Effects of Advanced Modeling on High-Performance, Multi-Rotor Drone Dynamics and Energy Efficiency

Samantha Hoang

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

I.Y. (Steve) Shen, Chair

Alberto Aliseda

Xu Chen

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Effects of Advanced Modeling on High-Performance, Multi-Rotor Drone Dynamics and Energy Efficiency

Samantha Hoang

Chair of the Supervisory Committee:
Professor I.Y. (Steve) Shen
Mechanical Engineering

This work focuses on using a dynamics-based approach to understand the effects of modeling choices on simulation of high-performance multi-rotor drones carrying heavy suspended payloads. Three aspects of the modeling and simulation of multi-rotor drones are examined: dynamics and system modeling choices, energy costs of controllers, and gust disturbance modeling. First, an 18-rotor drone is modeled as a rigid body using a quaternion formulation with the aerodynamics of the rotors blades simulated using HARP_Opt. To control the system, both a PID and a PD controller are developed using the same proportional and derivative gains. The gust disturbance is modeled deterministically as a simple boxcar function with a duration of one second.

First, the dynamics of the system are explored. Since these large multi-rotor drones are expected to carry heavy payloads, the effect of a heavy suspended payload is found through two means. Firstly, a linear stability analysis reveals that the model only has negative and zero eigenvalues which imply the the system is at least marginally stable. However, the stability of the nonlinear system cannot be guaranteed because of the zero-eigenvalues. To determine the stability of the nonlinear system, the system with a heavy suspended payload's response to a simple wind disturbance is simulated. The system's response reveals that the presence of the heavy suspended payload makes the system's response diverge significantly

from the case when there is no heavy suspended payload with as large as several orders of magnitude difference in the responses. These results demonstrate that the nonlinear stability of a large multi-rotor drone with a suspended payload must be considered when analyzing future drone designs for stability. Next, the effect of aerodynamic model choice for the rotors is explored by comparing a lumped blade (LB) model with a blade element theory (BET) model. The two aerodynamic models are simulated with identical inputs to follow four simple trajectories that involve vertical flight with no rotation as well as vertical flight with roll, pitch, and yaw rotation. The results show a maximum trajectory error of up to 91% in the case of vertical flight with yaw rotation. This demonstrates that aerodynamic model choice has a significant effect on simulation results and the BET model needs to be considered despite it being more computationally intensive than the LB model. The last portion of the system dynamics explored is the potential for using rotor groupings to reduce the total energy required to fly the drone. With 18 rotors, there are many ways in which the drone's trajectory can be flown. Two different rotor groupings are considered. The two groupings are used to fly the drone on almost identical trajectories. The resulting energy calculations reveals that the two groupings consistently maintain a 4-5% difference in energy cost for all the different trajectories simulated.

Second, the energy costs of controllers are explored. To do this, a method is proposed that uses the difference between open-loop system and closed-loop system energy costs to determine the energy cost of using a control system. As a case study, this method is then used to objectively compare a PID controller to a PD controller and characterize the energy cost of integral control. The resulting simulations reveal that both controllers perform similarly in tracking the desired trajectory with about a 5% average tracking error for low-amplitude trajectories and about a 20% tracking error for large-amplitude trajectories. However, the PID controller consistently uses more energy than the open-loop system (7-12.5% more) while the PD controller uses similar amounts of energy to the open-loop system. Additional simulations were completed looking at variations on the integral control. The resulting plots of energy and tracking error versus integral control gain magnitude demonstrate that there is a trade-off between tracking error and energy that needs to be balanced for each application to ensure all tracking and energy specifications are met.

Finally, the necessity of modeling gust disturbances using a stochastic model is evaluated. This is done by using a simple, deterministic gust disturbance with variable windspeed and angle and varying which rotors are affected by the gust. It is found that when all of the rotors are hit by the gust, then the drone responds mainly in the vertical direction. When half of the rotors are hit by the gust, then the drone exhibits about an order of magnitude more motion in the horizontal direction leading to very different overall responses. Changes in windspeed and angle also result in inconsistent variations in the final trajectory and energy usage as well. This unpredictability of the resulting trajectory and energy consumption shows that the drone's gust response is highly sensitive to changes in wind parameters. In addition, the drone's sensitivity to system parameter changes is tested by varying the control gains of the system. It is found that when the proportional, integral, and derivative control gains are all varied by the same amount, then the drone's trajectory and energy usage vary significantly. As the gains are reduced, the energy usage compared to the no-wind case varies by between 0.69-82.73%. In addition, when only the integral control is varied, the energy varies by 8.88-18.07% relative to the no-wind case. These results demonstrate that both trajectory and energy usage are sensitive to changes in the gust parameters which means that for more complicated gust disturbances, the drone's response will be even harder to characterize with deterministic models. Therefore, it is necessary to use stochastic gust modeling and simulations to fully characterize how a drone will respond to a random gust disturbance.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

DEDICATION

to my family and partner who have always given me their unconditional support

Chapter 1

INTRODUCTION

This chapter will introduce the background and motivations for the proposed research. Since the majority of research on multi-rotor drones has been done on quadcopters, much of the research referenced will be on quadcopters. The actual research being proposed in this document will specifically focus on 18-rotor drones.

1.1 Motivations

Multi-rotor drones are incredibly common in recreational uses, however, these drones can only carry small payloads with short flight times [1, 12–14, 21, 22, 24, 36, 37, 47]. This means that they are unable to carry heavy payloads and fly for extended periods of time; two tasks that are required for use in critical operations, such as mitigation of wild fires [17], cleaning and maintenance of power transmission towers at remote terrains [14], firefighting and rescue of high-rise buildings [59], search and rescue operations [14, 59]. To enable these critical operations, there is a strong demand for high-performance drones that can carry heavy payloads and fly for extended time periods with fail-safe capabilities that allow the drones to fly in gusty environments.

Currently, the design paradigm for simulating multi-rotor drone flight follows a classical control paradigm. In the current design paradigm, quadcopters use four rotor inputs to achieve the four control actions of roll, pitch, yaw, and vertical motion [42]. The total thrust for each rotor is calculated using aerodynamic models and then the thrust is used to determine the quadcopter's flight trajectory according to the 3D rigid-body dynamics [42]. The torque and rotation speed of the rotors governs the energy expenditure [10]. The desired flight trajectory is generated by minimizing flight time [5] or energy usage [51]. When the quadcopter deviates from the planned trajectory due to gusts [3, 16, 70] or unmodeled dynamics in the simulated model [38,54], the control law alters the four rotor inputs to return the quadcopter to the desired trajectory [71]. This current design paradigm has two major issues. First, it is not energy-centric [5, 23, 66, 74]. Second, it is not necessarily fail-safe [46]. Therefore the current paradigm faces many challenges in energy and safety management.

These challenges manifest themselves in multiple ways. First, many multi-rotor drones do not have enough redundancy. Redundancy is important for two reasons. First, the lack of redundancy could compromise the safety of the drone. If a rotor stalls or malfunctions, then the drone could lose control and crash [46, 69]. In addition, with fewer rotors, each rotor's speed must be changed frequently away from the rotation speeds where the rotor blades and motors have maximum energy efficiency. This results in the drone using extra energy unnecessarily and reducing the maximum achievable flight time.

Another way these challenges manifest is in the lack of high-fidelity modeling for both the rigid body dynamics and aerodynamics simultaneously. Currently, a lumped parameter approach is most common which describes the aerodynamics of the rotor blades using one lift coefficient and one drag coefficient that both depend on the rotor speed and induced velocity of the rotor [6]. This lumped-parameter approach ignores that the blade is not uniform and that each portion of the blade experiences different induced velocities. This approach also cannot accurately model stalls since often stalls occur over only a portion of the rotor blade [16]. As a result, making compromises in the accuracy of either calculation results in the incorrect power and energy calculations leading to inaccurate energy cost estimates. When a suspended payload is added to the system, it increases the energy consumption due to the increased mass moment of inertia and drag force on the payload. Thus, reducing energy consumption and maximizing flight times relies heavily on accurate modeling of the rigid body dynamics and aerodynamics.

In addition to improving energy calculations, high fidelity models also enable virtual environments to simulate drone flight using different controller designs [3, 65]. This aids in the design of advanced controllers. The high fidelity models also enable better prediction of stall conditions and how to recover from stall [9, 34]. These kinds of nonlinear phenomena are not captured in simpler dynamics models. As a result, the unmodeled dynamics will act as disturbances on the drone causing it to deviate from the designed trajectory and use even more energy to correct the deviation. If energy usage cannot be modeled accurately, then a truly energy efficient path cannot be generated [58].

The energy cost of control systems also are seldom quantified. Previously conducted research on controllers for multi-rotor drones is focused on developing the best controller for following a desired trajectory. The controllers developed rely on a range of techniques such as minimizing deviation from the desired trajectory [22, 43] and requiring the drone to pass through specified way points [12]. Other optimizations that have been considered are minimizing of flight time [49], avoiding obstacle [5], and minimizing payload swing [12,37]. In all of these controllers, the energy consumption is usually not considered. If it is considered at all, energy consumption is taken into account via a cost function when planning optimal trajectories [2, 43, 45, 53]. While this accounts for the nominal energy expenditure to fly a trajectory, it does not take into account the energy cost of the controller reacting other disturbances that occur during flight. In [27], two energy-optimal controllers are developed. While the energy cost to fly certain trajectories was discussed the actual energy used to fly the drone is never calculated. The main focus was on the effectiveness of these controllers at stabilization and minimizing trajectory error. Thus, even in the research on developing controllers that optimize for energy consumption, there is no distinction between the closed-loop and open-loop energy costs.

One final way that energy is not properly taken into account is through gust and disturbance modeling. It has been shown previously that gusts have a destabilizing effect on the flight of quadcopters [38]. Currently, gusts are only modeled as deterministic events [18, 23, 70, 74]. Therefore, the stochastic nature of gusts is entirely ignored. These deterministic models are chosen for simplicity of calculations and are often created to mimic the average gust behavior of the known region of interest either using mathematical functions [70] or computational fluid dynamics simulations [18]. In the area of wind turbine research wind is modeled as a stochastic process using the Weibull distribution [7, 61]. This has not been applied to multi-rotor drones in order to estimate energy costs in gusty environments. The first step of applying stochastic modeling to multi-rotor drones is to evaluate the sensitivity of the current deterministic model to changes in gust excitation parameters. High sensitivity to these parameters would indicate that it is more appropriate to use stochastic gust models as opposed to deterministic models. Thus, I am interested in determining if deterministic models are sufficient to model drone response to gusts or if applying stochastic models normally used in wind turbine research is necessary for more accurate simulation results and energy estimation.

1.2 Proposed Solution

To enable the creation and use of high-performance drones, this document proposes several novel strategies to gain a fundamental understanding of how multi-rotor drones fly with heavy payloads, how to increase flight times through energy saving strategies, and how to improve flight safety.

First, a new design for multi-rotor drones is proposed that uses a large number of rotors to make the drone less susceptible to failure due to rotor failure. Moreover, one can argue that a highly over-actuating drone is more energy-centric because the frequency and magnitude with which each rotor needs to be changed to achieve a certain trajectory can be reduced to a minimum.

