BAYESIAN REINFORCEMENT LEARNING WITH MCMC TO MAXIMIZE ENERGY OUTPUT IN HARDWARE-IN-THE-LOOP SIMULATIONS OF VERTICAL AXIS WIND TURBINE

by
USAMAH YAASEEN OSMAN

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

Sabancı University
June 2021

BAYESIAN REINFORCEMENT LEARNING WITH MCMC TO MAXIMIZE ENERGY OUTPUT IN HARDWARE-IN-THE-LOOP SIMULATION OF VERTICAL AXIS WIND TURBINE

Approved by:

Date of Approval: 28 June 2021

ABSTRACT

BAYESIAN REINFORCEMENT LEARNING WITH MCMC TOMAXIMIZE ENERGY OUTPUT IN HARDWARE IN THE LOOPSIMULATION OF VERTICAL AXIS WIND TURBINE

USAMAH YAASEEN OSMAN

MECHATRONICS M.Sc. THESIS, JUNE 2021

Thesis Supervisor: Assoc. Prof. Dr. AHMET ONAT

Abstract

Vertıcal Axıs Wınd turbınes(VAWT) are ideal for small-scale use in urban areas. The energy produced is optimized by controlling the wind speed to rotor tip speed ratio. The proposed control method which uses a neural network trained by Monte Carlo methods is run on a Hardware-in-the-Loop setup modelled after the VAWT. Once trained, the proposed method is compared to other commonly used control methods such as MPPT and SNC. It is found to produce satisfactory results with energy produced being more than MPPT all the time. It sometimes performs better than SNC.

ÖZET

DIKEY EKSENLI RUZGAR TURBINININ ENERJI CIKTISINI BUYUTMEK ICIN MZMC ILE BAYESCI PEKISTIRMELI OGRENME

USAMAH YAASEEN OSMAN

MEKATRONIKS YÜKSEK LİSANS TEZİ, HAZIRAN 2021

Tez Danışmanı: Doç. Dr. Ahmet Onat

Anahtar Kelimeler: Dinamik sistemlerde öğrenme yöntemleri, Koşullama ile öğrenme, Markov chain Monte Carlo yöntemleri, yüksek enerji verimlilikli rüzgar türbinleri.

Özet

Dikey Eksenli Rüzgar türbinleri(VAWT), kentsel alanlarda küçük ölçekli kullanım için idealdir. Üretilen enerji, rüzgar hızının rotor uç hızına oranı kontrol edilerek optimize edilir. Monte Carlo yöntemiyle eğitilmiş bir sinir ağını kullanan önerilen kontrol yöntemi, VAWT'den sonra modellenen bir Döngüde Donanım kurulumunda çalıştırılır. Eğitildikten sonra önerilen yöntem, MPPT ve SNC gibi yaygın olarak kullanılan diğer kontrol yöntemleriyle karşılaştırılır. Üretilen enerjinin her zaman MPPT'den fazla olması ile tatmin edici sonuçlar ürettiği bulunmuştur. Bazen SNC'den daha iyi performans gösterir

ACKNOWLEDGEMENTS

I am grateful for the immense support provided by my advisor Assoc. Prof. Dr. Ahmet Onat. His patience, encouragement and guidance have been invaluable in the completion of the thesis.

Assist. Prof. Dr. Vahid Tavakol Aghaei whose assistance and explanations made some complex problems easier to understand,

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

The popularity of renewable energy has been soaring as clean energy becomes a sustainable option, As of 2018, renewable energy sources made up 26.2 percent of generated electricity globally [1]. This is expected to further rise in the next few years.

Wind is at the forefront of viable options. Wind turbines offer unlimited power when placed in appropriate locations. They also bring with them a unique challenge in terms of control because of the uncertainty in wind speed. Wide variations of wind speed may occur within a short amount of time.

Vertical Axis Wind Turbines (VAWT), where the rotor axis is positioned in a vertical orientation, offer an advantage over the more popularly recognized Horizontal Axis Wind turbines (HAWT) in that they can be made in small sizes and offer more versatile mounting locations. They are also able to function using wind from all directions. They are the main focus in the thesis.

To harvest the maximum possible energy from the wind, appropriate control methods need to be applied. This is done to obtain the optimum amount of energy for every wind speed. Most control methods can determine the optimum obtainable power for a given wind speed fairly easily. The challenge with wind is that it changes rapidly and the performance of the wind turbine is mainly determined by how the control system reacts to the transients. A number of control methods exist including maximum power point tracker (MPPT), model predictive control (MPC) and non-linear methods including those which use neural networks.

In this thesis, machine learning methods are applied to the problem of load control of vertical axis wind turbines to maximize total energy (This is in contrast to instantaneous power maximization objective of many current methods). Machine learning methods are increasingly being applied to a wide variety of problems. Their general applicability to different problems makes them especially popular. The Monte Carlo Markov Chain (MCMC) methods are used to learn the parameters of a neural network which controls the electrical load connected to the output of the turbine. We verify the results not only through simulations, but also through hardware-in-the-loop exper-

iments. The performance of the neural network using parameters learned from MCMC is then compared to that of other existing methods including but not limited to the commonly used MPPT.

1.1 Thesis outline

The second chapter covers the progress made in the field of wind power and the application of reinforcement learning methods.The control methods analysed in the thesis including MPPT, Model Predictive Control(MPC) and the proposed control method are also introduced in this chapter. This is followed by Chapter 3, where a mathematical model of the VAWT is presented.

Chapter 4 gives a detailed explanation of the proposed control method. The control method structure is given with the specific algorithms used. Chapter 5 covers the Hardware-in-loop model used with the rapid control prototyping hardware dSPACE and Matlab Simulink framework. This covers the specific parts and validation of the performance. Parameters used for the experimental hardware are also described.

The specific results of HIL experiments are given and discussed in Chapter 6. Chapter 7 concludes the thesis with a brief discussion of future extensions and work that can be done based on the ideas in the thesis.

2.   Literature Survey & Background

With the drastic effects of climate change evident, renewable energy sources are becoming increasingly relevant. Wind energy is among the most promising options that are being explored. The most common structures associated with wind energy are shown in Fig. 2.1. Wind turbines are the most popular way of harvesting wind energy. Classified according to the axis of rotation, there can be vertical axis wind turbines (VAWT) and horizontal axis wind turbines (HAWT).
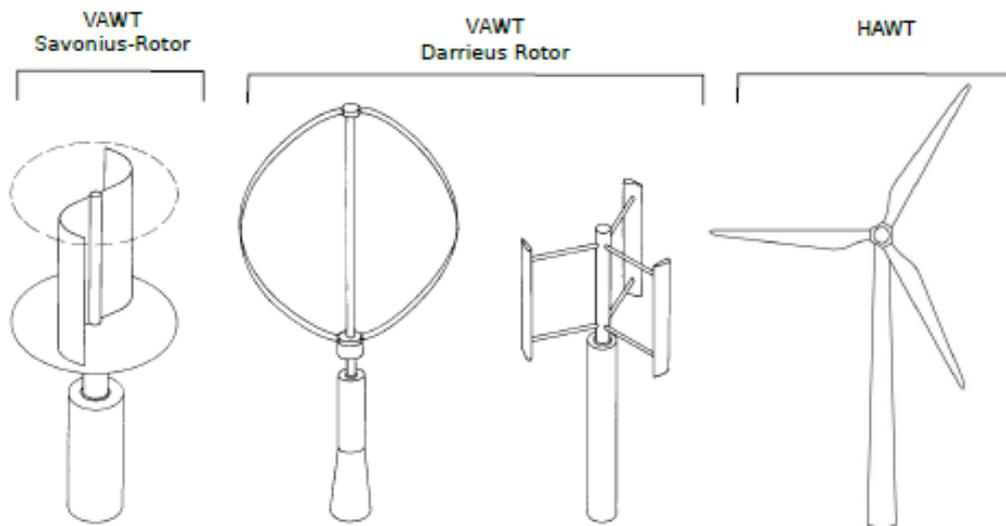
Figure 2.1 Types of Wind turbines

The wind energy device covered in the thesis is a Vertical Axis Wind turbine of the Darrieus type rotor. The rotor axis for the VAWT is vertical. As opposed to horizontal axis HAWT, which only utilize wind from certain directions, VAWT can use wind from all directions to generate energy. VAWTs are also smaller in size, hence ideal for use in the small-scale applications. This means they can be used in more diverse places. They are suitable for use even within the city as they produce less sound than HAWTs [2].

Control of wind turbines is is an important factor in their performance. Different

methods exist to achieve this. Variable speed control is used for the VAWTs due to their higher efficiency as compared to constant-speed control. In [3, 4] the general objective of control is to maximize the total energy produced by the VAWT system. This should be done by changing the the operating region of the system to adapt to existing wind conditions.

VAWT control involves regulating generator torque to get optimum energy from the wind. For every wind speed, there is an optimum rotor speed for the turbine. At the optimum rotor speed, the turbine is generating the maximum energy possible from the wind. This is subject to the physical limits of the system. VAWTs are controlled to balance speeding up with torque. Since the wind speed changes rapidly, the optimum rotor speed is never constant.

2.1 Common Control Algorithms for VAWT

Common control algorithms applied to VAWT control include MPPT, MPC, PID, predictive control and sliding mode control [5].

MPPT is one of the well-known control algorithms applied on a wide range of systems including the VAWT. MPPT operates by finding the optimal tip-speed ratio for any given generator rotor speed and wind speed pair by gradually changing the load operating point in each direction and determining the gradient of output power to increase it. While this method of control is suitable for maximizing instantaneous power, it does not necessarily lead to the maximum energy generation. This greedy approach to efficiency is not ideal for systems with fast changing operating point. This is especially true for the wind systems which fluctuate rapidly. MPPT approaches include standard MPPT algorithms [7, 8]. Adaptive MPPT algorithms [9] and fuzzy logic MPPT algorithms [10].

MPC is also a common algorithm used in wind systems. These outperform MPPT algorithms by allowing the instantaneous maximum point to be forgone in favour of total energy gains in the future [15]. Challenges include the computational expense for the method. It also requires the complete model of the system. Using it for small-scale applications of the VAWT would not be practical. The SNC algorithm was developed in [15] to deal with the drawbacks of MPC. Its main drawback is the prior model information required for good performance.

2.2 The proposed control method

A popular solution for the above-outlined limitations are Machine Learning (ML) methods. They have been applied on a wide range of robotic systems [11]. At the forefront of the currently used ML methods are the widely used Neural Networks (NN). Neural networks provide flexibility that other methods do not. The intricacies of the system need not be fully incorporated into the control algorithm. With the advancement in the processing power available, NNs are becoming very powerful tools in control. Their speed and relative ease of use makes them suitable for the VAWT system.

The NN brings with it the challenge of parameter tuning. The parameters will also vary with the wind conditions in the exact installation location of the VAWT. Even after initial tuning, the parameters will need to be re-tuned and further changed due to the obvious mechanical degradation that would occur over the lifetime of the system. A number of methods exist to find these parameters. Typical gradient based reinforcement learning methods could be used that would find the parameters through training. These run the risk of being trapped in a local optimum. They might also take a very long time to find the optimum due to the large search space for the NN parameters.

The MCMC methods are an ideal solution for finding appropriate parameters for the system. These methods are ideal for cases where samples are collected from complex distributions. The MCMC method used here was first explored in [12].

It when applied to a two link manipulator, promising results were obtained. The algorithm was then applied on a software simulation of the VAWT in [13]. It was found to perform better than the MPPT algorithm. To better validate the algorithm, there is a need to apply it experimentally on a physical system. This will give a better indication of how it would perform in a real-world setting. Hence, to further test the algorithm, we apply it on a HIL setup of the VAWT in this thesis. The particular setup used here was validated and tested in [14, 15]. HIL setups are a common testing method in vehicle systems and missile guidance systems [16, 17, 18, 19] and HIL has numerous advantages over software model based simulations [16, 20, 21]; they minimize the risk of faults in final testing phases in the field. Time and resources are saved on the testing and validation of designs. Mechanical conditions that might be complex to simulate such as friction could physically be created.

In systems with various components such as the VAWT (mechanical, electrical, electromagnetic), HIL is an effective tool to study the system from early on in terms of transient behaviour and dynamics [22]. The HIL setup is thus an effective tool for wind

turbine modelling [23, 24, 25, 26, 27, 28].

3.   Vertical Axis Wind Turbine and Control

Wind turbines consist of an aerodynamic rotor with blades and a generator. The rotor captures the wind power and converts it to mechanical rotation of the shaft. This mechanical energy is then converted to electricity. The mechanism for how this happens in the VAWT is explained in this chapter. The mechanics are presented in the first section and the intuition of control is given thereafter.

