


Article

A Dynamic Control Model of the Blades Position for the Vertical-Axis Wind Generator by a Program Method

Ivaylo Stoyanov ^{1,*} , Teodor Iliev ^{2,*} , Alina Fazylova ^{3,4}  and Gulsara Yestemessova ⁵¹ Department of Electric Power Engineering, University of Ruse, 8 Studentska Str., 7004 Ruse, Bulgaria² Department of Telecommunications, University of Ruse, 8 Studentska Str., 7004 Ruse, Bulgaria³ Department of Robotics and Engineering Tools of Automation, Satbayev University, 22a Satpaev Str., Almaty 050013, Kazakhstan; a.fazylova@au.es.kz⁴ Department Electronics and Robotics Almaty, University of Power Engineering and Telecommunications, Almaty 050013, Kazakhstan⁵ Department of Transport Construction and Production of Building Materials, LB Goncharov Kazakh Automobile and Road Institute, pr. Rayymbeka, 417, Almaty 050061, Kazakhstan; sara_rus@mail.ru

* Correspondence: stoyanov@uni-ruse.bg (I.S.); tiliev@uni-ruse.bg (T.I.)

Abstract: This article discusses the construction of a dynamic model for controlling the position of the blades of a vertical-axis wind generator using an automatic approach; a method is presented that relates the rotation of the motor to the position of the blades, which allows the optimization of the operation of the control system. In the research process, an automatic approach is used, which makes it possible to carry out numerical calculations that predict the behavior of the system at various values of motor rotation. The model allows us to analyze the dependence of the position of the blades on the rotation of the motor and determine the optimal parameters of the mathematical control model. The main goal of our study is to develop a mathematical model of the mechanism for further adjustment of the wind turbine blade position control system depending on the wind speed.

Keywords: wind turbine; automated control system; blades; lever control system; gear



Citation: Stoyanov, I.; Iliev, T.; Fazylova, A.; Yestemessova, G. A Dynamic Control Model of the Blades Position for the Vertical-Axis Wind Generator by a Program Method. *Inventions* **2023**, *8*, 120. <https://doi.org/10.3390/inventions8050120>

Academic Editors: Luigi Fortuna and Arturo Buscarino

Received: 4 August 2023

Revised: 17 September 2023

Accepted: 21 September 2023

Published: 25 September 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wind energy, which has received very wide development in many countries of the world, now represents significant competition for traditional energy sources [1]. A particularly high level of wind energy development is observed in Southeast Asia and Europe [2]. For example, in 2018, in Denmark, 42% of all electricity produced was produced by wind turbines; in Portugal—27%; in Spain—20%; in Ireland—19%; in Germany—18.8%; in the Mediterranean region on the Greek islands—more than 21% [3]; and in the European Union as a whole—7.5% [4]. It is important to note that the share of renewable energy sources in the total energy consumption of the countries of the European Union in 2018 reached 17.5% [5]. In 2022, for the first time in the EU, solar and wind energy came out on top in terms of generation, overtaking all other sources. Combined, wind and solar energy provided a record fifth of the EU's electricity (22.3%)—a contribution larger for the first time than either nuclear (21.9%) or gas (19.9%) generation [6]. According to the Union of the Electricity Industry, Eurelectric, two-thirds of the electricity generated in Europe in the first half of 2023 is carbon-free [7]. The share of renewable energy sources is 40% of the total production, while electricity generation based on fossil fuels has fallen by 18% [8]. By 2030, up to 80% of the EU's electricity could be produced without the use of fossil fuels [8]. Figure 1 shows the dynamics of simple electricity generation by different types of sources from 2000 to 2022.

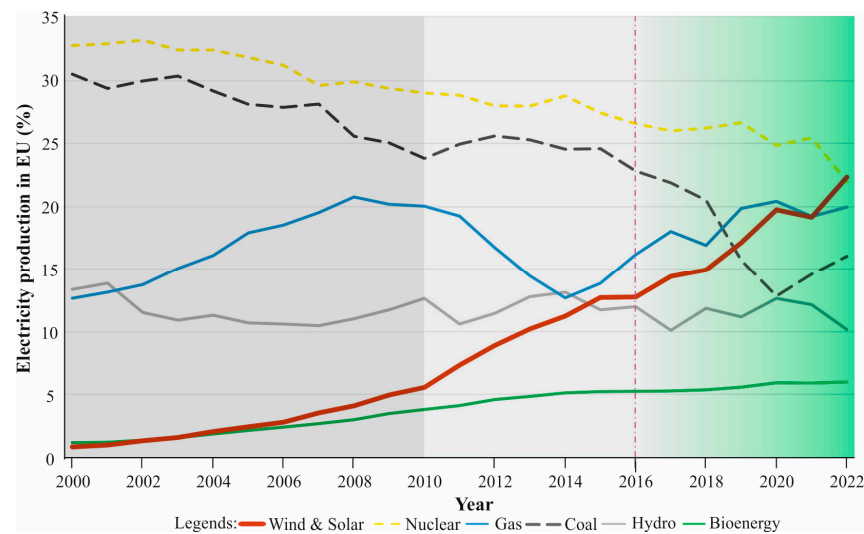


Figure 1. Electricity production by years in the EU in % [9].

The observed period can be divided into three parts. First, from 2000 to 2010, the main share of generated electricity was from nuclear and thermal power plants. In the second period, from 2010 to 2016, their relative share in the total gross production gradually began to decrease, while in the last period (after 2016), an increase in the electricity produced from renewable energy sources (RES) was observed. In the last three years, after 2020, the share of electricity produced by RES exceeds 50% of the total production.

Wind turbines are devices designed to capture the kinetic energy of the wind and convert it into useful electrical energy [10]. They play an important role in promoting sustainable energy solutions and come in two main types: Horizontal Axis Wind Turbine (HAW) and Vertical Axis Wind Turbine (VAW). At present, HCVs dominate the market, but HCVs remain the subject of active research [11]. Despite some comparative disadvantages, such as lower aerodynamic efficiency, problems with self-launching at low wind speeds, and design difficulties when scaling to large levels, HFWs have certain advantages, including their applicability on a small scale and the ability to generate power from various wind directions, which makes them suitable for distributed energy systems in urban environments [12].

One of the main tasks of wind turbine control is to regulate the output power in response to changes in wind speed, which is called the point of maximum power, which is critical for optimal power generation [13]. The various loads they experience when extracting energy from the wind are carefully analyzed when designing wind turbines. Proper structural design and choice of materials are key to the efficient operation of wind turbines [14]. The wind turbine blade position control system plays an important role in optimizing its operation when wind conditions change. This system provides accurate positioning of the turbine blades at the optimum angle to maximize the capture of wind energy and mitigate the impact of adverse meteorological events such as strong winds and gusts [15]. The angle of inclination of the blades directly affects the aerodynamic load and the extraction of energy from the wind. There are two main categories of blade position control systems: passive and active control systems.

Passive control systems use the built-in aerodynamic and mechanical characteristics of the blades to control their position [16]. Such an approach might include changing the profile of the blades or including flaps that automatically adjust the angle of attack based on wind speed. Although passive systems are easy to use and do not require active intervention, their effectiveness can be limited due to their static nature. In contrast, active control systems use complex electronics, sensors, and actuators to dynamically control the position of the blades [17]. This real-time adjustment capability allows blade rotation to be fine-tuned based on factors such as wind speed and generator load, resulting in

increased efficiency and adaptation to changing wind conditions. However, active systems require more complex infrastructure and complex equipment [14]. Algorithmic control systems represent another promising approach using advanced algorithms and software for optimization purposes [15]. Some authors suggested using PID controllers with the extension of derivation and integration orders from integer to non-integer numbers with extended functions in comparison to classical ones [18]. These algorithms may include wind speed prediction, adaptive blade angle control, generator optimization, and other techniques that provide the flexibility and potential for optimal generator performance. Other researchers proposed adaptive compensation by using a multilayer perceptron neural network. The results of this method offered a practical implementation [19].

However, the implementation of such systems requires significant computing resources and experience in programming algorithms [16]. Selecting an appropriate blade position control system for a wind turbine project requires a careful assessment of project requirements, costs, technical resources, and other relevant factors. Each system has its advantages and disadvantages, which makes the decision-making process important and complex [17]. However, the ultimate goal remains the same: to achieve the optimal position of the wind turbine blades for maximum efficiency and overall energy efficiency. With the pursuit of a green and environmentally friendly future, the continuous development of blade position control technologies paves the way for a more efficient and sustainable use of wind energy [20].

2. Related Work

In the ever-expanding field of renewable energy research, dynamic blade control systems for vertical-axis wind turbines have attracted considerable attention, and numerous studies are aimed at improving their performance and efficiency [21]. These studies have focused on the development of complex mathematical models that allow comprehensive analysis of the behavior of wind turbine blades as a function of wind speed, direction, and environmental conditions [22]. One common approach involves integrating advanced control algorithms that continuously collect real-time data from a network of sensors strategically placed on the wind turbine structure. These sensors closely monitor critical parameters such as wind speed, wind direction, temperature, and blade angles [23]. The collected data is then processed and transmitted to the control system, which independently adjusts the angles of the blades to achieve the optimal configuration for power production in any given wind. Such adaptive control systems are critical to maximizing power generation and improving the overall efficiency of wind turbines [24].

In addition, the search for innovative solutions has led to recent research into incorporating smart materials into the design of wind turbine blades. These smart materials have exceptional properties that allow them to respond dynamically to external stimuli. By integrating these materials into the blade design, researchers can use their unique capabilities to adjust the shape and orientation of the blade to optimize energy capture over a wide range of wind speeds and directions [25]. This adaptability improves the overall performance of wind turbines and contributes to a more stable and efficient power generation process. Despite significant progress in research on dynamic blade control systems, several problems remain in their development and implementation. One of the main challenges is to achieve a delicate balance between maximizing energy output and ensuring the durability of the turbine's mechanical components [26]. Excessive mechanical stress resulting from frequent knife adjustments can cause premature wear and shorten the life of the system. Consequently, researchers are constantly working to optimize control algorithms to minimize turbine wear and maximize power generation. The reliable operation of dynamic blade control systems relies heavily on advanced sensor technology and robust communications infrastructure. These systems must constantly communicate critical data between sensors and control units in real time to ensure fast and accurate blade adjustments [27]. Ensuring the reliability and accuracy of this communication network is vital to the smooth operation and performance of wind turbines. Ongoing research and

development in the field of dynamic blade control systems for vertical axis wind turbines holds great promise for the future of renewable energy [6]. The combination of advanced mathematical modeling, data-driven control algorithms, and innovative materials paves the way for significant advances in energy efficiency, system reliability, and profitability of wind power generation. As these technologies continue to evolve and improve, they will play a critical role in the global transition to cleaner and greener energy sources, with a significant impact on climate change mitigation and a sustainable energy future.

