

Article

A Novel Deep Learning Backstepping Controller-Based Digital Twins Technology for Pitch Angle Control of Variable Speed Wind Turbine

Ahmad Parvaresh ¹, Saber Abrazeh ², Saeid-Reza Mohseni ³, Meisam Jahanshahi Zeitouni ⁴, Meysam Gheisarnejad ⁵ and Mohammad-Hassan Khooban ^{6,*}

- ¹ Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman 76169-14111, Iran; parvaresh.ahmad@yahoo.com
- ² School of Electrical & Computer Engineering, Shiraz University, Shiraz 71946-84334, Iran; saber.abrazeh@yahoo.com
- ³ Electrical Engineering Department, Sharif University of Technology, Tehran 11365-11155, Iran; sr.mohseni@alum.sharif.edu
- ⁴ Electrical & Electronic Engineering Department, Shiraz University of Technology, Shiraz 71557-13876, Iran; jahanshahi.meisam@gmail.com
- ⁵ Department of Electrical Engineering, Islamic Azad University, Najafabad Branch, Najafabad, Esfahan 8514143131, Iran; me.gheisarnejad@gmail.com
- ⁶ DIGIT, Department of Engineering, Aarhus University, 8200 Aarhus N, Denmark
- * Correspondence: khooban@eng.au.dk

Received: 17 April 2020; Accepted: 5 June 2020; Published: 22 June 2020



Abstract: This paper proposes a deep deterministic policy gradient (DDPG) based nonlinear integral backstepping (NIB) in combination with model free control (MFC) for pitch angle control of variable speed wind turbine. In particular, the controller has been presented as a digital twin (DT) concept, which is an increasingly growing method in a variety of applications. In DDPG-NIB-MFC, the pitch angle is considered as the control input that depends on the optimal rotor speed, which is usually derived from effective wind speed. The system stability according to the Lyapunov theory can be achieved by the recursive nature of the backstepping theory and the integral action has been used to compensate for the steady-state error. Moreover, due to the nonlinear characteristics of wind turbines, the MFC aims to handle the un-modeled system dynamics and disturbances. The DDPG algorithm with actor-critic structure has been added in the proposed control structure to efficiently and adaptively tune the controller parameters embedded in the NIB controller. Under this effort, a digital twin of a presented controller is defined as a real-time and probabilistic model which is implemented on the digital signal processor (DSP) computing device. To ensure the performance of the proposed approach and output behavior of the system, software-in-loop (SIL) and hardware-in-loop (HIL) testing procedures have been considered. From the simulation and implementation outcomes, it can be concluded that the proposed backstepping controller based DDPG is more effective, robust, and adaptive than the backstepping and proportional-integral (PI) controllers optimized by particle swarm optimization (PSO) in the presence of uncertainties and disturbances.

Keywords: pitch angle control; DDPG algorithm; backstepping controller; digital twin (DT); DSP; software-in-loop; hardware-in-Loop

1. Introduction and Preliminaries

Nowadays, renewable energy sources have been playing a significant role in the achievement of reliable, efficient and affordable energy and they have good business development prospects.



In a comparison to these energy sources, wind energy is one of the fastest growing, economically cost-effective and most promising energy sources, and its development has progressed tremendously worldwide. Generally, the kinetic energy conversion of the wind into the electrical energy is done by wind turbine (WT). The operating region of every WT is mainly classified into two key areas: below and above-rated wind speed. The control objective at below-rated wind speed is to capture the maximum available power from the wind flow, using variable speed operation of WT. The pitch angle control is used to maintain the rated power at above-rated wind speed, while minimizing the load stress on the drive-train shaft at the same time [1]. Although the majority of WTs are fixed speed, numerous variable speed WTs are being increased because of this fact that they maximize the energy capture by functioning turbine at the maximum power coefficient.

A wide range of classical and modern control methods have been suggested to design pitch angle controllers at above-rated wind speed [2–4]. As highly sophisticated technologies, modern controllers can also increase the efficiency and performance of WTs, while keeping maintenance costs low [5,6]. In the last decades, the backstepping control strategy has been amply investigated and developed to access the stability goal of the whole system and state estimation obstacles. This control technique suggests good performance in both steady-state and transient operations, even in the presence of uncertainties, parameter variations, and load torque disturbances. The backstepping control laws are easily constructed and associated with Lyapunov functions [7,8]. Nonlinear integral backstepping (NIB), due to its recursive nature is the completely efficient controller, showed a great deal in stabilizing the nonlinear fixed-model WT systems with the presence of perturbations, and besides the integral action has been used to compensate the steady-state error [9]. On the other hand, 1111nonlinear characteristics of WTs lead to tough and almost impossible efforts to extract an exact model of a system. Furthermore, plant dynamics can be intensively changed with output disturbances, therefore we have no way to go through model-free controllers (MFCs). M. Fliess and C. Join in [10] have proposed an accurate definition of the MFC technique and its application in nonlinear systems to compensate modeling error.

Another key issue in NIBs is tuning its parameters to achieve the best outputs from the controller actions. Numerous studies have been done to find suitable optimization algorithms that are applicable to wind power generation systems [11,12]. Among other types of optimization and tuning methods, reinforcement learning (RL) has been increasingly developing [13]. There are lots of different online model-free value-function-based RL algorithms that use the deducted future reward criterion. Q learning [14], state–action–reward–state–action (SARSA) [15,16], and Actor-Critic (AC) methods [17] are well known, and there are also two more recent algorithms: QV learning [18] and AC learning automaton (ACLA) [18]. Furthermore, many policy search and policy gradient algorithms have been proposed [19,20], and there exist model-based [21] and batch RL algorithms [22]. Recently, the deep deterministic policy gradients (DDPG) algorithm has been widely using in a plethora of applications because of its strong learning ability and stability [23]. In this algorithm, there are two major neural networks (NNs): an actor NN (ANN) and a critic network (CNN). ANN is used to approximate the policy function and CNN is used to approximate the value function and besides, it works on approximation with deep neural networks for both the action-value function and the policy [24].

