAWT has over the HAWT, being omni-directional, puts it in a stochastic wind environment making it difficult to design for. What continues to limit the advancement of the VAWT is the accurate modeling of the unsteady aerodynamics that is highly non-linear and consequently limits the performance of classical controls. Today, machine learning techniques have become an important tool for both solving complex control tasks and for real-time control updates with little to no insight of the system dynamics. This makes machine learning the perfect candidate to help solve and alleviate the modeling of the VAWT unsteady aerodynamic environment.

1.3 Objectives and Potential Contributions

Although RL has experienced success in various applications, RL still requires a large set of data for training. Pitch control policies are normally trained using simulated data from computational fluid dynamics (CFD) simulations. While the CFD data provides insights into the aerodynamic characteristics of a turbine, it cannot guarantee the completeness of the data in representing the environment. The training process takes a long time as simulating the turbine dynamics can take hours or days iterating through solutions such that most of the time is dedicated to solving the turbine dynamics rather than learning. To solve this problem, previous work has designed a fast online RL control of VAWTs [38]. To ensure the speed of convergence, only one parameter was used to optimize a reconfigurable periodic motion, which could then be used for online learning. To build upon the previous work, we propose a collaborative RL pitching control method for VAWTs that work in parallel. Herein, we loosen the constraint on the formulation for a pitching trajectory, such that it can better approach the optimal pitching trajectory and introduce newer reinforcement learning techniques from [45] to bring the algorithm up to date. Rather than focusing on solving the turbine dynamics, we propose the use of an online model-free RL algorithm. To maintain fast convergence, a collaborative learning strategy is proposed, such that the involved turbines can explore effectively to achieve high energy conversion efficiency. The main contributions of this thesis are as follows:

(1) A collaborative online RL control of a pair of VAWTs is proposed, where the sampling efficiency is significantly improved through symmetric sampling which can be extended to wind farms.

(2) A smooth periodic function is designed to formulate a flexible pitching trajectory, which uses as few as six parameters to strike a balance between flexibility and accuracy.

(3) A simulation software kit is developed for studying collaborative pitching control of paired VAWTs, and it can be used by engineers to access their turbine control strategies.

1.4 Thesis Overview

The thesis is organized as follows. In Chapter 1 we discussed the importance of renewable energy and the current state of Vertical Axis Wind Turbines. In Chapter 2, the aerodynamics of a VAWT are formulated and the pitch angle for control is introduced. The fundamentals of the aerodynamics model, the Double Multiple Stream Tube model, are presented, and an introduction to the optimization algorithm Policy Gradient Parameter Exploration (PGPE) is provided. Chapter 3 presents the setup of the Dual Collaborative Wind Turbine model. An overview of the pseudo-code will be presented to demonstrate the implementation of newer reinforcement learning techniques and the use of PGPE symmetric sampling to extend the optimization algorithm to a second turbine. An overview of the problem and the expected outcomes is presented in detail. In Chapter 4, the numerical results are presented and discussed. Finally in Chapter 5 will end with the conclusion.

CHAPTER 2

Background

Based on gained knowledge from the literature review, the VAWT wind turbine dynamics are formulated in this chapter. In addition, the introduction to Policy Gradients and the implementation of the PGPE structure from [46] for the VAWT introduced and similar control structure found in literature.

2.1 Mathematical Modeling of the VAWT

2.1.1 Blade Element Theory

Blade element (BE) model is a useful method to determine the aerodynamic forces of the blade given geometric parameters. This method alone does not suffice as an additional model is required to determine the inflow of air but once obtained the aerodynamic forces can be calculated using the BE method. In this study we will be pairing the BE model with the momentum model and solve for the inflow iteratively. The working principle of a lift-driven VAWT is illustrated in Figure 2.1. Here, the resultant wind over a turbine blade is a combination of the incoming wind and the rotation-induced and therefore calculated as

$$V_R = \sqrt{V_\infty(1-a)^2 + \omega R^2} \tag{2.1}$$

where V_∞ is the undisturbed wind speed, ω is the turbine rotational velocity, R is the turbine radius, and a the induction factor. As the resultant wind passes over an airfoil, it generates lift, $\vec{L}$, and drag, $\vec{D}$, forces. These two forces are functions of the angle of attack α of the corresponding airfoil, which is defined as the angle between the chord line of the airfoil and the resultant wind velocity. Without pitch control, the angle of attack

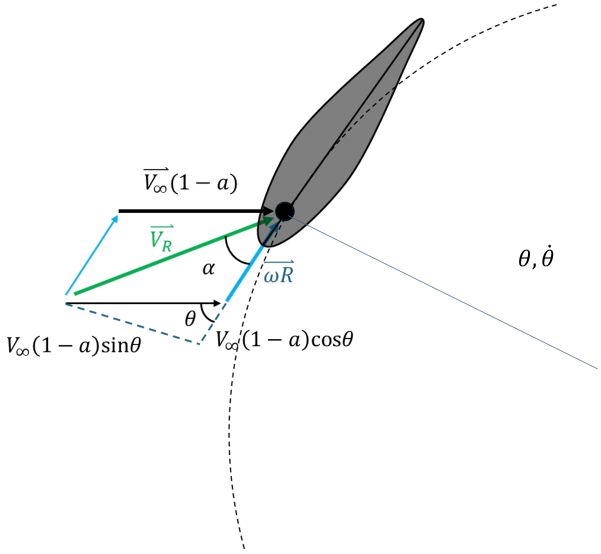


Figure 2.1: Working principle of a lift-driven VAWT.

of a VAWT can be calculated by

$$\alpha_0(\theta, \lambda) = \tan^{-1}\left(\frac{(1-a)\sin\theta}{(1-a)\cos\theta + \lambda}\right) \tag{2.2}$$

where θ is the azimuth angle of the blade, λ is the TSR, and a is the axial induction factor. The axial induction factor, a, gives an estimation factor for the incoming wind and how much the undisturbed wind velocity is affected by the actuator disk [47]. The induction factor will be discussed further in the next section but it is important to mention here as the blade element theory requires an estimation of the inflow to properly calculate α. TSR is the angular velocity of the turbine to the incoming free stream wind and is defined as

$$\lambda = \frac{\omega R}{V_\infty}. \tag{2.3}$$

Let the pitch control angle be β, then the effective angle of attack, or inflow angle, becomes

$$\psi = \alpha'(\theta, \lambda) + \beta. \tag{2.4}$$

From Eq. (2.2) and (2.3), it can be seen that the VAWT may experience dynamic stall at some positions when λ is small due to the high angles of attack. With pitch control, this situation can be mitigated or completely avoided.

The lift and drag forces generated by the airfoil can be determined using the blade element approach, given by

$$\vec{L} = \frac{1}{2}\rho V_R^2 C_L(\alpha) A_b \tag{2.5}$$

$$\vec{D} = \frac{1}{2}\rho V_R^2 C_D(\alpha) A_b \tag{2.6}$$

where ρ is the air density, C_L and C_D are the lift and drag coefficients, and A_b is the blade area defined as $A_b = ch$ for a straight blade with a chord length of c and a height of h. The blade element approach discretizes the length of the blade to find the performance of the blade in a 2D plane. These forces are then integrated along the length of the blade to determine the total thrust, torque, and power of the blade and the turbine as a whole. To determine the thrust force and torque in a simpler manner the lift and drag forces are decomposed into the form of normal and tangential components decomposed by the inflow angle given by

$$\vec{F_N} = \vec{L}\cos\psi + \vec{D}\sin\psi, \tag{2.7}$$

$$\vec{F_T} = \vec{L}\sin\psi - \vec{D}\cos\psi. \tag{2.8}$$

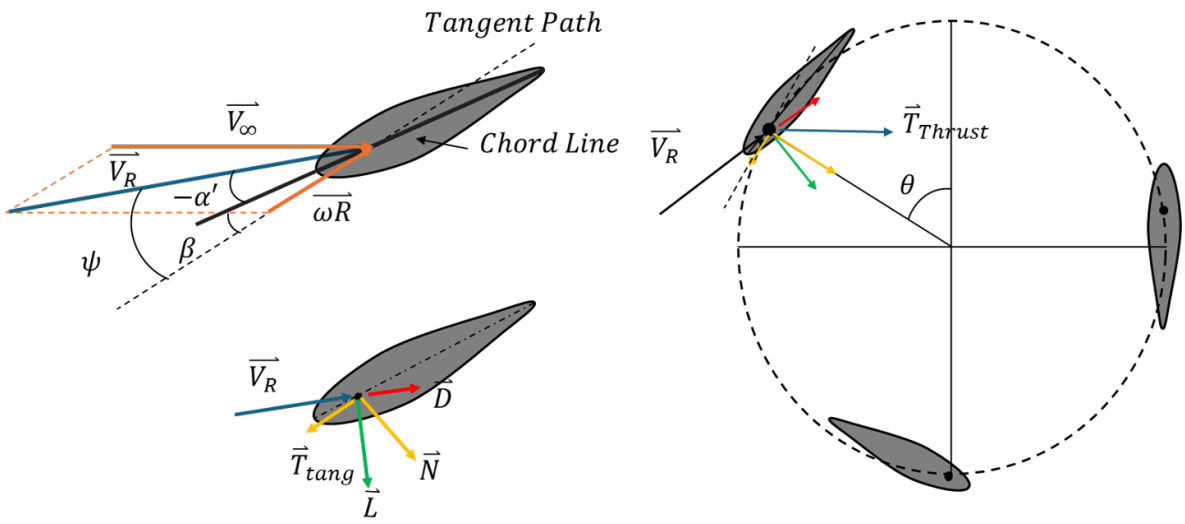


Figure 2.2: Aerodynamic forces acting on blades.

The average torque generated by the turbine is given by

$$\tau = \frac{1}{2\pi} \sum_{i=1}^{N_B} \int_0^{2\pi} \vec{F}_{T,i}(\theta) R d\theta \tag{2.9}$$

where N_B is the number of blades. Then, the average power generated by the turbine becomes,

$$P = \frac{1}{T} \sum_{i=1}^{N_B} \int_0^{2\pi} \vec{F}_{T,i}(\theta) R\omega d\theta \tag{2.10}$$

where $1/T$ is the period of the turbine. The available wind power traveling through the turbine can be calculated by

$$P_w = \frac{1}{2} \rho A_f V_\infty^3 \tag{2.11}$$

where $A_f = 2Rh$ is the frontal area of the turbine. Hence, the efficiency of a turbine, i.e., the coefficient of power, can be calculated by

$$C_p = \frac{P}{P_w} \tag{2.12}$$

2.1.2 Double Multiple Stream Tube Model (DMST)

The DMST model has been a popular aerodynamic model for the performance prediction of the VAWT amongst researchers due to its proven success over the years [15], [27], [33], [36]. Although considered a low-fidelity model, the DMST is considered a great tool for early development of the VAWT and fast analysis with dependable results that capture the overall performance of the VAWT, given its geometric parameters and expected operating environment. In total four stream tube models have been studied [48]. The DMST model is the latest evolution of the stream tube model growing in complexity from the Single-streamtube model that considered only one actuator disk and one stream tube to a dual actuator disk with multiple stream tube model, or simply the DMST. The DMST model relies fundamentally on both the momentum theory and blade element theory to deduce the induction factor, a, by equating the thrust force obtained by each and iteratively solving for said induction factor. Due to its low computational requirement and its ability to capture the VAWT general performance it is suitable for our pitching control studies. Although it should be noted the assumptions made by DMST which are listed in [47] and reiterated here,

- considers homogeneous, incompressible, steady-state fluid flow
- no frictional drag
- an infinite number of blades
- uniform thrust over the disk or rotor area
- a nonrotating wake

- static pressure far upstream and far downstream undisturbed ambient static pressure.

From the limitation above we can see the DMST model is limited to an in-depth analysis of the VAWT, as many models are. Still, the DMST model has been shown to reflect accurate performance predictions of the VAWT when used within the limits or scope of the model [15], [36].

The derivation of the DMST considers the conservation of linear momentum that is applied to a control volume around the VAWT as shown in Figure 2.3. The DMST model inserts two infinitesimal thin actuator disks in place of the turbine, one on the upper wind side of the turbine and the other in the downwind side of the turbine. Given the linear conservation of momentum, the thrust force across each disk can be determined for both the upwind actuation disk and lower as

$$T_{up} = \dot{m}_d(V_\infty - V_e) \tag{2.13}$$

$$T_{down} = \dot{m}_{d'}(V_e - V_w). \tag{2.14}$$

V_∞ is the undisturbed incoming wind velocity, V_e is the induced air velocity after passing the first actuator disk, and V_w is the induced velocity after passing the second actuator disk known as the wake velocity. Likewise, the thrust can also be found across each actuator disk by applying Bernoulli's function to the closed control volume across each

actuator. The following is for the upwind actuator disk,

$$\begin{aligned} p_\infty + 1/2\rho V_\infty^2 &= p_d^+ + 1/2\rho V_d^2 \\ p_d^- + 1/2\rho V_d^2 &= p_e + 1/2\rho V_e^2 \end{aligned} \tag{2.15}$$

and for the downwind actuator disk,

$$\begin{aligned} p_e + 1/2\rho V_e^2 &= p_{d'}^+ + 1/2\rho V_{d'}^2 \\ p_{d'}^- + 1/2\rho V_{d'}^2 &= p_w + 1/2\rho V_w^2 \end{aligned} \tag{2.16}$$

and finally recognizing thrust across each actuator can be defined as

$$T_{up} = A(p_d^+ - p_d^-) \tag{2.17}$$

$$T_{down} = A(p_{d'}^+ - p_{d'}^-). \tag{2.18}$$

The pressure values are obtained through Eqs. (2.15) for p_d^+ and p_d^- and the same for $p_{d'}^+$ and $p_{d'}^-$ using Eq. (2.16) thus the following equation for thrust is given as

$$T_{up} = 1/2\rho A_d(V_\infty^2 - V_e^2) \tag{2.19}$$

$$T_{down} = 1/2\rho A_{d'}(V_e^2 - V_w^2). \tag{2.20}$$

Finally equating Eqs. (2.19) and (2.13) and likewise for the downstream actuator disk (2.20) and (2.14) and recognizing $\dot{m} = \rho A_d V_d$ over the actuator disk, $\rho A_{d'} V_{d'}$ for the lower actuator disk, we can conclude the air stream velocity across each actuator disk is the

average of the upwind velocity and downstream

$$V_d = \frac{V_\infty + V_e}{2} \tag{2.21}$$

$$V_{d'} = \frac{V_e + V_w}{2}, \tag{2.22}$$

assuming the wake for the downstream actuator disk is fully developed [48]. The induction factor is defined as the fractional decrease of free stream air across the actuator disk,

$$a = \frac{V_\infty - V_d}{V_\infty} \tag{2.23}$$

$$a' = \frac{V_e - V_{d'}}{V_e}, \tag{2.24}$$

where a is the induction factor for the upper actuator disk and a' for the downwind actuator. These induction factors are then used to determine the velocity across each actuator disk for V_d and $V_{d'}$ using the previous Eqs. (2.21) and (2.22) to determine the inflow in terms of the induction factor to give,

$$V_d = (1 - a)V_\infty \tag{2.25}$$

$$V_{d'} = (1 - a')(1 - 2a)V_\infty. \tag{2.26}$$

The DMST model divides the turbine into two half cycles and splits the turbine into multiple adjacent aerodynamic stream tubes for analysis. The turbine is divided into equal arc lengths of $2N_{st}$, two times the number of stream tubes, and therefore arc

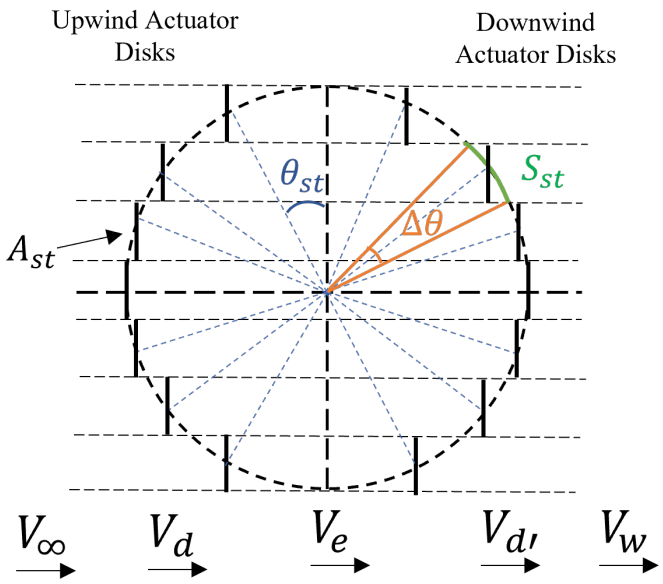


Figure 2.3: Double multiple stream tube model.

length of S_{st} and $\Delta\theta = \pi/N_{st}$. Each stream tube is located around the azimuthal of the turbine and located by its angle denoted as θ_{st} which corresponds to the center of the actuator disk. The frontal area of each actuator disk is defined as $A_{st} = dhR\Delta\theta \sin\theta_{st}$. See Figure 2.3 for depiction. To solve for the induction factor the thrust force obtained by the BE method is equated to the linear conservation of momentum thrust. Therefore the induction factor is solved iteratively. The thrust coefficient at each stream tube using the BE method is defined as the average thrust force at the stream tube location divided by the dynamic pressure and swept area of the actuator disk,

$$C_{t,s} = \frac{T_{st,avg}}{1/2\rho V_\infty^2 A_{st}} \tag{2.27}$$

where $T_{st,avg}$, and A_{st} are the average thrust force and the area of stream tube st. The average thrust force in the stream tube can be further written as $T_{st,avg} = B\frac{\Delta\theta}{2\pi}T_{inst}$ where $T_{inst} = 1/2\rho V_{rel}^2 c(-C_t cos(\theta) + C_n sin(\theta))$. The number of blades is defined by B and the

chord length as c. By submitting the calculation of the thrust force, the thrust coefficient can be further written as the average thrust coefficient of the turbine as

$$CT_{BET} = \frac{Bc}{2\pi R}\left(\frac{V_R}{V_\infty}\right)^2\left(-C_t\frac{\cos\theta}{\sin\theta}+C_n\right), \tag{2.28}$$

$$CT'_{BET} = \frac{Bc}{2\pi R}\left(\frac{V'_R}{V_e}\right)^2\left(-C_t\frac{\cos\theta}{\sin\theta}+C_n\right). \tag{2.29}$$

The momentum theory therefore also provides a relation between the induction factor and the stream tube loading derived from Eqs. (2.21) and (2.22) and Eqs. (2.25) and (2.26) to define two more important variables V_e and V_w. The downstream wake of the first actuator is defined as $V_e = V_\infty(1-2a)$ and the downstream wake of the second actuator is defined as $V_w = V_\infty(1-2a)(1-2a')$. These values are then used in the thrust equation derived from the Bernoulli principle, Eqs. (2.19) and (2.20), and divided by the dynamic force $1/2\rho V_\infty^2 A_{st}$ for the upwind and $1/2\rho V_e^2 A_{st}$ for the downwind actuator which then gives the thrust coefficient from the conservation of linear momentum,

$$CT_{MOM} = 4a(1-a), \tag{2.30}$$

$$CT_{MOM} = 4a'(1-a'). \tag{2.31}$$

The DMST solver equates Eqs. (2.28),(2.30) and (2.29),(2.31) to iteratively solve for the induction factor recognizing that wind velocities from the BE theory are also functions of the axial induction factor, a. Once the induction factors have been solved the summation of torque multiplied by rotational speed at each stream tube is obtained and divided by

the available wind power to find the overall coefficient of power of the wind turbine. The indices of the first summation indicate the frontal and downwind power generation, 1 and 2 respectively.

$$C_P = \sum_{i=1}^{2} \frac{\sum_{st=1}^{N_{st}} \frac{NB}{2\pi} \int_0^{2\pi} \tau_{i,st}(\theta, a_{st}) \omega d\theta}{1/2 \rho A_f V_\infty^3} \tag{2.32}$$

2.1.3 System Description and Dynamic Modeling of VAWT

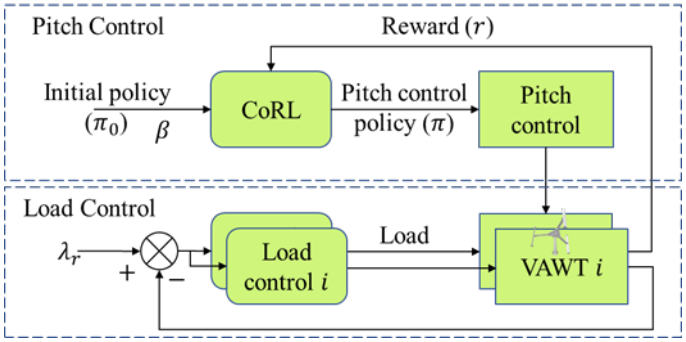


Figure 2.4: Control system block diagram.

The block diagram of the high-level overview of the control system is given in Figure 2.4, which illustrates the integration of the pitch control and load control subsystems, as well as their interactions with the aerodynamics of the turbine. The block diagram is an interpretation from [49] and [28] that describe the aerodynamic, mechanical, electrical, and control blocks in detail, but in this study, the mechanical and electrical blocks are not included due to their low-level detailed design analysis, which is not in the scope of this thesis. In addition, these systems, although important, will not add to the analysis of turbine performance as the turbine dynamics is considered a black box to the pitch control algorithm, the main focus of this thesis, and only requires the rotor aerodynamics. Furthermore, all inputs and outputs are considered high-level, requiring a subsystem to interpret these values. For example, the input load control value would need to

be interpreted by a micro-controller that would then require the need to switch high-efficiency electronic components to operate the resistive load [39] and therefore out of the scope of this study. The goal for load control is to ensure the turbine works at the designed optimal TSR which is a function of the geometric parameters of the turbine. Meanwhile, the pitch control optimizes the blade pitch angle, maximizing lift forces and thus improving the coefficient of power.

Load Control Block

The load control block is a single input and single output control system that keeps the TSR at a reference value, λ_r. The reference value is set at the optimal operating TSR of the turbine which depends mainly on the solidity of a turbine as previously discussed. The solidity of a turbine is a key parameter in determining the performance of the turbine and is defined as the ratio of the blade area to the frontal swept area of the turbine,

$$\sigma = \frac{N_B c}{d}. \tag{2.33}$$

A large number of high-fidelity computational fluid dynamics simulations concluded that the optimal TSR can be determined through only the knowledge of the turbine geometric parameters [22]. Rezaeiha et al. finds an optimal VAWT has a moderate to high solidity given it is a variable speed VAWT for urban applications. Moderate to high solidity VAWT achieves their maximum power coefficient at a lower TSR but at the cost of increased angles of attack. Therefore, using the findings of [22] the optimal TSR of the turbine in question is found using the solidity of the turbine, and a variable speed control

is implemented based on the correlation derived from [22],

$$\lambda_{opt} = 2.693\sigma^{-0.329} - 1.605. \tag{2.34}$$

For the implementation of the variable speed control intuitively, if the rotation speed of the turbine is too fast/slow, the control system will increase/decrease the load. This control behavior can be formulated using a simple PID control or any other control method where $\lambda_r = \lambda_{opt}$.

The dynamics model of a VAWT is given by a simple one-mass model,

$$J\dot{\omega} = \tau_a - \tau_l, \tag{2.35}$$

where J is the mass momentum of inertia, $\dot{\omega}$ is the angular rate, τ_l is the resistive load torque, and τ_a is the aerodynamic torque, given by

$$\tau_a(\theta, \psi) = \frac{1}{2}\rho A_f V_\infty^2 R C_\tau(\theta, \psi) \tag{2.36}$$

where C_τ is the total torque coefficient of the turbine and is a function of the turbine's angular position θ, the inflow angle of attack of each blade, and A_f the frontal area of the turbine. The resistive load, τ_l, is the controlling parameter, or the input control value. This value is used by the control algorithm chosen, which in this study is a simple PI controller and will be discussed in the next section. The inputs to the aerodynamic block are wind speed V_∞, rotor speed ω_r, and pitch angle β. The output of the aerodynamic block is the updated rotor speed ω_{new} and the coefficient of power C_P.

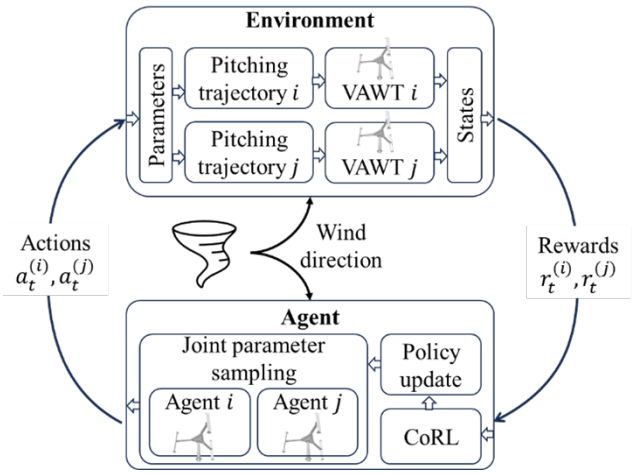


Figure 2.5: Structure of CoRL pitch control.

Pitch Control Block

Since the wind traveling through a VAWT can change at any time, increasing the learning speed is critical for the success of online RL pitch control. To improve the efficiency in exploration, thereby the speed of convergence, a collaborative RL pitch control method is proposed, and its structure is shown in Figure 2.5. Similar to a typical RL structure, it consists of an agent and the environment it interacts with. Differently, the collaborative RL (CoRL) algorithm strategizes the sampling and decision-making process to capture the possible nonlinear relationship between the actions and behavior of the turbines. Pitch control optimizes the pitch trajectory iteratively as it interacts with the environment, in this case, the aerodynamic block, by receiving a reward and adjusting the parameters of the pitching control policy accordingly. The actions are generated according to this pitching control policy and then sent back to the turbine in the environment, which updates the states of the turbines and provides feedback to the agents in the form of rewards or penalties. With enough trial-and-error explorations, the agent can learn a policy that optimizes its actions to maximize the cumulative rewards it receives

from the environment.

2.2 Control System of VAWT

2.2.1 Policy Gradient Policy Exploration (PGPE) for Pitch Optimization

Policy Gradient Policy Exploration (PGPE) introduced a new way to learn using model-free reinforcement learning, specifically designed for partially observable Markov decision problems. The method proposed by [46] estimates a likelihood gradient by sampling directly over the parameter space, leading to more accurate gradient estimations, compared to methods like REINFORCE. The PGPE algorithm is a deterministic policy that can determine the entire action-state history on a single sample, as opposed to a probabilistic policy that samples every action, leading to high variance in the gradient estimation. Like other PG methods, PGPE can be applied to a high dimensional, continuous state-action stochastic environment.