Second, I propose a method for accurate prediction of energy expenditure using high-fidelity models in rigid body dynamics, aerodynamics, and mechatronics simultaneously. The rigid body dynamics model will include a heavy suspended payload and the payload's aerodynamic drag. In addition, the model will use quaternions to represent the attitude of the drone to avoid gimbal lock occurring during the simulations. Gimbal lock is the mathematical phenomenon where the attitude cannot be uniquely represented using the three Euler angles; as such a fourth dimension is needed resulting in the use of quaternions which represent the attitude using four numbers [44]. The rotor blade aerodynamics are modeled using blade element and momentum theory (BET) to calculate the lift and drag on each section of the rotor blade. This approach allows for more granularity in calculating the lift and drag on the blade and help determine if stall occurs on any portion of the blade. This will greatly increase the accuracy of the lift and drag calculations. The mechatronics will include the limitations on the electrical motors turning the rotor blades.

Third, I propose a methodology for evaluating the energy costs of different controllers. In order to do this, the open-loop system energy costs will be calculated to develop a baseline for energy consumption to fly a specified trajectory. This is important to quantify because then the true cost of a control system can be understood by comparing the closed-loop system cost with that of the open-loop system to fly the same trajectory. Using the open-loop system energy consumption as a baseline gives an objective view of controller performance making it easier to compare to the energy cost of other controllers. This methodology also makes it possible to estimate the energy cost of the correction following a disturbance. For example, if the drone will fly in areas known to have sudden gusts, a the energy usage for the open-loop system to fly a control trajectory can be compared to the closed-loop system's energy usage to correct for a chosen gust disturbance. This process can be repeated for multiple controllers to compare the performance of the various controllers. Thus, this methodology can be used to find the true optimal controller for a given flight situation.

Finally, I hypothesize that modeling gusts as a stochastic process is necessary to more accurately approximate drone response and energy expenditure. To demonstrate that using stochastic modeling and simulation is necessary despite the computation and time cost, a deterministic gust model will be used as an input to simulations. Small variations will be made to the gust disturbance. If these variations result in large variations in the drone's response, then this suggests that the drone's response is very sensitive to small changes in the gust disturbance. This would imply that more complex gust disturbances, such as a gust disturbance modeled stochastically, would result in even larger variations in the response. These large variations would make it difficult to fully characterize the drone's response only using deterministic gust models. This would mean that it would be necessary to model gusts stochastically in order to fully characterize a drone's response to gust disturbances.

1.3 Research Aims

This project aims to answer several research questions in developing the above proposed solution. The overarching theme to these questions is that they are intended to evaluate current modeling and design choices and compare them to new methods for modeling and simulating multi-rotor drones that are conducive to creating a truly energy-optimal drone and control system.

1. How do modeling choices affect the simulation and flight trajectory of multi-rotor drones? The first question can be broken down into two main components.
    - How do modeling choices in the suspended payload and rotor aerodynamics model affect results when simulating the multi-rotor drones?
    - For highly overactuated drones, can rotors be grouped to optimize for energy usage in both an open- and closed-loop system? Can different rotor groupings fly the same trajectories or are there limitations based on how the rotors are grouped?
2. Can we quantify the energy costs inherent to different control systems? Will certain control systems have a more energy efficient response?
3. Is modeling gusts as a stochastic process necessary for more accurate simulations or is modeling gusts as a deterministic process sufficient?

1.3.1 Aim 1: Understanding effects of modeling choices on 3D rigid body dynamics and rotor aerodynamics

For this aim, I develop a high-fidelity model that simultaneously uses complex models in rigid body dynamics, rotor blade aerodynamics, and mechatronics. This advanced dynamical model uses quaternions to represent the attitude of the multi-rotor drone and includes the presence of a suspended payload. The rotor aerodynamics will be modeled using a blade element theory (BET) model in order to take into account the non-uniformity in geometry of each rotor blade. In addition to modeling the dynamics with more complex models, I will also simulate how the drone flies using different rotor groupings. These rotor groupings could lead to more energy-efficient flights by isolating the necessary rotor inputs to achieve a certain motion to independent groups of rotors.

Using the above model, I varied several model parameters to understand how they affect the dynamics of the open-loop system. This work can be broken down into three sub-aims as follows.

- **Aim 1a: Determine how payload mass and location affect the open-loop motion of the multi-rotor drone**
  In [28], I demonstrate that including the suspended payload has a significant effect on the motion of the drone. In open-loop systems, variations in the mass and location of the payload cause changes in the type of instability experienced. When there is no suspended payload (i.e., it is located at the center of the drone body) then the system will experience a diverging instability when hit by a gust of wind. When there is a suspended payload, then the system will experience an oscillating instability when hit by a gust of wind.

- **Aim 1b: Determine how significant the choice of aerodynamic model is for the rotors on the final simulated trajectory**
  In [29], I demonstrate that different aerodynamic models, specifically the BET and lumped-parameter models, result in different simulated trajectories. Specifically, I found that the lumped-parameter models severely under-predicted the motion in the vertical and yaw directions.

- **Aim 1c: Determine if rotor groupings can be optimized for more energy-efficient flight**

In [30], I demonstrate that different rotor groupings use different amounts of energy to fly. In addition, I demonstrate that the difference in energy does not compound with increasing complexity of the trajectory which implies that the energy differences between rotor groupings are inherent to the groupings themselves and examining simple trajectories is enough to determine which rotor grouping will be more energy efficient to fly.

1.3.2 Aim 2: Quantify the energy costs of controls

This aim focuses on the closed-loop system and comparing its performance to the open-loop system. There is an energy cost to using controllers in a closed loop system since the controllers must adjust the rotor speed inputs in order to correct errors in the trajectory. This should lead to the closed-loop system using more energy since the controller adjustments will cause the rotor speeds to deviate from their most energy-efficient values.

- **Aim 2a: Evaluate the difference between open- and closed-loop energy consumption**

  In [31], I develop a framework for quantifying the energy consumption of a control system by comparing it to the energy cost for the open-loop system to fly the same trajectories. I implement a PID control system for the multi-rotor drone and use this closed-loop system to demonstrate how to use the framework.

- **Aim 2b: Compare the energy costs of two controllers**

  In [31], I use the framework from Aim 2a to demonstrate its usefulness by comparing the energy cost of two controllers. Using the framework, I compare the energy costs of the PID control system from Aim 2a and a PD control system with the same proportional and derivative gains. Through this comparison, I quantify the energy cost of integral control and demonstrate how the framework can be used to perform a trade-off analysis in order to choose the most appropriate controller for a specific task.

1.3.3 Aim 3: Demonstrate that modeling gusts as a stochastic process is necessary

In this aim, I demonstrate that deterministic gust modeling is not sufficient to fully capture a multi-rotor drone's response to apply a simple, gust to the large multi-rotor drone and control system used in the previous aims. The resulting trajectories and energy calculations will be analyzed to demonstrate that the drone's response is very sensitive to variations in the gust disturbance. Several scenarios are considered and compared including different windspeeds, wind directions, and locations on the drone where the wind will hit it. The results of these comparisons demonstrate the drone's trajectory and energy usage in response to a gust disturbance are very sensitive to both variations in the gust and in the control system. These results lead me to conclude that stochastic modeling is necessary to fully understand how a drone will response to a gust disturbance.

- **Aim 3a: Evaluate sensitivity of system gust response to gust disturbance parameters**
  I use a simple gust disturbance to show that when some wind parameters are varied, the characteristics of the drone's response to the gust disturbance will change. The three wind parameters tested are the windspeed, wind angle, and which rotors the wind hits. The largest difference in drone response is seen when which rotors the wind hits is varied.

- **Aim 3b: Evaluate the effects of control gain modification when responding to gust disturbances**
  I vary the control gains in the PID controller used in Aim 2 to determine if stochastic simulations are required to evaluate controller performance to different gust scenarios. When all three control gains are varied, the drone's response characteristics change drastically demonstrating that the drone's response to a gust disturbance is highly sensitive to changes in the controller. Therefore, additional stochastic simulations are necessary to fully characterize the behavior of the new control system.

- **Aim 3c: Demonstrate sensitivity of total energy consumption to gust disturbances**

  I determine that the total energy used varies depending on the gust disturbance since many factors affect to the total energy used during the drone's trajectory, not just the presence of stall or rotor speeds. In addition, it was found that energy usage is also very sensitive to changes in the control gains, especially when all of the control gains are varied.

Chapter 2

AIM 1: UNDERSTANDING EFFECTS OF MODELING CHOICES ON 3D RIGID BODY DYNAMICS AND ROTOR AERODYNAMICS

In Aim 1, I demonstrate how different modeling choices affect simulation results for multi-rotor drones. Several modeling and design variations are examined including suspended payload mass and location, choice of aerodynamic models for rotor blades, and effect of rotor groupings on energy efficiency. Specifically, I found that payload mass and location have a significant effect on dynamics of the system for Aim 1a. In Aim 1b, I demonstrate that the different rotor aerodynamic models produce significantly different trajectories particularly in the vertical and yaw directions. In Aim 1c, I demonstrate the existence of rotor groupings that have differing energy efficiencies thus demonstrating the rotor groupings can be optimized for energy efficiency. The results demonstrating Aims 1a, 1b, and 1c are published in [28–30].

2.1 Prior Work

Multi-rotor drones have been used in a wide array of applications ranging from commercially-available quadcopters that are used by hobbyists and researchers to military-grade drones that are used for specialized missions. Quadcopters are commonly used by researchers [1, 12, 47] and hobbyists because they are commercially available, inexpensive, and easy to use. Because quadcopters are commercially-available, they are easy to apply to various research problems. Despite these advantages, quadcopters are low-powered and have low endurance. Hence, they cannot carry heavy payloads and cannot fly for extended time periods. Military drones, on the other hand, are high-powered and have high endurance because they are designed to carry out specific missions that often involve extended flight times, heavy payloads, and extreme maneuvers [74]. However, due to their specialization, military drones are expensive to produce. Currently, there has been little research or development on drones that fill the space between commercial-grade quadcopters and military-grade drones [23]. Thus, there is a need to understand these so-called high-performance drones that are capable military drone power and endurance while being generalizable and relatively inexpensive like quadcopters. These drones that occupy the space between quadcopters and military drone can be used to tasks such as search and rescue [59], agriculture [67], or remote utilities maintenance. Applications in agriculture and utilities maintenance, specifically, may require the drone to carry a heavy payload with a mass that may vary with time and to operate in environments with many disturbances. These tasks may be impossible for quadcopters to perform and too delicate for military drones to perform so high-performance drones must be developed.

2.1.1 Heavy Payloads

While a lot of research has been completed on multi-rotor drones, most research focuses on controls instead of dynamics. Within controls, the focus is on creating control algorithms and trajectory planners for quadcopters [19,40,43,62,64]. Only a small portion of work has been done on understanding the dynamics of quadcopters [20,22,32]. A portion of the controls research focuses specifically on control systems and trajectory planners for quadcopters carrying payloads [13,14,21,24,25,36,40,41,51,64]. The mass of the payloads carried in testing, however, only ranges from about 5-20% of the drone's mass [13,21] in theoretical simulations and 5-11% in experimental tests [25]. Heavier payloads can only be achieved when multiple quadcopters are used to carry a single payload [40,41]. Using multiple quadcopters to carry a heavy payload is disadvantageous because it increases the chance for errors and the severity of the consequences that result from these errors. Ideally, this heavy payload would be carried by a single high-performance drone because it has the power that quadcopters lack to carry a heavy payload.

