Data-Driven Optimal Control of Wind Turbines Using Reinforcement Learning with Function Approximation

November 13, 2022

Abstract

We propose a reinforcement learning approach with function approximation for maximizing the power output of wind turbines (WTs). The optimal control of wind turbines majorly uses the maximum power point tracking (MPPT) strategy for sequential decision-making that can be modeled as a Markov decision process (MDP). In the literature, the continuous control variables are typically discretized to cope with the curse of dimensionality in traditional dynamic programming methods. To provide a more accurate prediction, we formulate the problem into an MDP with continuous state and action spaces by utilizing the function approximation in reinforcement learning. The commonly used pitch angle is selected as a control variable we are concerned with, which is regarded as the system state along with some other controllable and uncontrollable variables proven to affect the power output. Computational studies of real data are conducted to demonstrate that the proposed method outperforms the existing methods in the literature in obtaining the optimal power output.

Keywords: Markov decision process; reinforcement learning; function approximation; optimal control, wind turbines.

1. Introduction

Wind energy is considered one of the promising alternative energy sources because it is renewable, cost-effective, and environmentally friendly [1]. To effectively utilize wind energy

under the ever-changing wind profile, it is imperative to optimally control the operations of wind turbines [2, 3]. In this research, we propose to use a Markov decision process (MDP) to model the optimal control problem of wind turbines, in which a reinforcement learning (RL) algorithm with function approximation is applied to maximize the power output under the stochastic wind profile.

As a competitive energy source, the growth of wind power capability is evidenced by its drastic increase from 24GW to 591GW worldwide since 2001 [4]. According to the Global Wind Energy Council [4], more than 50GW wind capacity has been installed annually since 2014. Along with the rapid growth of wind energy capacity, there is also an increase in the size and power output of wind turbines, resulting from the economic advantages of large wind turbines. The utility-scale wind turbines, starting from a height of 24 meters and an output of 50kW, have become as large as 114-meter high with 5GW of power output [5]. The growing size of wind turbines makes this industry highly capital-sensitive, in which a small fraction of the decrease in power output and operation time can lead to a significant monetary loss. With the average price of electricity assumed to be around \$0.1 per kWh [6], even 1% energy loss on a 100MW wind turbine is estimated to reduce the annual revenue by \$307,500 [5]. In such a capital-intensive industry, owners of large wind turbines can benefit greatly by optimizing the operation and maintenance of wind turbines. The active optimal control is imperative for the cost-effective operation of wind turbines. For example, megawatt-scale wind turbines with a variable speed become particularly attractive as their operation can be actively controlled [7].

The optimal control of wind turbines can be majorly achieved by using the maximum power point tracking (MPPT), which maximizes the power output when the wind profile deviates within the operating range of the wind turbines [7]. In most optimization models used in existing data-driven techniques, however, the power output is maximized at a single time point, which is less practical due to the time lag between the observation of signals and the implementation of optimal decisions. Moreover, such models fail to take into account the correlation between consecutive control decisions, making it difficult to satisfy the

constraint on the maximum changing rate of control variables. As a sequential decision-making procedure, the MPPT of wind turbines can naturally fit into the model of Markov decision processes (MDPs). In this paper, we model the MPPT problem as an MDP where the decision made at the current epoch is evaluated in the environment state at the next epoch, capturing the correlation between consecutive control decisions over time (instead of optimizing for a single time point).

Reinforcement learning is a modern MDP-solving process that approaches large MDPs when exact methods become infeasible. Equipped with a set of modern approaches highlighted by techniques including temporal difference (TD) learning and function approximation, reinforcement learning can solve MDP problems where the underlying state-transition dynamics is unknown or the state and action spaces are extremely large [8]. In our study, we formulate the MPPT of WTs as an MDP problem that is solved using RL with function approximation, and then validate our approach on a real operational dataset. By utilizing the function approximation technique, the state values or state-action values are approximated using a function, and then we bootstrap from the previously approximated value functions to carry out DP iterations. To train the model with historical data, a fitted Q-iteration is used to solve the problem offline, which is important for online applications. Our proposed algorithm is guaranteed to converge, while the existing ANN-based algorithms cannot guarantee convergence.

When the state or action space is high-dimensional or continuous, it is difficult or impossible to evaluate a function at every possible state or state-action pair. To avoid this problem, function approximation is a method to approximate the value or action value function with a parametric or nonparametric function [9]. In this paper, we explore different function approximations such as the K nearest neighborhood regression, Gaussian process regression, and kernel regression on the state-action value function, or the Q-function, to find the optimal control rule with the undiscounted reward and an infinite horizon. A fitted Q-iteration algorithm is implemented in the framework of off-policy reinforcement learning. A case study on real operational data of wind turbines is conducted to show the capability of

the proposed method to maximize the power output. We also develop an evaluation model to assess the results obtained from different function approximations and to show the superiority of the obtained optimal control policy compared with the originally employed control policy.

In summary, the main contributions of this paper are as follows:

- The MPPT problem is modeled as an MDP that captures the correlation between consecutive control decisions over time (instead of optimizing for a single time point).
- The MDP problem with continuous state and action spaces is solved using reinforcement learning with function approximation.
- We introduce a new model-free, offline fitted Q-iteration algorithm to solve the MDP problem, which is guaranteed to converge.
- An evaluation model is developed to assess the results obtained from different function approximations including the K nearest neighborhood regression, Gaussian process regression, and kernel regression.
- We demonstrate the superiority of the proposed method to the current operating policy using real operational data of wind turbines.

The remainder of the paper is arranged as follows. In Section 2, we extensively review the literature on the existing approaches to MPPT followed by the elaboration of the research gap we address. In Section 3, we start with the introduction of the physical mechanisms of the wind turbines and then propose our RL-based MPPT model which maximizes the long-term power output by adjusting the pitch angle. Section 4 contains the results we obtained from analyzing the real operational data of wind turbines. In Section 5, we conclude our research and discuss possible extensions and future studies.

2. Literature Review

Different models and algorithms have been developed to optimize the power output in the MPPT with various control variables, such as the linear regression with polynomial features, and time-series models (e.g., Kalman filter) [10]. Before the implementation of the supervisory control and data acquisition (SCADA) system, the maximization of the power output is achieved by adjusting the rotor speed based on the power signal feedback [11]. With the recent deployment of the SCADA system [12], modern data-driven techniques (e.g., ANN, support vector machine and Gaussian process) have been well utilized to model and optimize the power output at each time point, along with other variables that are continuously monitored [10, 13, 14]. However, existing methods fail to consider the time lag between the observation of signals and the time of implementing the decision in controlling turbines, which is critical especially when the elapsed time between the decision epochs is not negligible. In addition, existing studies ignore the ever-changing stochastic wind profile that crucially influences the operation of turbines. In this research, we address these limitations by using a Markov decision process (MDP) to model the optimal control problem of wind turbines, in which a reinforcement learning (RL) algorithm with function approximation is applied to maximize the power output under the stochastic wind profile.

