DROP: Deep Reinforcement Learning Based Optimal Perturbation for MPPT in Wind Energy

Salman Sadiq Shuvo
Electrical Engineering
University of South Florida
Tampa, FL, USA
salmansadiq@usf.edu

Md Maidul Islam
Electrical Engineering
Florida State University
Tallahassee, FL, USA
mi19b@my.fsu.edu

Yasin Yilmaz
Electrical Engineering
University of South Florida
Tampa, FL, USA
yasiny@usf.edu

2022 North American Power Symposium (NAPS) | 978-1-6654-9921-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/NAPS56150.2022.10012250

***Abstract*—The fluctuating nature of wind energy has inspired researchers to look for a fast, efficient Maximum Power Point Tracking (MPPT) algorithm. The MPPT method aims to harness maximum power at varying wind speeds by adjusting rotor speed. Our contribution to the wind MPPT task is twofold. First, we use a predictive model to map the current operating point of the turbine speed and output power to the optimal operating point (i.e., optimal turbine speed for maximum output power). Second, we propose a Deep Reinforcement Learning based solution that provides adaptive speed control to reach the MPP fast and precisely. Our experimental results demonstrate the superior performance of our method compared with the existing techniques.**

***Index Terms*—Deep reinforcement learning, markov decision process, MPPT, renewable energy, wind power.**

I. Introduction

A. Wind Energy

The increasing cost and adverse effects of fossil fuels on climate have increased the demand for an efficient renewable energy alternative. Wind energy can be a great source of clean and reliable energy, and there has been a rapid penetration of wind generators in modern power systems in the last decade. The global wind power generation capacity is expected to reach 840 GW by the end of 2022 [1]. The basic nature of wind energy is extremely fluctuating, and thus tracking of maximum power point (MPP) to extract maximum capture of energy at different varying wind speeds is of great interest.

B. Wind MPPT

Maximum power point tracking (MPPT) algorithms help to extract maximum power from wind energy conversion systems (WECS). The speed and direction of the wind change continuously and thus output from a WECS fluctuates. As per the Betz limit only 59% of total available wind energy can be harnessed by the wind turbine. The WECS system operating region is from the cut in wind speed V_{cutin} to rated wind velocity V_{rated}. MPPT algorithms take into account variables like voltage, optimal power, and duty cycle to ensure maximum power generation for corresponding wind velocity in the operating region. The MPPT tracking algorithms for WECS can be broadly categorized into four types, direct power control (DPC), Indirect Power control (IPC), smart or AI-based, and hybrid algorithms that utilize both conventional and smart methods [2]. Fig. 1 summarizes the classification of different MPPT methods for wind energy.

978-1-6654-9921-7/22/$31.00 ©2022 IEEE

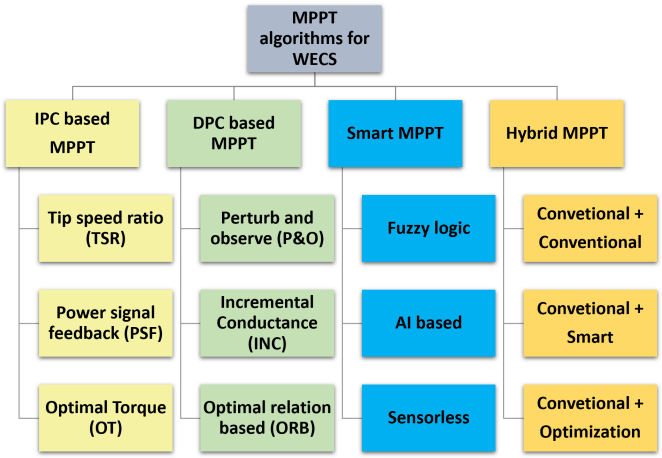

Fig. 1. Classification of MPPT algorithms for wind power.