2.1.2 High-Fidelity Modeling

Current research has not adopted dynamic models that use advanced modeling techniques for both multi-body dynamics and aerodynamics simultaneously [35, 55, 63]. Most drone models use aerodynamic models that take a lumped-parameter approach to calculating lift and drag coefficients for a rotor as a function of rotation speed and induced velocity [33]. This lumped-parameter approach may not be accurate for many drones since the size of the rotor blade is not significantly smaller than the size of the drone. As a result, how the induced velocity and other blade properties (e.g., pitch angle) change over the blade's length needs to be taken into account when calculating lift and drag. There has been research on using blade element theory to model the rotor blades that takes into account the varying induced velocity and blade profile. It was done, however, without the advanced multi-body dynamics modeling techniques [6]. In general, currently used drone models may not be able to determine endurance or to optimize trajectory accurately due to their lacking the capability to simultaneously model multi-body dynamics and aerodynamics.

2.1.3 Rotor Grouping

Another disadvantage of quadcopters is a lack of endurance because all four rotors are constantly accelerating and decelerating to perform even basic maneuvers in the vertical, roll, pitch, or yaw directions. This inefficiency in energy usage limits the length of missions that can be performed by a quadcopters. This means that quadcopters would be unable to perform extended missions such as search and rescue operations [59] that could last an indefinite amount of time.

The current design paradigm of multi-rotor drones, however, cannot be easily adapted to address these critical factors. There are several reasons. First, many multi-rotor drone platforms do not have enough redundancy. For example, quadcopters are controlled via four control actions (thrust, roll, pitch, and yaw) using four rotor speeds as inputs [42]. As a result, each rotor speeds up and down frequently, deviating from rated rpms where the blades

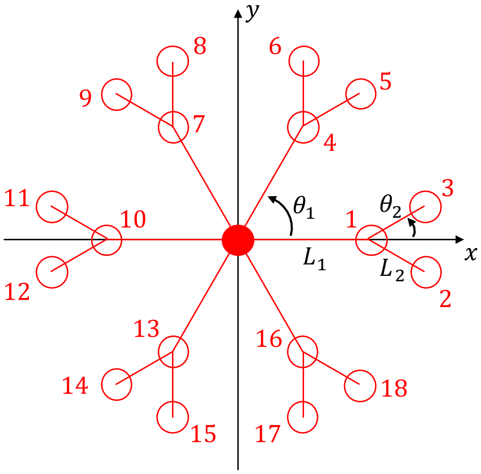


Figure 2.1: Proposed propeller geometry design for a high performance drone

and motors reach their peak efficiency. Energy is then wasted resulting in low endurance. The same design philosophy and control schemes are adopted even for overactuated systems such as hexacopters or octocopters [50, 72]. Lack of redundancy also means lack of safety. If a rotor stalls or malfunctions, the drone could lose control and crash [47].

This concern can be largely addressed by employing significantly more than 4 rotors in a drone (e.g., 18 rotors). The large number of rotors will not only increase the payload but also provide design and control flexibility. There is now a possibility of grouping rotors to deliver the four control actions independently for energy saving or trajectory optimization. For example, the rotors can be grouped in such a way that the total energy needed to fly a specific trajectory is minimized. This opens up a new space in which the control scheme can be optimized to minimize energy consumption while maximizing payload capacity. It also significantly improves safety margins for drones to recover from incidents such as rotor failures or stalls due to gusts.

One possible high-performance drone design is shown in Fig. 2.1. With eighteen propellers, this drone can generate enough thrust to carry payloads as heavy as the drone itself. In addition, the eighteen propellers can be divided into subgroups to control different basic movements such as roll, pitch, yaw, and vertical movement. However, in order to properly develop a drone with this design, a full understanding of the drone's flight dynamics and how the payload affects these dynamics is required since other researchers have proposed that the presence of a payload will inherently make a drone unstable when subjected to an external disturbance such as wind [6] or rotor-induced vibrations [54, 68]. Thus, it is important to understand what configuration will minimize this instability.

2.2 Modeling

Before any simulations can be done to understand the sensitivity of the drone's dynamics to different parameters and modeling choices, the dynamical model must be derived. Several different aerodynamic models for the rotor blades are used. Section 2.2.2 is taken from [28] and used in Aim 1a. Sections 2.2.2 and 2.2.2 are taken from [30]. For the variable definitions see Appendix A.

2.2.1 Drone Model Derivation

The following model derivation text is originally written in the appendix of [30]. This is the rigid body dynamics model that will be used throughout the completed and proposed research.

In order to derive the dynamical model for a drone with its center of mass located away from its geometric center, several aspects of the model must be addressed first. The most important is to determine how to represent the orientation of the drone. Typically, Euler Angles are used to represent the orientation using the roll, pitch, and yaw angles. The main issue with using Euler Angles is that they are subject to gimbal lock, a phenomenon when three state variables are no longer sufficient to uniquely represent the drone's orientation. In order to avoid gimbal lock when simulating the drone system, quaternions are used to represent the orientation. The quaternion representation of the orientation uses 4 state variables which means that any orientation can be represented uniquely.

When using quaternions, a special operation specific to quaternions is required. The $\otimes$

operator represents a quaternion multiplication operation that is used to convert between reference frames. This is defined for two quaternions $\bar{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = q_0 + \mathbf{q}$ and $\bar{p} = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k} = p_0 + \mathbf{p}$ as follows.

$$\bar{q} \otimes \bar{p} = q_0 p_0 - \mathbf{q} \cdot \mathbf{p} + q_0\mathbf{p} + p_0\mathbf{q} + \mathbf{q} \times \mathbf{p} \tag{2.1}$$

In order to use this operation on a three-component vector, the vector can be written as a quaternion. For example for a vector $\mathbf{v} = v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}$ to be used in a quaternion multiplication, it is rewritten as a quaternion as follows $\bar{v} = 0 + v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k} = 0 + \mathbf{v}$.

After establishing how to perform the quaternion multiplication, how it is used to convert between reference frames can be described. Consider a system where we must convert from a stationary frame to a moving frame. Let's consider a vector $\mathbf{r}_0^b$ that is written in the rotating frame. To convert this vector to the stationary frame, $\mathbf{r}_0^e$, the equation is

$$\mathbf{r}_0^e = \bar{q} \otimes \mathbf{r}_0^b \otimes \bar{q}^* \tag{2.2}$$

where $\bar{q}^* = q_0 - \mathbf{q}$ is the conjugate of the quaternion $\bar{q}$. With this information, the equations of motion can be derived.

First, the equations of motion for translation can be derived by using Newton's Second Law which states that $\sum \mathbf{F} = M\mathbf{a}_G$ where $\mathbf{a}_G = \ddot{\mathbf{r}}_G$. Here, r_G is the location of the center of mass of the drone. This is a difficult location to track so we want to instead find the location of the geometric center of the drone body without the payload and measure that position. In order to do that, we define a position vector pointing to the geometric center from the center of gravity, $\mathbf{r}_0^b$, such that the global position vector to the geometric center is $\mathbf{r} = \mathbf{r}_G + \mathbf{r}_0^e$. Since $\mathbf{r}_0^e$ is originally written in the body frame, then to take its derivative in the stationary frame, it must converted to the stationary frame. Thus the position vector becomes $\mathbf{r} = \mathbf{r}_G + \bar{q} \otimes \mathbf{r}_0^b \otimes \bar{q}^*$. The first derivative and second derivative are found to be

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}_G + \bar{q} \otimes (\Omega \times \mathbf{r}_0) \otimes \bar{q}^* \tag{2.3a}$$

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_G + \bar{q} \otimes \Big(\dot{\Omega} \times \mathbf{r}_0 + \Omega \times (\Omega \times \mathbf{r}_0)\Big) \otimes \bar{q}^* \tag{2.3b}$$

Then Eqn. (2.3b) can be solved for $\ddot{\mathbf{r}}_G$ to be used in Newton's Second Law.

In order to get the equation of motion for rotation, we use the equation $\sum \mathbf{M}_G = \frac{d}{dt}\left[I_G\Omega\right]$. We calculate the derivative on the right hand side of the expression is as follows.

$$\frac{d}{dt}[I_G\Omega] = I_G\dot{\Omega} + \Omega \times (I_G\Omega) \tag{2.4}$$

Finally, we can express the different forces that the drone will experience as external forces and torques ($\mathbf{F}^e_E$ and $\mathbf{T}^e_E$) and rotor forces and torques ($\mathbf{F}^b_B$ and $\mathbf{T}^b_B$). The rotor forces and moments need to be converted to the stationary frame in the equations and the location at which the forces are applied relative to the geometric center is ξ^e_E and ξ^b_B for the external forces and rotor forces, respectively. Thus the sum of the forces and moments can be expressed as follows.

$$\sum \mathbf{F} = \mathbf{F}^e_E + \bar{q} \otimes \mathbf{F}^b_B \otimes \bar{q}^* \tag{2.5a}$$

$$\sum \mathbf{M}_G = \mathbf{T}^b_B + \bar{q}^* \otimes \mathbf{T}^e_E \otimes \bar{q} + \left(\xi^b_B + \mathbf{r}^b_0\right) \times \mathbf{F}^b_B + \left(\xi^b_E + \mathbf{r}^b_0\right) \times \left(\bar{q}^* \otimes \mathbf{F}^e_E \otimes \bar{q}\right) \tag{2.5b}$$

Then we can combine the results of Eqns. (2.3b), (2.4), and (2.5) to get the final translational and rotational equations of motion as follows.

$$M\ddot{\mathbf{r}}^e - M\bar{q} \otimes \Big(\dot{\Omega} \times \mathbf{r}^b_0 + \Omega \times \left(\Omega \times \mathbf{r}^b_0\right)\Big) \otimes \bar{q}^* = \mathbf{F}^e_E + \bar{q} \otimes \mathbf{F}^b_B \otimes \bar{q}^* \tag{2.6}$$

$$I_G\dot{\Omega} + \Omega \times (I_G\Omega) = \mathbf{M}^b_B + \bar{q}^* \otimes \mathbf{M}^e_E \otimes \bar{q} + \left(\xi^b_B + \mathbf{r}^b_0\right) \times \mathbf{F}^b_B + \left(\xi^b_E + \mathbf{r}^b_0\right) \times \left(\bar{q}^* \otimes \mathbf{F}^e_E \otimes \bar{q}\right) \tag{2.7}$$

The last dynamical equations necessary is one that describes the dynamics of the quaternions. This is found by taking the derivative of the quaternion and simplifying the expression to get the following.

$$\dot{\bar{q}} = \frac{1}{2}\Omega \times \bar{q} \tag{2.8}$$

Thus the three sets of equations necessary to describe the motion of the drone are found. The translational equation of motion given in Eqn. (2.6) can be split into two first order equations. Eqn. (2.6) is converted from the stationary to the rotating frame and modified to describe the dynamics of the velocity in the body frame. Then another differential equation that describes the dynamics of the position given the velocity in the body frame is also formulated. The resulting two equations are given below.

$$\dot{\mathbf{r}}^e = \bar{q} \otimes \mathbf{v}^b \otimes \bar{q}^* \tag{2.9a}$$

$$\dot{\mathbf{v}}^b = \left(\dot{\Omega} \times \mathbf{r}_0^b + \Omega \times \left(\Omega \times \mathbf{r}_0^b\right)\right) + \bar{q}^* \otimes \tfrac{\mathbf{F}_E^e}{M} \otimes \bar{q} + \tfrac{\mathbf{F}_B^b}{M} \tag{2.9b}$$

Euler Angles Model

This section is taken from [28].