Existing literature has used the MDP to model the MPPT of wind turbines in which the dynamic programming (DP) approach was applied with the discretization of the control variables [15]. Dynamic programming has been developed as a solution to MDP problems with discrete and finite state and action spaces [16], which use a table to represent all state values (or state-action values) and are thus called tabular methods. DP methods can also be applied to some extensions of the original MDPs, such as semi-MDPs and partially observable MDPs [8]. Despite their higher efficiency than the exhaustive search over the policy space, DP methods cannot handle large, high-dimensional or continuous state and action spaces due to the "curse of dimensionality". The drawback is not alleviated until the recent introduction of reinforcement learning for solving the MDPs [9], which is also adopted

in this research.

There is already existing research that approaches the MPPT problem under the context of reinforcement learning [15, 17, 18]. To maximize the power output, Wei et al. used a tabular Q-learning algorithm to control the tip speed ratio of the turbine by discretizing the control variables [15], which inevitably introduces the discretization error. Wei et al. further approached the problem with continuous input and output variables using an ANN model-based Q-learning [17], which was later applied to yaw controlling by Saenz et al. [18]. The advanced actor-critic algorithm has been used for online pitch angle control [19, 20]. Although the ANN model-based Q-learning and the actor-critical algorithm performed well in the simulation study, they heavily rely on the online training process that is not practical in real-time applications. Consequently, these methods have not been applied to address the issues with real data, where online training is not available and the system dynamics is more complicated than the simulated environment. In this research, we propose to use the model-free reinforcement learning that can be trained offline on existing operational data.

3. Methodology

The optimal control of wind turbines concerns periodically adjusting the controllable variables (e.g., pitch angle, generator torque) according to the aerodynamic conditions (e.g., wind speed and direction) and the conditions of turbines monitored in real-time. It can be approached by using reinforcement learning techniques, where the problem is formulated as a Markov decision process. We first briefly introduce the physical mechanisms of wind turbines, and then describe our methodology including the preliminaries of MDP and reinforcement learning approaches with function approximation for maximizing the power output.

3.1 Physical Mechanisms of Wind Turbines

Wind turbines operate by capturing the kinetic energy in wind and transforming it into mechanical power, which is then used to generate electric energy by spinning a generator.

Wind turbines commonly consist of several main components including a rotor, a nacelle, a yaw system, a pitch system and a tower. The rotor and the nacelle are mounted on top of the tower, and the nacelle houses a set of gears and a generator [21]. The gears lie in the gearbox which connects to the rotor and the generator by the low-speed shaft and the high-speed shaft. The yaw system and the pitch system adjust the angles of the nacelle as well as the rotor blades, respectively, to align with the changes in wind direction.

The rotor, the gearbox, the low-speed shaft, the high-speed shaft and the generator together make up the drive train of the wind turbine. The drive train is the core component of the wind turbine as it is involved in the whole process of converting the wind power to the electric power. The rotor is made of the hub and the blades connected to it. When the wind passes through the rotor, it drives the rotor blades and the low-speed shaft connected to it to rotate. Then in the gearbox, the rotation of the low-speed shaft is converted to the rotation of the high-speed shaft, which directly drives the generator and generates electricity as it spins. To maximize the proportion of the wind power converted to the electric power, we usually need to change the rotational speed of the rotor according to the wind speed and adjust the pitch angle of the nacelle.

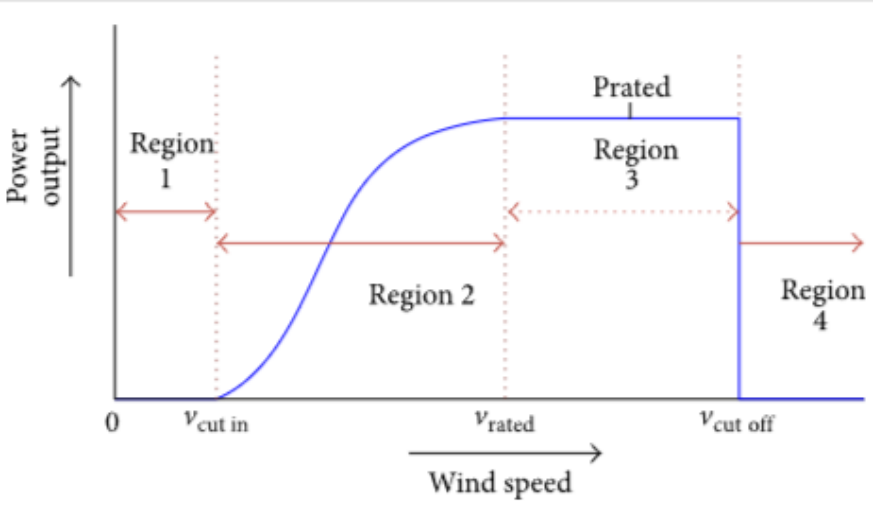

Figure 1: The power curve [22]

The relation between the power output of the WT and the wind speed can be briefly represented by the power curve in Figure 1. When the wind speed is below the cut-in speed,

$v_{cut\ in}$, the wind turbine does not spin and generates no power. When the wind speed is larger than $v_{cut\ in}$, the generated power increases with the wind speed. When the wind speed is between the rated wind speed, v_{rated}, and the cut-off wind speed, $v_{cut\ off}$, the power output stays at the standard power output point, P_{rated}. The wind turbine stops working when the wind speed is higher than $v_{cut\ off}$. An accurate formula describing the power output of a wind turbine is [23, 5, 13, 2, 24]

$$P = P_{wind} C_p(\lambda, \beta) = \frac{1}{2} \rho \pi R^2 v^3 C_p(\lambda, \beta) \tag{1}$$

where P_{wind} is the theoretical wind power available to a turbine, ρ is the air density, R is the rotor radius, and v is the wind speed before passing the rotor. $C_p(\lambda, \beta)$ is the power coefficient that evaluates the proportion of available wind power captured by the wind turbine, which is a nonlinear function of the blade pitch angle β and the tip-speed ratio $\lambda = \omega_r R/v$ with ω_r being the rotational speed of the rotor [13, 23, 24, 25, 1]. An empirical expression of $C_p(\lambda, \beta)$ proposed in the literature is [26, 24, 25]

$$C_p(\lambda, \beta) = \frac{1}{2}(\lambda - 0.022\beta^2 - 5.6)e^{-0.17\lambda} \tag{2}$$

which is theoretically bounded by the Betz limit $C_{p,max} = 0.593$ [24]. Eq. (2) implies that we can control the power output by controlling the pitch angle, β and the tip-speed ratio, λ. In practice, the pitch angle can be changed directly for variable pitch wind turbines [1], while the tip-speed ratio cannot be controlled directly and is often adjusted through the generator torque [27]. As the generator torque control is often carried out by electrical-mechanical feedback systems, we only investigate the control of pitch angle using the RL model in this research [3].