Markov Decision Process

PGPE basis its general framework of episodic reinforcement learning in a Markovian environment. It follows the Markovian decision process where the agent takes an action in an environment following a policy which leads to rewards based on states and actions $r(a_t, s_t)$ at a time step t. Almost all RL problems take the form of the Markov Decision Processes (MDP) where actions not only determine immediate rewards but also future rewards and future state interaction as well [50]. MDP is a class of sequential decision-making processes where an action influences the outcome of an event conditionally independent from the past given the present state. An MDP consists of 5 elements

that define the probability distribution from one state to another

$$M = (S, A, R, P, s_0) \tag{2.37}$$

where S the set of state space, $A \in IR^N$ is the action space, R is the reward function $r(s_t, a_t)$, P the transition dynamics $p(s_{t+1}|s_t, a_t)$, and s_0 the initial state distribution. In the RL problem, the action is taken according to a policy function that defines the agent's behavior. Of course, in the RL problem, the transition dynamics are unknown and must be learned through exploration and sampling methods such as the Monte Carlo method. Although exploration is key to finding optimal trajectories it can also lead to known unwanted state interactions that detract the agent from its goal, that is to maximize the long-term reward and therefore the need for efficient exploration methods.

Policy Gradients

Policy gradients are a very popular algorithm in RL methods as they expand into control for continuous states and actions as opposed to discrete values. This method searches the gradient in the policy space directly rather than from a value function and tries to derive the best policy by optimizing for the parameters ϑ in the parameterized function $\pi_\vartheta(a|s)$. The goal of PG methods is to find the set of parameters ϑ that maximize the agent's expected reward

$$J(\boldsymbol{\vartheta}) = \int_H p(h|\boldsymbol{\vartheta}) r(h) dh, \tag{2.38}$$

where T is the history rollout or sequence of state-action pairs defined $h = [s_{1:T}, a_{1:T}]$, $r(h)$ the reward function $\sum_{t=1}^{T} r(t)$, and $p(h|\boldsymbol{\vartheta})$ the probability density of the history

conditionally dependent on the parameters. To maximize the reward function then the gradient of the expected return is taken, in addition to using the log trick $\nabla_x y(x) = y(x)\nabla_x \log_x$, as

$$\nabla_\theta J(\boldsymbol{\vartheta}) = \int_H p(h|\boldsymbol{\vartheta}) \nabla_\vartheta \log p(h|\boldsymbol{\vartheta}) r(h) dh. \tag{2.39}$$

Because the environment is Markovian the states are conditionally independent from the parameters and the actions taken and therefore the probability density can be defined as

$$p(h|\boldsymbol{\vartheta}) = p(s_0) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t) p(a_t|st, \boldsymbol{\vartheta}) \tag{2.40}$$

and applying Eq. (2.40) to Eq. (2.39) we are left with

$$\nabla_\vartheta J(\boldsymbol{\vartheta}) = \int_H p(h|\boldsymbol{\vartheta}) \sum_{t=1}^{T} \nabla_\vartheta \log p(a_t|s_t, \boldsymbol{\vartheta}) r(h) dh. \tag{2.41}$$

Taking the integral of this gradient is unfeasible over the histories and therefore is approximated by using sampling methods such as the Monte Carlo method and expressed as

$$\nabla_\vartheta J(\boldsymbol{\vartheta}) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} \nabla_\vartheta \log p(a_t^n|s_t^n, \boldsymbol{\vartheta}) r(h^n). \tag{2.42}$$

Equation (2.42) is the base equation for the PG methods where different algorithms model $p(a_t|s_t, \boldsymbol{\vartheta})$ differently. Traditional PG methods model $p(a_t|s_t, \boldsymbol{\vartheta})$ as a probabilistic policy where the policy is defined as a parametric function approximator that outputs the probabilities of taking different actions, $a \sim \pi_\vartheta(a|s)$, where PGPE replaces this with a probability distribution over the parameters $\boldsymbol{\vartheta}$ themselves [51].

PGPE Algorithm

Policy Gradient Parameter Exploration as mentioned in the previous section deviates from typical PG methods in that it optimizes its search over the parameters. In taking actions from a policy at every time step, [46] contest leads to high variance when sampling over the histories and therefore leads to a noisy gradient due to the differentiation of policy to the parameters and sampling at each time step. On the other hand, PGPE employs a deterministic control and can generate an entire history from one parameter sample reducing the variance in the gradient estimation. Reducing "samples-per-history" is what [46] reports reduces high variance. This is defined as

$$p(a_t, s_t, \boldsymbol{\varrho}) = \int_{\vartheta} p(\boldsymbol{\vartheta}|\boldsymbol{\varrho})\delta_{F_\vartheta(s_t),a_t} d\boldsymbol{\vartheta}, \tag{2.43}$$

where $\boldsymbol{\varrho}$ is the parameters determining the distribution over the parameters $\boldsymbol{\vartheta}$, $F_\vartheta(s_t)$ the deterministic action chosen by the model with parameters $\boldsymbol{\vartheta}$ in s_t, and δ the Dirac delta function. The expected reward is then given $\boldsymbol{J}(\boldsymbol{\varrho})$ as

$$J(\boldsymbol{\varrho}) = \int_{\Theta}\int_{H} p(h, \boldsymbol{\vartheta}|\boldsymbol{\varrho})r(h)dhd\boldsymbol{\vartheta}. \tag{2.44}$$

Following the same steps as with the derivation of PG sampling methods by taking the gradient of $\nabla J(\boldsymbol{\varrho})$, using the "log trick" again, recognizing that h is conditionally independent of $\boldsymbol{\varrho}$ given the parameters $\boldsymbol{\vartheta}$, $p(h, \boldsymbol{\vartheta}|\boldsymbol{\varrho}) = p(h|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}|\boldsymbol{\varrho})$, we can express the

gradient estimation of $\nabla_{\varrho} J(\boldsymbol{\varrho})$ as

$$\nabla_{\varrho} J(\boldsymbol{\varrho}) \approx \frac{1}{N} \sum_{n=1}^{N} \nabla_{\varrho} \log p(\boldsymbol{\vartheta}|\boldsymbol{\varrho}) r(h^n). \tag{2.45}$$

Therefore Eq. (2.45) gives the sampling over the parameter distribution. See [51] for further derivation. Finally, the parameter $\boldsymbol{\varrho}$ in the basic PGPE form is represented as a set of mean parameters μ_i and deviations σ_i that define a normal distribution for each parameter of $\boldsymbol{\vartheta}$ and are represented as

$$p(\vartheta_i|\varrho_i) = \mathcal{N}(\vartheta_i|\mu_i, \sigma^2) \tag{2.46}$$

$$p(\vartheta_i|\varrho_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{\frac{1}{2}(\frac{\vartheta_i - \mu_i}{\sigma})^2}. \tag{2.47}$$

Taking the derivative of $\log p(\vartheta|\varrho)$ w.r.t. μ_i and σ_i we get the gradient representation to make the update rules of PGPE

$$\nabla \mu_i \log p(\boldsymbol{\vartheta}|\boldsymbol{\varrho}) = \frac{(\vartheta_i - \mu_i)}{\sigma_i^2} \tag{2.48}$$

$$\nabla \sigma_i \log p(\boldsymbol{\vartheta}|\boldsymbol{\varrho}) = \frac{(\vartheta_i - \mu_i)^2 - \sigma_i^2}{\sigma_i^3}. \tag{2.49}$$

Equations (2.48) and (2.49) can be substituted into Eq. (2.45) to approximate the reward gradient. Given enough samples, this approximation can be achieved up to arbitrary accuracy. Equation (2.45) also implies that each sample requires a rolling out of the entire state-action history.

2.2.2 PI Speed Control

The PI controller is implemented for variable speed control setting the optimal operating TSR for the turbine as the reference point. The error is therefore the difference between the actual and desired, $e = \lambda - \lambda_r$. The resistive torque load is then defined as

$$\tau_l = max(K_p\, e + K_i \int e\, dt, 0) \tag{2.50}$$

where K_p and K_i are the gains for the proportional and integral terms. This load controller applies load if the actual TSR is higher than the desired value.

CHAPTER 3

Collaborative Dual Wind Turbine Reinforcement Learning

In Chapter 2, the introduction to the base PGPE algorithm was introduced, and the general control structure was presented. In this Chapter, further enhancement of the base algorithm will be presented and extended to the development and formulation of our twin VAWT setup. In addition, the application of the base PGPE algorithm to the single wind turbine setup with added improvements where applicable. The use of the DMST software and MATLAB to study the VAWT base performance and develop ground truth for the PGPE algorithm will be presented. Finally, the collaborative learning between the two turbines will be explained in detail, and the forward Euler method will be used to step incrementally through time which will also be discussed.

3.1 Policy Updates with CoRL - Symmetric Sampling

To further enhance the base PGPE algorithm the author [51] updated the model to include reward baseline, symmetric sampling, and reward normalization to which we have included and taken advantage of here. To consider the previous explorations, sampling with a baseline reward is used. The baseline reward r_b is defined as a moving average reward over previous samples. By defining the step size as $\gamma = \gamma\sigma^2$, the parameter update can be given by

$$\begin{aligned} \Delta\mu_i &= \gamma_\mu(r - r_b)(\vartheta_i - \mu_i) \\ \Delta\sigma_i &= \gamma_\sigma(r - r_b)\frac{(\vartheta_i - \mu_i)^2 - (\sigma_i)^2}{\sigma_i}. \end{aligned} \tag{3.1}$$

The reward baseline ensures the gradient estimate converges faster as the parameter's respective rewards $r(t)$ will be compared against a moving average reward r_b. If the reward $r(t) > r_b$ then the parameters chosen will be more likely to be moving in the correct direction of the gradient whereas the opposite is true if $r(t) < r_b$ and therefore the parameters chosen will be likely avoided. Therefore adjusting $\boldsymbol{\varrho}$ in favor of the parameters $\boldsymbol{\vartheta}$ that will produce favorable results.

Another refinement made is symmetric sampling which allows for a better gradient estimation when the reward received is strongly skewed. Symmetric sampling samples from either side of the current mean of the parameter $\boldsymbol{\vartheta}$ where a perturbation is randomly chosen from $\epsilon \sim \mathcal{N}(0, \sigma)$ and added/subtracted from the parameter to give $\vartheta^+ = \mu + \epsilon$ and $\vartheta^- = \mu - \epsilon$. The reward produced by these trajectories are denoted as r^+ and r^- for ϑ^+ and ϑ^- respectively. Using these two samples the estimation using equation 2.45 gives

$$\nabla \mu_i J(\boldsymbol{\varrho}) \approx \frac{\epsilon_i (r^+ - r^-)}{2(\sigma_i)^2} \tag{3.2}$$

and using the same step size γ as before gives

$$\begin{aligned} \Delta \mu_i &= \frac{\gamma_\mu \epsilon_i (r^+ - r^-)}{2} \\ \Delta \sigma_i &= \gamma_\sigma \left(\frac{r^+ + r^-}{2} - b \right) \left(\frac{\epsilon_i^2 - (\sigma_i)^2}{\sigma_i} \right). \end{aligned} \tag{3.3}$$

Symmetric sampling is the foundation of collaborative learning for the dual wind turbine operation. In the learning environment, the symmetric samples imply each turbine, under the same policy, receives a different parameter distribution and a mirror trajectory and reward formed. By forming a mirror sample of each sample taken the weighted average

of the update improves even though it requires twice as much samples as they would for the single sampling method. This leads to a faster convergence and a better gradient estimation as is shown in [46] and will be shown here.

The final refinement is reward normalization. The reward normalization allows the updates to be independent of the rewards scale and therefore a poorly defined reward scale does not influence the learning. If the maximum reward that an agent can receive is known then this scalar will be defined as m and if it is not then the highest reward received thus far will be used. Then the μ updates will be divided by the difference of the maximum scalar reward m and the mean reward of the symmetric samples and the σ updates by dividing the maximum reward m and the reward baseline r_b giving

$$\begin{aligned} \Delta\mu_i &= \frac{\gamma_\mu \epsilon_i (r^+ - r^-)}{2m - r^+ - r^-} \\ \Delta\sigma_i &= \frac{\gamma_\sigma}{m - r_b} \left(\frac{r^+ + r^-}{2} - b \right) \left(\frac{(\epsilon_i)^2 - (\sigma_i)^2}{\sigma_i} \right). \end{aligned} \tag{3.4}$$

The policy formulation provides a flexible representation of pitching trajectories with six parameters, which makes online learning possible. Herein, the policy parameters are iteratively updated using CoRL policy gradient with parameter exploration. The policy parameters $\boldsymbol{\vartheta} = [c_1, ..., c_6] \in \Theta$ are assumed to be random mutually independent variables that follow a normal distribution $\vartheta \sim \mathcal{N}(\mu, \sigma^2)$, where μ and σ will be updated according to the explorative trials. Here, two turbines under the same wind conditions are used to explain the CoRL process. The two turbines are assumed to be identical and share the same optimal pitch control policy. In a typical PG method, the exploration method is usually carried out by perturbing the action found by the probabilistic policy, whereas

in PGPE, the perturbation is carried out directly over the parameter space by adding random noise to each parameter ϑ_i. For the two turbines new trails, i.e., $\vartheta^+ = \mu + \epsilon$ and $\vartheta^- = \mu - \epsilon$ are dependent and symmetrically sampled from the distribution $\epsilon \sim \mathcal{N}(0, \sigma)$. From the formulation of the pitching trajectory in Eq. (3.7), the pitching trajectory is entirely determined by the parameter samples $\boldsymbol{\vartheta_t}$. The state and reward at cycle t are $s_t = \{\lambda(t), C_P(t)\}$ and $r(t) = C_P(t)$, respectively. The expected reward given parameters $\varrho = \{\mu, \sigma\}$ and a sequence of state-action pairs $h = \{s_t, a_t | t = 1 : T\}$ can be found by the following integration over the state-action history space H and the parameter space Θ given by Eq. (2.44) and ultimately expressed as (2.45). The pseudo-code is shown in Algorithm 1. Here, the baseline reward is expressed as the exponential moving average

Algorithm 1: PGPE with Symmetric Sampling [46]

Initialize: $\boldsymbol{\mu}_0 = \mathbf{0}$
Initialize: $\boldsymbol{\sigma}_0 = \boldsymbol{\sigma_{init}}$
while *TRUE* **do**
  for *n=1 to N* **do**
    draw perturbation $\epsilon^n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I\sigma^2})$
    $\boldsymbol{\vartheta}^{+,n} = \boldsymbol{\mu} + \epsilon^n$
    $\boldsymbol{\vartheta}^{-,n} = \boldsymbol{\mu} - \epsilon^n$
    evaluate $r^{+,n} = r(h(\boldsymbol{\vartheta^{+,n}}))$
    evaluate $r^{-,n} = r(h(\boldsymbol{\vartheta^{-,n}}))$
  $\boldsymbol{T} = [t_{i,j}]_{i,j}$ with $t_{ij} := \epsilon_i^j$
  $\boldsymbol{S} = [s_{i,j}]_{i,j}$ with $s_{ij} := \frac{(\epsilon_i^j)^2 - \sigma_i^2}{\sigma_i}$
  $\boldsymbol{r}_T = [(r^{+,1} - r^{-,1}), ..., (r^{+,N} - r^{-,N})]^T$
  $\boldsymbol{r}_S = [\frac{(r^{+,1} - r^{-,1})}{2} - b, ..., \frac{(r^{+,N} - r^{-,N})}{2} - b]^T$
  update $\boldsymbol{\mu} = \boldsymbol{\mu} + \gamma_\mu \boldsymbol{T r_T}$
  update $\boldsymbol{\sigma} = \boldsymbol{\sigma} + \gamma_\sigma \boldsymbol{S r_S}$
  update r_b accordingly

over the previous rewards and defined as

$$r_b = m_b r_b + (1 - m_b) r_t \tag{3.5}$$

where m_b is the weight factor and r_t is the reward received at time t. The $\boldsymbol{\mu}$ center solutions in this study are initialized at zero although they can be initialized at an initial guess, which can speed up optimization. However, to show the effectiveness of the algorithm with additional updates to the code, we demonstrate the ability of the algorithm to find the optimal trajectory quickly without an initial guess.

3.1.1 Clip Up

Toklu et al., [45], proposed using new machine learning techniques to speed up gradient convergence by using adaptive update algorithms. The Clip Up optimizer ultimately combines the heavy ball momentum m_β, gradient normalization $\frac{\boldsymbol{g}}{||\boldsymbol{g}||}$, and lastly gradient clipping to control the rate at which the center solutions $\boldsymbol{\mu}$ are updated,

$$\boldsymbol{\mu}_{t+1} \leftarrow \boldsymbol{\mu}_t + ClipUp(\nabla f(\boldsymbol{\mu}_t)). \tag{3.6}$$

A common issue with gradient estimation is the gradient divergence or slow convergence when parameters pertaining to the optimization algorithm such as step size γ, parameter initialization $\boldsymbol{\mu_{init}}$, population size M, etc. are not set up properly. This is mainly due to the uncontrolled gradient estimation for the divergent case and the constant step size when updating the parameters for the slow convergence case. Clip Up uses heavy ball momentum m_β to follow the gradient direction, which is weighted heavier, allowing for the optimization to avoid outliers or directions that are not as frequent, thus helping the gradient converge faster. In other words, the gradient velocity will increase in the direction weighted favorably and negatively in those that are not. In addition to limiting the direction of the gradient, Clip Up optimizer also implements gradient normalization in

which the gradient magnitude is decoupled from its direction. Therefore gradient magnitudes are controlled by an additional mechanism. With a variable gradient magnitude, or non-normalized gradient, the step size must be adjusted accordingly to the reward scale therefore the sensitivity in the reward scale must be considered and changed throughout the training process or for different environment conditions. With gradient normalization, the step size γ is considered a hyperparameter for the Euclidean distance, $\frac{\boldsymbol{g}}{||\boldsymbol{g}||}$, and therefore independent of the reward scale. Finally, Clip Up implements gradient clipping that does not allow the gradient to go beyond the update speed threshold $\boldsymbol{v_{threshold}}$ can overshoot the local minimum and cause instabilities in the gradient. The Clip Up optimizer pseudo code is shown in Algorithm 2.

Algorithm 2: Clip Up Optimizer [45]

Initialize: Velocity $\boldsymbol{v}_0 = \mathbf{0}$
Hyperparameters: Step size γ_μ
Maximum speed v_{max}
Momentum m_β
Input: Estimated gradient $\nabla f(\boldsymbol{\mu}_k)$

$\boldsymbol{v'_{k+1}} \leftarrow m_\beta \cdot \boldsymbol{v}_k + \gamma_\mu \cdot (\nabla f(\boldsymbol{\mu})_k / ||\nabla f(\boldsymbol{\mu})_k||)$
if $||\boldsymbol{v'_{k+1}}|| > v_{max}$ **then**
  $\boldsymbol{v_{k+1}} \leftarrow v_{max} \cdot (\boldsymbol{v'_{k+1}} / ||\boldsymbol{v'_{k+1}}||)$
else
  $\boldsymbol{v_{k+1}} \leftarrow \boldsymbol{v'_{k+1}}$
return $\boldsymbol{v_{k+1}}$

3.2 Pitching Policy

The policy in an RL learning algorithm is usually represented by a parameterized function approximator. The goal of RL is then to optimize the parameters such that the function approximates the desired behavior. A feedback loop between the agent, environment, and the feedback reward function to the optimization algorithm achieves this. See Figure

2.5. Therefore there are various ways to define the pitching policy and even more ways to define the reward function. Given the previous pitching studies mentioned in the literature review, there are key takeaways of the pitching formulation to improve efficiency and power gains of the turbine listed here:

(1) pitching is dependent on individual blade position

(2) pitching amplitude is dependent on the turbine speed

(3) pitching must be smooth and continuous

(4) pitching ultimately influences torque fluctuation and power.

Given these insights, we employed two pitch policies that incorporate these key takeaways in the form of a mathematical expression, or policy.

3.2.1 Sine Policy

Given the cyclic operating nature of wind turbines, we know the control policy should also be a periodic function of the rotation angle of the turbine. Here, a transformed sine function is proposed to formulate the nonlinear policy which is taken from [52] with a slight modification that ensures continuity between the first and second half of the sine wave after independent vertical shifting. The policy is a continuous function necessary for the smooth operation of the turbine. Additionally, it can represent very flexible trajectories requiring as few as six parameters. The formulation of the policy is given by the following four operations, namely skewing, flattening, scaling, and shifting operations

$$\beta_{sine} = T_3 T_2 T_1(f_1) \tag{3.7}$$

where the base function is

$$f_1(\theta) = \sin(\theta + c_1) \tag{3.8}$$

where the skewing operation maps the above original sinusoidal function to a periodic function that has the same frequency and magnitude but with a skewness defined by the skewness factor $c_2 \in [-0.5, 0.5]$,

$$T_1 : f(\theta) \rightarrow \sin(\theta + c_1 + c_2 f(t)). \tag{3.9}$$

The flattening operation flattens the peak of the sinusoidal function,

$$T_2 : f(\theta) \rightarrow \sqrt{\frac{1 + c_3^2}{1 + c_3^2 f(t)^2}} f(\theta) \tag{3.10}$$

and the scaling and translating operations are given by

$$T_3 : f(\theta) \rightarrow \left(c_4 - \frac{c_4 - c_5}{1 + e^{-z_K((\theta - \pi) - y_K)}} \right) f(t) + c_6 \tag{3.11}$$

where the scaling term is defined as a scaled sigmoid membership function, it scales the first and second half cycle with a different factor of c_4 and c_5 and ensures the smooth and periodic properties of the scaling transformation. K is a constant factor that defines the slope of the transition in scaling and set as $K = [1, 10]$ for all simulations. The proposed pitching trajectory has the following advantages: 1) all the parameters have clear physical meaning and can be constrained according to the physical limits and prior information about the pitching trajectory; 2) it guarantees the smoothness of the pitch-

Table 3.1: Parameter definition.

Parameters	Definition
$\zeta(c_1)$	phase shift factor
$a(c_2)$	skewness factor
$b(c_3)$	flatness factor
$s_1(c_4)$	upper half scaling factor
$s_2(c_5)$	lower half scaling factor
$d(c_6)$	offset shift factor

ing trajectory; 3) build upon prior this formulation uses only six parameters c_1,c_2,. . . ,c_6, yet provides a very flexible trajectory representation. Figure 3.1 provides an illustration of the transformation process of the proposed pitching trajectory. Therefore, the param-

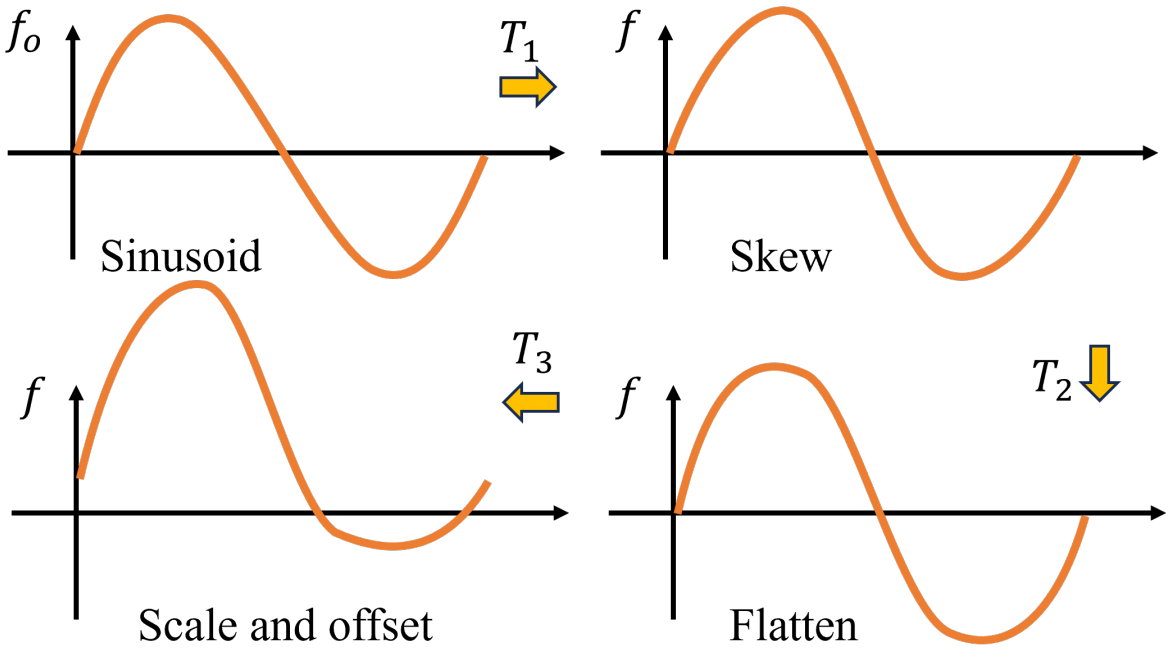


Figure 3.1: Illustration of pitching trajectory formulation.

eters, $\boldsymbol{\vartheta} = [c_1, c_2, c_3, c_4, c_5, c_6] = [\zeta, a, b, s_1, s_2, d]$, are the parameters optimized directly by the PGPE algorithm. Table 3.1 gives the summary of each parameter where each parameter is expressed by a center solution $\mu_{1,\ldots,6}$ and deviation $\sigma_{1,\ldots,6}$.

