BAYESIAN REINFORCEMENT LEARNING WITH MCMC TO MAXIMIZE ENERGY OUTPUT OF VERTICAL AXIS WIND TURBINE

by

Arda Ağababaoğlu

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

SABANCI UNIVERSITY

June 2019

BAYESIAN REINFORCEMENT LEARNING WITH MCMC TO MAXIMIZE ENERGY OUTPUT OF VERTICAL AXIS WIND TURBINE

APPROVED BY

Assoc. Prof. Dr. Ahmet Onat .....
(Thesis Supervisor)

Prof. Dr. Serhat Yeşilyurt .....

Prof. Dr. Müjde Güzelkaya .....

DATE OF APPROVAL: 20/06/2019

ABSTRACT

BAYESIAN REINFORCEMENT LEARNING WITH MCMC TO MAXIMIZE ENERGY OUTPUT OF VERTICAL AXIS WIND TURBINE

ARDA AĞABABAOĞLU

Mechatronics Engineering M.Sc. Thesis, June 2019

Thesis Supervisor: Assoc. Prof. Dr. Ahmet Onat

Keywords: Reinforcement Learning, Markov Chain Monte Carlo, Radial Basis Function Neural Network, Wind Energy Conversation Systems, Vertical Axis Wind Turbines

Optimization of energy output of small scale wind turbines requires a controller which keeps the wind speed to rotor tip speed ratio at the optimum value. An analytic solution can be obtained if the dynamic model of the complete system is known and wind speed can be anticipated. However, not only aging but also errors in modeling and wind speed prediction prevent a straightforward solution.

This thesis proposes to apply a reinforcement learning approach designed to optimize dynamic systems with continuous state and action spaces, to the energy output optimization of Vertical Axis Wind Turbines (VAWT). The dynamic modeling and load control of the wind turbine are accomplished in the same process. The proposed algorithm is a model-free Bayesian Reinforcement Learning using Markov Chain Monte Carlo method (MCMC) to obtain the parameters of an optimal policy.

The proposed method learns wind speed profiles and system model, therefore, can utilize all system states and observed wind speed profiles to calculate an optimal control signal by using a Radial Basis Function Neural Network (RBFNN). The proposed method is validated by performing simulation studies on a permanent magnet synchronous generator-based VAWT Simulink model to compare with the classical Maximum Power Point Tracking (MPPT). The results show significant improvement over the classical method, especially during the wind speed transients, promising a superior energy output in turbulent settings; which coincide with the expected application areas of VAWTs.

ÖZET

DİKEY EKSENLİ RÜZĞAR TÜRBİNİNİN ENERJİ ÇIKTISINI BÜYÜTMEK İÇİN MZMC İLE BAYESÇİ PEKİŞTİRMELİ ÖĞRENME

ARDA AĞABABAOĞLU

Mekatronik Mühendisliği Yüksek Lisans Tezi, Haziran 2019

Tez Danışmanı: Doç. Dr. Ahmet Onat

Anahtar Kelimeler: Pekiştirmeli Öğrenme, Markov Zincirli Monte Carlo, Dairesel Tabanlı Fonksiyon Sinir Ağı, Rüzgar Enerjisi Dönüştürme Sistemleri, Dikey Eksenli Rüzgar Türbini

Küçük ölçekli rüzgâr türbinlerinin (DERT) enerji çıkışının optimizasyonu, rüzgâr hızını rotor uç hızı oranını optimum değerde tutan bir kontrolör gerektirmektedir. Eğer dinamik model sistemin tamamı bilinir ve rüzgâr hızı tahmin edilebilirse, analitik bir çözüm elde edilebilir. Ancak, sadece yaşlanma değil aynı zamanda modelleme ve rüzgar hızı tahminindeki hatalar basit bir çözümü engeller.

Bu tezde, Dikey Eksenli Rüzgar Türbinlerinin enerji çıkış optimizasyonuna, sürekli durum ve aksiyon uzaylarına sahip dinamik sistemleri optimize etmek için tasarlanmış bir Pekiştirmeli öğrenme yaklaşımı uygulaması önerilmektedir. Rüzgar türbininin dinamik modellemesi ve yük kontrolü; tek süreç içinde ele alınmaktadır. Önerilen algoritma bir optimal politikanın parametrelerini elde etmek için Markov Zincirli Monte Carlo kullanarak modelden bağımsız Bayesçi Pekiştirmeli Öğrenmedir.

Önerilen yöntem rüzgar hızı profillerini ve sistem modelini öğrenir, bu nedenle, Dairesel Tabanlı Fonksiyon Sinir Ağı (DTFSA) kullanarak optimal kontrol sinyalini hesaplamak için tüm sistem durumlarını ve gözlenen rüzgar hızı profillerini kullanabilir. Önerilen yöntem, klasik Maksimum Güç Noktası Takipçisi (MGNT) ile karşılaştırmak üzere sabit mıknatıslı senkron jeneratör tabanlı DERT Simulink modeli için simülasyon çalışmaları yapılarak doğrulanır. Sonuçlar klasik yöntem ile kıyaslandığında, özellikle rüzgar hızı geçişlerinde, önemli bir gelişme göstermiştir, ayrıca değişken hızlar için umut vadeden enerji çıktısı göstermiştir ki bu Dikey Eksenli Rüzgar Türbinlerini (DERT) için istenilen bir durumdur.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Ahmet Onat, for his considerable encouragement, worthwhile guidance and insightful comments to complete this thesis. I feel honored for the opportunity to work under the supervision of him. Besides, I am thankful to Assoc. Prof. Dr. Onat for providing me continuous financial support during my master's studies.

I would like to thank Prof. Dr. Serhat Yeşilyurt and Prof. Dr. Müjde Güzelkaya for their careful evaluation of my thesis and useful comments.

I am obviously indebted to the best teammate, Vahid Tavakol Aghaei, for his tremendous contribution to this work as well as for being the amazing guy he is.

I am thankful to the lab members Umut Çalışkan, Özge Orhan. I also want to thank Hatice Çakır and Elif Saraçoğlu, who have made my graduate student life at Sabanci University more enjoyable. I would also like to state my special thanks to Cansu İşbilir for your support of my thesis and presentation. Special thanks also go to my close friends: Yiğit Cem Sarıoğlu, Özkan Menzilci, and Kutay Kızılkaya.

I am deeply grateful to my parents and brother, Şebnem, Altan and Berke for their immense love, endless support and trust.

Finally, I would like to express my heartfelt gratitude and sincere appreciation to my beloved girlfriend as well as my best friend, Berna Ünver, for her endless love, support (both emotional and technical), care and patience. I am very fortunate to have her by my side.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The main objective of the machine learning (ML) algorithm in control is to ensure a system that is optimally adapted to uncertain conditions. One of the most powerful ML methods for control problems in literature is reinforcement learning (RL) [1]. In RL, sequentially, the agent provides a transition from the current state to a next state by applying an action. The quality of this transition is defined by the reward function which is used to find an optimal policy for performance criterion based on the long-term goals. General scheme of RL is illustrated in Figure 1.1.

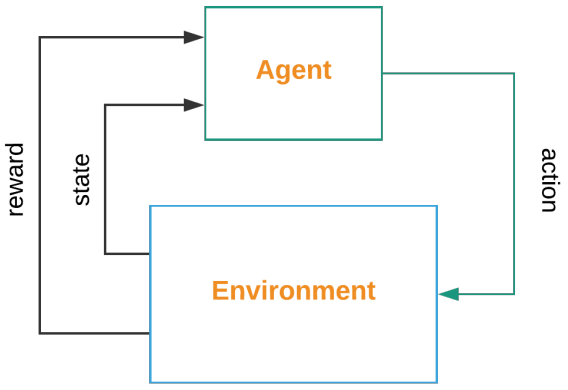


FIGURE 1.1: Reinforcement learning general scheme.

The RL approaches such as Policy Gradient (PG) algorithms are qualified to cope with continuous state spaces encountered in control problems [2, 3].

Due to growing environmental concerns, renewable energy systems have gained great importance worldwide. The wind energy conversion systems have recently been got attention among the renewable energy sources [4]. A wide variety of wind turbine models are available which can be classified mainly as horizontal axis wind turbines (HAWTs), in which the rotor axis is horizontal, in the direction of wind flow, and vertical axis wind turbines (VAWTs), in which the rotor axis is vertical, perpendicular to the direction of wind flow; the advantages of these have been examined in these studies [5–7].

(A) VAWT. (B) HAWT.

FIGURE 1.2: Examples of WECSs.

VAWTs do not need to rotate in to the wind and have fixed aerodynamic structure because they work independently from the wind direction. VAWTs are generally small-scale wind turbines due to physical limitations. The small-scale wind turbines have a power capacity of 1.4–20 kW according to [8]. Recently, the variable-speed

control for the wind energy conversion systems (WECSs) allows capturing more energy from the wind thanks to better power electronic components and pitch control. WECSs are mainly controlled by a electrical load and/or rotor blade pitch angle for obtaining the variable-speed control.

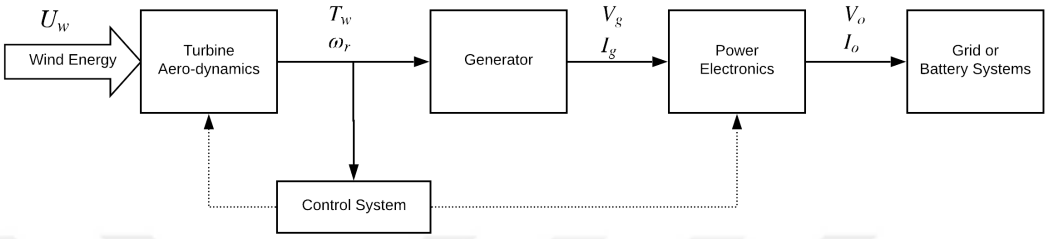


FIGURE 1.3: The schematic of wind energy conversion system.

The control schematic diagram of a WECS is illustrated in Figure 1.3. The kinetic energy of wind is converted to the mechanical energy by wind turbine aerodynamics. The mechanical power is a function of T_w, torque of wind, and ω_r, rotor angular velocity. Then, generator converts wind power to electric power. Generally, the torque of wind(T_w) and/or rotor angular velocity(ω_r) are monitored by the control system in order to calculate the control signal in the form of load current reference. Wind turbine power generation is a function of rotor aerodynamics and wind speed. The power extracted by the blades can be calculated by (1.1) as commonly referred in literature [4].

$$P_w = \frac{1}{2} C_p \rho A_{sw} U_w^3 \tag{1.1}$$

Here, ρ is air density, A_{sw} is swept area of the turbine and U_w is wind speed. The swept area of the VAWT used in this thesis (A_{sw}), is shown in Figure 1.4 and is calculated with (1.2).

$$A_{sw} = 2RL \tag{1.2}$$

The power coefficient C_p defines conversation ratio of the wind energy to the mechanical energy through aerodynamic design. To obtain optimum power output, the

wind speed to rotor tip speed ratio corresponding to a specific ω_r must be maintained. This condition brings C_p to its optimum value. Most of the research in energy or power output focuses on this goal.

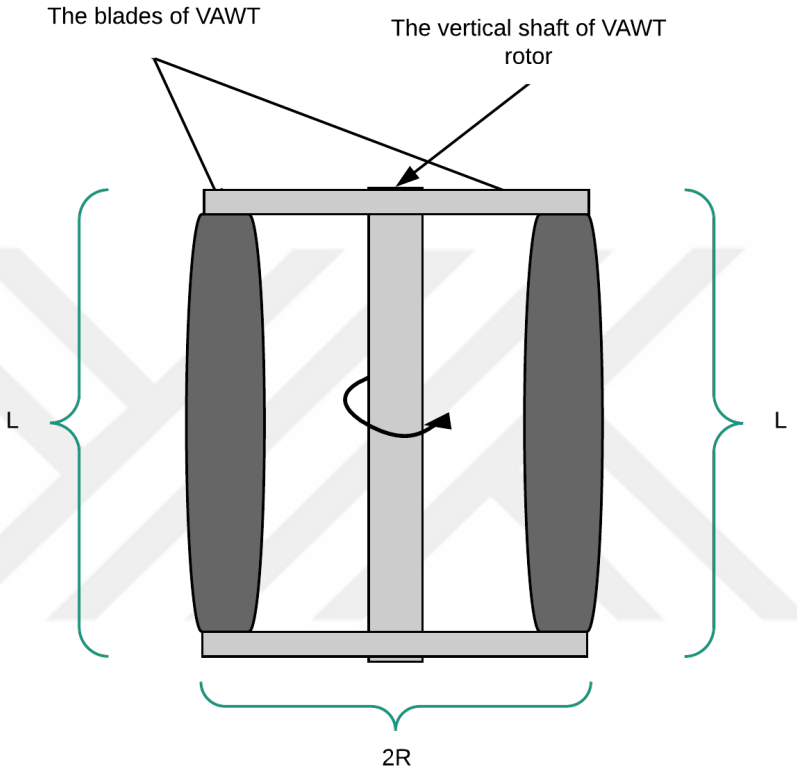


FIGURE 1.4: Swept area of the studied VAWT.

1.1 Motivation

The optimization of obtained energy is one of the most challenging issue for the wind energy conversion system (WECS). The classic control methods (PID, LQR, MPPT) and/or the more advanced control methods such as Model Predictive Control for wind turbine systems demonstrate optimality issue. For all of the mentioned methods, in order to apply a satisfactory control method for the variable wind profile, an online optimization approach is preferred. Furthermore, they demand linearization of the system around some working points. These challenges have inspired us to use an alternative approach which is based on data driven methods, which take into account all nonlinearities of the plant as an implicit or explicit model. Our proposed control strategy uses radial basis function neural network (RBFNN) as a controller that adjusts the value of load coefficient C_L as a control signal in order to maximize the energy output of the VAWT. The plant is a small-scale VAWT system that includes a three-straight-bladed rotor. The parameters of this nonlinear controller itself are learned by Bayesian Reinforcement Learning with Markov Chain Monte Carlo (MCMC) method. This control strategy enables us to learn the model of VAWT to calculate the value of load coefficient C_L as a control signal for given wind speed, rotor speed, load voltage and load current. In addition, unpredictable wind flows always become a significant challenge for wind energy systems; therefore our proposed control strategy addresses this issue explicitly by learning unpredictable wind profiles and the model of the VAWT. Furthermore, the proposed learning method is capable of learning the changes to the VAWT dynamics, such as rotor blade wear, friction coefficient change, aging of components and so on.

1.2 Outline of The Thesis

Chapter 2 presents a literature survey on control approaches for WECS and theoretical background of this research. Chapter 3 presents mathematical model of studied plant. This model consists of the aerodynamics of the VAWT, the generator and the load of VAWT. Chapter 4 presents the state-of-art of Bayesian Reinforcement Learning via MCMC and the RBFNN structure. Chapter 6 presents the parameters and simulation results for proposed Bayesian Reinforcement Learning via MCMC learning control strategy. Chapter 7 is the conclusion of this thesis and future works.