3. Materials and Methods

A 3D model of a vertical-axis wind turbine blade position control system was developed using computer-aided design (CAD) principles in the Solidworks® 2019 software environment. The core methodology of the 3D model is built on CAD principles [28]. Through the CAD process, the physical components of a wind turbine, including the blades, motor, and controls, were carefully and accurately modeled. This involved defining the geometry and orientation of each component to create a detailed digital representation of the entire control system. This 3D model serves as the main tool for a general understanding of the behavior of the system. This made it possible to carry out visualization, analysis, and optimization, and thanks to 3D modeling, it was possible to create a physical laboratory model of the control system on a 3D printer for conducting experiments.

This 3D model provides a comprehensive representation of the control system and facilitates an in-depth analysis of its dynamics [29]. The development of this virtual model is instrumental in predicting the behavior of the control system at various engine speed values, thus enabling the examination of multiple scenarios and input parameters. Through such analysis, optimal control strategies can be determined, and the performance of the control system can be thoroughly evaluated. The foundation of the dynamic model lies in the implementation of the Runge–Kutta method, a widely used numerical technique for solving differential equations. This iterative method yields an approximate solution with the desired level of accuracy, which is essential for understanding the relationship between motor rotation and blade rotation [30]. This insight into the motor–blade relationship further facilitates the analysis and optimization of the control system. Additionally, during the control system’s development process, a crucial aspect involved approximating the relationship between motor activation time and blade angle. The data obtained were subjected to polynomial approximation utilizing the SciPy library [31]. Given the blade’s angle dependency on time, the polynomial function serves as a fitting approach to accurately represent this intricate relationship [32]. The mathematical function derived from the polynomial approximation proves to be a valuable tool for comprehending and predicting the blade position control system’s behavior. Through a thorough examination of this function, it becomes possible to discern the optimal parameters for effectively controlling the wind turbine blade position and, subsequently, maximizing the overall system’s efficiency [33]. Figure 2 visually showcases the 3D model of the control system, emphasizing its significance as a powerful tool for simulations and analysis. This model allows researchers and engineers to visualize the intricate interplay between various components and gain a deeper understanding of the system’s behavior under different operating conditions. The development of a sophisticated 3D model of the blade position control system for vertical-axis wind generators, combined with numerical calculations and the utilization of the Runge–Kutta method [34], has yielded invaluable insights into system dynamics and performance. The polynomial approximation of the motor activation time and blade angle relationship further enhances the model’s accuracy, enabling researchers to make informed decisions regarding the optimal control strategies for maximizing wind turbine efficiency. This software environment represents a significant advancement in the field of wind turbine technology, contributing to the quest for sustainable and efficient renewable energy.

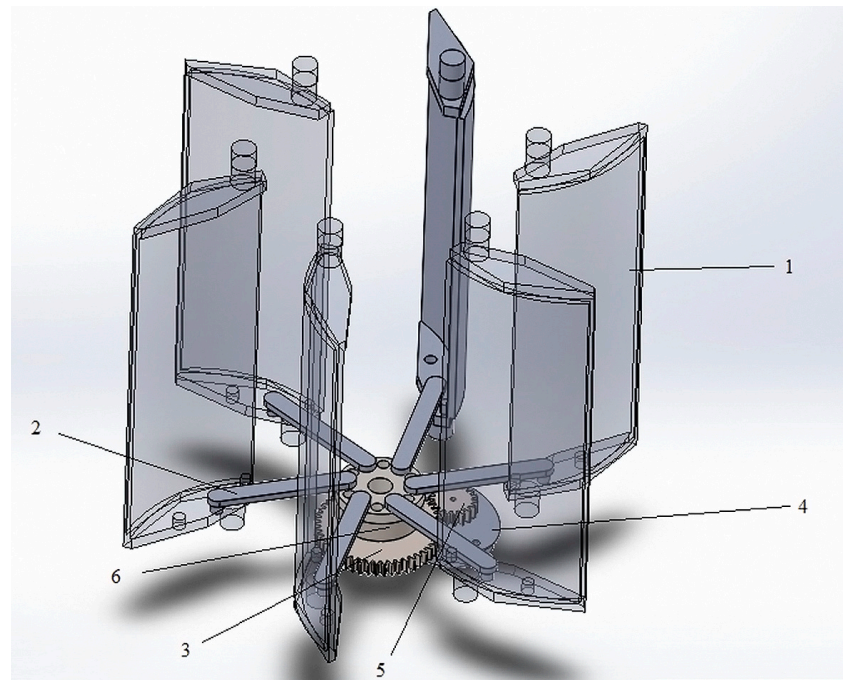


Figure 2. General 3D model of the wind turbine blade position control system [31]: 1—blade, 2—lever system, 3 and 5—gear wheels, 4—motor, 6—disk.

The mechanism described in the article is designed to control the blades of a wind generator (1) effectively. It utilizes a lever system (2) that is mounted on a movable disk (6). The key component in the system is the motor (4), which initiates the rotation. As the motor rotates, it drives a gear train (3, 5), which in turn causes the disk to rotate. This rotational motion of the disk ultimately adjusts the positions of the blades. By employing this lever system and motor-driven gear train, the mechanism allows for precise control over the blade positions. This control is crucial for optimizing the wind generator's performance in capturing wind energy. The ability to adjust the blade positions ensures that the generator can efficiently harness the available wind resources under different conditions. The appearance of the wind turbine in several modes of operation is shown in Figure 3 [35].

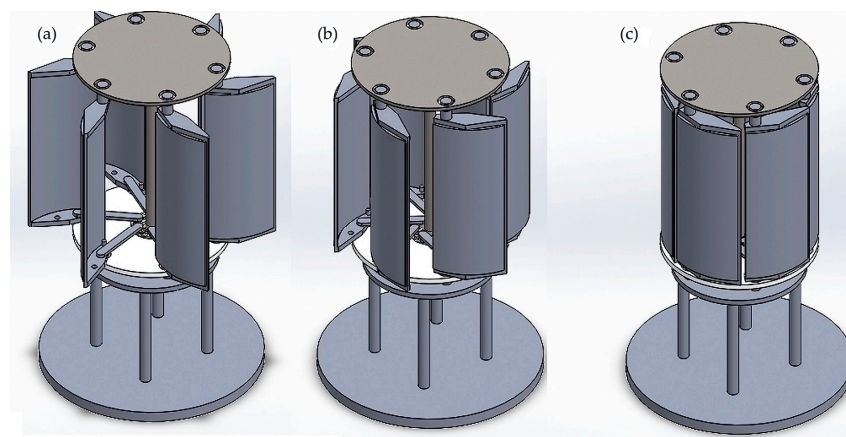


Figure 3. Wind turbine blade position control modes: (a)—normal mode of operation of the wind generator, in which the control system is not activated; (b)—adjustment of the position of the blades at wind speeds above 25 m/s [35]; (c)—emergency mode, in which the wind generator does not generate electricity.

The process of determining the optimal parameters of a mathematical model of a wind turbine control system involves a combination of modeling, experimentation, and control algorithm tuning. The model was used to predict the behavior of the system under conditions of changes in wind speed and engine activation time. A proportional-integral (PI) controller was used to control the position of the blades. The controller has two key parameters: k_p (proportional gain) and k_i (integral gain). The optimal values of those parameters have been determined experimentally. The laboratory experiments were conducted using a 3D-printed model of a wind turbine. This experiment involved exposing a model to airflow at 25 m per second and measuring the wind generator's output voltage over time (Figure 3). The stabilization time required for a wind turbine to achieve a stable rotation speed was determined to be 7 s. Based on the analysis of the dynamic properties of the wind power plant, a control program was developed. The control program is executed in real-time, and its performance was validated using a wind turbine model to achieve a blade angle of 90 degrees in 7 s at a wind speed of 25 m per second.

In this case, the selection of the optimal parameters of a wind turbine control system is a complex and multifaceted process, which is influenced by many factors, including changes in wind speed and engine rotation. However, determining the specific optimal parameters depends largely on the wind turbine materials, environmental conditions, and operational requirements. Factors such as the structural integrity of the turbine, the aerodynamic characteristics of its blades, the local climate, and even the desired power output come into play.

4. Dynamic Wind Generator Model

This section describes the dynamics of gear train movement, which allows you to calculate the angles, angular velocities, and accelerations of gears depending on external forces, system parameters, and initial conditions. Figure 4 shows a diagram of the gearing of gears.

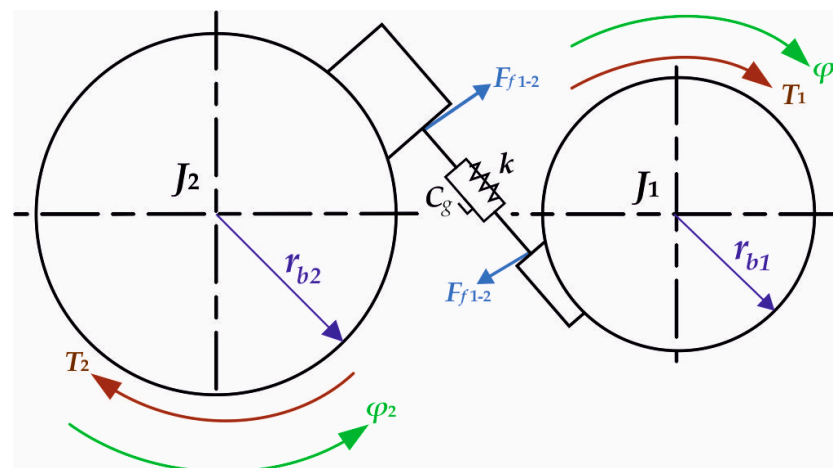


Figure 4. Gear diagram: k is the dynamic coefficient; J_1, J_2 are the moment of inertia; C_g is the elasticity coefficient; T is the torque.

