Autonomous drone for wind turbine blade maintenance

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Robotics and Autonomous Systems
April 2026
Nevil Sandaradura

Supervisors:
Adjunct Prof. Hashem Haghbayan
Prof. Juha Plosila

UNIVERSITY OF TURKU
Department of Computing

Nevil Sandaradura: Autonomous drone for wind turbine blade maintenance

Master of Science (Tech) Thesis, 51 p.
Robotics and Autonomous Systems
April 2026

The rapid growth of wind energy as a renewable energy source has shown the need for safer, faster, and more cost-effective solutions for wind turbine maintenance. It's essential to inspect turbine blades regularly to detect defects such as cracks and erosion, which can reduce the turbine's efficiency, increase its operational costs, or even produce safety risks if left unaddressed. The thesis proposes an autonomous drone-based solution for inspecting wind turbine blades in operation, to replace old inspection methods that are costly and time-consuming. The key challenge that this thesis tries to solve is the synchronization of the drone's movement with the rotating wind turbine blades, allowing the drone to recognize the defects in the turbine blades and to collect images or other sensor data of potential defects. Precise tracking of a moving blade point requires advanced control of the drone and careful consideration of the environmental conditions. To explore this, proof-of-concept simulation setups were developed in MATLAB Simulink, using a PID-based controller design to control the drone. Different controller designs were tested and compared to evaluate their effect on the drone's capability to follow the blade point. Tracking performance was evaluated using root mean square error (RMSE) and standard deviation (SD). The results show that synchronization between the drone and the moving wind turbine blade can be achieved under simplified simulation conditions. The results indicate that simulation design and control configuration play a more significant role in tracking accuracy than the environmental conditions alone. These findings provide a foundation for further research toward more realistic autonomous wind turbine blade inspection systems that are capable of inspecting real wind turbines, offering the potential to reduce their maintenance costs and increase their reliability.

Keywords: autonomous drone, wind turbine blade inspection, UAV tracking, real-time simulation

Contents

List of Figures

List of Tables

List of acronyms

6-DOF Six degrees of freedom

CNN Convolutional Neural Network

MPC Model Predictive Control

PID Proportional–Integral–Derivative controller

RMSE Root Mean Square Error

SD Standard Deviation

UAV Unmanned Aerial Vehicle

1 Introduction

Nowadays, we use more and more renewable energy. According to the International Energy Agency, renewable energy production has increased in recent years and is likely to continue growing under projected conditions. [1] One emerging renewable energy source is wind energy generated by wind turbines [2], [3]. Using more and more different kinds of solutions to produce renewable energy, it's likely that new kinds and bigger scales of problems from the machines will be encountered [4]. Therefore, it's necessary to develop new methods to address these problems and to make renewable energy production more sustainable and reliable. One way to achieve this goal is to make the maintenance of production machines easier and faster [4].

This study will focus on wind turbines and on how to make their inspection easier, faster, and cheaper using autonomous drones. The autonomous drone's purpose in the inspection is to detect surface-level damage, such as cracks, using a sensor, such as a camera [5]. Inspecting wind turbine blades requires synchronizing the drone's motion with the blades' motion so the drone can obtain a clear view of the blades' surface and collect reliable sensor data for analyzing defects [6]. This thesis, which includes a simulation built in MATLAB Simulink, was developed to assess the accuracy with which an autonomous drone can be synchronized with a wind turbine blade's motion. In chapter 1.1, the research questions and the goals of this study are described, while chapters 1.2 and 1.4 tell more about the overall structure.

1.1 Research questions

- **RQ1**: How accurate is a drone movement when synced with the movement of a wind turbine blade in different environments?
- **RQ2**: What are the properties that most affect the autonomous drone movement accuracy?
- **RQ3**: How does the movement accuracy affect the blade inspection?
- **RQ4**: How strongly does the drone simulation setup affect the accuracy of the wind turbine blade tracking?

1.2 Problem statement

Numerous studies and solutions for wind turbine blade inspection using autonomous drones already exist, but there are still some problems depending on the implementation [7]. Some implementations use LiDAR to generate 3D models of wind turbine blades. These kinds of solutions are clear and provide substantial data, but they're expensive, primarily because of the cost of the LiDAR sensor. With the help of LiDAR, obstacle avoidance and defect recognition are simpler than with a solution that uses only a camera. [8]

This study doesn't focus on defect recognition using a camera or other sensor, but rather on determining whether an accurate enough solution can be developed for tracking a wind turbine blade. One of the challenges of using a camera to inspect the wind turbine's surface is ensuring accurate inspection while the turbine is spinning and operating [9]. The turbine could be stopped, but that would be a waste of money and energy production [4]. The solution is to determine whether it's possible to synchronize a drone with the motion of the wind turbine blades, and then perform blade inspection using a camera and machine learning algorithms to detect potential

defects on the turbine surface.

The main goal of this study is to develop a simulation of a drone and a wind turbine and assess how well the drone can track a specific point on a turbine blade. Solution accuracy is the focus of the study and therefore, the simulation results will be analyzed in detail. This study also addresses other topics closely related to blade inspection, including limitations and existing solutions.

1.3 Research methods

Choosing the research methods for this thesis was quite trivial. To measure tracking accuracy and demonstrate the proof of concept for the autonomous drone inspecting moving wind turbine blades, a simulation of the concept was developed. In addition to this empirical study, this study also reviews prior research papers about the topic and analyzes them. For the simulation, MATLAB Simulink was used, as it provides a suitable environment for modeling, simulation, and analysis of dynamic, realistic systems. Matlab provides comprehensive materials and tutorials for using Simulink, in addition to the existing tools and blocks suitable for this kind of project.

1.4 Structure of the thesis

In the next chapter 2, prior research is reviewed, and the current state of the topic is summarized. Subsequently, in the 4 chapter, the implementation of the simulation setups is examined at a lower level. Chapter 5 covers the results from the simulation, while Chapter 7 shows the conclusion of the results and the simulation. Afterward, there is a discussion and a comparison of the conclusion, simulation, and results in Chapter 6. Lastly, Chapter 8 presents suggestions for future research on this topic.

1.5 Usage of AI and tools

In the process of writing this thesis, some technical tools have been used. In this section, information on the tools and their use is presented. The tool Grammarly was used to correct the text's grammar and suggest alternative words to improve readability. During the writing of this thesis, generative artificial intelligence tools were used to analyze and summarize studies and other reference materials, and to assist with the formation of some of the sentences. All the text is produced by the author of this thesis, and the correctness of the information has been checked.

2 Earlier studies

In this chapter, other studies on autonomous drones inspecting wind turbine blade defects will be analyzed to provide a clearer view of the current state of research on this topic [7]. Overall, the reviewed studies aim to achieve the same result as this study: identifying a reliable solution for autonomous wind turbine inspection. Several additional studies were reviewed to provide a more comprehensive overview of the current state of the topic.

The keywords and search queries used to identify related studies are presented in Table 2.1. The queries were used in IEEE and Google Scholar. The search was conducted from fall 2024 to spring 2026. IEEE was chosen for its domain-specific publications in engineering, and Google Scholar for its broader coverage. Papers were selected based on their titles and abstracts, and on their relevance to wind turbine inspection in general, and especially to drone-based autonomous wind turbine inspection. Some studies were identified through snowballing, whereby relevant studies were identified by exploring references and related studies from the studies identified [10].

2.1 Autonomous blade inspection

As is well known, wind energy and wind turbines have been a growing source of energy production for years [3]. At the same time, it has become clear that wind turbines require extensive maintenance and regular inspections to ensure safe and

Table 2.1: Initial search query

Query
"wind turbine blade inspection" OR "wind turbine inspection" OR "wind turbine blade defects" OR "blade inspection"
AND
"UAV" OR "drone" OR "unmanned aerial vehicle"
AND
"autonomous" OR "trajectory tracking" OR "control"
AND (optional; used to find simulation-related references)
"simulation" OR "modeling"

efficient operation [4]. To accelerate and improve turbine maintenance and inspection, particularly of turbine blades, several studies have investigated alternative approaches [11], [12]. In this section, some of them are analyzed and compared to provide a clearer overview of the current state of the research.

2.1.1 Levels of autonomy

It's important to identify the different levels of autonomy and to understand what a fully autonomous system means in the context of autonomous defect detection in wind turbine blades. When discussing levels of autonomy, there are various ways to classify them. Regardless of classification, the range typically spans non-autonomous to fully autonomous systems, with semi-autonomous systems between them [13], [14]. The Society of Automotive Engineers (SAE) has 5 levels of classification for on-road vehicles, where level 0 system is not autonomous at all, and levels 4 and 5 are fully autonomous, where the difference between level 4 and 5 is that a level 4 system is limited to a specific set of working scenarios, while level 5 can handle any scenario. The SAE classification is designed to categorize autonomous on-road vehicles but could also be applied to other use cases, including across various vehicle types, making it particularly suitable. [13] In this thesis, the term "autonomous"

is closest to level 4 of the SAE classification unless otherwise specified. Also, the Air Force Research Lab (AFRL) presented levels of UAV autonomy. AFRL's UAV autonomy levels range from 0 to 10, where level 0 denotes a remotely controlled aerial vehicle, whereas level 10 denotes a fully autonomous UAV that requires little guidance. [14] According to these classifications of the level of autonomy, the level of autonomy discussed in the thesis is as follows, in the context of autonomous wind turbine blade defect detection with a drone:

The drone is capable of the entire wind turbine blade inspection process without human intervention. This includes the drone's ability to perform autonomous takeoff and landing, navigate to the wind turbine, plan and execute the flight to the turbine, detect the blades and capture sensor data from them, and detect surface defects on the turbine blades. At this level, the drone has obstacle avoidance and adapts to the environmental conditions, including wind. The drone is limited to specific inspection scenarios, including environmental conditions, turbine types, and other factors. Outside of these boundaries, human intervention is required, as well as in the maintenance of the drone.

2.1.2 Current state of autonomous wind turbine inspection

The recent review of the studies on the topic from Heo and Na performed 2025 reviews studies related to drone-based technologies for wind turbine blade inspection [7]. The paper covers topics ranging from drone types to path planning and the various sensing techniques employed. The study reports that autonomous blade inspection is a promising alternative to manual inspections, potentially reducing manual inspections and improving the safety and reliability of turbines and the inspection process [7], [15]. The authors state that current drone technology is equipped with real-time data-processing systems, which eliminate or reduce the need for unsafe, slow methods of inspection, such as rope access or cranes that were previously nec-

essary. This is especially important in remote and offshore wind turbine farms [16]. Modern drones also have advanced navigation and detection systems that enable rapid inspection by optimizing flight routes and employing various sensing technologies. Optimized blade inspection can reduce the risk of wind turbine downtime and may even prevent the need to shut down the turbine for inspection. [7]

The study also highlights the challenges posed by autonomous wind turbine blade inspection, including battery limitations and weather effects [17], [18]. According to this study, the future of autonomous blade inspection will likely be enhanced by the use of more advanced AI models and sensor fusion [19].