3.2 MDP and Reinforcement Learning

A stationary MDP is characterized by a quintuple $\{\mathcal{T}, \mathcal{S}, \mathcal{A}_s, p(\cdot|s,a), r(s,a)\}$ consisting of a set of decision epochs $\mathcal{T}$, a state space $\mathcal{S}$, an action space $\mathcal{A}_s$ under state s, a stochastic transition function p, and a reward function r [28]. At each decision epoch, an action available for the current state s is selected and an instant reward $r(s,a)$ is received. The probability distribution of the next state $p(\cdot|s,a)$ completely depends on the current state and action, which is a core assumption of MDPs.

In an MDP model, a controller or agent seeks to find a policy $\pi : \mathcal{S} \to \mathcal{A}$, that maximizes a certain value criterion related to the reward. A commonly used criterion is the expected total discounted reward [28]

$$J_\infty^\pi(s) = \lim_{h\to\infty} \sum_{t=0}^{h} \mathbb{E}\left[\gamma^t r(s_t, \pi(s_t)) \middle| s_0 = s\right], \tag{3}$$

where γ is the discount factor. When the reward represents the revenue, it is proper to choose γ as the reciprocal of the risk-free rate. Such a discounted reward is referred to as the value function, or the V-function, of the state s under the policy π, denoted by v_π. The goal of MDP is to find an optimal policy π^* such that

$$v_{\pi^*}(s) \geq v_\pi(s) \quad \forall s \in \mathcal{S}, \pi.$$

In most cases, we write $v_{\pi^*}(s)$ as $v_*(s)$ for brevity. The value function under the optimal policy should satisfy the Bellman optimality equation, a central property of MDPs [9]:

$$v_*(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')].$$

In general, the optimization problem using the V-function is computationally intractable, unless an explicit model is assumed. The machine learning approach usually does not make any assumption about models. Instead, the problem is typically tackled using the state-

action value function, or the Q-function defined as

$$q_\pi(s,a) = \lim_{h\to\infty} \sum_{t=0}^{h} \mathbb{E}_\pi[\gamma^t r(s_t, \pi(s_t))|s_0 = s, a_0 = a].$$

The Bellman optimality equation characterizes the optimal q_* when an optimal control policy π^* is achieved [9]

$$q_*(s,a) = \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')], \tag{4}$$

where q_* is defined as the optimal Q-function achieved at π^*.

In this research, we maximize the long-term average power output without the existence of the discount factor as the average power output is what we are concerned about, which is equivalent to maximizing the long-term average reward defined as [9]

$$R^\pi(s) = \lim_{h\to\infty} \frac{1}{h} \sum_{t=0}^{h} \mathbb{E}\left[r(s_t, \pi(s_t))|s_0 = s\right]. \tag{5}$$

The relationship between the expected total discounted reward $J^\pi_\infty(s)$ in Eq. (3) and the long-term average reward $R^\pi_\infty(s)$ in Eq. (5) is given by [9]

$$J^\pi_\infty(s) = \frac{1}{1-\gamma} R^\pi(s). \tag{6}$$

Therefore, the results from modeling the MPPT of wind turbines as an MDP with a discounted reward can be readily used to obtain the optimal control policy and the optimal long-term average reward or power output.

3.3 MDP for WT Optimal Control

In this section, we formulate the MDP for the operation control problem of wind turbines, including the state space, action space, and reward function. In the MDP formulation of MPPT, actions are determined at fixed decision epochs given the state at the epoch. The

wind profile, power output and control variables in the previous decision epoch can be taken as the state of the current decision epoch, which is then used to determine the control variables at the current decision epoch. The overall MDP framework for MPPT is illustrated in Figure 2.

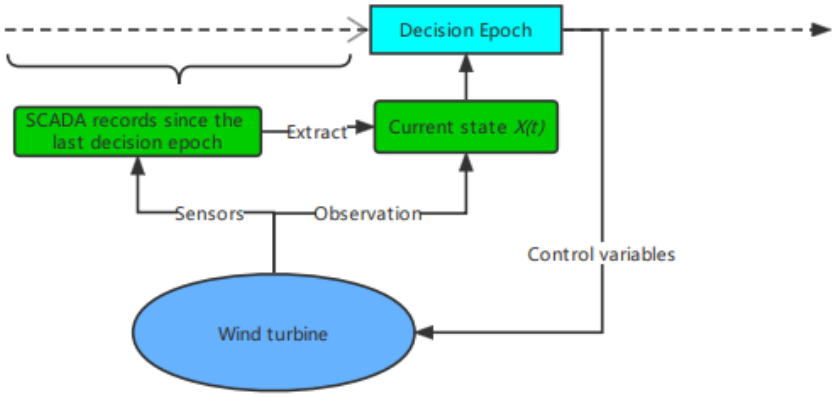

Figure 2: The MDP framework for MPPT.

The information collected from the SCADA system covers a wide range of variables, such as the rotation speed of shafts and bearings, the vibration measurements of components, the electrical measurements, the component temperature, and the wind speed and direction, among others [12]. Among these variables, some of them are the physical states that can be measured and controlled (e.g., the rotation speed of shafts and bearings, the pitch angle, the vibration measurements of components and the electrical measurements), and other variables are called the exogenous states that can be measured but are uncontrollable (e.g., the component temperature, the wind speed and direction). For the state space in our MDP, we choose the ones that are closely related to the power output, namely, the pitch angle representing the physical state, and the wind profile as the exogenous state [12]. The pitch angle is selected because it affects the power coefficient in Eq. (2), while the wind profile is selected as the wind speed directly determines the power output as shown in Eq. (1) [13]. The current power output is included in our state space, due to the autocorrelation in consecutive power output records [13].

Therefore, the state space of our MDP model is $s = (s_1, s_2, s_3, s_4, s_5)$ where s_1 denotes

the pitch angle measured in degrees, s_2 is the generator torque in Nm, s_3 is the average wind speed in m/s, s_4 is the corrected absolute wind direction in degrees, and s_5 is the power output in the previous epoch. For the action space, we only consider the pitch angle denoted by a_1, which represents the pitch angle in the next step in degrees. To avoid the discretization error, all the variables in the state and action spaces are kept continuous as they are measured, instead of being discretized, which is achieved by applying function approximation to the state-action value function.

MPPT aims to maximize the power output in the long term. Therefore, for each action, the reward is measured by the average power output generated in the next epoch. For the control of wind turbines, it is not practical to change the pitch angle constantly in time, since it can cause the machine subject to unnecessary stress. Therefore, we consider that control actions are taken at discrete times, e.g., every hour, every day. In this study, the decision epoch is chosen to be one hour to balance between timely adjusting control variables and preventing excessive stress on equipment. Our RL model is a model-free method, which means we do not rely on the transition function $p(\cdot|s, a)$ and the reward function $r(s, a, s')$ to carry out the optimization. Instead, we use sample transitions in the dataset and the instant reward is defined as the power output at the next decision epoch $r(t) = s_5(t + 1)$.