Chapter 2

Literature Survey and Background

In this chapter, control methodologies for VAWT systems in literature are investigated in terms of control performance, weakness, strength and robustness. Different applications of related reinforcement learning algorithms will be presented.

2.1 Control Approaches for Vertical Axis Wind Turbine Systems

The variable-speed control for wind turbine has recently gained attention, because it provides more energy output compared to the fix-speed control. There are several control approaches commonly used for the variable-speed control for wind turbines: Maximum Power Point Tracking (MPPT), model predictive control (MPC) and Reinforcement Learning methods to name a few. Especially, the variable-speed control is favored for VAWTs due to their simple fixed aerodynamic structure.

The first well-known control technique for vertical axis wind turbine systems is Maximum Power Point Tracking (MPPT). There is an optimal tip - speed ratio corresponding to a given generator rotor speed (ω_r) for each turbine for a specific wind speed, which provides the maximum power to be obtained. MPPT algorithms aim to maximize power output along the gradient of this ratio. MPPT is verified to be efficient in maximizing the instantaneous power; however, this is not equivalent to the maximization of the total energy output. There are three types of a MPPT algorithms, namely, Power Signal Feedback control, Tip Speed Ratio control, Perturb and Observe control as given in [9]. Moreover, there are two main control algorithms for controlling a small scale wind turbine by MPPT as outlined in [10]. First calculates the operating point based on preceding knowledge of turbine parameters. Second group's algorithms use iterative methods to explore the optimum control values. There are diverse MPPT approaches and implementations in literature such as adaptive MPPT algorithms [11], fuzzy logic MPPT algorithms [12], sensor-less MPPT algorithms [13–15], and standard MPPT algorithms [16, 17].

The other well-known control technique for VAWT in literature is model predictive control (MPC). MPC is useful when a precise model of the system is at hand and the objective function is convex [18]. This technique applies an on-line optimization approach with predictive look-ahead in order to obtain a satisfactory control signal for the variable wind profile. Since MPC must solve the optimization problem in a limited time, in some cases the objective function may not properly be maximized or minimized. Also, MPC requires the knowledge of future wind velocity that must be collected, and conveyed separately to the wind turbine. In general, drastically changing wind speeds may make the method inefficient, as MPC is a relatively slow control method. There are several implementations of MPC for wind energy conversation systems (WECS) given in [19–24]. Moreover, there are two examples of nonlinear model predictive control applications [25, 26].

Recently, Artificial Intelligence (AI) methods, especially Machine Learning (ML), are applied to dynamic systems such as control and robotic systems [27–29] for maximizing control performance. On the other hand, there are limited numbers of research on controlling wind energy conversation systems (WECS) by ML methods. In [30], the classic control approach MPPT is utilized by using Continuous Actor-Critic method. Reinforcement learning has been used for improving MPPT controller performance [31–33]. Besides, artificial neural-networks are used in WECS control by diverse approaches [34–37], and artificial neural-networks are used for system identification [38]. Furthermore, [39] and [40] practice deep machine learning methods to predict wind speed or wave speed.

Bayesian Reinforcement Learning is an extremely powerful technique for determining the optimal policy in stochastic dynamical systems regardless of system dynamics. This model-free RL approach works under the existence of some prior knowledge that is assumed to be in the form of a Markov decision processes (MDP). Bayesian Reinforcement Learning is a favorable approach for robotic and control problems, since it provides optimal policy without the exact knowledge of system model. The examples of Bayesian reinforcement learning application in robotics are given [28, 29, 41, 42]. The other approaches [43, 44] use Bayesian learning to optimize a controller. WECS can be modeled as MDP similar to robotics systems. However there are no published results for controlling WECS via Bayesian reinforcement learning to the knowledge of the author.

Radial basis function neural network (RBFNN) is a type of the three-layer feed-forward neural network, consisting of an input layer, a hidden layer and an output layer. RBFNN is commonly used for regression problems, pattern recognition, classification problems, and time series predictions. Recently, RBFNN is applied to control problems as a controller or to optimize diverse controllers. Different applications of RBFNN can be listed as; the robotic application shown in[45], [46] uses RBFNN to solve regression problems, [47] and [48] utilize RBFNN as controller for

proses control systems, power system controller implementations can be found in [49] and [50], the electro-mechanical systems are controlled by RBFNN controller in[51, 52].

Chapter 3

Vertical Axis Wind Turbine System Model

This chapter presents the used vertical axis wind turbine (VAWT) models, which consist of the aerodynamics of the VAWT, the generator, the power electronic part, and the load.

3.1 Vertical Axis Wind Turbine

The general schematic block diagram of the studied vertical axis wind turbine is illustrated in Figure 3.1. The studied vertical axis wind turbine system model is obtained from [6]; where all system parameters and all system equations are directly taken from.

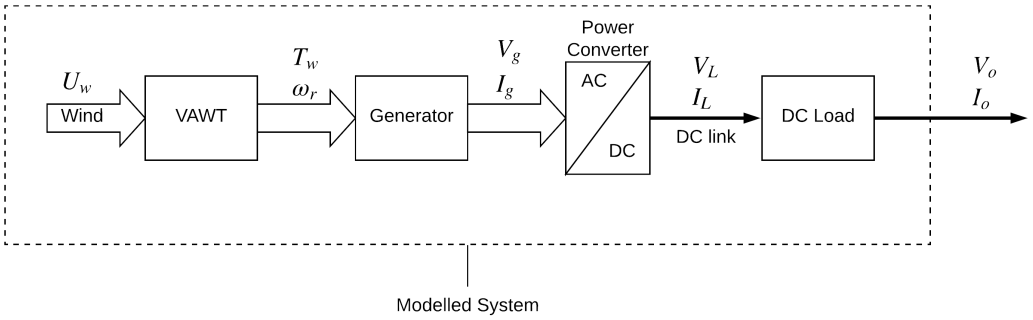


FIGURE 3.1: Block diagram of the studied system

3.1.1 The model of the vertical axis wind turbine

The power coefficient C_p defines conversation ratio of the wind energy to the mechanical energy through aerodynamic design, is generally expressed as a function of the tip-speed-ratio (TSR), λ, illustrated in equations 3.1, to the upstream wind speed.

$$\lambda = \frac{\omega_r R}{U_w} \tag{3.1}$$

where ω_r is the generator rotor speed, R is radius of VAWT and U_w is wind speed. Betz Limit, which indicates the maximum available power from the wind, is equal to 0.59259 [4]. This Betz Limit is also the maximum theoretical value of the power coefficient C_p which is used in (1.1). However, typically, the maximum amount of C_p is lower than Betz limit. Equation (3.2) is constructed in order to use in simulations for mimicking the actual aero-dynamical effects of wind and wind turbine design.

$$P_w = C_p(\lambda)\,\rho R L U_w^3 \tag{3.2}$$

where ρ, C_p and L is described in Table 3.1. Moreover, all parameters of the studied VAWT system are given in Table 3.1, which is obtained from [6].

TABLE 3.1: VAWT system parameters.

Parameter Name	Description	Value	Unit
J_r	Moment of inertia of the rotor	2	$kg-m^2$
R	Radius of the rotor	0.5	m
L	Length of a blade	1	m
b_r	Friction coefficient	0.02	Ns/rad
ρ	Air density	1.2	kg/m^3

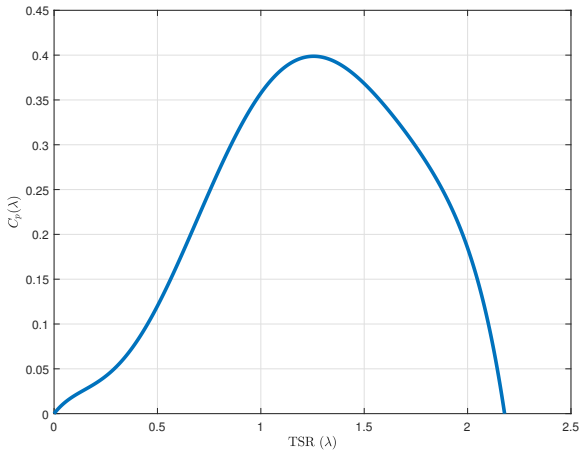


FIGURE 3.2: $\lambda - C_p$ curve of studied system

P_w is available in (3.2), so this equation can be used for calculating T_w (3.4). However, ω_r is not available yet, the generator (3.5) provides ω_r. Moreover, C_p is nonlinear equation of λ that is given in (3.3).

$$C_p(\lambda) = p_1\lambda^6 + p_2\lambda^5 + p_3\lambda^4 + p_4\lambda^3 + p_5\lambda^2 + p_6\lambda \tag{3.3}$$

$$T_w = \frac{P_w}{\omega_r} = \frac{C_p(\lambda)\,\rho RLU_w^3}{\omega_r} \tag{3.4}$$

The coefficients of $C_p(\lambda)$, which are ilustrated in Table 3.2, are taken from [6]. The plot of $C_p(\lambda)$(3.3) versus λ is illustrated in Figure 3.2 by using coefficients are given in Table 3.2.

TABLE 3.2: The coefficient values used in Cp model.

Coefficient	Value
p_1	-0.3015
p_2	1.9004
p_3	-4.3520
p_4	4.1121
p_5	-1.2969
p_6	0.2954

3.1.2 The permanent magnet synchronous generator and simplified rectifier model of the vertical axis wind turbine

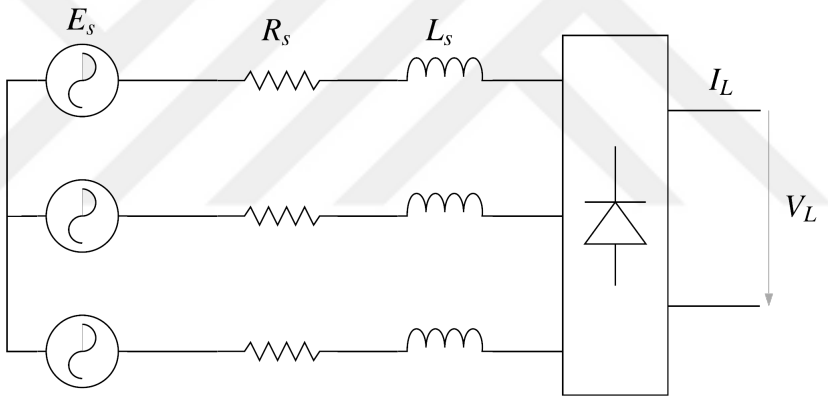


FIGURE 3.3: PMSG-Rectifier schematic.

The permanent magnet synchronous generator (PMSG) equation of motion for the rotor is given by:

$$J_r \frac{d\omega_r}{dt} = T_w - T_g - T_{rf} \tag{3.5}$$

where J_r is the equivalent inertia of the rotor, T_g is the generator torque on the rotor, T_{rf} is the viscous friction torque, which is assumed to be proportional to ω_r by a coefficient b_r as:

$$T_{rf} = b_r \omega_r \tag{3.6}$$

The permanent magnet synchronous generator (PMSG) and passive rectifier electric schematic diagram is illustrated in Figure 3.3 where E_s is electromotive force (EMF), L_s is phase inductance and R_s is the phase resistance of the PMSG. According to [6], the load voltage (V_L) is determined by the generator rotor angular speed (ω_r) and current draw [53]. V_L is maximum when the load current (I_L) is zero. The load voltage (V_L) deceases when I_L increases due to the generator torque (T_g) in (3.5). The generator torque (T_g), which is proportional to I_L by a coefficient torque constant K_t, can be expressed as follows;

$$T_g = K_t I_l \tag{3.7}$$

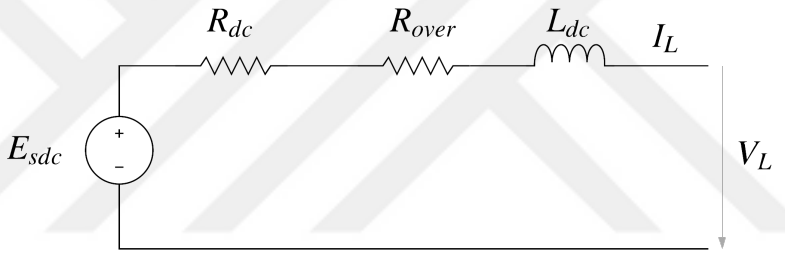


FIGURE 3.4: Simplified DC model of PMSG-Rectifier.

On the other hand, transformation of the 3-phase model to an equivalent DC model with the voltage drops for a given current and the generator rotor speed(ω_r) by ignoring fast dynamics of PMSG and the rectifier models is explained in [54, 55]. The PMSG-rectifier model and the simplified equivalent DC model are shown in Figure 3.4. The voltage drop (R_{dc}) represents PMSG and the rectifier resistive voltage drops. In addition to the resistive voltage drop, to obtain a realistic simplified DC model, R_{over} needs to be included to represent the average voltage drop due to the current commutation in the 3-phase passive diode bridge rectifier, armature reaction in the generator and overlapping currents in the rectifier during commutation intervals.

$$R_{over} = \frac{3L_s p\omega_r}{\pi} \tag{3.8}$$

E_{sdc}, L_{dc}, R_{dc} values in Figure 3.4, which represent the corresponding values between the 3-phase AC model and the equivalent DC model, can be calculated via the values in Table 3.3. Finally, according to [6], V_L is;

$$V_L = \sqrt{E_{sdc}^2 + (p\omega_r L_{dc} I_L)^2} - (R_{dc} + R_{over}) I_L \tag{3.9}$$

TABLE 3.3: PMSG and DC model values.

Variable	**PMSG**	**DC Model**
Flux	ϕ_s	$\phi_{dc} = 3\sqrt{6}\phi_s/\pi$
EMF	$E_s = \phi_s p\omega_r$	$E_{sdc} = 3\sqrt{6}E_s/\pi$
Inductance	Ls	$L_{dc} = 18L_s/\pi^2$
Resistance	R_s	$R_{dc} = 18R_s/\pi^2$
$\phi_s = 0.106Vs/rad,\ p = 6,\ L_s = 3.3mH,\ R_s = 1.7\Omega$		

3.1.3 The load model of the vertical axis wind turbine

In a real application of VAWT, the load consists of high efficiency power electronic elements such as the MOSFET, IGBT, low ESR capacitors and micro-controller for controlling power electronic elements. In this study, the load is represented by a simplified circuit. The load model is illustrated in Figure 3.5. R_L represents the input resistance of power converter or similarly its duty ratio.

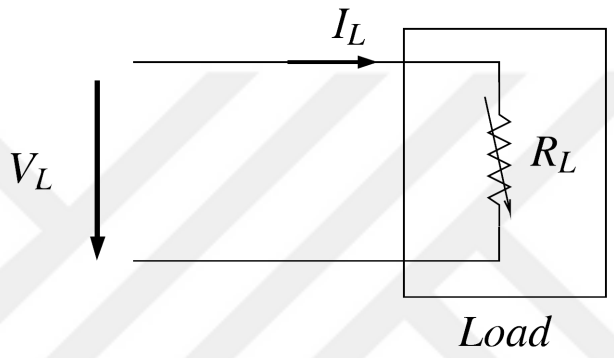


FIGURE 3.5: Simplified load model of VAWT.