Generalized coordinates of the change in the angle of the teeth q_1 and q_2 , which correspond to the angles φ_1 and φ_2 :

$$q_1 = \varphi_1, q_2 = \varphi_2. \quad (1)$$

Equation (2) is the mathematical expression of motion and describes Newton's second law for gearing. It connects the change in kinetic energy (dL/dq_1), dissipative forces (dD/dq_1) with external forces Q_1 acting on the system [36]. In this case, L represents the

Lagrange function. Since the degree of freedom of the above system is two, the number of Lagrange equations will be two:

$$\frac{d}{dt} \left(\frac{dL}{dq_1} \right) - \frac{dL}{dq_1} + \frac{dD}{dq_1} = Q_1, \quad (2)$$

where dL is the motion of the link driven by applied forces.

Next, the dependence of the moment of inertia J_1 on the angular velocity was determined φ'_1 :

$$\frac{dL}{d\varphi'_1} = J_1 \varphi'_1. \quad (3)$$

The time derivative was taken from Equation (3) and the angular acceleration was determined φ''_1 :

$$\frac{d}{dt} \left(\frac{dL}{d\varphi'_1} \right) = J_1 \varphi''_1. \quad (4)$$

Equations (5) and (6) describe the change in the Lagrangian function (L) concerning the angle φ_2 for the second gear in the transmission [37]:

$$\frac{dL}{d\varphi_1} = -k_{g1-2}(t) \times (r_{b1}\varphi_1 - r_{b2}\varphi_2) \times r_{b1} - \frac{1}{2}(r_{b1}\varphi_1 - r_{b2}\varphi_2)^2 \times \frac{dk_{g1-2}(t)}{d\varphi_1}. \quad (5)$$

Equation (5) describes the part of the Lagrange function (L) that depends on φ_1 . The Lagrange function in mechanics is used to determine the energy of a system. In this case, the terms of this equation determine the potential and kinetic energy of the system, as well as energy losses (dissipation) in the system. The coefficients $k_{g1-2}(t)$ and $dk_{g1-2}(t)/d\varphi_1$ can depend on time t and represent system characteristics. The coefficients $k_{g1-2}(t)$ describes the rigidity (or elasticity) of the connection between the teeth of the first and second gear train, and $dk_{g1-2}(t)/d\varphi_1$ can depend on time t and represent system characteristics. These coefficients r_{b1} and r_{b2} represent the radii of the first and second gears, respectively. They are used to determine the mechanical characteristics of the transmission, such as moment of inertia and torque, and can depend on the size and geometry of the teeth.

Equation (6) is a simplified version of Equation (5) and describes the part of the Lagrange function that depends only on φ_1 .

$$\frac{dL}{d\varphi_1} = -k_{g1-2}(t) \times (r_{b1}\varphi_1 - r_{b2}\varphi_2) \times r_{b1}. \quad (6)$$

With the help of Equation (6) it can be ignored dissipation and the additional term associated with the derivative $dk_{g1-2}(t)/d\varphi_1$.

Equation (7) defines the energy dissipation term for the first gear:

$$\frac{dD}{d\varphi'_1} = C_{g1-2} \times (r_{b1}\varphi'_1 - r_{b2}\varphi'_2) \times r_{b1}, \quad (7)$$

where C_{g1-2} is the friction force between the teeth of the first and second gears. It determines how quickly energy is lost during motion and can include both energy dissipation in the form of heat and other frictional effects. The specific meanings of coefficients $k_{g1-2}(t)$, C_{g1-2} , r_{b1} , r_{b2} , depending on the physical properties of the gears and the system being modeled.

The total force Q_1 is determined by the Formula [38]:

$$Q_1 = T_1/\varphi_1 - r_{f1} \times [k_{g1-2}(t) \times (r_{b1}\varphi_1 - r_{b2}\varphi_2) + C_{g1-2} \times (r_{b1}\varphi'_1 - r_{b2}\varphi'_2)] \times \mu_{1-2}, \quad (8)$$

where T_1 is the torque on the first shaft.

Equation (9) is the equation of motion for the first gear, relating the moment of inertia, kinetic and dissipative energies, external forces, and system parameters:

$$J_1 \varphi''_1 + k_{g1-2}(t) \times (r_{b1} \varphi_1 - r_{b2} \varphi_2) \times r_{b1} + C_{g1-2} \times (r_{b1} \varphi'_1 - r_{b2} \varphi'_2) \times r_{b1} = T_1 \quad (9)$$

Equations (10)–(12) are similar to Equations (2)–(4) but refer to the second gear:

$$\frac{d}{dt} \left(\frac{dL}{dq'_2} \right) - \frac{dL}{dq_2} + \frac{dD}{dq'_2} = Q_2; \quad (10)$$

$$\frac{dL}{d\varphi'_2} = J_2 \varphi'_2; \quad (11)$$

$$\frac{d}{dt} \left(\frac{dL}{d\varphi'_2} \right) = J_2 \varphi''_2. \quad (12)$$

Equation (13) describes the change in the Lagrangian function (L) concerning the angle φ_2 for the second gear in the transmission

$$\frac{dL}{d\varphi_2} = k_{g1-2}(t) \times (r_{b1} \varphi_1 - r_{b2} \varphi_2) \times r_{b1}. \quad (13)$$

This is only true for φ_2 and does not depend on $dk_{g1-2}(t) / d\varphi_2$, because in this case it is assumed that there is no derivative concerning φ_2 .

Equation (14) defines the energy dissipation function (D), which depends on the time derivative of φ_2 (φ'_2). The dissipation function is usually associated with energy losses in the system, for example, due to friction or other dissipative forces.

$$\frac{dD}{d\varphi'_2} = C_{g1-2} \times (r_{b1} \varphi'_1 - r_{b2} \varphi'_2) \times r_{b1} \quad (14)$$

All these equations form the basis for the analysis of motion and energy losses in a given mechanical system, using the generalized coordinates φ_1 and φ_2 .

Equation (15) is an expression for the external forces Q_2 acting on the second gear [11]:

$$Q_2 = T_2 / \varphi_2 - r_{f2} \times [k_{g1-2}(t) \times (r_{b1} \varphi_1 - r_{b2} \varphi_2) + C_{g1-2} \times (r_{b1} \varphi'_1 - r_{b2} \varphi'_2)] \times \mu_{1-2}. \quad (15)$$

where T_2 is the torque on the second shaft.

Thus, the equation of motion for the gear transmission will take the form [39]:

$$J_2 \varphi''_2 - k_{g1-2}(t) \times (r_{b1} \varphi_1 - r_{b2} \varphi_2) \times r_{b1} - C_{g1-2} \times (r_{b1} \varphi'_1 - r_{b2} \varphi'_2) \times r_{b1} = T_2 \quad (16)$$

Next, it is necessary to transform the dynamic equation into a simple system of equations of the first series $V_1 = \varphi_1, V_2 = \varphi'_1, V_3 = \varphi_2, V_4 = \varphi'_2, V'_1 = \varphi'_1, V'_2 = \varphi''_1, V'_3 = \varphi'_2, V'_4 = \varphi''_2$.

$$J_1 V'_2 + k_{g1-2}(t) \times (r_{b1} V_1 - r_{b2} V_2) \times r_{b1} + C_{g1-2} \times (r_{b1} V_2 - r_{b2} V_4) \times r_{b1} = (T_1 - r_{f1} \times [k_{g1-2}(t) \times (r_{b1} \varphi_1 - r_{b2} \varphi_2) + C_{g1-2} \times (r_{b1} \varphi'_1 - r_{b2} \varphi'_2)] \times \mu_{1-2}) = A; \quad (17)$$

$$J_2 V'_4 + k_{g1-2}(t) \times (r_{b1} V_1 - r_{b2} V_3) \times r_{b1} + C_{g1-2} \times (r_{b1} V_2 - r_{b2} V_4) \times r_{b1} = (T_2 - r_{f2} \times [k_{g1-2}(t) \times (r_{b1} V_1 - r_{b2} V_3) + C_{g1-2} \times (r_{b1} V_2 - r_{b2} V_4)] \times \mu_{1-2}) = B. \quad (18)$$

$$\begin{aligned}
 V_2 &= V_1; \\
 V_{t2} &= \frac{1}{J_1} \{A - K\}; \\
 V_{t4} &= \frac{1}{J_2} \{B + T\}; \\
 V_1|_{t=0} &= 0; \\
 V_2|_{t=0} &= 0; \\
 V_3|_{t=0} &= 0; \\
 V_4|_{t=0} &= 0.
 \end{aligned}$$

The technique described above shows the dynamics of a transmission system using gears. Variables V_1 , V_2 , V_3 , and V_4 represent the angular movements and speeds of the gears involved in the transmission (Figure 5). The following values were arbitrarily adopted for the system parameters [40]:

```

Mass of the first gear: m1 = 5 kg
Mass of the second gear: m2 = 8 kg
Radius of the first gear: rb1 = 0.3 m
Second gear radius: rb2 = 0.8 m
Stiffness coefficient: k = 100 N/m
Dissipation coefficient: C = 5 N s/m
Moment of inertia of the first gear: J1 = 2 kg × m2
Moment of inertia of the second gear: J2 = 1.5 kg × m2
External force Q1: Q1 = 50 N
External force Q2: Q2 = 60 N
Friction coefficient: μ = 0.2
First, let's agree that the initial values for all variables are
equal to zero:
V1(0) = 0
V2(0) = 0
V3(0) = 0
V4(0) = 0

```

Figure 5. Program code for the adopted system parameters.

The provided code implements a numerical method called the Runge–Kutta method to solve a system of differential equations describing the dynamics of a gear transmission system. Differential equations reflect the behavior of a system as it evolves over time. The differential equations that describe the behavior of the system are defined in the system equations function. These equations are based on the physical properties of gears and the forces applied to them. The variables V_1 , V_2 , V_3 , and V_4 represent the angular velocities and gear positions. The physical parameters of the system are defined as constants in the system equations function. These parameters include masses, radii, stiffnesses, damping, moments of inertia, external forces, and friction coefficients. The initial conditions of the system are set equal to zero for all variables ($V_1(0) = 0$, $V_2(0) = 0$, $V_3(0) = 0$, $V_4(0) = 0$).