3.4 Reinforcement Learning with Function Approximation

As both state space and action space are continuous, we consider using function approximation in reinforcement learning, where the Q-function is approximated. The method we introduce is also an offline learning method, which is important for capital intensive industries such as wind turbines because online training will be too expensive. In this approach, the state-action function, or Q-function $Q(s, a)$ receives continuous state and action values, instead of a table of values on finite state-action pairs. This setting allows us to keep the continuous sensor data and avoid the discretization error when estimating $Q(s, a)$ from existing samples.

One of the most straightforward algorithms to handle the approximated Q-function is

the fitted Q-iteration algorithm [29]. In the fitted Q-iteration, the approximation function can be in any form that approximates the state-action value function $\hat{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The algorithm assumes a greedy policy, given a training set (s_i, a_i, r_i, s_i') for $i = 1, \ldots, N$. In each iteration, we first estimate the Q-function for each training quadruple from the current approximation according to the Bellman equation

$$q_i \leftarrow r_i + \gamma \max_{a \in \mathcal{A}} \hat{Q}(s_i', a), \tag{7}$$

where γ is the discount factor. Then we update the function approximation with the new estimated Q-function value q_i for the training set. This process can be viewed as an operator $\hat{H}$ imposed on the function approximation $\hat{Q}$:

$$\hat{Q}_{n+1} = \hat{H}\hat{Q}_n. \tag{8}$$

When $\hat{Q}$ is in the form of a kernel regression described in Table 1 with the same kernel in each iteration, $\hat{H}$ is a contraction on the Banach space defined over $\mathcal{S} \times \mathcal{A}$ and the supremum norm, which guarantees $\hat{Q}$ to converge to a unique fixed point of $\hat{H}$ [29]. The whole algorithm is described in Algorithm 1.

Algorithm 1: Fitted Q Iteration Algorithm [29]

Input *The observation set* $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$*, and the maximum number of iterations* $MaxIter$*.*
Initialization: $n \leftarrow 0$, $\hat{Q}_0(s, a) \equiv 0$.
while $n < MaxIter$ *and* $\hat{Q}_n(s, a)$ *does not converge,* **do**
  $n \leftarrow n + 1$.
  $q_i \leftarrow r_i + \gamma \max_{a \in \mathcal{A}} \hat{Q}_n(s_i', a)$ for $i = 1, \ldots, N$.
  Obtain $\hat{Q}_{n+1}(s_i, a)$ according to q_i, for $i = 1, \ldots n$.
Output *The optimal Q-function* $\hat{Q}(s, a|\theta^*)$.

Since q_i in Eq. (7) is essentially the Bellman optimality equation in Eq. (4) at the sample transition (s_i, a_i, r_i, s_i'), the unique fixed point for $\hat{H}$ is also the unique solution to the Bellman optimality equation in (4). Therefore, the algorithm can find the policy that maximizes $J_\infty^\pi(s)$ [29]. In the discounted case with an infinite horizon, the algorithm converges to the optimal

policy that is a stationary point of the Bellman equation for qualified models. The detailed conditions are described in [9]. We can do the fitted Q-iteration according to Eq. (6) to reach the optimal policy and calculate the optimal long-term average reward. The choice of the discounted factor γ is validated in our numerical experiment.

3.5 Kernel Regression Function Approximation

The crucial element of the aforementioned fitted Q-iteration algorithm is the function approximation $\hat{Q}$. Without other restrictions, $\hat{Q}$ can take any parametric or non-parametric form, ranging from regression, and kernel models, to neural networks. However, when we switch from a tabular Q to function approximation $\hat{Q}$, the convergence property is no longer guaranteed for most forms of $\hat{Q}$ and their performances vary in terms of speed, stability and optimality. In our case, a kernel regression is adopted for its guaranteed convergence property, which is essential for estimating $\hat{Q}$ [30]. Theoretically, we can prove that the algorithm converges to $\hat{Q}(s, a; \theta^*)$ if it satisfies [30, 29]

$$Q(s,a) = \frac{\sum_{i=1}^{N} k((s,a),(s_i,a_i))Q(s_i,a_i)}{\sum_{i=1}^{N} k((s,a),(s_i,a_i))}.$$

The function approximation $Q(s, a)$ of this form is called kernel regression, and $k(x, x_i)$ is the kernel function.

In general, consider samples $\{(x_i, y_i)\}_{i=1}^{N}$, the kernel regression function $m_N(x)$ takes the form [31]

$$m_n(x) = \frac{\sum_{i=1}^{n} y_i k\left(x, x_i\right)}{\sum_{i=1}^{n} k\left(x, x_i\right)},$$

where $k : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$ is the kernel function we mentioned above. Common kernel functions include the uniform kernel, Epanechnikov kernel [32], Gaussian kernel, quadratic kernel [33] and tricube kernel [34]. In this research, we compare the performance of four different kernel functions for the function approximation: linear kernel, Laplacian kernel, Gaussian kernel, and quadratic kernel. The formulas of these kernels are given in Table

2. For the Gaussian kernel in the table, the parameter $u = \|x_i - x_j\|_2/h$ where h is the bandwidth of the kernel. When it comes to a kernel that is a function of u, it requires such a bandwidth that can be selected using Scott's rule of thumb [35].

In addition to these four kernels, we also consider the K-nearest neighborhood regression (KNN) and the linear regression as kernel regressions in the function approximation. Although different model selection and bandwidth selection methods exist for kernel regressions [36, 37], we can only compare our final fitted Q-iteration results because we have no sample for the true value of $Q(s, a)$.

Name	Formula
Linear kernel	$k(x_i, x_j) = x_i^T x_j$
Laplacian kernel	$k(x_i, x_j) = -\exp(\|x_i - x_j\|_1)$
Gaussian kernel	$k(x_i, x_j) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$
Quadratic kernel	$k(x_i, x_j) = (\gamma x_i^T x_j + 1)^2$

Table 1: Kernels considered in function approximation [32, 33, 36, 37]

4. Case Studies

4.1 Data Description

In this research, we consider the Senvion MM82 2.05MW wind turbines that have large turbines with a rotor diameter of 82m and a hub height of 80m [38]. The technical specifications for this turbine indicate that the allowed wind speed lies between the cut-in wind speed of 3.5 mph and the cut-off wind speed of 25 mph, i.e., $s_3 \in (3.5, 25)$ [38]. Although the ranges for other variables are not provided in the technical specifications, they can be estimated from historical data. The pitch angle ranges from -1 to 92.5 degrees in the dataset, i.e., $s_1 \in [-1, 92.5]$. The generator torque can be as low as zero when the wind speed is low, and it usually does not go beyond 6,000Nm, i.e., $s_2 \in [0, 6000]$. Finally, the wind can come from any direction and we have $s_4 \in [0, 360]$. As the power output should not exceed

2,050kW, we have $s_5 \in [0, 2050]$. The range of action a_1 is specified according to the current state to make sure that the next state lies in the normal range.