Chapter 4

Reinforcement Learning

In general, Reinforcement Learning (RL) problems can be defined as a Markov Decision Process (MDP) that is constructed by the tuple

$$(S, A, g, \eta, r) \tag{4.1}$$

where S and A stand for continuous state and action spaces, respectively. The state transition function g with an initial state function η is a probability distribution which determines the latent state of the agent in each time step given the current state and action $g(s_{t+1}|s_t, a_t)$. During the interaction of the state with its corresponding environment resulting in a transition from the current state s_t to new state s_{t+1} after executing action a_t, a scalar reward r is assigned for evaluating the quality of each individual state transition. This is demonstrated in Figure 1.1 schematically. A parameterized control policy denoted as $h_\theta(a_t|s_t)$ which plays the role of action selection scheme, is defined through a parameter vector θ which belongs to a parameter space Θ. Letting $X_t = (S_t, A_t)$ forms a Markov chain for $\{X_t\}_{t\geq 1}$ with the following transition law

$$f_\theta(x_{t+1}|x_t) := g(s_{t+1}|s_t, a_t)h_\theta(a_t|s_t). \tag{4.2}$$

4.1 Policy Gradient RL

Policy Search (PS) algorithms, which is a favorable RL approach for control problems, focus on finding parameters of policy for a given problem (4.1). The policy gradient algorithms as a PS RL method, which have recently drawn remarkable attention for control and robotic problems, can be implemented in high-dimensional state-action spaces. Because the robotic and control problems usually require dealing with large-dimensional spaces. The discounted sum of the immediate rewards up to time n is defined as

$$R_n(x_{1:n}) := \sum_{t=1}^{n-1} \gamma^{t-1} r(a_t, s_t, s_{t+1}). \tag{4.3}$$

where $\gamma \in (0, 1]$ is a discount factor and $r(a_t, s_t, s_{t+1})$ is reward function. The joint probability density of a trajectory $x_{1:n}$ until time $n-1$ is

$$p_\theta(x_{1:n}) := f_\theta(x_1) \prod_{t=1}^{n-1} f_\theta(x_{t+1}|x_t). \tag{4.4}$$

where $f_\theta(x_1) = \eta(s_1)h_\theta(a_1|s_1)$ and $x_t = (s_t, a_t)$, is the initial distribution for X_1. Furthermore, the general distribution of trajectory function is given in (4.5)

$$p_\theta(x_{1:n}|\theta) = f(s_1) \prod_{t=1}^{n-1} f_\theta(s_{t+1}|s_t, a_t)h_\theta(a_t|s_t, \theta) \tag{4.5}$$

where $p_\theta(x_{1:n})$ is the trajectory density function with an initial state $f(s_1)$. In a finite horizon RL setting, the performance of a certain policy, $J(\theta)$ is given by:

$$J_n(\theta) = \mathbb{E}_\theta[U(R_n(X_{1:n}))] = \int p_\theta(x_{1:n})U(R_n(x_{1:n}))dx_{1:n}. \tag{4.6}$$

The integral in (4.6) is intractable due to the fact that the distribution of the trajectory $p_\theta(x_{1:n})$ is either unknown or complex; thus calculation of $\nabla_\theta J_n(\theta)$ in (4.10)

is prohibitively hard. In order to deal with this problem, state-of-the-art policy gradient RL methods have been proposed by optimization techniques [2, 56–62], or exploring admissible regions of $J_n(\theta)$ via Bayesian approach [63, 64]. One of the very first methods in estimating $\nabla_\theta J_n(\theta)$in (4.10) is based on the idea of likelihood ratio methods. By taking the gradient in (4.6) we can formulate the gradient of the performance with respect to the parameter vector as:

$$\nabla J_n(\theta) = \int \nabla p_\theta(x_{1:n}) R_n(x_{1:n}) dx_{1:n}. \tag{4.7}$$

Next, by using (4.4) as well as the 'likelihood trick' identified by $\nabla p_\theta(x_{1:n}) = p_\theta(x_{1:n}) \nabla \log p_\theta(x_{1:n})$, where the product converted to summation according to logarithm's specifications, we can rewrite (4.7) as

$$\nabla J_n(\theta) = \int p_\theta(x_{1:n}) \left[\sum_{t=1}^{n} \nabla \log h_\theta(a_t|s_t) \right] R_n(x_{1:n}) dx_{1:n}. \tag{4.8}$$

Specifically, the goal of policy optimization in RL is to find optimal policy parameters θ^* that maximizes the expected value of some objective function of total reward R_n

$$\hat{\theta} = \arg\max_{\theta \in \Theta} J(\theta) \tag{4.9}$$

Although it is hardly ever possible to evaluate θ^* directly with this choice of Rn , maximization of $J(\theta)$ can be accomplished by policy gradient (PG) methods that utilize the steepest ascent rule to update their parameters at iteration i as:

$$\theta^{(i+1)} = \theta^{(i)} + \beta \nabla J_n(\theta^{(i)}). \tag{4.10}$$

where $x_t = (s_t, a_t)$. The objective of policy search in RL is to seek optimal policy parameters θ with respect to some expected performance of the trajectory $X_{1:n}$.

The RL methodology proposes to sample N trajectories from $p_\theta(x_{1:n})$ and then

performing Monte Carlo approximation over these N trajectories in order to approximate $\nabla_\theta J(\theta)$.

$$\nabla J_n(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{t=1}^{n} \nabla \log h_\theta(a_t^{(i)}|s_t^{(i)}) \right] R_n(x_{1:n}^{(i)}). \tag{4.11}$$

One of the RL methods that has successfully been applied in the domain of robotics which suit for the large-dimensional continuous state spaces, is a gradient based method called Episodic Natural Actor Critic (eNAC) which is extensively discussed in [65]. In this method $\nabla_\theta J(\theta)$ is calculated by solving a regression problem as:

$$\nabla_\theta J(\theta) = \left(\psi^T \psi\right)^{-1} \psi^T R\left(x_{1:n}\right) \tag{4.12}$$

where ψ is the gradient of the logarithm of the parameterized policy and is calculated for each iteration of the algorithm as:

$$\psi^{(i)} = \sum_{t=1}^{N} \nabla \log h_\theta \left(a_t^{(i)}|s_t^{(i)}\right) \tag{4.13}$$

A control policy which best represents the action selection strategy (calculation of the control signal) in robotics problems is usually a Gaussian probability distribution which takes into account the stochasticity of the system:

$$h_\theta(a|s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(a - \theta^T s)^2}{2\sigma^2} \right\}. \tag{4.14}$$

where μ is the vector of parameters to be learned. As a result, the gradient of this policy can easily be calculated as:

$$\nabla_\theta \log \pi\left(a|s, \theta\right) = \frac{a - s}{\sigma^2} s \tag{4.15}$$

The resulting strategy for the gradient based policy search then can be illustrated schematically in Figure 4.1 If the policy generates a reference trajectory, a controller

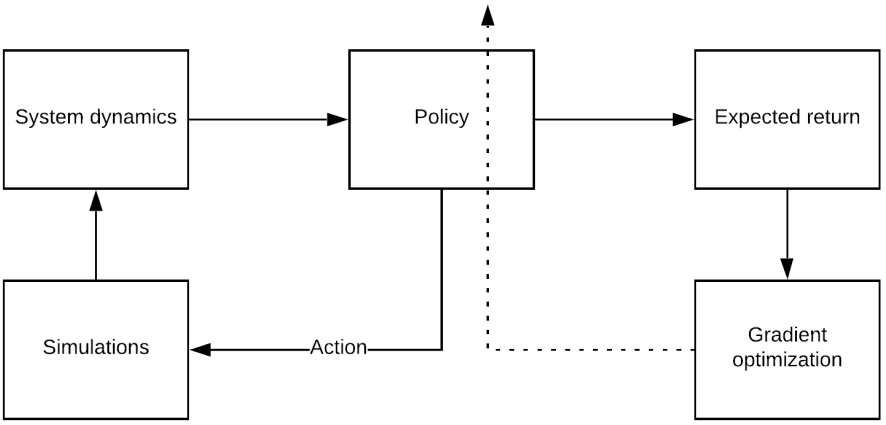


FIGURE 4.1: Gradient based policy search strategy

is required to map this trajectory (and the current state) to robot control commands (typically torques or joint angle velocity commands). This can be done for instance with a proportional-integral-derivative (PID) controller or a linear quadratic tracking (LQT) controller. The parameters of this controller can also be included in θ, so that both the reference trajectory and controller parameters are learned at the same time. By doing so, appropriate gains or forces for the task can be learned together with the movement required to reproduce the task.

4.2 Bayesian Learning via MCMC

In this section, we will discuss the benefits of the Bayesian method which concentrates on the control problem in Markov decision processes (MDP) with continuous state and action spaces in finite time horizon. The proposed method is a RL based policy search algorithm which uses Markov Chain Monte Carlo (MCMC) approaches. These methods are best applicable for complex distributions where sampling are difficult to achieve. The scenario here is to use risk-sensitivity notion, where a multiplicative expected total reward is employed to measure the performance, rather

than the more common additive one [28]. Using a multiplicative reward structure facilitates the utilization of Sequential Monte Carlo methods which are able to estimate any sequence of distributions besides being easy to implement. The advantage of the proposed method over PG algorithms, is to be independent of gradient computations. Consequently, it is safe from being trapped in local optima. Compared to PG when it comes to calculation of the performance measure $J(\theta)$, in Bayesian approach by considering the fact that $J(\theta)$ is hard to calculate due to intractability of the integral in (4.6), we establish an instrumental probability distribution $\pi(\theta)$ which is easy to take samples from and try to draw samples from this distribution without any need to calculate gradient information, as we did in PG RL algorithms.

The novelty of the presented approach is due to a formulation of the policy search problem in a Bayesian inference where the expected multiplicative reward, is treated as a pseudo-likelihood function. The reason for taking $J(\theta)$ as an expectation of a multiplicative reward function is the ability to employ unbiased lower variance estimators of $J(\theta)$ compare to methods that utilize a cumulative rewards formulation which lead in estimates with high variance. Instead of trying to come up with a single optimal policy, we cast the problem into a Bayesian framework where we treat $J(\theta)$ as if it is the likelihood function of θ. Combined with an uninformative prior $\mu(\theta)$ this leads to a pseudo-posterior distribution for the policy parameter. We then aim to target a quasi-posterior distribution and draw samples from via applying MCMC which is constructed as

$$\pi_n(\theta) \propto \mu(\theta) J_n(\theta)$$

MCMC methods are a popular family of methods used to obtain samples from complex distributions. Here, the distribution of our interest is $\pi(\theta)$, which is indeed hard to calculate expectations with respect to or generate exact samples from. MCMC methods are based on generating an ergodic Markov chain $\{\theta^{(k)}\}_{k\geq 0}$, starting from

the initial $\theta^{(0)}$¡, which has the desired distribution, in our case $\pi(\theta)$, as its invariant distribution. Arguably the most widely used MCMC method is the Metropolis–Hastings (MH) method where a parameter is recommended as a candidate value which is being derived from a proposal density as $\theta' \sim q(\theta'|\theta)$ Afterwards, the proposed θ' value is either accepted with a probability of $\alpha(\theta, \theta') = \min\{1, \rho(\theta, \theta')\}$ and the new parameter is updated as $\theta^{(k)} = \theta'$ or the proposed θ' is rejected and the value of new parameter does not change i.e. $\theta^{(k)} = \theta^{k-1}$. Here $\rho(\theta, \theta')$ is an acceptance ratio defined as:

$$\rho(\theta, \theta') = \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\pi(\theta')}{\pi(\theta)} = \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\mu(\theta')J(\theta')}{\mu(\theta)J(\theta)} \tag{4.16}$$

Because of difficulty in calculation of $J(\theta)$, computing this ratio is prohibitively hard. Despite this fact, one can select samples from $\pi(\theta)$ by applying SMC method to get an unbiased and non-negative estimate of $J(\theta)$ as demonstrated in Algorithm 2. The proposed method is summarized in Algorithm 1. Moreover, the detailed information of the proposed algorithm can be found in [28]. The other application of this Bayesian learning via MCMC is given [29], where estimates of the proportional and derivative(PD) controller coefficients using the proposed method for 2-DOF robotic system is determined.

Algorithm 1: Pseudo-marginal Metropolis-Hastings for RL

Input: Number of time steps n, initial parameter and estimate of expected performance $(\theta^{(0)}, \hat{J}^{(0)})$, proposal distribution $q(\theta'|\theta)$

Output: Samples $\theta^{(k)}$, $k = 1, 2, \ldots$

for *k = 1, 2, ...* **do**

Given $\theta^{(k-1)} = \theta$ and $\hat{J}^{(k-1)} = \hat{J}$, sample a proposal value $\theta' \sim q(\theta'|\theta)$.

Obtain an unbiased estimate $\hat{J}'$ of $J(\theta')$ by using Algorithm 2

Accept the proposal and set $\theta^{(k)} = \theta'$ and $\hat{J}^{(k)} = \hat{J}'$ with probability $\min\{1, \hat{\rho}(\theta, \theta')\}$ where

$$\hat{\rho}(\theta, \theta') = \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\mu(\theta')}{\mu(\theta)} \frac{\hat{J}'}{\hat{J}},$$

otherwise reject the proposal and set $\theta^{(k)} = \theta$ and $\hat{J}^{(k)} = \hat{J}$.

end

Algorithm 2: Simplified SMC algorithm for an unbiased estimate of $J(\theta)$

Input: Policy θ, number of time steps n, discount factor γ

Output: Unbiased estimate of $\hat{J}$

Start with $\hat{J}_0 = 1$.

for $t = 1, \ldots, n$ **do**

Sample $x_t \sim p(x_t|\theta)$ using (4.5)

Calculate $W_t = e^{\gamma^{t-1} r(x_t)}$.

Update the estimate: $\hat{J}_t = \hat{J}_{t-1} \times W_t$ **return** $\hat{J}$.

end

Chapter 5

Control Methodology

We aim to build a structure that can learn the internal system dynamics of the VAWT with all nonlinearities and observed wind speed profiles. This chapter presents the required Reinforcement Learning (RL) states and actions, radial basis function neural network (RBFNN) controller structure and explanation of the training stages of an MCMC Bayesian learning algorithm to obtain a proper MCMC controller for dealing with real wind profiles.

The proposed application of Reinforcement Learning is the optimization the instantaneous generator load current I_L to maximize the energy output and satisfy the conditions of the electrical constraints over a time horizon. In order to achieve this in the simplest form, we use RBFNN as a controller in order to calculate reference load current ($I_{L_{ref}}$).