One of the newest concepts of information technology is known by digital twin (DT), which is increasingly applied in wind energy conversion systems. The term digital twin "means an integrated multi-physics, multi-scale, probabilistic simulation of a complex product, which functions to mirror the life of its corresponding twin" [25]. The combination of physical and virtual data has many advantages. On one hand, the physical product can be made more intelligent to actively adjust its real-time behavior according to the recommendations made by the virtual product. On the other hand, the virtual product can be made more factual to accurately reflect the real-world state of the physical product [26]. Nevertheless, we gathered evidence during our research that digital twin in a wind turbine is still in the early stages of development. A new concept of a digital twin has been considered in this paper. Firstly, two separate tests, software-in-loop (SIL) and hardware-in-loop (HIL), have been

considered to show the abilities of the controller in real-time applications. Secondly, with a unique combination of these tests, it has been shown a new concept of digital twin, which is clearly efficient and effective.

In this paper, a new DDPG-based NIB-MPC controller has been proposed, to achieve the aforementioned key points for the promotion of pitch angle control of a variable speed wind turbine in above-rated wind speed. The parameters of the proposed controller have been tuned adaptively by the DDPG algorithm with the actor-critic structure. In this controller, there is no need for a system dynamics model and the system uncertainties have been estimated by ultra-local model and compensated via feedback signal. In NIB-MPC structure, NIB gains have been chosen as control parameters and then they have tuned adaptively by the DDPG algorithm. To highlight the capabilities of the proposed approach and achievement of similarity between output behavior of the system in software-in-loop (SIL) and hardware-in-loop (HIL) testing, a digital twin (DT) of proposed controller has been presented. This DT is implemented with presenting a novel strategy, on a TI digital signal processor (DSP) computing device.

This paper is organized as follows. Section 2 presents the nonlinear model of the variable speed wind turbine. Then, Section 3 introduces the proposed controller with detail. Section 4 focuses on the digital twin concept for implementing of the proposed controller and its SIL and HIL testing. The results of simulation in the Matlab/Simulink platform and also the implementation of the controller on TI DSP hardware has been presented in Section 5. Finally, Section 6 summarizes the main contributions and describes some additional avenues for continuing research.

2. Variable Speed Wind Turbine Nonlinear Model

As well known, wind energy is electricity produced by using mechanical components and electrical generators. A two-mass model is commonly used in the literature [27] to describe the variable speed wind turbine nonlinear dynamics. The use of a two-mass model is motivated due to this fact that the control laws derived from this model are more general and can be applied for wind turbines of different sizes. Particularly, these controllers are more adapted for high-flexibility wind turbines that cannot be properly modelled with a one mass model. In fact, it is also shown in [28] that the two-mass model can report flexible modes in the drive train model that cannot be highlighted with the one mass model. Full structure of a typical horizontal-axis wind turbine has been shown in Figure 1.



Figure 1. Structure of a typical wind turbine [28].

Lift and exerting a turbine force are generating. In nacelle, the rotating blades turn a shaft that goes into a gearbox. Wind power extract from the wind by the rotor which is limited by the Betz limit (maximum 59%). Therefore, the mechanical power is expressed in Equation (1) [3,27].

$$P_a = \frac{1}{2} \cdot \rho \cdot C_p (\lambda \cdot \beta) \cdot A \cdot V(t)^3 \tag{1}$$

In this case, ρ is the air density (kg/m³), C_P is the power coefficient, A is the swept area of the turbine (m²) and V is the wind speed (m/s). C_p denotes the power coefficient of wind turbines, which is a nonlinear function of pitch angle β and tip-speed ratio λ . λ is calculated by the blade tip speed and wind speed upstream of the rotor as [29]:

$$\lambda = \frac{R\omega_r}{V} \tag{2}$$

With ω_r being the rotor angular speed. Furthermore, the power coefficient can be obtained by:

$$C_p(\lambda .\beta) = 0.5176 \left(\frac{116}{\lambda_i} - 0.4\beta - 5\right) e^{-21/\lambda_i} + 0.0068\lambda$$
(3)

The parameter λ_i can be calculated as follow:

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1}$$
(4)

Nonlinear wind turbine model is shown in a generalized nonlinear form as follows:

$$\dot{X} = G(X) + Bu = \begin{bmatrix} \frac{P_r(x_1 \cdot x_4 \cdot V)}{x_1 J_r} - \frac{x_1 D_s}{J_r} + \frac{x_2 D_s}{N_g J_g} - \frac{x_3 K_s}{J_r} \\ \frac{x_1 D_s}{N_g J_g} - \frac{x_2 D_s}{N_g^2 J_g} + \frac{x_3 K_s}{N_g J_g} - \frac{T_g}{J_g} \\ x_1 - \frac{x_2}{N_g} \\ -\frac{1}{\tau_\beta} x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau_\beta} \end{bmatrix} u$$
(5)

In Equation (5) model nonlinear vector is G(X), X is state vector (Equation (6)), u is control input (Equation (6)). The system output (Y) is as Equation (7).

$$X = \begin{bmatrix} \omega_r & \omega_g & \delta & \beta \end{bmatrix}^T \cdot u = \beta_r$$
(6)

$$Y = \omega_r \tag{7}$$

With δ is twist angle, ω_g is generator speed and ω_r is rotor speed. In Equation (5), τ_β is time constants of pitch actuator and β_r is the pitch angle control. T_g is generator torque, J_r and J_g are the rotor and generator inertia, N_g is gear ratio, D_s and K_s are drive-train damping and spring constant, respectively.