3.2.2 Spline Policy

The spline function is a smooth, continuous, and flexible function that meets the criteria for a good pitching policy candidate. Therefore, a spline function, or parameterized curve, is also implemented as a pitching policy. The spline policy was also set to have the same number of parameters to compare the results with the six-parameter sine policy. In contrast to the sine policy, the spline function is not constrained to a base function. Therefore the spline function can take any shape that could be a positive or negative feature given the parameters $\boldsymbol{\vartheta} = [c_1, c_2, c_3, c_4, c_5, c_6]$ are initialized at $\mathbf{0}$ and therefore starts as a flat line. This could lead to slow convergence or an unstable policy. On the other hand, the spline policy can lead to a more elegant solution than what the sine policy could give given it is not constrained to any particular function. Therefore the spline function from MATLAB is implemented as:

$$\beta_{spline} = spline(x, \boldsymbol{\mu}, \boldsymbol{\theta}) \tag{3.12}$$

where $x = length(\boldsymbol{\mu})$ is the number of equally spaced segments of the spline equal to the number of parameters, $\boldsymbol{\mu}$ are the center solution values, and $\boldsymbol{\theta}$ is the discretized azimuthal positions around the turbine given by the aerodynamic solver. Therefore, the PGPE algorithm directly optimizes for the parameters $\boldsymbol{\mu}$ and corresponding deviations $\boldsymbol{\sigma}$ optimizing the probability distribution over $p(\boldsymbol{\vartheta}|\boldsymbol{\varrho})$ to obtain the maximum reward. Figure 3.2 shows a representation of the spline policy with the six equally spaced parameters. In addition, it shows a general representation of the formulation of the spline policy using PGPE with symmetric sampling that shows the mean solution with its initial search dis-

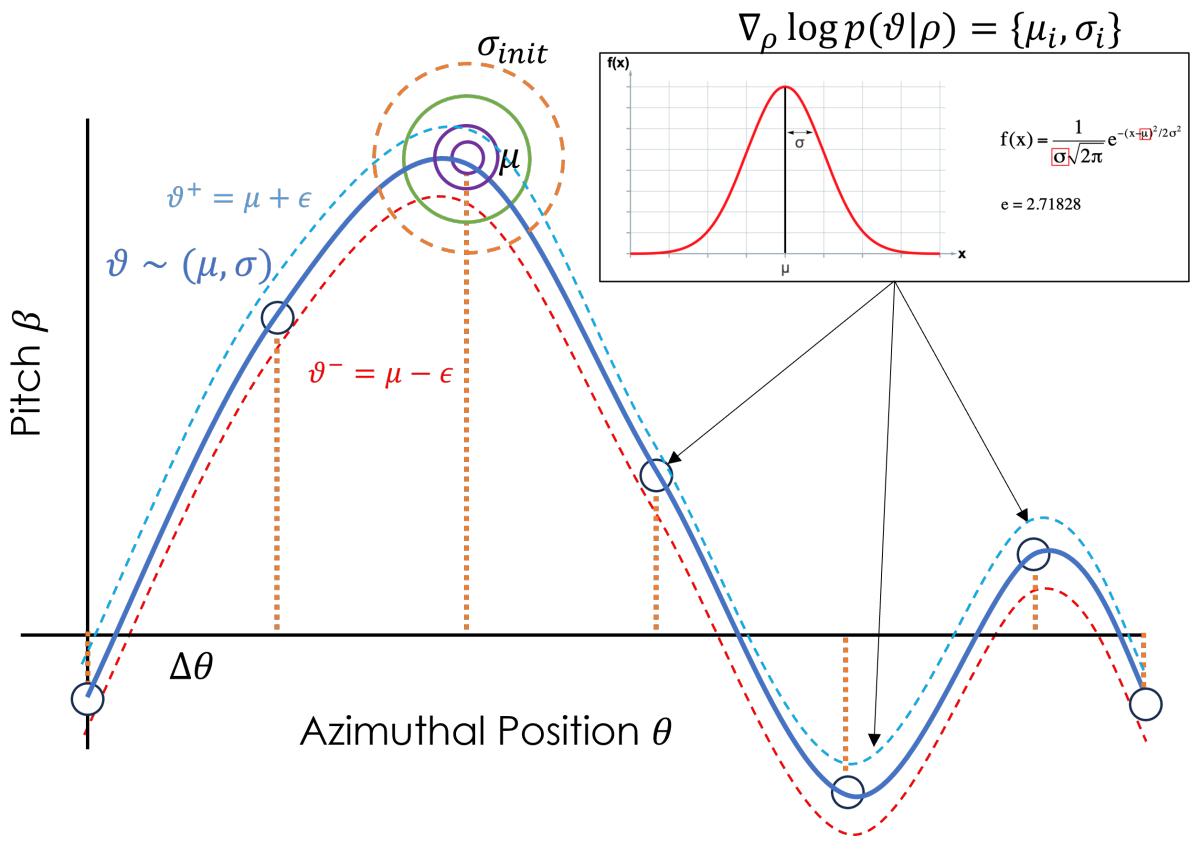


Figure 3.2: Spline policy formulation using PGPE with symmetric sampling.

tribution size. The parameter that samples from the normal distribution $\boldsymbol{\vartheta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is used by the base PGPE, whereas for the symmetrical sampling algorithm, the deviations are taken from $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma})$ and added to the center solutions where the parameters are $\boldsymbol{\vartheta} = \boldsymbol{\mu} \pm \epsilon$.

3.3 Twin VAWT Setup

For this study we chose to use the same VAWT dimensional parameters used in [28] and [53] which have experimental data as well as simulated data. In addition, we use the DMST model developed by [54], [55] which has been validated across other sources. Table 3.2 gives the turbine dimensions.

The solidity of the turbine is in the higher range which aligns with the goal of analyzing

Table 3.2: VAWT geometric parameters.

Geometric Parameters	**Values**	**Unit**
Chord Length c	0.246	m
Radius R	0.850	m
Number of Blades N	3	$-$
Moment of Inertia J	3	$Kg - m^2$
Blade Profile	NACA 0021	$-$
Aspect Ratio	1	$-$
Solidity σ	0.44	$-$

a VAWT that has its optimal operating speed in the lower range. Consequently, this also means the turbine will experience significant dynamic stall in its operation without any pitch control. Inputting the wind turbine geometric parameters given in Table 3.2 into Eq. (2.34) we get an estimated optimal TSR of 1.94. The discrepancy here, between Figure 3.3 and Eq. (2.34), lies in the fact that the optimal TSR given by Eq. (2.34) does not take into account any pitching though the coefficient of power is a function of both TSR and pitching, $C_p(\lambda, \beta)$. Whereas taking a closer look at Figure 3.3, we can see the curve for when the pitch is zero the optimal TSR leans much closer to 2.5. From the experimental results, and simulation, shown from [53] and [28] the optimal TSR, λ_{opt}, falls in the range of 2 - 2.5 which is what the current DMST gives. Figure 3.3 shows the performance of the given turbine under various TSRs and collective pitch. The best collective pitch/TSR combination is $\beta_{collective} = 6, \lambda = 2.5$ given by the DMST model.

3.3.1 MATLAB Pitch Optimization

Collective pitch is not as effective as it would be for the HAWT where the angle of attack α is constant throughout the azimuthal position of each blade. The VAWT, on the other hand, can benefit from individual blade pitching that is a function of both λ and θ. Because the pitch trajectory is an optimization problem that can be solved using

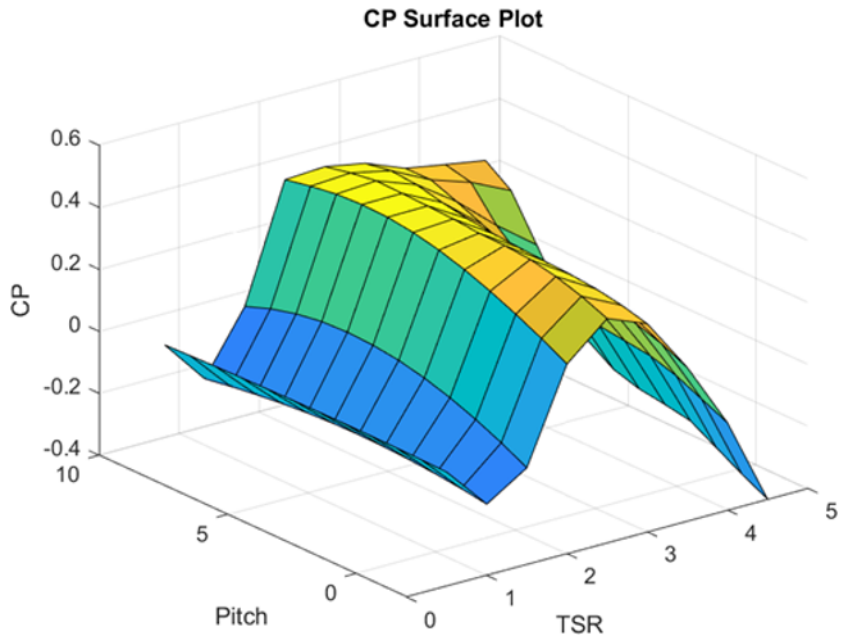


Figure 3.3: VAWT performance curve for collective blade pitching.

a constant TSR we turn to the MATLAB Optimization Toolbox to use their built-in nonlinear programming methods. The optimization algorithm used is the interior point (IPM) method. From the literature, optimal pitch trajectories follow a sinusoidal-like curve, and therefore, a sine wave was expected by using the spline function in MATLAB optimizing six parameters to formulate the pitch trajectory. The parameters, $c_1 ... c_6$, were equally spaced between $[0 : 2\pi]$ just as would be done for the PGPE algorithm. The optimization problem to minimize, or maximize in this case, is the power output from the turbine and the variable optimization variable β that represents pitch. Subject to a pitching range of - 20 to 20 degrees and the first and last parameters are constrained to

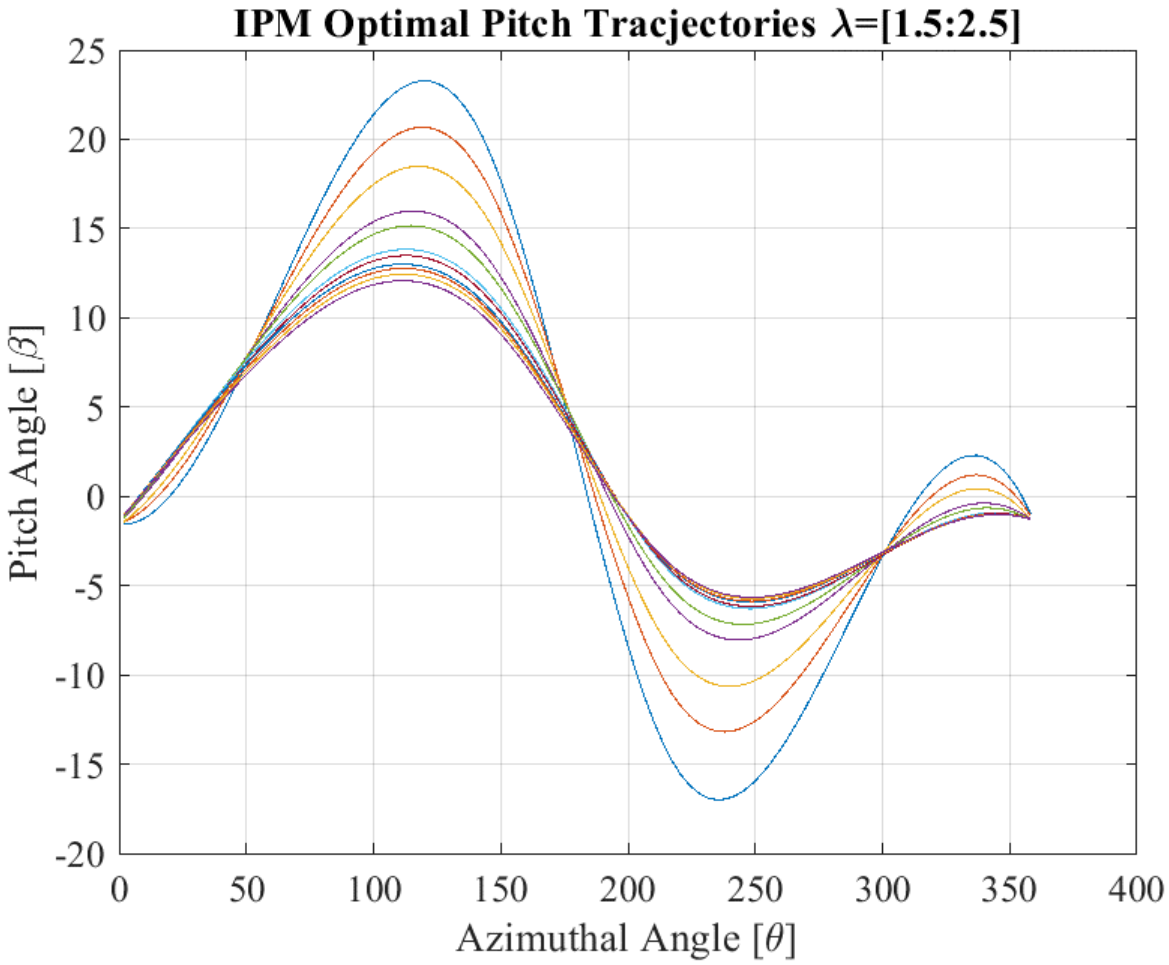


Figure 3.4: Optimal pitching trajectories under various tip speed ratios.

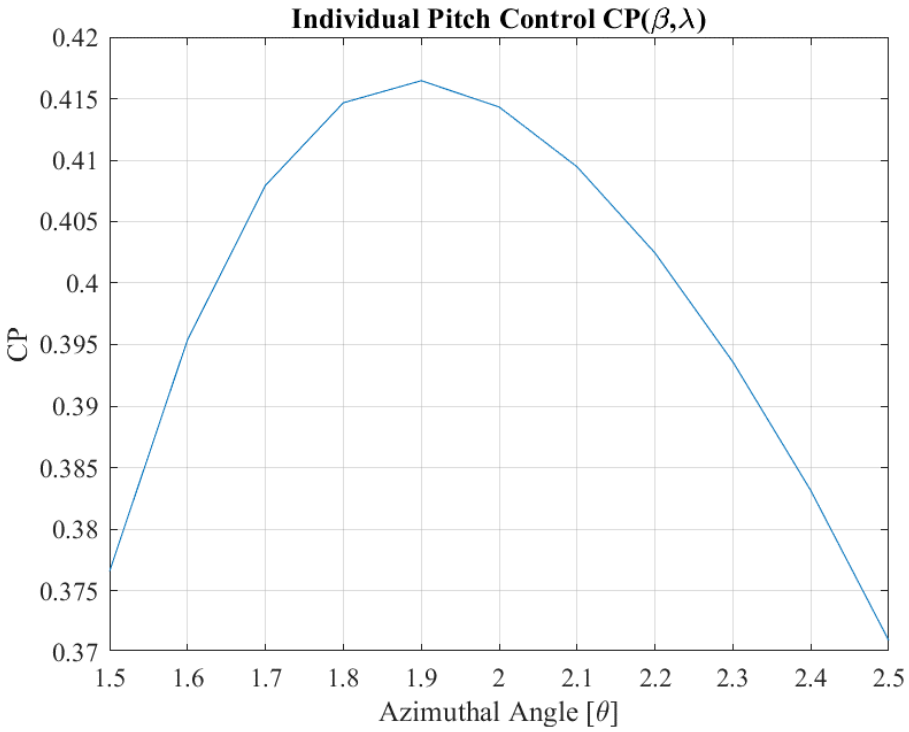
equal each other to ensure continuity between revolutions. In standard form:

$$
\begin{aligned}
&\textbf{Minimize:} \\
&min_{\beta} f(\boldsymbol{\beta}) : -P = 1/2 \rho A_f V_{\infty}^3 C_P(\lambda, \beta) \\
&\textbf{Subject to:} \\
&-20 \leq \boldsymbol{\beta} \leq 20 \\
&A_{eq} = [1, 0, 0, 0, 0, -1] \\
&b_{eq} = [\mathbf{0}]
\end{aligned}
\tag{3.13}
$$

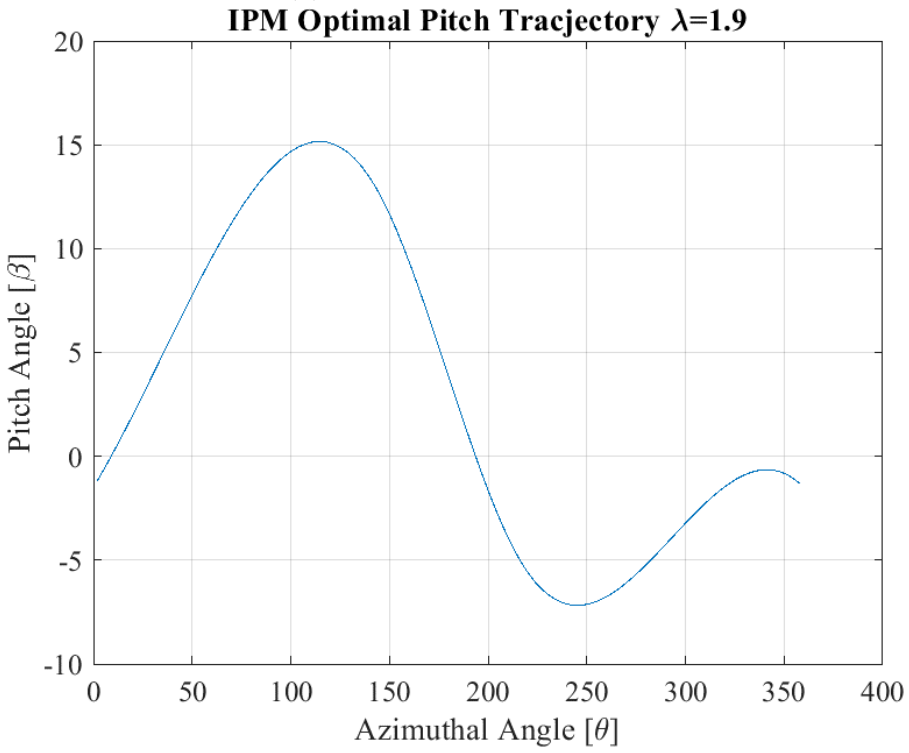
Figure 3.4 shows multiple optimal pitch trajectories for various TSRs, $\lambda = [1.5 : 0.1 : 2.5]$. We can see that, indeed, the optimal pitch trajectory follows a smooth sinusoidal-like trajectory. In addition, the lower the λ, the higher the pitching amplitude and vice-versa. From Figure 3.5a we can determine the pitch trajectory with the highest C_P falls at TSR of ~ 1.9 at $C_P = 0.4165$. Therefore, a TSR of 1.9, 1.94 as given by Eq. (2.34), will be the focus of the remaining study, and the optimized pitching trajectory from the MATLAB optimizer, Figure 3.5b, will be used as the ground truth for evaluating the PGPE-optimized pitch.

3.3.2 Environment

Symmetric sampling requires sampling two identical points in which these two points at time t are subject to the same observational state s_t. The algorithm thus perturbs the deterministic action by sampling from either side of the distribution by adding and subtracting noise $\epsilon \sim \mathcal{N}(0, \sigma)$ to the deterministic action determined by the parameters $\vartheta = \mu \pm \epsilon$ as discussed in Chapter 2. As mentioned earlier PGPE is meant to be implemented in an episodic case and therefore the algorithm is updated after each turbine revolution that ends a trajectory T. Consequently, following the deterministic action, pitching trajectory β_{pgpe} taken by the policy $\varrho(\mu, \sigma)$, both turbines end up in different states after the end of a trajectory, most importantly under different wind speeds and turbine rotational speeds. In a simulated problem, this is not an issue as the end of a turbine revolution ends the simulation and resets both turbines with the position $\theta = 0$ starting again, and thus all states to initial states $\boldsymbol{s_t} = \boldsymbol{s_{t_0}}$ or initial state distribution $p(\boldsymbol{s_0})$ as in this case. To bring this algorithm to a real-time online learning algorithm, the



(a) CP over various TSRs.



(b) Optimal Pitch Trajectory $\lambda_{opt} = 1.9$.

Figure 3.5: VAWT performance for individual pitch control.

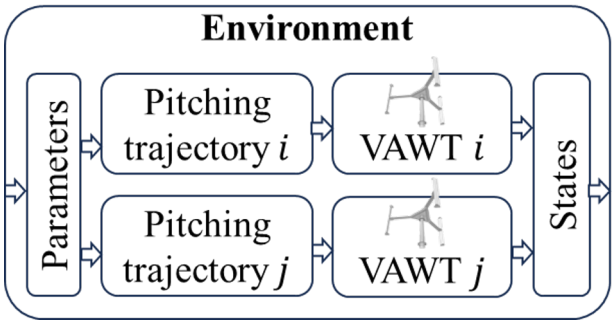


Figure 3.6: CoRL environment.

time scheme implemented when stepping through iterations requires special attention. There is also a need to overcome the discrepancy in observational states given that both turbines operate independently from one another and only share the policy as would be in a real environment. Although in a real environment time is absolute, here time is stepped forward using the Euler forward method given the turbine dynamics,

$$\begin{aligned} \omega_{t+1} &= \omega_t + \dot{\omega}\Delta t, \\ \Delta t &= \frac{2\pi}{\omega_t}, \\ \dot{\omega} &= \frac{\tau_{aero} - \tau_l}{J}. \end{aligned} \tag{3.14}$$

To overcome the issue of unsymmetrical samples the observational states had to be as close as possible, mainly the turbine speeds and perceived incoming wind, to keep both turbines operating under the same time scale and avoid gradient calculation issues. Thus, the major driver here is the turbine speeds, which influence the time stepped in the simulation as seen in Eq. (3.14), the interpolation of wind speed based on the time stamp, and the turbine position. Therefore, the importance of the PI controller that moderates both turbine speeds and from this, a constraint in the optimization process

arises:

$$TSR_1 - TSR_2 << 1 \tag{3.15}$$

The speed difference between turbines must be as small as possible to ensure both turbines operate under the same time scale and keep the policy updates from diverging and becoming unstable.

Following the sequence of events from Figure 3.6 the agent formulates two pitch trajectories by the method described in the beginning of this section based on the current policy ϱ_{pgpe}. The wind profile is either a constant value for the duration of training or is a variable wind profile $V_\infty(t)$ with a length larger or equal to the duration of simulation time. For the variable wind case, the wind values are interpolated based on the running 'real-time', $t_{new} = t_{old} + \Delta t$, and thus the importance of having both turbines run on the same time scale. Given the pitch trajectories, $\beta_{pgpe} = [\beta_i, \beta_j]$ and turbine states $s_t = [V_{\infty_i,j}, \lambda_{i,j}]$, the turbine plant is simulated and the output from the environment are the new states $\boldsymbol{s_{t+1}}$ and the reward $\boldsymbol{r_t} = [r_i, r_j] = [C_{P_i}, C_{P_j}]$ which is fed to the agent for optimization updates. The reward received from the environment is the coefficient of power out from each turbine. Therefore, the updates are made solely based on the power output.

3.4 Policy Updates for Single VAWT

3.4.1 PGPE for Single Turbine Operation

For the single turbine operation, the base PGPE algorithm is used. This is due to the non-symmetric application of the single turbine. In a simulation environment, the simulation of multiple turbines and their symmetric counteraction leads to a more robust

gradient estimation. The multiple simulation candidates refer to the population size in evolutionary algorithms such as PGPE and in a simulation environment the gradient estimation is greatly improved by having a larger batch. For either case in this study, dual and single turbine operation, the population size is one. The goal of this work is to show the use of PGPE as an online algorithm that does not rely on simulated virtual environments but rather learns as an online algorithm and therefore all turbine plants are assumed to be 'real' and continuous where states are passed on from the current episode to the next. One of the main problems with traditional controls, and even more modern controls such as Model Predictive Control (MPC), is the need for an accurate description of the aerodynamic model of the VAWT. If such a model was available the use of PGPE with a large population size would suffice to determine an optimal pitch control without the need for an online implementation, and for that matter, the other control methods described earlier as well. Yet there is no such model available and the reliance on simulated environments does not suffice to guarantee an optimal pitch control therefore increasing the population size in the simulated environment would not be as unuseful. Therefore, to show the usefulness of an online reinforcement learning algorithm to interact with the real-world environment and approximate the dynamics of the VAWT, we propose the use of PGPE with updates to demonstrate the effectiveness in learning the VAWT dynamics overcoming the shortfall of traditional control methods. The base PGPE pseudo code is shown in Algorithm 3.

The reward baseline r_b is updated the same as in the dual turbine operation, Eq. (3.5). In addition, the Clip Up optimizer is also implemented to update the center solutions $\boldsymbol{\mu}$ and standard deviation parameters $\boldsymbol{\sigma}$. The main difference as noted earlier

Algorithm 3: PGPE [46]

Initialize: $\boldsymbol{\mu}_0 = \mathbf{0}$
Initialize: $\boldsymbol{\sigma}_0 = \boldsymbol{\sigma}_{init}$
while *TRUE* **do**
  for *n=1 to N* **do**
    draw perturbation $\boldsymbol{\vartheta}^n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I\sigma^2})$
    evaluate $r^n = r(h(\boldsymbol{\vartheta^n}))$
  $\boldsymbol{T} = [t_{i,j}]_{i,j}$ with $t_{ij} := (\vartheta_i^j - \mu_i)$
  $\boldsymbol{S} = [s_{i,j}]_{i,j}$ with $s_{ij} := \frac{(t_{i,j})^2 - \sigma_i^2}{\sigma_i}$
  $\boldsymbol{r}_T = [(r^1 - b), ..., (r^N - b)]^T$
  update $\boldsymbol{\mu} = \boldsymbol{\mu} + ClipUp\boldsymbol{Tr_T}$
  update $\boldsymbol{\sigma} = \boldsymbol{\sigma} + \gamma_\sigma \boldsymbol{Sr_S}$
  update r_b accordingly

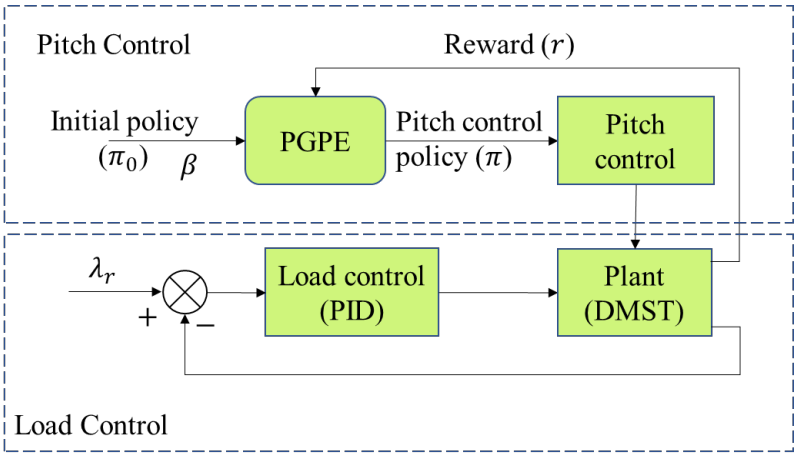


Figure 3.7: Control block diagram for single turbine.

is the optimization in which symmetric sampling is not done and therefore the gradient calculation differs. Here the parameter is directly sampled from $\mathcal{N} \sim (\boldsymbol{\mu}, \boldsymbol{I\sigma}^2)$ and evaluated directly. Thereafter, the gradient for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is estimated using Eq. (3.1) and finally applying reward normalization.

3.4.2 Environment

The environment for the single turbine operation is the same the dual turbine operation with the exception of the second turbine. The sequence of events is the same and the

inputs to the environment are the parameters by the agent and output the rewards and next states $\boldsymbol{s_{t+1}}$. Figure 3.7 shows the block diagram for the single turbine which is identical to the dual turbine operation. What differs the most is in the optimization technique as described in the previous section.

CHAPTER 4

Numerical Results

The performance of the proposed CoRL developed in Chapter 3 is verified using three wind profile simulation cases to showcase the difference in modifying the policy described in Chapter 3 and the comparison to a single wind turbine operation under constant and varying wind. The proposed sine policy function is used for the first simulation case with the dual turbine collaborative operation under a constant wind profile. The sine policy was used as an initial starting point as it is the more stable policy of the two. Variations in the sine policy parameters would not be as completely random as in the spline policy function where there were no constraints placed. In addition, due to the stability of the sine policy, the PI parameters were found experimentally during the training of the sine policy. After this, the spline policy was also trained under the constant wind profile with the same PI parameters as the sine policy case. Convergence time for all three cases under the constant wind profile and unsteady wind profiles were found and compared as well as their rewards. For the constant wind case the optimal pitch trajectory was compared to the optimized trajectory given by the MATALB optimizer shown in Chapter 3. Note for the dual collaborative operation, both policies used the same hyperparameters. The main difference was in the policy formulation. This is true for the remainder of the simulated cases. In the single turbine operation, the learning rate had to be reduced from $\gamma = 0.005$ to $\gamma = 0.0025$ to get the algorithm to stabilize and eventually converge.

The PI controller greatly impacted the stability and convergence of the collaborative pitch control in keeping both turbines operating at the same speed. Only once the

turbines reached their optimal operating speed could the algorithm begin optimizing the pitch trajectory irrespective of λ, keeping in mind that C_P is a function of both λ and β.