Further, the presented calculation method was written in the form of a program code in the Python language, taking into account the numerical parameters of the system for performing the calculation (Figure 6):


```

import numpy as np
def system_equations(t, V):
    V1, V2, V3, V4 = V
    m1 = 5
    m2 = 8
    r_b1 = 0.3
    r_b2 = 0.8
    k = 100
    C = 5
    J1 = 2
    J2 = 1.5
    T1 = 50
    T2 = 60
    mu = 0.2
    r_f1 = 1
    r_f2 = 1
    V1_dot = V2
    V2_dot = (T1 - r_f1 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
V2 - r_b2 * V4))) * mu / J1 - (k * (r_b1 * V1 - r_b2 * V2) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J1
    V3_dot = V4
    V4_dot = (T2 - r_f2 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
V2 - r_b2 * V4))) * mu / J2 - (k * (r_b1 * V1 - r_b2 * V3) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J2
    return [V1_dot, V2_dot, V3_dot, V4_dot]
V_initial = [0, 0, 0, 0]
t_start = 0
t_end = 10
dt = 0.01
t = np.arange(t_start, t_end, dt)
V_solution = np.zeros((len(t), 4))
V_solution[0] = V_initial
for i in range(len(t) - 1):
    k1 = dt * np.array(system_equations(t[i], V_solution[i]))
    k2 = dt * np.array(system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k1))
    k3 = dt * np.array(system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k2))
    k4 = dt * np.array(system_equations(t[i] + dt, V_solution[i] +
k3))
    V_solution[i + 1] = V_solution[i] + (1 / 6) * (k1 + 2 * k2 + 2 *
k3 + k4)
for i in range(len(t)):
    print(f"time: {t[i]}, Values: {V_solution[i]}")

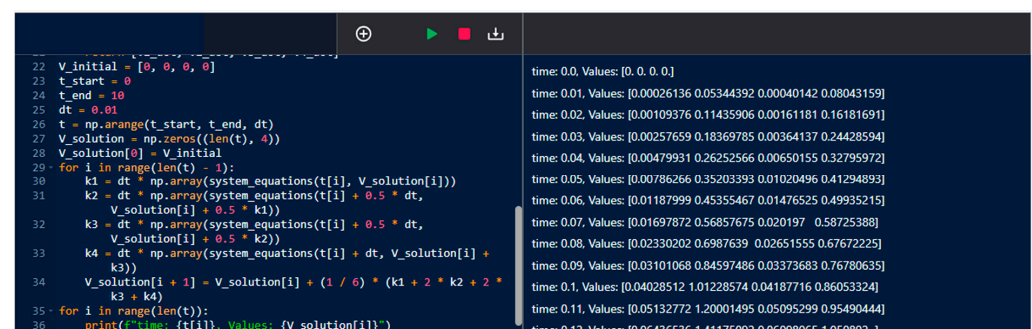
```

Figure 6. Python code of the calculation method.

The simulation time interval is determined by the parameters t_{start} , t_{end} , and dt (time step). In code, it models the system from t_{start} to t_{end} with a time step dt . Numerical integration (Runge–Kutta method): the main loop is repeated in time steps from t_{start} to t_{end} . At each time step, it calculates the derivatives of the variables ($V1_{\text{dot}}$, $V2_{\text{dot}}$, $V3_{\text{dot}}$, $V4_{\text{dot}}$) using the Runge–Kutta method. This involves calculating the values of $k1$, $k2$, $k3$, and $k4$, which are used to update the variables at the next time step. The simulation results, i.e., the values of $V1$, $V2$, $V3$, and $V4$ at each time step, are stored in the V_{solution} array. Finally, the code prints the time and the corresponding values of $V1$, $V2$, $V3$, and $V4$ at each time step.

The presented program code plays a crucial role in determining the speed of rotation of the gear. It accomplishes this by utilizing a series of expressions, specifically expressions (1)–(18), which are programmed into the code. These expressions represent the mathematical calculations and algorithms necessary to calculate the gear's rotational speed accurately. By inputting the relevant parameters and variables into the program, such as the wind speed, gear ratio, and other relevant factors, the code can calculate and determine the optimal rotational speed for the gear.

There is a visualization of the program's output in Figure 7. This figure provides a graphical representation of the results obtained from running the program. It may include plotted data points, curves, or other relevant visual elements that demonstrate the relationship between the input parameters and the resulting rotational speed of the gear.



```

22 V_initial = [0, 0, 0, 0]
23 t_start = 0
24 t_end = 10
25 dt = 0.01
26 t = np.arange(t_start, t_end, dt)
27 V_solution = np.zeros((len(t), 4))
28 V_solution[0] = V_initial
29 for i in range(len(t) - 1):
30     k1 = dt * np.array(system_equations(t[i], V_solution[i]))
31     k2 = dt * np.array(system_equations(t[i] + 0.5 * dt,
32                                     V_solution[i] + 0.5 * k1))
33     k3 = dt * np.array(system_equations(t[i] + 0.5 * dt,
34                                     V_solution[i] + 0.5 * k2))
35     k4 = dt * np.array(system_equations(t[i] + dt, V_solution[i] +
36                                     k3))
37     V_solution[i + 1] = V_solution[i] + (1 / 6) * (k1 + 2 * k2 + 2 *
38                                     k3 + k4)
39 for i in range(len(t)):
40     print(f"time: {t[i]}, Values: {V_solution[i]}")

```

time: 0.0, Values: [0. 0. 0. 0.]

time: 0.01, Values: [0.00026136 0.05344392 0.00040142 0.08043159]

time: 0.02, Values: [0.00109376 0.11435906 0.00161181 0.16181691]

time: 0.03, Values: [0.00257659 0.18369785 0.00364137 0.24428594]

time: 0.04, Values: [0.00479931 0.26252566 0.00650155 0.32795972]

time: 0.05, Values: [0.00786266 0.35203393 0.01020496 0.41294893]

time: 0.06, Values: [0.01187999 0.45355467 0.01476525 0.49935215]

time: 0.07, Values: [0.01697872 0.56857675 0.020197 0.58725388]

time: 0.08, Values: [0.02330202 0.6987639 0.02651555 0.67672225]

time: 0.09, Values: [0.03101068 0.84597486 0.03373683 0.76780635]

time: 0.1, Values: [0.04028512 1.01228574 0.04187716 0.86053324]

time: 0.11, Values: [0.05132772 1.20001495 0.05095299 0.95490444]

time: 0.12, Values: [0.06436536 1.41175092 0.06088065 1.050892 1

Figure 7. Visualization of program output for determining gear rotation speed.

Following Figure 2, the energy from the gear transmission is transferred to the disk, and then the position of the blades is controlled through the lever system.

Figure 8 shows the complete program code for calculating the position of the blades:

```
import numpy as np
import pandas as pd
def gear_system_equations(t, V):
    V1, V2, V3, V4 = V
    m1 = 10
    m2 = 8
    r_b1 = 1
    r_b2 = 0.8
    k = 100
    C = 5
    J1 = 2
    J2 = 1.5
    T1 = 50
    T2 = 60
    mu = 0.2
    r_f1 = 1
    r_f2 = 1
    V1_dot = V2
    V2_dot = (T1 - r_f1 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
V2 - r_b2 * V4))) * mu / J1 - (k * (r_b1 * V1 - r_b2 * V2) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J1
    V3_dot = V4
    V4_dot = (T2 - r_f2 * (k * (r_b1 * V1 - r_b2 * V3) + C * (r_b1 *
V2 - r_b2 * V4))) * mu / J2 - (k * (r_b1 * V1 - r_b2 * V3) * r_b1 + C *
(r_b1 * V2 - r_b2 * V4) * r_b1) / J2
    return [V1_dot, V2_dot, V3_dot, V4_dot]
def lever_system_equations(t, theta):
    motor_torque = 10
    motor_ramp_time = 2
    lever_length = 1
    blade_length = 0.5
    lift_coefficient = 0.1
    if t < motor_ramp_time:
        motor_torque_current = motor_torque * t / motor_ramp_time
    else:
        motor_torque_current = motor_torque
    angular_acceleration = motor_torque_current / (blade_length *
lever_length)
    theta_dot = angular_acceleration * t
    return theta_dot
V_initial = [0, 0, 0, 0]
theta_initial = 0
t_start = 0
t_end = 10
dt = 0.01
t = np.arange(t_start, t_end, dt)
V_solution = np.zeros((len(t), 4))
V_solution[0] = V_initial
theta_solution = np.zeros_like(t)
theta_solution[0] = theta_initial
for i in range(len(t) - 1):
    k1 = dt * np.array(gear_system_equations(t[i], V_solution[i]))
    k2 = dt * np.array(gear_system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k1))
    k3 = dt * np.array(gear_system_equations(t[i] + 0.5 * dt,
V_solution[i] + 0.5 * k2))
    k4 = dt * np.array(gear_system_equations(t[i] + dt,
V_solution[i] + k3))
    V_solution[i + 1] = V_solution[i] + (1 / 6) * (k1 + 2 * k2 + 2 *
k3 + k4)
    theta_dot = lever_system_equations(t[i], theta_solution[i])
    theta_solution[i + 1] = theta_solution[i] + dt * theta_dot
R = 0.8
D = 0.5
V_disk = [(R / D) * V[1] for V in V_solution]
data = {'Time (sec)': t, 'V1': V_solution[:, 0], 'V2': V_solution[:,
1], 'V3': V_solution[:, 2], 'V4': V_solution[:, 3], 'V_disk': V_disk,
'Blade angle (rad)': theta_solution}
df = pd.DataFrame(data)
df['Blade angle (deg)'] = np.rad2deg(df['Blade angle (rad)'])
print(df)
```

Figure 8. Program code for calculating the position of the blades.