C. Wind MPPT Existing methods

The most popular IPC-based conventional MPPT algorithm is tip speed ratio (TSR). In this method, the reference rotor speed is generated by estimating the rotor and wind speed. Using this reference speed and other system parameters, power extraction is optimized. The TSR algorithm can either utilize mechanical sensors employed in anemometers in the wind turbine swept area or estimate the wind speed through mathematical modeling [3], [4]. TSR is simple to implement and shows rapid response in regulating the rotor speed under changing environments. The drawbacks are increased installation and maintenance costs, lower efficiency, and lack of reliability. The optimum torque (OT) method uses an optimal torque curve for multiple wind speeds to regulate the generator torque. Although the technique is simple and yields higher efficiency under ideal conditions, it is greatly dependent on the climate and wind turbine characteristics [5]. Any mismatch between the assumed optimal torque curve based on the prior knowledge and the actual climate and wind turbine characteristics may cause significant performance drops. The power signal feedback (PSF) method uses a lookup table for optimal power for a wind turbine that is generated by an experimental setup or simulation. This

method shows good performance in tracking MPP at low wind speeds, but requires prior knowledge of system and wind speed sensors [6]. Similar to the other IPC methods, PSF is based on a static prior knowledge base, which may suffer from the potential mismatch between the assumed and actual characteristics of climate and turbine.

As opposed to the model-based IPC methods, DPC methods follow a data-driven approach, e.g., Perturb and observe (P&O) and Incremental conductance (INC) methods [1], [6]. In P&O, the control variables are adjusted and their effect on the performance is observed to decide on the next steps. The advantage of this algorithm is that it does not require additional measurement sensors and prior knowledge of the wind turbine parameters. However, choosing the appropriate direction and step size for perturbation is a challenging task, and thus the conventional P&O algorithm may be slow in convergence, and oscillate near MPP but fail to achieve it. The INC method observes the rectifier output power to decide the direction of perturbation. This data-driven method also suffers from slow convergence and oscillation around MPP, like P&O.

With advances in AI methods, several machine learning (ML) MPPT algorithms have been proposed recently. Researchers in [7], [8] presented radial basis function (RBF) neural network and Wilcoxon RBF neural (WRBFN) network for wind MPPT. Authors in [9] presented an artificial neural network (ANN) based MPPT algorithm by using the electric power and rotor speed of the generator as the input and the action values of the WECS as the output. While those ML methods improved the accuracy of previous model-based MPPT methods in terms of mapping the operating point to MPP based on the prior knowledge, they still cannot deal with the model mismatch in an effective way.

Several hybrid methods have also been proposed by researchers such as ANN and PSF [10]; ORB and particle swarm optimization [11]; TSR, PSF, and hill climb search (HCS) control [12]. Authors in [13] presented fuzzy logic and ANN based Adaptive Neural-Fuzzy Interface System (ANFIS) technique for MPPT. The hybrid methods aim to combine the advantages of several methods to improve performance, but are typically complex in terms of time and computational complexity.

D. RL for energy optimization & MPPT

Reinforcement learning (RL) is an AI technique which has been extensively used for data-driven optimization in various applications such as robotics, Internet of Things (IoT) [14], and power systems [15]. Recently, RL for MPPT in solar energy has also been studied [16]. There are also some RL works for wind MPPT. In [17], [18], authors presented similar Q-learning based MPPT techniques where the RL agent learns the MPP by interacting with the environment using the model-free Q-learning algorithm. One advantage of RL is that it does not require prior knowledge of the wind turbine characteristics or deployment of wind speed measurement sensors. The data-driven nature of RL enables adapting to the practical operating conditions, as opposed to completely relying on models built using prior knowledge. Compared to the traditional DPC methods, such as P&O, RL methods can converge to the MPP faster and more accurately since they take consider expected future impacts of actions (i.e., perturbations).

However, the existing Q-learning based wind MPPT methods [17], [18] lack two important aspects. The model-free Q-learning algorithm completely ignores easily accessible prior knowledge on the wind turbine, such as optimal power curves under different wind speeds that can be obtained from the manufacturer. A fully data-driven approach which does not use any prior knowledge may suffer slower convergence and loss of energy as a result, as we empirically demonstrate in Section IV. Moreover, due to the discrete nature of Q-learning, [17], [18] use look-up tables to store the expected values of state-action pairs (i.e., Q values) for decision making, which is limited to low-dimensional, low-resolution, discrete state-action spaces. Critical system state variables, such as rotor speed and output power, and action variables, such as change in rotor speed, are inherently continuous-valued. Hence, discretizing them into a low-resolution space inevitably causes performance loss. On the other hand, trying to store huge high-resolution look-up tables requires large memory spaces and intractable training time and data.