4.1 Initialization of Dual Turbines

The initialization of the PGPE borrows from [45], where the hyperparameters with the greatest attention on the maximum velocity v_{max}, step size for both center solution and deviation updates $\gamma_{\mu,\sigma}$, and the initialization of the standard deviation parameters $\boldsymbol{\sigma}_{init}$. The initialization of $\boldsymbol{\sigma}_{init}$ requires a distribution size that is not too small where exploration would be minimized and not too large that would cause an unsteady search policy and/or a slow convergence. The standard deviation is defined as the search radius from the center solution $\boldsymbol{r}_{pgpe} = ||\boldsymbol{\sigma}_{init}||$ and $\sigma_{init} = qv_{max}$ where q has a value between 10 to 20, although for our case we found a value of 30 to be adequate. In addition, due to the parameter space being in the radian space, the search radius was also brought to the radian space giving the standard deviation initialization as

$$\sigma_{init} = qv_{max}\left(\frac{\pi}{180}\right). \tag{4.1}$$

The maximum velocity is set as $v_{max} = \gamma_\mu/2$ and therefore all hyperparameters are in the same distance scale and therefore tuning of the parameters can be made intuitively over the center solution distribution $\varrho_\vartheta \sim (\mu, \sigma)$. Finally, the momentum parameter m_β is set to 0.9.

Table 4.1: Hyperparameters for CoRL.

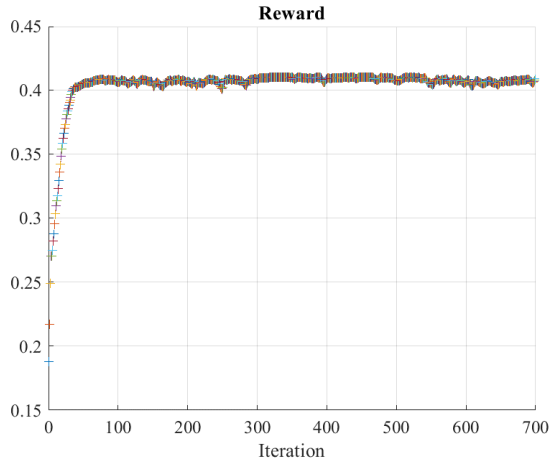
Hyperparameters	**Values**
Learning rate γ_μ	0.005
Max velocity v_{max}	0.01
Search radius q	30
Sigma learning rate γ_σ	0.05
Sigma max learning rate $\gamma_{\sigma_{max}}$	0.2
Momentum m_β	0.9

4.2 Optimal Pitching Trajectory using Online PGPE Algorithm

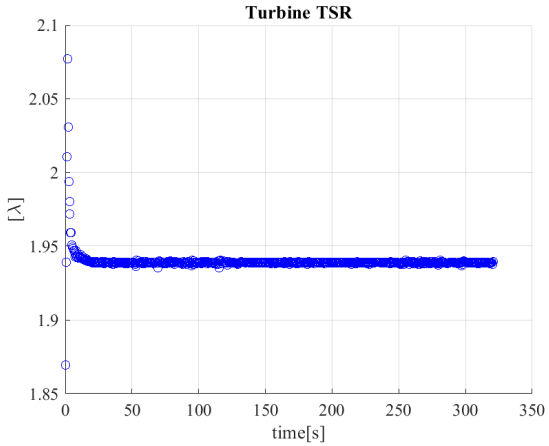
4.2.1 Constant Wind Condition

CoRL Wind Turbine Setup: To begin training, a constant wind profile of 6m/s was chosen as the starting point to verify whether the algorithm could indeed converge under a steady state environment. The indicated free stream wind velocity is those found on a typical windy day in a costal area or on top of a hill but is not considered too windy. As previously mentioned, the sine policy was first simulated with the hyperparameters shown in Table 2. The solution parameters, $\boldsymbol{\mu}$, were all initiated at zero and the distribution over the solution parameters was set at $\mathbf{r_{pgpe}} = v_{max} q(\pi/180)$ as done in [45]. Since the parameters are in the radian space, the conversion factor in the parentheses was added to keep the solution space within this space otherwise the search radius would be too large and have a hard time converging. We found a search radius of 30 to cover an adequate initial search space as opposed to the 15 recommended by [45].

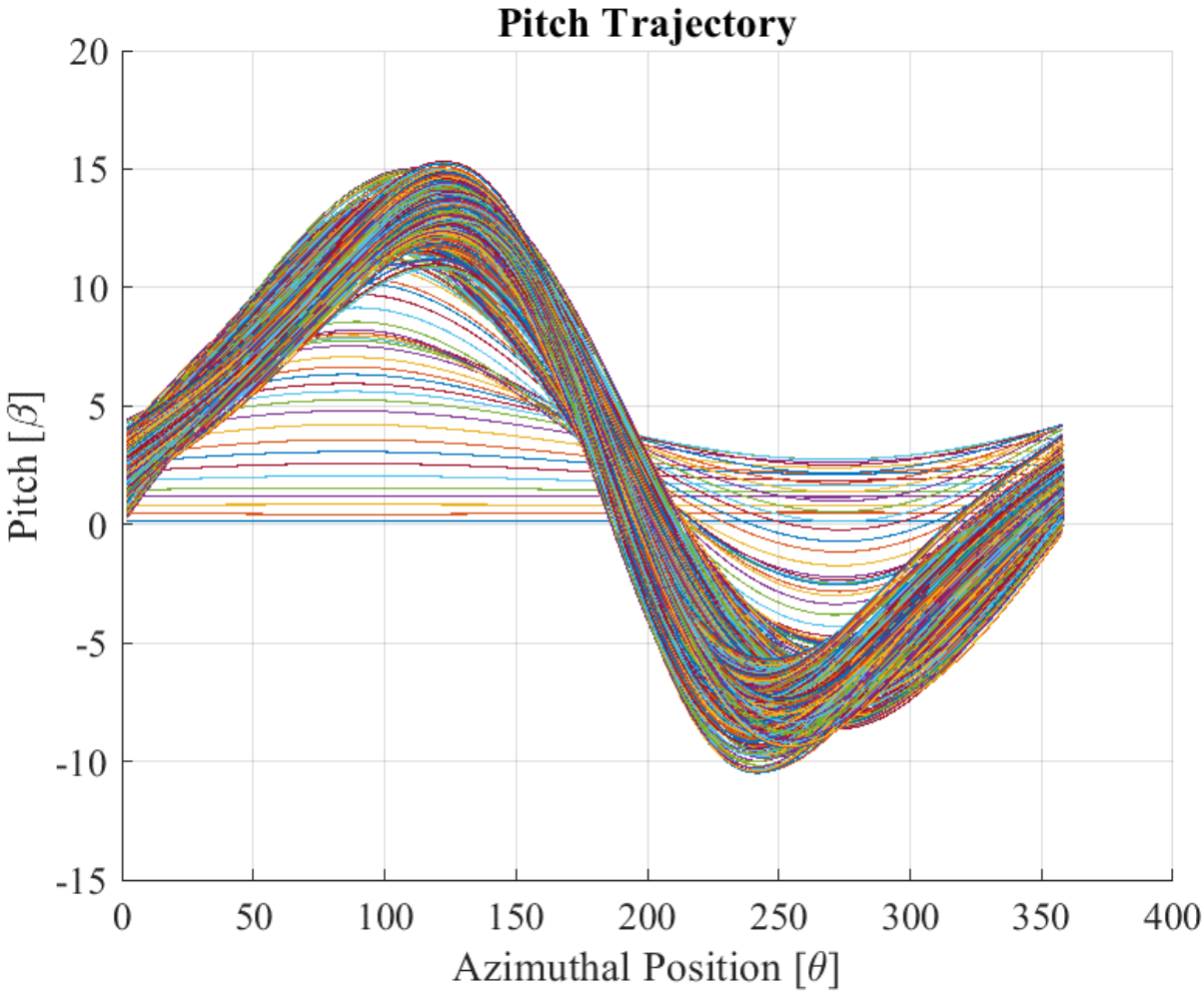
After tuning, the PID was able to stabilize the wind turbine speed of $\lambda_{opt} \approx 1.94$ under 10 seconds of operation taking the settling time of 1% the desired value. Although there is a relatively high overshoot, the controller is able to eventually correct the speed, applying corrections only after every turbine revolution. In reality, the PI controller



(a) Policy reward.



(b) PI speed regulation.



(c) Pitch evolution.

Figure 4.1: CoRL sine policy - Constant wind.

would be updated in real-time or every 5ms. Still, due to the limitations of accessing the dynamics in between revolutions using the DMST model, updates were given only after a turbine revolution. From the reward graph in Figure 4.1, it can be seen the algorithm converges right around 60 iterations which is within 5% of the highest C_P achieved by the MATLAB optimization and at 40 iterations surpassed a C_P of 0.400. In physical time the turbine would need approximately 15.9 seconds to reach an optimal pitch trajectory. After which the turbine mean reward is 0.4072 which is the average C_P of the turbine. In addition, the pitch evolution is seen in the figure to show the transition between every iteration which is mostly confined to the dense trajectory plots in 4.1c. Figure 4.1c gives a glimpse of the inner workings of the PGPE algorithm as it shows the range of distribution over the solution parameters. Lastly, the final pitch trajectory is shown in Figure 4.2. Given the relatively small optimal TSR, the turbine experiences large angles of attack during operation. Therefore, the pitch angle has to greatly compensate to avoid dynamic stalling, increasing pitch to 14 degrees in the frontal half and -7 degrees in the rear half of the turbine. Shown in this figure are two pitch trajectories; one of them is the center solutions once converged to the criteria mentioned above, and the other with the parameters that got the highest reward, C_P of 0.4101. Although the max C_P is less than the one achieved by the MATLAB optimization toolbox, it was expected as the pitching trajectory under the sinuosodial policy is confined to a sine curve.

Following the same procedure as with the sinusoidal policy, the spline function was implemented to formulate the pitch trajectory. As mentioned previously, the same hyper-parameters were used as well as PID parameters. All $\boldsymbol{\mu_{spline}}$ parameters were initialized at zero as before, and the same search radius used $\boldsymbol{r_{pgpe}}$. Like in the sine policy, the wind

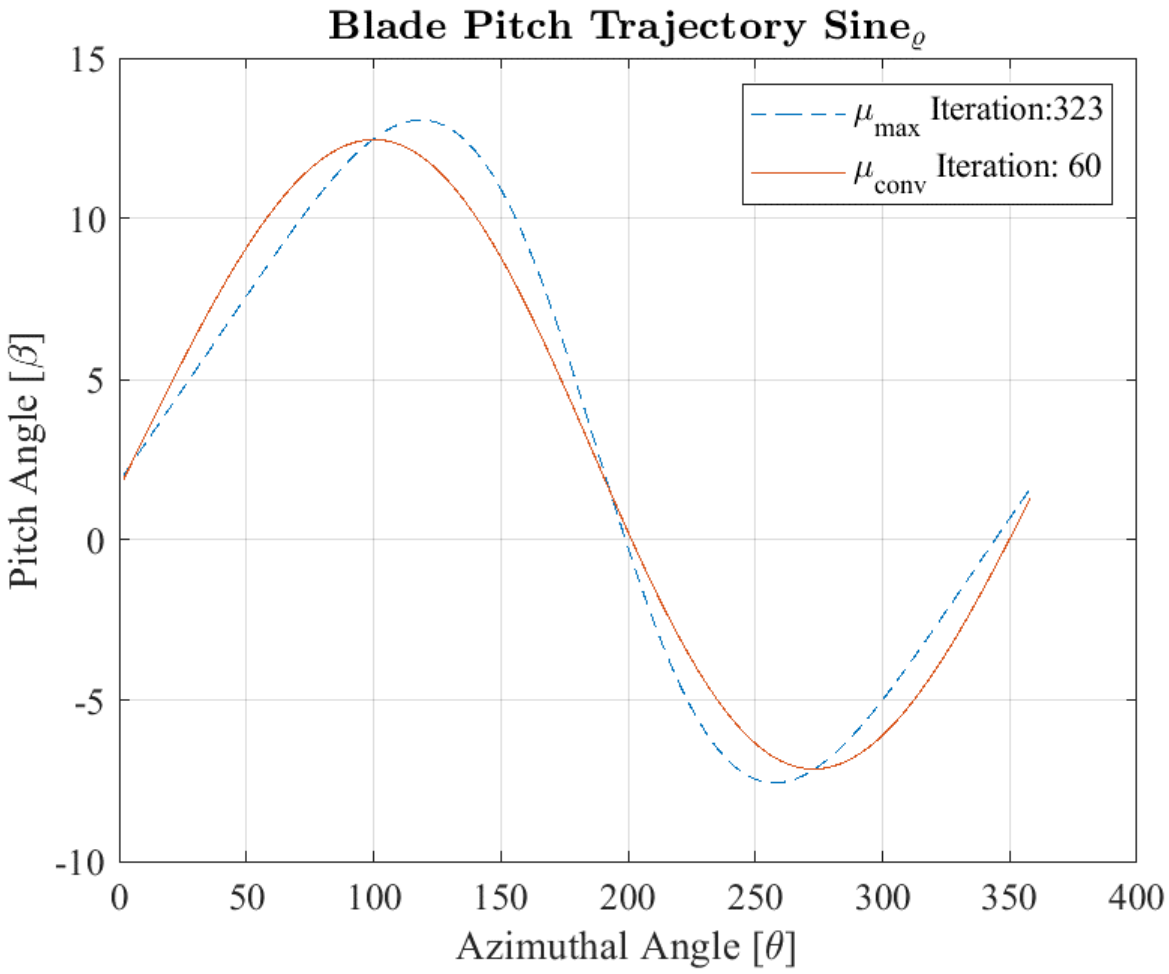


Figure 4.2: Optimal sinusoidal pitch trajectory - Constant wind.

turbine speed is stabilized within a few seconds and settled in under 10 seconds. It can be seen from the reward figure, Figure 4.3a, that the spline policy converges around the same time frames as the sine policy at right around 60 iterations which translates to 13.6 seconds. Although the spline policy converges to a slightly higher reward and ultimately outperforms the MATLAB optimization by receiving a max reward of $C_P = 0.4176$. The average C_P of the policy is 0.4152 after convergence. In Figure 4.4, we can see the spline policy reached the C_P output by the MATLAB optimization at iteration 57 of 0.4165, which is roughly 15 seconds of training. Furthermore, at iteration 281, corresponding

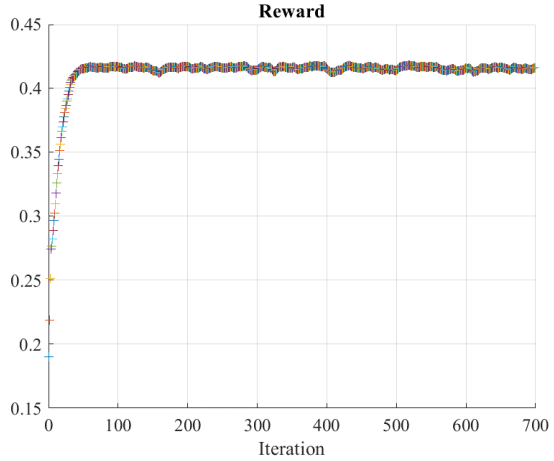
to 26 seconds the spline function receives its highest reward. The spline optimal pitch trajectory reaches a little over 15 degrees in the first half and -7 pitching degrees as seen in the previous policy.

Table 4.2: Hyperparameters for single operation WT.

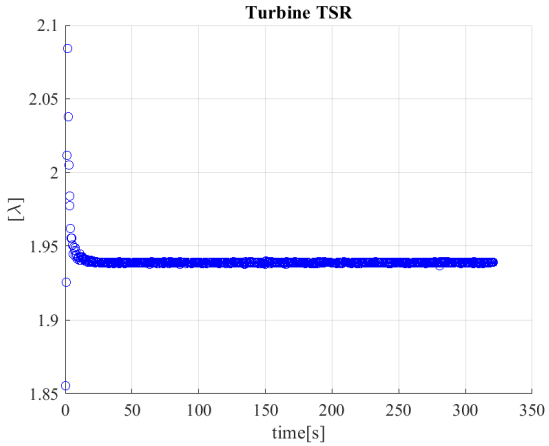
Hyperparameters	**Values**
Learning rate γ	0.0025
Max velocity v_{max}	0.005
Search radius q	30
Sigma learning rate γ_σ	0.05
Sigma max learning rate $\gamma_{\sigma_{max}}$	0.2
Momentum m_β	0.9

Single Operation Wind Turbine: To provide a comparison of the single vs dual wind turbine algorithm performance, a single turbine was simulated under the same environment as the dual wind turbine configuration. The goal here is to demonstrate the advantage of the symmetric sampling of the PGPE in a real environment to achieve a faster convergence speed. The single wind turbine algorithm used the non-symmetric PGPE version with the Clip Up enhancement. In addition, the sinusoidal policy was used due to its stability. All hyperparameters were used similarly to the dual turbine configuration except for the learning rate which was decreased to $\gamma_\mu = 0.0025$, as the algorithm had a hard time converging with the larger learning rate used in the dual wind turbine configuration. As such, one can immediately see that the single turbine algorithm is already at a disadvantage.

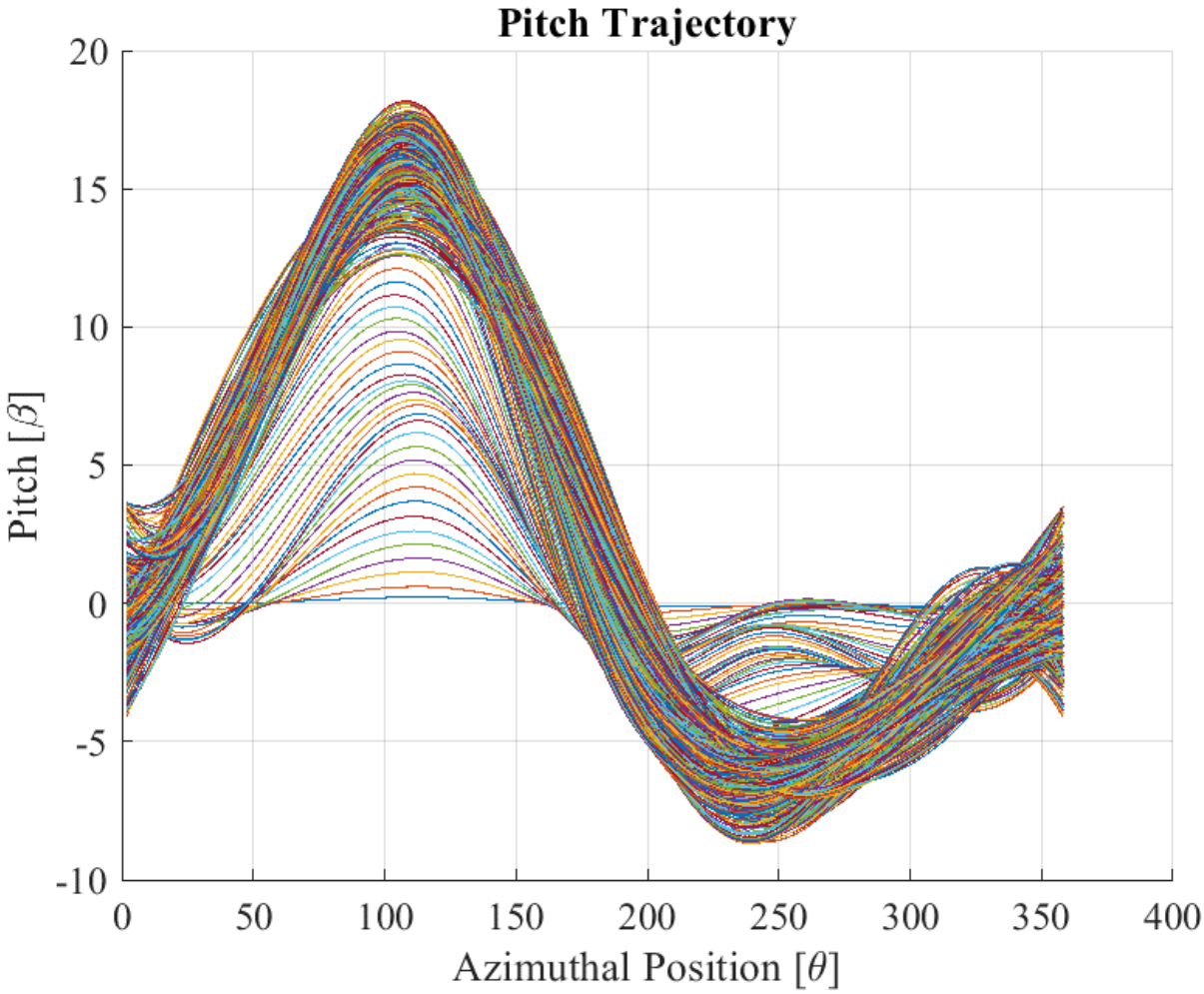
From the pitch trajectory evolution figure, Figure 4.5, the effects of the small learning rate can be seen by the dense pitch trajectory curves throughout the plot. The smaller step size means the gradient is updated slower and, therefore, the dense pitching curves



(a) Policy reward.



(b) PI speed regulation.



(c) Pitch evolution.

Figure 4.3: CoRL spline policy - Constant wind.

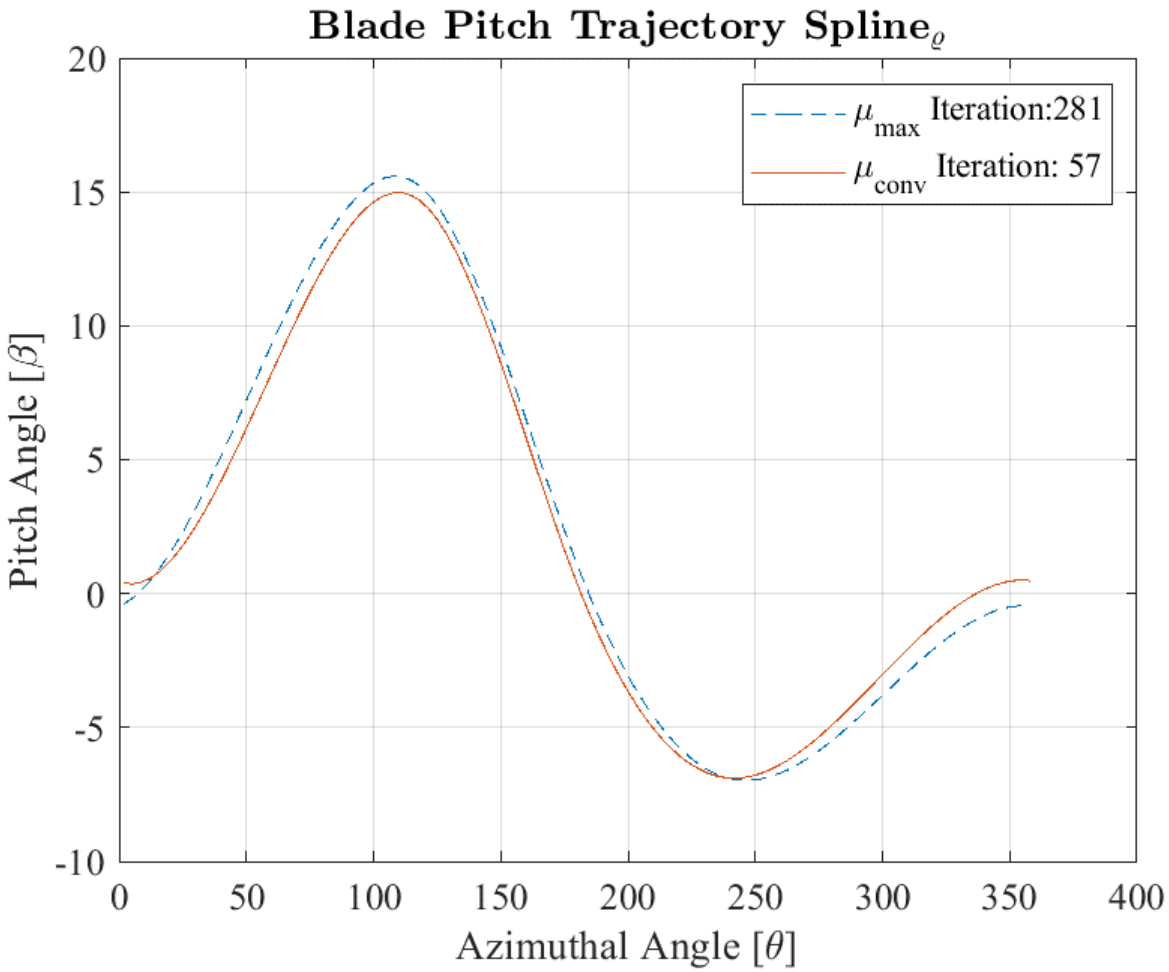
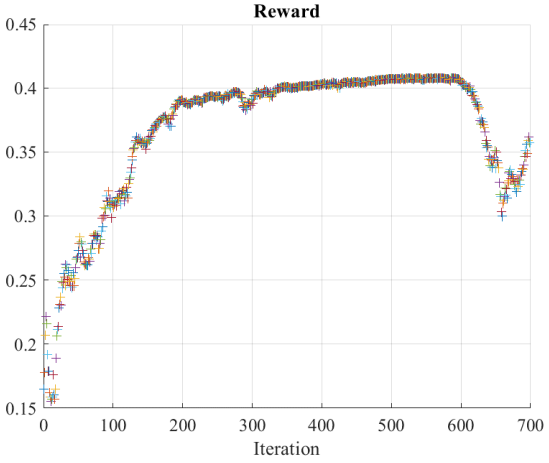
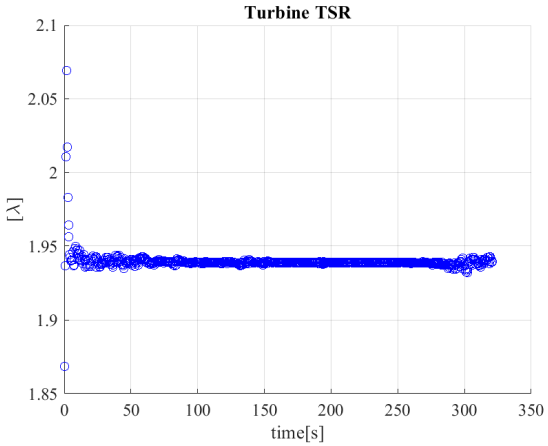


Figure 4.4: Optimal spline pitch trajectory- Constant wind.