The initial conditions and simulation time steps are set up, and arrays to store the solution are initialized. The for loop iterates over time steps, calculating k_1 , k_2 , k_3 , and k_4 values based on the Runge–Kutta formula. The solution is updated at each time step using the weighted sum of these k values. The result is stored in the $V_solution$ array, which represents the system's state variables (e.g., velocities). Polynomial approximation is employed to approximate the relationship between motor activation time and blade angle. The `lever_system_equations` function defines the equations governing the lever system, including parameters like motor torque, ramp time, lever length, blade length, and lift coefficient. A conditional statement is used to vary the motor torque with time, simulating

a ramp-up in torque. Angular acceleration is calculated based on the motor torque. The $\theta_{\dot{}}$ value represents the rate of change of blade angle with time.

After running this program code, calculations of the angles of rotation of the blades were obtained depending on the time the motor was turned on (Table 1).

Table 1. Result of program execution.

No	Time (s)	Blade Angle (rad)	Blade Angle (deg)
0	0.00	0.000000	0.000000
1	0.02	0.000040	0.002303
2	0.04	0.000322	0.018427
3	0.06	0.001085	0.062190
4	0.08	0.002573	0.147414
5	0.10	0.005025	0.287918
6	0.12	0.008683	0.497523
7	0.14	0.013789	0.790048
8	0.16	0.020583	1.179314
9	0.18	0.029307	1.679141
10	0.20	0.040201	2.303348
11	0.22	0.053508	3.065756
12	0.24	0.069467	3.980185
13	0.26	0.088322	5.060455
14	0.28	0.110312	6.320387
15	0.30	0.135678	7.773799
16	0.32	0.164663	9.434513
17	0.34	0.197508	11.316348
18	0.36	0.234452	13.433125
19	0.38	0.275739	15.798663
20	0.40	0.321608	18.426783
21	0.42	0.372302	21.331305
22	0.44	0.428060	24.526049
23	0.46	0.489126	28.024834
24	0.48	0.555739	31.841482
25	0.50	0.628141	35.989811
26	0.52	0.706573	40.483643
27	0.54	0.791276	45.336797
28	0.56	0.882492	50.563094
29	0.58	0.980462	56.176352
30	0.60	1.085427	62.190394
31	0.62	1.197628	68.619038
32	0.64	1.317307	75.476105
33	0.66	1.444704	82.775414

Based on the data obtained, the following observations can be made: when the engine is turned on with the previously set parameters, the blades begin to rotate from zero angle and gradually increase their angle of rotation (column 3 in Table 1 shows the value of the blade rotation angle in radians, column 4 in degrees). The increase in the angle of rotation of the blade occurs gradually and, by the end of the simulation, reaches a maximum value of 83 degrees. Thus, to fully bring the wind generator into emergency operation, as shown in Figure 3c, it is necessary to keep the engine on for 0.6 s. Next, you need to derive a mathematical function according to Table 1. To do this, an approximation of the dependence between the engine on time and the angle of rotation of the blades is carried out.

Using the SciPy library, a polynomial approximation of the data was performed. Since the angle of rotation of the blades depends on time, a polynomial function can be used. Below is the code that performs the approximation and outputs the resulting mathematical function [41]. Further, according to the data obtained, it is necessary to derive a mathematical function according to the data in Table 1. To do this, an approximation of the dependence between the time of switching on the motor and the angle of rotation of the blades is carried out. Since the angle of rotation of the blades depends on time, a

polynomial function can be used. Figure 9 shows the program code that performs the approximation and outputs the resulting mathematical function [41]:

```
import numpy as np
from scipy.optimize import curve_fit

time = np.array([0.00, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16,
0.18, 0.20, 0.22, 0.24, 0.26, 0.28, 0.30, 0.32, 0.34, 0.36, 0.38, 0.40,
0.42, 0.44, 0.46, 0.48, 0.50, 0.52, 0.54, 0.56, 0.58, 0.60, 0.62, 0.64,
0.66])

angle_rad = np.array([0.000000, 0.000040, 0.000322, 0.001085, 0.002573,
0.005025, 0.008683, 0.013789, 0.020583, 0.029307, 0.040201, 0.053508,
0.069467, 0.088322, 0.110312, 0.135678, 0.164663, 0.197508, 0.234452,
0.275739, 0.321608, 0.372302, 0.428060, 0.489126, 0.555739, 0.628141,
0.706573, 0.791276, 0.882492, 0.980462, 1.085427, 1.197628, 1.317307,
1.444704])

def polynomial_func(t, a, b, c):
    return a * t^2 + b * t + c

params, _ = curve_fit(polynomial_func, time, angle_rad)
a, b, c = params
print(f" Blade angle (rad)) = {a:.4f} × t^2 + {b:.4f} × t + {c:.4f}")
```

Figure 9. Program code of polynomial approximation.

After running the program code, a mathematical function was obtained that approximates the dependence of the angle of rotation of the blades on the time the motor is turned on according to the data from Table 1:

$$\varphi(t) = at^2 + bt + c. \quad (19)$$

Here t represents the time the motor is turned on, and the values of the coefficients ($a = 293.6769$, $b = +81.0778$, $c = 4.9338$) are determined by approximation.

These techniques enable the creation of a dynamic model that can predict the behavior of the control system and provide valuable insights into optimizing wind turbine efficiency. The code in (Figure 8) demonstrates the practical implementation of these techniques used in the automatic approach.

5. Results

Computer simulation was carried out in the TIA Portal 14 software environment, where various values of the control coefficients were studied. The results obtained make it possible to choose the optimal control parameters.

In accordance with Equation (19), for further development of a control system based on this equation, it was decided to use feedback to maintain a given angle of rotation of the blades of a vertically axial wind generator. Let φ_{desired} be the desired angle of rotation of the blades and φ_{actual} be the current angle of rotation as measured by the sensor. The rotation angle error will be determined by the difference between the desired and the current rotation angle [42]:

$$\text{error} = \varphi_{\text{desired}} - \varphi_{\text{actual}}. \quad (20)$$

This error will be used further to calculate the control signal that will affect the operation of the motor. One approach to control the system is to use a PI (proportional-integral) controller. The formula for calculating the control signal in the PI controller is as follows [43]:

$$u(t) = k_p \times \text{error}(t) + k_i \int \text{error}(t) dt, \quad (21)$$

where $u(t)$ is the control signal applied to the motor; k_p is the coefficient of proportionality (proportional coefficient); k_i is the integration coefficient (integral coefficient); $\text{error}(t)$ is the rotation angle error at time t ; $\int \text{error}(t) dt$ is the integral of the rotation angle error from the initial time to t .

Determining the optimal coefficients (k_p and k_i) for a PI controller, as demonstrated in the research, can be accomplished through numerical simulations and experiments.

This offers a relatively straightforward path to tuning the controller to achieve the desired system performance. The integral component of a PI controller ($k_i \int \text{error}(t) dt$) accounts for accumulated errors and ensures the elimination of steady-state errors, which can be crucial in systems requiring precise control.

The values of k_p and k_i can be adjusted experimentally to achieve the required performance of the control system. Thus, using this PI controller formula and replacing φ_{desired} with the angle of rotation obtained from the approximation function $\varphi(t)$, we can calculate the control signal $u(t)$ for the motor at each time t .

As a result, the control system will look like this:

$$u(t) = k_p \times (\varphi_{\text{desired}} - \varphi_{\text{actual}}) + k_i \int (\varphi_{\text{desired}} - \varphi_{\text{actual}}) dt, \quad (22)$$

where φ_{desired} is the calculated according to Equation (19), φ_{actual} is the measured angle of rotation of the blades at time t .

A test was carried out and the coefficients were determined for a wind turbine blade control system. This model includes equations for the gear transmission system and the blade angle control system. To test the dynamic model, computer simulation was carried out using the TIA Portal software environment. The software allowed the exploration of various control coefficients. A proportional-integral (PI) controller was used to control the wind turbine blades. The PI controller calculates the control signal ($u(t)$) based on the error between the desired angle (φ_{desired}) and the actual angle (φ_{actual}) measured by the sensor. The proportional gain (k_p) and integral gain (k_i) were adjusted experimentally during the simulation to achieve the required control system performance to minimize the error between the desired and actual blade angles and ensure stable operation.

The general scheme of the control system for the position of the wind turbine blades is shown in Figure 10.

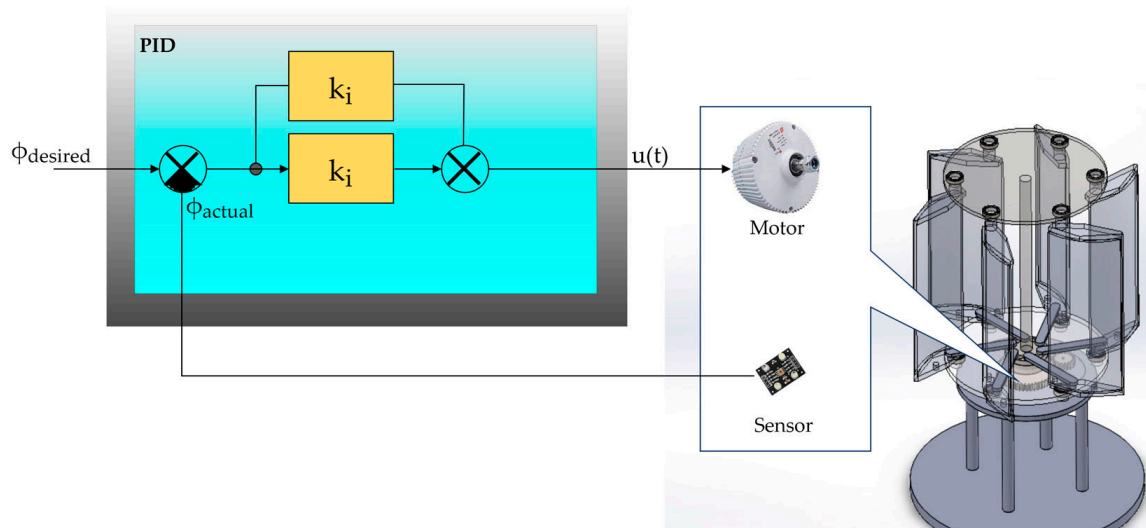


Figure 10. Scheme of the control system for the position of the vertical-axis wind turbine blades.

Criteria for selecting odds: Proportional gain (k_p): A value of 0.5 was selected. The choice of k_p affects the proportional component of the control and affects the speed of response of the control system to errors. The ratio was chosen to balance the control characteristics without causing excessive fluctuations. Integral factor (k_i): A value of 50 was selected. The coefficient (k_i) affects the integral component of the control, which helps eliminate steady-state errors. These coefficients and equations will be used in a real control system to maintain the desired blade angle in a real wind turbine.