The equivalent equations of motion using Euler Angles are given by the following equations.

$$M\ddot{\mathbf{r}}^e - M\mathbb{R}^T \left(\dot{\Omega} \times \mathbf{r}_0^b + \Omega \times \left(\Omega \times \mathbf{r}_0^b\right)\right) = \mathbf{F}_E^e + \mathbb{R}^T \mathbf{F}_B^b \tag{2.10}$$

$$\begin{aligned} I_G\dot{\Omega} + \Omega \times (I_G\Omega) = \mathbf{M}_B^b + \mathbb{R}\mathbf{M}_E^e + \left(\xi_B^b + \mathbf{r}_0^b\right) \times \mathbf{F}_B^b \\ + \left(\xi_E^b + \mathbf{r}_0^b\right) \times \left(\mathbb{R}\mathbf{F}_E^e\right) \end{aligned} \tag{2.11}$$

$$\dot{\Phi} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \Omega \tag{2.12}$$

A simplification is completed for the translational equations of motion given by Eqn. (2.10) by recognizing that a rotation from the body frame to the earth frame is given by $\mathbf{v}^e = \mathbb{R}^T\mathbf{v}^b$, and a rotation from the earth frame to the body frame is given by $\mathbf{v}^b = \mathbb{R}\mathbf{v}^e$. The resulting first-order equations are given below.

$$\dot{\mathbf{r}}^e = \mathbb{R}^T v^b \tag{2.13a}$$

$$\dot{\mathbf{v}}^b = \left(\dot{\Omega} \times \mathbf{r}_0^b + \Omega \times \left(\Omega \times \mathbf{r}_0^b\right)\right) + \mathbb{R}\frac{\mathbf{F}_E^e}{M} + \frac{\mathbf{F}_B^b}{M} \tag{2.13b}$$

2.2.2 Aerodynamic Models

Three different aerodynamic models for the rotor blades will be used throughout this research. The first model is taken from [6] and is used in Aim 1a. The second and third models are developed to use the results of Computation Fluid Dynamics (CFD) simulations using HARP_Opt [48]. These models, described in Secs. 2.2.2 and 2.2.2 are used in Aims 1b and 1c.

First Aerodynamic Model from [6]

The main forces to be modeled are the lift and drag forces acting on the propellers of the drone. The forces on the arms of the drone will be ignored. The model used to estimate the lift and drag forces is taken from Bristeau [6]. The lift force (**LF**) from the propellers of a quadcopter is given by

$$\mathbf{LF} = \rho c R^3 \omega^2 C_{L\alpha}\left(\frac{\alpha_t}{3} - \frac{1}{2R|\omega|}\left(\varepsilon_1 q - \varepsilon_2 p\right)\right)\mathbf{z_b} \tag{2.14}$$

where ε_1 and ε_2 are dictated by the location of the motors for which the forces are being calculated. The parameters ε_1 and ε_2 can be derived for each propeller by considering their location relative to the drone's body. For example, ε_1 and ε_2 for the i-th propeller on an eight-propeller drone shown in Fig. 2.2 can be derived as follows.

$$\varepsilon_{1,i} = L\cos\frac{(i-1)\pi}{4}$$
$$\varepsilon_{2,i} = L\sin\frac{(i-1)\pi}{4}$$

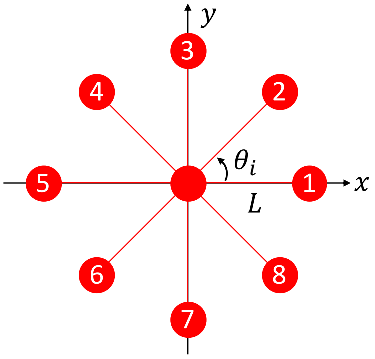


Figure 2.2: Geometry of an eight-armed drone as an example of how the geometry parameters in the lift force equation are derived.

The remaining lift and drag forces and moments are given as follows [6]. Unlike the lift force, the drag force is broken up into three components; the nominal drag due to the pitch angle of the blade ($\mathbf{DF}_0$), the drag induced by the propeller rotation ($\mathbf{DF}_i$), and the drag due to lift ($\mathbf{DF}_L$), as shown below.

$$\mathbf{DF}_0 = \frac{1}{2}\rho c R^2 C_{D0}\left(|\omega|\left(\bar{u}\mathbf{x_b} + \bar{v}\mathbf{y_b}\right) - sgn(\omega)\bar{w}\left(p\mathbf{x_b} + q\mathbf{y_b}\right)\right) \tag{2.15a}$$

$$\mathbf{DF}_i = \rho c R^3 C_{Di}\alpha_t\left(\frac{\bar{w}}{R^2}\left(\bar{u}\mathbf{x_b} + \bar{v}\mathbf{y_b}\right) + \frac{\omega}{3}\left(p\mathbf{x_b} + q\mathbf{y_b}\right)\right) \tag{2.15b}$$

$$\mathbf{DF}_L = -\frac{1}{2}\rho c C_{L\alpha}\left(R\alpha_t\bar{w}\left(\bar{u}\mathbf{x_b} + \bar{v}\mathbf{y_b}\right) + \left(sgn(\omega)R^2\bar{w} - \frac{R^3\alpha_t\omega}{3}\right)\left(p\mathbf{x_b} + q\mathbf{y_b}\right)\right) \tag{2.15c}$$

The lift moment ($\mathbf{LM}$) is simply given by

$$\mathbf{LM} = \rho c R^4 C_{L\alpha}\left(\frac{|\omega|}{8}\left(p\mathbf{x_b} + q\mathbf{y_b}\right) + \frac{\bar{w}sgn(\omega)}{4R^2}\left(\bar{u}\mathbf{x_b} + \bar{v}\mathbf{y_b}\right)\right). \tag{2.16}$$

The drag moment ($\mathbf{DM}$) on the propeller is given by

$$\mathbf{DM} = -sgn(\omega)\rho c R^4\omega^2 C_{L\alpha}\left(\frac{C_{D0} + C_{Di}\alpha_t^2}{4} - \frac{C_{L\alpha}\alpha_t\bar{w}}{3R|\omega|}\right)\mathbf{z}_b. \tag{2.17}$$

The forces and moments listed in Eqns. (2.14–2.17) are all applied at the axis of rotation of the propeller that is creating it. As such, when forming the forcing terms for the moment balance, we need to include the moment from each of the propellers as well as the moment due to the lift from each propeller.

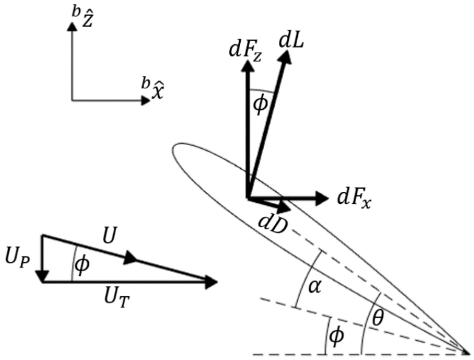


Figure 2.3: Airfoil representing a cross section of the i-th element of the blade [69, Fig. 4]

Blade Element Theory Model

The BET model divides the rotor blade into n elements where the lift and drag are calculated for each element and then summed together to get the total lift and drag on each rotor blade [69]. The BET model described here treats each element of the blade as an airfoil as pictured in Fig. 2.3 and the lift and drag of the i-th element are calculated as follows.

$$L_i = \frac{1}{2}\rho U_i(\omega, V, r_i)^2 C_{L,i}(\omega, U_i, r_i)A_i \tag{2.18a}$$

$$D_i = \frac{1}{2}\rho U_i(\omega, V, r_i)^2 C_{D,i}(\omega, U_i, r_i)A_i \tag{2.18b}$$

where $C_{L,i}$ and $C_{D,i}$ are the i-th element's coefficient of lift and drag, U_i is the i-th element's induced velocity, r_i is the i-th element's distance from the rotor axis, and A_i is the i-th element's effective surface area. The total lift and drag can then be calculated by summing together the lift and drag for each element of the blade.

$$L = \sum_{i=1}^{n} L_i \tag{2.19a}$$

$$D = \sum_{i=1}^{n} D_i \tag{2.19b}$$

From the lift L_i and drag D_i, the total thrust $(F_z)_j$ produced by the j-th rotor can be found by summing the vertical components of lift and drag, as indicated in Fig. 2.3, as follows.

$$(F_z)_j = \sum_{i=1}^{n} L_i \cos(\phi_i) - D_i \sin(\phi_i) \tag{2.20}$$

where ϕ_i is the angle ϕ indicated in Fig. 2.3. The thrust from each of the propellers can then be summed to get the total thrust applied to the drone in the body frame. Similarly, the torque on the j-th rotor produced by the rotors spinning can be calculated as the sum of the torques caused by the horizontal components of lift and drag as indicated in Fig. 2.3.

$$T_j = \sum_{i=1}^{n} L_i r_i \sin(\phi_i) + D_i r_i \cos(\phi_i) \tag{2.21}$$

Lumped Blade Model

The LB model, unlike the BET model, does not divide the blade into n elements and instead uses the average of the values for the coefficients of lift and drag, the induced velocity, and the average angle ϕ of the rotor blade to calculate the total force and torque acting on the propeller. The lift and drag for the LB model is calculated as follows.

$$L = \frac{1}{2}\rho U_{avg}(\omega, V)^2 C_L(\omega, U_{avg}) A \tag{2.22a}$$

$$D = \frac{1}{2}\rho U_{avg}(\omega, V)^2 C_D(\omega, U_{avg}) A \tag{2.22b}$$

where C_L and C_D are the coefficients of lift and drag, U_{avg} is the average induced velocity of the rotor blade, and A is the effective surface area of the entire blade. The total thrust and torque produced by each rotor can then be found by finding the vertical and horizontal components, respectively, of the lift and drag and summing them together as done in Eqns. (2.20) and (2.21).

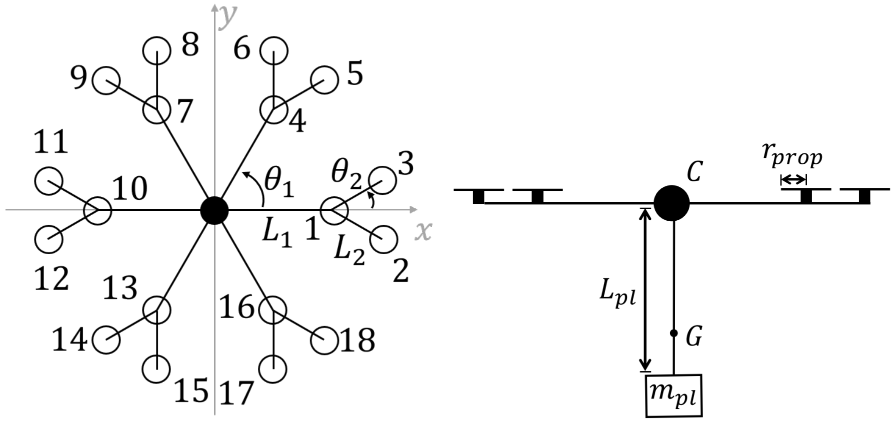


Figure 2.4: Top (left) and side view (right) of reference drone

$$(F_z)_j = L\cos(\phi_{avg}) - D\sin(\phi_{avg}) \tag{2.23a}$$

$$T_j = \frac{r_{prop}}{2}\left(L\sin(\phi_{avg}) + D\cos(\phi_{avg})\right) \tag{2.23b}$$

where ϕ_{avg} is the average angle ϕ indicated in Fig. 2.3 over the length of the rotor blade.