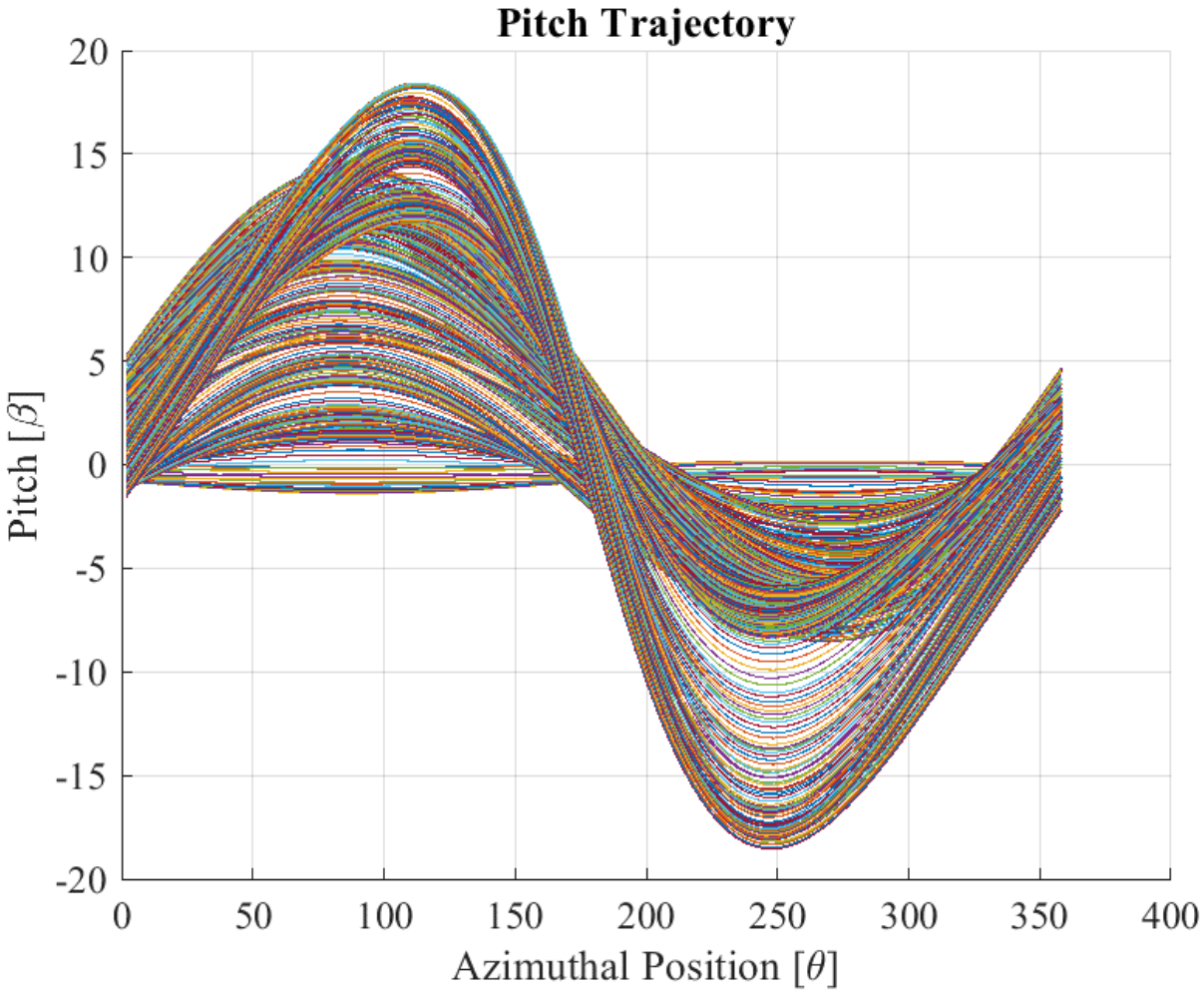
throughout the plot. In addition, the trajectories don't seem to converge on any particular range of trajectory as is obvious in the CoRL spline and sinusoidal policies. However, the base algorithm still received a relatively high C_P of 0.4083 but only achieved a C_P of 0.400 at 153 seconds much slower than the other simulations. In addition, the turbine speed is not as steady as the previous simulations, and therefore, the algorithm is unable to optimize fully under the optimal speed λ_{opt}. Because of the lack of convergence, the simulation was run again, this time with initiated values to improve convergence time. The sine wave was initiated with the parameters $\mu = [0, 0.1, 0.04, 0.36, 0.25, 0.05]$. In



(a) Policy reward.



(b) PI speed regulation.



(c) Pitch evolution.

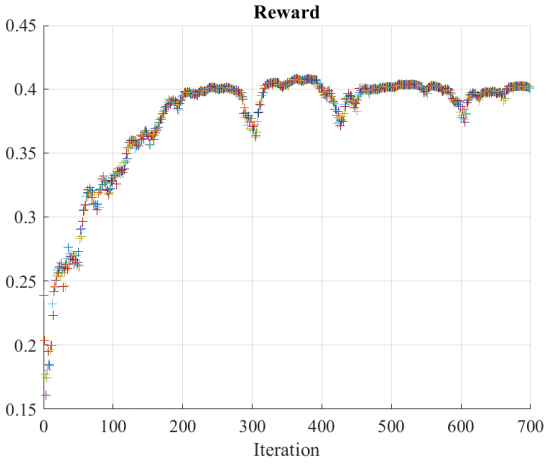
Figure 4.5: Single WT sinusoidal policy - Constant wind.

summary, the sine wave was not skewed, had very little flatness, an upper amplitude of 20 degrees, and lower amplitude of 15 degrees, and a small phase shift of 3 degrees.

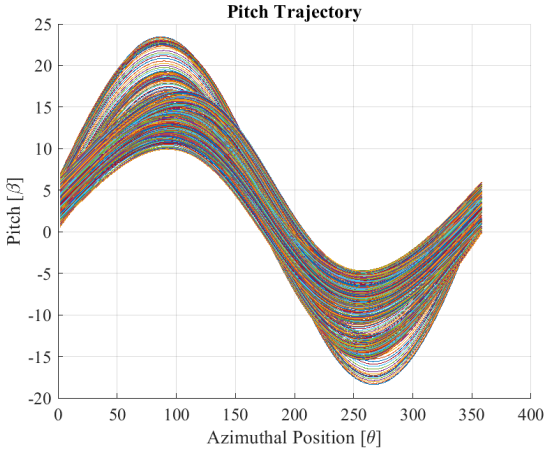
Figure 4.6 shows the updated run after initializing the center solutions with an initial guess. This time around, we see the pitch evolution is focused on a smaller range of solutions, and the solution begins to stabilize when referring to the reward graph. The highest reward received in the second run is a C_P of 0.4081 and a mean reward of 3965 after convergence. It also crossed the C_P threshold of 0.400 much faster at 107.7 seconds, 45 seconds faster than the first run. Still, we can see the performance of the dual turbine is far superior. Although the algorithm of the single and dual turbine configuration is not the same, they draw from the same underlying PGPE algorithm, and one can see the advantage of the symmetric sample over the single sample by converging faster and to a higher mean reward due to a better estimate of the gradient.

Figure 4.7 shows the best pitching trajectories for all the cases run under a constant wind environment. The spline function had the highest reward and was the closest to our ground truth pitch trajectory provided by the MATLAB optimization toolbox. The sinusoidal pitching policies are shifted up and have lower pitching amplitude in the front half than those in the spline policies. Yet all pitching trajectories fall under a similar area, and thus, moving forward, we can conclude that PGPE's solutions are valid candidates.

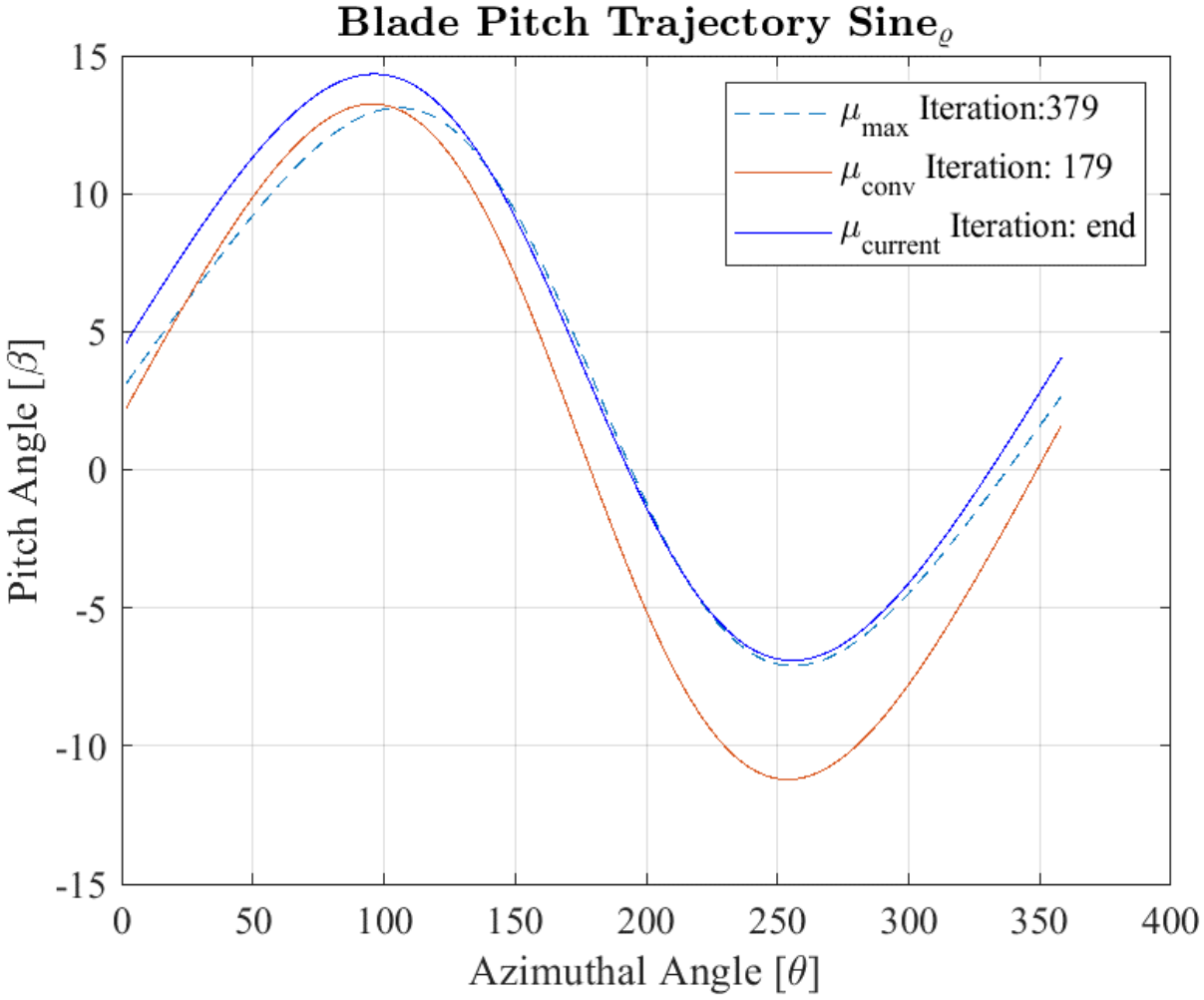
In Figure 4.8 we can see the trajectory evolution from each of the policies. The CoRL policies are initiated at zero and find the range of possible pitch trajectories fairly quickly. From the subfigure 4.8c we can see the single turbine policy with the solutions initiated at zero has a difficult time converging on a range of possible solutions whereas the same policy with initiated center solutions converges on what seem to be two sets of pitch



(a) Updated Reward.



(b) Pitch evolution.



(c) Optimal Trajectory Comparison.

Figure 4.6: Single WT sinusoidal policy - Updated initialization.

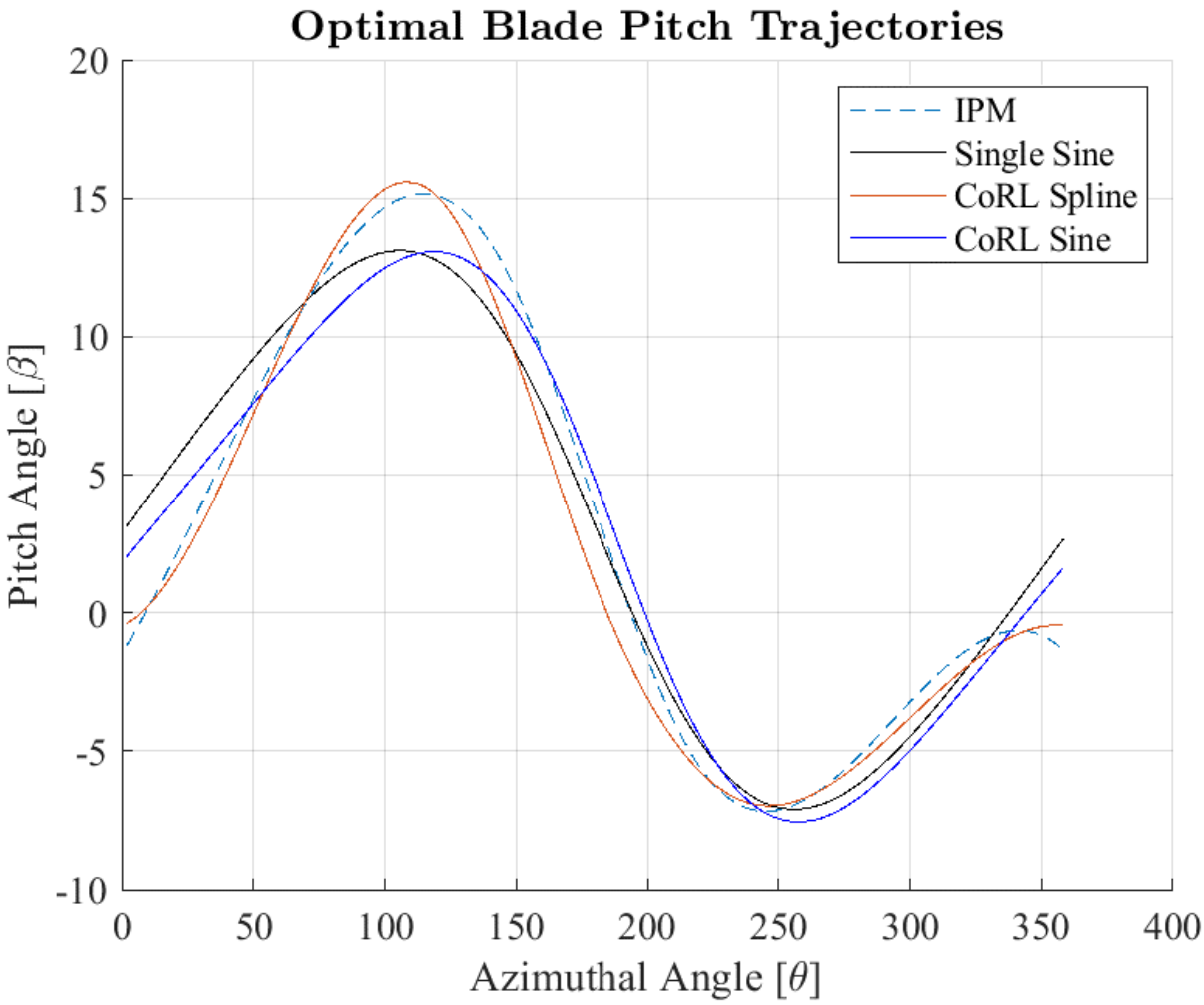
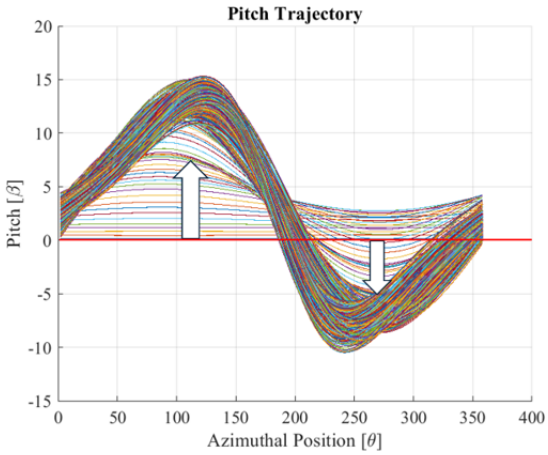


Figure 4.7: Optimal pitch trajectories comparison for constant wind.

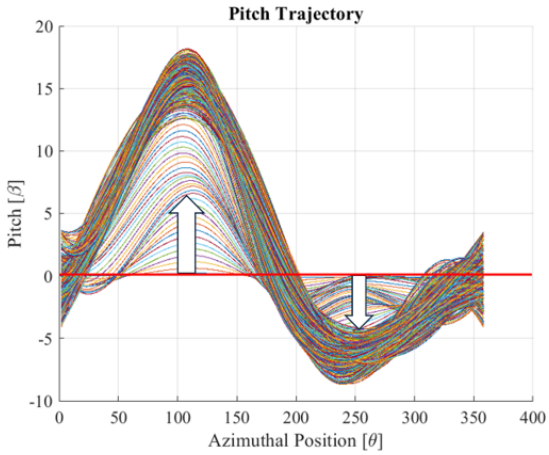
trajectory ranges and therefore the constant dip of reward in Figure 4.6a.

4.2.2 Variable Wind Condition using a Ramp Wind Function

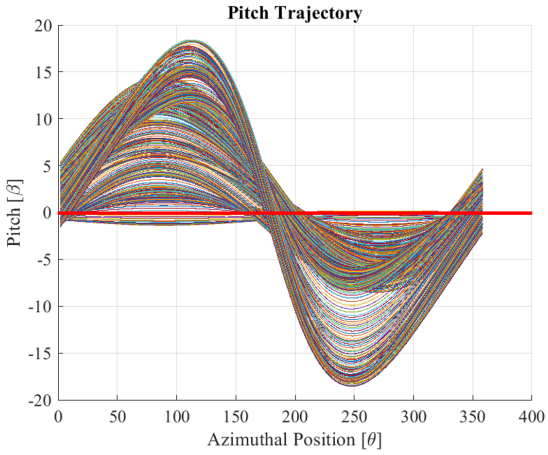
A wind ramp function was used to test the stability of the algorithm to see the effects the varying wind conditions would have on the algorithm. The wind function sampled had a windspeed frequency update of 10 over the span of the simulated time and was generated to change the slope of the wind after each update randomly. The windspeed graph is shown in Figure 4.9. This wind profile is used for all three simulation cases using the sinusoidal policy, spline policy and lastly the single wind turbine operation.



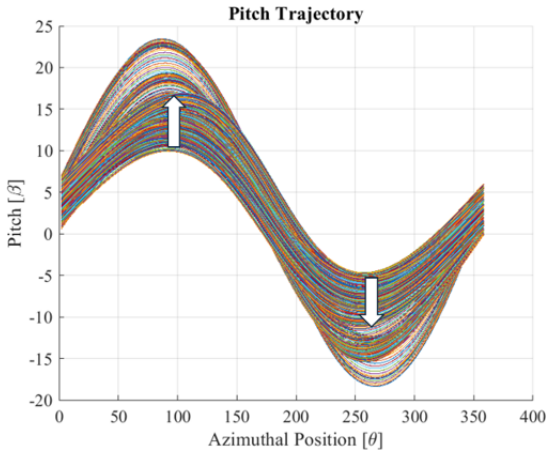
(a) CoRL sine policy pitch evolution.



(b) CoRL spline policy pitch evolution.



(c) Single WT sine policy pitch evolution with $\boldsymbol{\mu} = 0$.



(d) Single WT sine policy pitch evolution with $\boldsymbol{\mu} = init$.

Figure 4.8: Pitch evolution for constant wind environment.

Table 4.3: Result summary of constant wind case.

Performance Parameters	Single WT Initialized	Sine Policy	Spline Policy
Mean reward	0.3965	0.4072	0.4152
Max reward	0.4081	0.4101	0.4176
Convergence time [s], $r = 0.4$	107.7	15.9	13.7
Converged	Yes	Yes	Yes

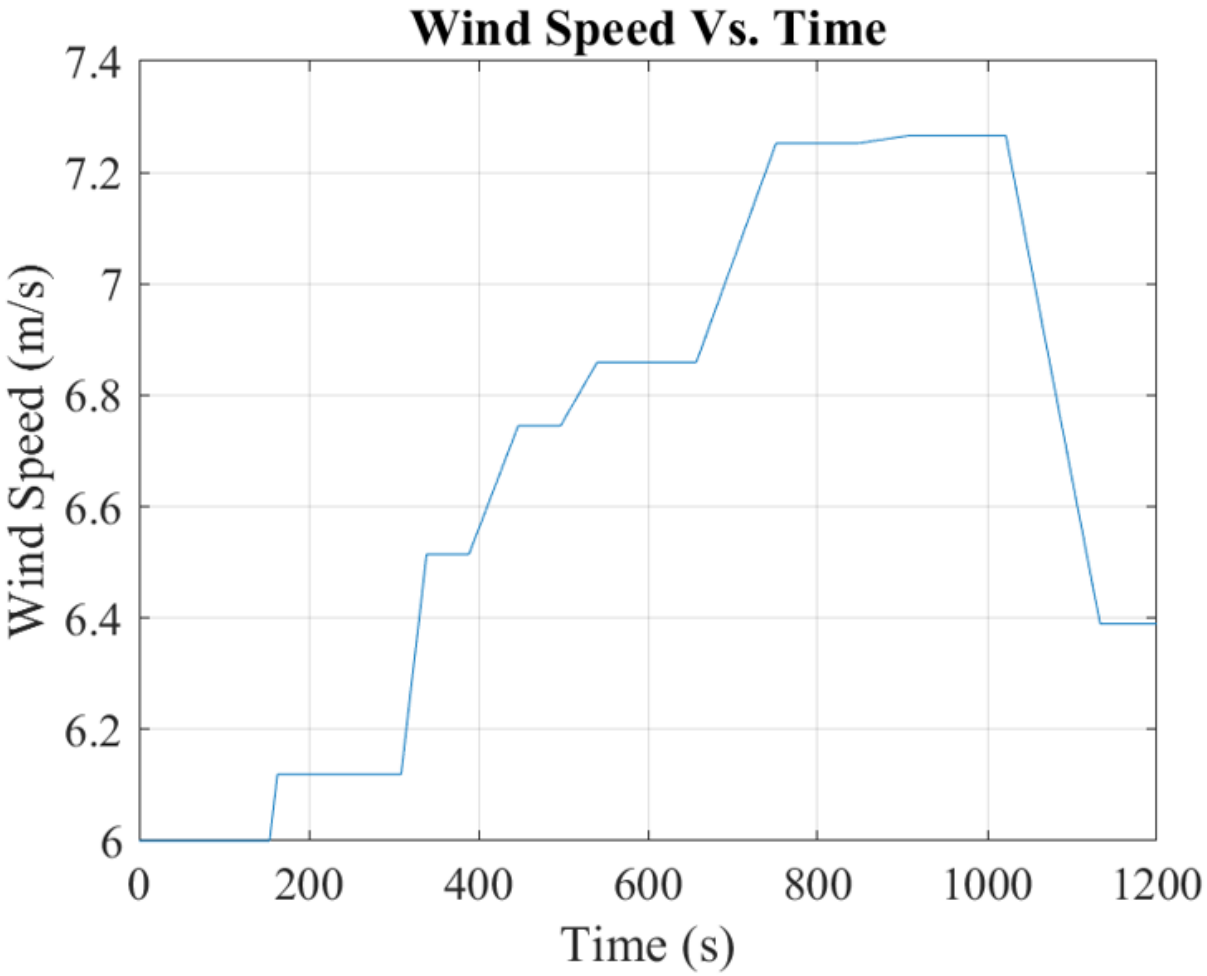
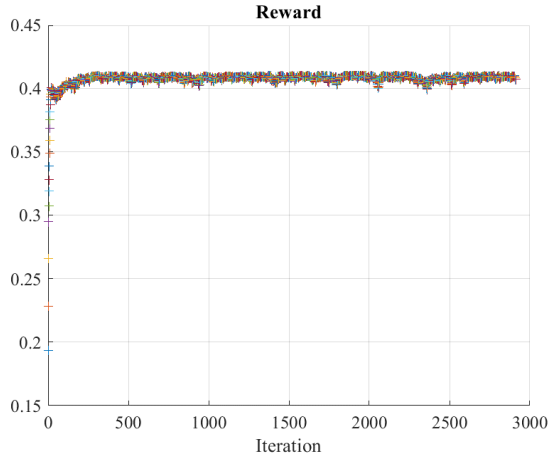


Figure 4.9: Variable wind profile - Ramp function.

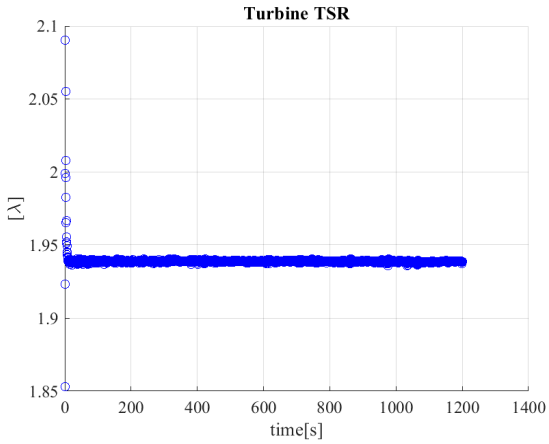
CoRL Wind Turbine Setup: For the sinusoidal and spline functions the same hyperparameters were used as in the first case in addition the same PID parameters were also used. All $\boldsymbol{\mu}$ center solutions were initialized at zero and $\boldsymbol{\sigma} = \boldsymbol{r_{pgpe}}$ as done in the previous constant wind simulations. Again, the first policy simulated was the sinusoidal policy. Seen in Figure 4.10 is the reward, final pitch trajectory, and lastly the TSR. Since the wind profile is dynamic, the TSR has more oscillations as the PID controller tries to maintain optimal speed control. Consequently, this is also reflected in the reward plot as there are many more oscillations in the reward as well. The PID had a similar

initial performance settling the wind speed in under 10 seconds. The mean reward of the sinusoidal policy after converging was 0.4081 which is comparable to the constant wind simulations. Interestingly, the parameters converged in roughly the same time as before, reaching the C_P threshold of 0.4 in roughly 10 seconds, 5 seconds faster than in the constant wind environment. This can be attributed to the TSR varying a little more than in the constant wind environment leading to differences in the C_P performances as $C_P(\lambda, \beta)$. The final pitch trajectory was observed to be similar when compared to the constant wind profile velocity. The curve has a positive offset at 3 degrees and upper amplitude peak is at 14 degrees and the lower amplitude is at roughly -9 degrees.