The energy output (E) that we want to maximize can be computed by integrating power output (P) over a specific time period as:

$$E = \int_0^t P dt \tag{5.1}$$

The reference maximum energy output is obtained from the integration of the optimal aerodynamic power, P^*, which is the power that can be generated by the rotor when the power coefficient is kept at its maximum value, C_p^*, continuously:

$$E^* = \int_0^t P^* dt \tag{5.2}$$

where E^* is optimal energy output amount. Finally, we can calculate error for the energy which is defined as the difference between the energy output and the reference one as:

$$e = E^* - E \tag{5.3}$$

Furthermore, the derivative of error ($\dot{e}$) is defined as:

$$\dot{e} = e\frac{d}{dt} \tag{5.4}$$

The state space of the learning agent, S, of wind turbine model is comprised of one continuous component which is error dot, $s_t = (\dot{e})$. The action space A, which is reference load current (I_L), is one-dimensional and continuous, as well. The current state of the agent is defined according to the previous state and the current action as $S_t = G(S_{t-1}, A_t)$ where the corresponding relation $G : S \times A \to S$ is a deterministic function. In addition, we must add V_L and I_L constrains for actual system power electronic parts. The output voltage and current of generator is bounded by the minimum and maximum limits

$$V_{min} \leq V_L \leq V_{max} \tag{5.5}$$

$$I_{min} \leq I_L \leq I_{max} \tag{5.6}$$

5.1 Radial basis function neural network

The control policy here is provided by a Radial Basis Function Neural Network (RBFNN) which is used for implementing the controller to calculate the reference load current ($I_{L_{ref}}$). An RBFNN is shown in Figure 5.1 where inputs are $x_i, i = 1, 2, \ldots, n$ and output is $y = F(x, \theta)$, and m is the number of hidden nodes.

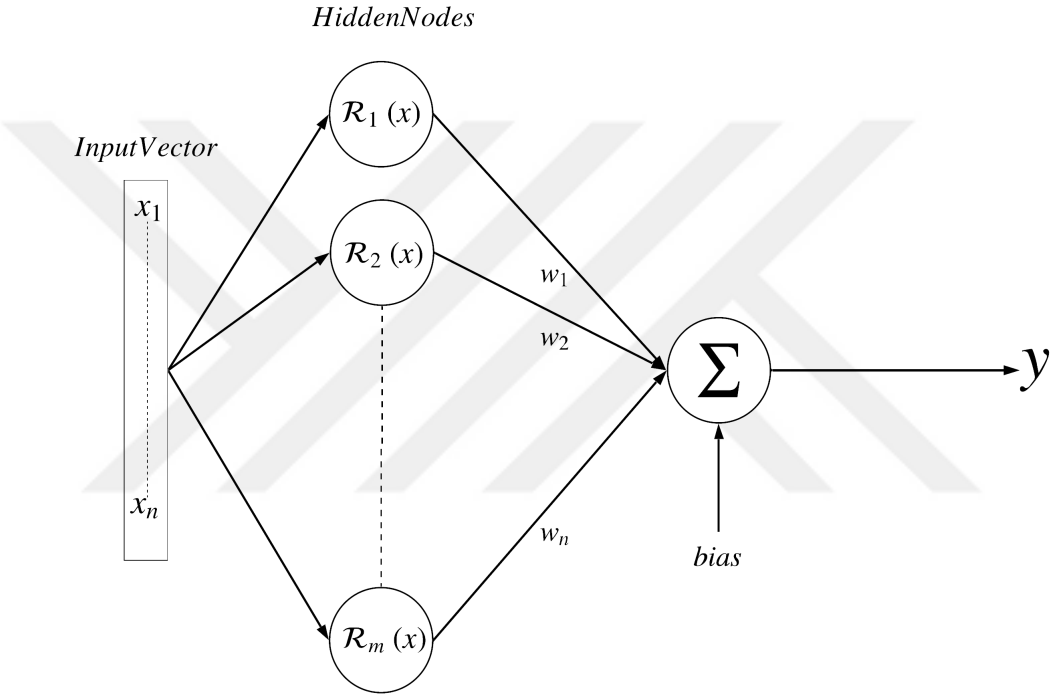


FIGURE 5.1: Radial basis function neural network

The output equation of the RBFNN in Figure 5.1 is denoted as:

$$y = F(x, \theta) = \sum_{i=0}^{n_r} w_i \mathscr{R}_i(x) + bias \tag{5.7}$$

where $\mathscr{R}_i(x)$ is the output of i^{th} hidden node, bias is a scale parameter and weights are represented by $w = [w_1 w_2 \dots w_m]$. The receptive field $\mathscr{R}_i(x)$ is defined as:

$$\mathscr{R}_i(x) = exp\left(-\frac{\|(x - c_i)\|^2}{2b_i^2}\right) \tag{5.8}$$

where c is the center of each RBF and b is the corresponding standard deviation for each RBF.

$$c = \begin{bmatrix} c_{11} & \dots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{n1} & \dots & c_{nm} \end{bmatrix} \tag{5.9}$$

$$b = \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix}^T \tag{5.10}$$

Detailed information about RBFNN can be found in [66]. c_{ii} parameters are predefined coefficients for the simplicity of reinforcement learning model. The center matrix of RBFNN hidden nodes c is given as follow:

$$c = \begin{bmatrix} 4.66 & 5.99 & 7.32 & 8.65 & 9.98 & 11.31 \\ -8.33 & -5 & -1.67 & 1.66 & 4.99 & 8.32 \\ 0.83 & 02.49 & 4.15 & 5.81 & 7.47 & 9.13 \\ 3.32 & 9.96 & 16.6 & 23.24 & 29.88 & 36.52 \\ 5 & 11 & 17 & 23 & 29 & 35 \\ -4.998 & -3 & -1.002 & 0.996 & 2.994 & 4.992 \end{bmatrix} \tag{5.11}$$

where those center parameters are extracted from every RBFNN input signal working interval. Moreover, the bias is selected as 3.5, because it is meaningful for reasonable wind speed interval(6m/s - 12m/s) even if hidden nodes stay zero.

In order to achieve learning system dynamics of VAWT and all possible wind speed profiles, RBFNN inputs are defined as in Table 5.1. Wind speed(U_w) and derivative of wind speed($\dot{U_w}$) are selected for perceiving wind speed and/or change of wind

speed. Load current(I_L), load voltage(V_L), PMSG rotor angular speed(ω_r) and derivative of PMSG rotor angular speed($\dot{\omega_r}$) are added RBFNN input space in order to give network VAWT internal states.

TABLE 5.1: Description of RBFNN Inputs.

RBFNN Input Number	Input Symbol	Input Description
x_1	U_w	Wind Speed
x_2	$\dot{U_w}$	Derivative of Wind Speed
x_3	I_L	Load Current
x_4	V_L	Load Voltage
x_5	ω_r	PMSG Rotor Speed
x_6	$\dot{\omega_r}$	Derivative of PMSG Rotor Speed

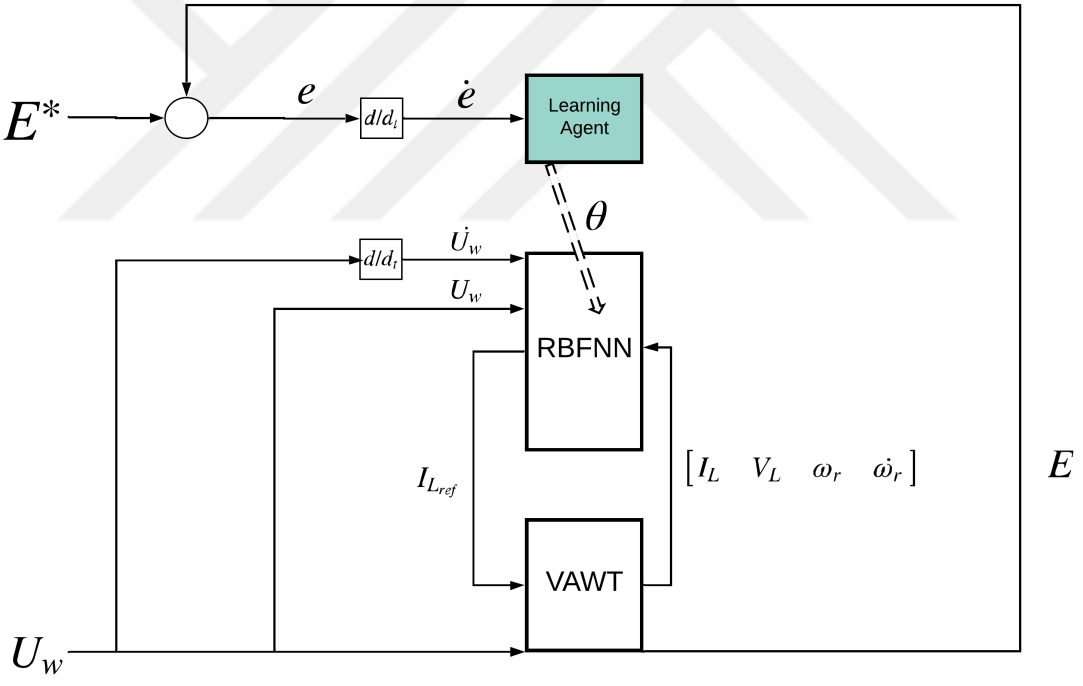


FIGURE 5.2: Radial basis function neural network control block diagram

The parameters of this nonlinear controller(RBFNN) itself will be learned by using Bayesian learning via MCMC method. This learning method will facilitate not only response to realistic wind profiles, but also the whole model of the system.

Therefore, our proposed method will be able to learn the optimal value of $I_{L_{ref}}$ for the probable wind conditions not just for a known wind profile. After establishing the framework for RBFNN controller, by using Bayesian learning via MCMC method methods, will learn the parameters of RBFNN controller. The block diagram of the control methodology is depicted in Figure 5.2. The learned parameters θ, which consists of weights and standard deviations of RBFNN controller, are updated in each learning iteration.

5.2 MCMC Bayesian Learning Algorithm Training Method

In this section, we describe how the policy parameters can be trained with progressively more complex wind speed patterns. We start with the initial parameter set θ_{S0}, the proposed learning system is trained with step wind profile, sinusoidal wind and finally realistic wind profile to obtain parameter sets θ_{S1}, θ_{S2}.

The nature of Bayesian Reinforcement Learning via MCMC is that the learning iterations start with a random parameter set θ_{S0}, which indicates initial policy parameters. For the first stage of training, step wind reference is selected as start point of MCMC, because it is a basic pattern which requires rotor energy management strategy. As a result of stage 1 of MCMC training pattern, we obtain the MCMC controller, which can work under step wind profile, with θ_{S1} parameters. After learning step wind, the second stage of training aims to obtain a controller that can respond to a variable speed wind pattern. Therefore, the reference signal of training second stage is sinusoidal wind that has close frequency to realistic wind. The second stage of MCMC training pattern start with θ_{S1} parameters as a initial policy parameters of MCMC learning iterations. As a result of stage 2 of MCMC

training pattern, MCMC controller, which is trained by sinusoidal wind, has θ_{S2} policy parameters as an outcome. Off-line part of MCMC training pattern finishes after obtaining θ_{S2} that can deal with realistic wind profiles. After off-line part of MCMC training, we proposed on-line learning with real wind, which implies that MCMC controller can continue learning in an actual VAWT installation, using only a small microprocessor to improve its control performance under local wind conditions. The MCMC training pattern is summarized in the schematic diagram in Figure 5.3.

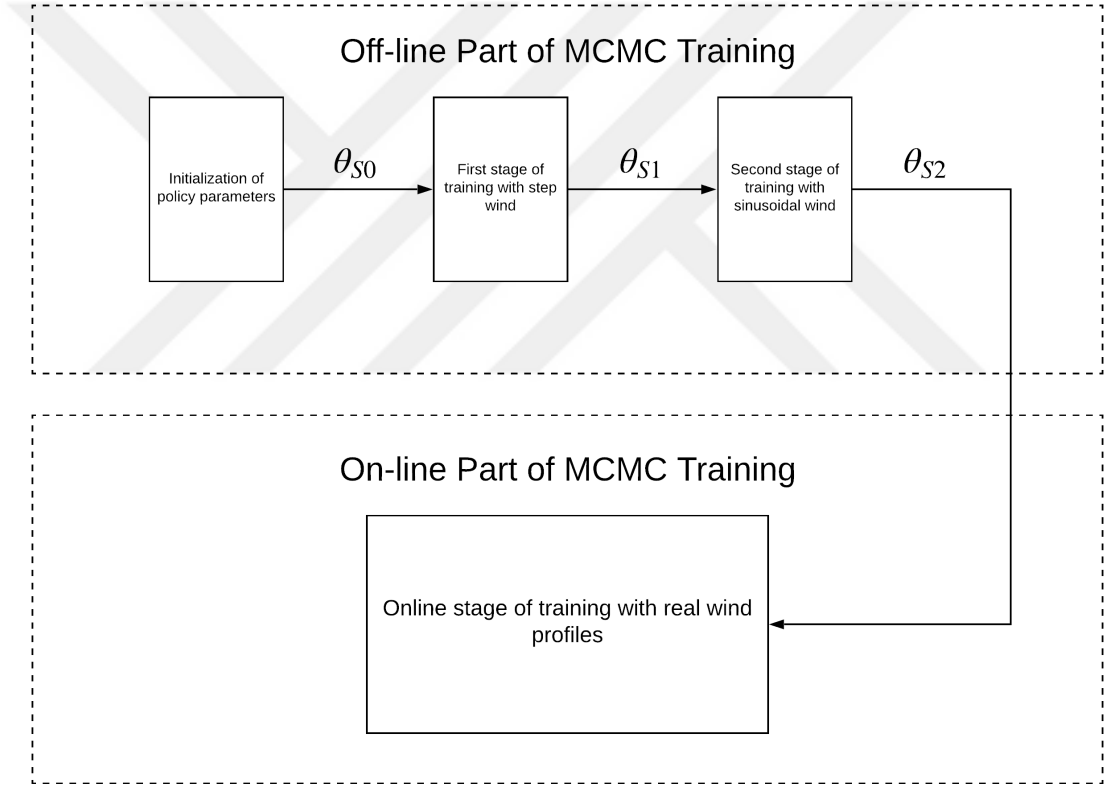


FIGURE 5.3: MCMC training pattern schematic diagram.

5.2.1 Parameters of The Learning Method

The initial parameters and the reward function structure of the MCMC algorithm are determined precisely for initialization. The reward function of MCMC Bayesian learning algorithm is defined as $r(s_t) = -s_t^T Q s_t$, where Q is 10^5 and s_t is $\dot{e}$. In this study the reward structure is defined as average reward, so the discount factor for the overall reward function R_n is $\gamma = 1$. Gaussian random walk is defined with $q(\theta'|\theta) = \mathscr{N}(\theta'; \theta, \Sigma_q)$, where diagonal covariance matrix Σ_q is $diagonal\left(\begin{bmatrix} 1 & \dots & 1 \end{bmatrix}_{1x12}\right)$. The prior distribution of policy is $\mathscr{N}\left(0; diagonal\left(\begin{bmatrix} 10000 & \dots & 10000 \end{bmatrix}_{1x12}^T\right), \Sigma_q\right)$. The essence of MCMC Bayesian learning algorithm is not knowing our prior $\mu(\theta)$, it is a challenging issue in finding optimal control policy. The sampling time of VAWT dynamics is 1ms. Also, RBFNN structure has already been given in section 5.1.

Chapter 6

Simulation Results

This chapter presents simulation results for the MCMC training parts and the comparison between the MCMC controller and a MPPT controller.