To determine the control coefficients, computer simulation was carried out in the TIA portal software 14 environment. In this study, 40 was chosen as the desired value of the

blade installation angle. This value depends on the wind speed, on the dimensions of the wind turbine, and on the characteristics of climatic conditions, which are not considered within the framework of this article. The simulation results are shown in Figures 11–13. The proportionality coefficient was chosen as 0.5 since, with it, the P component does not exceed the desired value SP_INT; when the integration time was changed, the following results were obtained, which are shown in Table 2. At small values of T_i , the control signal at the beginning of the transient process has a pronounced oscillatory character (Figure 11). In the analysis of the results in Figure 11, an important observation emerges: the transient process, represented by the characteristic S-shaped curve, concludes at approximately 7 s. This finding underscores the inherent necessity for developing a control system capable of effecting the required change in the wind generator blade pitch angle within a 7-s window. It is particularly noteworthy that this study was exclusively focused on the maximum steady wind speed. Clearly, at lower wind speeds, this time window will expand. Consequently, the developed control system proves itself capable of effectively responding and ensuring a 90-degree change in blade position, even in the face of sudden wind gusts at or near maximum speed.

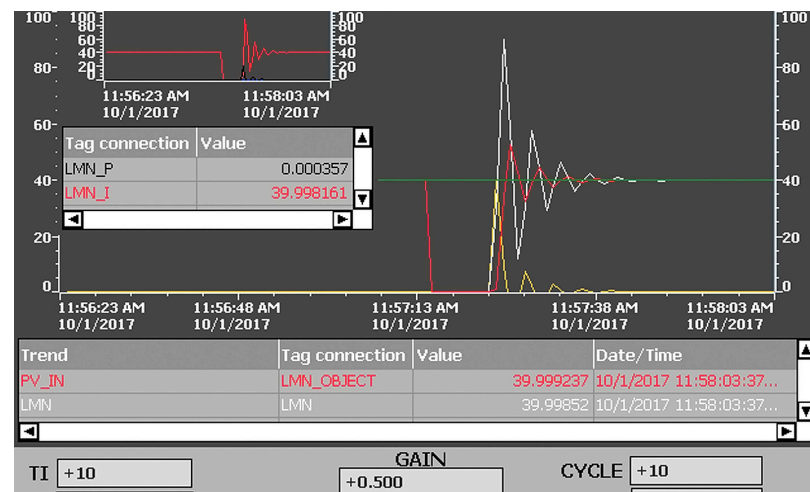


Figure 11. PI controller at $T_i = 10$ ms: green line—setpoint; red line—control action of the I-component; yellow line—setting signal; white line—resulting control action (output control signal).

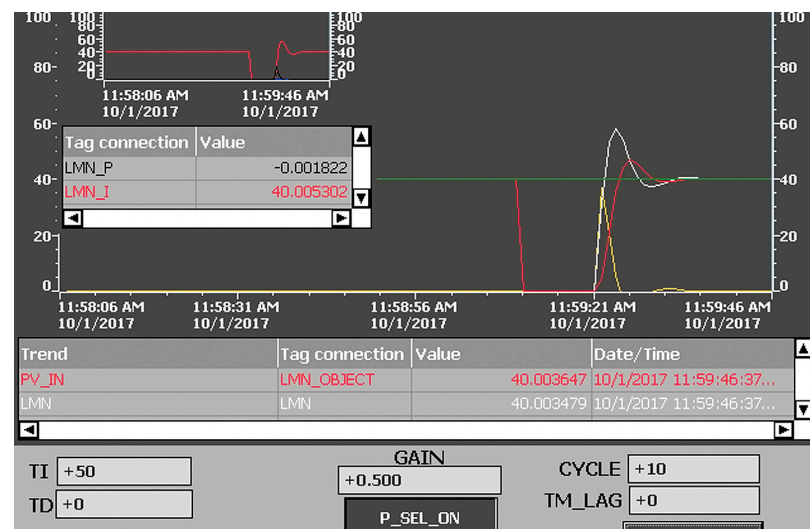


Figure 12. PI controller at $T_i = 50$ ms: green line—setpoint; red line—control action of the I-component; yellow line—setting signal; white line—resulting control action (output control signal).

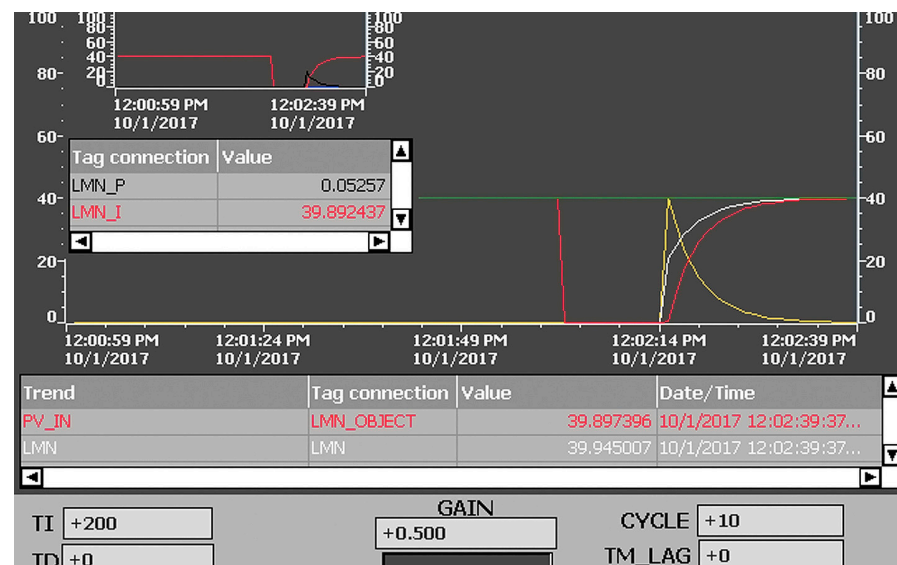


Figure 13. PI controller at $T_i = 200$ ms: green line—setpoint; red line—control action of the I-component; yellow line—setting signal; white line—resulting control action (output control signal).

Table 2. Study of the PI controller.

Ti (ms)	Peak Value LMN	Transient Time (s)	Character
10	93	≈30	oscillatory
50	59	≈17	oscillatory
100	45	≈13	oscillatory
200	40	≈25	exponential
500	40	≈65	exponential

The component completely covers the P component, so it has a much larger value. With a further increase in the integration time, the initial fluctuations decrease (Figure 12), resulting in the transient process taking on an exponential form after the influence of the P component ceases. However, this longer integration time prolongs the overall duration of the transient process (Figure 13).

Thus, during the simulation carried out in Figures 11–13, the following values were chosen:

$$\varphi_{\text{desired}} = 0.1;$$

$$\varphi_{\text{actual}} = 0;$$

$$K_p = 0.5;$$

$$K_i = 50.$$

Equation (22) would then look like this:

$$u(t) = 0.5(0.1 - 0) + 50 \int (0.1 - 0) dt. \quad (23)$$

Integrating, we get:

$$u(t) = 0.5(0.1 - 0) + 50 \times 0.1t \quad (24)$$

Simplifying, we get the control signal:

$$u(t) = 0.05 + 5t. \quad (25)$$

The desired value of the blade installation angle was set at 40, depending on wind speed, wind turbine dimensions, and climatic conditions. The simulation results in Figures 11–13 illustrated the behavior of the control system using a proportional-integral (PI) controller with specific coefficients. For the given coefficients ($K_p = 0.5$ and $K_i = 50$),

the control signal ($u(t)$) was derived using Equation (25). The influence of the integration time (T_i) on the transient process and the wind generator's operating modes were examined. A smaller T_i led to pronounced oscillatory behavior, while a moderate T_i reduced initial fluctuations, and a larger T_i resulted in an exponential transient process.

6. Discussion

The experiment used a laboratory model of a wind turbine created using a 3D printer (Figure 14). The model was exposed to an airflow at 25 m per second from a stationary state. Using the software and hardware complex "Retom-51", a series of measurements of the output voltage of the wind generator depending on time was carried out to obtain the acceleration characteristic. This dependence provides important information about the dynamics of the wind generator and its response to changes in the airflow. The results of these experiments are presented in Table 3.



Figure 14. Implemented laboratory 3D model of a wind turbine.

Table 3. Initial data.

t, s	0	0.001	0.002	0.003	0.004
y (voltage)	0	0.342	1.101	2.008	2.914

By inserting the corresponding y -values into the system of equations:

$$\begin{cases} B_3y_1 + B_2y_2 + B_1y_3 + y_4 = 0 \\ B_3y_2 + B_2y_3 + B_1y_4 + y_5 = 0. \\ B_3y_3 + B_2y_4 + B_1y_5 + y_6 = 0 \end{cases} \quad (26)$$

Equation obtained:

$$\begin{cases} 1.101B_1 + 0.342B_2 + 0B_3 + 2.008 = 0 \\ 2.008B_1 + 1.101B_2 + 0.342B_3 + 2.914 = 0. \\ 2.914B_1 + 2.008B_2 + 1.101B_3 + 3.747 = 0 \end{cases}$$

where $B_1 = -2.4218$; $B_2 = 1.9255$; $B_3 = -0.5037$. Obtain:

$$\lambda^3 - 2.4218 \times \lambda^2 + 1.9255 \times \lambda - 0.5037 = 0.$$

Given the fact that one of the roots is equal to 1, we divide the resulting equation by $(\lambda - 1)$, which leads to the following result:

$$\lambda^3 - 2.4218\lambda^2 + 1.9255\lambda - 0.5037;$$

$$\lambda - 1;$$

$$\lambda^3 - \lambda^2;$$

$$-1.4218\lambda^2 + 1.9255\lambda - 0.5037;$$

$$\lambda^2 - 1.4218\lambda + 0.5037;$$

$$-1.4218\lambda^2 + 1.4218\lambda.$$

$$0.5037\lambda - 0.5037$$

$$0.5037\lambda - 0.5037$$

$$0.$$

Next, the roots of the resulting equation after division are determined:

$$\lambda_1 = 0.67 = p;$$

$$\lambda_2 = 0.7518 = q.$$

It can be calculate T_1 and T_2 by:

$$T_1 = \frac{-\Delta t}{\ln(p)} = 2.497;$$

$$T_2 = \frac{-\Delta t}{\ln(q)} = 3.505.$$

Substituting the values:

$$y_i = k_i \left(1 + \frac{T_1}{T_2 - T_1} \times e^{\frac{-t}{T_1}} + \frac{T_2}{T_1 - T_2} \times e^{\frac{-t}{T_2}} \right), \quad (27)$$

where $k = \sum \frac{k_i}{n} = 7.501$.