The SCADA data we analyzed in this research were collected by ENGIE at the La Haute Borne wind farm located in Meuse, France for 25,000 hours [39]. Specifically, we selected four turbines with the ID numbers R80711, R80721, R80736 and R80790. Due to the storage and IO limit, we are only able to access the maximum, minimum, and average values of the signals in a 10-minute interval, although the original data were collected at a higher frequency.

During the operations of wind turbines, it is not practical to take real-time actions that constantly change the control variables, due to machine stress induced by the adjustment. Therefore, we use one hour as the decision epoch in this research and take the average value in a one-hour interval as the observed value for each variable.

4.2 Preliminary Analysis

A preliminary analysis is conducted on the WT operation data to help select the training data and the control variables. We first analyze and visualize some SCADA variables that are related to the physical model including the power output, the power coefficient, and the generator torque. According to Eq. (1), we can calculate the theoretical power available to the turbine, P_{wind}, where the air density, ρ, is approximately $1.175 kg/m^3$ at 411m above the sea level. The power output P, the wind power P_{wind} and the power coefficient $C_p(\lambda, \beta)$ are plotted for a sample window of 450 hours in Figure 3. As shown in Figure 3, the power output P is below the available wind power, P_{wind}, and the power coefficient lies in a reasonable range below the Betz's limit, which indicates the validity of the dataset.

Then we investigate the relationship between the wind speed and the power output of the whole dataset to verify the dataset for our study. As shown in Figure 4, at some time points such as the range between around 20,000 hours and 24,500 hours, the power output keeps zero where the wind speed is normal. It is reasonable to assume that the corresponding turbine was not operating during those time points. Therefore, we exclude these data from

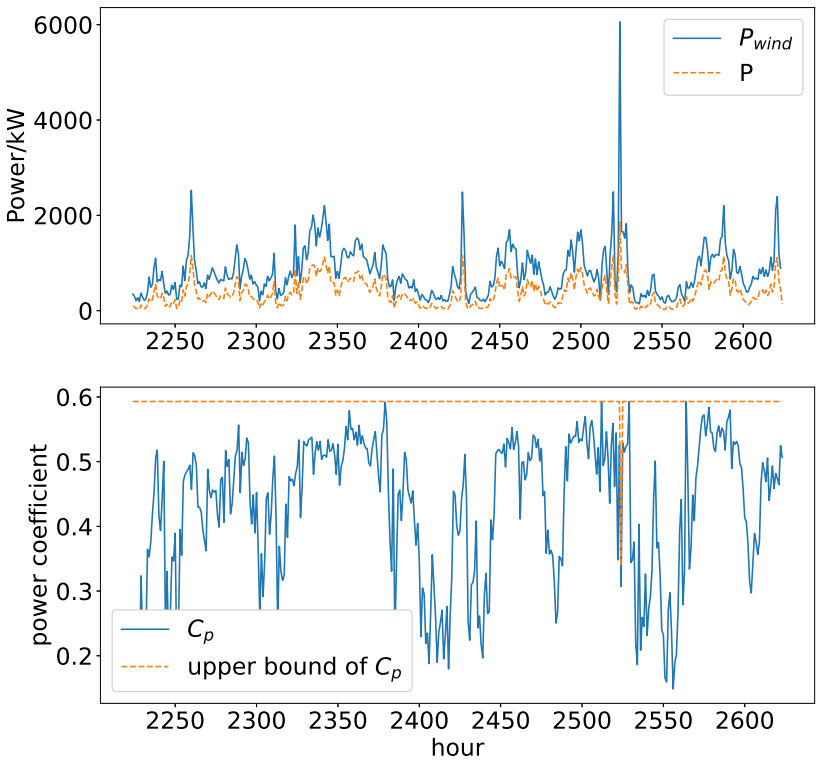

Figure 3: The wind power, power output and power coefficient in a sample window of 450 hours

our analysis and modeling.

4.3 Evaluation Model

As different forms of function approximation in the fitted Q-iteration may give us different results in terms of the optimal control policies and the power output, we need an approach to evaluate the performance of different RL models developed with different function approximations. The power output can also serve as the reward function $r(s, a, s')$ in our MDP model, which evaluates the instant reward based on the full state transition. Without access to wind turbines, we decide to develop an independent model to evaluate their performance which can be described as

$$f = f(s_1, s_3, s_4, s_5, a_1, s_3', s_4').$$

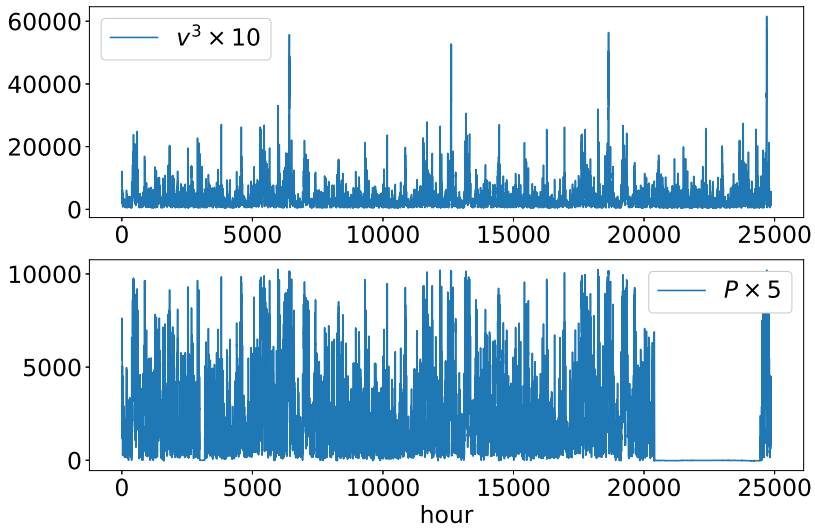

Figure 4: v^3 and power output of the whole dataset

Rather than the Q function that only depends on the current state and action, our evaluation model also takes into account the wind profiles at the next decision epoch which is not available in a control model. The model for performance evaluation uses the wind profile at time t to predict the power output at time t. According to [13], a good model can be obtained with the aforementioned predictors with an artificial neural network (ANN) model. Then with the evaluation model, we can evaluate the performance based on the existing operational data without simulation on real turbines using our new control policy. We considered linear regression, random forest, Gaussian process regression, KNN and multilayer perceptron when selecting an evaluation model. The final evaluation model will be chosen according to its out-of-sample predictive power.