3. Design of Proposed Controller

3.1. Nonlinear Integral Backstepping Model-Free Control (NIB-MFC)

In this section, the method of nonlinear backstepping model-free control (NIB-MFC) and system stability will be proposed. The wind turbine dynamics can be illustrated by the following nonlinear system [30]:

$$x^{(n)} = f(x) + bu \tag{8}$$

where u and f(x) are the system input and model system dynamics respectively. Equation (8) can be written as:

$$x^{(n)} = f(x) + f_e(\cdot) + \beta u \tag{9}$$

where β is the estimate of the unknown gain of parameter *b* and $f_e(\cdot)$ is the un-modeled and uncertainties dynamics of WT, therefore $f_e(\cdot)$ can be formulated as:

$$f_e(\cdot) = Model Uncertainties + (b - \beta)u$$
(10)

To reduce the error of certain state variables, the ultra-local model can be used for the known and modeled nonlinear dynamics of WT.

$$u = \frac{F - x_d^{(n)} - u_c}{\beta} \tag{12}$$

$$x^{(n)} = f(x) + x_d^{(n)} + u_c$$
(13)

The state variable of the wind turbine can be formulated as follows:

$$x_1 = \omega_r \tag{14}$$

$$x_2 = \dot{x}_1 \tag{15}$$

But in practice, the actual and desired values of state variable (x_1) is not the same so the error between them is represented by:

$$e_1 = x_1 - x_1^d = x_1 - x_d \tag{16}$$

The position tracking and velocity tracking error can be convergence to a certain variable by using the theory of NIB-MFC. The block diagram of this control loop has been illustrated in Figure 2 [30,31].



Figure 2. Illustration of nonlinear integral backstepping model-free control (NIB-MFC) loop control.

The Lyapunov function is chosen to guarantee the convergence stability of the nonlinear WT system for this purpose. The Lyapunov function $V(e_1)$ will be defined to be positive definite around the state variable and can be written as:

$$V_1(e_1) = \frac{1}{2}e_1^2 \tag{17}$$

The derivative of this function is shown as follows:

$$\dot{V}_1(e_1) = e_1 \dot{e}_1 = e_1 \left(x_2 - \dot{x}_d \right)$$
 (18)

Since, x_2 is not our control input, there will be a dynamic error between it and its desired value, x_2^d . Therefore, the velocity tracking error can be offered to compensate for the dynamics error:

$$e_2 = x_2 - x_2^d$$
 (19)

The error will go to zero if the Lyapunov function is chosen semi-negative. The implicit input x_2 can be written as:

$$x_2^d = \dot{x}_d - k_1 e_1 \tag{20}$$

$$e_2 = x_2 - \dot{x}_d + k_1 e_1 \tag{21}$$

The modeling error and uncertainties lead to steady-state error. This error can be eliminated by using the integral term to the system as shown in below:

$$x_2 = \dot{x}_d - k_1 e_1 - k_3 \int e_1 \tag{22}$$

$$e_2 = x_2 - \dot{x}_d + k_1 e_1 + k_3 \int e_1 \tag{23}$$

As a result, the derivative of velocity and position tracking can be described as:

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_d = e_2 + x^* - \dot{x}_d = e_2 - k_1 e_1 - k_3 \int e_1$$
(24)

$$\dot{e}_2 = \dot{x}_2 - \ddot{x}_d + k_1 \dot{e}_1 - k_3 e_1 = \dot{x}_2 - \ddot{x}_d + k_1 e_2 - k_1^2 e_1 - k_1 k_3 \int e_1 - k_3 e_1$$
(25)

The Lyapunov function $V_1(e_1, \int e_1)$ and $V_2(e_1, e_2, \int e_1)$ will be defined for the position and velocity tracking error and formulated as [31]:

$$V_1\left(e_1, \int e_1\right) = \frac{1}{2}e_1^2 + \frac{k_3}{2}\left(\int e_1\right)^2 \tag{26}$$

$$\dot{V}_1\left(e_1, \int e_1\right) = e_1\dot{e}_1 + k_3e_1\int e_1 = e_1e_2 - k_1e_1^2$$
(27)

$$V_2\left(e_1.e_2.\ \int e_1\right) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{k_3}{2}\left(\int e_1\right)^2$$
(28)

$$\dot{V}_2(e_1.e_2.\int e_1) = e_1\dot{e}_1 + e_2\dot{e}_2 + k_3e_1\int e_1 = e_1e_2 - k_1e_1^2 + e_2(\dot{x}_2 - \ddot{x}_d + k_1e_2 - k_1^2e_1 - k_1k_3\int e_1 - k_3e_1)$$
(29)

To guarantee the convergence of the e_2 to zero the $\dot{V}_2(e_1.e_2, \int e_1)$ should be semi-negative definite. This can be satisfied by choosing the Equation (30).

$$e_1e_2 + e_2\left(\dot{x}_2 - \ddot{x}_d + k_1e_2 - k_1^2e_1 - k_1k_3\int e_1 - k_3e_1\right) = -k_2e_2^2$$
(30)

Consequently, \dot{x}_2 can be written as:

$$\dot{x}_2 = f(x) + \ddot{x}_d + u_c = \ddot{x}_d + (k_1^2 + k_3 - 1)e_1 - (k_1 + k_2)e_2 + k_1k_3 \int e_1$$
(31)

$$u_{c} = \left[\left(k_{1}^{2} + k_{3} - 1 \right) e_{1} - \left(k_{1} + k_{2} \right) e_{2} + k_{1} k_{3} \int e_{1} \right] - f(x)$$
(32)

3.2. Reinforcement Learning

Reinforcement learning (RL), due to its generality, is studied in many areas such as control theory, operations research, simulation-based optimization, multi-agent systems, statistics, and genetic algorithms [32]. The problems of interest in RL have also been studied in the theory of optimal control, which is concerned mostly with the existence and characterization of optimal solutions, particularly in the absence of a mathematical model of the environment. Therefore, in wind turbine plant, where the system is nonlinear and has huge complexities, RL is a powerful and practical tool to estimate controller parameters which control wind turbine blade pitch angle and consequently rotor speed under the various level of wind speed variations.

RL is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. But to implement RL operationally which are mainly continuous control problems, there are many problems including the divergence of learning, continuous nature of inputs and outputs and temporal correlation of data.