3.1 VAWT mechanics

The ratio of conversion of energy from wind to mechanical is defined by the power coefficient, C_p. This determines how much of the available wind power is actually captured. The value is limited by the Betz limit [6]. The Betz limit is the maximum possible power obtainable from the wind for conversion through aerodynamic design. This is equal to 0.59259. In practice, the C_p reaches a value much lower than that.

The power coefficient, C_p is expressed as a function of tip speed ratio, λ, which is given as;

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

where ω_r is the generator speed, R is the radius of the area swept by the VAWT and U_w is the wind speed.
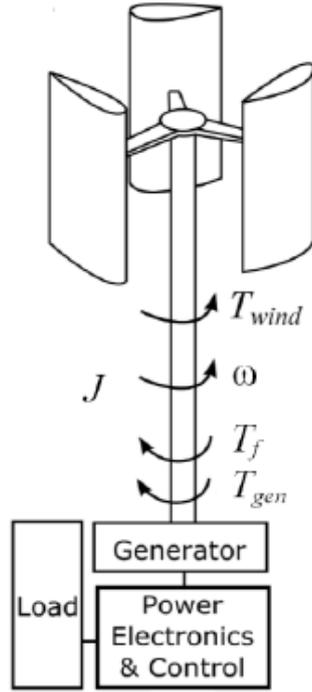
Figure 3.1 VAWT model

Generally, moving wind exerts a torque on the turbine. The blades start speeding up when the wind torque is higher than the sum of opposing torques such as the frictional torque and the generator torque. The main system equation central to the model is;

$$\frac{d\omega}{dt} = \frac{T_w - T_g - T_f}{J} \tag{3.2}$$

T_w is the wind torque generated by the blades, T_g is the torque from the generator and T_f is the torque due to the friction. The acceleration of the rotor shaft is thus affected by torques mentioned above. The one torque that can be influenced by control is the generator torque. We can control the turbine speed by controlling the generator torque. The generator torque can in turn be affected by the load current.

The Torque of the wind (T_w) and the wind power(P_w) are defined as;

$$T_w = \frac{P_w}{\omega} \tag{3.3}$$

$$P_w = C_p \rho L R U_w^3 \qquad (3.4)$$

where C_p is the power coefficient, ρ is the air density, L is the rotor height, R is the rotor radius and U_w is the wind velocity. C_p, representing the aerodynamic efficiency of the turbine is often expressed as a function of λ, the Tip Speed Ratio (TSR), determined as;

$$\lambda = \frac{\omega_r R}{U_w} \qquad (3.5)$$

In this study, a λ vs. $Cp(\lambda)$ curve from computational fluid dynamics simulations is used as shown in Fig. 3.2.
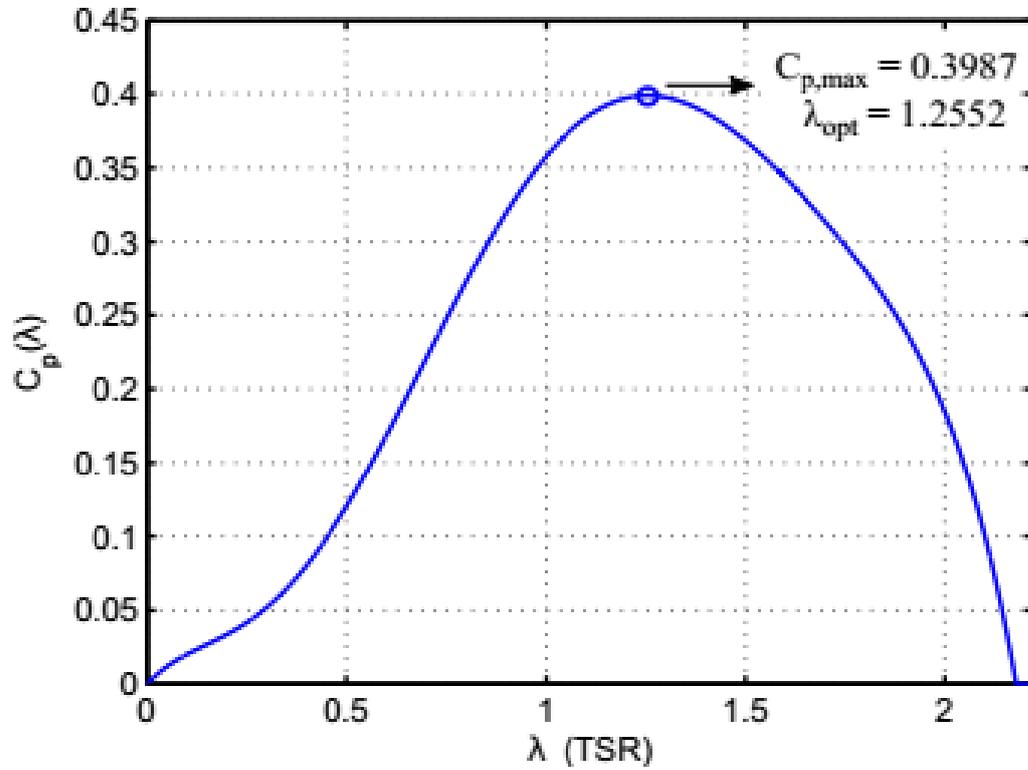
Figure 3.2 λ vs. $Cp(\lambda)$ curve

For our studies, the curve is approximated using a 6th order polynomial as follows;

$$C_p = p_1 \lambda^6 + p_2 \lambda^5 + p_3 \lambda^4 + p_4 \lambda^3 + p_5 \lambda^2 + p_6 \lambda \qquad (3.6)$$

The coefficient values are listed in Table 3.1

Table 3.1 C_p polynomial coefficient values.

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

On the electrical side, the generator torque and and the armature voltage are given by;

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

$$v_L = K_b \omega \tag{3.8}$$

The generator load torque can thus be varied by changing the current passing through it, for example, through a variable resistance load as in (3.9), or through other current control means.

$$T_g = K_t i_L = K_t \frac{v_L}{r_L} \tag{3.9}$$

3.2 Control

The general objective of control is to run the turbine at the operating region of high efficiency. In our case, this means increasing the total energy produced over time. For every wind speed, there is an optimal rotation speed for the turbine where the power coefficient is C_{pmax}. Here it would produce the maximum possible power for the system. This means when the wind speed U_w changes, the turbine speed ω should be brought to the optimal value as quickly as possible.

Controlling the turbine speed to obtain the maximum possible energy is central to the control of the VAWT in the thesis.

$$T_w = J\frac{d\omega}{dt} + T_g + T_f \tag{3.10}$$

To control the amount of torque available for speed changes to the optimal value,

the T_g torque may be regulated. The generator torque is thus controlled to regulate how much is available for the blades to speed up or down as quickly as required. For example, keeping T_w constant, decreasing T_g would speed the blades up faster. The converse is also true. T_g, in turn, is controlled by regulating the amount of current to the generator load as explained above.

Generally, once the wind speed increases, the turbine would begin gaining speed. To increase the speed gain so that the turbine reaches the optimum quickly, the opposing torque, T_g must be reduced. This will increase the torque available for speeding up the turbine. Reaching the optimum speed quickly means that the optimum power is attained quickly.

Greedy approaches such as MPPT, however, aim to maximize the instantaneous available power. This means that in transients, the algorithm will try to maximize the power, increase the load torque T_g, and thus delay the rotor speedup to reach the optimum value. This will decrease the total energy output.

4.   The Proposed Learning Method

The commonly used methods of control for the VAWT are not always sufficient to fully exploit the available wind power due to a number of reasons. Primarily, regulating the generator torque under the influence of the widely and rapidly fluctuating wind speed is not simple. The methods do not account for the possible degradation that a VAWT would go through in its lifetime either. Once the degradation occurs, the control algorithm is not tuned to perform at its best. Finally, the current control methods do not provide a way of adapting to the local wind patterns of the installation site. While the available control algorithms do perform fairly well, there is room for improvement.

The method of control proposed in this chapter is able to tackle the problems outlined above. In most cases, as suggested in [12, 13], The proposed method, using neural networks and MCMC, outperforms the currently available methods.

Neural networks are becoming increasingly popular in a wide variety of applications. Their applications in control are especially convenient as compared to traditional methods that require in-depth understanding of the systems to which they are applied. The RBFNN is used in the control structure for the VAWT in the thesis as a convenient non-linear approximator [13].

The MCMC is then used as the learning method for the parameters of the neural network. Although the theoretical foundations of the MCMC algorithm used here are complex, the practical application of the algorithm is fairly simple. The advantage of the general control framework used is that it;

- Can run on the same simple processors that can run MPPT. This means there is no need for a more costly processor implementation.

- Can use batch learning. Batch learning provides a time and processor utilization advantage. It again does not require a powerful processor.

- Can train for as long as is required. The training can continue even after the system has been commissioned.

The Chapter is organized as follows: The general structure of control is first explained. This is followed by an explanation on the RBFNN structure used in the thesis. Finally, the MCMC algorithm applied to tune the RBFNN parameters is explained. The chapter concludes by a brief explanation of the actual training process.

4.1 General architecture
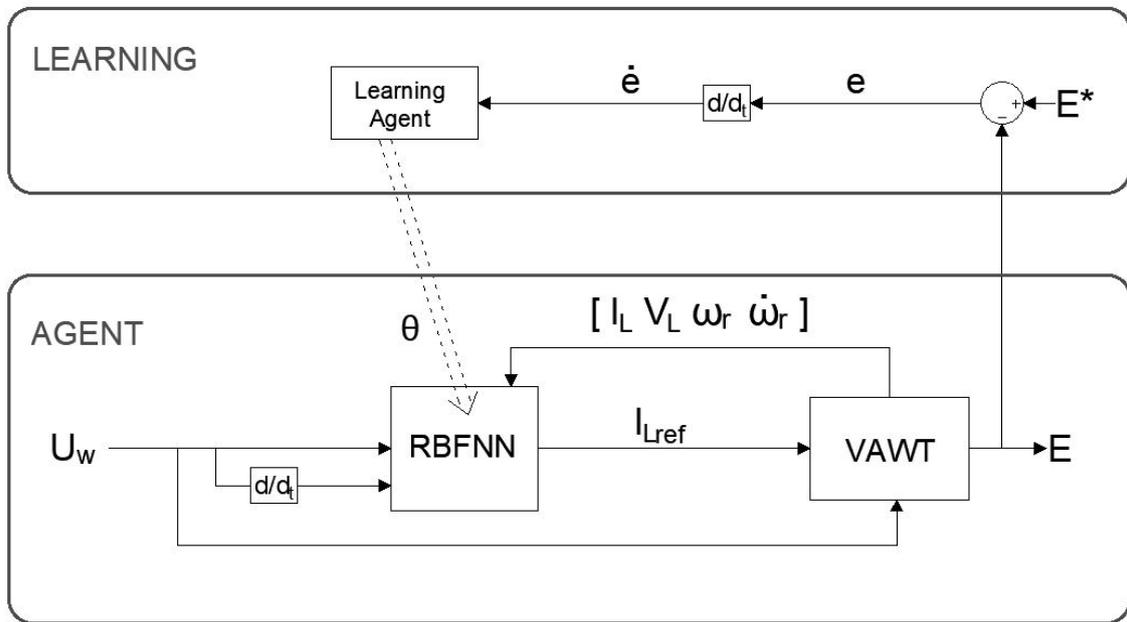
Figure 4.1 General architecture

The general architecture of the proposed system consists of two broad parts; the agent and the learning. The agenta part consists of the VAWT and the RBFNN, which is the neural network used for control of the VAWT. As explained in Chapter 3, the control input in this case is the load current. This is determined by the the RBFNN. To do this, the RBFNN takes as input 6 values which sufficiently describe the system. These are given in Table 4.1. As seen from Figure 4.1, four of the parameters are taken from the VAWT itself while the other two are the wind speed and its derivative. The wind speed and its derivative give information on the nature of the wind, since the RBFNN, lacking dynamics, cannot compute a derivative itself. The rest of the input values give information about the internal state of the VAWT system. These input values then pass through the neural network and a single variable representing the reference

current is given as output.