To address these shortcomings of Q-learning based wind MPPT methods, in this work we propose a deep RL solution which uses both prior knowledge for faster convergence and observed data for adapting to the operating environment. The main **contributions** of this work are:

- A novel deep RL method to control the turbine rotor speed under variable wind velocity;
- A machine learning-based optimal power curve predictor to utilize prior knowledge in the proposed deep RL method;
- Performance evaluation of the deep RL method with existing techniques.

The remainder of the paper is organized as follows: section II gives the necessary background. The proposed technique is explained in Section III, and the experimental results are presented in Section IV. Finally, concluding discussions and remarks are given in Section V.

II. Problem Statement

The electrical power generated by a wind turbine,

$$P = \eta_G \eta_C P_m, \tag{1}$$

depends on the generator efficiency η_G, converter efficiency η_C, and the mechanical power P_m captured by the turbine from wind. The mechanical power captured by a wind turbine is given by,

$$P_m = \frac{1}{2}\rho v^3 A C_p,$$

where $\rho,\ v^3,\ A = \pi R^2,\ R,\ C_p$ are respectively the air density, air velocity, area swept by the turbine blades, blade radius, and turbine power coefficient. C_p is a function of the turbine speed ratio $\lambda = \frac{\omega R}{v}$, where ω denotes the rotor speed,

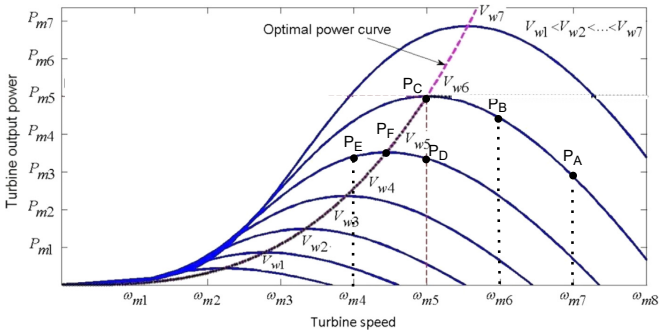

Fig. 2. Turbine power versus turbine speed for different wind speeds.

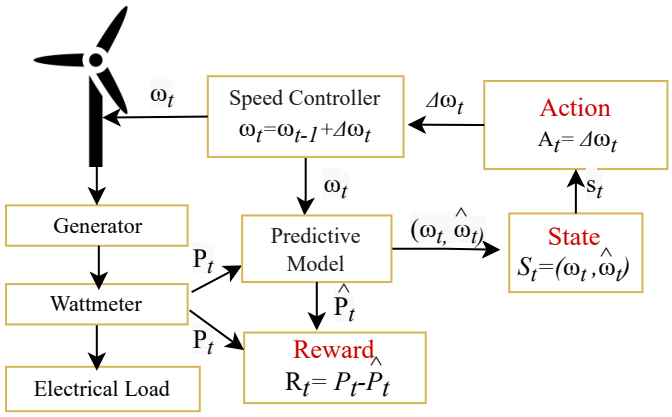

Fig. 3. Proposed wind MPPT model. RL components are marked in red.

and the blade pitch angle β. As the air density is mostly constant, for a particular turbine, Eq. (1) can be rewritten as

$$P = \underbrace{\frac{1}{2}\eta_G\eta_C\rho A}_{constant} \times \underbrace{v^3 C_p}_{f(v,\omega)}. \tag{2}$$

Eq. (2) indicates that the electrical power output for a given turbine varies according to the wind velocity v and the rotor speed ω, as shown in Fig. 2. The solid curved lines in Fig. 2 represent the output power for different turbine rotor speed and wind velocity. Higher wind speed results in more electrical power, as evident by the increasing power curves for higher wind velocity (i.e., $v_{w1} < v_{w2} < ... < v_{w7}$). The concave power curves show that for a fixed wind velocity there is a unique optimal rotor speed ω^* that yields maximum output power P_{max}. Connecting these points, we get the optimal power curve shown by the dashed magenta line.