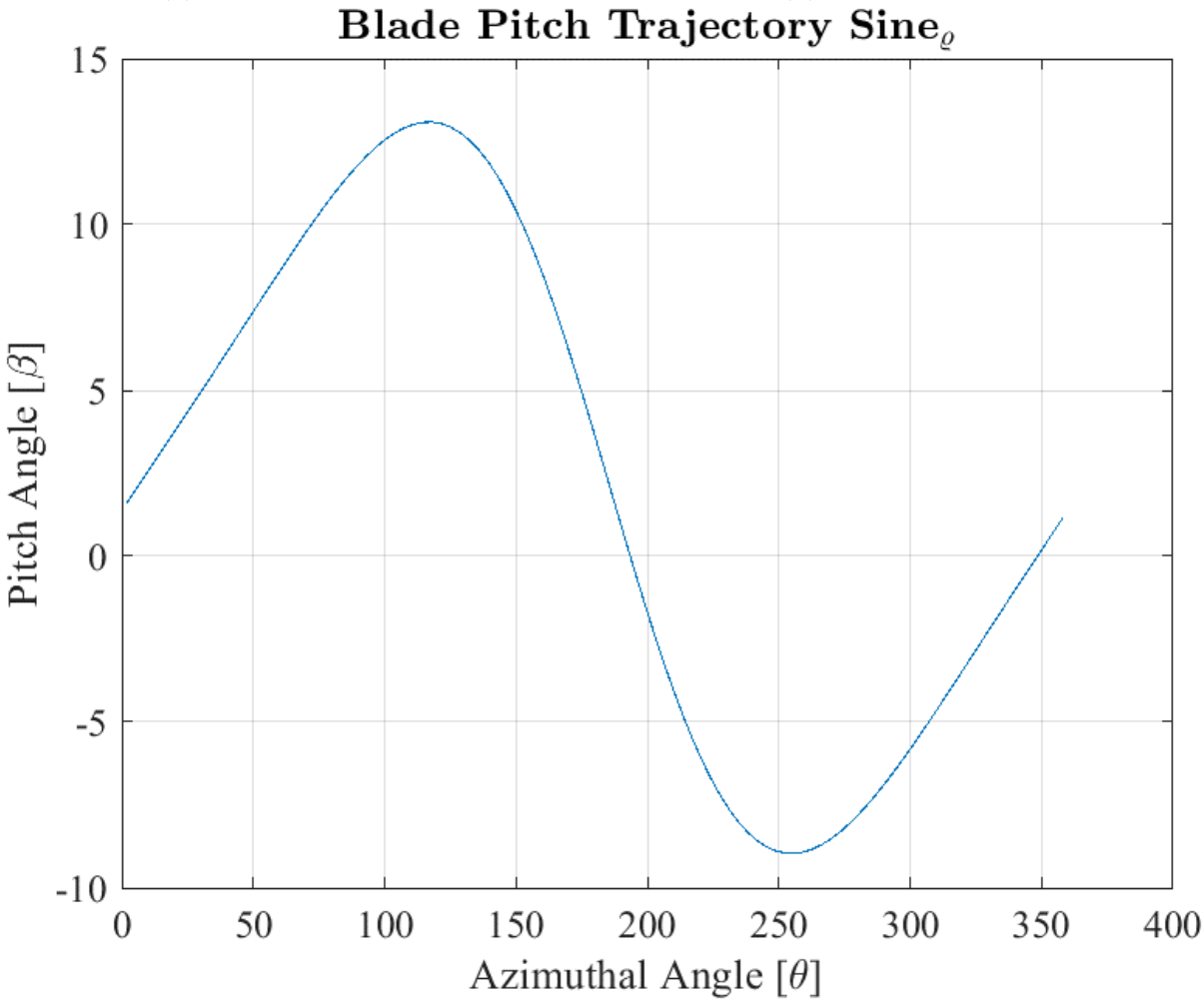
Similarly, the same initialization technique was used for the spline function and the same hyperparameters as for the sine policy function. The PID controller was able to stabilize the turbine speed in under 10 seconds, and the solution parameters converged at a slower but comparable rate than the sinusoidal policy reaching the C_P threshold of 0.400 at 18.2 seconds compared to the 10 seconds of the sinusoidal policy and slower than in the constant wind environment which is expected. The mean reward after convergence comes out to 0.4159 which is higher than the constant wind pitch trajectory and the sinusoidal policy. Shown in Figure 4.11 is the optimal pitch trajectory. We can start seeing the problem with the spline function as although it is continuous it is not smooth from one trajectory to the other. There must be another constraint placed on the spline policy such that the end points meet in a smooth differentiable curve. Nevertheless, this pitching curve is one of many optimal pitching trajectories sampled from the distribution $\mathcal{N} \sim (\mathbf{0}, \boldsymbol{\sigma})$ of the current policy and is the current $\boldsymbol{\mu}$ center solutions at the end of the simulation.



(a) Policy reward.

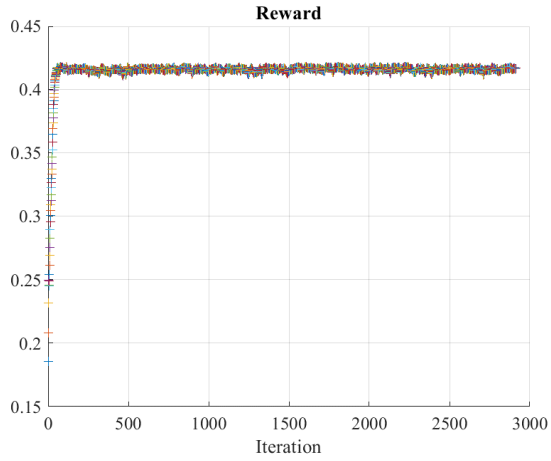


(b) PI speed regulation.

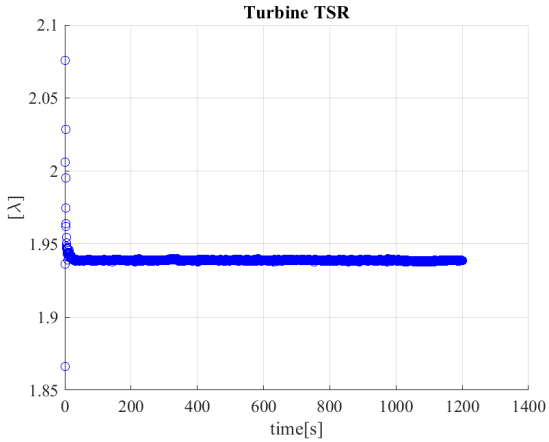


(c) Optimal Pitch.

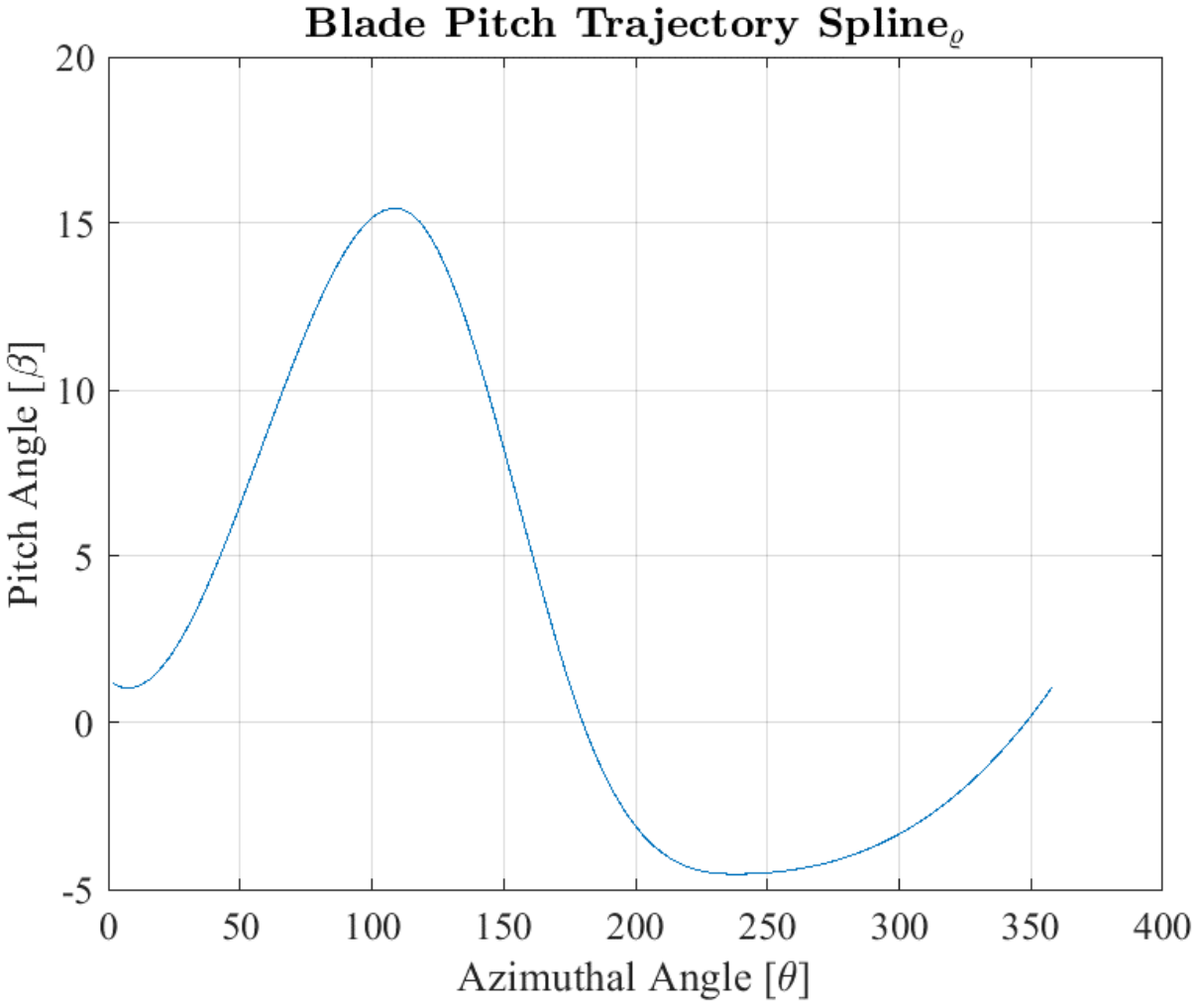
Figure 4.10: CoRL sinusoidal policy – Varying wind.



(a) Policy reward.



(b) PI speed regulation.



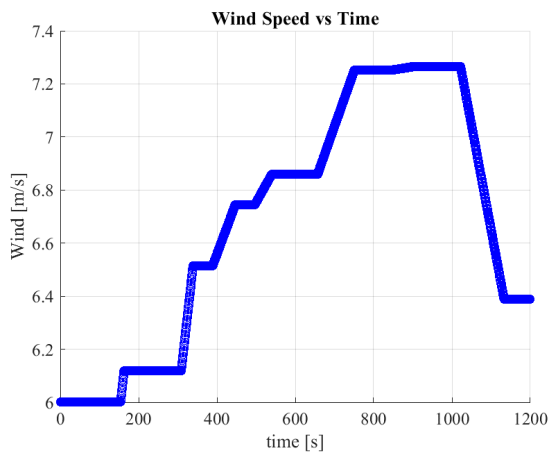
(c) Optimal Pitch.

Figure 4.11: CoRL spline policy – Varying wind.

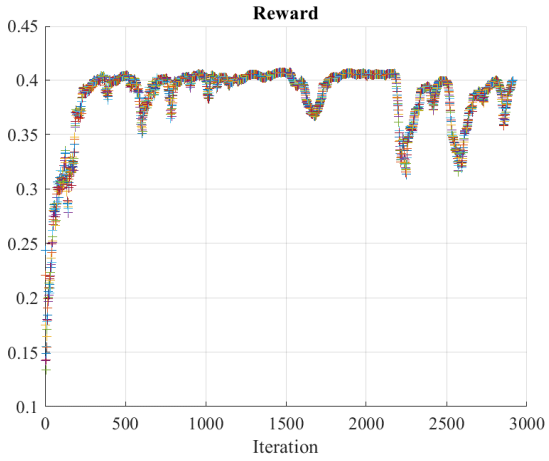
Table 4.4: Result summary of varying wind case.

Performance Parameters	Single WT Initialized	Sine Policy	Spline Policy
Mean reward	0.3925	0.4081	0.4159
Max reward	0.4071	0.4103	0.4179
Convergence time [s], $r = 0.4$	138.0	9.1	18.2
Converged	Yes	Yes	Yes

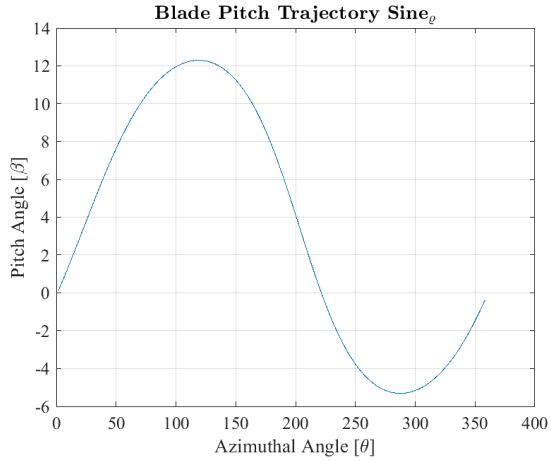
Single Wind Turbine Operation: The same initialization was followed, using the same approach as with the constant wind profile for the single wind turbine operation. The solution parameters were initialized at an initial guess as before. The PID controller settled the turbine wind speed in roughly 5 seconds but was much more noisy when compared to the other simulations which reduces the maximum C_P achievable if not operating in the optimal speed. One reason for this difficulty in obtaining the optimal speed is the inability of the algorithm to converge to an optimal range of solutions, giving pitching trajectories that cause unstable forces that slow or speed the turbine at a faster rate than can be managed by the PI controller. That is why we can see the reward graph is correlated to the turbine speed graph. The solution parameters in this study crossed the C_P threshold in roughly 138 seconds or 320 iterations. The highest reward is a respectable 0.4081, and the mean reward C_P of 0.4071. Due to the time convergence and instability in the reward graph, we can see the single wind turbine operation is not up to par with the dual wind turbine configuration although capable of generating decent pitch trajectories. Therefore, in the next simulation case, the single-operation turbine configuration will not be used. See Figure 4.12 for results.



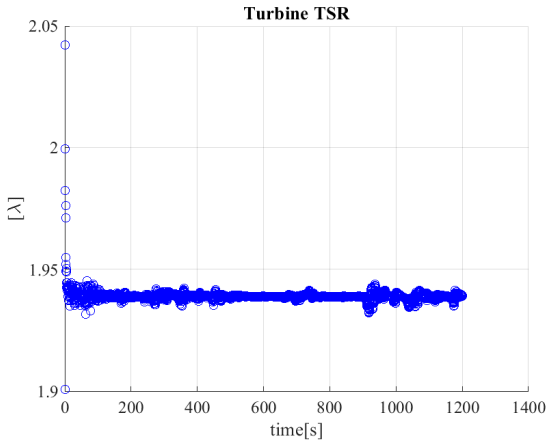
(a) Wind speed.



(b) Policy reward.



(c) Optimal Pitch.



(d) PI speed regulation.

Figure 4.12: Single WT sinusoidal policy – Varying wind.

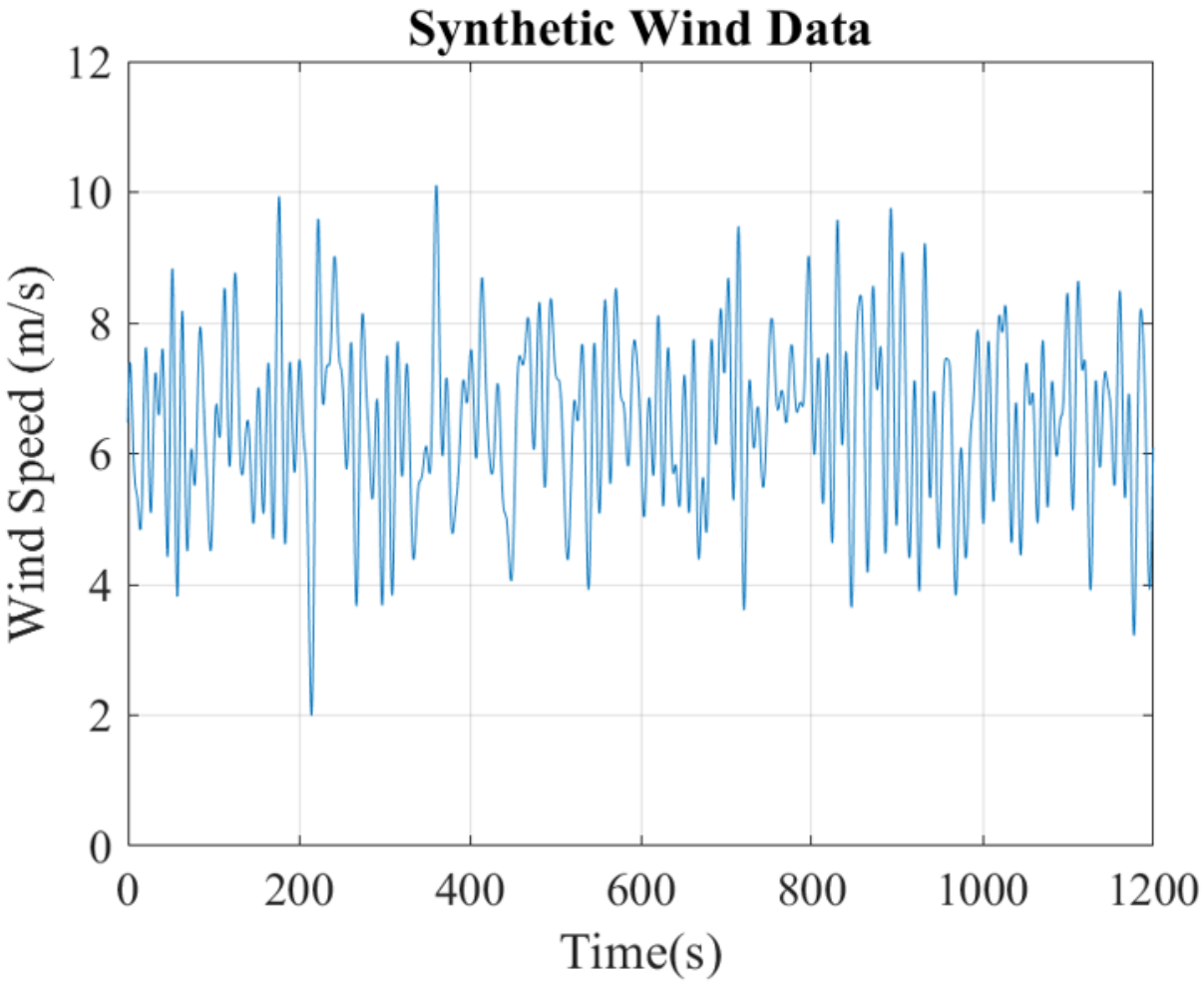


Figure 4.13: Synthetic random wind.

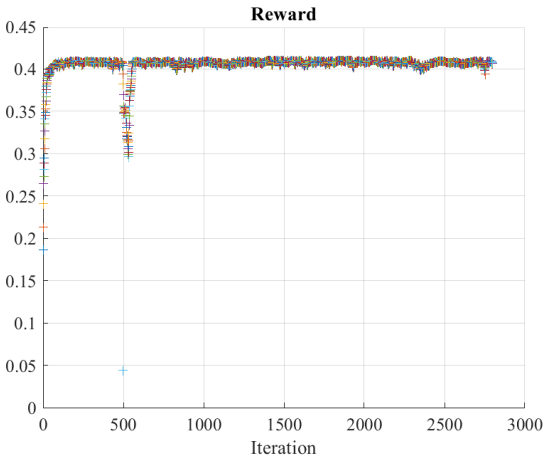
4.2.3 Synthetic Random Wind

In the final case we tested the algorithm performance with a synthetic wind profile. The average speed of the wind profile is 6.5 m/s with the simulated total time is 20 minutes. The wind profile frequency update is set at 0.15 Hz shown in Figure 4.13.

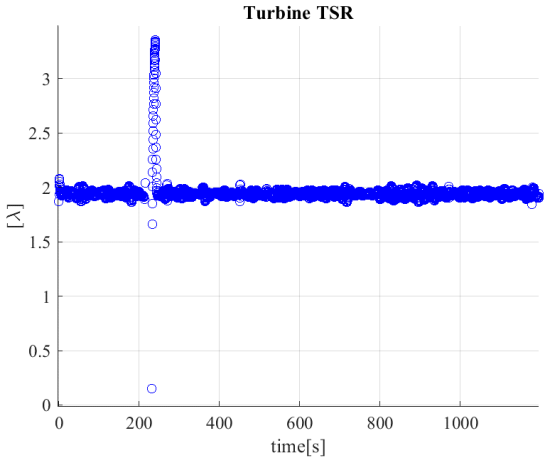
CoRL Wind Turbine Operation: The sinusoidal policy was first tested vs the synthetic wind data. Although the sine policy was the more stable of the two it failed to learn adequately within the given time frame. The deviation step size $\gamma_{\sigma max}$ had to be increased from 0.2 to 0.5 to stabilize the distribution update of the parameters while the other

hyperparameters were left the same. The step size was increased to allow the distribution over the center solution to increase and, therefore, be able to adapt faster given the rapid change in TSR. From the turbine wind speed graph, Figure 4.14, we can immediately see the effects of the noisy wind speed as the TSR greatly fluctuates at one point. However, the PID controller is able to maintain the desired optimal TSR value within $\pm 4\%$ for most of the simulated time, yet we can see the two peaks at two extremes where the turbine speed almost reaches zero and the other where the turbine exceeds a TSR of 3. There was a major drop in TSR when the wind speed dropped to 2 m/s towards the beginning of the simulation. At this point, the TSR dropped to almost zero. Here we see a massive performance drop as the turbine cannot produce power due to the lack of wind and PID response to remove the turbine load, producing almost no power, abrupt shift in dynamics leading to poor performance of deterministic control, and consequently receiving a small reward. For the drop in TSR, we can attribute this to low wind speed and, therefore, not enough aerodynamic torque to sustain the given load by the PID. In addition, because this range in TSR has not been experienced by the algorithm, the deterministic control does not have the proper parameters to overcome this dip and, therefore, the loss of C_P and, as later will be discussed, the highly unsymmetrical samples from both turbines. Yet the PID controller is able to return the TSR to its optimal TSR and the PGPE algorithm is also able to return to its optimal pitch trajectory without diverging. The solution parameters crossed the C_P threshold of 0.400 in 20.4 seconds, the slowest yet but not surprising, and maintained an average of 0.4066 after convergence.

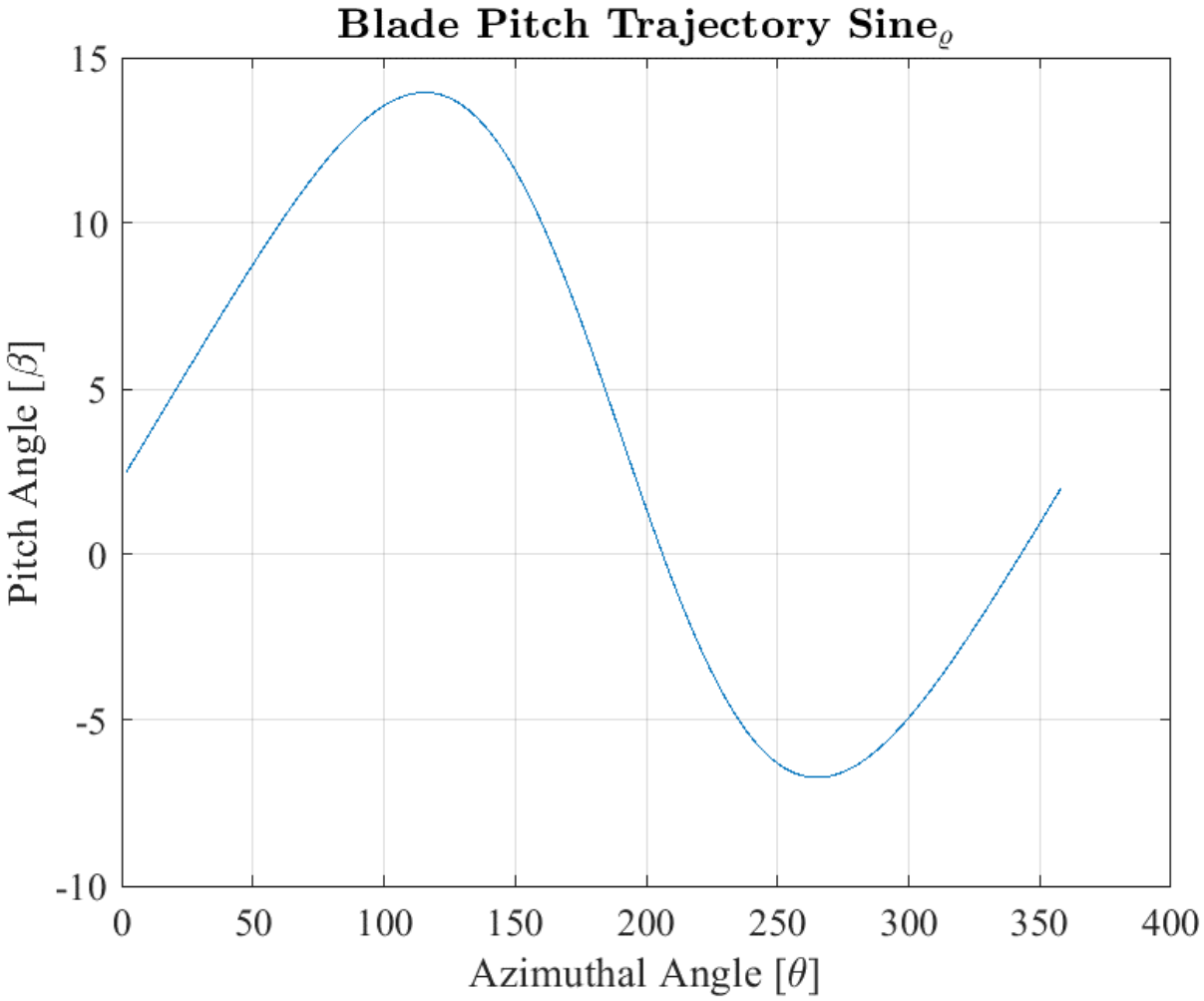
For the spline policy, the same hyperparameters were used as those used for the sinusoidal policy adjusting the max learning rate of the deviation parameter. As can



(a) Policy reward.



(b) PI speed regulation.



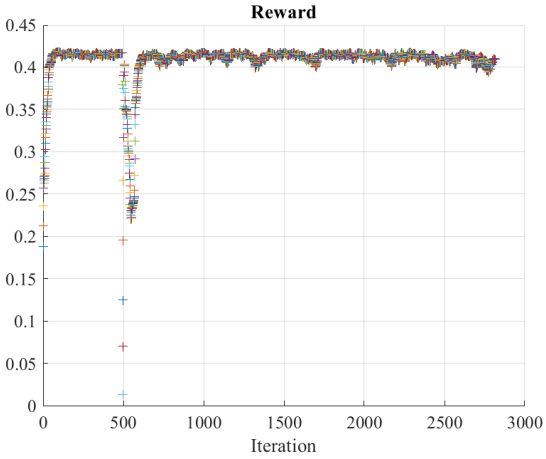
(c) Optimal Pitch.

Figure 4.14: CoRL sine policy – Synthetic wind.

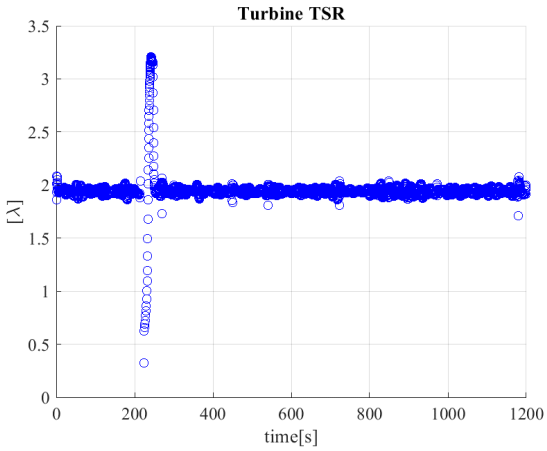
Table 4.5: Result summary of synthetic wind case.