The performance of the VAWT is measured by the energy output, E produced by the system. This is computed by the integration of the power output, P measured over a specified period of time;

$$E = \int_0^t P \, dt \tag{4.1}$$

The reference energy output, E^* is obtained by integrating the optimum instantaneous power, P^*. The optimum instantaneous power is the power obtained from the VAWT rotor when the power coefficient value C_p is kept at its optimum over the whole period.

$$E^* = \int_0^t P^* \, dt \tag{4.2}$$

The error for the system is calculated as the difference between the optimum energy and the actual energy output.

$$e = E^* - E \tag{4.3}$$

The performance of the RBFNN is in turn determined by the network parameter values, θ. Tuning of the θ values are described in the learning part of the system. The learning agent takes the derivative of the error, $\dot{e}$, the instantaneous power error, as an input throughout training to improve the parameters θ of the RBFNN.

$$\dot{e} = \frac{de}{dt} \tag{4.4}$$

Table 4.1 RBFNN input value.

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	ω_r	Derivative of PMSG Rotor Speed

4.2 The RBFNN for control

The RBFNN is the control policy used to calculate the reference load current. It is a non-linear approximator. The structure is given in Figure 4.2;
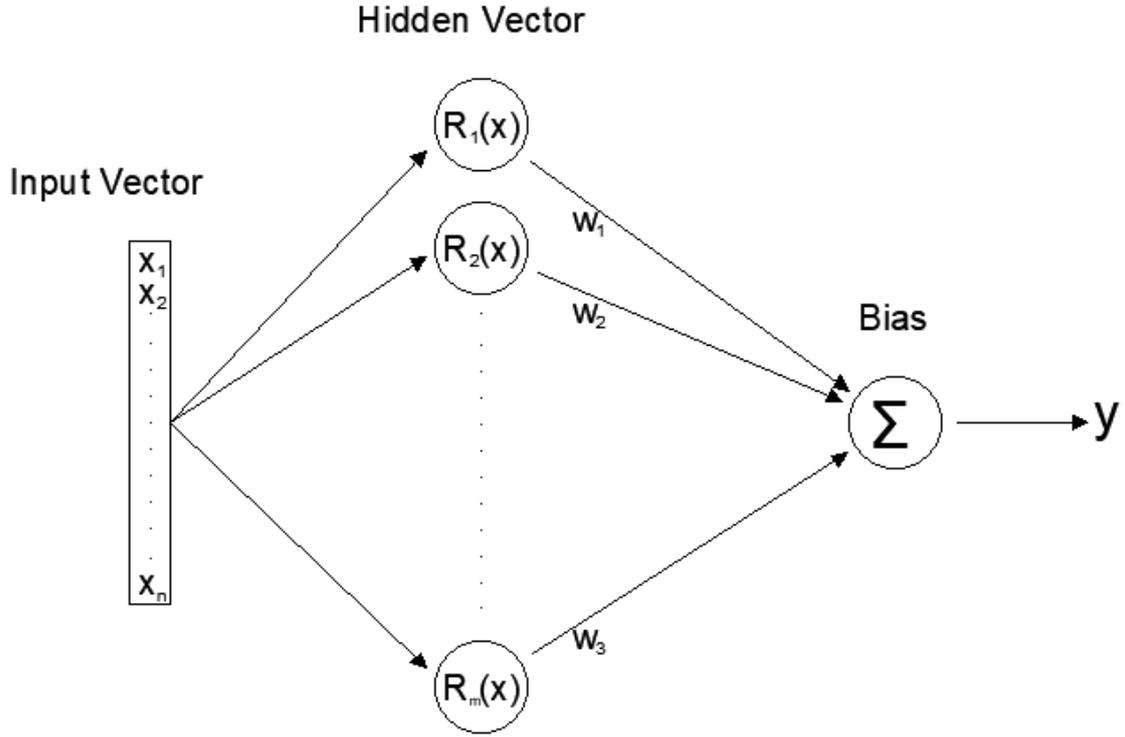
Figure 4.2 RBFNN structure

The equation for the network is given as;

$$y = F(x,\theta) = \sum_{i=0}^{n_r} w_i R_i(x) + bias \qquad (4.5)$$

where, w represents the weights while bias is a scale parameter. The receptive field $R_i(x)$ is the ith hidden node output. It is defined as;

$$R_i(x) = exp(\frac{||x - c_i||^2}{2b_i^2}) \qquad (4.6)$$

c and b represent the center and the corresponding standard deviation respectively for

each RBF.

$$c = \begin{bmatrix} c_{11} & \dots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{n1} & \dots & c_{nm} \end{bmatrix} \qquad (4.7)$$

$$b = \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix}^T \qquad (4.8)$$

The hidden node standard deviation, b and weights, w are designated as θ parameter values. These parameters affect the performance of the RBFNN. They are;

$$\theta = \begin{bmatrix} b & w \end{bmatrix} \qquad (4.9)$$

A more detailed explanation of the RBFNN is found in [29]. The challenge of finding the right θ parameter values for the RBFNN is tackled using a Markov Chain Monte Carlo (MCMC) method that has been developed in [12]. The method is explained in the next section.

4.3 Reinforcement learning using MCMC

Reinforcement learning is a form of machine learning where an agent develops rules for taking actions, based on an evaluation of rewards received thus far, with the objective of maximizing the cumulative rewards over a finite or infinite time horizon. The agent thus 'learns' what action to take given an observation, 'reinforced' by rewards.
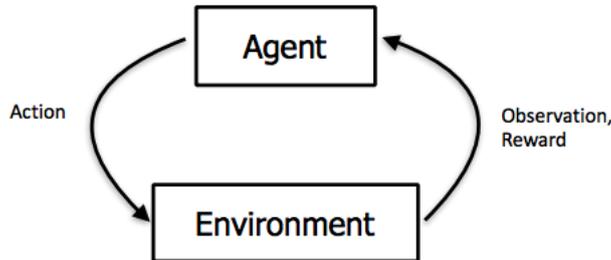
Figure 4.3 Reinforcement learning cycle

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 is difficult to achieve.

.

4.3.1 The Metropolis-Hastings algorithm

The Metropolis-Hastings(MH) algorithm is one of the most common MCMC methods. A candidate parameter is proposed using a proposal distribution. The proposed value is then applied to the system to obtain the state-action trajectory. The trajectory is then evaluated to get an unbiased estimate of J. Depending on the acceptance mechanism of MH algorithm it is either accepted or rejected according to $min(1, \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\mu(\theta')J_n(\theta')}{\mu(\theta)J_n(\theta)})$

where q is the proposal distribution, μ is the prior distribution and J is the cost function.

$$\rho(\theta, \theta') = \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\pi_n(\theta')}{\pi_n(\theta)} = \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\mu(\theta')J_n(\theta')}{\mu(\theta)J_n(\theta)} \tag{4.10}$$

Algorithm 1: Pseudo-marginal Metropolis-Hastings for RL

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

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

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

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

  Obtain an unbiased estimate J of $J(\theta)'$;

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

$$\rho(\theta, \theta') = \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{\mu(\theta')J_n(\theta')}{\mu(\theta)J_n(\theta)} \tag{4.11}$$

  else proposal rejected and set $\theta^{(k)} = \theta$ and $j^{(k)} = j$

end

4.3.2 MCMC

Most real-world problems involve using trajectory data to estimate some unknown parameters. in order to estimate these unknown parameters, there exact analytical methods. however, due to the high dimensionality and complexity of the systems at hand, these analytical methods are not feasible. to tackle this problem, the SMC methods are used where sampling is preferred over analytical methods [32].

In constructing the mcmc algo, the knowledge available can be incorporated into the prior distribution. after that, a posterior distributon is constructed with respect to the proportionality can be shown as

$$\pi(\theta) \propto \mu(\theta) \times J(\theta) \tag{4.12}$$

where μ is the prior distribution, J is the cost function and π is the resultant posterior distribution.

4.4 Training

While training, the algorithm functions by minimizing the cost function. The cost function, in this case, is found using the total quadratic reward. The reward function is calculated from the instantaneous power error, $\dot{e}$ which can easily be measured in a real wind turbine by comparing the wind speed and electrical power. This was described in detail in Section 4.1.

Once the initial parameters are set, one iteration of the setup is run. The performance from these parameters is gauged using the cost function. At the end of the iteration, a change of parameters is proposed θ_{prop} to improve the cost function.

The next iteration is run using the proposed parameters and a new cost function is generated. The new cost function value is compared to the cost function value associated with the previously accepted parameter values. The proposed parameters are accepted if the cost function value is lower than the previously accepted one. Otherwise, it is rejected and the next iteration is run. For every stage, a suitable number of iterations is chosen.

5. Hardware-In-the-Loop Experiments

Ideally, an actual VAWT system would be used for the simulation. However, due to the expense and difficulty of specifying a wind pattern for a real VAWT, a Hardware-In-the-Loop (HIL) setup is used.

To get a realistic result as close to the actual system as possible, a HIL setup developed in[14] for a small-scale VAWT is used. The setup is adapted to train the algorithm used in the thesis. It is further used to compare the performance of the RBFNN-with-MCMC algorithm with the other commonly used control algorithms mentioned in Chapter 3. The HIL setup used has been widely validated, tested and applied in [14, 15] where a detailed explanation can be found.

5.1 HIL model

A representation of the VAWT system and HIL model system is shown in Fig. 5.1 An actual VAWT system is shown in Fig. 5.1(a). It consists of the turbine blades that are connected to the generator by a shaft. The generator is connected to the load through power electronics and control unit. Figure 5.1(b) is the HIL system model. The wind speed and torque at the generator shaft are mimicked through a motor and gearbox, which are driven by actual wind data. The generator, load and control are the same components as would be present in the VAWT product.

The HIL setup consists of a computer, motor drive and motor, a gearbox, a generator and an electronic load. The block marked dSPACE is a real-time system for rapid prototyping of control algorithms, used here for the running of control algorithms in real-time, measuring and generating the analog signals of the system, and controlling the electronic load as well as generating the wind patterns at the shaft of the motor. In the next section, the components used in the HIL setup of the VAWT are introduced.

Figure 5.1 VAWT system and HIL model

The function for each is explained.

The main components of the HIL model are;

- Fast real-time control prototyping computer system (dSPACE [**?**])

- Motor & Driver

- Generator

- Electronic load

A photograph of the experimental setup is shown in Fig. 5.2. Leftmost is the electronic load, then to the right are: The motor drive and motor, coupling gearbox, generator (with a rectifier in front of it), electrical connector box, and finally the computer containing the dSPACE rapid control prototyping system.
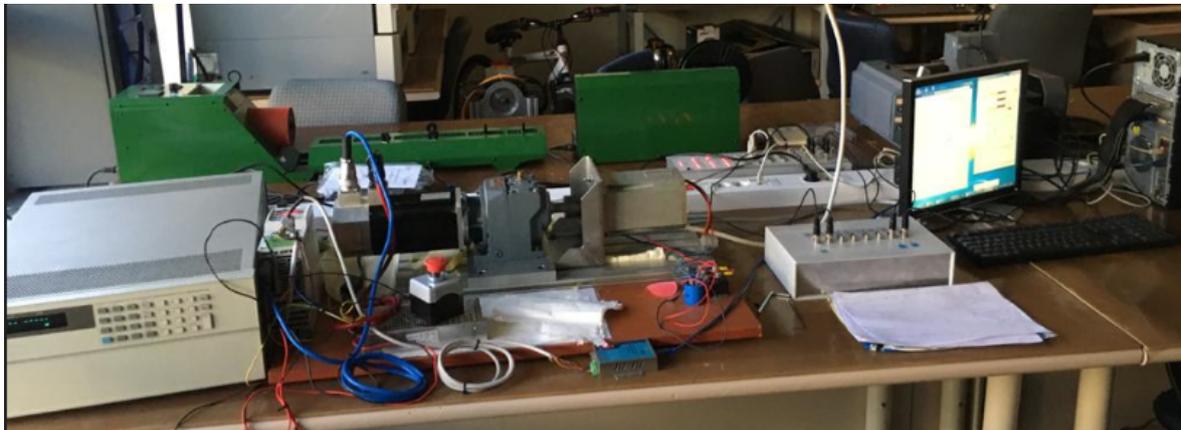
Figure 5.2 A photograph of the experimental setup.

5.2 Electromechanical Components

5.2.1 Motor

The motor is used to simulate desired wind patterns in the setup by generating the velocity and torque from simulated wind patterns.The Femsan 5F100810001 motor, a Permanent Magnet Synchronous Motor (PMSM) is used in the setup. The parameters as specified by the manufacturer are given in Table 5.1.

Table 5.1 Femsan 5F100810001 motor parameters.