After the preliminary analysis, we continue to develop a reliable model for evaluating our results. We intend to select an appropriate evaluation model that has the best predictive performance on the power output. Particularly, we use the first 80% sample data as our training data, and the modeling performance is evaluated on the last 20% test data. We consider four metrics of the test data to measure the predictive performance, namely, the mean squared error (MSE), mean absolute error (mean AE), maximum absolute error (max

AE) and minimum absolute error (min AE). In order to predict $P(t)$, the power output at time t, the inputs of our model include the wind speed and direction at t and $t-1$, $v(t), v(t-1), \alpha_\omega(t), \alpha_\omega(t-1)$, the pitch angle at t and $t-1$, $\alpha_p(t), \alpha_p(t-1)$, and the power output at t, $P(t-1)$. As given in Eq. (1), we also include $v^3(t)$ and $v^3(t-1)$ in the analysis which proves to improve the results significantly. The performance of linear regression (LR), k-nearest neighborhood (KNN), random forest (RF), Gaussian process regression (GPR) and multi-layer perceptron (MLP) is listed in Table 2. For Gaussian process regression, we considered different basis functions and kernel functions. Among these models, the GPR without the v^3 feature achieves the best result in almost all metrics. It achieves the least MSE which is half as much as that of the RF mode. It also obtains the least mean AE and Max AE which indicates the high robustness of the model. The selected GPR model is a Gaussian process over the predictors with a constant mean $\beta = 672.56$ and an radial basis function kernel $k(s, s') = \sigma_f^2 \exp\left(-\|s - s'\|_2/(2\sigma_l^2)\right)$, whose kernel parameters are $\sigma_l = 3594.68$ and $\sigma_f = 2089.41$.

Predictors	Model	MSE	Mean AE	Max AE	Min AE
No v^3	LR	197.74	10.97	94.88	0.0057
	KNN(k=4)	84.04	6.36	117.55	0.00043
	GPR(constant basis, rbf ker)	22.19	3.21	39.90	0.0020
	RF(n=93)	45.24	4.71	45.18	0.000002
	3-hidden layer of size 100 MLP	574.96	14.56	166.06	0.0047
With v^3	LR	197.543	11.01	91.62	0.0013
	KNN(k=4)	87.13	6.32	117.55	0.0017
	GPR(constant basis, matern52 ker)	23.12	3.25	48.77	0.00037
	RF(n=12)	46.04	4.72	51.23	0.005
	3-hidden layer of size 100 MLP	190.34	9.79	125.18	0.0099

Table 2: Out-of-sample performance of the evaluation models

4.4 Fitted Q-Iteration Results

To verify whether the linear model is appropriate for our function approximation, we start with conducting fitted Q-iterations with different values of γ by using a linear model in this experiment. As the sum of the absolute change in q_i, the convergence threshold does not exceed 1,000. When $\gamma = 0.4, 0.6$ and 0.8, the estimated long-term average rewards obtained in Eq. (5) are 465.31 kW, 467.40 kW and 468.00 kW, respectively. The results are

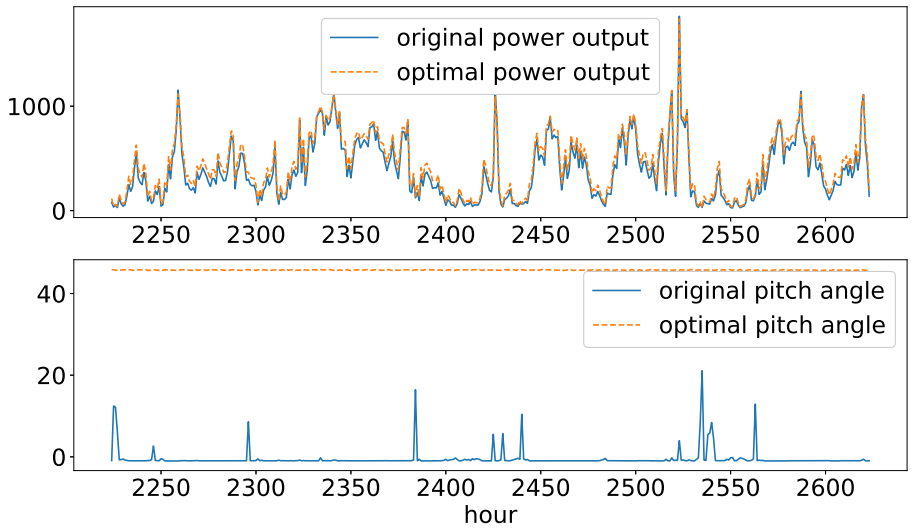

Figure 5: The optimal power output and optimal control (linear kernel)

about a few kilowatts higher than the observed average reward of 460.11kW. The optimal action is to minimize the pitch angle (-1), which is a natural optimal condition for a linear model. However, under this control policy, the mean average power output is smaller than the original power output, which indicates that the linear model is not appropriate for our function approximation.

We then carry out the fitted Q-iteration algorithm on the aforementioned function approximation in Table 1 including Gaussian kernel regression, Laplacian kernel regression, linear kernel regression, quadratic kernel ridge regression (KRR) and KNN. A snapshot of the optimal power output and optimal control is shown for each of the five kernels in Figure 5-9, respectively. In the figures, the x-axis denotes the time step in hours. The y-axis is the angle in degrees for the pitch angle, and the original and the optimal power output given by our evaluation model.

For comparison, the corresponding optimal average power outputs calculated using our evaluation model are shown in Table 3, where the reference average power output is the original average power output of 460.11 kW. The percentage of improvement is also provided in the last column of Table 3. From the last column, we can see that the optimal policies

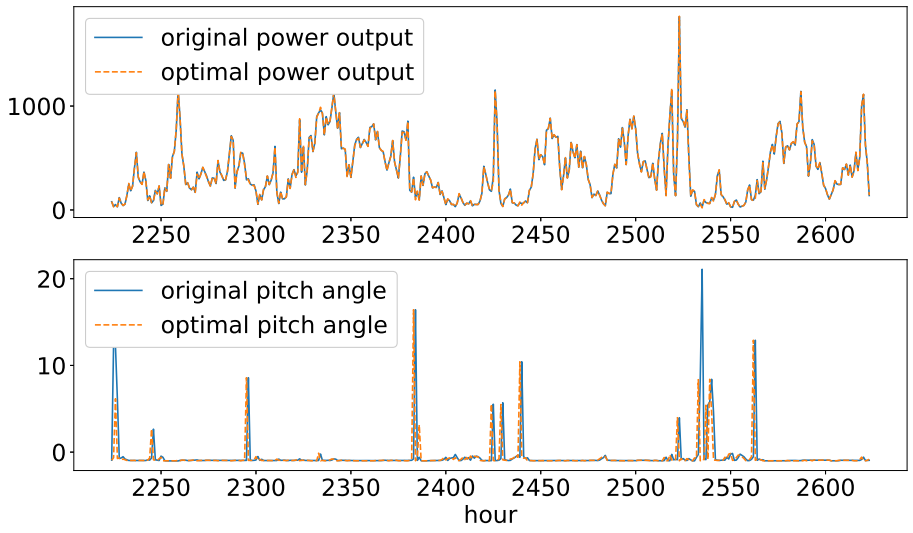

Figure 6: The optimal power output and optimal control (Gaussian kernel)