Another recent study suggests that defect inspection can be significantly improved using the YOLO-Wind computer vision framework. The framework is designed for UAVs and, therefore, performs better on such devices. By using YOLO-Wind, it's possible to detect even small cracks and dirt on the wind turbine blades, even in their early stages. This level of accuracy is significant and will reduce maintenance costs and improve the reliability of defect detection. [20]

Similar results to those obtained with YOLO-Wind have been achieved with AI-based visual inspection systems that detect and classify defects. With these kinds of models, it's possible to detect and classify different types of defects with high accuracy. [15] These models are also designed to collect data on defects, in addition to merely detecting them [21].

There is also data on the various hardware solutions for blade inspection. Most solutions employ cameras, including event cameras, LiDAR, and other sensors. It has been shown that combining thermal and RGB image data yields better defect detection performance [22]. These sensors, combined with an efficient CNN-based detection model, perform better on systems with severely limited computational resources, such as drones [21]. Additionally, lower energy consumption allows the drone's operational time to be longer [22].

In this section, recent updates on autonomous wind turbine blade inspection are covered. However, it should be noted that there are numerous studies on the topic, and it remains actively researched.

2.2 Limitations

There are some limitations that very likely affect the drone's performance while inspecting a moving wind turbine blade. The most important limitations from this study's point of view are related to the wind turbine inspection itself or the drone's movement synchronization with the wind turbine blade. Some difficulties that affect the wind turbine inspection arise from environmental change, such as varying weather conditions [23]. One of the greatest environmental difficulties is the wind, since wind turbines are often placed in areas where it is windy [24]. This is one reason why the drone controller should be fast enough to react to sudden changes that are caused by the wind or other factors. The wind could either move the drone to distract its synchronization with the wind turbine blade, or it could move the wind turbine blade suddenly, causing the same outcome. [25] Other limitations related, for example, to the computational constraints, data transmission, and power source of the drone are excluded from this study.

2.2.1 Environmental challenges

Modern drones perform well across various environments and can withstand challenging weather conditions. However, for some drones, this can introduce issues during defect detection. [23], [26] Wind turbines can pose challenges for drones if they are not stopped for inspection [7]. To operate under adverse weather conditions, the drone controller must be designed to keep the drone stationary while analyzing the wind turbine blades that are moving [27]. With advanced path planning, it's

possible to reduce environmental factors that disrupt the drone's operation by selecting the optimal path for the environment, for example, by avoiding high-wind areas [23].

2.2.2 Controller related limitations

As mentioned earlier, the drone should be able to synchronize its movement with the wind turbine and remain stationary so the inspection of the blade can be performed. One of the most important factors related to this is the drone's controller. The drone controller should react quickly to disturbances during inspection, and even without the disturbances, the drone should be able to follow quickly moving wind turbine blades if the wind turbine remains operational during the inspection [23], [28]. In case the controller response time is too slow, it's possible that the drone loses the desired position, while its movement should be synchronized with the wind turbine blade.

A good performing drone controller should maintain a precise relative position to the moving blade, and if this cannot be achieved, it can result in tracking errors that accumulate over time [29]. The faster the wind turbine blade is rotating, the more difficult it is for the drone to keep up with the blade's position. On top of that, the quickly rotating blades can generate turbulence and aerodynamic disturbances that the drone should be able to react to [23]. The controllers tuned for stability may react too slowly to the disturbances or the rotating blade's position, but on the other hand, the controllers that are tuned for high responsiveness may become unstable [30]. Unstable controllers could increase errors instead of correcting them by causing oscillatory motion or unstable flight behavior [31].

3 Methodology

This chapter presents the methodology used in the study, concentrating on how the drone was tested. In this chapter 3.1 section describes what kind of controller structure was used, what inputs and outputs it had, and more about the type of drone used for testing. While section 3.2 explains how the tracking of the wind turbine with a drone was designed.

3.1 Drone

The specifications of the drone used in this study resemble the DJI Flip quadcopter displayed in figure 3.1, which is a relatively small commercial drone and weighs under 249 g according to the manufacturer [32]. The weight of the drone used for testing is 200 g, which is slightly less than the DJI Flip drone. The DJI flip drone could be very likely used in real-life testing for wind turbine inspection as it has 31 minutes of flight time and a max distance of 14 km, and on top of these properties, it has a high quality built in camera with a 2 GB internal storage and a long range video transmitter that would allow video processing from a remote location with an external computer in a real-time [32]. For a real wind turbine inspection operation, the DJI Flip might not be the optimal choice, since it might be hard to attach external sensors to it, and the longer flight time would be beneficial in wind turbine inspection, especially in the testing phase, while different kinds of problems could occur [33].

Figure 3.1: DJI Flip drone [32]

The following subsections describe the drone controller design in more detail. Section 3.1.1 presents the drone's controller structure and the design choices selected. Section 3.1.1.1 presents what inputs were given to the controller and why, while the section 3.1.1.2 describes what data the controller outputs.

3.1.1 Controller Structure

The drone controller should have a fast response time, so it can keep up with the moving wind turbine blade [30]. Things affecting the controller response time are the sampling rate of the control system, PID controller tuning, and the drone model's dynamic properties, including the mass of the drone, the ratio of weight, and the thrust generated by the motors [34].

For the drone controller, I chose to use PID-type controllers to keep the simulation simple and more computationally efficient [34]. Alternative control approaches, for example, Model Predictive Control (MPC), were excluded from this study. [35] In this study, the term PID-type controller refers to any controller that includes proportional (P), integral (I), and/or derivative (D) components [36]. PID-type controllers are commonly used in drone flight control because of their robustness, simplicity, and because they are suitable for real-time implementation [34].

Two existing solutions were found that have a compatible drone controller struc-

ture that could be utilized for the wind turbine blade inspection, since the inputs and outputs of those were as needed, and these controller also based on PID-type controllers [37], [38]. So instead of developing a drone controller from scratch, these two existing solutions were used as a base for the controller design.

The first existing controller, later referred to as controller A, and its structure are described in more detail in the paper written by Ahmed et al. [37]. This controller is originally designed for a liquid-carrying plant protection UAV, and it allows the drone to follow a desired trajectory while keeping it stable. The controller is based on PID-type controllers, and it receives trajectory signals and compares them with the measured position and orientation of the drone. Feedback loops are used in the controller to correct the drone's position and orientation and guide it towards the target trajectory by generating control signals using the error between the actual values and desired values of the drone's position and orientation. [37]

The other existing controller, later referred to as controller B, is more deeply described in the paper written by Ferry [38]. The structure of this drone controller consists of three main components: the controller itself, the sensor subsystem, and the plant model. The purpose of the sensor subsystem is to measure the drone's state, such as position and orientation, and provide this data to the controller so it is able to adjust the controller inputs, while the plant model represents the drone's physical behavior. This controller comprises four PID-type control loops that stabilize the drone. These separated control loops are responsible for controlling the drone's pitch, roll, yaw, and altitude. All of those four control loops work simultaneously to first stabilize the drone and then control its position by applying thrust and changing the orientation of the drone. To achieve this, the controller outputs are converted into speed commands for each of the four motors of the drone. [38]

Both of these controller structures are based on PID-type controller feedback loops, where the error is continuously calculated by comparing the actual state and

the desired state of the drone to be able to correct the movement by adjusting the outputs so that the error is minimal and the drone maintains stable flight.

3.1.1.1 Inputs

Usually, the controller inputs consist of the reference and the feedback signals. The reference signals describe the desired motion to the controller by defining the target behavior of the system. Reference signals can include data such as the desired position, altitude, or orientation of the drone. For the wind turbine blade inspection, these reference signals define the path that the drone should follow to keep its movement synchronized with the blade. The feedback signals describe the current state of the drone and are obtained from the sensors that are responsible for calculating. These signals can be such as a drone's current position, speed, or orientation. By using the reference signals and feedback signals, it's possible to calculate the error between the actual and desired state of the drone. [36] The control error can be used by a PID-type controller to correct the state of the drone [34].

In controller A, the input reference signals are the desired XYZ position, trajectory, and velocity. The input feedback signals are the current position, velocity, and the pitch, roll, and yaw angles. [37] In controller B, the input reference signals are desired altitude, desired pitch and roll angles, and desired yaw rate, while the input feedback signals for this controller are the current altitude, pitch and roll angles, and yaw angular velocity [38]. Even though the inputs of these two controllers are different, they both allow the controller to continuously adjust the drone's motion to follow the desired path.

3.1.1.2 Outputs

The outputs of the drone controller are control signals that are used for determining the speeds of the drone's motors [30]. By adjusting the speeds of the motors, it's

possible to control the drone's motion, including its speed and orientation. Quadcopters such as DJI Flip have four motors that are used to control the orientation and the lift of the quadcopter. The control signals are calculated based on the inputs and the errors obtained from the feedback loop [36]. The control of the motor speeds of each of the four motors allows precise control of the altitude, and the pitch, roll, and yaw of the drone.

The controller A outputs total thrust, and yaw, pitch, and roll angles that are used to generate angular velocities for each of the motors of the drone, to control the drone [37]. Controller B outputs rotor angular velocities for each rotor that directly control the drone motors [38]. Both of these outputs allow controlling the drone in a way that it can accurately follow the given flight path and maintain stability during the wind turbine blade inspection.

3.2 Wind turbine

3.2.1 Inputs

For the wind turbine, there are several inputs that straightly affects to how the wind turbine operates [39]. Some of the input values are related to the energy generation of the turbine, but do not directly affect the turbine's behaviour. In the table 3.1, the most important inputs of the wind turbine that are critical when inspecting the turbine blade with a drone are presented [40].

3.2.2 Tracking

For the drone to be able to synchronize its speed and position with the wind turbine blade, it was necessary to make the drone know which blade it should synchronize its movement with. To accomplish this, x points were added to each of the wind turbine blades. In this study, the number selected was 10. Later in this study, these

Table 3.1: Main wind turbine inputs affecting turbine behaviour in the simulation

Input parameter	Unit	Value
Wind speed (v_{wind})	m/s	depending on the wind profile
Air density (ρ)	kg/m^3	1.15
Rotor radius (R)	m	35.25
Rotor blade inertia (J_r)	kg·m^2	3 963 845
Blade pitch angle (β)	deg	Variable
Pitch actuator ramp rate	deg/s	20
Cut-in wind speed	m/s	4
Rated wind speed	m/s	12
Cut-out wind speed	m/s	20
Turbine cut-in speed	rpm	11
Turbine cut-out speed	rpm	22

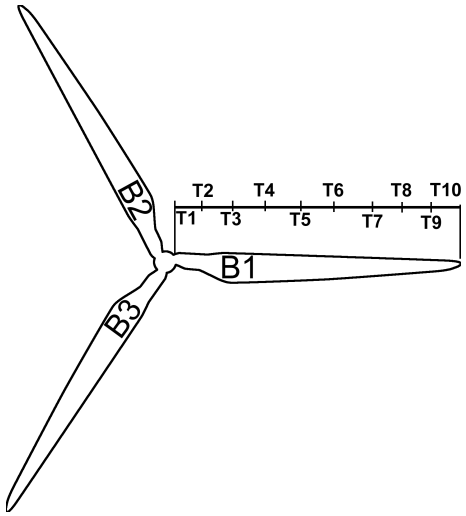

Figure 3.2: There are ten blade points T[1-10] on each blade B[1-3]

points are referred to as blade points. Each blade was configured to have 10 blade points, each of them containing information about the point's current location as shown in Figure 3.2. Ten blade points were selected because this number is sufficient to demonstrate the concept without making the simulation overly complex. [41] The blade points are evenly spaced along the blade's length.