6.1 First stage of training

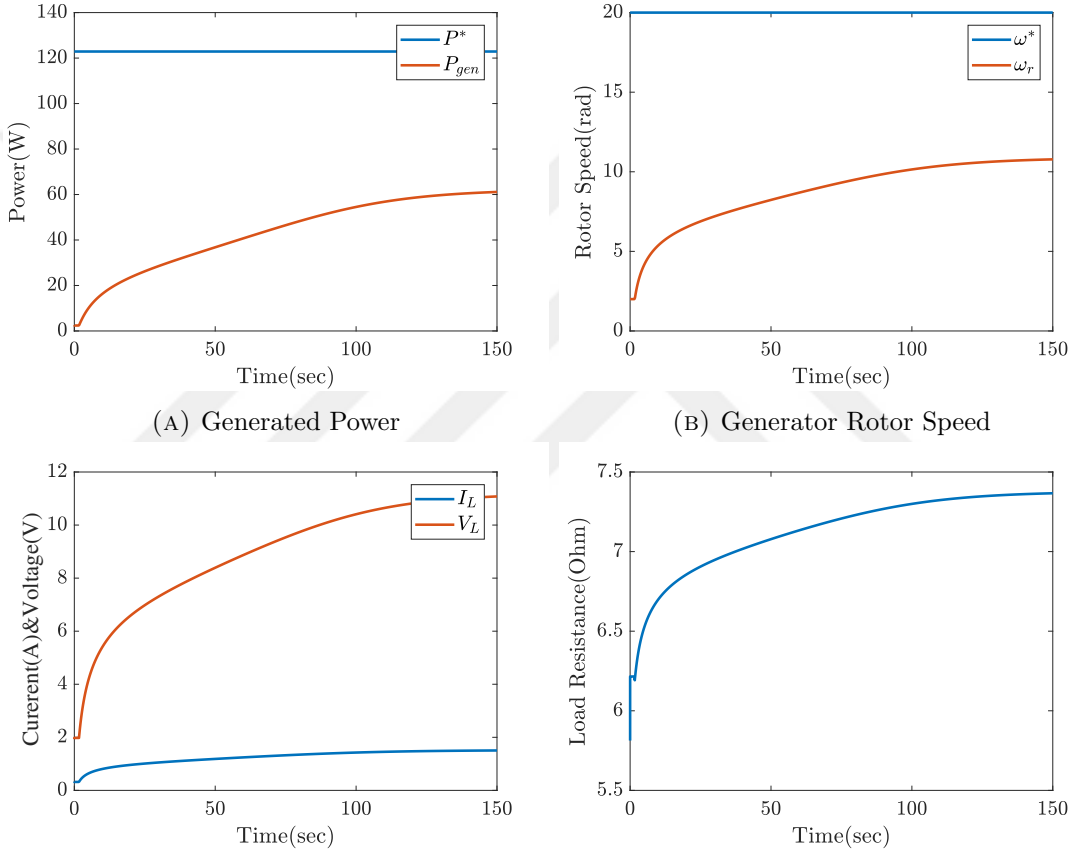
The reference signal of first training is step wind, is equals to 8 m/s, since the studied VAWT works in the wind speed range of 6m/s to 12m/s. Then, simulation time set to 150 second to observe transient behavior of VAWT. After this selection, initial policy parameters (θ_{S0}) have to be determined to create initial distribution of MCMC Bayesian learning algorithm. The policy parameters θ consist of RBFNN hidden node standard deviations and weights. θ_{S0} is defined as follows;

$$\theta_{S0} = \begin{bmatrix} b_{S0} & w_{S0} \end{bmatrix}$$

$$b_{S0} = \begin{bmatrix} 20 & 20 & 20 & 20 & 20 & 20 \end{bmatrix}$$

$$w_{S0} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{6.1}$$

where b_{S0} as initial standard deviation matrix of RBFNN and w_{S0} as initial weight matrix of RBFNN. Initial weights are selected as close to zero and non-zero coefficient and initial standard deviation parameters are chosen as (6.1); however they should cover the vector space of hidden nodes.



(A) Generated Power

(B) Generator Rotor Speed

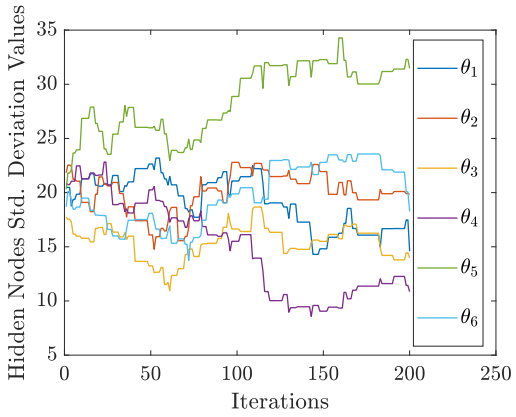
(C) Load Voltage & Load Current

(D) Load Resistance

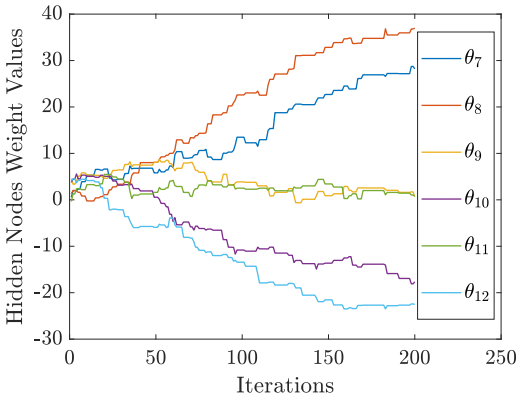
FIGURE 6.1: The MCMC controller, beginning of first stage training with θ_{S0} parameters, simulation result P, ω_r, V_L, I_L and R_L.

The 1^{st} iteration of MCMC first stage training simulation result with θ_{S0} is illustrated in Figure 6.1. As shown in Figure 6.1, the 1^{st} iteration of MCMC first stage training is not successful in terms of control. This is expected because of the random initial

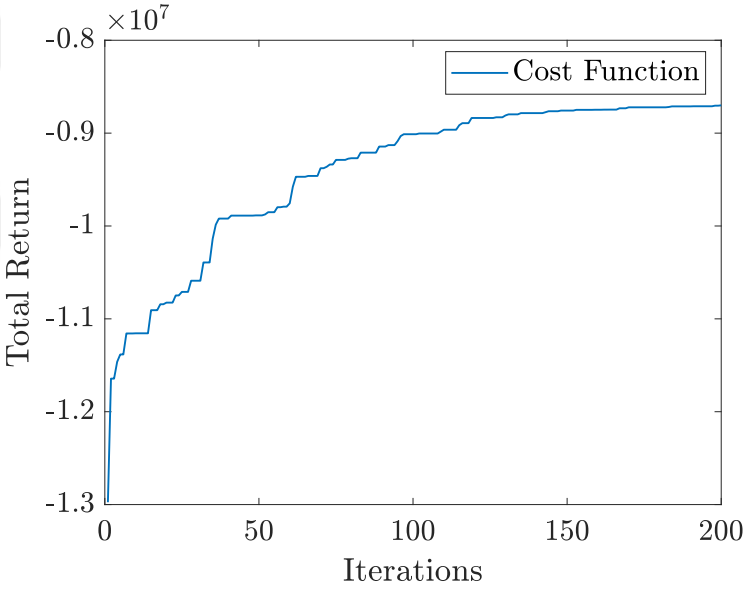
θ_{S0}. Figure 6.1 is intended as baseline and to represent the power of MCMC learning algorithm at the end of first stage training.



(A) First stage of training standard deviation.



(B) First stage of training weights.



(C) First stage of training total return.

FIGURE 6.2: Learning plots of MCMC first stage training.

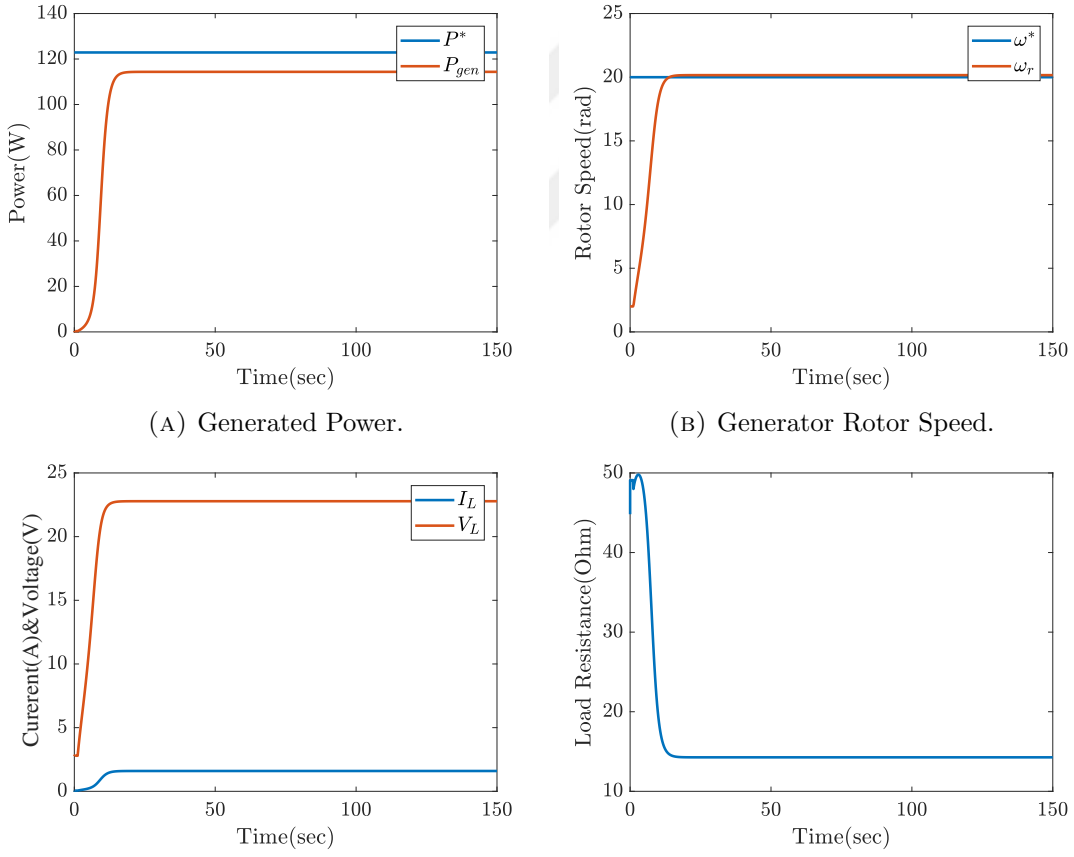
The evaluation of policy parameters during stage 1 training are shown in Figure 6.2 (A) and (B). The total return of first stage training in Figure 6.2 (C) has not converged to zero or steady state. It can be seen in Figure 6.2 (A) and (B) that policy parameters of stage 1 training has not converged to a specific value either. We stop learning early to prevent over-fitting, since we want to the MCMC controller to

learn dynamic wind speed as well. Therefore MCMC learning is interrupted at 200^{th} iteration. The resulting first stage training parameters θ_{S1}, which are a mean of the policy parameter values obtained in last quarter of 200 iterations, can handle step wind reference. The obtained first stage training parameters θ_{S1} are given below;

$$\theta_{S1} = \begin{bmatrix} b_{S1} & w_{S1} \end{bmatrix}$$

$$b_{S1} = \begin{bmatrix} 14.917 & 20.089 & 16.421 & 8.097 & 33.399 & 21.958 \end{bmatrix}$$

$$w_{S1} = \begin{bmatrix} 31.989 & 39.962 & 5.227 & -17.091 & 3.60 & -22.349 \end{bmatrix} \tag{6.2}$$



(A) Generated Power.

(B) Generator Rotor Speed.

(C) Load Voltage and Load Current.

(D) Load Resistance.

FIGURE 6.3: The MCMC controller, end of first stage training with θ_{S1} parameters, simulation result P, ω_r, V_L, I_L and R_L.

The simulation result of the proposed MCMC controller with θ_{S1} parameters is shown in Figure 6.3. The rotor speed of generator for the obtained MCMC controller, illustrated in Figure 6.3 (B), is close to optimal rotor speed of generator. Furthermore, the load current increase of the proposed MCMC controller, that is be shown in Figure 6.3 (C), is not aggressive which is better for achieving optimal control performance. Because it allows the rotor to speed up quicker to optimal value in the absence of load torque.

6.2 Second stage of training

The next step of training aims to obtain a controller which can cope with rapidly changing wind pattern. For this purpose, the reference signal is selected to a sinusoidal wind $(10 + 2sin(0.2t))$ in order to cover the studied VAWT working wind speed range is 6m/s to 12m/s. The reference signal illustrated in Figure 6.4.

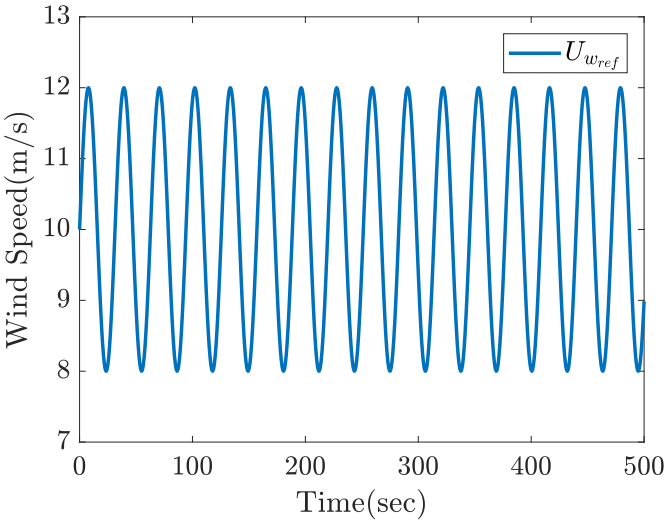
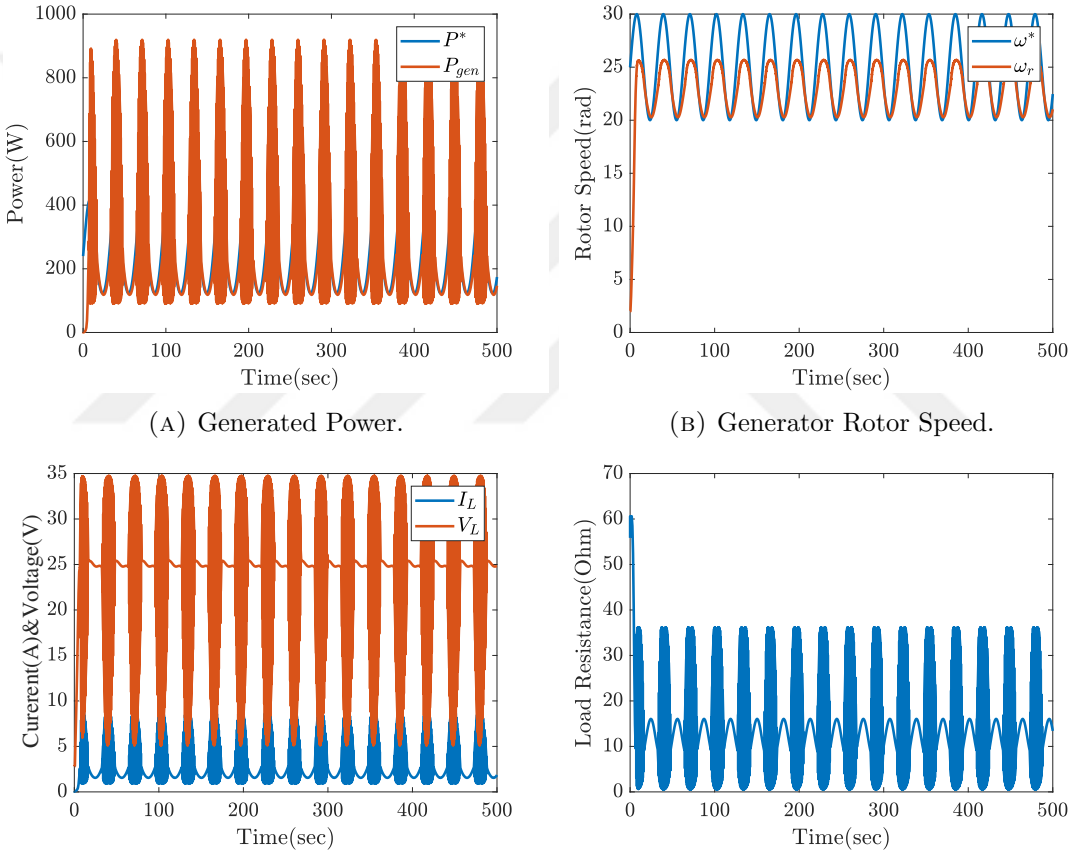


FIGURE 6.4: Stage 2 training wind speed reference.