Parameters	Value
Nominal torque	2.72 Nm
Nominal current	6.8 A
Nominal speed	4000 RPM
Nominal power	1139 W
Voltage constant	25.5 V/kRPM
Torque constant	0.4 Nm/A
Poles	8
Speed sensor	Resolver - 15 Pins
Rotor inertia, $J_{m,rotor}$	$4.4 \times 10^{-4} kgm^2$

The motor is controlled by the Mini Open Drive EXP 400V motor drive from TDE Macno. The appropriate motor parameters have been set using the OPD Explorer interface software from the PC. There is a direct connection from the PC to the driver (a connection module, RS-485 communication port, is used).

The motor is connected to the generator in the setup using a gearbox with a speed reduction ratio of 9.87. The gearbox (Yilmaz Reduktor MN002- B07) is used to decrease the speed and increase the torque to mechanical characteristics appropriate for the generator used in the VAWT system.

5.2.2 Generator

A custom built Permanent Magnet Synchronous Generator produced by FEMSAN was used in the setup. The specific parameters of the generator were then experimentally obtained in [14] and are given in Table 5.2.

The 3-phase AC output from the PMSG is converted to DC using a 3-phase diode bridge rectifier IXYS VUO160-12NO7. The specifics of the AC/DC rectifier with capacitor are given in [14].

5.2.3 Electronic Load

Table 5.2 PMSG and rectifier parameters[14].

Parameter	Value	Use
Number of pole pairs, p	6	$\omega_e = p\omega_{gen}$ and in simplified dc model
Equivalent Inertia, J	$4.88 \times 10^{-4} kgm^2$	VAWT inertia emulation
Friction torque, $T_f(\omega_m)$	$T_f(\omega_m) = -1.418 \times 10^{-6}\omega_m^2$ $+1.327 \times 10^{-3}\omega_m + 0.175 Nm$	VAWT inertia emulation, torque constant estimation
Line resistance, R_s	1.6 Ω	simplified DC model
Line inductance, L_s	4.6 mH	simplified DC model
Flux, ϕ_s	0.1069 Vs/rad	open circuit line/phase voltage and simplified dc model
Torque constant. K_t ϕ_s	1.307 Nm/A	$T_g = K_t i_L$
Threshold voltage for each diode, V_{th}	0.77 V	Simplified dc model

The Agilent N3306A is used to load the generator to the specified current calculated by the learning algorithm. It is a programmable electronic load used as a power sink in the setup. The desired load value is specified by the connection through the dSPACE board.

5.2.4 Rapid control prototyping system

A dSPACE DS1104 R&D controller board is used to implement a real-time control loop. This is used with the CP1105 connector panel. The connector panel is used to send analog signals to the motor drive. The electronic load is also controlled using the RS-232 communication port on the connector panel

The key advantage of the system is the integration with MATLAB and Simulink. This allows for models designed in Simulink to be directly implemented through ControlDesk software of dSPACE. This use with the exact application in the wind sector is shown in [30].

The dSPACE interface software environment in the PC is ControlDesk. It allows for manipulation of values collected through the board while also sending desired values for the motor and electronic load. Due to its compatibility with MATLAB, the integration of algorithms through ControlDesk and dSPACE is convenient.

5.2.5 Computer

The computer is used as the main interface between the user and the components in the setup. It is connected to the motor driver and electronic load through the dSPACE board. This connection is used to send signals to control the motor. The ControlDesk application is used to interface with dSPACE.

6. Experimental Results

The hardware-in-the-loop (HIL) setup is a good representative of the real VAWT system. The relevant control algorithms were thus run on the HIL system to compare performance. Foremost, it was used in the training of the proposed method using RBFNN with MCMC algorithm. This was to tune the RBFNN module which will act as the controller. Our objective is to compare the energy output of the proposed method to that of MPPT and SNC methods.

The proposed method can be trained in various styles. We implemented the following training approaches:

- Train in stages by applying input signals of increasing complexity at each stage, while using constant initial parameters in the initial stage.

- Train in stages by applying input signals of increasing complexity at each stage, while using random initial parameters in the initial stage. 5 such trials were performed

- Training completed in a single stage, with actual wind patterns applied from the beginning. 3 such trials were performed to average the results.

The reasoning behind using input signals of increasing complexity is to guide the learning agent to an appropriate part of the parameter space. The RBFNN parameters were initialized before training. The first stage trains it on a step input, followed by a sinusoidal input in the second stage. The third stage finally trains it on a random wind input, conforming to a realistic wind pattern.

In the single stage training approach, the RBFNN was initialized with random initial parameters and 3 trials were performed to determine the average of their performance. More trials will give a better average of performance, at the expense of time spent on physical experimentation.

Once the RBFNN is trained such that its control parameters have been found using the MCMC algorithm, it is compared to other algorithms. In this case, a comparison

of the total energy obtained with a random wind pattern input is done for the RBFNN, MPPT and SNC. The performance of the algorithm is thus determined by how much energy it produces at the end of a certain period of time.

6.1 RBFNN

The general structure for the RBFNN was introduced in chapter 4. The center matrix, c used for the RBFNN is given below [13]; The center parameters are taken from the working interval of each input for the RBFNN, where each row belongs to one input and the center values are evenly spread in the range of values expected for that input.

$$\begin{bmatrix} 4.66 & 5.99 & 7.32 & 8.65 & 9.98 & 11.31 \\ -8.33 & -5 & -1.67 & 1.66 & 4.99 & 8.32 \\ 0.83 & 2.49 & 4.15 & 5.81 & 7.47 & 9.13 \\ 3.32 & 9.96 & 16.6 & 23.24 & 29.88 & 36.52 \\ 5 & 11 & 17 & 23 & 29 & 35 \\ -4.998 & -3 & -1.002 & 0.996 & 2.994 & 4.992 \end{bmatrix}$$

6.2 Constant Initialized Parameters (Training in stages)

The initial parameters of the RBFNN are set to be the initial distribution for the MCMC algorithm. These parameters are the hidden node standard deviations and weights. They are set as follows;

$$\theta_0 = \begin{bmatrix} b_0 & w_0 \end{bmatrix}$$

$$b_0 = \begin{bmatrix} 20 & 20 & 20 & 20 & 20 & 20 \end{bmatrix}$$

$$w_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Where b is the standard deviation and w is the weight. The weight is chosen as the minimum non-zero coefficient close to zero. Both should cover the vector space of hidden nodes.

As done in [13], the neural network was trained in 3 stages. The first stage was a step

input, this was followed by a sine input. Finally a realistic random wind data was used as input. The stages and the results from each are presented next.

6.2.1 First stage of training

The reference wind in the first stage is step wind of 8m/s, applied at time $t = 0$. This is well within the limits of the VAWT setup working range which is 6 m/s to 12 m/s. As the most important part of the of the simulation is the transient behaviour, the time for every iteration is set at a relatively short 120 seconds. This is enough time for the system to settle. The first iteration with the untrained RBFNN with initial parameters θ_0 yields performance as shown in Figure 6.1. The plot shows the change in instantaneous power with time.

Three key points are to be noted in the figure. These determine the performance of the system;

- Start of ascent: How quickly the system reacts to a change.

- Rate of ascent: How fast the system reaches the final change value after it reacts

- Final constant level: The final value around which the generated power reaches.

Start of Ascent
The instantaneous power generated is found to start increasing around the 20th second. This means the system is quite slow to react. Ideally, the quicker the system reacts the better.
Rate of Ascent
The instantaneous power keeps increasing at a fairly constant rate until the 50th second. The rate of increase is also a key factor in gauging the performance of the system. A steeper slope means the system is reacting faster.
Final constant level
The final constant level is around 54 Watts. This value is far from the optimum instantaneous **power which is close to 79 Watts**
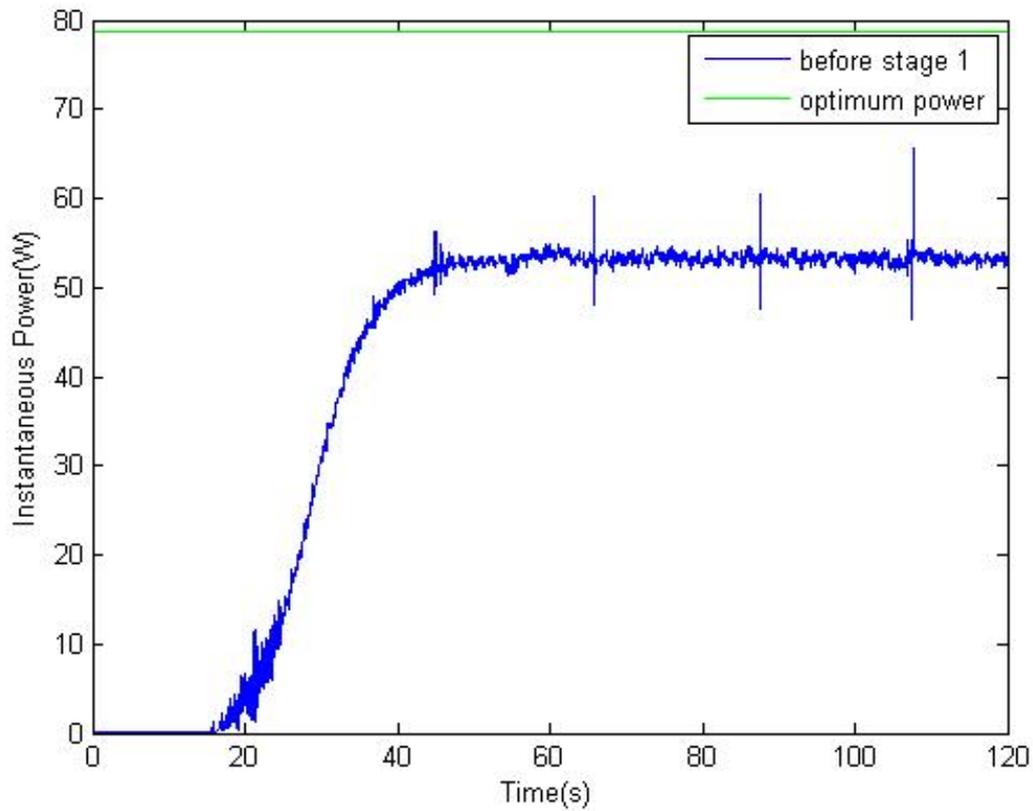
Figure 6.1 Performance before training, response to a step wind speed change.

(a) Cost   (b) Proposed   (c) Accepted
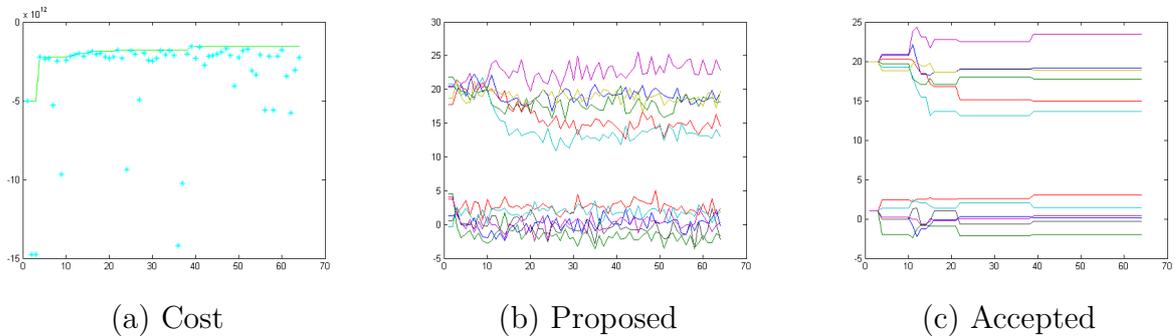
Figure 6.2 Cost function and parameter changes in the first stage of training

The actual training was done once the base performance was recorded. The plots in Fig. 6.2 illustrate the training. In Fig. 6.2(a), the cost function is shown, together with the best values achieved as a continuous line appearing at the top. During training many θ parameters are proposed, their performance (in terms of cost function value) are tested, and those which improve over the best existing parameters are kept, whereas those with worse performance are rejected. The proposed parameters θ at each iteration are shown in Figure 6.2(b), and those which do improve the cost function value, and accepted, are shown in Fig. 6.2(c).

28

Figure 6.2(c) shows the accepted value with respect to iterations. A set of θ values are accepted if the resulting cost function is higher than the one stored. Once it has been accepted, all the iterations that follow have their cost function compared to the stored value. Figure 6.2(a) shows the cost function values. the accepted cost value that has become the best so far, is shown as a line. The cost function from every iteration is shown on the graph as a '+' mark. The continuous line representing the accepted cost value thus only changes in the direction to become less than its current accepted value. It is otherwise discarded (discarded values are represented by the '+' marks on Fig.6.2(a)).