4 Simulation setups

The simulation was designed to be as minimal as possible while attempting to replicate real-world wind turbine blade inspection using a drone [12]. The simulation setup comprised two components: the wind turbine model simulation and the drone model simulation. After creating these models, it was time to merge them and see how the system worked. This chapter presents the full details of both simulations and integrates them into a single account.

4.1 Wind turbine simulation

The wind turbine model was primarily developed by following the tutorial on MATLAB's website. In the tutorial, the wind turbine model comprises five components: supervisory control, turbine with grid and transformer, turbine pitch controller, power controller, and visualizer. [40] The created wind turbine model had the first four components implemented as subsystems, and the visualizer component was integrated directly into the top-level model.

The reason for following the tutorial provided by MATLAB for creating the wind turbine simulation was the clear explanation of the operating principle and the desire to avoid design errors. From the outset, it was evident that the simulation setup required modifications not covered in the tutorial, and therefore, it was justified to use a well-designed simulation as a baseline. Additionally, given the simulation's structure, implementing the drone simulation would be easier after drawing on infor-

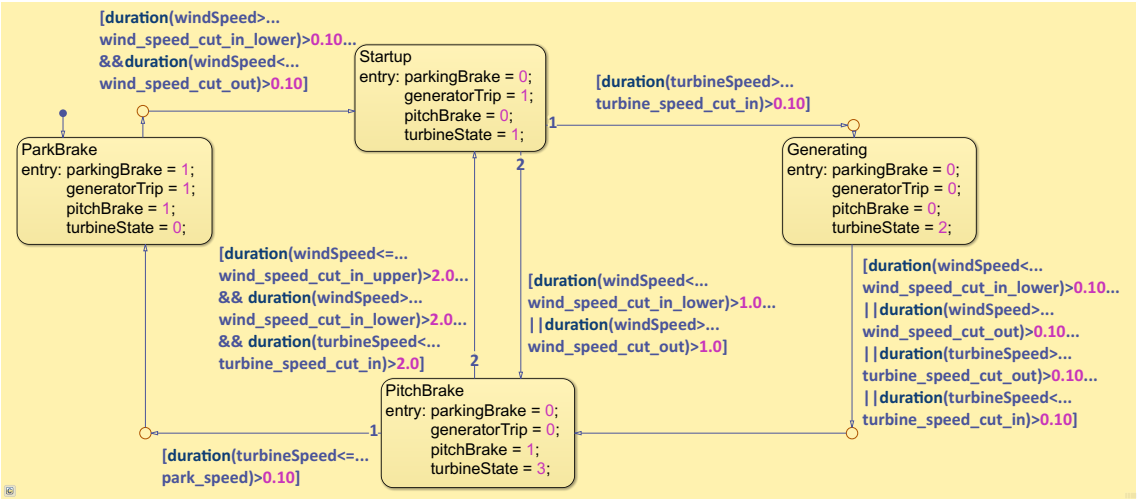

Figure 4.1: Supervisory control system state machine

mation and design practices from a tutorial that explains them. Next, more detailed information is provided on each subsystem, followed by the modifications made in the simulation.

4.1.1 Supervisory control subsystem

The supervisory control subsystem in this simulation is a relatively simple state machine with four states: park break mode, startup mode, generating mode, and pitch break mode. [40] These states represent different operational states that the turbine can have. The technical implementation of this subsystem was relatively simple, as it is a state machine with only a few states. Initially, understanding the state-transition conditions in Simulink was somewhat challenging.

It is evident in the state machine structure in Figure 4.1, which includes all states and the conditions that can change the state. Next, there is a more detailed overview of each state of the supervisory control subsystem's state machine to better understand the wind turbine's operating principles.

4.1.1.1 Park brake mode

The park brake mode is accessed only from pitch brake mode and is the simulation's entry mode. Its purpose is to stop the blades' movement completely. In this state, the turbine's hydraulic brake is activated, and the blades are positioned to slow the turbine's rotation. In park brake mode, the wind turbine speed is below the safe operating limit and therefore, the generator doesn't produce energy. [40]

4.1.1.2 Startup mode

Startup mode is enabled either after park brake mode or after pitch brake mode, depending on the situation. In this state, the wind turbine's hydraulic brake is released, and the blades are positioned to maximize turbine speed as quickly as possible. In this state, the generator is deactivated, and therefore, the turbine doesn't produce energy. [40]

4.1.1.3 Generating mode

This state is activated after the start-up state, when the blade rotation speed is sufficiently high. When the desired speed is achieved, the blades are positioned to maximize energy production. The generator is activated in this state and connected to the transformer. [40]

4.1.1.4 Pitch brake mode

In this state, the turbine's rotational speed is reduced by positioning the blades appropriately. The generator remains on, and the turbine produces electricity from the ongoing rotation of the blades. This state is accessed either from startup mode or from generating mode, depending on wind speed and turbine rotational speed. [40]

4.1.2 Turbine with grid and transformer

This subsystem contains a wind turbine and a grid with a transformer. The grid and transformer are included for demonstration only and are simplified, as they are not necessary for the simulation's objective. This wind turbine module consists of three main components: the rotor hub and the nacelle.

The rotating blades are probably the most well-known part of a wind turbine. The rotor hub is a built-in block in Simulink. The blades are designed to capture wind as efficiently as possible, and their angle can be adjusted, which is typically done by the wind turbine itself, based on its speed and the wind speed [39]. The Simulink wind turbine block takes wind velocity and pitch angle as inputs and outputs thrust. In this simulation, the thrust is then sent to the nacelle and the velocity sensor.

In general, the nacelle is a complex component of the wind turbine, as it contains components such as cooling, electronics, and the drivetrain. In some wind turbines, the power conversion system may also be included. [42] Though in this simulation, it contains brakes, a gear train, a generator, and the wind turbine's velocity sensor. In this simulation, the system is simplified to improve speed and computational performance. In this simulation setup, the nacelle receives the turbine's thrust and parking-brake state as inputs and outputs three energy-phase components, denoted A, B, and C. The grid and transform components are very simple, as they do not affect the results and are for demonstration purposes in this study. It takes the energy-phase components as inputs and consists of a transformer, a grid, and parasitic conductance. The parasitic conductance is added only to increase the simulation speed [40].

4.1.3 Turbine pitch controller

This subsystem controls the turbine pitch angle. It takes the turbine state, wind speed, and generator power as inputs and outputs the pitch angle. The purpose of this subsystem is to set the pitch angle based on the current state, as obtained from the turbine state machine described in 4.1.1. [39], [40]

4.1.4 Power controller

The power controller subsystem models power demand and sets the optimal torque for the wind turbine by simulating the generator input speed. It takes wind speed, generator trip value, and pitch brake as inputs and outputs active power demand and the power reference. This is an important part of the setup, as it simulates how power demand and torque optimization behave in this simulation. [39], [40]

4.1.5 Data visualizer

The data visualizer is not a subsystem, but it only combines Simulink scope blocks and multiplexers to view and log signals, enabling the signal data to be viewed in Simulink's Simulation Data Inspector [40]. Although it's the simplest part of the system, it's probably the most important because, with the data, it's possible to observe what is happening in the simulation and whether the signal values are proper [43]. After all, it's difficult to tell how the system works without seeing the data. The most important data are displayed in the scope block as a graph. As in Matlab's tutorial, it was configured to visualize the following properties of the system: all states of the supervisory control system, pitch angle, generator power output, and wind speed. Other data that wasn't needed as often could be viewed in the Simulation Data Inspector. [40]

4.1.6 Modifications

This section covers the modifications to the simulation made to make it suitable for this study. For the wind turbine simulation, it was necessary to add the blade points to the wind turbine blades so that the movement and position of each blade could be determined, providing the drone with a reference for the location it needs to track. To implement these blade points for each blade, Simulink's blocks, mostly joints, solids, rigid transforms, and transform sensors were used. The solids present the blades and are connected to each other through the central piece using weld joints. The middle piece is then rotated and connected to the rotor hub velocity sensor. Each solid has 10 rigid transforms and transform sensors to represent the blade-tracking points. Each rigid transform has its own XYZ coordinates, so it's simple to read the values of those using the transform sensors. With this setup, the location data can be saved to a MATLAB workspace variable and used in real time within the simulation, which is important when implementing the drone simulation. Using Multibody blocks like these also enabled visualization of the turbine rotor motion in MATLAB's Mechanics Explorer, although this was unnecessary given the comprehensive data logging. [40]

The modification described above was the most important for this wind turbine simulation from the perspective of the drone simulation. Otherwise, the simulation presented in Matlab's tutorial served its purpose well. Some minor modifications were made to simplify the simulation and facilitate future modifications. One of these steps was to replace Simulink's Goto and From blocks with Import and Output blocks. This made it easier to see how the data actually moves between different parts of the system [40].

4.2 Drone simulation setups

Several simulation approaches were evaluated to meet the drone system's control requirements, and these approaches were refined to improve the accuracy of the drone controllers. The main limitation of the drone controllers tested was that none were fast enough to maintain the drone's position accurately at the selected blade point on all axes.

The initial simulation setup was as follows: each simulation ran for 60 seconds, providing sufficient coverage of performance and accuracy. On top of that, a longer time frame would have caused issues since the machine used to run the simulations took a lot of time, even hours, to run some of the simulation setups, and the simulation run time had to be the same for every setup, so the data of the different setups could be compared. On the other hand, too short a simulation time wouldn't yield sufficient data for analysis and could also lead to inaccurate results.

For each tested simulation setup, one blade point was selected for the drone to track. The simulation setup outputs RMSE for each axis (X, Y, Z) to indicate the accuracy of tracking in that axis [44]. At the top of the RMSE value, a graph is also created to visualize the difference between the drone's position and the blade point's position on each axis, making it easier to understand how the solution performed. Each simulation ran for three different blade points, B1_T1, B1_T5, and B1_T10, to see if and how the blade point location affects the tracking accuracy.

In the following subsections, the tested drone simulation setups and their differences are discussed. In the next chapter 5, the results of those simulation setups are analyzed and compared.

4.2.1 Simulation setup A: 6-DOF quadcopter based setups

These simulation setups are strongly based on the 6-DOF Quadcopter simulation originally developed by Kong Chun-Wei. The original purpose of developing this

drone was to simulate the dynamics of a liquid-carrying UAV. The quadcopter comprises the main body frame, brushless DC motors, and a PID-type velocity controller [36]. The structure and operational principles of the system are more detailed in the paper Ahmed et al. [37].

The first step in this solution was to preprocess the wind turbine simulation data to ensure that the quadcopter simulation could interpret it correctly. The data were formatted as a MATLAB matrix, with each row containing a timestamp and the X, Y, and Z coordinates of each blade point. This preprocessed dataset was sampled at 1 Hz. After the simulation was run for the first time without any modifications, the result showed a significant amount of error. Therefore, it was clear that modifications were needed to reduce the error.