In that case, the transfer function takes the following form:

$$W(p) = \frac{7.501}{(2.497p + 1)(3.505p + 1)}.$$

The acceleration characteristics of the obtained transfer functions are illustrated in Figure 15.

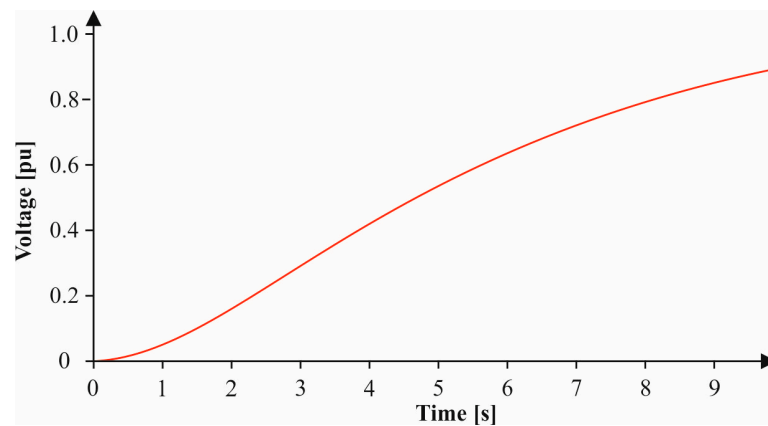


Figure 15. Acceleration characteristic.

Thus, the laboratory model of the wind turbine was identified by analyzing the static and acceleration characteristics. From the obtained experimental data, the type of connection was determined, which corresponds to the inertial connection of the second order. As a result of the analysis of the acceleration characteristics of the wind generator, it was determined that the installation time required to achieve a steady rotation speed is 7 s. This time reflects the dynamic properties of the wind turbine and its ability to achieve stable operation after changing external influences. Based on the calculations and analysis of the control system, a program was developed that allows you to adjust the position of the wind turbine blades. The program for regulating the position of the wind turbine blades includes performing calculations, determining the optimal parameters, and setting up the control system (Figure 16). The implementation of this program allows you to control the wind turbine, fine-tune the position of the blades, and maintain optimal system operation.

```
import time
def calculate_desired_angle(time):
    # Function to calculate the desired angle of blade rotation
    return 90.0 * (time / 7.0)
def pi_controller(current_angle, desired_angle, kp, ki, prev_error,
integral):
    # PI controller for controlling the blade position
    error = desired_angle - current_angle
    integral += error
    output = kp * error + ki * integral
    return output, error, integral
def control_system():
    # PI controller parameters
    kp = 1.0
    ki = 0.1
    # Initial values
    current_angle = 0.0
    integral = 0.0
    prev_error = 0.0
    start_time = time.time()
    while True:
        current_time = time.time() - start_time
        if current_time <= 7.0:
            # Calculate the desired angle of blade rotation
            desired_angle = calculate_desired_angle(current_time)
            # Use the PI controller to control the blade position
            control_signal, prev_error, integral =
pi_controller(current_angle, desired_angle, kp, ki, prev_error,
integral)
            # Update the current blade position (simulate the
operation of the wind turbine)
            current_angle += control_signal
            # Output the current time and blade angle
            print("Time: {:.2f}s, Angle: {:.2f}
degrees".format(current_time, current_angle))
        else:
            break
        # Delay to simulate real-time operation
        time.sleep(0.5)
    # Run the control system
    control_system()
```

Figure 16. Program code for regulating the position of the wind turbine blades.

As can be seen in Figure 17, the developed control program reaches the maximum closing of the wind generator blades within 7 s. This time corresponds to the acceleration period of the wind turbine from its stationary state to operation at a speed of 25 m per second. Thus, thanks to the control program, the wind generator can quickly and efficiently adapt to changes in wind speed.

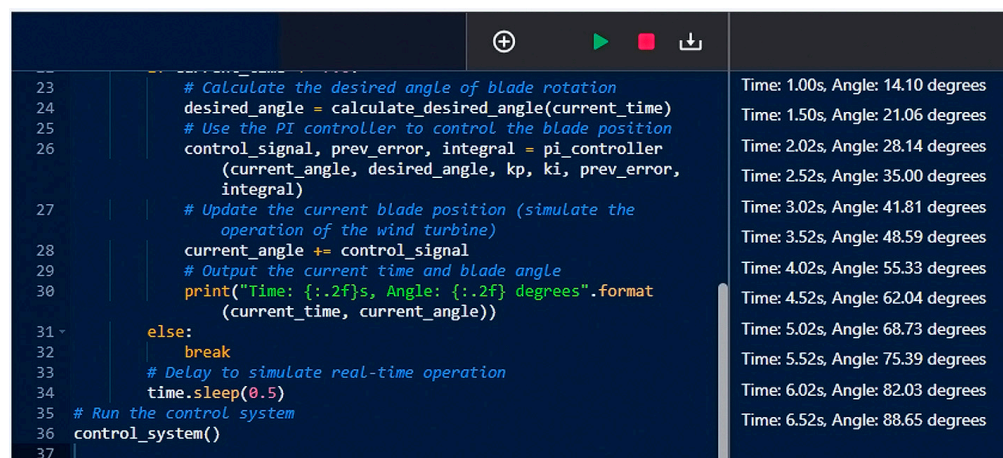


Figure 17. Operation of the control program.

In this scenario, the wind turbine operates under weak wind conditions, not exceeding 25 m per second, and functions normally without the control system's interference. However, when the wind speed reaches 25 m per second, a PI-based control system comes into play to achieve a desired blade angle of 90 degrees within 7 s. The control law, given as $u(t) = 0.05 + 5t$, adjusts the position of the blades to control the speed of rotation of the wind generator. The PI controller uses the current and desired rotation angles, while the integral component compensates for errors to achieve the target angle.

7. Conclusions

In conclusion, this article presents an integrated approach to the development and research of a control system for the position of the blades of a vertical-axis wind turbine. Using the automatic approach, a dynamic model was built to predict the behavior of the system at various values of the engine speed. This model made it possible to analyze the dependence of the blade position on the rotation of the engine and to determine the optimal parameters of the mathematical model of the control system. The main goal of the study was to create a mathematical model that would further optimize the wind turbine blade position control system depending on the wind speed. The PI controller-based system proved to be effective in achieving the desired 90-degree turn angle within 7 s when the wind speed reached 25 m per second. The presented 3D model of the control system has become a modeling and analysis tool that allows a deeper understanding of the behavior of the system in various operating conditions. Using the Runge–Kutta method and polynomial approximation, the virtual model accurately represented the complex relationship between motor activation time and blade pitch angle, which facilitated informed decisions about optimal control strategies to maximize wind turbine efficiency. In addition, the study explored the use of a pi controller to maintain the desired blade angle. Computer simulation in the TIA portal software environment revealed the optimal control parameters that provide a stable and efficient transient process. The developed program for adjusting the position of the wind turbine blades has demonstrated effective control and fine-tuning capabilities, helping to maintain the optimal operation of the system. A laboratory experiment with a 3D-printed model of a wind turbine confirmed the dynamic properties of a wind turbine and its ability to provide stable operation in response to changes in airflow. The settling time required for a stable rotational speed was

determined to be 7 s, which highlights the dynamic performance of the wind turbine. The settling time and dynamic performance of a vertical-axis wind turbine are critical factors that impact its ability to capture energy effectively, ensure safety and reliability, integrate with the grid, and maximize its overall output. Investing in control systems and designs that minimize settling time can lead to more successful and productive vertical-axis wind turbine installations in real-world conditions. The obtained result can be implemented in practice for optimizing the operation of wind turbine blade position control systems, thereby increasing the overall efficiency and reliability of wind energy use. Continuous improvement in wind turbine control technology is critical as the world moves towards a greener and cleaner energy future. The wind turbine control technology and system optimization continue to evolve, driven by the need for cleaner and more sustainable energy sources. Ongoing research and development efforts in these areas are crucial to unlocking the full potential of wind energy and addressing the global challenges of climate change and energy security.

Author Contributions: Conceptualization, I.S., T.I. and A.F.; methodology, A.F., T.I., I.S. and G.Y.; software, A.F., T.I. and I.S.; validation, I.S., T.I., A.F. and G.Y.; formal analysis, I.S., T.I., A.F. and G.Y.; investigation, I.S., T.I. and A.F.; resources, A.F. and T.I.; writing—original draft preparation, I.S., T.I., A.F. and G.Y.; writing—review and editing, I.S., T.I. and A.F.; visualization, I.S., T.I. and A.F.; contributed to the interpretation of the results, A.F., T.I., I.S. and G.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: This article is based on research conducted within the framework of the project “Development of a control and monitoring system for a multi-blade vertical-axis wind generator” of the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No: AP19679162).