The simulation time is set to 500 seconds for capturing sinusoidal behavior of wind speed reference. The initial policy parameter is set to first stage of training result θ_{S1} parameters.

In order to demonstrate the learning power of MCMC Bayesian learning algorithm, we present the performance of the generator with θ_{S1} policy parameters to the sinusoidal input speed in Figure 6.5 as a baseline to compare with MCMC controller with θ_{S2} that will be presented later. As shown in Figure 6.5 (D), the load resistance has



(A) Generated Power.

(B) Generator Rotor Speed.

(C) Load Voltage & Load Current.

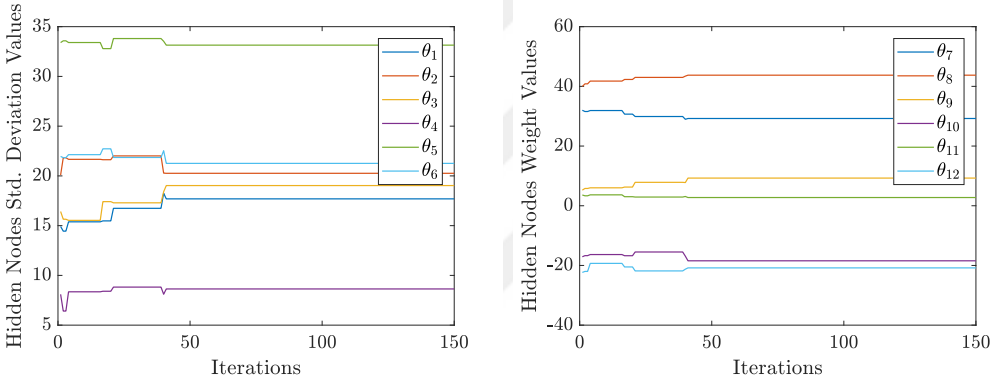
(D) Load Resistance.

FIGURE 6.5: The MCMC controller, beginning of second stage training with θ_{S1} parameters, simulation result P, ω_r, V_L, I_L and R_L.

some noise peaks, which leads to the load voltage and the output power have same peak structure, it is because MCMC controller with θ_{S1} parameters has not been trained sinusoidal reference. Significantly, derivative inputs of RBFNN parameters

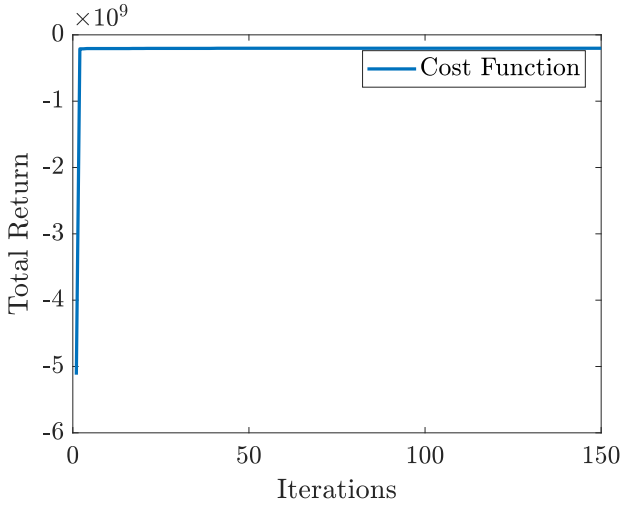
have not yet been trained properly to work under rapidly changing wind speeds. Bayesian Reinforcement Learning via MCMC improves policy parameters of stage 2 training to obtain the optimal resulted parameter θ_{S2}, presented next.

The policy parameters of stage 2 training evolution is shown in Figure 6.6 (A) and (B). The total return of second stage training, shown in Figure 6.6 (C), converges. It can be seen in Figure 6.6 (A) and (B) that MCMC controller has learned sinusoidal reference after approximately 40 iterations, since the policy parameters of stage 2 training do not change after approximately 40 iterations. This also implies that all proposed parameter values are rejected by MCMC learning algorithm.



(A) Second stage of training standard deviation.

(B) Second stage of training weights.



(C) Second stage of training total return.

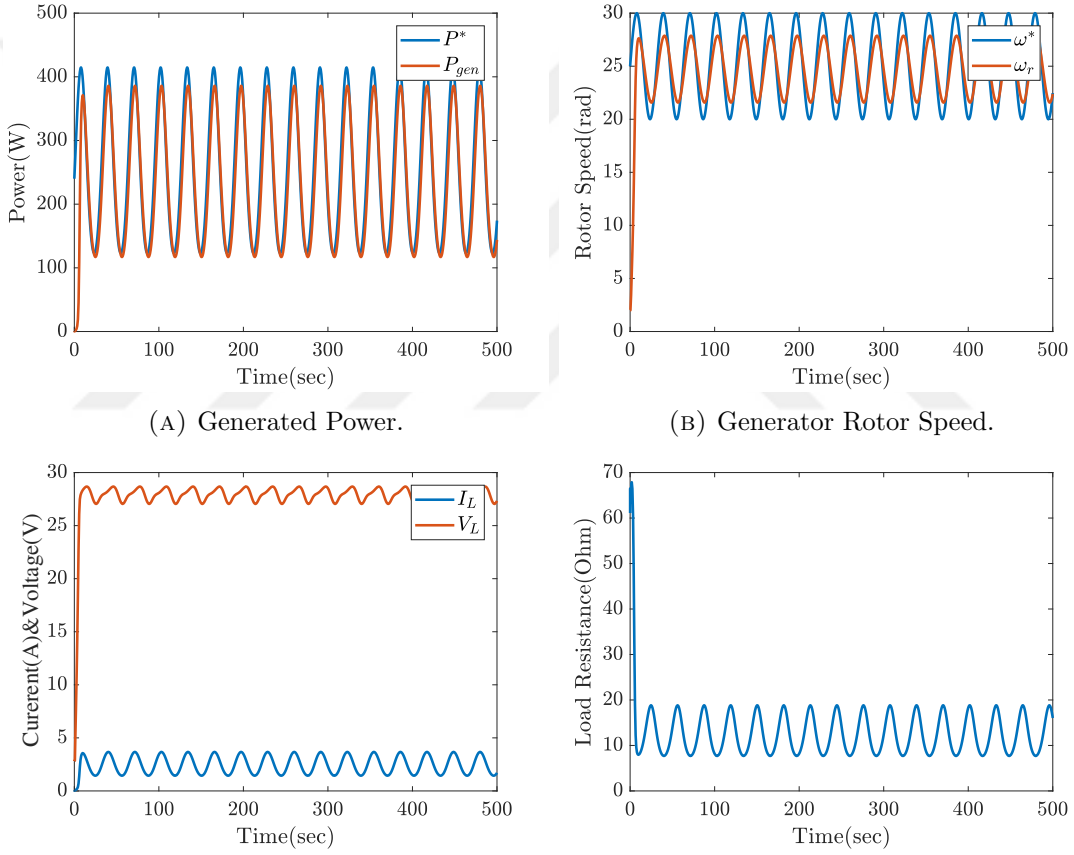
FIGURE 6.6: Learning plots of MCMC second stage training.

The MCMC learning iterations are continued until 150 to ensure MCMC controller performance. After MCMC stage 2 learning, the resulting θ_{S2} parameters of MCMC controller is given below;

$$\theta_{S2} = \begin{bmatrix} b_{S2} & w_{S2} \end{bmatrix}$$

$$b_{S2} = \begin{bmatrix} 17.84 & 20.31 & 18.9 & 8.59 & 33.26 & 21.18 \end{bmatrix}$$

$$w_{S2} = \begin{bmatrix} 29.19 & 43.73 & 9.28 & -18.45 & 2.74 & -20.82 \end{bmatrix} \tag{6.3}$$



(A) Generated Power.

(B) Generator Rotor Speed.

(C) Load Voltage and Load Current.

(D) Load Resistance.

FIGURE 6.7: The MCMC controller, end of second stage training with θ_{S2} parameters, simulation result P, ω_r, V_L, I_L and R_L.

The simulation results θ_{S2} for second stage training are shown in Figure 6.7. As a result, MCMC controller with θ_{S2} parameters follows optimal power output well, especially compared to results in Figure 6.5. The performance of the VAWT at the end of second stage training can be considered to be the state of generator as a product shipped from factory.

6.3 Comparison of Proposed Method with MPPT

In this section, the proposed MCMC Controller trained by Bayesian learning algorithm is compared to the commonly used maximum power point tracking (MPPT) algorithm for WECS in terms of control performance and energy output. The comparison is done in two steps; first step is that MCMC controller is compared to MPPT with step reference (10m/s) to illustrate start performance of control algorithms, second step is the comparison with realistic wind speed profile to show control performance and energy output. In a realistic setting, MCMC parameters are taken as θ_{S2}.

To better understand the comparison, MPPT algorithm will be explained. MPPT aims to maximize instantaneous power generation, which is a greedy approach for WECT. The detailed explanation of MPPT algorithm can be found in [16, 17]. For this study, two different MPPT controllers, shown at Table 6.1, are defined for comparison to MCMC controller. $mppt_2$ has faster convergence speed to optimal rotor speed than $mppt_1$ under fixed wind speed, yet $mppt_1$ is more successful under realistic wind profiles due to variable wind speeds.

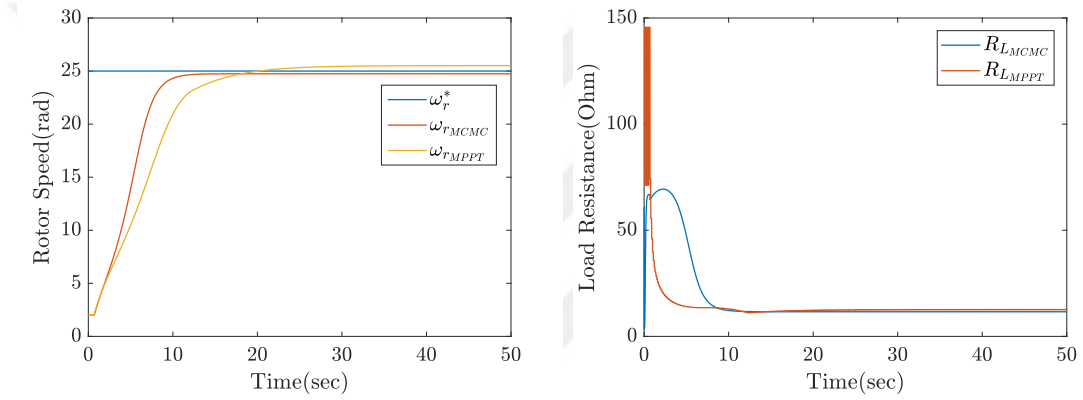
Table 6.1: The MPPT controllers description.

	Sampling Time	ΔI_{ref}
$mppt_1$	0.1s	0.02A
$mppt_2$	0.1s	0.01A

6.3.1 Step wind speed reference performance comparison

Our aim for this test is to compare starting performance of the control methods. Drastic wind speed change is a major problem for WECS, step wind reference is an easy way to mimic these drastic changes.

The magnitude of step wind speed is 10 m/s. The simulation time is set to 50 seconds due to observe transient behavior of compared controllers. MCMC controller, which uses θ_{S2}, is compared to $mppt_1$ controller given in Table 6.1.



(A) Generator Rotor Speed.

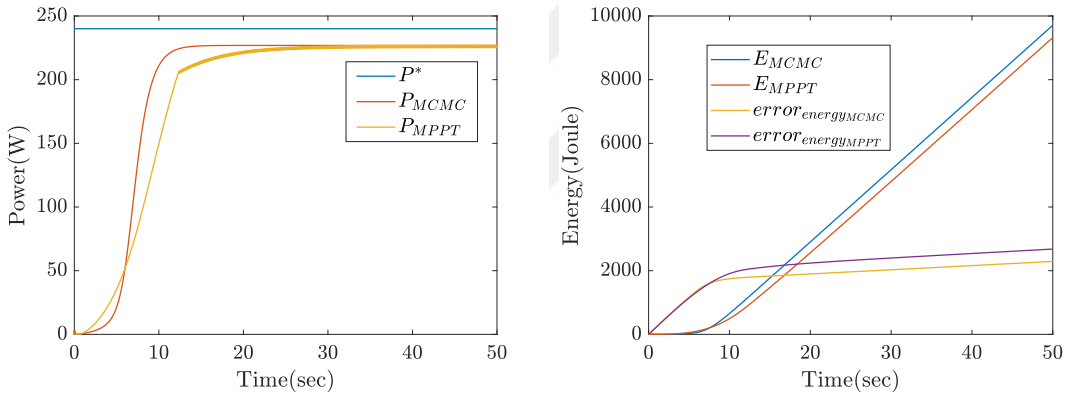
(B) Load Resistance.

(C) Load Voltage.

(D) Load Current.

FIGURE 6.8: The MCMC controller with θ_{S2} parameters and $mppt_1$ simulation results under step wind speed profile (10m/s).