The first step in the simulation was to retune the PID-type controllers for X, Y, Z, and the yaw controller modules of the drone. Initially, the PID-type controllers were tuned using MATLAB's PID tuner tool to obtain preliminary settings. Subsequently, these were manually fine-tuned through trial and error to improve accuracy. Even after these changes, the results were insufficiently accurate. [34]

4.2.1.1 Simulation setup A1: Minimal modifications

Table 4.1: RMSE on each axis for different blade points, after controller modifications and PID-tuning

Error on Axis (RMSE)	B1_T1	B1_T5	B1_T10
X	163.8038	167.9352	215.0175
Y	23.2805	28.7241	14.5587
Z	47.3066	142.6118	131.6333

The next step was to make small modifications to the controllers by adding derivative (D) and/or integral (I) terms to the proportional (P) controller. After several combinations and considerable PID retuning, a solution with relatively low RMSE was identified [34], [44]. The results of this solution can be seen in table 4.1

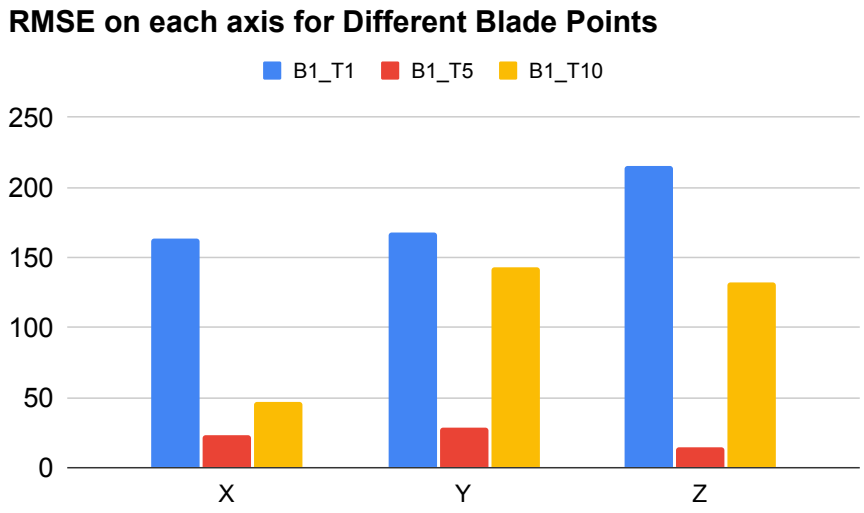

Figure 4.2: RMSE on each axis for different blade points in simulation setup A1

and in figure 4.2. For an unknown reason, after these changes, the simulation run time was much slower than before. As the simulation results were satisfactory at this point, it was clear that even greater precision was still required, because when a drone tracks defects on a wind turbine blade, precision is critical, and error should be minimal. Therefore, further improvement was pursued to improve accuracy.

4.2.1.2 Simulation setup A2: Redesigning the X and Y controller modules

The next attempt used the same quadcopter simulation as a baseline, but this time by redesigning the drone's controller modules. Instead of using multiple P-controllers, PID-type controllers were used in the X and Y controller modules. The Z and yaw controllers were already highly simplified and did not produce significant errors relative to the X and Y, so they were retained as is. In the X and Y controller modules, different input values were also tested for the controllers, instead of the original ones, which were based on previous designs but did not provide sufficiently accurate results for this setup [34]. After a significant number of designs, the

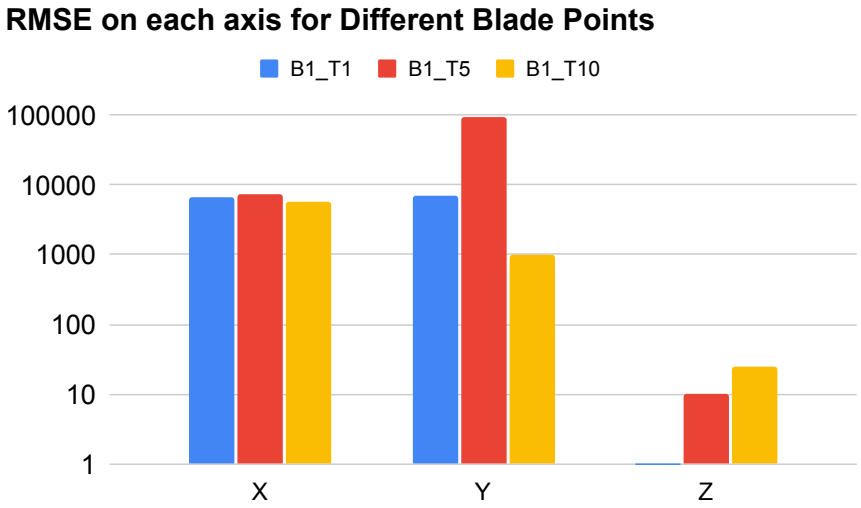

Figure 4.3: RMSE on each axis for different blade points in simulation setup A2

best possible solution was identified again. For each change, retuning the PID-type controllers was required and was done primarily using MATLAB's PID Tuner tool, but when the design appeared promising, the tuning was fine-tuned by hand [34]. Additionally, modifications to the Z and yaw controller modules were attempted, but these further degraded the results and were therefore left unchanged.

Table 4.2: RMSE for Different Blade Points after redesigning X and Y controller modules.

Error on Axis (RMSE)	B1_T1	B1_T5	B1_T10
X	6.608×10^3	7.5201×10^3	5.7654×10^3
Y	6.9725×10^3	9.6242×10^4	1.0213×10^3
Z	0.9395	10.1259	24.9110

After these modifications, which appeared promising, the system was tested with various blade points and wind profiles, and some performed better than others. As shown in table 4.2 and figure 4.3, the RMSE increased significantly for the X and Y axes, and this was the case for all tests conducted with this setup. The only positive change relative to the previous solution was that the RMSE on the Z-axis decreased substantially and became tolerable. [44] Additionally, the simulation runtime was

brought back close to that of the original baseline setup, making it computationally less demanding to run the simulation multiple times for different blade points and wind profiles. The results of the simulations with this setup and the one described in section 4.2.1.1 were not sufficiently accurate, so a different approach was adopted, as described in section 4.2.2.

4.2.2 Simulation setup B: Real-time drone

This drone simulation is completely different than the ones described above. For this setup, the quadcopter plant model and control system design are based on the paper Ferry, where it's described in more detail [38]. Despite its high level of detail and complexity, the model required several modifications to be suitable for use with wind turbine simulation data.

Instead of saving the data to a dataset, as in the drone simulation setups described earlier, this approach directly fed wind turbine blade data from the wind turbine simulation to the drone simulation in real time. With this approach, data preprocessing was unnecessary, enabling the drone to respond more dynamically to wind turbine blade motion and better reflect real-world operating conditions. The plan was to run both simulations simultaneously, with the drone responding as needed in real time.

To feed the wind turbine's blade data to the drone, Simulink's inport block was added to the model which inputs multiplexed data of a blade point which is then demultiplexed in the quadcopter model using Simulink's demux block, which outputs X, Y and Z locations of the wind turbine's blade point and fed those into the drone's guidance module, that had inputs for desired XYZ positions and actual XYZ postion coming from the plant model in a form of feedback loop. From the guidance module's output, roll, pitch, and yaw were computed, which were then used to determine rotational speeds. [38] After having the rotational speeds, those were fed

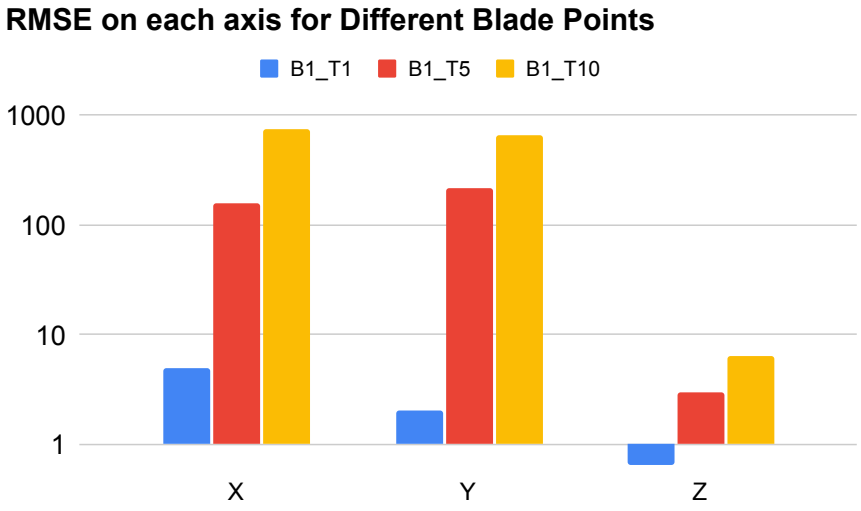

Figure 4.4: RMSE on each axis for different blade points in simulation setup B

to the plant model, whose operational logic is more deeply explained in Ferry [38]. The plant model outputs the drone's location data, which we used to calculate the error using the drone's location and the blade point's actual location [44].

Overall, the simulation's basic working logic was straightforward, yet it provided an advantage over the other two setups by enabling real-time interaction between the drone and the wind turbine simulation. Also, this real-time drone was more realistic and, therefore, provided better testing conditions for the drone's control system.

Table 4.3: RMSE for Different Blade Points of real-time drone simulation setup.

Error on Axis (RMSE)	B1_T1	B1_T5	B1_T10
X	5.0225	155.7701	745.2005
Y	2.0118	216.3959	658.6624
Z	0.6407	2.9464	6.3341

After testing this solution numerous times and refining it, improving the results seemed impossible. As shown in table 4.3 and figure 4.4, the RMSE results for the chosen blade points weren't particularly high but were higher than expected. It can

also be seen that the closer the blade points were to the center of the wind turbine, the smaller the error was.

In the next chapter 5, a closer look at each of the simulation setups is taken, and the results of each of those setups will be compared with each other, and then it will be discussed whether any of these solutions would be accurate enough to use in a real-life wind turbine surface defects inspection.

5 Results

In the previous chapter 4.2, the results of all the drone simulation setups whose purpose was to track the specific blade point of a wind turbine blade were presented. As shown, the results from some setups were significantly inaccurate and, therefore, those setups are not very useful for real-world applications. However, the purpose of this chapter is to compare the results and see which setup is most suitable for real-world use.

5.1 Comparison of RMSE value of the results

It was clear that the simulation setup A2 presented in section 4.2.1.2 was the least accurate among the three setups: A1, A2, and B. Only the RMSE error values for the z-axis were quite accurate and even more accurate than the first setup A1 presented in section 4.2.1.1, but still a bit more inaccurate than the last simulation setup B presented in section 4.2.2. Because of the significant inaccuracy of simulation setup A2, it is excluded from the comparison of results, and therefore, simulation setups A1 and B will be compared.

5.1.1 X-axis error

The results indicate that the farther the drone is from the center of the wind turbine, the higher the RMSE. This was evident in the results for setup A1 and setup B as well. In setup B, the RMSE is significantly lower when the drone tracks the blade

point B1_T1. While in setup A1, the RMSE was 163.8083, whereas in setup B it was 5.0225. However, this wasn't the case for all blade points on which the simulation setups were tested. So the difference in RMSE for this blade point was 158.7858.