As a test case, let us assume the turbine is operating at ω_{m7} rotor speed and produces power P_A. Under ideal conditions, with the knowledge of power curves for different wind speeds, the MPPT problem can be easily solved by changing the rotor speed ω_{m5} to obtain the maximum output power P_C. However, due to practical limitations, sophisticated MPPT solutions are needed in practice. Firstly, wind speed needs to be accurately measured using a sensor to exactly determine the optimal rotor speed ω_{m5}. Due to the critical dependence of the MPPT performance on the accuracy of wind sensors in this scenario and the cost of regular maintenance of wind sensors for accurate measurements, alternative approaches which do not require wind speed knowledge are studied. To this end, we train a predictive model to map the operating point P_A to the optimal rotor speed ω_{m5} without requiring the wind speed.

Moreover, without exactly knowing the optimal rotor speed, an MPPT method requires a step size $\Delta\omega = \omega_{m2} - \omega_{m1}$ to change the rotor speed while searching for the optimal rotor speed. While a big step size ensures fast tracking, it may also cause convergence issues. On the other hand, a small step size provides a smoother convergence to the MPP at the cost of slower tracking. For instance, in Fig. 2, starting with the operating point P_A, a search-based MPPT algorithm (e.g., Hill Climb Search (HCS) method [1]) with a step size of $\Delta\omega$ would first decrease the rotor speed from ω_{m7} to ω_{m6} and then to ω_{m5}. Consider the wind speed dropping to v_{m5} afterwards, hence causing the operating point to move from P_C to P_D at the same rotor speed ω_{m5}. In that case, the step size $\Delta\omega$ in Fig. 2 would cause the rotor speed to oscillate between ω_{m4} and ω_{m5}, whereas a smaller step size could prevent or decrease the oscillation.

Also with aging, the turbine capacity and performance decline, and the optimal power curve shifts downward. This non-stationary optimal power curve makes RL a suitable optimization technique for the wind MPPT problem.

III. Model Developement

Our method consists of a predictive model and an RL agent which controls the rotor speed of the turbine. The proposed control agent utilizes the advantage actor-critic (A2C) deep RL framework, which is suitable for fine-grained state-action spaces [19]. Fig. 3 shows that at time step t, the speed controller controls the turbine rotation speed ω_t. The turbine is connected to a generator that converts the mechanical power into electrical power, which serves the load. The generated power P_t is measured by a wattmeter. Then P_t and ω_t are fed into the predictor that maps the operating point to the optimal rotor speed $\hat{\omega}_t$, as explained in Section III-A. The current rotor speed and predicted optimal speed define the system state for the RL agent. The RL agent takes the action of rotor speed change $\Delta\omega_t$ to control the turbine speed. The generated power P_t and the predicted optimal power $\hat{P}_t$ define the reward to train the RL framework.

A. Predictive Model

Fig. 2 illustrates that the power curves for different wind speeds are non-overlapping. For a particular power curve, there is a rotor speed that achieves the maximum power and the resulting optimal operating points form the optimal power curve. That means a predictive model can be trained on data obtained from the manufacturer to output the optimal rotor speed that would maximize the generated power. The input data to the predictive model consists of current rotor speed ω_t and generated power P_t, and the output of the

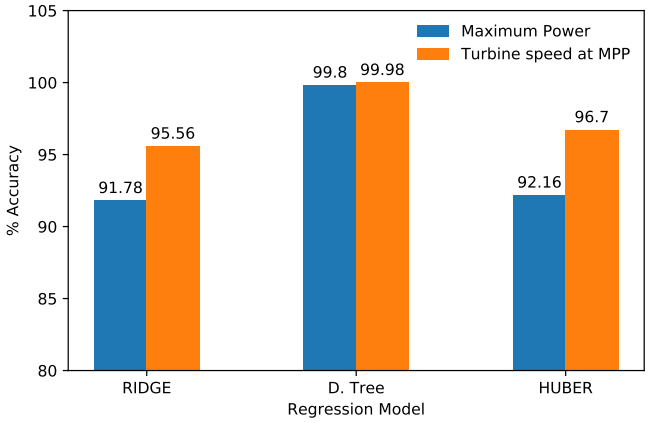

Fig. 4. Average accuracy for different regressor models for predicting maximum power and optimal rotor speed for different wind velocities.

predictive model is the predictions for optimal rotor speed $\hat{\omega}_t$ and maximum power $\hat{P}_t$.