Recently, deep Q-network (DQN) has introduced a new set of features to solve most of the problems mentioned. However, a number of these challenges such as continuous states which are especially related to practical applications, cannot be resolved by this algorithm. In this regard, deep deterministic policy gradients (DDPG) had been proposed by Lillicrap et al. [33] based on the significant progress in DQN and the new approach named actor-critic paradigm expressed in [34] as a method which tackles continuous control issues.

3.3. The Learning Process

Firstly, the following concepts which are related to RL are explained below:

- Markov decision process (MDP): It is the form in which the RL environment is typically stated, and it is because many RL algorithms for this context utilize dynamic programming techniques.
- Agent: The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human, by maximizing its reward and minimizing its penalty.
- Environment: The environment is the physical world in which the agent operates. The agent's current state and action are considered as its input, and the agent's reward and its next state are its output.
- State: State *s* is the current situation of the agent in the environment, and S is the set of all possible states of the agent.
- Policy: Policy π is the method by which the agent's state is mapped to an appropriate action leading to the highest reward.
- Action: A is the set of all possible moves *a*, that the agent can make.
- Reward: This value is the feedback from the environment as an evaluation criterion that determines the success or failure of an agent's actions in a given state.
- Value function: The value function V^π is defined as the long term expected to return with a discount. The discount factor (γ ε (0, 1]) dampens the rewards' effect on the agent's choice of action to make future rewards worth less than immediate rewards. Roughly speaking, the value function estimates "how good" it is to be in a given state.
- *Q*-value: *Q*-value or action value is used to measure how effective taking an action at a state is.

From the practical point of view, the interaction between an active decision-making agent and its environment happens in all RL applications. On the other words, the agent tries to achieve a goal, through maximizing reward, despite uncertainty about its environment.

The standard reinforcement learning theory expresses that an agent obtains a policy, which maps every state $s \in S$ to an action $a \in A$. It maximizes the expectation of a long-term discounted reward as below:

$$J = \mathbb{E}_{r_i.s_i \sim E.a_i \sim \pi}[R_1] \tag{33}$$

where $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the total long term discounted reward at each step.

A value function V^{π} which is formulated in Equation (34), depicts the total discounted reward R_t for each *s* ϵ \mathbb{S} .

$$V^{\pi}(s) = \mathbb{E}_{\pi}[R_t|s_t = s] \tag{34}$$

The value function, V^{π} can be recursively described as Equation (35) according to the Bellman Equation:

$$V^{\pi}(s) = \mathbb{E}_{\pi}[r_t + \gamma V^{\pi}(s_{t+1}) | s_t = s]$$
(35)

An equivalent of the value function is represented by the action-value function Q^{π} in Equation (36), given as [35]:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[r_t + \gamma Q^{\pi}(s_{t+1}.a_{t+1})|s_t = s.a_t = a]$$
(36)

The policy shall be chosen in such a way that it maximizes the action-value function. On the other hand, $\pi^* = \operatorname{argmax} Q^*(s.a)$.

The DDPG algorithm having a great ability to solve continuous problems consists of two neural networks (NNs) $\mu(s_t|\theta^{\mu})$ and $Q(s_t.a_t|\theta^{Q})$ named actor NN (ANN) and critic network (CNN), where θ^{Q} and θ^{μ} are the weights of the CNN and ANN, respectively. According to the stochastic gradient descent, the CNN is updated by minimizing the loss function below [36,37]:

$$\mathcal{L}(\theta^{Q}) = E_{(s,a)} \Big[\Big(y_t - Q \Big(s_t, a_t \big| \theta^{Q} \Big) \Big)^2 \Big]$$
(37)

where

$$y_t = r_t(s_t, a_t) + \gamma \mathcal{Q}\left(s_{t+1}, \mu(s_t | \theta^{\mu}) | \theta^{\mathcal{Q}}\right)$$
(38)

Based on the policy of Equation (39), the coefficients of ANN are updated.

$$\nabla_{\theta^{\mu}} J^{\theta^{\mu}} \approx \mathbb{E}_{s_{t} \sim \rho^{\beta}} \Big[\nabla_{\theta^{\mu}} Q \Big(s.a \big| \theta^{Q} \Big) \Big|_{a = \mu(s|\theta^{\mu})} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu}) \Big]$$

$$= \mathbb{E}_{s_{t} \sim \rho^{\beta}} \Big[\nabla_{a} Q \big(s.a \big| \theta^{Q} \big) \Big|_{a = \mu^{\theta}(s)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu}) \Big]$$
(39)

In the above equation, β is a specific policy to the current policy π and ρ is the discounted distribution.

Due to correlations existing in the input experiences, a replay buffer \mathcal{D} is used to weaken that in the DDGP algorithm. To enable a robust learning DDPG agent, two separate NNs $\mu'(s|\theta^{\mu'})$ and $Q'(s.a|\theta^{Q'})$ named target NNs (TNNs) are utilized in addition to the main ANN and CNN. The additional NNS are same in shape to the main NNs but have distinct coefficient weights θ' [38–40].

4. Digital Twin Controller of WT System

4.1. The Concept of Digital Twin

Nowadays, modeling and simulation is a standard process in system development. The digital twin (DT) concept refers to the accurate reproduction of a physical wind turbine in a computational system to facilitate understanding and study its behavior [41,42]. This digital twin can empower wind asset owners and turbine manufacturers operating wind turbines to predict and plan for faults and optimize the performance of their assets. The technology involves creating a digital copy or "twin" of physical assets, processes, systems, and devices to allow real-time remote monitoring that can save the wind industry significant downtime and maintenance costs while increasing production. Real-time

data from sensors is fed into the digital twin and compared to simulated theoretical parameters under the same working conditions. Then similarities and discrepancies are analyzed to diagnose the health of the asset.