Reference Model

For all simulations in this section, the same drone model will be used. Figure 2.4 shows the rotor configuration and relevant geometries for the reference drone model. The values for the relevant geometries of the drone are given in Tab. 2.1. Previously, two variations on two features (i.e., grouping and aerodynamic model) of the drone model have just been described. The reference drone model chooses one variation for each of these features to use in all simulations unless otherwise stated. The BET model will be used for the aerodynamic model of the rotors, and Grouping 1 will be used to dictate how each motion will be initiated.

In addition to describing the geometries and configurations of the drone itself, it is also important to establish the model for the rotor blade. Figure 2.5 gives a side and three-dimensional view of the rotor blade design used for the drone model. The blade's aerody-

Table 2.1: Reference drone dimensions

Dimension	M	m_{pl}	r_{prop}	L_1	L_2	L_{pl}	θ_1	θ_2
Value	141.78 kg	30 kg	0.50 m	1.31 m	0.842 m	0.20 m	60°	30°

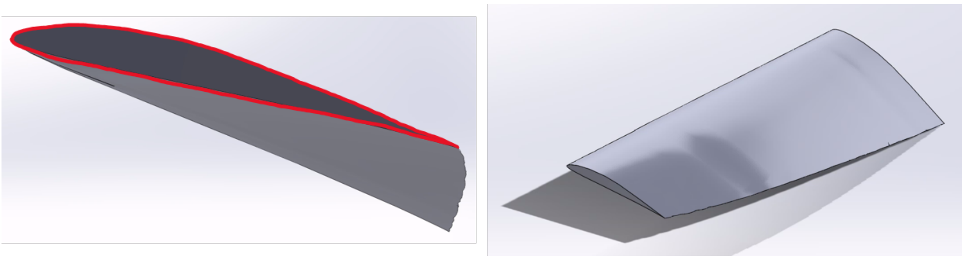


Figure 2.5: Cross-section of rotor blade (left) outlined in red and 3D view of rotor blade (right)

namic properties are calculated using the Horizontal Axis Rotor Performance Optimization (HARP_Opt) program from the National Renewable Energy Lab (NREL) [48]. HARP_Opt calculates the coefficients of lift and drag for different rotor speeds and induced velocities. The data from HARP_Opt is consolidated into N-dimensional tables, and the coefficients of lift and drag are calculated from these tables using interpolation methods at each time step. The same HARP_Opt data is used for both the BET model and the LB model so that the resulting trajectories can be compared directly. The aerodynamic data from the HARP_Opt simulations will be averaged over the span of the rotor blade in order to produce the average coefficients of lift and drag used in the LB element model.

Motor Power

In order to compare the different aerodynamic models and rotor groupings, a method for calculating power consumed by the rotor must be chosen. The drone's power usage is approximated using the torque on the rotor blade and its angular velocity. The equation for power consumed by the j-th motor is written as follows.

$$P_j = \omega T_j \tag{2.24}$$

where T_j is the torque on the j-th rotor as given in Eqn. (2.21) or Eqn. (2.23b) depending on if the BET or LB model is used.

The total energy used by the drone to fly a trajectory is calculated by integrating the total power for each of the j rotors.

$$E = \sum_j \int_0^t P_j dt \tag{2.25}$$

where t is the time it takes to fly the trajectory and the power P_j is found through Eqn. (2.24).

2.3 Aim 1a: Determine how payload mass and location affect the open-loop motion of a multi-rotor drone

This section seeks to quantify the degree to which the presence of the payload causes instability in the system and the magnitude of this instability when the system is subjected to external disturbances. The two main parameters that will dictate the drone's stability are the length of the arm or cable connected to the payload and the mass of the payload. Both of these parameters will be varied individually in order to determine which has a larger effect, if any. In addition, a linear stability analysis will be completed and the results compared to nonlinear simulations of drone flight when subjected to varying payload masses and payload arm lengths. The comparison will reveal how the effect of the payload varies between the linear and nonlinear models. The results from this comparison will determine the effectiveness of using a linearized model in the control system for a drone carrying a payload.

The following text is published in [28].

2.3.1 Linearization

The first step in understanding the stability of the drone-payload system is to linearize the system about different equlibrium and trim points. An equilibrium point is a position in which the system is at a steady state equilibrium. For a drone in a disturbance-free environment, hovering is an equilibrium point. A trim point is a flight condition wherein the system is moving, but the flight is not changing. Vertical flight would be considered a trim point. Understanding the stability of these equilibrium and trim points will provide insight into how the payload location affects the stability of the system. The linearized system can then be modeled as shown below

$$\dot{\xi} = A\xi + B\eta \tag{2.26}$$

where A and B are the system matrices, ξ is the state vector, and η is the input vector. A and B are found by taking the Jacobian of the original system of equations with respect to the states and the inputs, respectively, and evaluating the Jacobian at various equilibrium and trim points. The state vector is a column vector of the partials of each state ($\partial \mathbf{r}^e$, $\partial \mathbf{v}^b$, $\partial \Omega$, and $\partial \bar{q}$ or $\partial \Phi$) and the input vector is $\eta = [\partial \omega_i]$ for $i = 1,\ 2,\ \ldots,\ n$ where n is the number of propellers in the system.

The eigenvalues of A determine the system's stability and thus provide a quick method to check the stability of new payload configurations. As such, a study of how the eigenvalues of A changes with system parameters will provide insight into how the parameters affect the system's stability.

Equilibrium and Trim Points

Two of the equilibrium and trim points that are important to understand are hovering and vertical flight. Both hovering and vertical flight occur during takeoff and landing. Since these two actions must be completed by the drone before beginning its mission, it is important to understand the drone's stability in these sensitive conditions to avoid crashes. This is especially true in gusty conditions because the drone's control system may not provide

Table 2.2: Summary of equilibrium and trim conditions

Equilibrium or Trim point	Conditions
Hovering	$\phi = \theta = \dot{\phi} = \dot{\theta} = \dot{\psi} = 0$ $u = v = w = \dot{u} = \dot{v} = \dot{w} = 0$ $p = q = r = \dot{p} = \dot{q} = \dot{r} = 0,$ $\psi = \psi_e$
Vertical Flight	$\phi = \theta = \dot{\phi} = \dot{\theta} = \dot{\psi} = 0$ $u = v = \dot{u} = \dot{v} = \dot{w} = 0$ $p = q = r = \dot{p} = \dot{q} = \dot{r} = 0,$ $\psi = \psi_e$, $w = w_e$

enough margin to prevent a crash. Thus it is important to make sure the system is as stable as possible. In addition, it is also important to study the stability of the level, forward flight condition since the drone will need to fly forward while traveling to its mission location.

The Euler Angles formulation of the system will be linearized since the linearized system should never reach gimbal lock. Therefore the Euler Angles formulation will never be invalidated. In addition, using the Euler Angles formulation makes it easier to define the equilibrium and trim points. The conditions for the state variables and their derivatives for each equilibrium and trim point are listed in Tab. 2.2.

Simulation Results

This section will specifically examine vertical flight, since this condition must be stable or stabilizable for the drone to be able to safely take off before and land after completing a mission. This flight condition is critical to ensuring the safety of the drone and completion of its mission. The flight conditions for vertical flight are given in Tab. 2.2 with the vertical

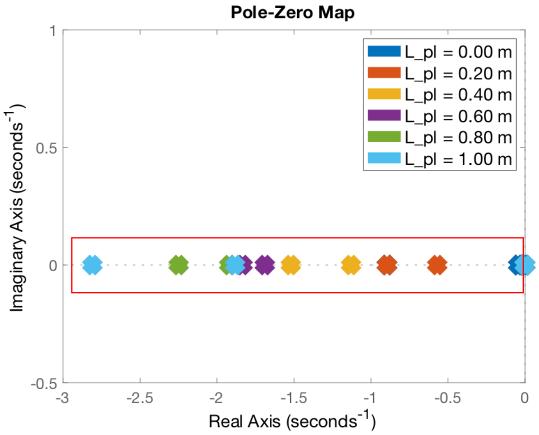


Figure 2.6: Poles of system matrix with $m_{pl} = 30$kg as payload arm length is varied.

velocity chosen as $w_e = 0.5$ meters per second and the yaw angle arbitrarily chosen as $\psi_e = 5°$ (0.0873 radians). The required inputs to the propellers ω_i is then calculated from Eq. 2.10. There are no other inputs or disturbances included in the linearized system.

For ease of simulation, an eight-armed drone will be used in all following results. Equations (2.11), (2.12), and (2.13) were rewritten as a nonlinear state equation, and then the Jacobians with respect to the state variables and input variables were calculated at the vertical flight conditions. The length of the arm suspending the payload from the drone was varied between 0 meters and 1 meter, while the payload mass was held constant at 30 kg. Then, the mass of the payload was varied between 0 kg to 50 kg while the payload arm length was held constant at 0.2 meters. The eigenvalues of the A matrix are then plotted in Fig. 2.6 for the varying payload arm length and in Fig.2.7 for the varying payload masses.

It is important to note that for the case of vertical flight, A has five negative, nonzero-eigenvalues and seven zero-eigenvalues. The stability of the zero-eigenvalues cannot be predicted because they are neither stable nor unstable. The non-zero eigenvalues can be examined to determine if they will become positive.

From Fig. 2.6, it is clear that as the length of the arm suspending the payload increases,

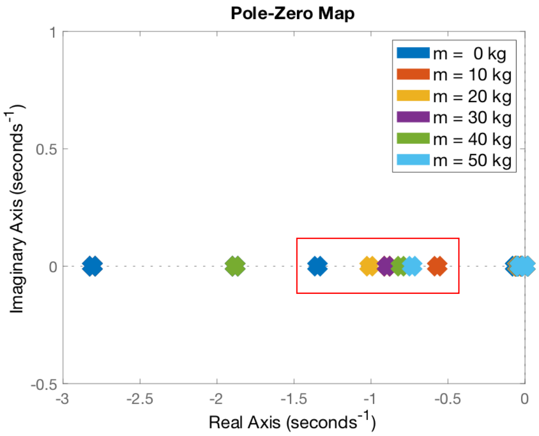


Figure 2.7: Poles of system matrix with $L_{pl} = 0.2$m as payload mass is varied.

the boxed eigenvalues become more negative. However, this does not guarantee the system's stability due to the seven zero-eigenvalues which may contribute to the nonlinear system's instability. One possible explanation for the negative eigenvalues becoming more negative is that the increased distance between the payload and drone increases the inertia of the system, which in turn makes the five nonzero-eigenvalues more negative.