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

References

1. Hassan, M.; Guedes Soares, C. Dynamic Analysis of a Novel Installation Method of Floating Spar Wind Turbines. *J. Mar. Sci. Eng.* **2023**, *11*, 1373. [CrossRef]
2. Shchur, I.; Klymko, V.; Xie, S.; Schmidt, D. Design Features and Numerical Investigation of Counter-Rotating VAWT with Co-Axial Rotors Displaced from Each Other along the Axis of Rotation. *Energies* **2023**, *16*, 4493. [CrossRef]
3. Daskalaki, D.; Fantidis, J.; Kogias, P. Feasibility of Small Wind Turbine via Net Metering in Greek Islands. *J. CIEES* **2021**, *1*, 23–28. [CrossRef]
4. Yao, Y.; Wang, G.; Fan, J. WT-YOLOX: An Efficient Detection Algorithm for Wind Turbine Blade Damage Based on YOLOX. *Energies* **2023**, *16*, 3776. [CrossRef]
5. Roga, S.; Bardhan, S.; Kumar, Y.; Dubey, S.K. Recent technology and challenges of wind energy generation: A review. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102239. [CrossRef]
6. Jard, T.; Snaiki, R. Real-Time Repositioning of Floating Wind Turbines Using Model Predictive Control for Position and Power Regulation. *Wind* **2023**, *3*, 131–150. [CrossRef]
7. Fazylova, A.; Tultayev, B.; Iliev, T.; Stoyanov, I.; Beloev, I. Development of a Control Unit for the Angle of Attack of a Vertically Axial Wind Turbine. *Energies* **2023**, *16*, 5202. [CrossRef]
8. Shah, K.A.; Meng, F.; Li, Y.; Nagamune, R.; Zhou, Y.; Ren, Z.; Jiang, Z. A Synthesis of Feasible Control Methods for Floating Offshore Wind Turbine System Dynamics. *Renew. Sustain. Energy Rev.* **2021**, *151*, 111525. [CrossRef]
9. Dunne, D. Wind and Solar Supplied More of the EU's Electricity than Any Other Power Source for the First Time Ever in 2022, New Analysis Finds. Available online: <https://www.carbonbrief.org/wind-and-solar-were-eus-top-electricity-source-in-2022-for-first-time-ever/h> (accessed on 4 June 2023).
10. Hu, C.; Deng, J.; Liu, C.; Luo, S.; Li, X.; Lu, H. Achieving an Optimal Decision for the Joint Planning of Renewable Power Supply and Energy Storage for Offshore Oil–Gas Platforms. *Appl. Sci.* **2023**, *13*, 8833. [CrossRef]
11. Beier, D.; Schnepf, A.; Van Steel, S.; Ye, N.; Ong, M.C. Fatigue Analysis of Inter-Array Power Cables between Two Floating Offshore Wind Turbines Including a Simplified Method to Estimate Stress Factors. *J. Mar. Sci. Eng.* **2023**, *11*, 1254. [CrossRef]
12. Zhang, J.H.; Lien, F.-S.; Yee, E. Investigations of Vertical-Axis Wind-Turbine Group Synergy Using an Actuator Line Model. *Energies* **2022**, *15*, 6211. [CrossRef]

13. Genç, C.; Sakalli, A.; Stoyanov, I.; Iliev, T.; Mihaylov, G.; Beloev, I. Development of Wind Energy and the Installed Wind Power Plants in Turkey. *E3S Web Conf.* **2020**, *207*, 02013. [CrossRef]
14. Rivas, C.E.; Malo, G.D.; Minchala, L.I.; Probst, O. Self-Optimizing Control System to Maximize Power Extraction and Minimize Loads on the Blades of a Wind Turbine. *Machines* **2023**, *11*, 601. [CrossRef]
15. Zhou, M.; Wang, M.; Li, J.; Li, G. Multi-area generation-reserve joint dispatch approach considering wind power cross-regional accommodation. *CSEE J. Power Energy Syst.* **2017**, *3*, 74–83. [CrossRef]
16. Genç, C.; Sakalli, A.; Stoyanov, I.; Iliev, T.; Mihaylov, G.; Beloev, I. Wind Energy Potential for Electricity Production in Hatay, Turkey. *E3S Web Conf.* **2020**, *207*, 02014. [CrossRef]
17. Al-Faruk, A.; Sharifian, A. Flow Field and Performance Study of Vertical Axis Savonius Type SST Wind Turbine. *Energy Procedia* **2017**, *110*, 235–242. [CrossRef]
18. Caponetto, R.; Fortuna, L.; Porto, D. Parameter tuning of a non integer order PID controller. In Proceedings of the Fifteenth International Symposium on Mathematical Theory of Networks and Systems, South Bend, IN, USA, 12–16 August 2002. Available online: <https://mtns.math.nd.edu/papers/7434.pdf> (accessed on 12 September 2023).
19. Fortuna, L.; Muscato, G. A roll stabilization system for a monohull ship: Modeling, identification, and adaptive control. *IEEE Trans. Control Syst. Technol.* **1996**, *4*, 18–28. [CrossRef]
20. Lam, H.F.; Peng, H.Y. Measurements of the wake characteristics of co- and counter-rotating twin H-rotor vertical axis wind turbines. *Energy* **2017**, *131*, 13–26. [CrossRef]
21. Li, X.; Qian, J.; Tian, D.; Zeng, Y.; Cao, F.; Li, L.; Zhang, G. Maximum Power Tracking Control of Wind Turbines Based on a New Prescribed Performance Function. *Energies* **2023**, *16*, 4022. [CrossRef]
22. Cheng, M.; Zhu, Y. The state of the art of wind energy conversion systems and technologies: A review. *Energy Convers. Manag.* **2014**, *88*, 332–347. [CrossRef]
23. Chowdhury, A.M.; Akimoto, H.; Hara, Y. Comparative CFD analysis of vertical axis wind turbine in upright and tilted configuration. *Renew. Energy* **2016**, *85*, 327–337. [CrossRef]
24. Knopper, L.D.; Ollson, C.A. Health effects and wind turbines: A review of the literature. *Environ. Health* **2011**, *10*, 78. [CrossRef]
25. Márquez, F.; Tobias, A.; Pérez, J.; Papaelias, M. Condition monitoring of wind turbines: Techniques and methods. *Renew. Energy* **2012**, *46*, 169–178. [CrossRef]
26. Belmili, H.; Cheikh, R.; Smail, T.; Seddaoui, N.; Biara, R.W. Study, design and manufacturing of hybrid vertical axis Savonius wind turbine for urban architecture. *Energy Procedia* **2017**, *136*, 330–333. [CrossRef]
27. Abdullah, M.A.; Yatim, A.; Tan, C.W. A study of maximum power point tracking algorithms for wind energy system. In Proceedings of the 2011 IEEE Conference on Clean Energy and Technology (CET), Kuala Lumpur, Malaysia, 27–29 June 2011; pp. 321–326. [CrossRef]
28. Musial, W.; Beiter, P.; Spitsen, P.; Nunemaker, J.; Gevorgian, V.; Cooperman, A.; Hammond, R.; Shields, M. 2019 Offshore Wind Technology Data Update; NREL/TP-5000-77411; National Renewable Energy Laboratory (NREL): Golden, CO, USA, 2020. Available online: <https://www.nrel.gov/docs/fy21osti/77411.pdf> (accessed on 4 July 2023).
29. Łukaniszyn, M.; Baron, B.; Kolańska-Płuska, J.; Majka, Ł. Inrush Current Reduction Strategy for a Three-Phase Dy Transformer Based on Pre-Magnetization of the Columns and Controlled Switching. *Energies* **2023**, *16*, 5238. [CrossRef]
30. Li, F.; Wang, D.; Liu, D.; Yang, S.; Sun, K.; Liu, Z.; Yu, H.; Qin, J. A Comprehensive Review on Energy Storage System Optimal Planning and Benefit Evaluation Methods in Smart Grids. *Sustainability* **2023**, *15*, 9584. [CrossRef]
31. Partovi-Mehr, N.; Branlard, E.; Song, M.; Moaveni, B.; Hines, E.M.; Robertson, A. Sensitivity Analysis of Modal Parameters of a Jacket Offshore Wind Turbine to Operational Conditions. *J. Mar. Sci. Eng.* **2023**, *11*, 1524. [CrossRef]
32. Zalkind, D.; Ananda, K.; Chetan, M.; Martin, D.; Bay, C.; Johnson, K.; Loth, E.; Griffith, T.; Selig, S.; Pao, L. System-level design studies for large rotors. *Wind Energy Sci.* **2019**, *4*, 595–618. [CrossRef]
33. Bortolotti, P.; Bottasso, C.L.; Croce, A. Combined preliminary–detailed design of wind turbines. *Wind Energy Sci.* **2016**, *1*, 71–88. [CrossRef]
34. Abbas, N.; Zalkind, D.; Pao, L.; Wright, A. A reference open-source controller for fixed and floating offshore wind turbines. *Wind Energy Sci. Discuss.* **2021**, *7*, 53–73. [CrossRef]
35. Fazylova, A.; Iliev, T. A New Type of Wind Generator Blades Mechanical Automatic Control System. In Proceedings of the 2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Veliko Tarnovo, Bulgaria, 24–26 November 2022; pp. 1–5. [CrossRef]
36. Alassi, A.; Ahmed, K.H.; Egea-Alvarez, A.; Foote, C. Transformer Inrush Current Mitigation Techniques for Grid-Forming Inverters Dominated Grids. *IEEE Trans. Power Deliv.* **2023**, *38*, 1610–1620. [CrossRef]
37. Nurgizat, Y.; Ayazbay, A.-A.; Galayko, D.; Balbayev, G.; Alipbayev, K. Low-Cost Orientation Determination System for CubeSat Based Solely on Solar and Magnetic Sensors. *Sensors* **2023**, *23*, 6388. [CrossRef]
38. Rodríguez, A.G.; González, A.; Payán, M. Estimating wind turbines mechanical constants. *Renew. Energy Power Qual. J.* **2007**, *1*, 697–704. [CrossRef]
39. Kanev, S. Extreme turbulence control for wind turbines. *Wind Eng.* **2017**, *41*, 353–366. Available online: <https://www.jstor.org/stable/90013672> (accessed on 15 June 2023). [CrossRef]
40. Zalkind, D.S.; Dall’Anese, E.; Pao, L.Y. Automatic controller tuning using a zeroth-order optimization algorithm. *Wind Energy Sci.* **2020**, *5*, 1579–1600. [CrossRef]

41. Knudsen, T.; Bak, T.; Soltani, M. Prediction models for wind speed at turbine locations in a wind farm. *Wind Energy* **2011**, *14*, 877–894. [[CrossRef](#)]
42. Soltani, M.N.; Knudsen, T.; Svenstrup, M.; Wisniewski, R.; Brath, R.; Ortega, R.; Johnson, K. Estimation of rotor effective wind speed: A comparison. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1155–1167. [[CrossRef](#)]
43. Jukes, T.N. Smart control of a horizontal axis wind turbine using dielectric barrier discharge plasma actuators. *Renew. Energy* **2015**, *80*, 644–654. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.