When observing the blade point B1_T5, it is observable that the RMSE values of setup A1 and setup A2 are closer to each other, with only a 12.1651 difference, where simulation setup A1's value was higher. At the blade point B1_T10, A1's RMSE value was significantly lower than setup B's. Simulation setup A1 has an RMSE of 215.0175, while setup B has a corresponding value of 745.2005, resulting in a 530.183 difference in favor of setup A1.

How should it be determined which of the simulation setups performed better for the X-axis? It is important to identify the setup that performs most consistently across all blade points, rather than the one with the lowest overall RMSE. For this reason, both the mean RMSE and the standard deviation (SD) of the RMSE value were analyzed. By analyzing both, the setup that balances accuracy and consistency most effectively could be identified. While a lower average RMSE across blade points indicates better overall accuracy, a lower SD means more reliable performance across all the blade points. The same method was used in the next chapters, where Y and Z-axis accuracies are analyzed.

Table 5.1: Comparison of the mean and standard deviation of RMSE values for the X-axis

Simulation setup	**RMSE (X)**	**SD of RMSE (X)**
A1	182.2522	28.4
B	301.9975	391.3

An analysis of the values in table 5.1 and figure 5.1 shows that simulation setup A1 has both a lower overall RMSE and SD. The SD is significantly lower, while the overall RMSE is lower, so the difference is less significant. Accordingly, setup A1 outperformed setup B on the X-axis. For setup B, the RMSE was lower than the SD, indicating substantial variability and strong outliers in the data.

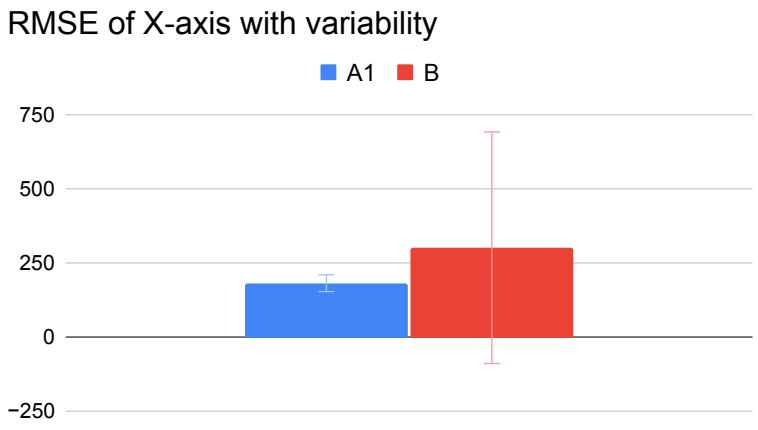

Figure 5.1: Comparison of mean RMSE on the X-axis for simulation setups A1 and B, with error bars representing the standard deviation

5.1.2 Y-axis error

The Y-axis in setup B seemed to behave quite similarly compared to its X-axis. This is not the case for setup A1, as the results showed that the highest RMSE value was from blade point B1_T5, whereas the lowest was B1_T10, and therefore, the RMSE does not increase similarly to the X-axis.

Table 5.2: Comparison of mean and standard deviation of RMSE values for the Y-axis

Simulation setup	**RMSE (Y)**	**SD of RMSE (Y)**
A1	22.1878	7.141
B	292.3567	334.49

As shown in Table 5.2 and figure 5.2, both SD and mean RMSE are substantially lower for simulation setup A1 than for setup B. Therefore, we can follow the same approach as for the X-axis and determine that setup A1 also performed better on the Y-axis. The setup B has a higher SD compared to RMSE, also in the Y-axis results.

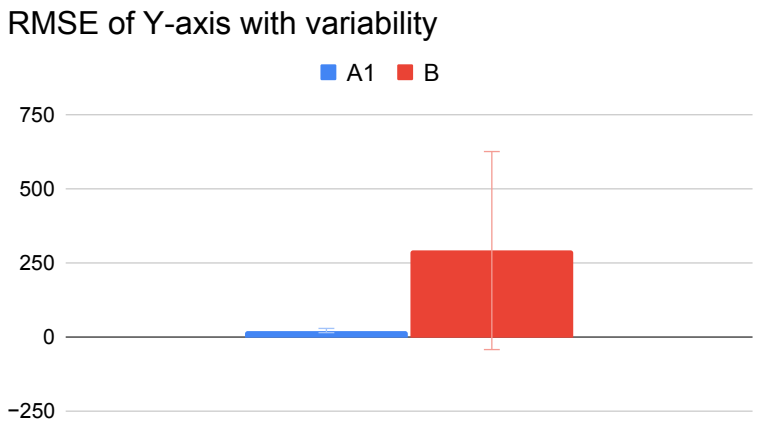

Figure 5.2: Comparison of mean RMSE on the Y-axis for simulation setups A1 and B, with error bars representing the standard deviation

5.1.3 Z-axis error

In the Z-axis of simulation setup A1, the lowest RMSE was observed at bladepoint B1_T1, the highest at B1_T5, and the intermediate value at B1_T10. In setup B, the Z-axis values increase more intuitively, so the lowest value is at bladepoint B1_T1, and the highest is B1_T10. However, when comparing the values of the two setups, it was clear that setup B had significantly lower values.

Table 5.3: Comparison of mean and standard deviation of RMSE values for the Z-axis

Simulation setup	**RMSE (Z)**	**SD of RMSE (Z)**
A1	107.1839	52.24
B	3.3071	2.863

The RMSE values for both simulation setups indicated that, along the Z-axis, setup B performed substantially better than setup A1. This can also be seen when looking at the mean RMSE and SD in the table 5.3 and in the figure 5.3. In the Z-axis, setup B performed more accurately.

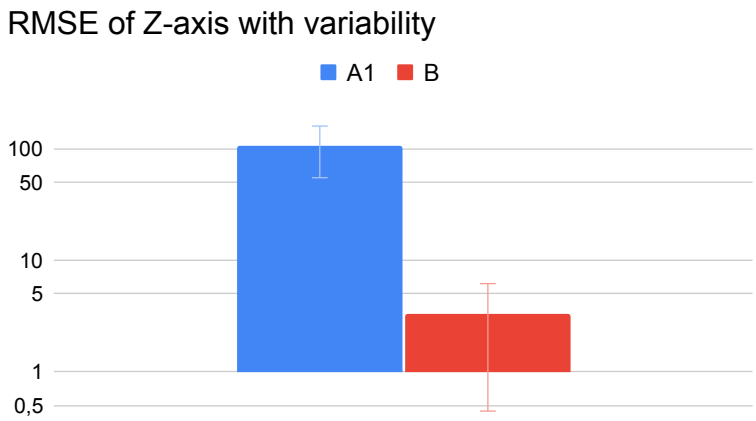

Figure 5.3: Comparison of mean RMSE on the Z-axis for simulation setups A1 and B, with error bars representing the standard deviation

5.2 Overall results

After examining the results for the XYZ axes for both simulation setup A1 and setup B, setup A1 performed better in the X and Y axes, while setup B performed better in the Z-axis. Both simulation setups have advantages, but overall, setup A1 is more stable due to smaller fluctuations in its RMSE values. Accordingly, the next step is to test the performance of setup A1 under different wind profiles and assess whether the accuracy of the results remains.

5.2.1 Deeper testing

Although setup A1 performed reliably in the simulation environment, it's difficult to predict how it would perform in real-world conditions. To obtain a clearer view, it was run with a different wind profile to analyze and compare the RMSE for each blade point on a single blade.

The first wind profile, that were used before for each setup, is a stable linear wind profile, and it is well-suited for testing the basic accuracy of this simulation setup, but for deeper tests, it is a good idea to use another wind profile so the results

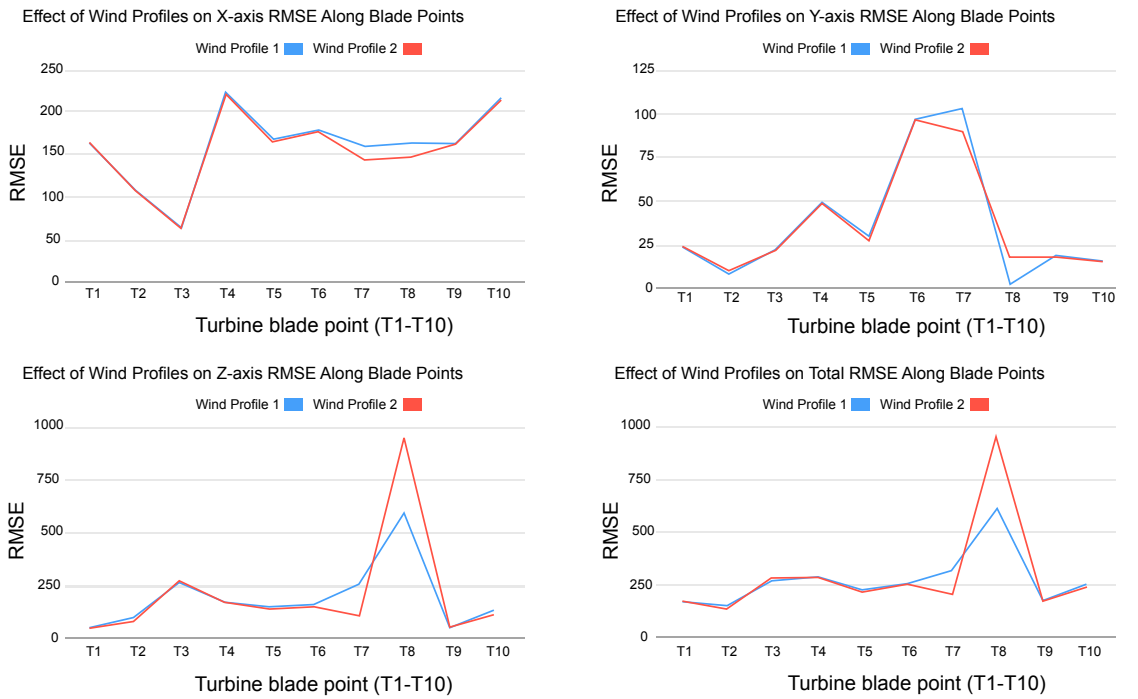

Figure 5.4: RMSE on x, y, z axes and total error across transform points, comparing the effect of different wind profiles with setup A1

can be compared and the effect of the wind on the accuracy can be tested. There may be multiple other wind profiles, but because the simulation takes a significant amount of time to run, only one other wind profile was used for testing in this study.

The second wind profile used for testing exhibited greater vertical variability and higher peak velocities than the original wind profile. The second wind profile produces more turbulence than the original. This significant difference in the wind profiles clearly demonstrated how the drone performs under varying environmental conditions. It's important to note that the wind profiles don't affect the drone itself, but they only affect the wind turbine, which rotates its blades according to wind speed. This is because the simulations are constructed in this way, and for this study, it's sufficient to gather information on how the drone can track a point on a wind turbine blade.

In the figure 5.4, there are four subplots, one for each axis, and then a comparison

of total RMSE. On the Y-axis of each graph, there is the RMSE value, and on the Z-axis, the transform point of the blade is shown. Overall, the difference is not substantial. However, closer examination of these graphs and analyses reveals a clearer picture of the accuracy and performance of setup A1 under different environmental conditions.