In Fig. 2.7, the boxed eigenvalues are changing with the changing payload mass. The eigenvalues show a trend towards being less negative as the payload mass is increased. Despite this trend towards being less negative, the eigenvalues still remain negative when the payload mass is at a maximum of $m_{pl} = 50$kg. This trend is the opposite of the trend seen in Fig. 2.6, despite the fact that an increase in L_{pl} and an increase in m_{pl} both result in the center of gravity of the system moving further away from the drone's geometric center. This contradiction implies that the results of the linear stability analysis cannot be directly applied to the nonlinear system.

To determine what states will be affected by the zero-eigenvalues, take as an example a drone-payload system with $m_{pl} = 30$ kg and $L_{pl} = 1.0$ m. The eigenvectors corresponding to the zero-eigenvalues for the described system are shown in Tab. 2.3 with the order of

Table 2.3: Eigenvectors for all zero-eigenvalues

State	Eigenvectors for Zero-Eigenvalues						
x	0	1	0	0	0	0	0
y	0	0	1	0	0	0	0
z	0	0	0	1	0	0	0
u	0	0	0	0	0	0	0
v	0	0	0	0	0	0	0
w	0	0	0	0	0	0	0
p	0	0	0	0	0	0	0
q	0	0	0	0	0	0	0
r	1	0	0	0	0	0	0
ϕ	0	0	0	0	1	0	0
θ	0	0	0	0	0	1	0
ψ	0	0	0	0	0	0	1

the states in the state vector shown in the left column. Examining Tab. 2.3 reveals that the eigenvectors for the zero eigenvalues correspond to the states x, y, z, r, ϕ, θ, and ψ which means that these states are marginally stable in the linearized system. The nonzero eigenvalues are: -0.0421, -0.0421, -0.896, -1.882, and -2.809. The eigenvectors for these eigenvalues correspond to the remaining five states (u, v, w, p, and q), which means that these states are stable in the linearized system. However, due to the large number of zero eigenvalues in the system, the stability of the nonlinear system cannot be predicted based on studying the eigenvalues of the linearized system. A study of the behavior of the nonlinear system is required to fully understand the stability of the drone-payload system.

2.3.2 Payload Effects in the Nonlinear System

In order to fully understand how the instability caused by the presence of a suspended payload affects the flight dynamics, simulations of the nonlinear system were also completed. The same parameters used in the linearized model are used here for consistency. The results of the nonlinear simulation can then be compared to the linear system stability. Like previously, one set of simulations was completed when the payload arm length was varied between 0 meters and 1 meter while the payload mass was kept constant at 30 kg. The other was completed when the payload mass was varied between 0 kg and 50 kg while the payload arm length was kept constant at 0.2 meters. The system is subjected to a one-second gust of wind at the start of the simulation to induce different initial conditions on the system, and then the flight is simulated for an additional nine seconds to study the effect of the induced initial condition over time. The case without a gust of wind was also simulated as a comparison to ensure that the deviation due to the different initial conditions is significant. Without the gust, the presence of the suspended payload will still cause the drone to deviate slightly from the exact vertical path. In addition, both the case with and without gust was simulated for a system where the payload is instead lumped at the drone's geometric center since this case should not deviate from the perfect vertical trajectory. The quadrature sum of the difference between the suspended and lumped payload is then computed for both the case with wind and without wind. The results of these simulations are presented and discussed in the next section.

Simulation Results

Figures 2.8 and 2.9 show the quadrature sum error between the simulated flight where the payload is suspended and the simulated flight where the payload is lumped into the geometric center. This error is calculated from the position of the drone as follows

$$\epsilon = \sqrt{(x_s^e - x_l^e)^2 + (y_s^e - y_l^e)^2 + (z_s^e - z_l^e)^2} \tag{2.27}$$

where the position of the drone with the suspended payload is given by $\begin{bmatrix} x_s^e & y_s^e & z_s^e \end{bmatrix}$ and the

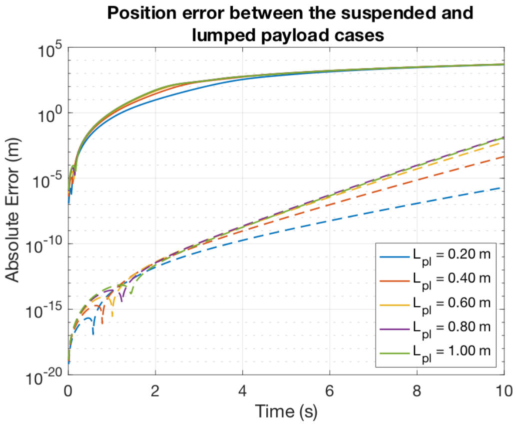


Figure 2.8: Deviation of position from expected with $m_{pl} = 30$kg as L_{pl} is varied. Wind case is shown in solid lines and no-wind case is shown in dotted lines.

position of the drone with the lumped payload is given by $\begin{bmatrix} x_l^e & y_l^e & z_l^e \end{bmatrix}$. In Figs. 2.8 and 2.9 the dotted line shows the error ϵ for the case without wind. The solid line (after one second) shows the error ϵ for the case when there are induced initial conditions due to a sudden wind disturbance that lasts for one second and has magnitude $\mathbf{v}_w^e = \begin{bmatrix} 5 & 0 & 1 \end{bmatrix}$ m/s.

Figure 2.8 shows that for the simulations with wind-induced initial conditions, the flight of a drone with a suspended payload deviates significantly from the flight of a drone with the payload lumped at the drone's geometric center. The deviation is on the order of $10^3 - 10^4$ meters. This can be compared to the simulations without wind. Without the gust, the deviation is significantly smaller (on the order of $10^{-6} - 10^{-2}$ meters). Since there is a minimum difference of five orders of magnitude between these two errors, the results suggest that the suspended payload makes a significant difference on how the drone behaves when subjected to a major disturbance, such as a sudden gust of wind.

Figure 2.9 shows the same trend as Fig. 2.8 where the presence of a payload, no matter its mass, has a huge effect on how the drone reacts to a sudden disturbance. When there is no wind, the error between the suspended and lumped payload cases is very small (on the order of $10^{-8} - 10^{-4}$ meters). Compared to the case when there is wind, there is a seven orders of magnitude difference between the errors. Again, the results indicate that presence

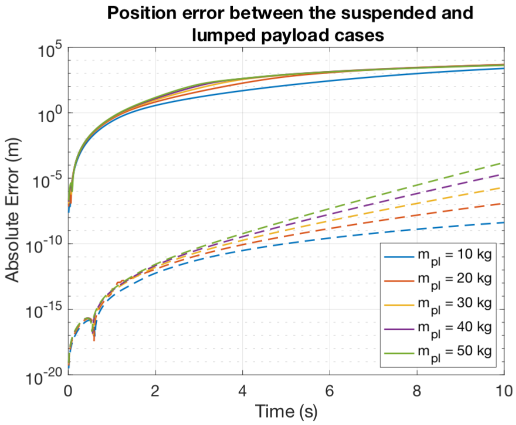


Figure 2.9: Deviation of flight position from expected value with $L_{pl} = 0.2$m as m_{pl} is varied. Wind case is shown in solid lines and no-wind case is shown in dotted lines.

of a suspended payload significantly affects the drone's behavior when sujected to a sudden gust of wind.

Since the error in positions has been confirmed to be significant, the trajectories of the drone can now be examined to determine if the payload does make the system unstable. To do this, both the payload mass and payload arm length are varied as in the previous simulations. Based on Figs. 2.10 and 2.11, the difference in the trajectories between the case without wind (dotted lines) and the case with a sudden gust (solid line) confirms the previous conclusion that the presence of the payload makes the system extremely unstable. The trajectories of the case without wind may be difficult to see since they closely follow the trajectory for $L_{pl} = 0$ meters in Fig. 2.10 and the trajectory for $m_{pl} = 0$kg in Fig. 2.11. In both Figs. 2.10 and 2.11, the cases with wind immediately deviate from the expected, purely vertical path due to the presence of a payload. From Fig. 2.10, it is clear that the longer payload arms make the drone deviate significantly from the case when $L_{pl} = 0$ meters (the payload is not suspended). Similarly from Fig. 2.11, it is apparent that the heavier the payload, the more the drone's flight will deviate from the case when $m_{pl} = 0$ kg (there is no payload present). Since the drone's reaction to a disturbance varies greatly between the case when there is a suspended payload present and when there is not a suspended payload, it can

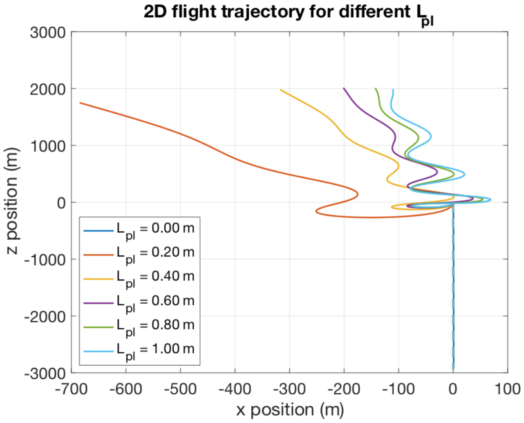


Figure 2.10: Deviation of flight position from expected value with $m_{pl} = 30$kg as payload arm length is varied. Wind case is shown in solid lines and no-wind case is shown in dotted lines.

be concluded that the presence of a suspended payload makes the drone inherently unstable. This conclusion fills in the gap left by the linear stability analysis from which no conclusion about the stability of the nonlinear system could be made due to the seven zero-eigenvalues.

2.3.3 Conclusions

The goal of this section is to quantify how the payload affects the stability of the system and the severity of the instability it causes when the system is subjected to an external disturbance. The linear stability analysis reveals that when the drone model is linearized around vertical flight, the system only has negative and zero-eigenvalues which indicate that the linearized system is at least marginally stable. This is true no matter how L_{pl} and m_{pl} are varied. Because the system has seven zero-eigenvalues, however, the stability of the nonlinear system cannot be concluded directly from the system's linear stability, and so the stability of the nonlinear system must be examined directly. An analysis for nonlinear stability could not be done because there is none that can be applied. Since linear stability does not imply nonlinear stability, numerical simulations must be used to show the stability of the open-loop system. In the future, it is possible that a Lyapunov function could be used to prove that the

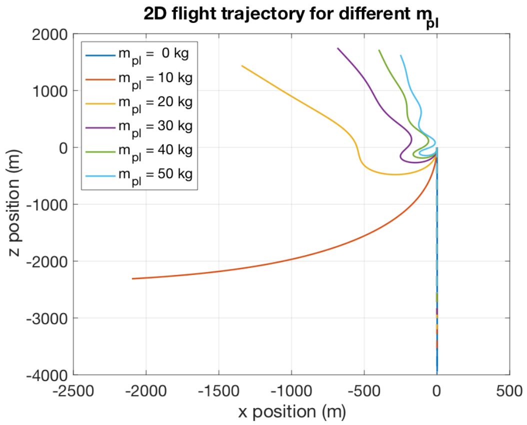


Figure 2.11: Deviation of flight position from expected value with $L_{pl} = 0.2$m as payload arm length is varied. Wind case is shown in solid lines and no-wind case is shown in dotted lines.

system is unstable. For now, analysis of the open-loop simulations results is the extent to which the nonlinear stability analysis can be completed. The nonlinear simulations confirms the initial assumption that the payload makes the drone more unstable in the presence of wind. The simulations show that no matter how L_{pl} and m_{pl} are varied, the drone will become unstable when subjected to a wind-induced initial condition. From the plots, the severity of the instability can be quantified to be on the order of five to seven orders of magnitude difference between the expected paths of the simulated flights without wind and the actual paths of simulated flights with wind. An examination of the trajectories results in the same conclusion that the payload causes the drone to severely deviate from the expected flight trajectory when the drone with a lumped payload is subjected to wind. Thus, it can be concluded that the presence of a payload will always cause the drone to become unstable and that the severity of the instability is very large.