The behaviour is expected. The frequency of acceptance at the beginning is high as the proposed values converge to those suitable for the system to give better performance. Once the θ values found give a fairly good performance, further improving the performance becomes difficult.

The parameters obtained from the first stage are a mean of the last quarter iterations as explained in section 4.4. The obtained parameters θ_1 at the end of stage 1, give satisfactory performance with step inputs as discussed next.

$$\theta_1 = \begin{bmatrix} b_1 & w_1 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} 19,1546 & 17,7314 & 14,9783 & 13,6768 & 23,5067 & 18,8672 \end{bmatrix}$$

$$w_1 = \begin{bmatrix} -0,3496 & 0,1080 & -1,9467 & 2,9907 & 1,4315 & 0,3527 \end{bmatrix}$$
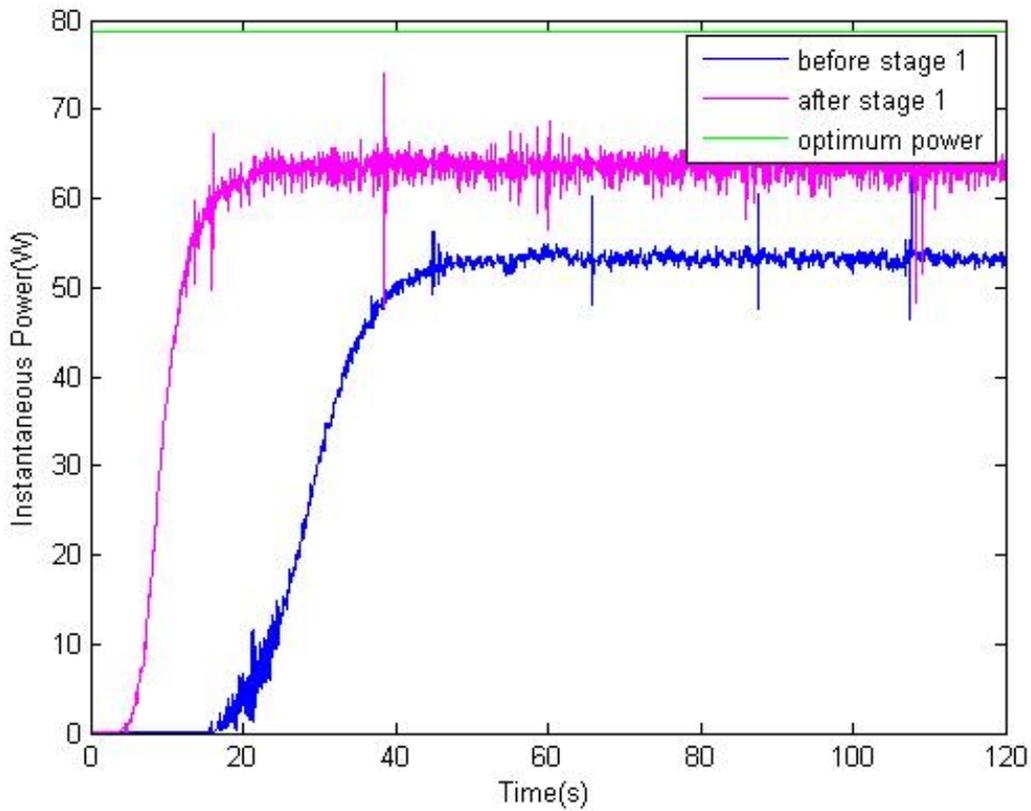
Figure 6.3 Performance before and after stage 1 training

Figure 6.3 shows a comparison of performance with parameters θ_0 and θ_1, before and after training. The resultant instantaneous power after the training is higher, as well as the response time, which is faster. It can be concluded that the RBFNN thus performs better after 64 iterations.

6.2.2 Second stage of training

The second stage of training is on a sinusoidal input. This is to train the network to deal with periodically changing patterns. A sinusoidal wind speed input with the characteristic of $U_\omega = 8.5 + 1.5 sin(0.2t)$ is given. The length of the simulation is taken as 200 seconds which is longer compared to Stage 1, to allow for sufficient time for the learning of changing input. The starting parameters of this stage (θ_1) are the results of the first stage. The result of the training is shown in Figure 6.4;

It can be seen that a further improvement on the cost function was obtained but the performance is already good. Therefore most of the improvement came in the first

(a) Cost   (b) Proposed   (c) Accepted
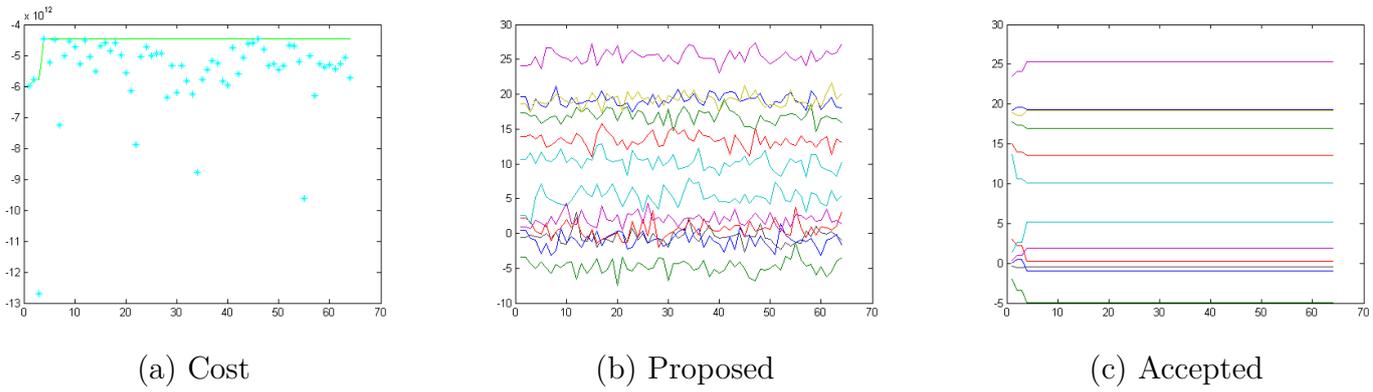
Figure 6.4 Second stage of training

part of the stage and the cost function did not change afterwards as observed from the
steady values in Figure 6.4(c) which shows the last accepted parameters. The resulting
parameters from the second stage are a mean of the last quarter iterations, θ_2. The
obtained parameters give satisfactory performance with sinusoidal inputs. As a point
of comparison, the performance before and after training is shown in Fig. 6.5;

$$\theta_2 = \begin{bmatrix} b_2 & w_2 \end{bmatrix}$$

$$b_2 = \begin{bmatrix} 19.3178 & 16.9057 & 13.5480 & 10.1003 & 25.3663 & 19.1908 \end{bmatrix}$$

$$w_2 = \begin{bmatrix} -0.4878 & -0.9533 & -4.9919 & 0.2965 & 5.1840 & 1.9230 \end{bmatrix}$$
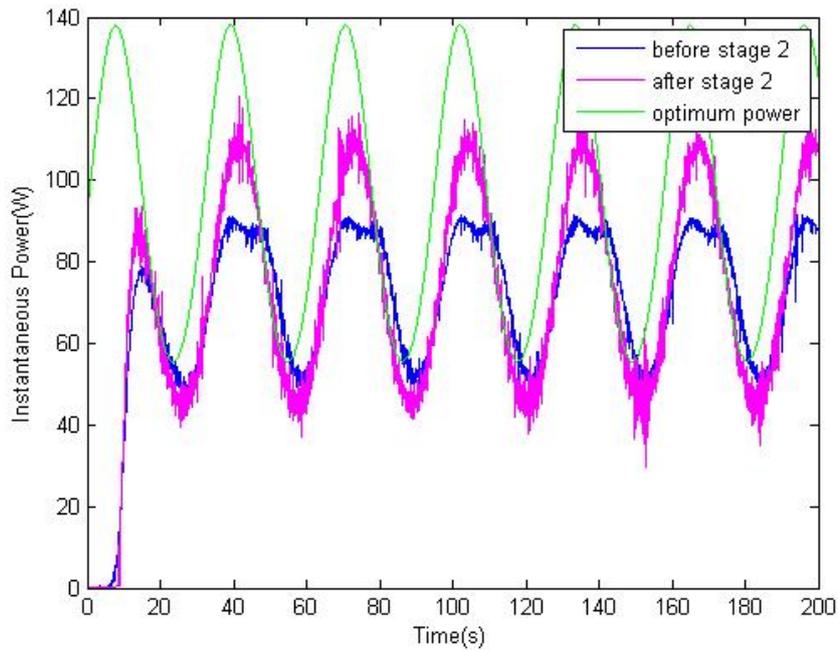
Figure 6.5 Performance before and after stage 2 training

31

6.2.3 Third stage of training

A real wind profile is used in the third stage of training. With the initial parameters set as θ_2, the training time is set as 300 seconds. The result of the training is shown in Figure 6.6.

(a) Cost   (b) Proposed   (c) Accepted
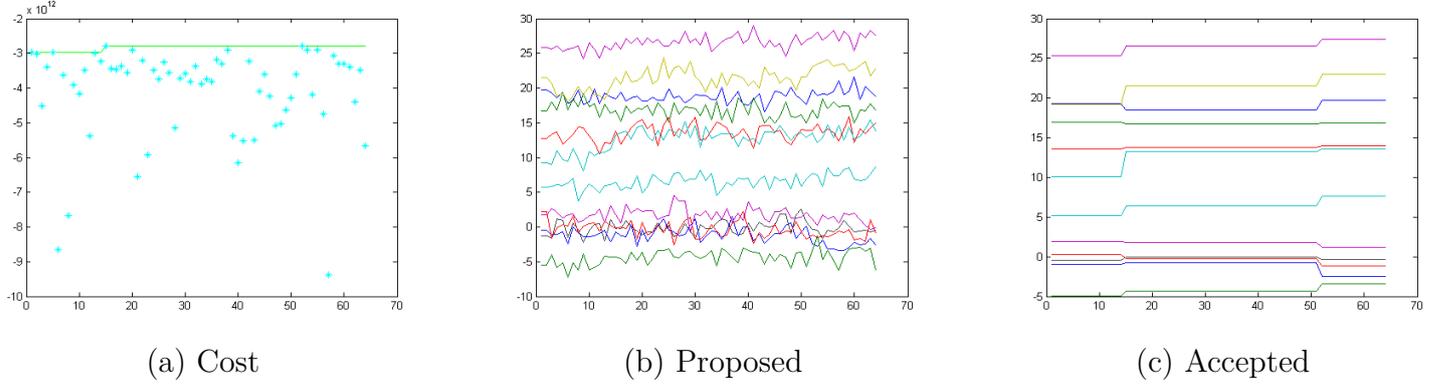
Figure 6.6 Third stage of training

The resulting parameters from the third stage are a mean of the last quarter iterations. The obtained parameters give satisfactory performance with random inputs.

$$\theta_3 = \begin{bmatrix} b_3 & w_3 \end{bmatrix}$$

$$b_3 = \begin{bmatrix} 19.5056 & 16.8061 & 13.9125 & 13.4814 & 27.2057 & 22.7501 \end{bmatrix}$$

$$w_3 = \begin{bmatrix} -0.2862 & -2.1477 & -3.5553 & -0.9850 & 7.4361 & 1.2864 \end{bmatrix}$$

As a point of comparison, the performance before and after training is shown in Figure 6.7.

6.2.4 Comparison of RBFNN with MPPT and SNC

The plot 6.8 shows power obtained from running RBFNN, MPPT and SNC algorithms for control of the system. The same random wind input is given and the total energy obtained from each is recorded.