This testing using two different wind profiles provided insight that the drone could endure different wind conditions, which is an important trait for a drone doing wind turbine inspection operations, because weather conditions, including the wind, can't be predicted accurately enough for every region, and the drone shouldn't be dependent on specific ordinary weather conditions, since it should be operational whenever possible to get the real benefit using the autonomous drone instead of a human or a remote controlled drone.

Further testing with different wind profiles and other environmental circumstances, such as different temperatures and rain, could be tested to obtain more information how the drone would perform in different environments, but to get most of these other tests the simulation should be more complex, containing at least battery and other electronics for the drone to see how the temperature would affect to the drone's battery life and how durable the electronics of the drone would be. For real-world use, the drone should be evaluated to determine how wind affects its motion while it synchronizes its motion with a point on a wind turbine blade. In this study, the drone simulation was kept simple to assess how well the drone can synchronize its motion with a specific blade point.

While these results indicate that the simulation is functioning and is relatively accurate, they do not address how the drone would perform in real-world conditions. Even so, it provides insight into performance under different environmental conditions and, in some sense, answers **RQ1**. The wind profile affects the accuracy of this simulation, but not as much as expected. The more aggressive wind profile

is associated with slightly increased tracking error.

These results are sufficient to demonstrate that the system is overall operational, at least under specific conditions. In the next chapter, the results and other information gathered through this study will be discussed, and the remaining research questions will be addressed using these data.

6 Discussion and analysis

This chapter analyzes the results of the simulation setups presented in the chapter 4, focusing on the accuracy of the drone's motion under different environmental conditions, identifying the key factors that affect this accuracy, and evaluating the overall quality of the simulation setups.

6.1 Drone movement accuracy in different environments

The wind profile is the only changing environmental condition in these setups. In chapter 5, the most accurate drone simulation setup was tested with an additional wind profile to see how the simulation performs under different environmental conditions. It should be noted that the wind only affects the wind turbine blades, not the drone itself, because of the simplicity of the drone simulation setups. As noted in the previous chapter, different wind profiles clearly affected the drone's accuracy in tracking the blade point of the wind turbine blade [45]. For some of the blade points, the change of RMSE caused by the wind is marginal, whereas for others, the change is more noticeable.

6.2 Properties affecting tracking accuracy

Multiple factors affect the accuracy of the drone's motion when it is synchronized with a wind turbine blade point [46]. For **RQ2**, this section concentrates on the properties that are not directly related to environmental conditions but still have a notable impact on the drone's movement accuracy.

6.2.1 Design limitations of the simulation models

All the tested simulation setups were based on PID-type control architectures, which is a simple and computationally efficient approach, but it introduced several design constraints. In the 6-DOF quadcopter-based setups presented in section 4.2.1, the simplifications caused the drone to respond either too quickly or too slowly to changes in blade motion, resulting in accumulated positional tracking error. The simulation assumed perfect position and orientation measurements without latency or noise [47]. While this simplification made the system easier to configure, it likely caused an overestimation of how well the PID-type controllers would perform in real-world use.

The real-time simulation setup described in section 4.2.2 addressed some of these design flaws by directly feeding wind turbine blade data into the drone model in real time. Nevertheless, this setup also introduced synchronization timing issues and limited computational capacity. Even small timing delays between the drone and wind turbine simulations resulted in small positional drift, particularly on the X and Y axes.

6.2.2 Control configuration and tuning

In simulation setups A1 and A2, multiple PID-type controllers were tuned separately for each axis (X, Y, Z) and yaw. The tuning was first performed with MATLAB's

PID Tuner and then refined manually through trial and error, resulting in moderate accuracy in the neutral condition, but it was not robust across different blade points or wind profiles. The controller tuning required balancing of its responsiveness and stability. Increasing proportional gains led to overshooting and oscillations, while lower gains reduced responsiveness. The integral and derivative terms helped reduce steady-state error but increased simulation runtime and numerical fluctuation. Redesigning the X and Y controller modules and testing with alternative input signals in simulation setup A2 caused a significant increase in RMSE across the horizontal axes. These results suggest that the chosen PID structure had limited capability to handle the fast-rotating turbine blade point.

In the simulation setup B, the drone control setup performed relatively better due to the coupling between the guidance module and the plant model. The PID-type controllers remained sensitive to rapid position changes, particularly for blade points farther from the nacelle (B1_T10), indicating that the control loop could not fully compensate for large accelerations and that a more advanced control method would likely improve performance [29].

6.3 Effect of movement accuracy on blade inspection quality

The drone's movement accuracy indicates how reliably it can perform a wind turbine blade inspection. The goal of this section is to understand how positional accuracy would affect the potential inspection quality in a real-world scenario, based on speculations derived from the synchronization accuracy between the drone and the wind turbine blade point. It should also be noted that the simulation setups in this study allow the drone to inspect only the wind turbine blades from the front side, since the back side of the blades cannot be inspected with this setup.

6.3.1 Positional accuracy and inspection coverage

In the context of blade inspection, these variations would directly affect how consistently the drone could maintain the desired relative position with respect to the blade surface. Accuracies along the X- and Y-axes determine how closely the drone follows the blade's motion. The tracking errors on these axes indicate the drone drifting away from the blade point, resulting in inconsistent surface coverage.

The results showed that simulation setup A1 provided the most stable side-to-side blade tracking, with relatively low mean RMSE and SD, corresponding to more consistent inspection coverage in slower-moving blade regions. Simulation setup B, which uses real-time synchronization, showed greater side-to-side deviation, leading to less predictable inspection results.

6.3.2 Vertical accuracy

Vertical accuracy, represented by the Z-axis RMSE, determines how well the drone maintains a consistent inspection distance from the blade, helping to minimize issues related to image scaling and loss of focus. The results in chapter 5 showed that simulation setup B achieved far better accuracy than setup A1. In a real-world inspection environment, this would improve image stability and reduce the risk of collisions or loss of visibility. [7], [47] However, vertical stability alone is insufficient if horizontal tracking remains inaccurate.

6.3.3 Overall implications for inspection quality

Side-to-side accuracy (X–Y axes) specifies whether the drone can maintain proper coverage of the blade surface, while vertical accuracy (Z-axis) affects proximity and image stability. The results suggest that, in real-world settings, further improvements in the control system and movement synchronization would be necessary to ensure consistent inspection data quality.

6.4 Impact of simulation setup on tracking accuracy

This section answers **RQ4**. The results in chapter 5 demonstrate that the choice of simulation setup significantly influences tracking accuracy, and in several cases, the setup itself has a greater impact on overall performance than either the blade point or the wind profile.

6.4.1 Differences between the tested simulation setups

Simulation setup A1 showed the lowest variation among blade points, suggesting that its simplified yet consistent model structure supported stable tracking, while setup A2 produced the most inaccurate results of all three setups. Although both of them used the same base, with minor differences in controller design and input structure. This indicates that the tracking accuracy was highly dependent on the design of the simulation setup and the controller structure.

Setup B achieved the highest accuracy on the Z-axis but was considerably less accurate on the X and Y axes. The higher horizontal error was related to limitations in real-time synchronization. This made the performance more dependent on the timing of the simulation setup than on the controller itself [48].

6.4.2 Influence of timing and data flow

The most significant finding is that the simulation design, specifically the transfer of the blade data to the drone model, directly affected accuracy. In simulation setup A1, the drone followed preprocessed blade-point data, yielding stable results. Setup B relied on real-time data, and even small delays or fluctuations in update rates caused significant changes in the drone's sideways position. These timing differences were especially clear at blade point B1_T10, where the blade motion was fastest. Simulation setup B was clearly behind the blade point, indicating that real-time

coupling between simulation elements can degrade accuracy unless the update rate is consistent enough.

6.4.3 Sensitivity of accuracy to model design choices

The results also show that even small design decisions affect blade-point tracking performance [49]. For example, modifying the controller structure in simulation setup A2 led to a significant reduction in accuracy. This indicates that the simulation setups were not only different in performance but also in sensitivity, meaning that the controller design itself is a main factor in how well the drone can follow a blade point.

6.4.4 Overall impact of the simulation setup

Based on the results, the differences between setups caused far larger accuracy changes than environmental conditions or blade point selection. Improving the simulation design, particularly the data flow to the drone simulation and the controller behavior, would likely produce greater accuracy.

6.5 Summary

The results showed that the drone's movement accuracy varied across axes, blade points, and environmental conditions, with the fastest blade regions posing the greatest tracking challenges. It also indicated that the key factors affecting movement accuracy were not environmental conditions alone, but rather simulation setup and the controller design, particularly the sensitivity of PID-type controllers and tuning parameters. Although the simulation setups used in this study are not yet ready for real-world use, they still provide useful insights into how drone design and its various aspects influence autonomous blade tracking.

7 Conclusion

This study researched the use of an autonomous drone to track and inspect wind turbine blades, with a primary focus on blade tracking accuracy rather than defect detection, to help with the maintenance of wind turbine blades. The primary objective was to develop a proof-of-concept simulation in MATLAB Simulink to evaluate how well a drone can synchronize with a moving blade point under varying environmental conditions, controller configurations, and simulation setups. The following conclusions summarize the insights obtained from the simulation experiments, concentrating on the factors that influence drone tracking performance and their implications for potential real-world blade inspection applications.

7.1 Answers to the research questions

- **RQ1**: The results showed that a drone can follow a wind turbine blade with reasonable accuracy when movement synchronization is handled through control logic, even when the blade motion differs due to environmental changes. Tracking performance, measured by RMSE and SD, remained relatively stable across most blade points under neutral conditions and decreased only slightly when the wind profile was changed. The effect of the wind varied by blade point, with some showing minimal change in RMSE. These results indicated that the drone can maintain sufficient synchronization for tracking tasks in simple environments. However, accuracy was highly dependent on which blade

point was observed and how strongly the wind affected its trajectory. Because the drone is not directly affected by wind in the current simulation model, the results reflect best-case tracking behavior rather than fully representing real-world dynamics.

- **RQ2**: The results show that the properties that have the greatest influence on drone movement accuracy are related to the simulation and controller design. The PID-based controllers used in all setups provided a functional baseline. Still, their performance was hardly constrained by simplifications in the drone model, including the lack of actuator dynamics, aerodynamic effects, and sensor uncertainty. These design choices made the system easier to tune but also made it less realistic and limited its responsiveness, especially on fast-moving parts of the blade. The tuning process itself was highly sensitive. Small adjustments could significantly improve or degrade performance, and changes to the controller design led to significant accuracy loss in some of the setups. Overall, movement accuracy was determined more by the model's design and configuration than by external conditions, especially the central importance of controller design, tuning strategy, and realistic system modeling in future implementations.