This knowledge must be taken into consideration when designing a high-performance to carry heavy payloads. High-performance drones, such as the previously-suggested design, are large and have many propellers, so there is a high risk of danger should these drones become unstable during takeoff or landing. By studying both the linear and nonlinear stability of the system, the payload parameters for a high-performance drone can be optimized such that the instability of the system is minimized. By comparing the linear stability analysis and nonlinear simulations results, the effect of the payload was determined to be significant and not captured by the linearized model. This suggests that a linearized model is not sufficient to capture the dynamics of a static payload. Thus, future work on high-performance drones with heavy payloads must use the nonlinear model since payload dynamics are not properly captured in the linear model. The results presented here provide one method through which the stability of the proposed high-performance drone design can be analyzed and quantified.

2.4 Aim 1b: Determine the significance of aerodynamic models for modeling the rotor blades

The following sections are excerpted from [30].

To explore the impact of high-fidelity modeling, this section examines the effect of using blade element theory to model the aerodynamics of the rotor blades together with a multi-body dynamical analysis of the drone. Two identical sets of simulations are performed in order to determine how the lumped-blade model that most drone models use differs from using a blade element theory model that takes into account the variation in induced velocity and blade geometry over the length of the blade as well as computational fluid dynamics (CFD) simulation results.

2.4.1 Comparing Rotor Blade Models

To determine how rotor blade model affects the resulting trajectory, and consequently the energy consumed, basic trajectories are generated using the same rotor speed inputs for the two rotor blade aerodynamic models. Simulations for four basic motions are completed for each aerodynamic model. The four basic motions are

1. Pure vertical motion starting at some initial height and steadily accelerating upward,
2. Small angle rotation in the roll direction starting and ending at 0°,

3. Small angle rotation in the pitch direction starting and ending at 0°, and
4. Small angle rotation in the yaw direction starting and ending at 0°

The second through fourth basic motions are all superimposed over the first basic motion. The resulting trajectories for the two models are then compared to determine how the trajectory (and consequently the calculated rotor thrust and torque) varies between the two models.

Results

Figure 2.12 shows the rotor speed inputs that is used for both the LB and BET models and the resulting trajectories for each simulation. As previously stated in Section 2.5.1, these rotor speed inputs are formulated so that, when using the BET model, the drone is able to perform a complete motion in each of the roll, pitch, yaw, and vertical directions. For the roll, pitch, and yaw motions, the pairs of rotors indicated previously have their speeds increased or decreased in order to produce an overall torque on the drone that causes angular motion in all three directions.

From the plots in Fig. 2.12, it is clear that the chosen aerodynamic model greatly affects the trajectory of all motions. Specifically, yaw and vertical motions show the largest deviation in final trajectory. To quantify the difference between the aerodynamic models, the maximum percent difference between the trajectories produced by each model is calculated as follows.

$$\%\text{ difference} = \frac{\max|x_{BET}| - \max|x_{LB}|}{\max|x_{BET}|} \times 100\% \tag{2.28}$$

where x_{BET} is the state variable for the BET trajectory and x_{LB} is the state variable for the LB trajectory. For example, when calculating the percent difference for the vertical trajectory, $x_{BET} = z_{BET}$ and $x_{LB} = z_{LB}$. The percent difference between the maximum displacements of each model are shown in Tab. 2.4.

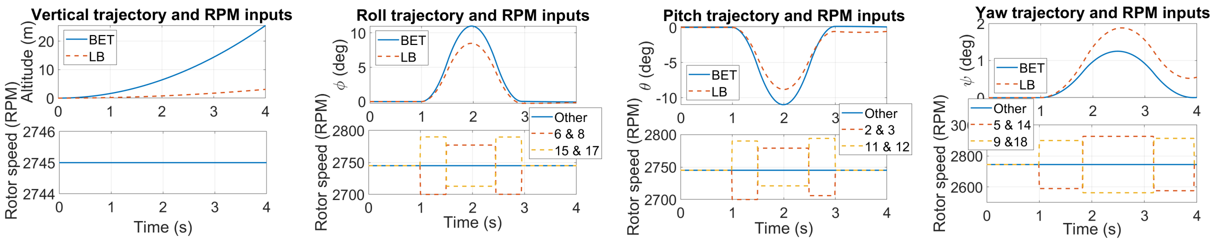


Figure 2.12: Plots comparing the BET and LB model simulation results for the four basic motions with their respective rotor speed inputs

Table 2.4: Summary of differences in maximum displacement between BET and LB models

Trajectory	Vertical	Roll	Pitch	Yaw
Difference (%)	87.9	22.9	19.8	52.5

As shown in Tab. 2.4, there is a large difference between the resulting trajectories for all motions. The difference is particularly prominent for vertical and yaw motions. To determine why the large difference in these two trajectories exist, the coefficients of lift and drag must be examined and compared to the angle ϕ of the rotor blade. The coefficients of lift and drag are shown in Figs. 2.13 and 2.14, respectively, and the variation of the angle ϕ that determines the direction of the lift and drag forces relative to the plane of rotation of the blade is shown in Fig. 2.15. The coefficients are plotted as a function of position on the rotor blade for a blade that is spinning at 2750 RPM and has axial velocities of 0, 5, and 10 m/s. The rotor speed is chosen because it is the average of the rotor speeds used in the simulations, and the axial velocities are chosen to span the range of axial velocities present in the simulations. In Figs. 2.13 and 2.14, the solid lines indicate the average value of C_L and C_D at each of the indicated velocities and are the values for C_L and C_D used in the LB model.

Recall that yaw motion is initiated by creating a difference in torques between the rotors.

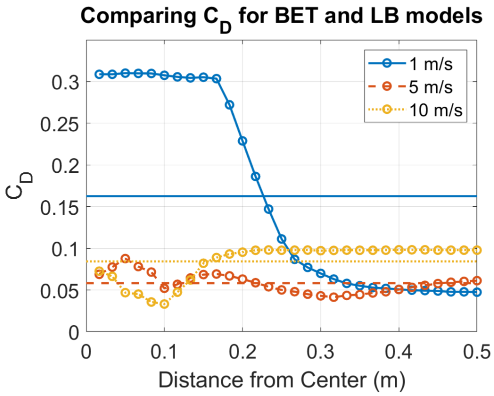


Figure 2.13: Plot showing variation of C_D along the rotor blade length for three different axial velocities. Solid lines indicate the average values for C_D used in the LB model

The large difference in the yaw trajectory indicates that the total torque for each rotor varies significantly when calculated using the BET model as opposed to the LB model. As such the coefficient of drag must be examined. Only the coefficient of drag needs to be examined since the angle ϕ is very small along the entire length of the blade for the maximum axial velocity achieved (see Fig. 2.15) meaning that drag contributes most to the torque. Figure 2.13 shows how the drag coefficient varies for different axial velocities along the length of the rotor blade. We see that for low axial velocities, the coefficient of drag varies greatly. This means that the total drag experienced by the blade is significantly different when using the BET model as compared to the LB model when calculating aerodynamic forces for these low velocities. Since the drone starts at zero velocity at the beginning of the simulation, then this would cause an early deviation of the calculated torque values. This means that the trajectory produced by the BET model will differ from the LB model early on in the simulation since the coefficient of drag varies greatly along the rotor blade for low velocities. Thus, even though the coefficient of drag is more constant at higher velocities, the initial large variation causes the 52.5% difference in the trajectories.

By comparison, the lift contributes the most to the total thrust calculated due to the small angle ϕ of the rotor blade (see Fig. 2.15). From Fig. 2.14, it is clear that at high velocities,

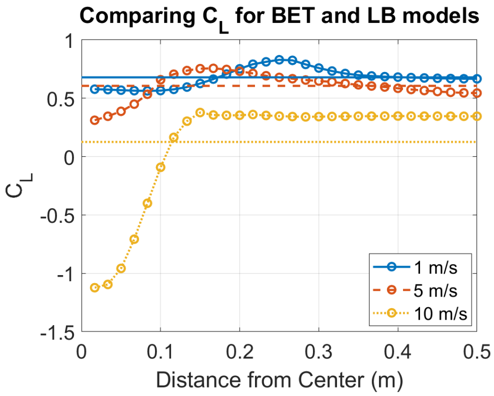


Figure 2.14: Plot showing variation of C_L along the rotor blade length for three different axial velocities. Solid lines indicate the average values for C_L used in the LB model

Table 2.5: Summary of difference in energy consumption between BET and LB Models

Trajectory	Vertical	Roll	Pitch	Yaw
Difference without Efficiency Curve (%)	36.35	36.35	36.35	36.34

the coefficient of lift varies significantly. Thus, as the velocity increases, the calculated lift will be significantly lower for the LB model as compared to the BET model. This means that as the vertical velocity of the drone increases, the LB model will have a lower total lift resulting in deviation of the trajectory later in the simulation. This matches the results shown in Fig. 2.12 where the LB model achieves 87.9% less altitude than the BET model since the deviations largely occur later in the simulation. The results from both the yaw and vertical motion simulations show that using a LB approach will yield significantly different results than using the BET model because the coefficients of lift and drag are dependent on the axial velocity and location of the element.

In addition to changing the simulated trajectory, the different models will also yield different calculated energies. Table 2.5 shows the difference in total energy calculated (without

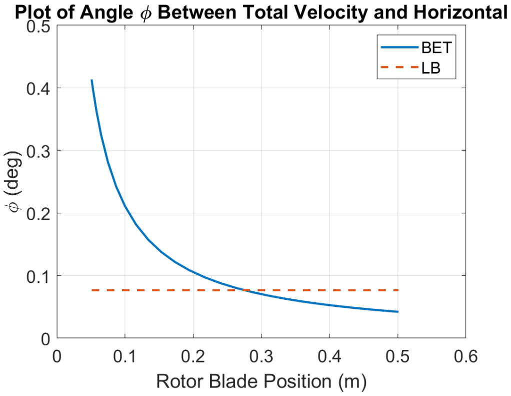


Figure 2.15: Plot of ϕ as a function of position along the length of the blade for the max achieved axial velocity of 12.9 m/s for the BET model and 1.6 m/s for the LB model

any motor efficiency curve) given by the following equation

$$\%\text{ difference} = \frac{\max(E_{LB}) - \max(E_{BET})}{\max(E_{BET})} \times 100\% \tag{2.29}$$

where E_{BET} is the energy calculated for the trajectories calculated using the BET model and E_{LB} is the energy calculated for the trajectories calculated using the LB model. As seen in Tab. 2.5, the energy calculated using the LB model is consistently 36.35% higher than the energy calculated by using the BET model. The consistent energy difference no matter the trajectory flown shows that the energy is due to the different aerodynamic model used. If the BET model is more accurate, then the 36.35% overestimation of the energy consumption by the LB model may result in the drone carrying more energy than necessary. The increased amount of energy carried by the drone would reduce the additional payload weight the drone could carry.