We experimented with several regression models (Ridge, Decision Tree, and Huber) using the Python Scikit-learn package and selected Decision Tree regressor for the simulations. Although, all the regressors attain more than 90% accuracy, Decision Tree achieves the best accuracy for both maximum power and optimal turbine speed, as shown in Fig. 4.

B. MDP Model

In this paper, we design a Markov Decision Process (MDP) model to formulate the problem for the RL agent. The MDP model is based on the Markov property; i.e., given the current state and action the future state does not depend on the past states and actions. We next explain the elements of our MDP model for the MPPT controller, shown with red in Fig. 3.

1) State, S_t: The agent collects the rotor speed ω_t from the speed controller and predicted optimal speed $\hat{\omega}_t$ from the predictive model to form the MDP state as:

$$S_t = (\omega_t, \hat{\omega}_t).$$

Both of the state variables are positive real numbers, which can be effectively handled by deep RL methods like A2C without discretization, as opposed to the tabular RL methods such as Q-learning.

2) Action, A_t: The RL agent's action A_t is to select the change of turbine rotation speed $\Delta\omega_t$. So, $\Delta\omega_t = 0$ indicates no change in speed. Positive and negative $\Delta\omega_t$ values represent increase and decrease of turbine speed, respectively. Theoretically, continuous-valued speed changes can give the highest flexibility and performance optimization; however, the turbine speed controller can only accommodate a limited number of discrete values. We consider a fine-grained action space $A_t = \Delta\omega_t \in \{-0.05, -0.04, ..., 0.04, 0.05\}$, where the numbers indicate the percentile change of the nominal/-nameplate turbine speed (provided by the manufacturer). As a result, the RL agent has 11 possible actions.

Table I: GE Energy 1.5MW Wind Turbine Technical Specifications (GE 1.5 S).

Parameter	Definition	Value
P_{max}	Rated capacity	1500 kW
v_{rated}	Rated wind speed	12 m/s
v_{min}	Cut-in wind speed	4 m/s
	Number of rotor blades	3
$D = 2R$	Rotor diameter	70.5 m
A	Swept area	3.904 m^2
ω	Rotor speed (range)	11.1 – 22.2 rpm

The turbine speed change, $\Delta\omega_t$, is executed by the Turbine Speed Controller (TSC). This TSC includes necessary mechanical and electrical devices to achieve the target speed $\omega_t = \omega_{t-1} + \Delta\omega_t$.

3) Reward, R_t: In RL, the reward function guides the agent towards optimal action. The reward is observed from the environment but requires modeling to provide meaningful insight to the RL agent. We use the difference between the predicted and generated power as reward,

$$R_t = P_t - \hat{P}_t. \tag{3}$$

The agent aims to maximize the reward, i.e., maximize output power P_t. The selected reward in Eq. (3) provides the RL agent with a stable target to reach. The highest reward the agent can achieve is zero, i.e., generated power being equal to the predicted maximum power. Since changing turbine speed will incur a negative reward once the MPP is reached, the agent selects $A_t = 0$ ($\Delta\omega_t = 0$) in that case.

4) Next State, S_{t+1}: Given the agent's action and predicted optimal turbine speed, the next state is given by

$$S_{t+1} = (\omega_t + \Delta\omega_t, \hat{\omega}_{t+1}).$$

If the wind velocity remains the same, this transition is deterministic. Changing wind speed incurs randomness in the state due to the changing $\hat{\omega}_{t+1}$.