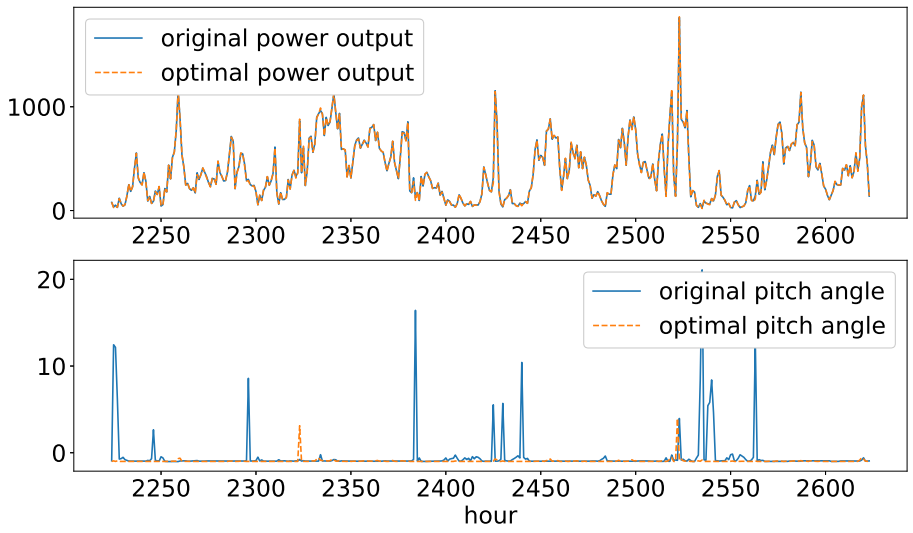

Figure 7: The optimal power output and optimal control (Laplacian kernel)

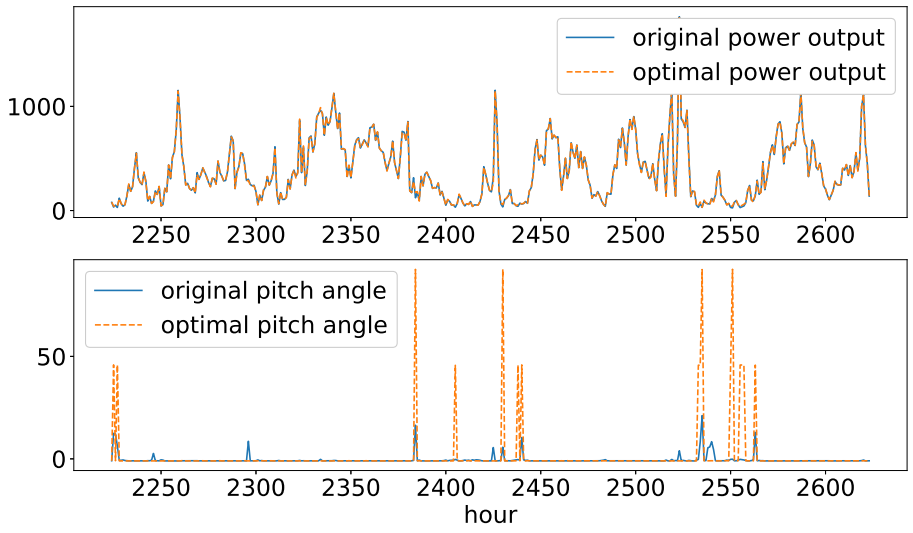

Figure 8: The optimal power output and optimal control (Quadratic kernel)

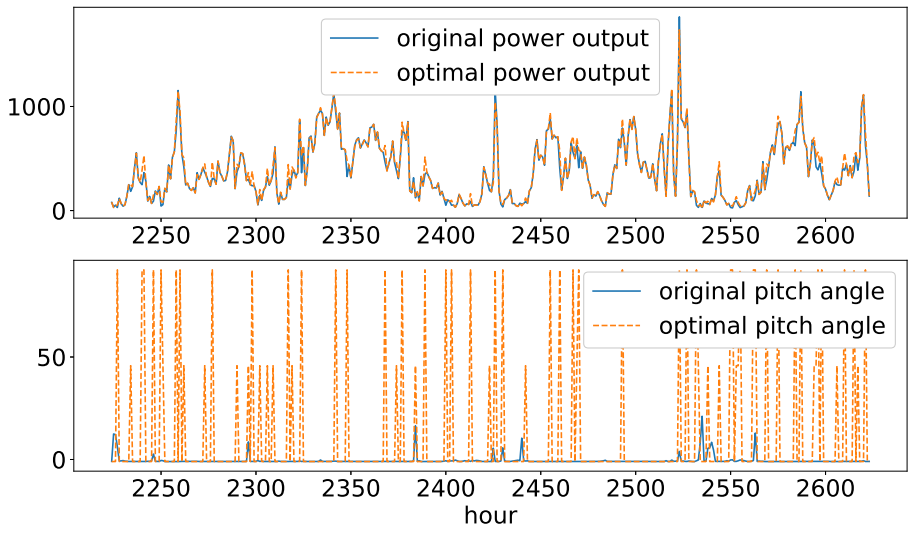

Figure 9: The optimal power output and optimal control (KNN regressor with $k = 4$)

Predictor	Optimal Average Power Output (kW)	Improvement(%)
KRR with linear kernel	509.93	10.83
KRR with Gaussian kernel	460.12	0.002
KRR with Laplacian kernel	459.81	-0.065
KRR with Quadratic kernel	460.48	0.080
KNN when k=4	469.36	2.01
Reference	460.11	-

Table 3: Results from Different Predictors of Function Approximation

obtained by the fitted Q-iteration with most function approximation predictors have higher average power output than the reference value of 460.11kW, except for the one from the KRR with Laplacian kernel. Overall, KRR function approximations with Gaussian, Laplacian and Quadratic kernels provide nearly the same results as the reference. This is due to their tendency to make the same control decision as the training data that can be seen from Figure 6-8. The KRR function approximation with a linear kernel shows the most significant improvement in the optimal average power output, and the KNN function approximation also improves the power output. Meanwhile, these two function approximation predictors provide more radical optimal policies, judging from Figure 5 and Figure 9. The benefit of KNN and KRR with linear kernel can be explained by their relatively simple structures, which prevent them from the local optima during the optimization steps. In conclusion, a linear kernel should be adopted in the MPPT problem.

5. Discussion and Conclusions

In this research, we maximize the power output of wind turbines under the stochastic wind profile by formulating the problem as a Markov decision process with continuous state and action spaces. As exact methods become infeasible for the large MDPs, we utilize the function approximation in reinforcement learning to overcome the curse of dimensionality of DP methods. In computational studies using real data, we derive the optimal control policy of the pitch angle by applying the fitted Q-iteration algorithm to the MPPT task of operating wind turbines with a linear kernel. We also use a GPR model for evaluating our results of MDP using all available information, which achieves a high precision in predicting

the power output compared with previous results using the ANN. The evaluation model demonstrates the superiority of the optimal control policy obtained by our RL algorithm.