Performance Parameters	Sine Policy	Spline Policy
Mean reward	0.4066	0.4159
Max reward	0.4103	0.4179
Convergence time [s], $r = 0.4$	20.4	18.5
Converged	Yes	Yes

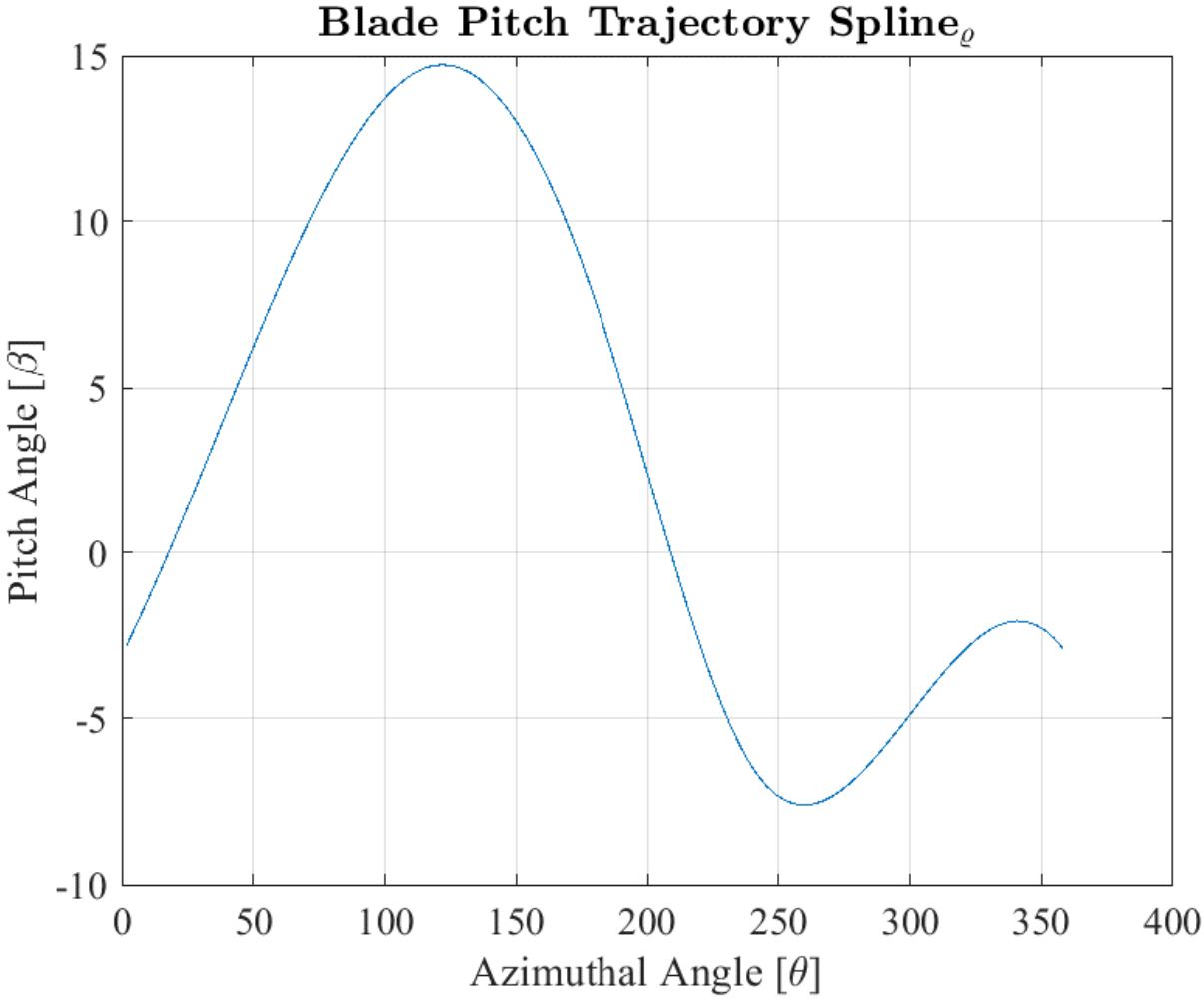
be seen in Figure 4.15 the PID controller was able to keep the turbine speed constant throughout the simulated time but had its trouble as in the previous simulation. The PID controller was able to bring the speed to a steady state similar to the previous simulations under 10 seconds with the exception of when the wind speed dropped to a low 2 m/s. But nevertheless, the controller was able to compensate by removing the turbine load and allowing the turbine's optimal speed to return. From the reward graph, we can see a major reward dip in the same section as where the wind speed drops below 2 m/s. The reward loss in the spline policy is much greater than seen in that of the sinusoidal policy. As mentioned earlier the spline function can take various forms and therefore not as stable as the other policy. In this case, it was still able to return to its nominal state without becoming unstable and failing, which shows the robustness of the PGPE algorithm and the CoRL setup. The average reward after convergence comes at 0.4088, lower than the previous simulations but nevertheless respectable. Looking at the pitching trajectory after the simulation, we can see it is similar to the optimized pitch trajectory output by the MATLAB optimization algorithm and under the constant wind environment.



(a) Policy reward.



(b) PI speed regulation.



(c) Optimal Pitch.

Figure 4.15: CoRL Spline policy – Synthetic wind.

4.3 Dual Turbine Performance Metric

Convergence of Parameters

To determine the convergence of the parameters, $\boldsymbol{\mu}$ we mainly used the reward plot as an indicator and used the individual parameter evolution throughout the simulation as well. The PGPE algorithm updates the solution parameter, μ, in the direction of the gradient and updates the search distribution, σ, of each parameter as well. As the parameters begin to converge the search distribution stabilizes within a range of possible solutions. In some cases, the search distribution is so small that it may seem as though the parameter converges on a single value. Figure 4.16 shows the optimization of all six parameters of the sinusoidal policy for the synthetic wind data. We can see all parameters are initialized at zero and from there, they begin to move to their final range of values slowly. The search distribution radius is initialized as $\boldsymbol{r_{pgpe}} = q * vel_{max}\pi/180 = \mathbf{0.0052}$. At the end of the simulation, the search distribution is $\sigma = [0.0035, 0.0052, 0.0020, 0.0032, 0.0094, 0.0022] = [a, b, d, s_1, s_2, \zeta]$. From the values and the graphs of Figure 4.16, we can see the center solutions settled in a solution range rather than a value. What we can deduce from these values is the offset shift factor d and s_1 are not as greatly affected by the varying environment whereas the lower half scaling factor s_2 is influenced a little more by the environment and therefore does not converge on a single parameter but rather a range of values depending on the combination of the environment and the other parameters. Yet all the center solutions, or parameters $\boldsymbol{\mu}$, are in the same order of magnitude, and therefore, all parameters are influential to the optimal pitch trajectory.

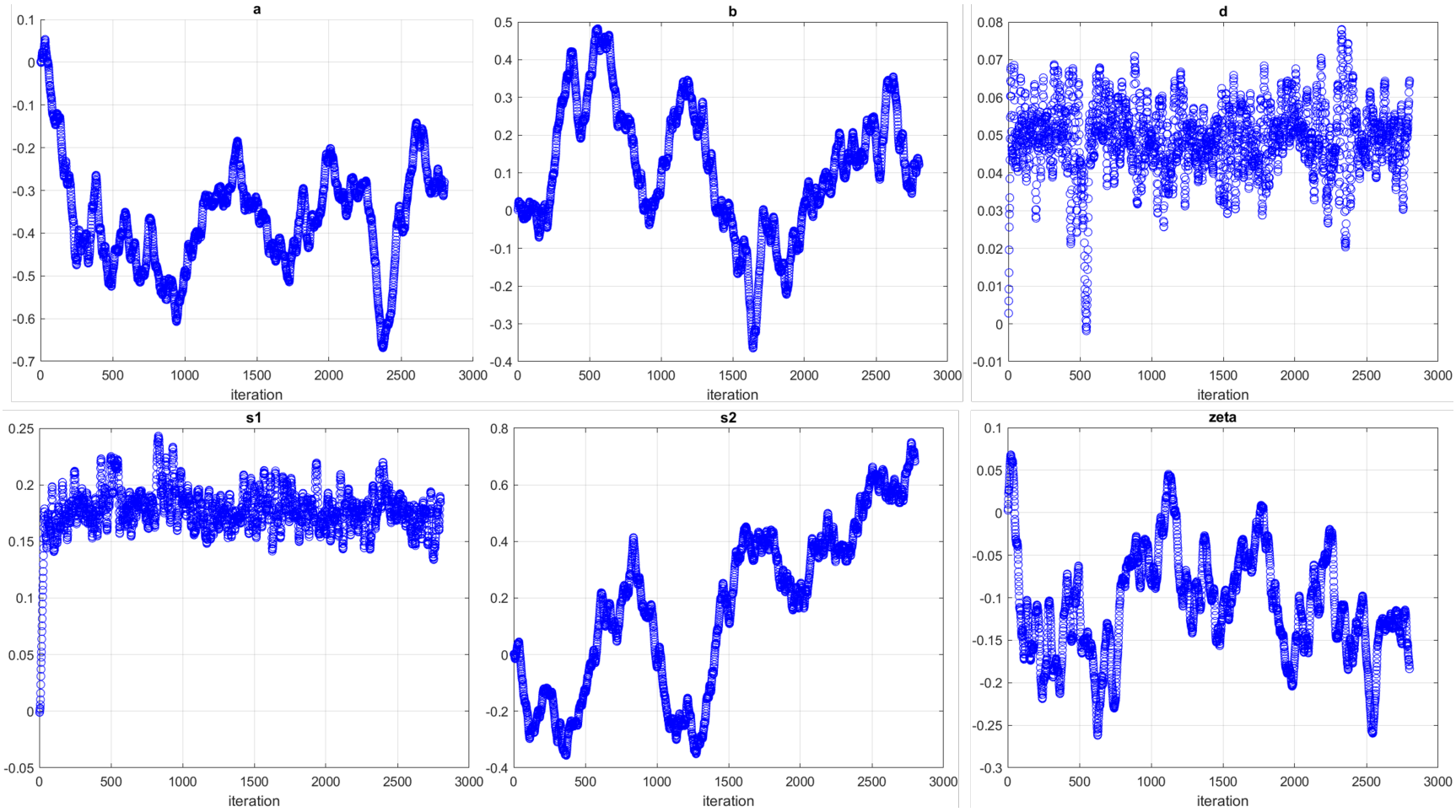


Figure 4.16: Sinusoidal policy parameter convergence.

Dual Turbine Environmental Operation and Algorithm Robustness

For the symmetric PGPE algorithm to work properly, both turbines must operate with the same environmental input. This means both turbines had to be operating at the same speed and under the same wind conditions or cause the PGPE algorithm to become unstable. The reality is that getting an in-sync wind turbine operation is challenging as there could be wind variations from one location to the next and/or speed controllers that can't perfectly sync the turbine speeds. In our simulations these imperfections were taken into account as the turbine dynamics were generated based on the policy actions. In this case, two turbines took two mirroring actions where noise was added to one turbine and subtracted from the other. This means one turbine could take a pitching trajectory that dramatically affects the dynamics of the turbine decreasing the turbine speeds by inducing more aerodynamic drag than lift. Whereas the other turbine could have chosen a more optimal pitching trajectory keeping the turbine speed stable. Given the erratic action of the first turbine, the speed controller cannot react fast enough, and the environmental input is now a semi-asymmetric input. From the Figure 4.17 we can see the TSRs of both turbines during the simulation. If we take a closer look at some of the peaks, we can see the two turbines operate at a different TSR suggesting either a difference in wind speed or a difference in turbine angular speed. But as can be seen, the dual turbine operation is, for the most part, in sync, operating within 4% of each other, and regardless of the discrepancies in turbine operation, the algorithm is still able to find the optimal distribution of parameters and converge within a reasonable time. This shows the algorithm's stability even with non-ideal symmetric samples. Taking a closer look at

the large gap at iteration 500 we can see the large difference in turbine speeds causing a sharp decrease in rewards as seen in Figure 4.14. In this scenario, the turbines were at different speeds for nearly 20 seconds, reducing the reward but ultimately overcoming the reward loss with both the help of the PI controller and the larger deviation learning rate allowing a faster recovery. If we take a closer look at Figure 4.17 we can see that Turbine 2 is much more stable than Turbine 1 in which case the TSR for Turbine 1 did not fluctuate as much as the other and is also reflected in the reward, Figure 4.19. Because both turbines share the same policy and the solution is updated taking into account both rewards, where the better of the two is favored positively, the stability of Turbine 2 helps bring back Turbine 1 to a stable region. In addition, since the PI controller is unable to adjust the speed of the turbines when the speed is lower than the optimal speed, then it is the pitching control which allows the turbine to return back to its optimal speed. If we recall the performance of the VAWT is a function of both TSR and pitching, $C_P(\lambda, \beta)$.

Periodic and Continuous Pitching Trajectory

Given that the wind turbine's fundamental operation is periodic, it makes sense to use a continuous periodic pitch trajectory. From the two policies used the spline policy was able to generate a higher average reward than the sinusoidal policy. Although the spline pitch trajectory was able to produce higher rewards, in practice the pitch trajectory is not one that would be feasible given the abrupt pitching action after each revolution. Plotted in Figure 4.20 is the periodic sinusoidal policy pitching trajectory along with the spline policy. As can be seen near the end of the trajectory, the slope is negative and then must quickly change to a positive slope for the spline policy. Because of the lack of

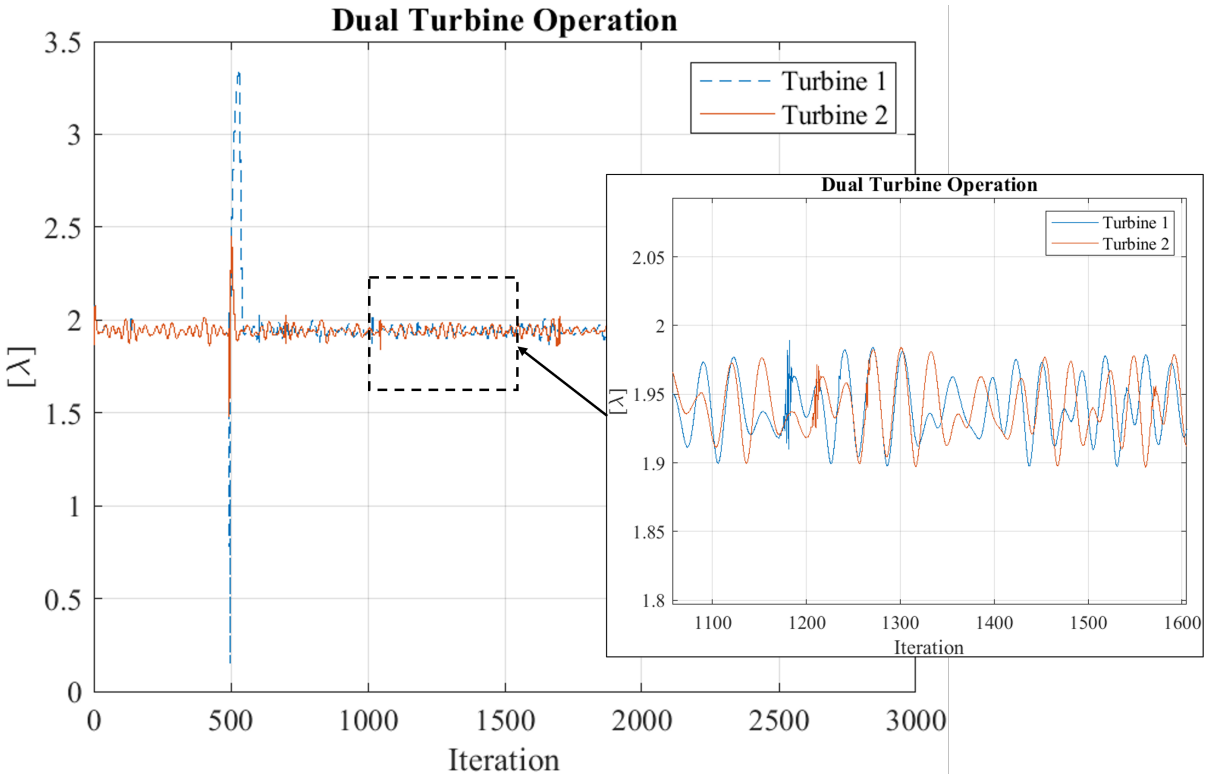


Figure 4.17: Dual turbine speed comparison.

individual blade dynamics, the algorithm has no transitional information as the DMST algorithm assumes no boundary layer separation. In addition, this abrupt movement is sure to introduce unnecessary vibration to the turbine, and given the frequency at which the turbine operates may not even be possible to move at such a high rate. Therefore, although a good test for the PGPE algorithm this policy can be ruled out. In contrast, the sinusoidal policy has a much smoother transition between revolutions due to the periodic nature of the sine curve and is close in average power efficiency.

Comparison to a Non-Controlled Turbine

Finally, Figure 4.21 shows the optimal power coefficient of the wind turbine without any pitching. The optimal turbine speed is at 2.4 with a C_P of 0.2749. In comparison, the controlled turbine for the sinusoidal policy had a mean reward of C_P 0.4066 giving an

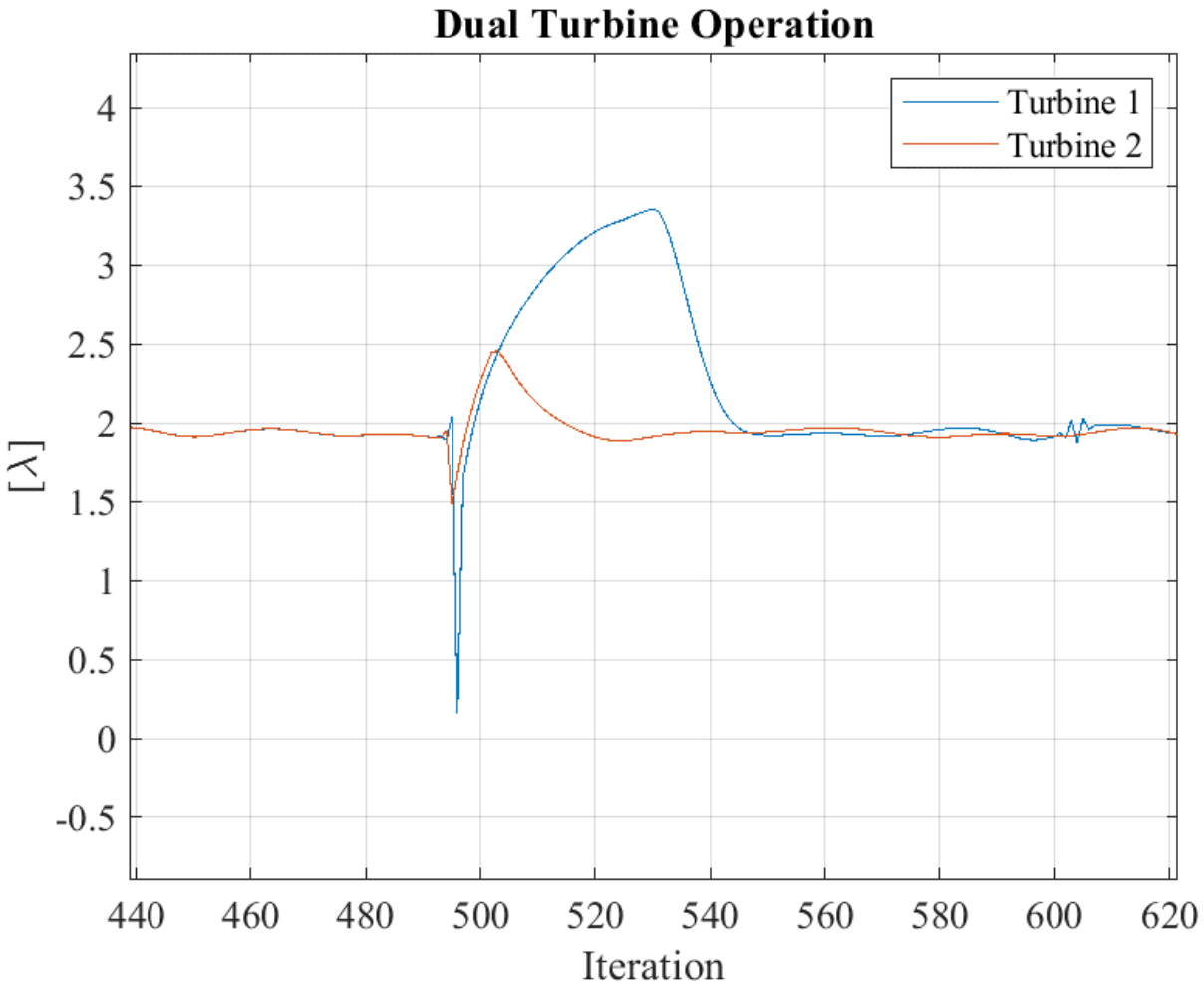


Figure 4.18: Large difference between turbine speeds.

improvement of 47.9% when compared to a non-controlled turbine. Lastly, to show the mitigation of dynamic stall due to the pitch control implementation, Figure 4.22 shows the reduction in the angle of attack during turbine operation in which the pitch angle is added to the inflow angle to give us the effective angle of attack, α', in which is now the angle used to calculate turbine performance. Refer to Eq. 2.4.

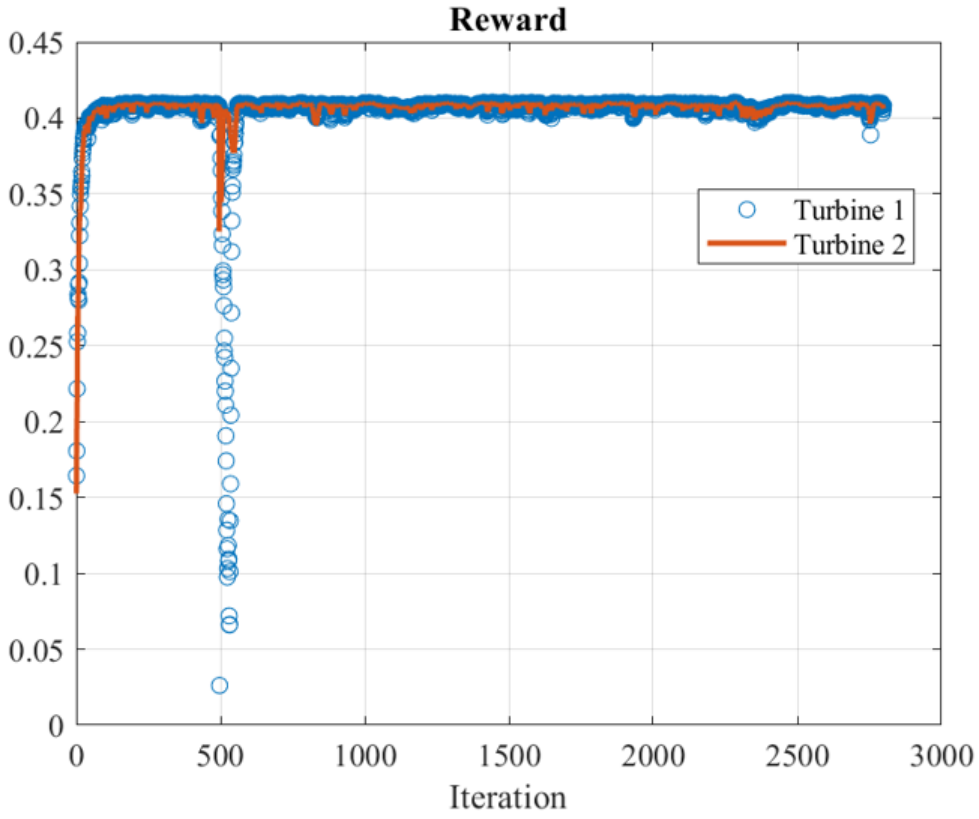
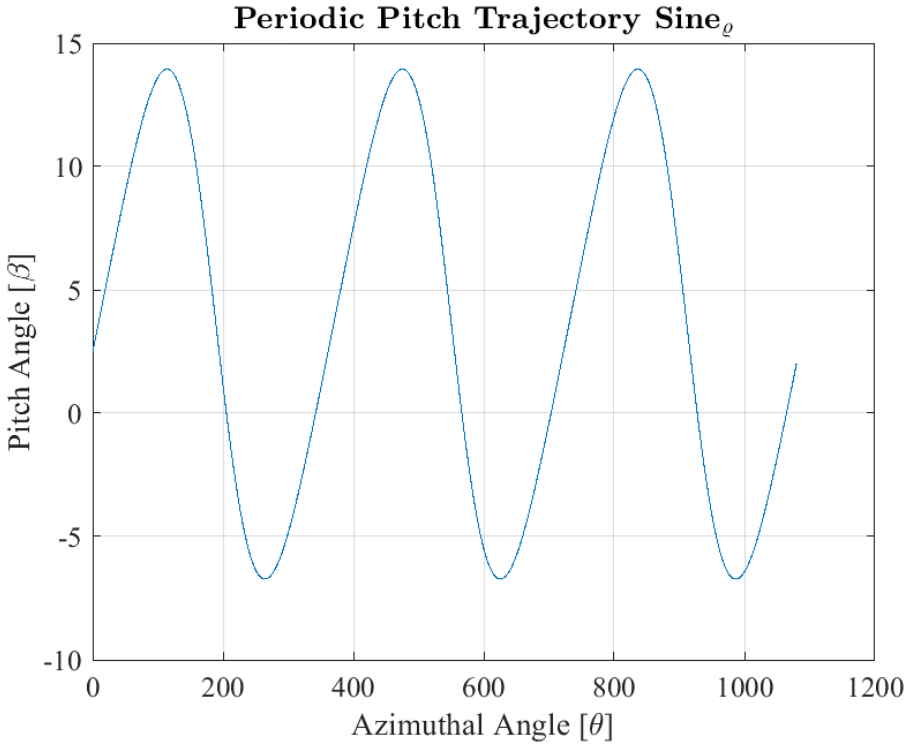
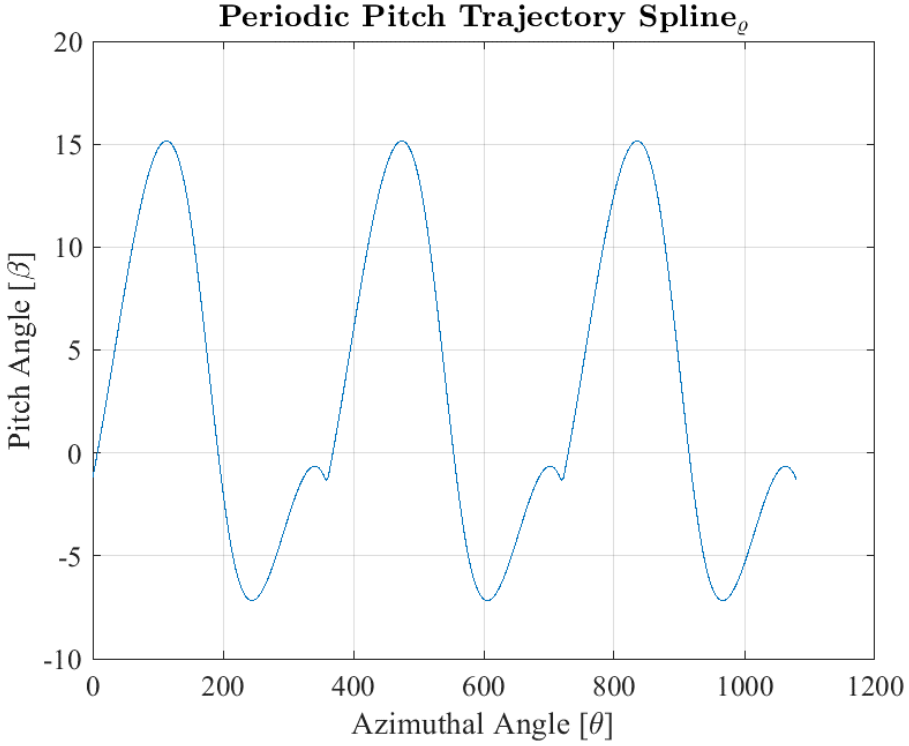


Figure 4.19: Reward comparison of turbines.