Algorithm	Energy
RBFNN	2.3551×10^4
MPPT	2.1723×10^4
SNC	2.6382×10^4
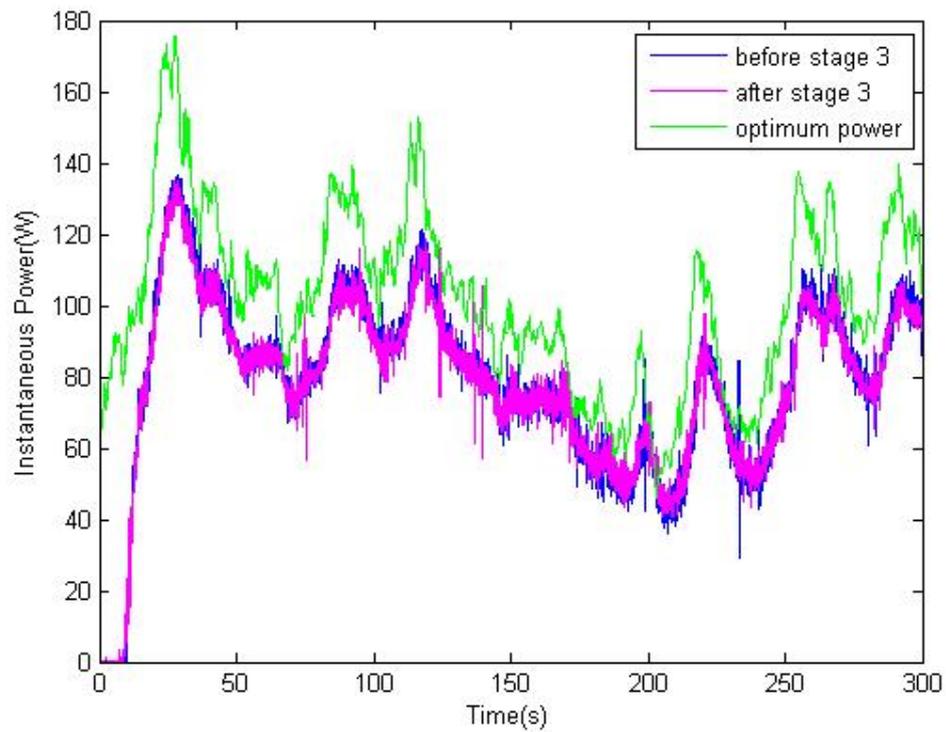
Figure 6.7 Performance before and after stage 3 training

The RBFNN power plot is better than MPPT. Over the simulation length, this causes a significant energy difference. Over longer periods, this difference becomes markedly important.
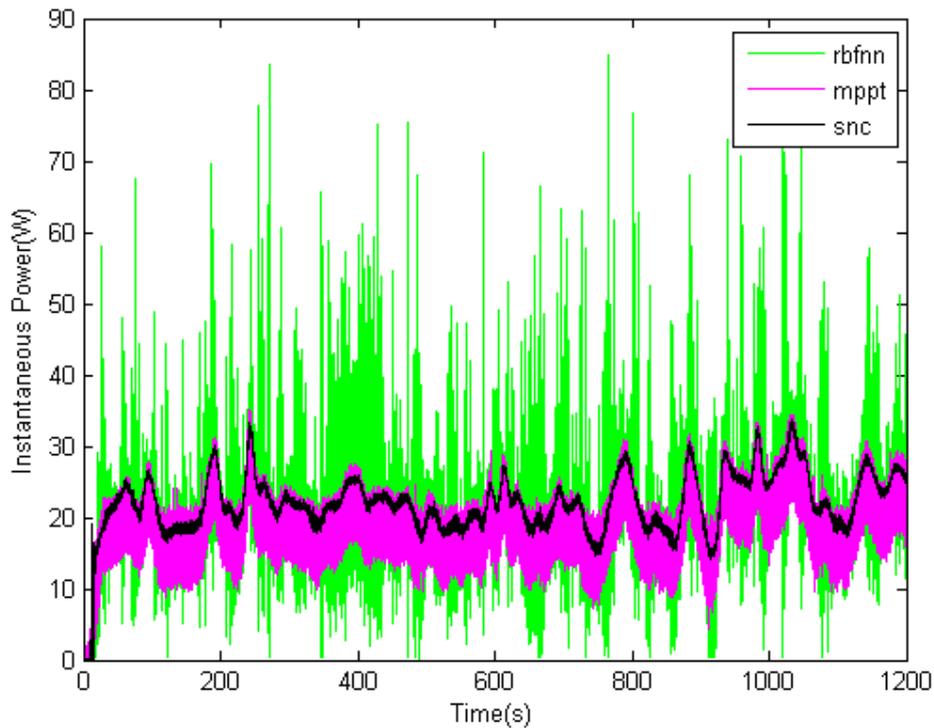
Figure 6.8 Power comparison of RBFNN, MPPT and SNC

6.3 Random Initialized Parameters (Training in stages)

In this test, parameters were initialized to random values and then the three stages were run to completion. This is in contrast to what was done in Section 6.1 where constant parameters were chosen as initial values. Although it would have been preferable to run the trials with random initial parameters a large number of times, to obtain statistically meaningful results. However, due the time-constraints, only 5 trials were run to observe and ensure the repeatability of the experiment.

For each trial, the starting parameters of the neural network were randomly initialized. the network is then trained using a step wind pattern as the first stage. The initialized parameters for each of the trials are shown in Table 6.1.

At the end of training stage 1, the average value of the last quarter accepted parameters is taken as the new theta value. The new theta values are used as the initialised parameters for training in stage 2. The same is done for resultant theta values from stage 2. The initialized values used for both stage 2 and 3 are shown in Table 6.2 and Table 6.3

34

Table 6.1 Stage 1 Initialized Parameters

trial 1 b	8.76	18.81	0.12	12.21	16.02	4.66
trial 1 w	18.65	15.27	16.53	11.47	15.85	6.58
trial 2 b	4.62	8.07	2.44	5.37	5.16	6.63
trial 2 w	3.04	6.96	2.43	17.68	1.89	18.60
trial 3 b	7.36	13.59	11.36	13.04	9.82	7.97
trial 3 w	9.55	1.33	8.22	19.38	15.61	14.58
trial 4 b	3.41	13.54	18.05	7.04	2.00	15.86
trial 4 w	9.37	13.75	13.66	8.43	13.19	12.67
trial 5 b	3.82	17.14	18.86	2.07	5.93	15.98
trial 5 w	8.83	9.85	0.59	11.45	11.45	14.83

Table 6.2 Stage 2 initialized Parameters

trial 1 b	3.94	14.14	1.29	6.45	13.96	1.81
trial 1 w	10.49	−1.42	15.36	−5.99	11.59	−0.86
trial 2 b	9.04	13.09	24.35	−0.31	1.16	20.42
trial 2 w	4.94	5.50	−7.94	3.18	12.21	15.99
trial 3 b	−2.73	5.63	0.66	−3.54	17.25	8.07
trial 3 w	6.29	−2.49	−2.30	−0.23	3.75	4.30
trial 4 b	−5.59	13.51	12.39	4.47	−0.72	18.11
trial 4 w	−2.96	−4.46	8.94	8.84	9.93	10.37
trial 5 b	9.04	13.09	24.35	−0.31	1.16	20.42
trial 5 w	4.94	5.50	−7.94	3.18	12.21	15.99

Table 6.3 Stage 3 Initialized Parameters

trial 1 b	5.47	14.21	−0.64	6.51	15.33	1.96
trial 1 w	10.12	−1.52	12.36	−6.21	11.12	−1.25
trial 2 b	9.04	13.09	24.35	−0.31	1.16	20.42
trial 2 w	4.94	5.50	−7.94	3.18	12.21	15.99
trial 3 b	1.40	8.84	1.65	−3.30	17.73	9.31
trial 3 w	3.17	−3.13	−1.23	−0.13	3.43	8.44
trial 4 b	−5.95	14.28	15.16	0.08	−4.07	17.23
trial 4 w	−2.68	−1.36	1.18	6.04	7.73	5.78
trial 5 b	9.04	13.09	24.35	−0.31	1.16	20.42
trial 5 w	4.94	5.50	−7.94	3.18	12.21	15.99

Table 6.4 Stage 3 Final Parameters

trial 1 b	5.44	14.18	−0.66	6.57	15.32	2.04
trial 1 w	10.16	−1.54	12.34	−6.18	11.18	−1.31
trial 2 b	−0.68	3.59	1.90	−0.54	1.26	13.49
trial 2 w	−2.80	8.71	4.41	15.58	5.97	10.10
trial 3 b	1.40	8.84	1.65	−3.30	17.73	9.31
trial 3 w	3.17	−3.13	−1.23	−0.13	3.43	8.44
trial 4 b	−5.77	13.97	14.23	0.05	−3.90	17.33
trial 4 w	−2.75	−1.60	1.76	6.84	8.59	5.76
trial 5 b	9.10	13.14	24.37	−0.28	1.14	20.46
trial 5 w	4.92	5.43	−7.90	3.24	12.24	15.88

6.3.1 Convergence of cost function

It is noteworthy that, while the final accepted parameters in the neural network for every trial is different, the cost function converges around the same value. This indicates that the final performance from all the trials is the best achievable after training. This is shown in Table 6.5.

The performance of stages 1, 2 and 3 for each random initialization of the trials can be seen in Appendix A.

Table 6.5 Cost function values.

trial 1	$-2.96 \cdot 10^{12}$
trial 2	$-2.79 \cdot 10^{12}$
trial 3	$-2.60 \cdot 10^{12}$
trial 4	$-2.74 \cdot 10^{12}$
trial 5	$-2.74 \cdot 10^{12}$

6.4 Random Initialized Parameters (Direct Training)

Since the concept of training in stages was observed to work well, the necessity of the training in stages was questioned. the reasoning for training in stages was to guide the parameters in the parameter space towards the desired performance. It was conjectured that due to the large vector space of the parameters, finding the satisfactory performance parameters would be difficult. However, since multiple sets of parameters can be used to obtain satisfactory performance, direct training in the form of using simulated wind data directly from initial parameters, was attempted in the thesis.

Five trials of learning were performed. Each trial had 250 iterations of training. The initial parameters before training and the resulting parameters after training are given in Table 6.7 and 6.8 the final cost function values are given in Table 6.6. The cost function values can be compared to those found in Table 6.5

Table 6.6 Cost function values.

trial 1	$-3.10 \ \cdot 10^{12}$
trial 2	$-8.10 \ \cdot 10^{12}$
trial 3	$-7.61 \ \cdot 10^{12}$
trial 4	$-3.35 \ \cdot 10^{12}$
trial 5	$-1.08 \ \cdot 10^{13}$
trial 999	$-2.75 \ \cdot 10^{12}$

Two out of the five trials reached close to the minimum cost function values previously estimated from training in stages. While the results are not ideal, they are promising for a number of reasons. Running more cycles is expected to give a more conclusive result on the suitability of direct training.

Only 250 iterations in each cycle were run during training due to time constraints. Running more iterations might have resulted in finding the appropriate parameters. Pursuing direct cycle training is also desirable as the process is considerably less complicated as compared to the training in stages.

One extra trial was run with 999 iterations. This gave promising results. While it would be desirable to run more trials with 999 iterations, the real-time nature of the experiments presents a challenge due the time expense involved.

Table 6.7 Direct Training Initialized Parameters

trial 1 b	4.85	0.37	14.41	11.18	7.73	2.22
trial 1 w	5.17	9.85	14.60	8.10	16.06	10.73
trial 2 b	10.83	9.12	5.78	8.46	13.10	0.47
trial 2 w	16.19	18.84	12.38	10.34	18.12	3.27
trial 3 b	5.85	0.79	1.04	0.65	6.71	3.03
trial 3 w	18.34	10.77	7.84	12.99	10.42	2.28
trial 4 b	7.98	5.36	16.65	19.91	13.00	14.08
trial 4 w	18.65	13.75	11.37	7.62	12.69	7.26
trial 5 b	13.40	6.95	13.39	5.91	13.61	7.88
trial 5 w	7.84	4.14	14.49	16.39	18.53	10.28

6.5 Performance Comparison

Table 6.8 Direct Training Final Parameters

trial 1 b	−0.38	−2.86	4.85	15.81	2.12	3.43
trial 1 w	6.44	3.10	−1.36	3.65	15.04	0.62
trial 2 b	10.83	9.12	5.78	8.46	13.10	0.47
trial 2 w	16.19	18.84	12.38	10.34	18.12	3.27
trial 3 b	5.56	−2.35	2.38	0.16	6.39	4.93
trial 3 w	18.33	11.46	9.66	12.38	9.86	1.71
trial 4 b	3.80	−1.32	0.29	19.02	4.21	6.41
trial 4 w	18.65	12.46	15.35	3.54	14.46	0.89
trial 5 b	13.40	6.95	13.39	5.91	13.61	7.88
trial 5 w	7.84	4.14	14.49	16.39	18.53	10.28

The performance of the proposed method is compared to MPPT and SNC. The proposed method is applied on the system with the parameters obtained from the system training. The algorithms are run on the system for equal periods of time. The total energy produced by each algorithm is then compared.