C. Solution Approach

The RL agent's objective is to maximize the expected discounted total reward for a time horizon T,

$$E\left[\sum_{t=0}^{T} \gamma^t R_t\right], \tag{4}$$

where $\gamma \in (0, 1)$ is the discount factor for future rewards. Two of the most popular approaches for finding the optimal policy $\{A_t\}$ are value-based methods (i.e., deep Q-learning) and policy-based methods (i.e., policy gradient). We consider the Advantage Actor-Critic (A2C) algorithm for this continuous-state MDP [19]. A2C is a hybrid deep RL method consisting of a policy-based actor network and value-based critic network. A pseudo code for the A2C algorithm is given in Algorithm 1.

IV. Experiments

A. Setup

The GE Energy 1.5MW Wind Turbine (GE 1.5 S) is a popular model, that we have used in our experiments. Its

Algorithm 1 A2C algorithm for wind MPPT.

Input: discount factor γ, learning rate, and number of episodes e
Input: Wind velocity $\{v_t\}$, and turbine speed $\{\omega_t\}$
Initialize: Actor network with random weights and critic network with random weights
for episode $= 1, 2, ..., e$ **do**
  for $t = 1, 2, ..., T$ **do**
    Observe ω_t and P_t
    Predict $\hat{\omega}_t$, and $\hat{P}_t$
    Select action A_t for state $S_t = (\omega_t, \hat{\omega}_t)$ (Actor network)
    Calculate reward $R_t = P_t - \hat{P}_t$
    Store transitions (S_t, A_t, R_t, S_{t+1}).
    Compute the advantage function (Critic network)
    Update actor network via advantage function.
    Update critic network through back propagation.
  end for
end for

Table II: Computational statistics for the experiments.

Hardware	Software	Task	Computation time
Intel® Core i7 3.60GHz 16 GB RAM	Python 3.8.5 Pytorch 1.8.1 sklearn 0.23.2	ANN Predictor Train	6 sec
		DRL Convergence	80 min
		DRL Decision	2.4 sec

technical specifications are given in Table I. The turbine's generation depends on the wind velocity and its rated wind speed is 12 m/s. However, for safety purposes the turbine is operational only between $4-25$ m/s. In our experiment, we generate a test signal where wind velocity ranges between $6-11$ m/s as shown by the dashed line and right y-axis in Fig. 5. We use air density 0.0013 kg/m^3 and other parameters from Table I.

In the A2C architecture, the actor and critic networks have similar structures with 2 input neurons, followed by 16, 64, and 16 neurons for both of them. However, the actor network has 11 output neurons (number of actions) compared to the 1 output neuron of the critic network. The Adam optimizer and a blend of linear and ReLU activation functions worked well for us. We set the learning rate to 0.001 and the discount factor to 0.95. We show the computation metrics in Table II for our proposed model. The A2C algorithm converges in 2000 episodes and takes 80 min computation time (each episode takes 2.4 sec). Each decision requires 0.024 sec, which provides a real-time applicability for our method.

B. Benchmark Policies

We compare our method with two policies.

1) Hill Climb Search (HCS) [1]: We use the popular Hill Climb Search method [1] as our baseline policy. We set the step size as 0.02 for hill climbing through a grid search.

2) Q-Learning Method [17]: Wei et al. [17] proposed an RL method for wind MPPT. They use the tabular Q-learning algorithm, and hence their state space is dicretized. The state space consists of the rotor speed and output power for the wind turbine, and the action space is the change of rotor speed (similar to our approach). However, they model the reward function with +1, 0, and -1 for positive, zero, and negative incremental change in power generation.

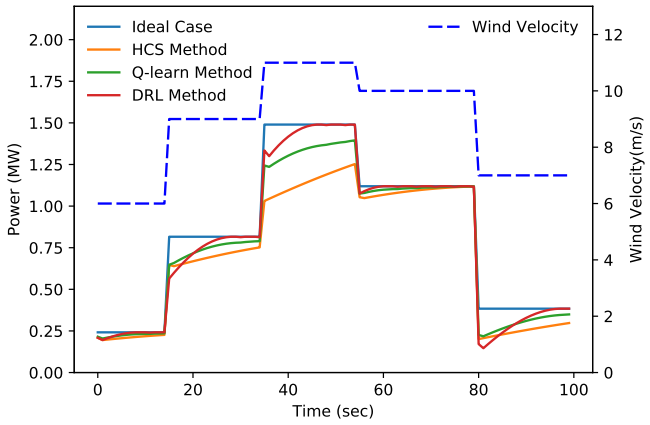

Fig. 5. Output power for different methods for varying wind speed.