In this paper, to achieve the mentioned purposes, it has been illustrated how a digital twin of the pitch angle controller implements on a TMS320F28379D Dual-Core Delfino[™] Microcontroller device in Texas Instrument (TI). The hardware-in-loop (HIL) idea develops a controller algorithm to diagnose the health of its behavior on the wind turbine. However, the software-in-the-loop (SIL) idea permits the test of the algorithms but neglects the test of the controller hardware. In this paper, it provides some strategies for HIL and SIL model with DT concept that it is outlined in the list below:

- Define the system and simulation of closed-loop control in software;
- Implementation of the proposed controller on a TI microcontroller board;
- Upgrade the controller coefficients and achieving the desired output using the DDPG-NIB method in HIL mode with real-time data;
- Optimization of control coefficients of SIL controller reusing NIB-DDPG method (criteria: similarity of SIL and HIL outputs).

It would appear logical to conclude that these strategies refer to the outputs and performances of the two systems are similar. It is possible to estimate the behavior of the system in HIL mode by changing its parameters in SIL mode. At the first step, the NIB-MFC of the HIL setup is regulated to reduce the rotor speed deviations in the WT system. Following this, the NIB-MFC of SIL is designed in such a way that it minimizes the difference between the outcome of the WT in the HIL and SIL environments. Thus, the design of the DT controller for the WT plant is carried out in two distinct steps which are illustrated in Figure 3.

Figure 3. A proposed strategy for the combination of hardware-in-loop (HIL) and software-in-loop

4.2. The Proposed DDPG Tuned Backstepping Control Method

(SIL) testing.

The parameter estimation accuracy of the backstepping controller highly affects the quality of its output actions. Therefore, the proposed DDPG algorithm is used to design the coefficients embedded in the NBI controller structure to offers a new as an adaptive tuner mechanism (instead of tuning manually). In the backstepping controller, the NIB block has a nonlinear attitude, especially in temporary variable variations which leads to controller performance deterioration. Thus, the best solution to tackle this issue is to adaptively calculate the NIB gains (k_1 , k_2 and k_3) based on the DDPG algorithm. The structure of the proposed DDPG backstepping method leading to have a constant output rotor angular velocity is represented in Figure 4. In this structure, the DDPG algorithm provides tuner signals to adjust the NIB-MFC gains adaptively.



Figure 4. Illustration of an actor-critic network.

The critic network in the proposed structure is responsible for the effectiveness evaluation of the actor policy, and according to the critic network data, the ANN adjusts the NIB gains to reach the controller objective. The ANN senses the state variables and then generates three continuous control signals for tuning of the NIB gains (k_1 , k_2 and k_3). After that, the CNN receives the state variables and turning signals, and then reward signal r_t is calculated. Following that, the critic network weights are trained to lead to an updated DDPG network with adaptive tuning action signals to feed the controller.

The terms of rotor speed, rotor speed error, and rotor speed error integral as are chosen here in both the HIL and SIL to form a three-dimensional vector of state space, represented as:

$$s_t = \left\{ \omega_r, e, \int e dt \right\} \tag{40}$$

where s_t is the state of the MDP in the HIL and SIL environments.

In this application, the structure of the NNs of the DDPG algorithm for the design of the HIL and SIL controllers is the same with two hidden layers of 200 and 100 units. The architecture of the ANN and CNN for online tuning of the HIL and SIL controllers are illustrated in Figure 5, where the rectified linear unit (ReLU) is chosen as the activation function. As depicted in Figure 5, the inputs of the ANN are the system states while the ANN output and system states are inserted into the CNN.



Figure 5. Structure of the actor neural network (ANN) and critic neural network (CNN).

To determine the optimal control coefficients based on the DT concept, the reward function of the DDPG algorithm for the design of HIL and SIL controllers is defined as Equations (41) and (42), respectively.

reward in HIL =
$$\frac{1}{(rotor speed error)^2}$$
 (41)

reward in SIL =
$$\frac{1}{\left| difference \ between \ the \ rotor \ speed \ of \ HIL \ and \ SIL \right|}$$
(42)

4.3. Implementing the Adaptive NIB Controller Based DDPG

The training procedure of the DDPG mechanism for online tuning of the NBI controller coefficients in the HIL and SIL environments are the same which is described in the following manner.

The ANN and CNN $\mu(s_t|\theta^{\mu})$ and $Q(s_t, a_t|\theta^Q)$, with coefficient weights θ^{μ} and θ^Q , respectively, are initialized randomly. The TNNs $\theta^{Q'}$ and $\theta^{\mu'}$ are updated with weights of $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^{\mu'} \leftarrow \theta^{\mu}$, respectively. A replay buffer with the capacity \mathcal{D} is constructed. The initial state is s_1 stored. The action $a_t = [k_1, k_2, k_3] = \mu(s_t|\theta^{\mu}) + \text{nosie}$ is chosen based on ANN. The action a_t is applied to the system (HIL or SIL controllers) to obtain the next state s_{t+1} and reward r_t —the reward is calculated by Equations (41) and (42) for HIL and SIL, respectively. The term (s_t, a_t, r_t, s_{t+1}) , which is the experience set at each time step, is saved in the *R*-sized experience memory. During each step of the training process, a mini-batch of experiences saved previously are uniformly sampled from the memory \mathcal{D} to update the NNs at each time step. $y_t = r_t(s_t, a_t) + \gamma Q'(s_t, \mu(s_t|\theta^{\mu'})|\theta^{Q'})$ is calculated, the CNN is updated by minimizing the loss $\mathcal{L}(\theta^Q) = E_{(s,a)} \left[(y_t - Q(s_t, a_t|\theta^Q))^2 \right]$. The policy of ANN is updated by using the following policy gradient: $\nabla_{\theta^{\mu}} J^{\theta^{\mu}} = \mathbb{E}_{s_t \sim \rho^{\beta}} \left[\nabla_a Q(s_t, a_t|\theta^Q) \Big|_{a=\mu^{\theta}(s)} \nabla_{\theta^{\mu}} \mu(s_t|\theta^{\mu}) \right]$. The TNN's are updated by the following learning mechanism:

$$\theta^{Q'} \leftarrow \tau \theta^{Q} + (1 - \tau) \theta^{Q'} \text{ and } \theta^{\mu'} \leftarrow \tau \theta^{\mu} + (1 - \tau) \theta^{\mu'}$$
 (43)

where $\tau \ll 1$.