The wind input data used is from actual readings taken on-campus at Sabanci University in Istanbul. Readings were taken over a 24 hour period at 10 second intervals. The points were then interpolated. Due to the limited valid working range of the system, a bias was added to the wind speed data so that it falls within the valid working range. Table 6.9 gives the total energy produced from the five trials performed. These are compared to MPPT and SNC which were each run thrice. This is because the performance of the algorithms is slightly different each time. Finally, the performance of the directly trained proposed method is given. Only one of the five cycles is found to give performance comparable to the other algorithms. the most likely reason for this is the number of iterations for which the algorithm was trained. As the training was was not guided by stages, more iterations could be needed for the algorithm to find parameters that would give comparable performance.

The proposed method is found to perform better than MPPT every time it is trained. This is expected, as the MPPT algorithm is greedy and does not account for long-term performance. The SNC, on the other hand, is found to sometimes perform better than the proposed method. This happens because of the randomized nature of the training for the proposed method.

Table 6.9 Energy comparison.

Trial 1	2.67	$\cdot 10^{04}$
Trial 2	2.58	$\cdot 10^{04}$
Trial 3	2.45	$\cdot 10^{04}$
Trial 4	2.68	$\cdot 10^{04}$
Trial 5	2.49	$\cdot 10^{04}$
MPPT	2.18	$\cdot 10^{04}$
MPPT	2.17	$\cdot 10^{04}$
MPPT	2.17	$\cdot 10^{04}$
SNC	2.61	$\cdot 10^{04}$
SNC	2.61	$\cdot 10^{04}$
SNC	2.61	$\cdot 10^{04}$
Direct training 1	2.44	$\cdot 10^{04}$
Direct training 2	71.74	
Direct training 3	1.73	$\cdot 10^{04}$
Direct training 4	1.97	$\cdot 10^{04}$
Direct training 5	71.25	
Direct training 999	2.46	$\cdot 10^{04}$

7.   Conclusion and Future work

The VAWT currently has a number of commonly used algorithms to achieve efficiency in power generation. However, due to the rapid change in wind, they do not capture the maximum possible energy. Machine learning methods are thought to provide a good solution to the problem. Specifically a neural network is used in the thesis.

A neural network is equipped to deal with the dynamic nature of control required for wind. This is because of the constant change in the control variable that wind requires. The challenge that results is to find appropriate parameters for the neural network. Due to the large number of parameters, this can be especially challenging.

To train the neural network, an MCMC algorithm is used. MCMC algorithms are considered part of the general reinforcement methods available today. Due to the large vector space to explore to find suitable parameters for the neural network, the MCMC structure is found to be appropriate for the search.

In the search for optimum parameters, a gradual method (in stages), developed previously in [13] was used. 3 stages were configured. The first training stage was with a step input. The second with a sine input and the final one with simulated wind data.

One group of training in three stages was designated as a trial. As such, 5 trials were run. At the end of the trials, 5 different groups of parameters were obtained. Once the training was done, comparison of results was done with the performance of MPPT and SNC .

The comparison was done with actual wind data input. The parameters of each trial were used on the RBFNN . The final energy values obtained from each trials was recorded. The MPPT and SNC algorithms were also used with the data. the final energy values obtained were also recorded in the same table.

Finally, the obtained energy values from the above were compared to each other.

The RBFNN performance from each trial was found to fluctuate widely. It was however always better than the MPPT algorithm. Out of the 5 trials, 2 had higher energy values

than the SNC while the other three were lower. The RBFNN control method is still desirable as it is not as computationally expensive as the SNC algorithm. The SNC algorithm also has the added challenge of requiring model information.

At the end of the experiments, the need for gradual training was put into question. thus the RBFNN was directly trained using the random wind pattern data to gauge performance at the end of training. The results were promising as the direct training method gave performance at par with the stages training method in some of the trials.

In the thesis, all the training and testing was done on a HIL setup, where the rotor speed and torque were simulated by a servo motor, and the generator and load were implemented as physical systems. This was the optimum setup for the research as it provided the closest testing to the actual VAWT and allowed the testing of arbitrary wind patterns.

The RBFNN control method applied here may be compared with the Deep deterministic Policy gradient (DDPG) method. This was attempted but found to take too long. Despite the time cost, DDPG is considered to be a powerful method in reinforcement learning. Implementation on an actual VAWT system would be the final step in the application of the algorithm

BIBLIOGRAPHY

[1] https://www.c2es.org

[2] H. Riegler, "HAWT versus VAWT: Small VAWTs find a clear niche." *Refocus,* 4(4), 44-46, 2003

[3] S. D. L. Salle, D. Reardon, W. E. Leithead , & M. J. Grimble, "Review of wind turbine control." *International Journal of Control*, 52(6), 1295-1310, 1990.

[4] Johnson, K. E., Pao, L. Y., Balas, M. J., & Fingersh, L. J., "Control of variable speed wind turbines: standard and adaptive techniques for maximizing energy capture." *IEEE Control Systems*, 26(3), 70-81, 2006

[5] IXYs *VUO160-12NO7 Datasheet.*

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

[7] 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

[8] 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.

[9] 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.

[10] 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.

[11] reference for white paper on ML in robotics

[12] V. Tavakol Aghaei, *Markov Chain Monte Carlo Algorithm for Bayesian Policy Search* Master's thesis, Sabanci University, Istanbul, 2019.

[13] A. Agabaoglu *Bayesian Reinforcement learning with MCMC to maximize energy output of vertical axis wind turbine* Master's thesis, Sabanci University, Istanbul, 2019.

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

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

[16] A. Bouscayrol, "Different types of hardware-in-the-loop simulation for electric drives," in Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on, pp. 2146-2151, IEEE, 2008.

[17] "Ruzgar enerji potansiyel atlasi."

[18] D. Lin, P. Zhou, and Z. Cendes, "In-depth study of the torque constant for permanent-magnet machines," Magnetics, IEEE Transactions on, vol. 45, no. 12, pp. 5383-5387, 2009.

[19] N. Mohan and T. M. Undeland, *Power electronics: converters, applications, and design.* John Wiley & Sons, 2007.

[20] Ugur-14 T. Wildi, *Electrical Machines, Drives and Power Systems: Pearson New International Edition.* Pearson Education Limited, 2013.

[21] H. Rashid, *Power Electronics: Circuits, Devices, and Applications.* Pearson/Prentice Hall,2004.

[22] T. M. H. Nicky, K. Tan, and S. Islam, "Mitigation of harmonics in wind turbine driven variable speed permanent magnet synchronous generators," *in Power Engineering Conference, 2005. IPEC 2005.* The 7th International, pp. 1159-1164, IEEE, 2005.

[23] S. Hansen, L. Asiminoaei, and F. Blaabjerg, "Simple and advanced methods for calculating six-pulse diode recti
er line-side harmonics," in *Industry Applications Conference, 2003. 38th IAS Annual Meeting.* Conference Record of the, vol. 3, pp. 2056-2062, IEEE, 2003.

[24] A. Urtasun, P. Sanchis, I. San Martin, J. Lopez, and L. Marroyo, "Modeling of small wind turbines based on pmsg with diode bridge for sensorless maximum power tracking," *Renewable energy,* vol. 55, pp. 138-149, 2013.

[25] F. D. Adegas, J. A. V. Ale, F. S. dos Reis, G. C. da Silva Simioni, and R. Tonkoski, "Maximum power point tracker for small wind turbines including harmonic mitigation," in *2006 European Wind Energy Conference and Exhibition,* pp. 1-10, 2006.

[26] K.-Y. Lo, Y.-M. Chen, and Y.-R. Chang, "Mppt battery charger for stand-alone wind power system," *Power Electronics, IEEE Transactions on,* vol. 26, no. 6, pp. 1631-1638, 2011.

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

[28] D.-H. Tran, B. Sareni, X. Roboam, and C. Espanet, "Integrated optimal design of a passive wind turbine system: an experimental validation," *Sustainable Energy, IEEE Transactions* on, vol. 1, no. 1, pp. 48-56, 2010.

[29] J. Liu, "Radial Basis Function (RBF) Neural Network Control for Mechanical Systems," pp. 339-362. 01 2013.

[30] L. Barote, C. Marinescu, and M. N. Cirstea, "Control Structure for Single-Phase Stand-Alone Wind-Based Energy Sources," *IEEE Transactions on Industrial Electronics,* vol. 60, no. 2, pp. 764-772, Feb. 2013.

[31] www.dspace.com

[32] A.D., N. De Freitas and N.J. Gordon, "An introduction to Sequential Monte Carlo Methods", *SMC in Practice*, 2001

APPENDIX A

The simulations described in Sec. (6.3) are described here. Figures (A1.9, A1.10 and A1.11) show the evolution of the cost function, the proposed parameter values θ and the accepted values in sub figures (a), (b) and (c) respectively for each individual initialization. Please note that the vertical axes of each graph may not be the same.

It can generally be seen from the graphs that progressive improvement of the cost function results in many accepted proposals, and if a good set of parameters θ are found from the beginning of the trial, then acceptance rate is not as high.

The results for the direct cycle training are given in figure (A1.12).

(a) Cost   (b) Proposed   (c) Accepted

(a) Cost   (b) Proposed   (c) Accepted

(a) Cost   (b) Proposed   (c) Accepted

(a) Cost ($\times 10^{13}$)   (b) Proposed   (c) Accepted

(a) Cost   (b) Proposed   (c) Accepted
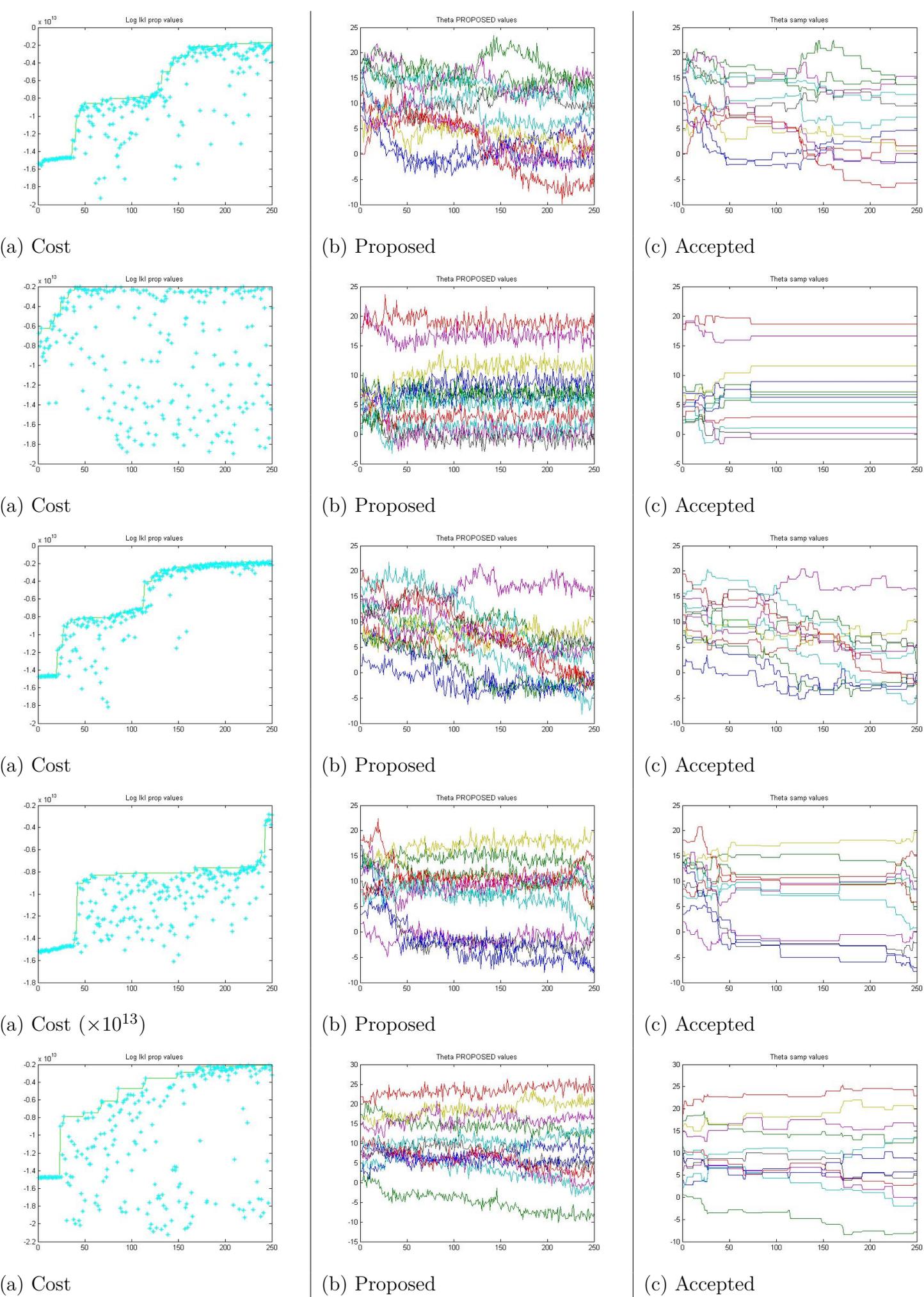
Figure A.1 Stage 1 training for trials 1-5