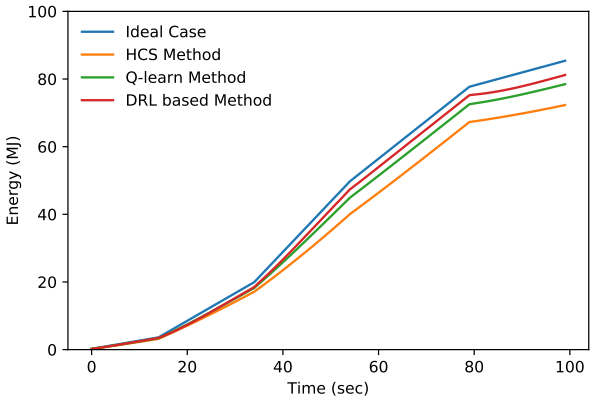

Fig. 6. Energy generation for different methods for the 100 sec experimental timeline.

C. Results

For the experiments, we have generated a 100 sec sequence. Ideal Case in Figs. 5 and 6 indicates the situation where the wind turbine always operates on the optimal power curve (MPP), hence providing the ideal benchmark to evaluate the other methods. Our deep RL method is the fastest to reach the MPP. The Q-learning method suffers from its discretized state space. However, it does better than the HCS method by considering expected future rewards in its actions. For the HCS method, deciding on the step size is critical. We followed the suitable step size from our grid search; however, a dynamic step size might perform better. The proposed deep RL method performs significantly better than the other methods, especially while adapting to a big change in the environment, as evident in the sudden drop of wind speed at 80^{th} sec (from 10 m/s to 7 m/s).

As a consequence of slow MPPT, the other methods yield a lower amount of wind energy, as seen in Fig. 6. The energy generation for different methods is consistent with the MPPT performance explained previously. Table III shows the yield

Table III: Comparison of energy generation for different methods.

After	Energy Generation (MJ)	Reduction
Ideal Case	85.39	0 %
HCS Method [1]	72.31	15.32 %
Q-Learning Method [17]	78.48	8.09 %
Proposed DRL Method	81.19	4.92 %

of energy in the $100\ sec$ timeline for different methods. The deep RL method lags the ideal case by 4.92 %; compared to 8.09 % and 15.32 % for the Q-learning and HCS methods.

V. Conclusion

This research provides a state-of-the-art deep RL solution (based on the actor-critic paradigm) to the MPPT task for wind energy. The proposed method utilizes a predictor for mapping the operating point to the maximum power point in the turbine speed vs. output power graph. Using the prior knowledge (power curves under different wind speeds) on the wind turbine, the predictor helps the deep RL agent to form an informative state space and a stable target to train its actor and critic networks. Integrating the prior knowledge into a state-of-the-art deep RL approach which can work with continuous-valued state variables and a fine-grained action space, the proposed method significantly outperforms the popular benchmarks, Hill Climb Search (HCS) [1] and Q-learning based MPPT method [17], in terms of generated power and total energy under varying wind speeds.

Acknowledgement

The authors would like to thank Huruy Gebremariam for his valuable insights.

References

[1] H. H. Mousa, A.-R. Youssef, and E. E. Mohamed, "State of the art perturb and observe mppt algorithms based wind energy conversion systems: A technology review," *International Journal of Electrical Power & Energy Systems*, vol. 126, p. 106598, 2021.

[2] J. Pande, P. Nasikkar, K. Kotecha, and V. Varadarajan, "A review of maximum power point tracking algorithms for wind energy conversion systems," *Journal of Marine Science and Engineering*, vol. 9, no. 11, p. 1187, 2021.