- **RQ3**: The results show that movement accuracy had a direct impact on the quality and reliability of blade inspection. Stable tracking along the X and Y axes is particularly important, as it determines the drone's ability to follow the blade point's path and maintain consistent surface coverage. Z-axis accuracy determines how well the drone maintains a steady inspection distance, resulting in clear and consistent imagery. Simulation setup A1 showed better side-to-side tracking accuracy. It would therefore enable more consistent inspection coverage in slower blade regions. The more realistic real-time simulation setup B had better altitude stability but showed greater horizon-

tal drift, which would lead to inconsistent coverage. Temporal response also plays a critical role by delaying synchronization, leading to inspection data being captured at the wrong moment and resulting in blur or missed surface areas. Overall, the results indicated that insufficient motion accuracy directly reduces inspection quality, especially near the blade point closest to the blade tip, where motion is fastest. This means that improved tracking performance is essential for reliable real-world turbine blade inspections.

- **RQ4**: The results clearly show that the simulation setup itself has a major impact on blade point tracking accuracy, in many cases more than the selected blade point or wind profile. All three tested simulation setups showed that structural differences in control design, data processing, and timing could produce considerably different results. Simulation setup A1 had the most balanced tracking performance, while minor controller changes in A2 led to a significant drop in accuracy. The real-time setup B improved altitude stability but introduced larger horizontal deviations due to synchronization delays. These results highlighted that performance tracking is highly sensitive to the implementation of the drone model and to data transfer between system components. In practice, improving simulation design, especially real-time data flow and update rate consistency, would likely have a greater impact on tracking accuracy than environmental adjustments alone. Therefore, the simulation setup and especially the controller structure are the most critical factors determining how accurately a drone can follow a moving wind turbine blade.

7.2 Limitations of the study

Although the simulation results provide useful insights into drone synchronization with wind turbine blade points, the study also has several limitations that affect the

generalization of the results. The drone simulation setups lacked important physical effects, such as aerodynamic forces, actuator dynamics, power supply, and sensor noise, resulting in tracking performance that reflects ideal behavior rather than a fully realistic one. The drone was also not exposed to wind or other environmental disturbances, whereas the wind profile affected only the blade motion, reducing the validity of the results. Also, wind was the only environmental factor that varied, and its complexity was limited to two predefined wind profiles. The real-time simulation setup also faced update rate and synchronization constraints, which affected accuracy and may not accurately reflect how a real embedded autonomous system would behave. These limitations should be considered when analyzing the results, as they show that accuracy values may vary significantly under real-world wind turbine blade inspection conditions. These limitations also highlight several directions for further research, which are discussed in the next chapter.

Overall, this study shows that a drone can be synchronized with a wind turbine blade for blade inspection, at least under predefined simulation conditions. The simulation setups in this study offer a foundation for automating wind turbine blade inspections. While the current models contain several limitations, the results showed clear directions for improving control performance and simulation realism in future work. By improving these setups, autonomous drones could play an increasingly valuable role in safely and efficiently maintaining renewable energy infrastructure at a lower cost [4]. The next chapter discusses how these conclusions can be expanded and used as a base for further research.

8 Future research

This study investigated the accuracy with which an autonomous drone can track a moving point on a wind turbine blade in a controlled simulation environment. While the results demonstrated that synchronizing drones' motion with wind turbine blade motion is achievable and provided insights into controller behavior, the simplified simulation environment limits the direct translation of these findings to real-world use cases. To improve the simulation setup for practical use, several areas require further research. This chapter summarizes the most relevant directions for future work, focusing on improvements to more realistic simulation, controller design, sensor use, and practical validation.

8.1 Improving the simulation to better correspond to the real-world use

One of the main limitations of the current study is the simplified physical model of the drone. The simulation excluded aerodynamic forces, wind disturbances, actuator dynamics, and sensor imperfections, all of which play a significant role in real flight [45]. Future research should therefore incorporate a more realistic representation of drone behavior by modeling aerodynamic drag, thrust limitations, turbulence effects, and blade-induced airflow. Additionally, including sensor noise, latency, and drift would allow the simulation to reproduce realistic navigation un-

certainty. Combining these elements would provide a more robust understanding of how controller performance and tracking accuracy translate to real-world conditions and would allow more reliable comparisons between controller strategies.

8.2 Energy consumption and flight feasibility

This study did not account for battery limitations or energy consumption, even though these factors strongly affect real-world blade inspection operations. Future studies could include a detailed battery model in the simulation setup, allowing examination of how more aggressive maneuvers, altitude changes, and tracking precision affect the drone's flight time and energy consumption [50]. Such modeling would also help identify potential inspection durations, optimal flight paths, and energy efficiency. Evaluating endurance with realistic movement would provide a clearer picture of whether the tested controllers and trajectories are practical for full-blade inspections at operational wind farms.

8.3 More advanced controller designs

All controllers used in the simulation setups for this thesis were based on PID-type controllers. While PID controllers are simple and widely used, they offer limited capabilities and struggle with highly dynamic or nonlinear motion, such as that near the blade tip [51]. Therefore, future research should investigate the use of advanced control methods to improve accuracy and robustness. For example, MPC could anticipate future blade motion and minimize tracking errors through predictive optimization. Adaptive controllers could adjust their parameters in response to changing blade speeds and environmental conditions. Robust control approaches, such as sliding mode control (SMC), can improve stability in the presence of disturbances. Data-driven and reinforcement learning techniques are also promising,

particularly for learning more complex blade motion patterns directly from simulation [45]. Comparing these methods with PID controllers would clarify the trade-offs between computational cost, stability, and performance.

8.4 Sensors for actual blade inspection

Although this study focused exclusively on movement tracking, real inspection quality depends on the sensors carried by the drone. Future research should incorporate camera and depth-sensor models into the simulation setup to examine how movement accuracy affects image clarity, surface coverage, and defect detection. Different types of cameras offer distinct advantages. RGB cameras capture surface defects, thermal cameras reveal moisture ingress or delamination, and event cameras are particularly effective in high-speed scenarios where conventional cameras would blur images [27]. Depth imaging methods, such as stereo vision or structured light systems, can help with geometric reconstruction, whereas LiDAR provides precise 3D structural information. Combining multiple sensors through sensor fusion algorithms could improve both navigation accuracy and inspection quality. Incorporating these sensing considerations into the simulation would offer a more complete understanding of how tracking performance translates into inspection outcomes.

8.5 Better scope for the blade inspection and flight trajectory design

The current simulation tracked only a single point on the blade's front surface. However, real inspections require complete coverage of the entire blade surface, including the trailing edge, the suction side, and often also the wind turbine tower itself. Future research should explore more complex flight trajectories that support

full 3D surface coverage. This could include trajectory optimization methods that ensure complete turbine imaging, as well as tracking strategies that guide the drone along the blade rather than a single point on it. Multi-drone systems also represent an emerging approach in which multiple drones collaborate to reduce inspection time. Studying these cases would provide more realistic insights into how drones could perform comprehensive blade inspections during turbine operation [52].

8.6 Real-world tests with hardware

Simulation-based findings must be verified through real-world testing. Future experiments could begin with controlled indoor environments using motion-capture systems, enabling accurate evaluation of controller performance and sensor behavior. [28] Mock-up blades mounted on rotating test rigs could provide good enough validation under predictable conditions, while outdoor experiments with slowly moving or parked wind turbine blades would add environmental complexity and introduce real wind disturbances. Over time, these increasingly realistic tests would help determine how well the proposed methods transfer to operational turbines and whether the achieved tracking accuracy was good for reliable wind turbine blade inspection [28]. This kind of validation is necessary for evaluating the performance of the system developed through simulation.

8.7 Summary

Future research should focus on developing more realistic simulation environment, exploring more advanced control systems, including sensor systems, expanding inspection scenarios, and validating results on real hardware. These factors would increase the reliability and applicability of drone-based wind turbine inspections, bringing the technology closer to deployment in real operational environments.

References

[1] International Energy Agency (IEA), "World energy outlook 2025", International Energy Agency, Paris, Tech. Rep., 2025. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2025.

[2] F. Capizzi, A. Das, T. Dauwe, I. Moorkens, R. Juhana, and M. Tomescu, "Renewable energy in Europe", European Environment Information and Observation Network, Tech. Rep., 2019, pp. 21, 25–26. [Online]. Available: https://www.eionet.europa.eu/etcs/etc-cme/products/etc-cme-reports/renewable-energy-in-europe-2019-recent-growth-and-knock-on-effects/.

[3] Global Wind Energy Council (GWEC), "GWEC Impact Report: Powering The Future With Wind", Global Wind Energy Council, Tech. Rep., 2025. [Online]. Available: https://www.gwec.net/reports/gwec/impactreport.

[4] C. A. Walford, "Wind turbine reliability :understanding and minimizing wind turbine operation and maintenance costs.", Sandia National Laboratories, Tech. Rep., Mar. 2006. DOI: 10.2172/882048. [Online]. Available: https://www.osti.gov/biblio/882048.

[5] L. Wang and Z. Zhang, "Automatic detection of wind turbine blade surface cracks based on uav-taken images", *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7293–7303, 2017. DOI: 10.1109/TIE.2017.2682037.

[6] H. Tanrıverdi, G. Karakuş, and A. Ulukan, "Wind turbine inspection with drone: Advantages and disadvantages", *Journal of Energy Systems*, vol. 7, no. 1, pp. 57–66, 2023. DOI: 10.30521/jes.1148877.

[7] S.-J. Heo and W. S. Na, "Review of drone-based technologies for wind turbine blade inspection", *Electronics*, vol. 14, no. 2, 2025, ISSN: 2079-9292. DOI: 10.3390/electronics14020227. [Online]. Available: https://www.mdpi.com/2079-9292/14/2/227.

[8] M. Car, L. Markovic, A. Ivanovic, M. Orsag, and S. Bogdan, "Autonomous wind-turbine blade inspection using lidar-equipped unmanned aerial vehicle", *IEEE Access*, vol. 8, pp. 131 380–131 387, 2020. DOI: 10.1109/ACCESS.2020.3009738.

[9] S. Minaeian, J. Liu, and Y.-J. Son, "Effective and efficient detection of moving targets from a uav's camera", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 497–506, 2018. DOI: 10.1109/TITS.2017.2782790.

[10] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering", in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '14, London, England, United Kingdom: Association for Computing Machinery, 2014, ISBN: 9781450324762. DOI: 10.1145/2601248.2601268. [Online]. Available: https://doi.org/10.1145/2601248.2601268.

[11] M. Memari, P. Shakya, M. Shekaramiz, A. C. Seibi, and M. A. S. Masoum, "Review on the advancements in wind turbine blade inspection: Integrating drone and deep learning technologies for enhanced defect detection", *IEEE Access*, vol. 12, pp. 33 236–33 282, 2024. DOI: 10.1109/ACCESS.2024.3371493.

[12] C. Yang et al., "Bladeview: Toward automatic wind turbine inspection with unmanned aerial vehicle", *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 7530–7545, 2025. DOI: 10.1109/TASE.2024.3464640.

[13] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles", SAE International, Recommended Practice, Standard J3016_202104, Apr. 2021, Revised April 2021; originally issued January 2014. DOI: 10.4271/J3016_202104. [Online]. Available: https://doi.org/10.4271/J3016_202104.

[14] E. Sholes, "Evolution of a uav autonomy classification taxonomy", in *2007 IEEE Aerospace Conference*, 2007, pp. 1–16. DOI: 10.1109/AERO.2007.352738.