(a) Cost (b) Proposed (c) Accepted

(a) Cost (b) Proposed (c) Accepted

(a) Cost (b) Proposed (c) Accepted

(a) Cost (b) Proposed (c) Accepted

(a) Cost (b) Proposed (c) Accepted
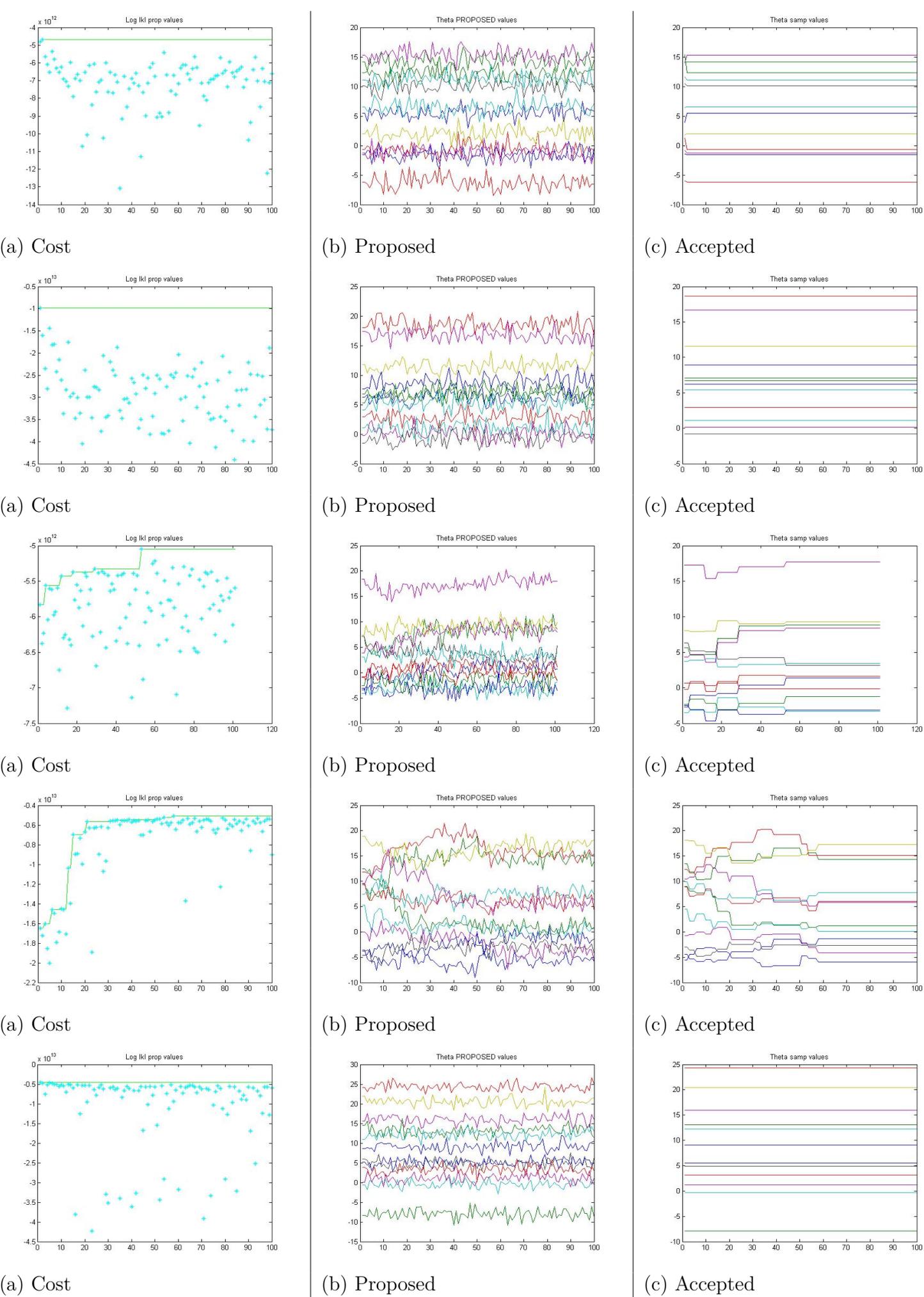
Figure A.2 Stage 2 training for trials 1-5

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted
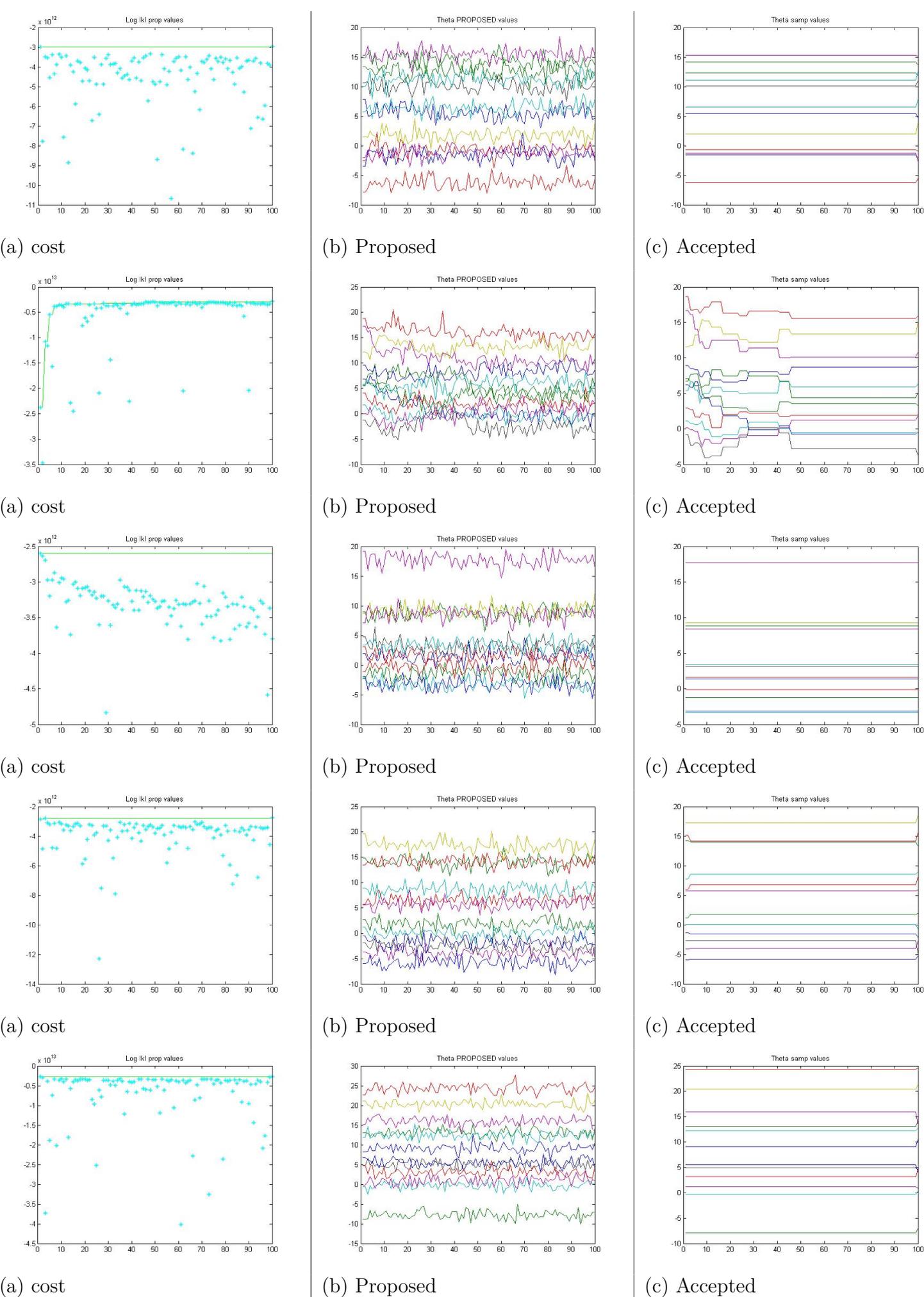
Figure A.3 Stage 3 training for trials 1-5

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted

(a) cost   (b) Proposed   (c) Accepted
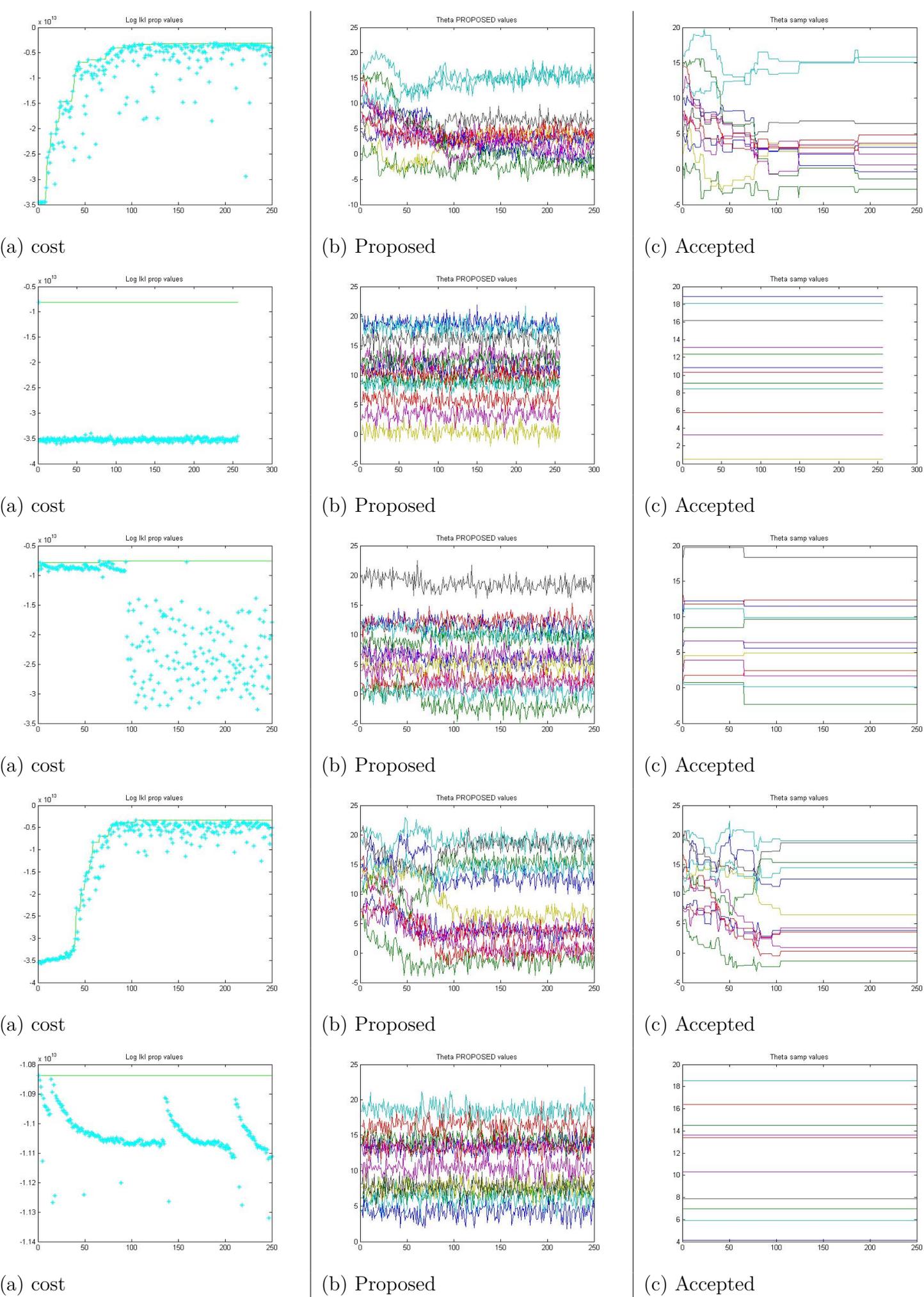
Figure A.4 Direct training cycles 1-5

49