5. Results

The NIB-MFC scheme, which is a model-free scheme with an ultra-local model, offers optimal performance to compensate system output of the WT plant in the digital twin framework. For this purpose, the NBI-MFC controller has been adopted in the HIL and SIL environments. The gains of the NIB-MFC technique, which play a critical role in the pitch angle control of a WT plant, are considered as the control coefficients which should be adjusted by the DDGP tuner mechanism. The DDPG method throughout 200 episodes, which is equal to 2500 training steps.

It can be said that the target of using digital twin is the similarity system's output behaviors in the SIL and HIL. If this purpose is satisfied, then it is possible to estimate the behavior of the HIL system by changing the parameters in the SIL system. To achieve similarity behaviors of system output in the digital twin system, firstly deep deterministic policy gradient (DDPG) based nonlinear integral backstepping (NIB) in combination with model free control (MFC), (DDPG-NIB-MFC), is used to obtain optimal controller parameters by reducing differences between reference input and output in HIL and after that, HIL pitch angle output is applied as reference input in SIL. In the subsequent section, the performance of the suggested control system is performed by real-time software-in-the-loop (RT-SIL) MATLAB simulation experiments, as well as real-time hardware-in-the-loop (RT-HIL) TI board. Moreover, the backstepping and proportional-integral (PI) controllers are also designed by the particle swarm optimization (PSO) algorithm in the digital twin framework for the comparative purposes. By minimizing the objective function, the controller coefficients are optimally designed. In this application, the inverse values of the reward functions for the HIL and SIL controllers (defined in Equations (41) and (42)) were defined as objective functions.

5.1. Scenario I

At the first stage, a multi-step wind speed variation in the range of [14 m/s, 21 m/s] is applied to the non-linear WT system. The profile of the wind speed disturbance is depicted in Figure 6 while the rotor speed curves of the HIL system for the backstepping based DDPG, PSO optimized backstepping and PI controllers are shown in Figure 7. From the comparative outcomes of Figure 7, the suggested backstepping controller based DDPG offers a superior dynamic performance having a lesser settling time and smaller amplitude of fluctuations than the backstepping controller based PSO in the terms. It is also seen that the outcome of the PI controller based PSO experiences large-angle rotor speed fluctuation and thus it cannot compensate for the multi-step wind speed variation. The curve of the average reward for the full-simulated training phase under the wind disturbance is depicted in Figure 8. Looking at the details, as it regards Figure 8, the reward started at 200,000, then the value goes up significantly since episode 5, at which point it almost constant. The increasing trend of the reward measured in HIL is an indicator of the reduction of rotor speed error which confirms the correctness of the suggested NBI controller designed by the DDPG algorithm.







Figure 7. HIL output comparative results of backstepping controller based DDPG in combination with model free control (MFC), backstepping and PI controllers optimized by PSO according to Scenario I.



Figure 8. The trend of average reward in the HIL environment.

Similarly, the rotor speed outcomes of the SIL for the backstepping based DDPG, PSO optimized backstepping and PI controllers are compared as illustrated in Figure 9. Critical observation of the SIL outcomes reveals that the suggested controller gives a higher quality transient and steady-state behavior of rotor speed compared to the PSO optimized backstepping and PI controllers. Figure 10 depicts that the DDPG agent is trained over 200 episodes to adaptively tune the backstepping controller coefficients. From Figure 10, it is clear that the average reward is increased and stabled during the 200 episodes which means the difference between the system outcomes in HIL and SIL is minimized. This affirms the efficiency of the DDPG agent in tuning the backstepping controller in the digital twin concept.



Figure 9. SIL output comparative results of backstepping controller based DDPG, backstepping controller based PSO and PI controller based PSO according to Scenario I.



Figure 10. The trend of average reward in the SIL environment.

The dynamic specifications of the WT system with the multi-step wind speed in the terms of settling time, overshoot and error output are noted and furnished in Table 1. For comparison, the obtained outcomes reached for both HIL and SIL environments are provided in Table 1. From the statistical analysis, by employing the backstepping controller based DDPG, an improvement in the dynamic specifications of digital twin-based system is achieved for both the HIL and SIL.

Performance Measurements	DDPG Based NIB-MFC		PSO Based NIB-MFC		PSO Based PI	
	HIL	SIL	HIL	SIL	HIL	SIL
Settling time	0.02	0.03	0.06	0.15	21.6	23.2
Overshoot	0.196%	0.24%	0.28%	0.36%	30.7%	35.1%
Error	0.076	0.16	0.178	0.23	11.21	13.92

Table 1. Settling time, overshoot and error comparison outcomes according to the Scenario I.

5.2. Scenario II

In this case, the applicability of the suggested digital twin controller is explored when the wind speed is randomly fluctuated within [14 m/s, 22 m/s]. The profile of the random wind speed (which is numerically produced by an additive Gaussian noise with noise power 0.0003 to DC and slope levels at different time intervals) is presented in Figure 11 and the comparative dynamic outcomes for the HIL and SIL controllers are illustrated in Figures 12 and 13, respectively. The outcomes of these figures prove the superiority of the backstepping controller based DDPG to damp the rotor speed in the HIL environment. In addition, it is demonstrated that the curves of rotor speed are very close to each other in both HIL and SIL dynamic outcomes.



Figure 11. Random distribution of wind speed.



Figure 12. HIL output comparative results of backstepping controller based DDPG, backstepping controller based PSO and PI controller based PSO according to Scenario II.



Figure 13. SIL output comparative results of backstepping controller based DDPG, backstepping controller based PSO and PI controller based PSO according to Scenario II.