The simulation results ω_r, R_L, V_L and I_L comparison MCMC controller to MPPT under step wind speed profile (10m/s) are shown in Figure 6.8. Correct ω_r is the most crucial parameter for reaching optimal C_p value. The optimal generator rotor speed (ω_r^*) is calculated by (3.1) where U_w is given. Figure 6.8 (A) illustrates ω_r^* response, MCMC controller ω_r response ($\omega_{r_{MCMC}}$) and MPPT ω_r response ($\omega_{r_{MPPT}}$). It can be easily seen that $\omega_{r_{MCMC}}$ is closer to ω_r^*, although MCMC controller and $mppt_1$ controller have similar performance under 10m/s step wind. On the other hand the most remarkable difference between these two controllers is that MCMC controller has relatively smooth load current increase as shown in Figure 6.8 (D). This increase contributes $\omega_{r_{MCMC}}$ to reach the optimal ω_r^* value in a fast way. MCMC controller also maintains a power output that closer to optimal than MPPT.



(A) Power output.

(B) Energy output and error of energy.

FIGURE 6.9: The MCMC controller with θ_{S2} parameters and $mppt_1$ power and energy output under step wind speed profile (10m/s).

The proposed MCMC controller allows ω_r to reach the optimal by initially keeping the electrical load small. This allows the rotor to speed up to ω_r^* quicker, thus output more energy. On the other hand, MPPT controller immediately increases the electrical load greedily then decreases as shown in Figure 6.8 (B). The load voltage response of these two controllers are given in Figure 6.8 (C). $\omega_{r_{MPPT}}$ is higher than ω_r^* value. It is consistent with Figure 6.8 (A) because V_L is proportional to ω_r.

Figure 6.9 (A) demonstrates the power output for MCMC controller and $mppt_1$. The power outputs are approximately equal, yet the transient behaviors of these two controller are different thanks to differences between $R_{L_{MCMC}}$ and $R_{L_{MPPT}}$. The generated total energy for given simulation time is shown in Figure 6.9 (B). E_{MCMC} and E_{MPPT} have similar trends but E_{MCMC} peforms better during the transient. For rapidly and continuously changing wind patterns the difference will become significant.

6.3.2 Real wind speed reference performance comparison

Our aim for this test is to demonstrate real wind profile performance of the proposed control method. In real wind turbine applications, wind can be modeled by the sum of a variable speed and noise element. Fluctuating wind speed change is a major problem for WECS, the proposed method has to deal with this type of winds. It is important to emphasize that realistic wind speed signal used for this test is obtained by Simulink Aerospace Toolbox wind generator block. The reference wind speed signal is illustrated in Figure 6.10.

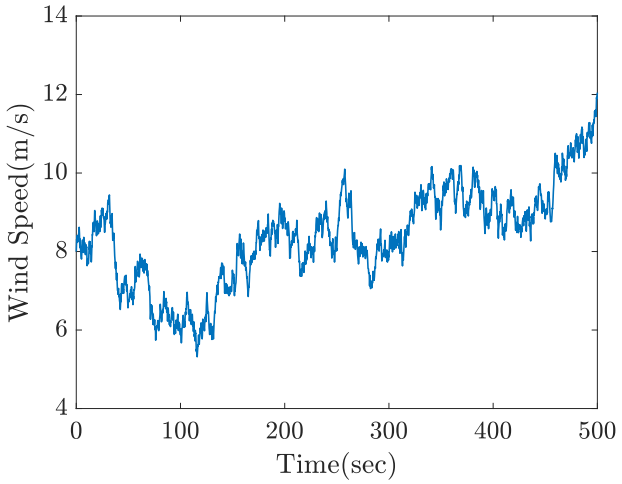


FIGURE 6.10: Real Wind Speed References for Comparison of MCMC and MPPT.

The simulation time is set to 500 seconds to show energy output performance of controllers. MCMC controller, which uses θ_{S2}, compares to $mppt_1$ controller given in Table 6.1.

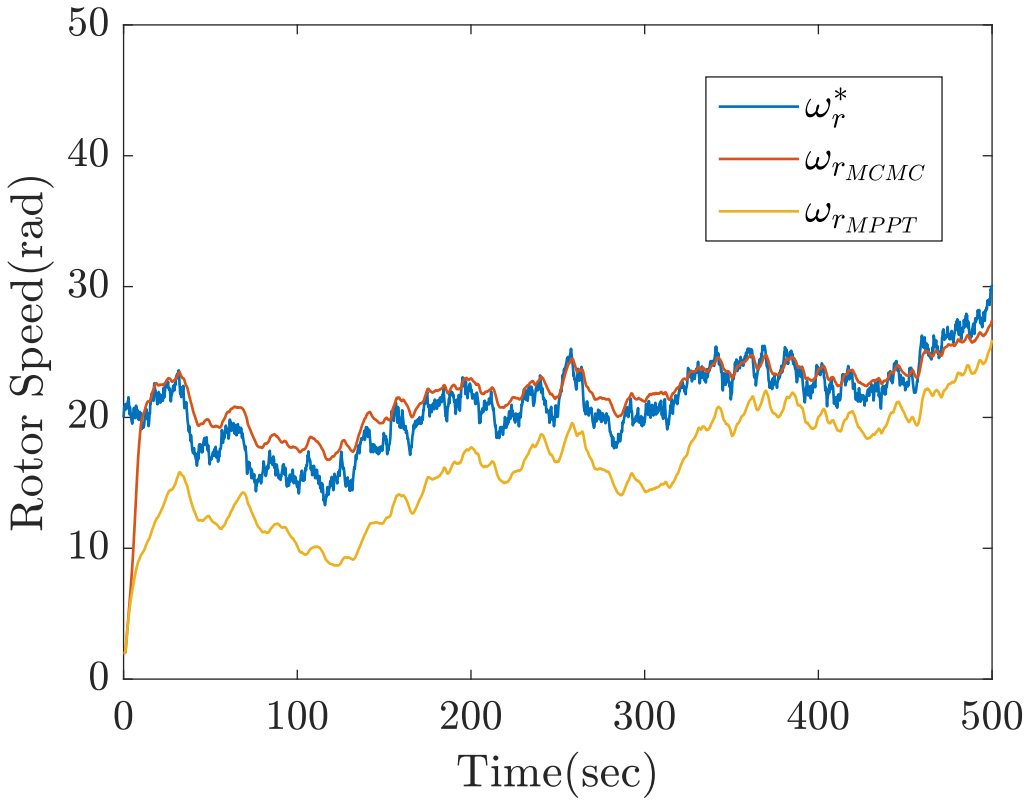


FIGURE 6.11: The MCMC controller with θ_{S2} parameters and $mppt_1$ generator rotor speed under realistic wind speed profile.

As mentioned before, correct ω_r is the most crucial parameter for reaching optimal C_p value. The optimal generator rotor speed (ω_r^*) is calculated by (3.1) where U_w is given by Figure 6.10. Figure 6.11 illustrates the responses of MCMC controller ($\omega_{r_{MCMC}}$) and MPPT ($\omega_{r_{MPPT}}$). It can be easily seen in Figure 6.11 that MCMC controller works closer to optimal generator rotor speed from around 8 s. However, MPPT controller never reaches optimal rotor speed (ω_r^*) according to Figure 6.11.

The electrical load response for these two controllers can be found in Figure 6.12. It can be seen that $R_{L_{MCMC}}$ reacts to wind speed changes better than $R_{L_{MPPT}}$.

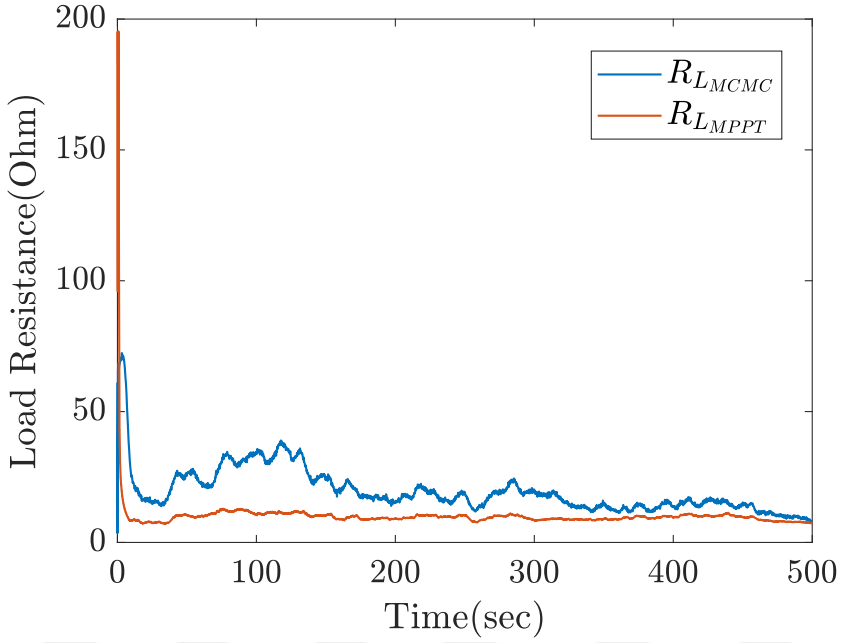
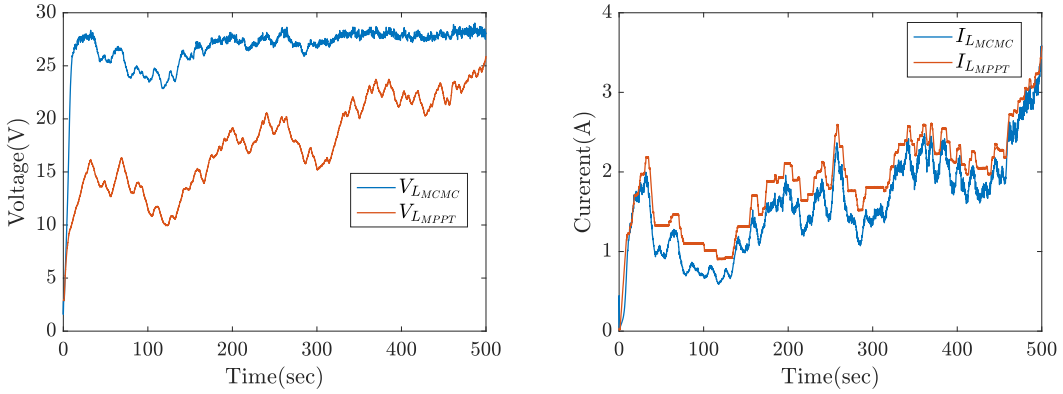


FIGURE 6.12: The MCMC controller with θ_{S2} parameters and $mppt_1$ load resistance under realistic wind speed profile.

The load current and load voltage responses can be observed in Figure 6.13.



(A) Load voltage.

(B) Load current.

FIGURE 6.13: The MCMC controller with θ_{S2} parameters and $mppt_1$ load voltage and load current under realistic wind speed profile.

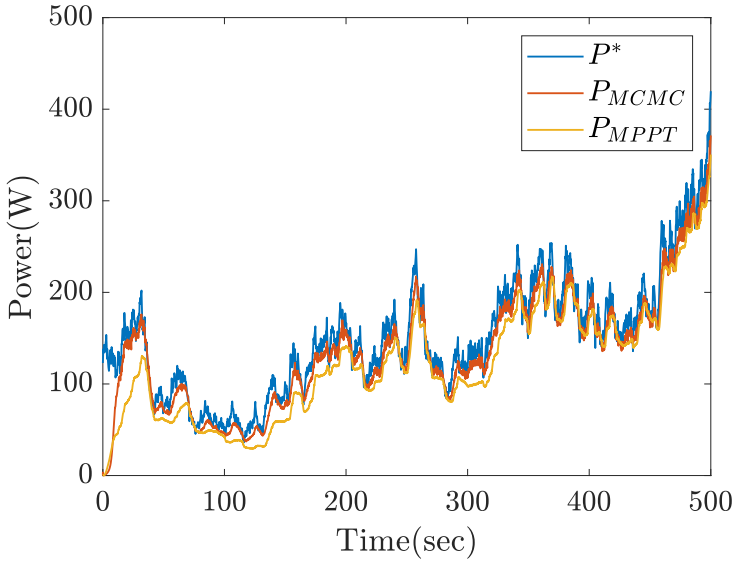


FIGURE 6.14: The MCMC controller with θ_{S2} parameters and $mppt_1$ power output under realistic wind speed profile.

MCMC controller follows the optimal power (P^*) more closely than MPPT. The total energy output for given simulation time is shown in Figure 6.15. It can be seen that E_{MCMC} is higher than E_{MPPT}.

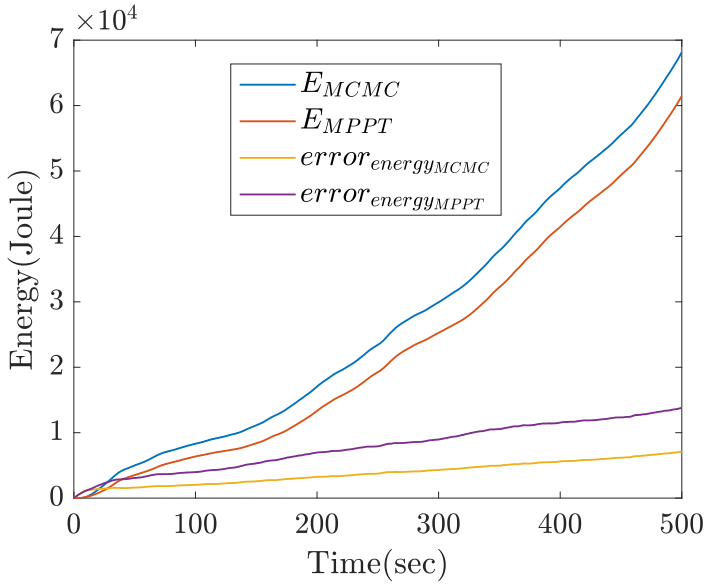


FIGURE 6.15: The MCMC controller with θ_{S2} parameters and $mppt_1$ energy output under realistic wind speed profile.

RL approaches have statistical background, the results must be statistically consistent; Therefore this simulation has been done 10 times with 10 different realistic wind profiles in order to demonstrate how consistent the two controllers react in realistic wind performance. The results for 3 controllers, which are MCMC, $mppt_1$ and $mppt_2$, are listed in Table 6.2. Corresponding mean values and standard deviation values for these 3 controllers are calculated in Table 6.3.

TABLE 6.2: Experiment results difference of energy output from optimal (Joule).

Experiment No	**MCMC**	$mppt_1$	$mppt_2$
1	8117.7	19844.5	12511.7
2	8254.7	9933.5	32410.5
3	8671.9	9654.2	11929.7
4	5917.6	35108.9	8024.1
5	7726.5	33325.8	38513.7
6	7695.6	31365.4	45678.7
7	7757.4	45690.7	8003.5
8	7787.8	57123.1	7802.2
9	8114.1	22900.9	13489.9
10	8117.4	17834.3	10523.4

It must be noted that $mppt_1$ and $mppt_2$ have almost 4 times higher mean of energy output error than MCMC controller. Furthermore, it can be seen in Table 6.3 that $mppt_1$ and $mppt_2$ have large standard deviation values, it means that wind speed changes affect MPPT controllers performance significantly, yet MCMC controller has consistent control performance under diverse real wind profiles.

TABLE 6.3: Experiment results means and standard deviations of difference of energy output from optimal (Joule).