The proposed algorithms using the KNN function approximation and the KRR function approximation with a linear kernel achieve 2% and 10% improvement on the optimal average power output over the operational records, respectively, which is competitive compared with the existing methods on power output [13, 14]. Considering that the existing methods do not take account of the lag between the observation of signals and the time of implementing the decision in controlling turbines, the performance of the proposed method has demonstrated its practical benefits compared with previous methods.

One issue with the MDP formulation lies in the delay in the consecutive decision-making and control embedded in the model, as the control variables in the next decision epoch are determined based on the state at the current decision epoch. For future research, we can generalize the reinforcement learning to the online version with policy gradient techniques, if the experimental wind turbines are accessible or a system dynamics model is available [9]. Furthermore, instead of the linear model, more complex models can be used for function approximation, such as non-linear models, supervised learning, or a neural network method.

References

[1] Eduardo José Novaes Menezes, Alex Maurício Araújo, and Nadège Sophie Bouchonneau da Silva. A review on wind turbine control and its associated methods. *Journal of cleaner production*, 174:945–953, 2018.

[2] Majid A Abdullah, AHM Yatim, Chee Wei Tan, and Rahman Saidur. A review of maximum power point tracking algorithms for wind energy systems. *Renewable and sustainable energy reviews*, 16(5):3220–3227, 2012.

[3] Shilpa Mishra, Sandeep Shukla, Nitin Verma, et al. Comprehensive review on maximum power point tracking techniques: wind energy. In *2015 Communication, Control and Intelligent Systems (CCIS)*, pages 464–469. IEEE, 2015.

[4] Karin Ohlenforst, Steve Sawyer, Alastair Dutton, Ben Backwell, Ramon Fiestas, Joyce Lee, Liming Qiao, Feng Zhao, and Naveen Balachandran. *GLOBAL WIND REPORT 2018*. 04 2019.

[5] Lucy Y Pao and Kathryn E Johnson. Control of wind turbines. *IEEE Control systems magazine*, 31(2):44–62, 2011.

[6] U.S. Energy Information Administration. U.s. electricity price sept. 2019. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a, 2019.

[7] Jackson G Njiri and Dirk Soeffker. State-of-the-art in wind turbine control: Trends and challenges. *Renewable and Sustainable Energy Reviews*, 60:377–393, 2016.

[8] Shenglin Peng et al. Reinforcement learning with gaussian processes for condition-based maintenance. *Computers & Industrial Engineering*, 158:107321, 2021.

[9] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[10] Debashisha Jena and Saravanakumar Rajendran. A review of estimation of effective wind speed based control of wind turbines. *Renewable and Sustainable Energy Reviews*, 43:1046–1062, 2015.

[11] Kazmi Syed Muhammad Raza, Hiroki Goto, Hai-Jiao Guo, and Osamu Ichinokura. A novel algorithm for fast and efficient maximum power point tracking of wind energy conversion systems. In *2008 18th international conference on electrical machines*, pages 1–6. IEEE, 2008.

[12] Jannis Tautz-Weinert and Simon J Watson. Using scada data for wind turbine condition monitoring–a review. *IET Renewable Power Generation*, 11(4):382–394, 2016.

[13] Andrew Kusiak and Haiyang Zheng. Optimization of wind turbine energy and power factor with an evolutionary computation algorithm. *Energy*, 35(3):1324–1332, 2010.

[14] T Li, AJ Feng, and L Zhao. Neural network compensation control for output power optimization of wind energy conversion system based on data-driven control. *Journal of Control Science and Engineering*, 2012, 2012.

[15] 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*, 62(10):6360–6370, Oct 2015.

[16] Henk C.Tijms. *A First Course in Stochastic Models*. John Wiley & Sons, Ltd, 2004.

[17] 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*, 31(11):7837–7848, 2016.

[18] Aitor Saenz-Aguirre, Ekaitz Zulueta, Unai Fernandez-Gamiz, Javier Lozano, and Jose Manuel Lopez-Guede. Artificial neural network based reinforcement learning for wind turbine yaw control. *Energies*, 12(3):436, 2019.

[19] M Sedighizadeh and A Rezazadeh. Adaptive pid controller based on reinforcement learning for wind turbine control. In *Proceedings of world academy of science, engineering and technology*, volume 27, pages 257–262. Citeseer, 2008.

[20] Q. Bin, L. Pengcheng, W. Xin, and Z. Wanli. Pitch angle control based on renforcement learning. In *The 26th Chinese Control and Decision Conference (2014 CCDC)*, pages 18–21, 2014.

[21] Elaheh Taherian-Fard, Ramin Sahebi, Taher Niknam, Afshin Izadian, and Mokhtar Shasadeghi. Wind turbine drivetrain technologies. *IEEE Transactions on Industry Applications*, 56(2):1729–1741, 2020.

[22] Vaishali Sohoni, SC Gupta, and RK Nema. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. *Journal of Energy*, 2016, 2016.

[23] Boubekeur Boukhezzar, Houria Siguerdidjane, and M Maureen Hand. Nonlinear control of variable-speed wind turbines for generator torque limiting and power optimization. 2006.

[24] Alfred Wanyama Manyonge, RM Ochieng, FN Onyango, and JM Shichikha. Mathematical modelling of wind turbine in a wind energy conversion system: Power coefficient analysis. 2012.

[25] Yuanye Xia, Khaled H Ahmed, and Barry W Williams. Wind turbine power coefficient analysis of a new maximum power point tracking technique. *IEEE transactions on industrial electronics*, 60(3):1122–1132, 2012.

[26] PM Anderson and Anjan Bose. Stability simulation of wind turbine systems. *IEEE transactions on power apparatus and systems*, (12):3791–3795, 1983.

[27] Brian Ray Resor, David Charles Maniaci, Jonathan Charles Berg, and Phillip William Richards. Effects of increasing tip velocity on wind turbine rotor design. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2014.

[28] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[29] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

[30] Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2):161–178, 2002.

[31] László Györfi, Michael Kohler, Adam Krzyzak, Harro Walk, et al. *A distribution-free theory of nonparametric regression*, volume 1. Springer, 2002.

[32] VA Epanechnikov. Nonparametric estimates of a multivariate probability density. theor. *Probab. Appl*, 14:153–158, 1969.

[33] William S Cleveland and Susan J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.

[34] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[35] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[36] Michiel Debruyne, Mia Hubert, and Johan AK Suykens. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9(10), 2008.

[37] Max Köhler, Anja Schindler, and Stefan Sperlich. A review and comparison of bandwidth selection methods for kernel regression. *International Statistical Review*, 82(2):243–274, 2014.

[38] Senvion S.A. Mm82 wind turbine 2mw. https://www.senvion.com/global/en/products-services/wind-turbines/mm/mm82/, 2019.

[39] ENGIE. Engie opendata. https://opendata-renewables.engie.com/explore/index, 2019.