5.3. Scenario III (The Parametric Uncertainty in the Turbine Model)

For an illustration of the robustness ability of suggested backstepping controller based DDPG, some uncertainties are imposed on the WT model parameters under the following: $R_B = +30\%$, $J_R = +40\%$ and $T_B = +50\%$. Two standard error measurement criteria including the mean square error (MSE) and the root mean square error (RMSE) are considered, which are defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left[e_i^2 \right]$$
 (44)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left[e_i^2\right]}$$
(45)

The MSE and RMSE values for the HIL and SIL are provided in Figure 14a,b, respectively. From the bar comparison graphs, it is proved that the backstepping controller based DDPG has less sensitivity than other controllers to increasing of rotor radius, rotor inertia and pitch actuator time constant. It is also confirmed that by employing the suggested controller, the behavior of SIL output is the same as HIL output variations and this means that the concept of the digital twin is fulfilled.



(a) Mean square error (MSE) standards for the parametric uncertainties



(b) Root means square error (RMSE) standards for the parametric uncertainties

Figure 14. Comparison of mean square error (MSE) and root mean square error (RMSE) standards for the parametric uncertainties.

6. Conclusions

This paper focuses on presenting a novel backstepping controller based DDPG for a pitch angle control of variable speed WT in the digital twin framework. Initially, the backstepping controller based DDPG is adopted for control of WT in HIL to damp the rotor speed fluctuations in this environment. Following this, the digital twin of the WT system is constructed in SIL and DDPG algorithm is employed to tune the NIB controller coefficients by the measured data from the WT in the HIL environment. The digital twin realization of the suggested scheme has been implemented on a TMS320F28379D dual-core Delfino[™] microcontroller device in Texas Instrument (TI). The results revealed that the dynamic responses of the WT speed rotor are improved with the backstepping controller based DDPG. Moreover, the suggested control scheme can tune the SIL controller coefficients and make the digital twin WT system has the same operation with the HIL. From the analysis, it is found that the presented controller is more efficient and reliable than the PSO optimized backstepping and classical PI controllers.

Author Contributions: Data curation, S.A. and M.J.Z.; Methodology, A.P.; Software, S.-R.M.; Supervision, M.-H.K.; Validation, M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Rezaei, V. Advanced control of wind turbines: Brief survey, categorization, and challenges. In Proceedings of the 2015 American Control Conference (ACC), IEEE, Chicago, IL, USA, 1–3 July 2015; pp. 3044–3051.
- 2. Chen, J.; Chen, J.; Gong, C. New Overall Power Control Strategy for Variable-Speed Fixed-Pitch Wind Turbines within the Whole Wind Velocity Range. *IEEE Trans. Ind. Electron.* **2012**, *60*, 2652–2660. [CrossRef]
- 3. Boukhezzar, B.; Siguerdidjane, H. Nonlinear Control of a Variable-Speed Wind Turbine Using a Two-Mass Model. *IEEE Trans. Energy Convers.* **2010**, *26*, 149–162. [CrossRef]
- 4. Leithead, W.E.; De La Salle, S.A.; Reardon, D. Classical control of active pitch regulation of constant speed horizontal axis wind turbines. *Int. J. Control.* **1992**, *55*, 845–876. [CrossRef]
- 5. Wright, A.D. *Modern Control Design for Flexible Wind Turbines;* National Renewable Energy Laboratory: Golden, CO, USA, 2004.
- 6. Lather, J.; Dhillon, S.; Marwaha, S. Modern control aspects in doubly fed induction generator based power systems: A review. *Int. J. Adv. Res. Electr. Electr. Instrum. Eng.* **2013**, *2*, 2149–2161.
- 7. Trabelsi, R.; Khedher, A.; Mimouni, M.F.; M'Sahli, F. An Adaptive Backstepping Observer for on-line rotor resistance adaptation. *Int. J. IJ-STA* **2010**, *4*, 1246–1267.
- 8. Ullah, N.; Wang, S. High performance direct torque control of electrical aerodynamics load simulator using adaptive fuzzy backstepping control. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2014**, 229, 369–383. [CrossRef]
- 9. Rajendran, S.; Jena, D. Backstepping sliding mode control of a variable speed wind turbine for power optimization. *J. Mod. Power Syst. Clean Energy* **2015**, *3*, 402–410. [CrossRef]
- Fliess, M.; Join, C. Model-free control and intelligent pid controllers: Towards a possible trivialization of nonlinear control? *IFAC Proc. Vol.* 2009, 42, 1531–1550. [CrossRef]
- 11. Fuglsang, P.; Madsen, H.A. Optimization method for wind turbine rotors. *J. Wind. Eng. Ind. Aerodyn.* **1999**, *80*, 191–206. [CrossRef]
- 12. Selig, M.S.; Coverstone-Carroll, V.L. Application of a Genetic Algorithm to Wind Turbine Design. *J. Energy Resour. Technol.* **1996**, *118*, 22–28. [CrossRef]
- 13. Wiering, M.; Van Hasselt, H. Ensemble Algorithms in Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2008**, *38*, 930–936. [CrossRef]
- 14. Watkins, C.J.C.H. Learning from Delayed Rewards; King's College: London, UK, 1989.
- 15. Rummery, G.A.; Niranjan, M. *On-Line Q-Learning Using Connectionist Systems;* University of Cambridge, Department of Engineering Cambridge: Cambridge, UK, 1994.