	Mean	**Standard Deviation**
MCMC Controller	7816	732
$mppt_1$	28278	15309
$mppt_2$	23889	27411

Chapter 7

Conclusion and Future Works

This section presents conclusion and contribution of this thesis as well as possible future research direction via this thesis.

7.1 Conclusion

In this thesis, it has been shown that the proposed Bayesian Reinforcement Learning via MCMC method is able to learn VAWT system dynamics and wind profiles. Performance has been improved by progressing learning from step wind to sinusoidal wind reference in order to deal with realistic wind profiles. It has been shown that the proposed method has superior performance compared to the common MPPT method in terms of total energy output. MCMC controller has shown 89% efficiency while MPPT has shown 78% efficiency for realistic wind patterns. Also, real wind simulations have been performed 10 times with 10 different wind speed profiles to demonstrate the consistency of the MCMC controller. MCMC has similar performance with respect to wind speed changes, while MPPT performance has more variation as shown in Table 6.2.

7.1.1 Contribution

Our proposed MCMC controller enables us to directly apply the control signal to WECS via model learning of VAWT. Furthermore, the proposed MCMC method use an approach, which is based on data driven methods, takes into account all the nonlinearities of the plant as an implicit or explicit model. Bayesian learning algorithm with MCMC has powerful policy to represent instantaneous states of VAWT in order to calculate control signal for maximizing energy output. Unpredictable wind flows always become a significant challenge for wind energy systems; therefore our proposed control strategy addresses this issue explicitly by learning unpredictable wind profiles and the model of the VAWT. The simulation results justify this claim because MCMC controller outperforms the most well-known control algorithm MPPT.

7.2 Future Works

The designed learning mechanism is not only off-line learning in factory but also on-line learning in the field, after the commissioning of the VAWT, since wind speeds vary according to location. Bayesian Reinforcement learning via MCMC provides on-line learning option with local winds to maximize energy generation in installation area of VAWT.

Bibliography

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: Adaptive computation and machine learning. MIT Press, 1998.

[2] V. Gullapalli, J. A. Franklin, and H. Benbrahim, "Acquiring robot skills via reinforcement learning," *IEEE Control Systems*, vol. 14, pp. 13–24, feb 1994.

[3] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Beijing, China), 2006.

[4] A. da Rosa, "Chapetr 1 - generalites," in *Fundamentals of Renewable Energy Processes (Third Edition)* (A. da Rosa, ed.), Boston: Academic Press, third edition ed., 2013.

[5] A. Özgün Önol, *Modeling, hardware-in-the-loop simulations and control design for a vertical axis wind turbine with high solidity*. Master's thesis, Sabancı University, Istanbul, 2016.

[6] U. Sancar, *Hardware-in-the-loop simulations and control designs for a vertical axis wind turbine*. Master's thesis, Sabancı University, Istanbul, 2015.

[7] M. Islam, S. Mekhilef, and R. Saidur, "Progress and recent trends of wind energy technology," *Renewable and Sustainable Energy Reviews*, vol. 21, pp. 456 – 468, 2013.

[8] A. Tummala, R. K. Velamati, D. K. Sinha, V. Indraja, and V. H. Krishna, "A review on small scale wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1351 – 1371, 2016.

[9] M. Lasheen, F. Bendary, A. Sharaf, and H. M. El-Zoghby, "Maximum power point tracking of a wind turbine driven by synchronous generator connected to an isolated load using genetic algorithm," *Journal of Electrical Engineering*, vol. 15, p. 21, 04 2015.

[10] R. Kot, M. Rolak, and M. Malinowski, "Comparison of maximum peak power tracking algorithms for a small wind turbine," *Mathematics and Computers in Simulation*, vol. 91, pp. 29 – 40, 2013. ELECTRIMACS 2011 - PART II.

[11] J. Hui and A. Bakhshai, "A new adaptive control algorithm for maximum power point tracking for wind energy conversion systems," in *2008 IEEE Power Electronics Specialists Conference*, pp. 4003–4007, June 2008.

[12] D. Biswas, S. S. Sahoo, P. M. Tripathi, and K. Chatterjee, "Maximum power point tracking for wind energy system by adaptive neural-network based fuzzy inference system," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, pp. 1–6, March 2018.

[13] H. Fathabadi, "Maximum mechanical power extraction from wind turbines using novel proposed high accuracy single-sensor-based maximum power point tracking technique," *Energy*, vol. 113, pp. 1219 – 1230, 2016.

[14] M. Pucci and M. Cirrincione, "Neural mppt control of wind generators with induction machines without speed sensors," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 37–47, Jan 2011.

[15] S. M. R. Kazmi, H. Goto, H. Guo, and O. Ichinokura, "A novel algorithm for fast and efficient speed-sensorless maximum power point tracking in wind energy conversion systems," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 29–36, Jan 2011.

[16] E. Koutroulis and K. Kalaitzakis, "Design of a maximum power tracking system for wind-energy-conversion applications," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 486–494, April 2006.

[17] D. Zammit, C. S. Staines, A. Micallef, M. Apap, and J. Licari, "Incremental current based mppt for a pmsg micro wind turbine in a grid-connected dc microgrid," *Energy Procedia*, vol. 142, pp. 2284 – 2294, 2017. Proceedings of the 9th International Conference on Applied Energy.

[18] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335 – 348, 1989.

[19] A. Körber and R. King, "Model predictive control for wind turbines," *European Wind Energy Conference and Exhibition 2010, EWEC 2010*, vol. 2, 01 2010.

[20] P. F. Odgaaard and T. G. Hovgaard, "Selection of references in wind turbine model predictive control design," *IFAC-PapersOnLine*, vol. 48, no. 30, pp. 333 – 338, 2015. 9th IFAC Symposium on Control of Power and Energy Systems CPES 2015.

[21] A. Onol, U. Sancar, A. Onat, and S. Yesilyurt, "Model predictive control for energy maximization of small vertical axis wind turbines," 10 2015.

[22] C. Bottasso, P. Pizzinelli, C. Riboldi, and L. Tasca, "Lidar-enabled model predictive control of wind turbines with real-time capabilities," *Renewable Energy*, vol. 71, pp. 442 – 452, 2014.

[23] A. Jain, G. Schildbach, L. Fagiano, and M. Morari, "On the design and tuning of linear model predictive control for wind turbines," *Renewable Energy*, vol. 80, pp. 664 – 673, 2015.

[24] D. Song, J. Yang, M. Dong, and Y. H. Joo, "Model predictive control with finite control set for variable-speed wind turbines," *Energy*, vol. 126, pp. 564 – 572, 2017.

[25] A. Bektache and B. Boukhezzar, "Nonlinear predictive control of a dfig-based wind turbine for power capture optimization," *International Journal of Electrical Power and Energy Systems*, vol. 101, pp. 92 – 102, 2018.

[26] A. El Kachani, E. M. Chakir, A. A. Laachir, T. Jarou, and A. Hadjoudja, "Nonlinear model predictive control applied to a dfig-based wind turbine with a shunt apf," in *2016 International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 369–375, Nov 2016.

[27] Y. P. Pane, S. P. Nageshrao, J. Kober, and R. Babuška, "Reinforcement learning based compensation methods for robot manipulators," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 236 – 247, 2019.

[28] V. T. Aghaei, A. Onat, and S. Yıldırım, "A markov chain monte carlo algorithm for bayesian policy search," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 438–455, 2018.

[29] V. T. Aghaei, A. Ağababaoğlu, A. Onat, and S. Yıldırım, "Bayesian learning for policy search in trajectory control of a planar manipulator," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0240–0246, Jan 2019.

[30] J. L. Wattes, A. J. S. Dias, A. P. S. Braga, P. P. Praça, A. U. Barbosa, and D. S. Oliveira, "A continuous actor-critic maximum power point tracker applied

to low power wind turbine systems," in *2016 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 3231–3236, March 2016.

[31] W. Meng, Q. Yang, and Y. Sun, "Reinforcement learning controller for variable-speed wind energy conversion systems," in *Proceedings of the 33rd Chinese Control Conference*, pp. 8877–8882, July 2014.

[32] C. Wei, Z. Zhang, W. Qiao, and L. Qu, "Reinforcement-learning-based intelligent maximum power point tracking control for wind energy conversion systems," *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 6360–6370, Oct 2015.

[33] C. Wei, Z. Zhang, W. Qiao, and L. Qu, "An adaptive network-based reinforcement learning method for mppt control of pmsg wind energy conversion systems," *IEEE Transactions on Power Electronics*, vol. 31, pp. 7837–7848, Nov 2016.

[34] W.-M. Lin and C.-M. Hong, "Intelligent approach to maximum power point tracking control strategy for variable-speed wind turbine generation system," *Energy*, vol. 35, no. 6, pp. 2440 – 2447, 2010. 7th International Conference on Sustainable Energy Technologies.

[35] A. Rezvani, A. Esmaeily, H. Etaati, and M. Mohammadinodoushan, "Intelligent hybrid power generation system using new hybrid fuzzy-neural for photovoltaic system and rbfnsm for wind turbine in the grid connected mode," *Frontiers in Energy*, vol. 13, pp. 131–148, Mar 2019.

[36] J. de Jesús Rubio, "Interpolation neural network model of a manufactured wind turbine," *Neural Computing and Applications*, vol. 28, pp. 2017–2028, Aug 2017.

[37] C.-H. Lin, "Recurrent modified elman neural network control of permanent magnet synchronous generator system based on wind turbine emulator," *Journal of Renewable and Sustainable Energy*, vol. 5, no. 5, p. 053103, 2013.

[38] Y. Bao, H. Wang, and J. Zhang, "Adaptive inverse control of variable speed wind turbine," *Nonlinear Dynamics*, vol. 61, pp. 819–827, Sep 2010.

[39] Y. Liu, R. J. Patton, and S. Shi, "Wind turbine load mitigation using mpc with gaussian wind speed prediction," in *2018 UKACC 12th International Conference on Control (CONTROL)*, pp. 32–37, Sep. 2018.

[40] L. Li, Z. Yuan, and Y. Gao, "Maximization of energy absorption for a wave energy converter using the deep machine learning," *Energy*, vol. 165, pp. 340 – 349, 2018.

[41] A. Wilson, A. Fern, and P. Tadepalli, "Using trajectory data to improve bayesian optimization for reinforcement learning," *J. Mach. Learn. Res.*, vol. 15, pp. 253–282, Jan. 2014.

[42] M. Castronovo, *Offline Policy-search in Bayesian Reinforcement Learning*. Phd thesis, University of Liege, Liège, Belgium, 2016.

[43] M. Schillinger, B. Hartmann, P. Skalecki, M. Meister, D. Nguyen-Tuong, and O. Nelles, "Safe Active Learning and Safe Bayesian Optimization for Tuning a PI-Controller," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5967–5972, 2017.

[44] R. R. Duivenvoorden, F. Berkenkamp, N. Carion, A. Krause, and A. P. Schoellig, "Constrained Bayesian Optimization with Particle Swarms for Safe Adaptive Controller Tuning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11800–11807, 2017.

[45] M. E. Daachi, T. Madani, B. Daachi, and K. Djouani, "A radial basis function neural network adaptive controller to drive a powered lower limb knee joint orthosis," *Applied Soft Computing Journal*, vol. 34, pp. 324–336, 2015.

[46] D. Kosic, "Fast Clustered Radial Basis Function Network as an adaptive predictive controller," *Neural Networks*, vol. 63, pp. 79–86, 2015.

[47] V. A. Akpan and G. D. Hassapis, "Nonlinear model identification and adaptive model predictive control using neural networks," *ISA Transactions*, vol. 50, no. 2, pp. 177–194, 2011.

[48] M. Aliasghary, A. Naderi, H. Ghasemzadeh, and A. Pourazar, "Design of radial basis function neural networks controller based on sliding surface for a coupled tanks system," *Proceedings - 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2011*, vol. 1, pp. 8–12, 2011.

[49] S. Datta and A. K. Roy, "Radial basis function neural network based STATCOM controller for power system dynamic stability enhancement," in *India International Conference on Power Electronics, IICPE 2010*, pp. 1–5, IEEE, 2011.

[50] Y. Zhan, Y. Chen, and B. Zhang, "A radial basis function Neural Networks based space-vector PWM controller for voltage-fed inverter," in *2014 16th European Conference on Power Electronics and Applications, EPE-ECCE Europe 2014*, pp. 2–10, 2014.

[51] A. Rawat and M. J. Nigam, "Comparison between adaptive linear controller and radial basis function neurocontroller with real time implementation on magnetic levitation system," in *2013 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*, pp. 1–4, IEEE, 2013.

[52] J. Lin and R. J. Lian, "Hybrid self-organizing fuzzy and radial basis-function neural-network controller for gas-assisted injection molding combination systems," *Mechatronics*, vol. 20, no. 6, pp. 698–711, 2010.

[53] K. Lo, Y. Chen, and Y. Chang, "Mppt battery charger for stand-alone wind power system," *IEEE Transactions on Power Electronics*, vol. 26, pp. 1631–1638, June 2011.

[54] B. Sareni, A. Abdelli, X. Roboam, and D. Tran, "Model simplification and optimization of a passive wind turbine generator," *Renewable Energy*, vol. 34, no. 12, pp. 2640 – 2650, 2009.

[55] D. Tran, B. Sareni, X. Roboam, and C. Espanet, "Integrated optimal design of a passive wind turbine system: An experimental validation," *IEEE Transactions on Sustainable Energy*, vol. 1, pp. 48–56, April 2010.

[56] H. Kimura and S. Kobayashi, "Reinforcement learning for continuous action using stochastic gradient ascent," *The 5th International Conference on Intelligent Autonomous Systems*, 1988.

[57] C. J. Maddison, D. Lawson, G. Tucker, N. Heess, A. Doucet, A. Mnih, and Y. W. Teh, "Particle value functions," in *Workshop track - ICLR 2017*, 2017.

[58] M. Toussaint and A. Storkey, "Probabilistic inference for solving discrete and continuous state markov decision processes," in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, (New York, NY, USA), pp. 945–952, ACM, 2006.

[59] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7, pp. 1180 – 1190, 2008. Progress in Modeling, Theory, and Application of Computational Intelligenc.

[60] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pp. 1594–1601, 2005.

[61] H. J. Kappen, V. Gomez, and M. Opper, "Optimal control as a graphical model inference problem," *Machine Learning*, vol. 87, pp. 159–182, 2012.

[62] P. Dayan and G. Hinton, "Using Expectation-Maximization for reinforcement learning," *Neural Computation*, vol. 9, pp. 271–278, 1997.

[63] M. Hoffman, A. Doucet, N. D. Freitas, and A. Jasra, "Bayesian policy learning with trans-dimensional MCMC," in *Advances in Neural Information Processing Systems 20* (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), pp. 665–672, Curran Associates, Inc., 2008.

[64] D. Wingate, N. D. Goodman, D. M. Roy, L. P. Kaelbling, and J. B. Tenenbaum, "Bayesian policy search with policy priors," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 16–22, July 2011.

[65] J. R. Peters, *Machine learning of motor skills for robotics.* Phd thesis, University of Southern California, 2007.

[66] J. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*, pp. 339–362. 01 2013.