[3] C. M. Parker and M. C. Leftwich, "The effect of tip speed ratio on a vertical axis wind turbine at high reynolds numbers," *Experiments in Fluids*, vol. 57, no. 5, pp. 1–11, 2016.

[4] Y. Errami, M. Ouassaid, and M. Maaroufi, "Optimal power control strategy of maximizing wind energy tracking and different operating conditions for permanent magnet synchronous generator wind farm," *Energy Procedia*, vol. 74, pp. 477–490, 2015.

[5] M. Yin, W. Li, C. Y. Chung, L. Zhou, Z. Chen, and Y. Zou, "Optimal torque control based on effective tracking range for maximum power point tracking of wind turbines under varying wind conditions," *IET Renewable power generation*, vol. 11, no. 4, pp. 501–510, 2017.

[6] D. Kumar and K. Chatterjee, "A review of conventional and advanced mppt algorithms for wind energy systems," *Renewable and sustainable energy reviews*, vol. 55, pp. 957–970, 2016.

[7] C.-H. Chen, C.-M. Hong, and T.-C. Ou, "Wrbf network based control strategy for pmsg on smart grid," in *2011 16th International Conference on Intelligent System Applications to Power Systems*, 2011, pp. 1–6.

[8] T. Li and Z. Ji, "Intelligent inverse control to maximum power point tracking control strategy of wind energy conversion system," in *2011 Chinese Control and Decision Conference (CCDC)*, 2011, pp. 970–974.

[9] C. Wei, Z. Zhang, W. Qiao, and L. Qu, "Intelligent maximum power extraction control for wind energy conversion systems based on online q-learning with function approximation," in *2014 IEEE energy conversion congress and exposition (ECCE)*. IEEE, 2014, pp. 4911–4916.

[10] S. Azzouz, S. Messalti, and A. Harrag, "A novel hybrid mppt controller using (p&o)-neural networks for variable speed wind turbine based on dfig a novel hybrid mppt controller using (p&o)-neural networks for variable speed wind turbine based on dfig," 1874.

[11] M. A. Abdullah, T. Al-Hadhrami, C. W. Tan, and A. H. Yatim, "Towards green energy for smart cities: Particle swarm optimization based mppt approach," *IEEE Access*, vol. 6, pp. 58 427–58 438, 2018.

[12] J. Hussain and M. K. Mishra, "Adaptive maximum power point tracking control algorithm for wind energy conversion systems," *IEEE Transactions on Energy Conversion*, vol. 31, no. 2, pp. 697–705, 2016.

[13] M. R. Javed, A. Waleed, U. S. Virk, and S. Z. ul Hassan, "Comparison of the adaptive neural-fuzzy interface system (anfis) based solar maximum power point tracking (mppt) with other solar mppt methods," in *2020 IEEE 23rd international multitopic conference (INMIC)*. IEEE, 2020, pp. 1–5.

[14] A. Nassar and Y. Yilmaz, "Reinforcement learning for adaptive resource allocation in fog ran for iot with heterogeneous latency requirements," *IEEE Access*, vol. 7, pp. 128 014–128 025, 2019.

[15] S. S. Shuvo and Y. Yilmaz, "Home energy recommendation system (hers): A deep reinforcement learning method based on residents' feedback and activity," *IEEE Transactions on Smart Grid*, 2022.

[16] S. S. Shuvo, H. Gebremariam, and Y. Yilmaz, "Deep reinforcement learning based optimal perturbation for mppt in photovoltaics," in *2021 North American Power Symposium (NAPS)*. IEEE, 2021, pp. 1–6.

[17] C. Wei, Z. Zhang, W. Qiao, and L. Qu, "Reinforcement-learning-based intelligent maximum power point tracking control for wind energy conversion systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6360–6370, 2015.

[18] A. Kushwaha, M. Gopal, and B. Singh, "Q-learning based maximum power extraction for wind energy conversion system with variable wind speed," *IEEE Transactions on Energy Conversion*, vol. 35, no. 3, pp. 1160–1170, 2020.

[19] P.-H. Su, P. Budzianowski, S. Ultes, M. Gasic, and S. Young, "Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management," *arXiv preprint arXiv:1707.00130*, 2017.