- Sutton, R.S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*; University of Massachusetts: Amherst, MA, USA, 1996; pp. 1038–1044.
- 17. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [CrossRef]
- Wiering, M.A.; Van Hasselt, H. Two Novel On-policy Reinforcement Learning Algorithms based on TD(λ)-methods. In Proceedings of the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, Honolulu, HI, USA, 1–5 April 2007; pp. 280–287. [CrossRef]
- Sutton, R.S.; McAllester, D.A.; Singh, S.P.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*; AT&T Labs-Research: Florham Park, NJ, USA, 2000; pp. 1057–1063.
- 20. Baxter, J.; Bartlett, P. Infinite-Horizon Policy-Gradient Estimation. J. Artif. Intell. Res. 2001, 15, 319–350. [CrossRef]
- 21. Moore, A.W.; Atkeson, C.G. Prioritized sweeping: Reinforcement learning with less data and less time. *Mach. Learn.* **1993**, *13*, 103–130. [CrossRef]
- 22. Riedmiller, M. Neural Fitted Q Iteration—First Experiences with a Data Efficient Neural Reinforcement Learning Method. In *European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3720, pp. 317–328.
- Pang, H.; Gao, W. Deep Deterministic Policy Gradient for Traffic Signal Control of Single Intersection. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5861–5866.
- Ghouri, U.H.; Zafar, M.U.; Bari, S.; Khan, H.; Khan, M. Attitude Control of Quad-copter using Deterministic Policy Gradient Algorithms (DPGA). In Proceedings of the 2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE), Islamabad, Pakistan, 6–7 March 2019; pp. 149–153.
- 25. Glaessgen, E.; Stargel, D. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials, Honolulu, HI, USA, 23–26 April 2012; p. 1818.
- 26. Tao, F.; Sui, F.; Liu, A.; Qi, Q.; Zhang, M.; Song, B.; Guo, Z.; Lu, S.C.-Y.; Nee, A.Y.C. Digital twin-driven product design framework. *Int. J. Prod. Res.* **2018**, *57*, 3935–3953. [CrossRef]
- 27. Rajendran, S.; Jena, D. Control of Variable Speed Variable Pitch Wind Turbine at Above and Below Rated Wind Speed. *J. Wind. Energy* **2014**, 2014, 1–14. [CrossRef]
- 28. Boukhezzar, B.; Siguerdidjane, H. Comparison between linear and nonlinear control strategies for variable speed wind turbines. *Control Eng. Pr.* **2010**, *18*, 1357–1368. [CrossRef]
- 29. Ren, Y.; Li, L.; Brindley, J.; Shangguan, X.-C. Nonlinear PI control for variable pitch wind turbine. *Control. Eng. Pr.* **2016**, *50*, 84–94. [CrossRef]
- Al Younes, Y.; Drak, A.; Noura, H.; Rabhi, A.; El Hajjaji, A. Robust Model-Free Control Applied to a Quadrotor UAV. J. Intell. Robot. Syst. 2016, 84, 37–52. [CrossRef]
- 31. Al Younes, Y.; Drak, A.; Noura, H.; Rabhi, A.; El Hajjaji, A. Nonlinear Integral Backstepping–Model-Free Control Applied to a Quadrotor System. In Proceedings of the 10th International Conference on Intelligent Unmanned Systems, Montreal, Canada, 29 September–1 October 2014.
- Wei, C.; Zhang, Z.; Qiao, W.; Qu, L. Reinforcement-Learning-Based Intelligent Maximum Power Point Tracking Control for Wind Energy Conversion Systems. *IEEE Trans. Ind. Electron.* 2015, 62, 6360–6370. [CrossRef]
- Gu, S.; Lillicrap, T.; Sutskever, I.; Levine, S. Continuous Deep Q-Learning with Model-based Acceleration. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 12–18 July 2016; pp. 2829–2838.
- Srinivasan, S.; Lanctot, M.; Zambaldi, V.; Perolat, J.; Tuyls, K.; Munos, R.; Bowling, M. Actor-Critic Policy Optimization in Partially Observable Multiagent Environments. In *Advances in Neural Information Processing Systems*; Advances in Neural Information Processing Systems: New York, NY, USA, 2018; pp. 3422–3435.
- 35. Hasanvand, S.; Rafiei, M.; Gheisarnejad, M.; Khooban, M.-H. Reliable Power Scheduling of an Emission-Free Ship: Multi-Objective Deep Reinforcement Learning. *IEEE Trans. Transp. Electrif.* **2020**, *1*. [CrossRef]

- 36. Gheisarnejad, M.; Khooban, M.H. An Intelligent Non-integer PID Controller-based Deep Reinforcement Learning: Implementation and Experimental Results. *IEEE Trans. Ind. Electron.* **2020**, *1*. [CrossRef]
- Hajihosseini, M.; Andalibi, M.; Gheisarnejad, M.; Farsizadeh, H.; Khooban, M.-H. DC/DC Power Converter Control-Based Deep Machine Learning Techniques: Real-Time Implementation. *IEEE Trans. Power Electron.* 2020, 1. [CrossRef]
- 38. Khooban, M.H.; Gheisarnejad, M. A Novel Deep Reinforcement Learning Controller Based Type-II Fuzzy System: Frequency Regulation in Microgrids. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, 1–11. [CrossRef]
- Rodriguez-Ramos, A.; Sampedro, C.; Bavle, H.; De La Puente, P.; Campoy, P. A Deep Reinforcement Learning Strategy for UAV Autonomous Landing on a Moving Platform. J. Intell. Robot. Syst. 2018, 93, 351–366. [CrossRef]
- Gheisarnejad, M.; Farsizadeh, H.; Tavana, M.-R.; Khooban, M.H. A Novel Deep Learning Controller for DC/DC Buck-Boost Converters in Wireless Power Transfer Feeding CPLs. *IEEE Trans. Ind. Electron.* 2020, 1. [CrossRef]
- 41. Zeitouni, M.J.; Parvaresh, A.; Abrazeh, S.; Mohseni, S.-R.; Gheisarnejad, M.; Khooban, M.-H. Digital Twins-Assisted Design of Next-Generation Advanced Controllers for Power Systems and Electronics: Wind Turbine as a Case Study. *Inventions* **2020**, *5*, 19. [CrossRef]
- 42. He, R.; Chen, G.; Dong, C.; Sun, S.; Shen, X. Data-driven digital twin technology for optimized control in process systems. *ISA Trans.* **2019**, *95*, 221–234. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).