(a) Sine policy.



(b) Spline policy.

Figure 4.20: Periodic pitching trajectories.

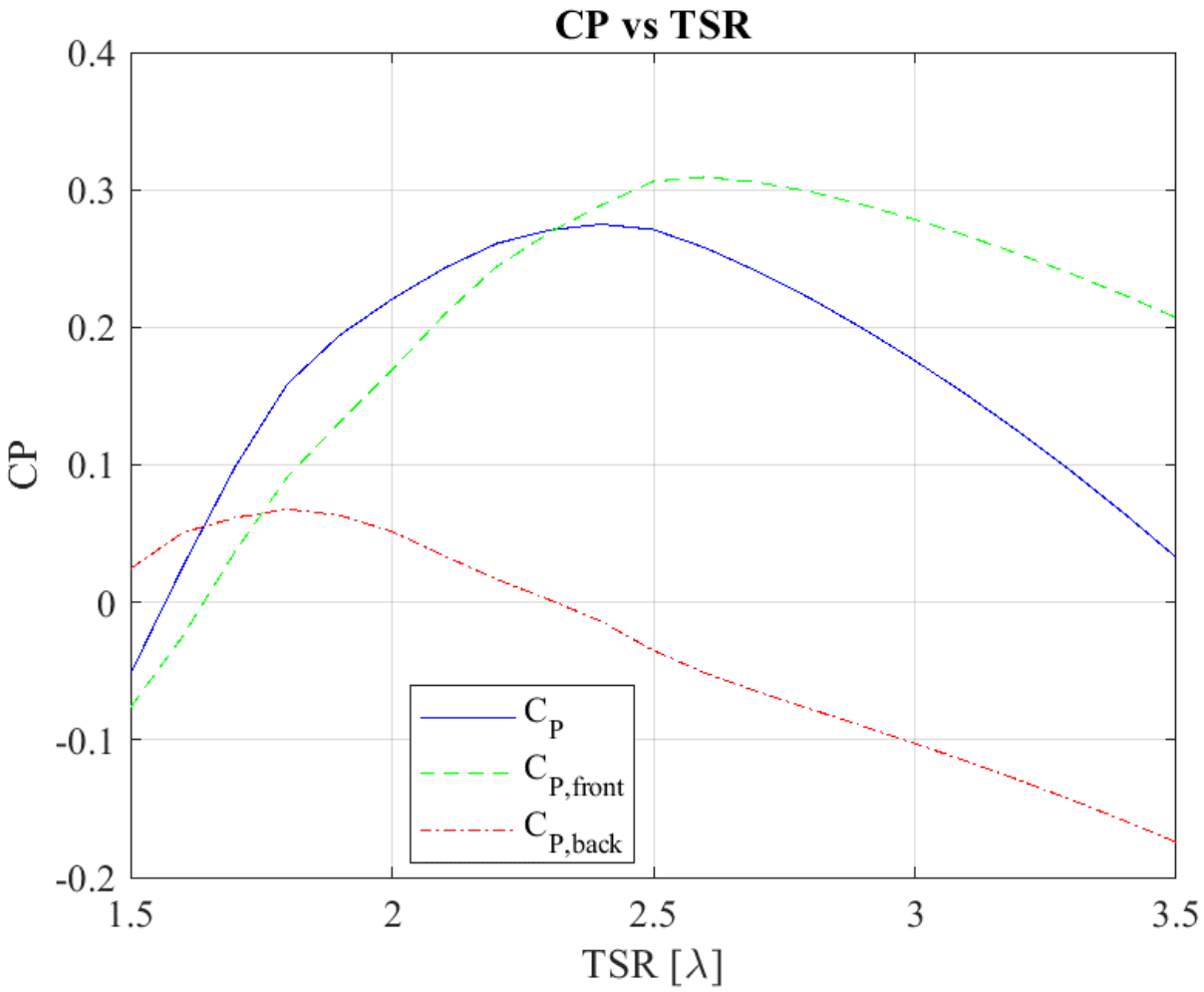


Figure 4.21: Non-pitched turbine.

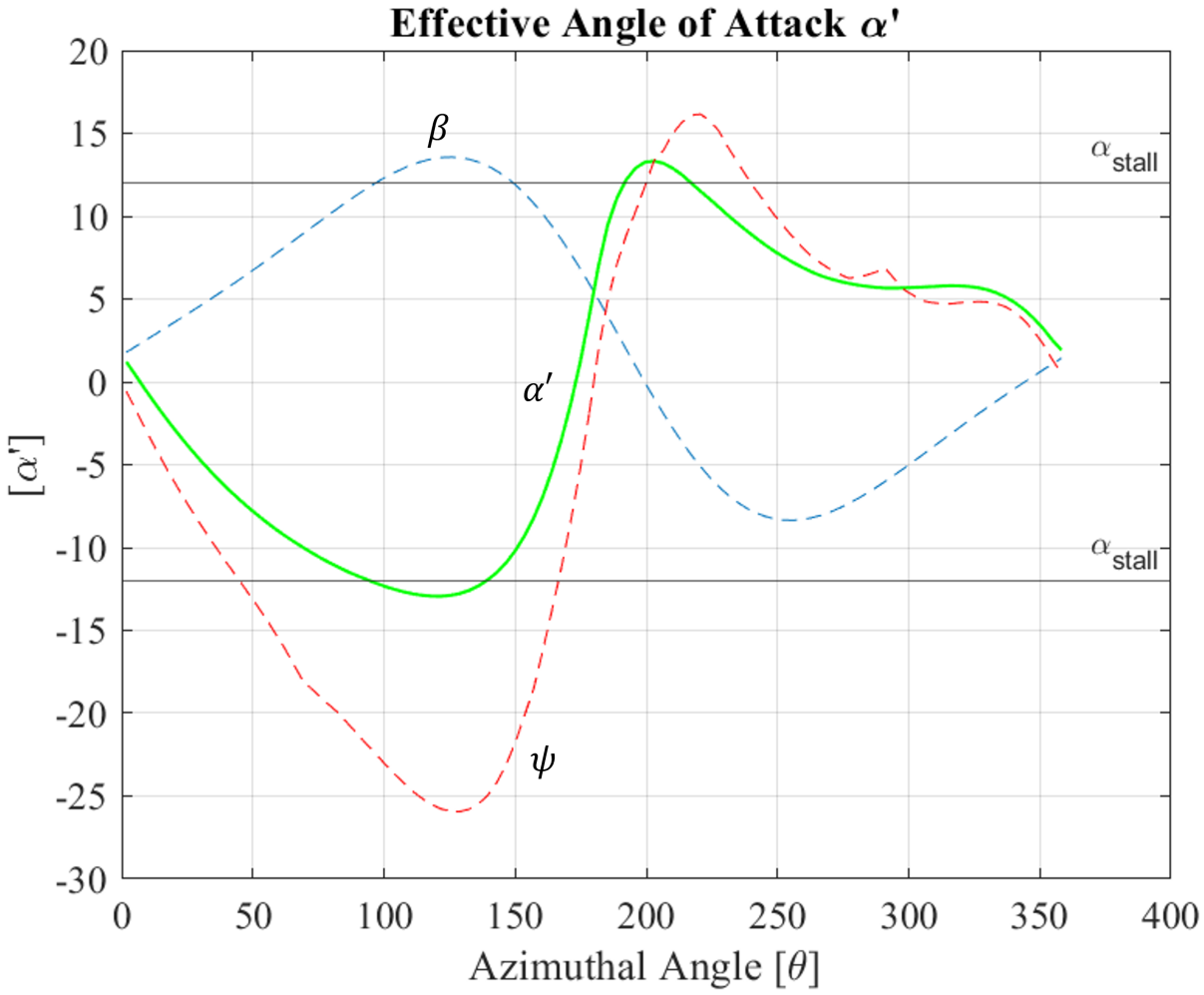


Figure 4.22: Angle of attack reduction during operation for $\lambda = 1.94$.

CHAPTER 5

Conclusion and Future Works

5.1 Conclusion

This work has demonstrated the use of a reinforcement learning algorithm to learn the VAWT dynamics by directly interacting with the environment and learning by trial and error. We demonstrated the use of a newer PGPE algorithm which incorporated newer machine learning techniques that greatly improved the learning time. We also demonstrated the use of PGPE as an online RL algorithm and its stability even under non-ideal symmetric samples. In the case of the wind turbine operating under unsteady air, the PGPE algorithm was able to compensate for the changing environment with minimal fine-tuning of hyperparameters. In addition, the sinusoidal policy was demonstrated to be a great candidate for the VAWT pitching that was intuitive and gave a great start to the formulation of an optimized pitch trajectory based on only 6 parameters.

The development of future renewable energy systems intertwined with new state-of-the-art technology is advancing the direction of renewable energy developments. Where lack of technology once limited the extraction of energy under non-ideal environments, such as urban areas for wind energy, are now being reconsidered. Technology has allowed advancements in electronics, manufacturing techniques, and new methods for energy harvesting that were once considered a far reach. Continued research efforts are still ongoing as there is still much to be discovered and challenges to be solved.

5.2 Future Direction

Due to the theoretical approach of the turbine environment the implementation of PGPE on a physical turbine system is to be investigated. Challenges of implementation of PGPE to a turbine system require a strong control system that moderates the speed of both turbines and an investigation of the setup of both turbines such that the wake of one system does not affect the other. In addition, switching from an episodic time frame to a continuous application by using methodologies from existing RL techniques. Finally, the extension of PGPE to a wind farm system from the dual turbine setup. This would allow the use of a better gradient estimation by using a larger population size readily implemented on PGPE.

REFERENCES

[1] U.S. Energy Information Administration. How much of U.S. energy consumption and electricity generation comes from renewable energy sources? https://www.eia.gov/tools/faqs/faq.php?id=92&t=4, 2023.

[2] U.S. Energy Information Administration. Monthly Energy Review. Technical report, U.S. Energy Information Administration, 2024.

[3] D. Feldman, K. Dummit, J. Zuboy, R. Margolis. Spring 2023 Solar Industry Update. Technical report, National Renewable Energy Laboratory, 2023.

[4] Global Renewables Alliance, COP28 Presidency, et al. Tripling renewable power and doubling energy efficiency by 2030: Crucial steps towards 1.5°C. *COP 28 UAE*, 2023.

[5] Global Wind Energy Council. GWEC Global Wind Report 2023. *Global Wind Energy Council: Brussels, Belgium*, 2023.

[6] L. Cozzi, D. Wetzel, Gianluca T., and J. Hyppolite II. For the first time in decades, the number of people without access to electricity is set to increase in 2022. 2022.

[7] Environmental Protection Agency. Climate Change Indicators: Ecosystems. https://www.epa.gov/climate-indicators/ecosystems, 2023.

[8] U.S. National Park Service. A History of Earth's Climate. https://www.nps.gov/cajo/learn/nature/history-of-earths-climate.htm, 2023.

[9] NASA Earth Observatory. Global Temperatures. https://earthobservatory.nasa.gov/world-of-change/global-temperatures#:~:text=According%20to%20an%20ongoing%20temperature,1.9%C2%B0%20Fahrenheit)%20since%201880., 2022.

[10] Environmental Protection Agency. Overview of Greenhouse Gases. https://www.epa.gov/ghgemissions/overview-greenhouse-gases#f-gases, 2024.

[11] U.S. Energy Information Administration. History of Wind Power. https://www.eia.gov/energyexplained/wind/history-of-wind-power.php, 2023.

[12] Paul G. Austrian was first with wind electric turbine, not Byth or De Goyon. https://wind-works.org/austrian-was-first-with-wind-electric-turbine-not-byth-or-de-goyon/, 2023.

[13] Renewable Energy World. History of wind turbines. https://www.renewableenergyworld.com/storage/grid-scale/history-of-wind-turbines/#gref, 2014.

[14] B. S. Linscott, P. Perkins, J.T. Dennett. Large Horizontal-Axis Wind Turbines. Official Memorandum, NASA, 1984.

[15] H. J. Sutherland, D. E. Berg and T. D. Ashwill. A Retrospective of VAWT Technology. *Sandia National Laboratories*, pages 1–64, 2012.

[16] P. Maegaard, A. Krenz, and W. Palz, editors. *Wind power for the world: the rise of modern wind energy.* CRC Press, Boca Raton, FL, 2013.

[17] O. Peter and C. Mbohwa. Renewable energy technologies in brief. *International Journal of Scientific and Technology Research*, 8:1283–1289, 2019.

[18] International Renewable Energy Agency. Renewable power generation costs in 2022, 2023.

[19] B. Summerville. Grand challenges for small wind: Results of the european academy study. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2023.

[20] CJ S. Ferreira, H. Bijl, G. van Bussel, and G. Van Kuik. Simulating dynamic stall in a 2D VAWT: Modeling strategy, verification and validation with Particle Image Velocimetry data. In *Journal of physics: conference series*, volume 75, page 012023. IOP Publishing, 2007.

[21] C. N. Eastlake. An aerodynamicist's view of lift, bernoulli, and newton. *The Physics Teacher*, 40(3):166–173, 2002.

[22] A. Rezaeiha, H. Montazeri, and B. Blocken. Towards optimal aerodynamic design of vertical axis wind turbines: Impact of solidity and number of blades. *Energy*, 165:1129–1148, 2018.

[23] M. A. Miller, S. Duvvuri, W. D. Kelly, and M. Hultmark. Rotor solidity effects on the performance of vertical-axis wind turbines at high reynolds numbers. In *Journal of Physics: Conference Series*, volume 1037, page 052015. IOP Publishing, 2018.

[24] Y. Peng, Y. Xu, and S. Zhan. A hybrid dmst model for pitch optimization and performance assessment of high-solidity straight-bladed vertical axis wind turbines. *Applied Energy*, 250:215–228, 2019.

[25] W. Tjiu, T. Marnoto, S. Mat, M. H. Ruslan, and K. Sopian. Darrieus vertical axis wind turbine for power generation i: Assessment of darrieus vawt configurations. *Renewable Energy*, 75:50–67, 2015.

[26] S. Brusca, R. Lanzafame, and M. Messina. Design of a vertical-axis wind turbine: How the aspect ratio affects the turbine's performance. *International Journal of Energy and Environmental Engineering*, 5:333–340, 2014.

[27] P. Jain and A. Abhishek. Performance prediction and fundamental understanding of small scale vertical axis wind turbine with variable amplitude blade pitching. *Renewable Energy*, 97:97–113, 2016.

[28] G. Abdalrahman, W. Melek, and F. Lien. Pitch angle control for a small-scale darrieus vertical axis wind turbine with straight blades (h-type vawt). *Renewable Energy*, 114:1353–1362, 2017.

[29] W A Moran. Giromill wind tunnel test and analysis. volume ii. technical discussion. final report, june 1976–october 1977. 10 1977.

[30] R. D. McConnell. Giromill overview. In *Wind Energy Innovative Systems Conference Proceedings: May 23-25, 1979, The Four Seasons Motor Inn, Colorado Springs, Colorado*, volume 49, page 319. AWEA, 1979.

[31] Q. Zhang, M. Bashir, W. Miao, Q. Liu, C. Li, M. Yue, and P. Wang. Aerodynamic analysis of a novel pitch control strategy and parameter combination for vertical axis wind turbines. *Renewable Energy*, 216:119089, 2023.

[32] A. Sagharichi, M. J. Maghrebi, and A. ArabGolarcheh. Variable pitch blades: An approach for improving performance of darrieus wind turbine. *Journal of Renewable and Sustainable Energy*, 8(5), 2016.

[33] Y. Staelens, F. Saeed, and I. Paraschivoiu. A straight-bladed variable-pitch vawt concept for improved power generation. In *Wind Energy Symposium*, volume 75944, pages 146–154, 2003.

[34] C. Li, Y. Xiao, Y. Xu, Y. Peng, G. Hu, and S. Zhu. Optimization of blade pitch in h-rotor vertical axis wind turbines through computational fluid dynamics simulations. *Applied Energy*, 212:1107–1125, 2018.

[35] Y. Ma, L. Zhang, Z. Zhang, and D. Han. Optimization of blade motion of vertical axis turbine. *China Ocean Engineering*, 30:297–308, 2016.

[36] I. Paraschivoiu, O. Trifu, and F. Saeed. H-darrieus wind turbine with blade pitch control. *International Journal of Rotating Machinery*, 2009:Article ID 505343, 7 pages, 2009.

[37] L. Zhang, J. Gu, H. Zhu, K. Hu, X. Li, L. Jiao, J. Miao, J. Liu, and Z. Wang. Rationality research of the adjustment law for the blade pitch angle of h-type vertical-axis wind turbines. *Renewable Energy*, 167:484–496, 2021.

[38] H. Shen, A. Ruiz, and N. Li. Fast online reinforcement learning control of small lift-driven vertical axis wind turbines with an active programmable four bar linkage mechanism. *Energy*, 262:125350, 2023.

[39] V.T. Aghaei, A. Ağababaoğlu, B. Bawo, P. Naseradinmousavi, S. Yıldırım, S. Yeşilyurt, and A. Onat. Energy optimization of wind turbines via a neural control policy based on reinforcement learning markov chain monte carlo algorithm. *Applied Energy*, 341:121108, 2023.

[40] Z. Xu, J. Chen, and C. Li. Research on the adaptability of dynamic pitch control strategies on h-type vawt close-range arrays by simulation study. *Renewable Energy*, 218:119231, 2023.

[41] Y. Chen, D. Zhang, X. Li, Y. Peng, X. Zhang, Z. Han, Y. Cao, and Z. Dong. Surrogate models for twin-vawt performance based on kriging and artificial neural networks. *Ocean Engineering*, 273:113947, 2023.

[42] Y. Chen, L. Kuang, J. Su, D. Zhou, Y. Cao, H. Chen, Z. Han, Y. Zhao, and S. Fu. Investigation of pitch angles on the aerodynamics of twin-vawt under staggered arrangement. *Ocean Engineering*, 254:111385, 2022.

[43] M. Hassanpour and L. N. Azadani. Aerodynamic optimization of the configuration of a pair of vertical axis wind turbines. *Energy Conversion and Management*, 238:114069, 2021.

[44] W. Chen, C. Chen, C. Huang, and C. Hwang. Power output analysis and optimization of two straight-bladed vertical-axis wind turbines. *Applied Energy*, 185:223–232, 2017.

[45] N. E. Toklu, P. Liskowski, and R. K. Srivastava. Clipup: a simple and powerful optimizer for distribution-based policy evolution. In *International Conference on Parallel Problem Solving from Nature*, pages 515–527. Springer, 2020.

[46] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber. Policy gradients with parameter-based exploration for control. In *Artificial Neural Networks-ICANN 2008: 18th International Conference, Prague, Czech Republic, September 3-6, 2008, Proceedings, Part I 18*, pages 387–396. Springer, 2008.

[47] J. F. Manwell, J. G. McGowan, and A. L. Rogers. *Wind energy explained: theory, design and application*. John Wiley & Sons, West Sussex, UK, 2010.

[48] D. De Tavernier, C. Ferreira, and A. Goude. Vertical-axis wind turbine aerodynamics. In *Handbook of Wind Energy Aerodynamics*, pages 1317–1361. Springer, 2022.

[49] M. Singh and S. Santoso. Dynamic models for wind turbines and wind power plants. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2011.

[50] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, Cambridge, Massachusetts, 2018.

[51] F. Sehnke. *Parameter exploring policy gradients and their implications*. PhD thesis, Technische Universität München, 2012.

[52] Z. Li and J. W. Langelaan. Parameterized trajectory planning for dynamic soaring. In *AIAA Scitech 2020 Forum*, page 0856, 2020.

[53] F. Balduzzi, A. Bianchini, R. Maleci, G. Ferrara, and L. Ferrari. Critical issues in the cfd simulation of darrieus wind turbines. *Renewable Energy*, 85:419–435, 2016.

[54] D. V. Cabrera. Study on vertical-axis wind turbines using streamtube and dynamic stall models. B.S. thesis, Universitat Politècnica de Catalunya, 2014.

[55] D. Vallverdu. VAWT Analysis. `https://www.mathworks.com/matlabcentral/fileexchange/46909-vawt-analysis`, 2024. Retrieved May 8, 2024.

Appendix A

Simulating Environment using DMST Model

Provided here is the Collaborative Online Reinforcement Learning environment implemented in MATLAB using the DMST model from [55]. This environment will be simulated twice for each turbine in which case each will have its own dynamic trajectory with different outputs that will be used by the PGPE algorithm. The global variable 'param' and 'param2' contains all the geometrical parameters and states for the VAWTs, the PI load control, PGPE parameters, and wind parameters. This is defined at the beginning of the code to allow all functions to use these variables without redefining. In addition to giving the geometric parameters to the DMST software to create the VAWT, the aerodynamic coefficients of the airfoil being used is needed. The airfoil data provided by the software was used for the 'NACA 0021' airfoil as it was used to validate its model.

```
function [reward,t,omega,windspeed,TSR,IER] = env_dyn(mu,param)
%% Load turbine parameters      % geometric, wind data, and PI parameters
myTurbine = param.myTurbine;    % creates turbine model in DMST software
R = param.R;                    % turbine radius
t = param.t;                    % time of the overall sim
tsr = param.tsr;                % TSR init/last iteration
ier = param.ier;                % integral error from last iteration
theta = myTurbine.theta;        % discretized angular positions

%% load wind speed and turbine speed
```

```
if t == 0
    wind = param.wind_init;
    omega = tsr*wind/R;
else
    wind = para.windspeed;
    omega = tsr*wind/R;
end

%% Generate pitch trajectory based on policy using
policy = param.policy;
switch policy
    case 'spline'
    pitch = spline(0:(2*pi/(param.M)):2*pi,[mu(1:param.M);mu(1)],theta);
    case 'sine'
    pitch = sinuosodial_f(mu,theta);
end

%% Simulate turbine under conditions
ii = 1;         % counter
num_iter = 1;   % number of revolutions to simulate
reward = zeros(length(num_iter));
while ii <= num_iter
```

```
%% Solve using DMST model. See reference [55].
myTurbine.set  ('TSR' , tsr );
myTurbine.set  ('pitch' , pitch );
myTurbine.set  ('U' , wind );
myTurbine.solve
Cp = myTurbine.solution.power.CP;       % used as the reward
Tor = myTurbine.solution.torque.CT;     % used to step time
Torque = zeros(1,length(theta));
    for k = 1: length(theta)
    Torque(k) = 0.5*param.rho*2*R*wind^2*R*Tor(k);
end
Tau = mean(Torque);


%% Update load control
er = param.tsr_ref - tsr;
ier = ier + er*2*pi/omega;
Load = max(-param.Kp*er - param.Ki*ier,0);
T =  Tau - Load;
omega = omega+T/param.J*2*pi/omega;


if omega < 0 % For the rest of the simulation the reward will be zero
break
end
```

```
    tsr = omega*R/wind;     % new TSR
    t = t+2*pi/omega;       % new time
    % interpolate new wind data
    wind = interp1(param.wind_data(:,1),param.wind_data(:,2),t);


    %% Reward
    reward(ii) = CP;


    ii = ii + 1;
    windspeed = wind;   % update wind
    TSR = tsr;          % update tsr
    IER = ier;          % update integral error for PI control
end
```

Policy formulation for sinusoidal pitch trajectory implemented in MATLAB.

```
function sine_pitch = sinuosodial_f(mu,theta)


a_ = mu(1);                             % skewness factor
b = mu(2);                              % flatness parameter
d = mu(3);                              % offset parameter
s1 = mu(4);                             % amplitude for upwind half
s2 = mu(5);                             % amplitude for downwind half
```

```
zeta_ = mu(6);                          % phase shift point

fo = sin(theta+zeta_);                  % initial sine curve

T1 = sin(theta+a_*fo+zeta_);            % first transformation

T2 = sqrt((1+b^2)./(1+b^2*T1.^2)).*T1; % second transformation

T3 = zeros(1,length(theta));

s = s1-(s1-s2)*sigmf(theta-pi,[1 10]); % third transformation

T3 = s.*T2+d;


sine_pitch = T3;

end
```

Appendix B

PGPE Symmetric Sampling with Clip Up

Code provided here is the code used in this thesis for the RL algorithm. Here PGPE with Symmetric Sampling from [51] and the Clip Up optimizer from [45] is implemented in MATLAB to optimize the parameterized policies.

```
%% Initialize all PGPE parameters
num_sol = #;                                % how many center solutions
mu = zeros(num_sol,1);                      % init center solutions
r_pgpe = param.q*param.vel_max*pi/180;      % radian space
sig = ones(length(mu),1)*r_pgpe;            % init deviation


i =1
while i < param.iters    % param.iters time to train
    for n = 1:param.Ns   % param.Ns = 1 for our case
        sig_sq = eye(length(sig)).*sig;
        epsilon = diag(normrnd(0,sig_sq));  % E ~ N(0,I*sig)
        mu_plus = mu+epsilon;               % symm sample 1
        mu_minus = mu-epsilon;              % symm sample 2
        [r_plus] = env_dyn(mu_plus,param);  % sim turbine 1
        r_plus = sum(r_plus);               % rew from turbine 1
        [r_minus] = env_dyn(mu_minus,param2);% sim turbine 2
        r_minus = sum(r_minus);             % rew from turbine 2
```

```
    T(:,n) = epsilon;
    S(:,n) = (epsilon.^2-sig.^2)./sig;
    rt(n) = (r_plus-r_minus)/(2*param.m-r_plus-r_minus);
    rs(n) = ((r_plus+r_minus)/2-param.rb)/(param.m-param.rb);
end

%% Gradient update
est_grad = T*rt; % estimated gradient

%% Clip Up. See reference [45]
eucl_est_grad = norm(est_grad);
if eucl_est_grad == 0 % Check if zero
    grad_norm = zeros(size(est_grad));
    vel_prime = param.momentum*vel;
else
    grad_norm = est_grad/eucl_est_grad; % nomalized gradient
    vel_prime = param.momentum*vel + param.alpha*grad_norm;
end
eucl_vel_prime = norm(vel_prime); % norm of velocity
if eucl_vel_prime > param.vel_max % cap vel update
    vel = param.vel_max*(vel_prime/eucl_vel_prime);
else
    vel = vel_prime;
```

```
    end


    %% update center solution and stdev
    % update center solution
    mu = mu + vel; % update the center solution


    % update stdev
    old_sig = sig;
    sig = sig + param.sig_alpha*S*rs; % update stdev
    allowed_change = abs(old_sig) * param.sig_max;
    sig_min_allowed = old_sig - allowed_change;
    sig_max_allowed = old_sig + allowed_change;
    sig = clip(sig, sig_min_allowed, sig_max_allowed);
end
```