[15] C. Yang et al., "Bladeview: Toward automatic wind turbine inspection with unmanned aerial vehicle", *IEEE Transactions on Automation Science and Engineering*, vol. PP, pp. 1–16, Jan. 2024. DOI: 10.1109/TASE.2024.3464640.

[16] K. Zhang, V. Pakrashi, J. Murphy, and G. Hao, "Inspection of floating offshore wind turbines using multi-rotor unmanned aerial vehicles: Literature review and trends", *Sensors*, vol. 24, no. 3, p. 911, 2024, ISSN: 1424-8220. DOI: 10.3390/s24030911. [Online]. Available: https://www.mdpi.com/1424-8220/24/3/911.

[17] P. Garg, "Characterisation of fixed-wing versus multirotors uavs/drones", *Journal of Geomatics*, vol. 16, no. 2, pp. 152–159, Oct. 2022. DOI: 10.58825/jog.2022.16.2.44. [Online]. Available: https://onlinejog.org/index.php/journal_of_geomatics/article/view/44.

[18] D. Vohra, P. Garg, and S. Ghosh, "Problems and prospects of flying rotor drones particularly quadcopters", *Türkiye İnsansız Hava Araçları Dergisi*,

vol. 4, no. 1, pp. 1–7, 2022. DOI: 10.51534/tiha.1068613. [Online]. Available: https://izlik.org/JA85NK32MB.

[19] S. Gibb, H. M. La, T. Le, L. Nguyen, R. Schmid, and H. Pham, "Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection", *Journal of Field Robotics*, vol. 35, no. 6, pp. 988–1004, 2018. DOI: https://doi.org/10.1002/rob.21791. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21791. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21791.

[20] Z. Zhanfang and L. Tuo, "Enhancing wind turbine blade damage detection with yolo-wind", *Scientific Reports*, vol. 15, no. 1, p. 18 667, 2025, ISSN: 2045-2322. DOI: 10.1038/s41598-025-03639-8. [Online]. Available: https://doi.org/10.1038/s41598-025-03639-8.

[21] Y. Zhu and X. Liu, "A lightweight cnn for wind turbine blade defect detection based on spectrograms", *Machines*, vol. 11, no. 1, p. 99, 2023, ISSN: 2075-1702. DOI: 10.3390/machines11010099. [Online]. Available: https://www.mdpi.com/2075-1702/11/1/99.

[22] S. Svystun, O. Melnychenko, P. Radiuk, O. Savenko, A. Sachenko, and A. Lysyi, *Thermal and rgb images work better together in wind turbine damage detection*, 2024. arXiv: 2412.04114 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2412.04114.

[23] B. H. Wang, D. B. Wang, Z. A. Ali, B. T. Ting, and H. Wang, "An overview of various kinds of wind effects on unmanned aerial vehicle", *Measurement and Control*, vol. 52, no. 7-8, pp. 731–739, 2019. DOI: 10.1177/0020294019847688. eprint: https://doi.org/10.1177/0020294019847688. [Online]. Available: https://doi.org/10.1177/0020294019847688.

[24] S. Pranupa, A. Sriram, and S. Nagaraja Rao, "A review of wind farm layout optimization techniques for optimal placement of wind turbines", *International Journal of Renewable Energy Research*, vol. 13, no. 2, 2023.

[25] E. Bregu, N. Casamassima, D. Cantoni, L. Mottola, and K. Whitehouse, "Reactive control of autonomous drones", in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16, Singapore, Singapore: Association for Computing Machinery, 2016, pp. 207–219, ISBN: 9781450342698. DOI: 10.1145/2906388.2906410. [Online]. Available: https://doi.org/10.1145/2906388.2906410.

[26] K. Wang et al., "Weatherclean: An image restoration algorithm for uav-based railway inspection in adverse weather", *Sensors*, vol. 25, no. 15, p. 4799, 2025, ISSN: 1424-8220. DOI: 10.3390/s25154799. [Online]. Available: https://www.mdpi.com/1424-8220/25/15/4799.

[27] S. Zhang et al., "Uav based defect detection and fault diagnosis for static and rotating wind turbine blade: A review", *Nondestructive Testing and Evaluation*, vol. 40, no. 4, pp. 1691–1729, 2025. DOI: 10.1080/10589759.2024.2395363. eprint: https://doi.org/10.1080/10589759.2024.2395363. [Online]. Available: https://doi.org/10.1080/10589759.2024.2395363.

[28] A. Amer, M. Mehndiratta, J. le Fevre Sejersen, H. X. Pham, and E. Kayacan, *Visual tracking nonlinear model predictive control method for autonomous wind turbine inspection*, 2023. arXiv: 2310.14030 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2310.14030.

[29] K. Pereida and A. P. Schoellig, "Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions", in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7831–7837. DOI: 10.1109/IROS.2018.8594267.

[30] F. Santoso, M. A. Garratt, S. G. Anavatti, and I. Petersen, "Robust hybrid nonlinear control systems for the dynamics of a quadcopter drone", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 3059–3071, 2020. DOI: 10.1109/TSMC.2018.2836922.

[31] Y. Liu, X. Huang, Y. Zhang, and Y. Zhou, "Dynamic stability and control of a manipulating unmanned aerial vehicle", *International Journal of Aerospace Engineering*, vol. 2018, no. 1, p. 3 481 328, 2018. DOI: https://doi.org/10.1155/2018/3481328. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/3481328. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2018/3481328.

[32] DJI. "Dji flip specs". [Online]. Available: https://www.dji.com/fi/flip/specs.

[33] Z. Ameli, Y. Aremanda, W. A. Friess, and E. N. Landis, "Impact of uav hardware options on bridge inspection mission capabilities", *Drones*, vol. 6, no. 3, p. 64, 2022, ISSN: 2504-446X. DOI: 10.3390/drones6030064. [Online]. Available: https://www.mdpi.com/2504-446X/6/3/64.

[34] Q. Wang, "Selection of pid parameters for uav control system", *AIP Conference Proceedings*, vol. 3144, no. 1, p. 030 038, Jun. 2024, ISSN: 0094-243X. DOI: 10.1063/5.0218152. eprint: https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0218152/20010900/030038_1_5.0218152.pdf. [Online]. Available: https://doi.org/10.1063/5.0218152.

[35] K. S. Holkar and L. M. Waghmare, "An overview of model predictive control", *International Journal of control and automation*, vol. 3, no. 4, pp. 47–63, 2010.

[36] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Prentice Hall, 2015, ISBN: 0133496597. [Online]. Available:

http://books.google.com/books?id=WaozngEACAAJ&dq=isbn:0133496597&hl=&source=gbsapi.

[37] S. Ahmed, B. Qiu, C.-W. Kong, H. Xin, F. Ahmad, and J. Lin, "A data-driven dynamic obstacle avoidance method for liquid-carrying plant protection uavs", *Agronomy*, vol. 12, no. 4, p. 873, 2022, Additional documentation: https://jordan787878.github.io/firstweb/6dofQuadcopter/6dofQuad.html. DOI: 10.3390/agronomy12040873. [Online]. Available: https://doi.org/10.3390/agronomy12040873.

[38] N. Ferry, "Quadcopter plant model and control system development with matlab/simulink implementation", *Kate Gleason College of Engineering Rochester Institute of Technology Rochester, New York*, 2017.

[39] P. J. Schubel and R. J. Crossley, "Wind turbine blade design", *Energies*, vol. 5, no. 9, pp. 3425–3449, 2012, ISSN: 1996-1073. DOI: 10.3390/en5093425. [Online]. Available: https://www.mdpi.com/1996-1073/5/9/3425.

[40] MathWorks. "Wind turbine tutorial". Additional documentation: https://se.mathworks.com/help/simulink/. [Online]. Available: https://se.mathworks.com/help/sps/ug/wind-turbine.html.

[41] W. Chen, W. Hong, H. Zhang, P. Yang, and K. Tang, "Multi-fidelity simulation modeling for discrete event simulation: An optimization perspective", *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1156–1169, 2023. DOI: 10.1109/TASE.2022.3173296.

[42] M. O. Siddiqui, P. R. Feja, P. Borowski, H. Kyling, A. R. Nejad, and J. Wenske, "Wind turbine nacelle testing: State-of-the-art and development trends", *Renewable and Sustainable Energy Reviews*, vol. 188, p. 113 767, 2023, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2023.113767. [On-

line]. Available: https://www.sciencedirect.com/science/article/pii/S136403212300624X.

[43] J. P. Kleijnen, "Verification and validation of simulation models", *European Journal of Operational Research*, vol. 82, no. 1, pp. 145–162, 1995, ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(94)00016-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377221794000166.

[44] Y. Song, Z. Hu, T. Li, and H. Fan, "Performance evaluation metrics and approaches for target tracking: A survey", *Sensors*, vol. 22, no. 3, p. 793, 2022, ISSN: 1424-8220. DOI: 10.3390/s22030793. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/793.

[45] Y.-H. Huang, E.-J. Liu, B.-C. Wu, and Y.-J. Ning, "Safe uav control against wind disturbances via demonstration-guided reinforcement learning", *Drones*, vol. 10, no. 1, 2026, ISSN: 2504-446X. DOI: 10.3390/drones10010002. [Online]. Available: https://www.mdpi.com/2504-446X/10/1/2.

[46] A. Altan and R. Hacıoğlu, "Model predictive control of three-axis gimbal system mounted on uav for real-time target tracking under external disturbances", *Mechanical Systems and Signal Processing*, vol. 138, p. 106 548, 2020, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2019.106548. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327019307691.

[47] D. R. Unger, I.-K. Hung, D. L. Kulhavy, Y. Zhang, and K. Busch-Petersen, "Accuracy of unmanned aerial system (drone) height measurements", *International Journal of Geospatial and Environmental Research*, vol. 5, no. 1, Article 6, 2018, Also available in Faculty Publications, SFA ScholarWorks:

https://scholarworks.sfasu.edu/forestry/481. [Online]. Available: https://dc.uwm.edu/ijger/vol5/iss1/6/.

[48] P. Chen, Y. Dang, R. Liang, W. Zhu, and X. He, "Real-time object tracking on a drone with multi-inertial sensing data", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 131–139, 2018. DOI: 10.1109/TITS.2017.2750091.

[49] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi, "Input uncertainty sensitivity enhanced nonsingleton fuzzy logic controllers for long-term navigation of quadrotor uavs", *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 725–734, 2018. DOI: 10.1109/TMECH.2018.2810947.

[50] Y. Chen, D. Baek, A. Bocca, A. Macii, E. Macii, and M. Poncino, "A case for a battery-aware model of drone energy consumption", Oct. 2018, pp. 1–8. DOI: 10.1109/INTLEC.2018.8612333.

[51] I. Sadeghzadeh, A. Mehta, Y. Zhang, and C.-A. Rabbath, "Fault-tolerant trajectory tracking control of a quadrotor helicopter using gain-scheduled pid and model reference adaptive control", in *Annual Conference of the PHM Society*, vol. 3, 2011.

[52] F. Hosoda, Y. Tamura, and Y. Hirata, "Development of a cooperative multi-drone system for offshore wind turbine inspection", in *2025 SICE Festival with Annual Conference (SICE FES)*, 2025, pp. 187–192. DOI: 10.23919/SICEFES67750.2025.11236586.