2.4.2 Discussions

The effect of the rotor blade aerodynamic model on thrust and torque is examined by simulating the roll, pitch, yaw, and vertical motions for a BET model and a LB model. The resulting trajectories of the BET and LB models differ drastically despite the same input being given to them. Of the four simple trajectories simulated, the yaw and vertical motions show the most significant differences of 87.9% and 52.5%, respectively. The results from the BET and LB comparison study indicate that the chosen aerodynamic model changes the result of a flight simulation drastically. Thus it can be concluded that the BET models take into account aerodynamic effects that are ignored when using a LB model and should be used in flight simulations where accuracy is important. In addition to a difference in the trajectory, the energy consumed during flight is also different under the different models as shown in Tab. 2.5. Therefore, the choice of aerodynamic model is critical in cases where energy consumption needs to be accurately calculated, such as in trajectory planning.

2.4.3 Conclusions

This section demonstrates the importance of using the blade element theory model when modeling the aerodynamics of the rotors of multi-rotor drones. As the drone becomes larger, the blades are no longer accurately represented as a single lumped element as done in many drone models. The different velocities experienced by different sections of the blade are ignored in the lumped blade model which changes the calculated thrust and torque on the blade. This difference in the thrust and torque change the resulting trajectory and, as a result, the accuracy of the simulated trajectory could be compromised. In addition to this, the accuracy of the calculated energy consumption may also be compromised since the BET and LB models calculate different energy usages for the same rotor speed inputs.

2.5 Aim 1c: Determine the significance of rotor groupings on energy efficiency

The following sections are excerpted from [30].

To explore the effect of rotor grouping, this section compares the energy consumption of two different rotor groupings of the 18-rotor drone. Rotor speeds in these two groupings are different, but they are chosen in such a way that the drone flies the same trajectory. Therefore, a direct comparison of the energy consumption is possible since the same drone model is used for both groupings. First, the energy consumption for three basic motions

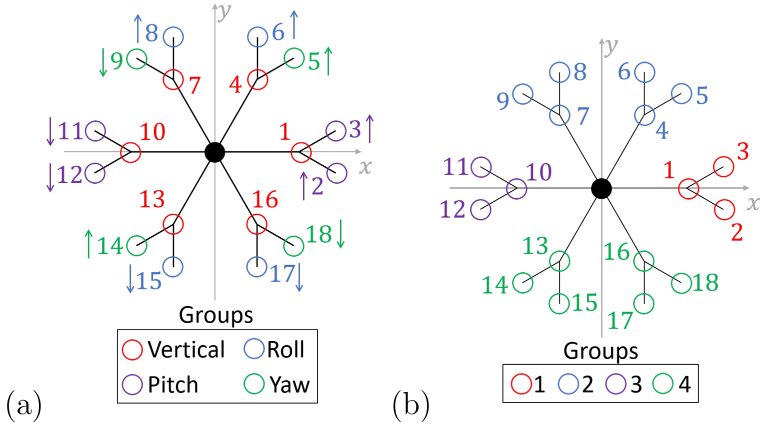


Figure 2.16: (a) Grouping 1 and (b) Grouping 2 for 18-rotor drone

(i.e., roll, pitch, and yaw) is compared. Then a complex trajectory is examined to determine if there is an energy compounding effect due to the combination of basic motions to form a complex trajectory.

2.5.1 Rotor Groupings

The concept of rotor groupings must also be discussed. For multi-rotor drones with more than 4 rotors, grouping the rotors may make it easier to apply control strategies for specific motions. Rather than allowing a controller to randomly choose rotors to adjust to follow a specified trajectory, rotor groupings allow the designer to choose how and which rotors can be adjusted to achieve a specified trajectory. The groupings can be chosen in order to minimize energy consumption or any other value as desired. The two groupings described here are just two possible ways to group the rotors. They are used to demonstrate how grouping choice affects energy consumption during flight [29].

Grouping 1

The first grouping is shown in Fig. 2.16(a). For this grouping, each group handles one of each of the roll, pitch, yaw, and vertical motions. The colored up and down arrows for the roll, pitch, and yaw groups show how the rotor inputs are adjusted in pairs in order to initiate a motion in those directions.

For example, in the vertical group all rotors are controlled with one input. This means that an increase in rotor speed to all rotors will result in the drone accelerating upward and a decrease in the rotor speed input to all rotors will result in the drone decelerating. As a result, any change in the input will cause the drone to accelerate in the vertical direction with no change to its angular orientation. By comparison, a roll motion is initiated by increasing the input to rotors 6 and 8 and decreasing the input to rotors 15 and 17 by the same amount as shown by the four blue arrows in Fig. 2.16(a). The difference in total thrust between these two pairs of rotors causes the drone to move in the roll direction only. The pitch and yaw groups are controlled in a similar manner to the roll group with the arrows indicating which pairs of rotors are controlled as a single input and how they are adjusted. For example, in order to initiate a pitch motion, rotors 2 and 3 are increased by the same amount that rotors 11 and 12 are decreased by in order to make the drone rotate in the pitch direction. Similarly for the yaw motion, inputs to rotors 5 and 14 are increased by the same amount the inputs to rotors 9 and 18 are decreased by. These ways in which vertical, roll, pitch, and yaw motions are initiated is fixed, so if a control system were to be imposed on the system, the chosen rotors would be used to initiate corrections in each direction independently.

Grouping 2

The second grouping is shown in Fig. 2.16(b). The 18 rotors are divided into 4 groups that allow the drone to behave like a quadcopter. In order for each of the groups to produce the same total thrust, the rotor inputs to Groups 1 and 3 must be different than the rotor inputs to Groups 2 and 4. The base rotor speed for Groups 2 and 4 must be lower than the base rotor speed for Groups 1 and 3 since Groups 2 and 4 have twice the number of propellers. Each of these groups will be controlled using a single input like a quadcopter.

In order for the 4 groups to act as 4 “rotors” they each must have a base rotor speed from which they are adjusted to initiate each of the four actions. The base rotor speed is

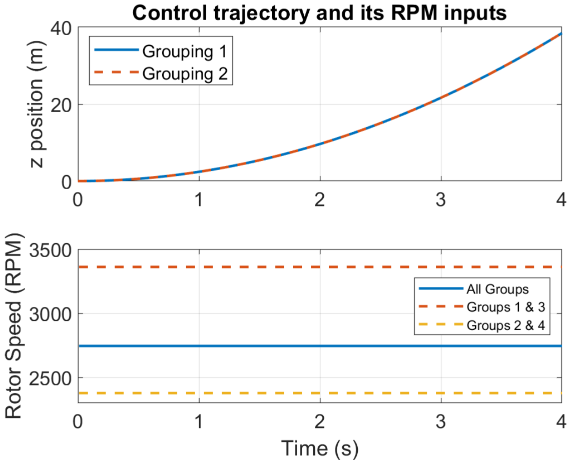


Figure 2.17: Trajectory used to determine rotor speed inputs that produce the same total lift for both groupings (top) and the base rotor speeds (bottom)

found by using a control upward trajectories for both groupings. This trajectory is used to ensure that the total lift of both groupings is the same when all rotors are held constant at a base rotor speed value. A trajectory is chosen for Grouping 1. Then the rotor speeds for Grouping 2 are calibrated such that both groupings will follow the same trajectory. This results in Grouping 1 having a base rotor speed of 2395 RPM and Grouping 2 having a base rotor speed of 2933 RPM for Groups 1 and 3 and a base rotor speed of 2074 RPM for Groups 2 and 4. The reference trajectory produced and the resulting rotor speed inputs for each grouping are plotted in Fig. 2.17. The solid lines indicate the trajectory and rotor speeds for Grouping 1 while the dotted lines in both plots indicate the trajectory and rotor speeds for Grouping 2.

In order to facilitate each of the roll, pitch, yaw, and vertical motions, the four groups will be controlled in pairs. Groups 2 and 4 will be used to initiate roll motions, and Groups 1 and 3 will be used to initial pitch motions. For yaw motions, Groups 1 and 3 must be given the same input and Groups 2 and 4 must be given the same input in order to create a difference in torques. The difference in torques produced by each pair of Groups will cause the drone to move in the yaw direction. Finally, all groups will be controlled together to control vertical motion. Like for Grouping 1, time histories of the inputs demonstrating these motions will be given in later sections.

2.5.2 Comparing Rotor Groupings for Simple Trajectories

As shown in the previous section, the aerodynamic model chosen affects the trajectory greatly, resulting in a similar effect on the energy calculation. The rotor groupings also play a role in the energy consumption since depending on how the rotors are controlled, the amount of energy required to fly a single trajectory can change. With more than 4 rotors, there are many ways in which the drone rotors can be controlled. This section examines two rotor groupings that are used to simulate roll, pitch, and yaw motions. Like in the previous set of simulations, the reference drone model is used for both rotor groupings and all three motions are simulated for both groupings in order to compare their performance. The total energy consumption for each flight is calculated from the rotor power which is found in Eqn. (2.24).

The rotor speed inputs for the two groupings to complete the remaining three motions are determined by using the rotor speeds in Fig. 2.17 as the central value about which to vary the motor speeds for each group. This means that for all rotors in Grouping 1, the motor speeds to accomplish these motions are varied about 2395 RPM. For Grouping 2, the motor speeds for groups 1 and 3 are varied about 2933 RPM, and the motor speeds for groups 2 and 4 are varied about 2074 RPM.

Results

The roll, pitch, and yaw motions are constructed as shown in Fig. 2.18 with the rotor speed inputs for each trajectory plotted immediately below them. In all three plots only the rotor speed inputs of the relevant groups are plotted. All rotor inputs that are not plotted are equal to the inputs from the control trajectory as shown in Fig. 2.17. For each of the motions, the inputs are adjusted until close agreement of the trajectories is reached so the energy required to fly each trajectory can be directly compared.

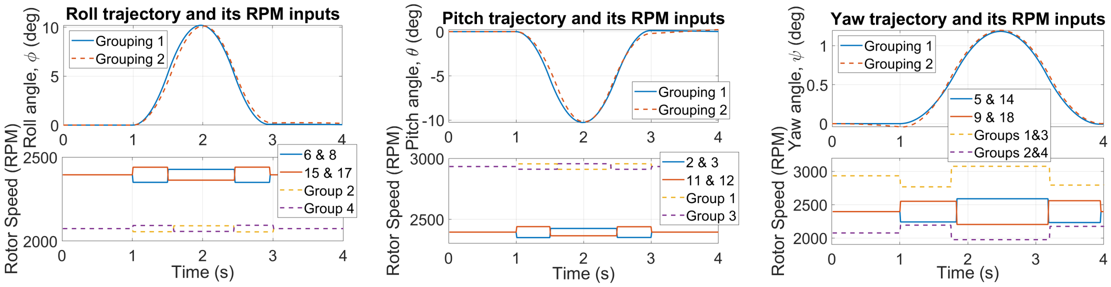


Figure 2.18: Plots showing the trajectories simulated for two different groupings. From left to right, the plots show the roll, pitch, and yaw trajectories with their respective rotor speed inputs plotted below

The percent difference in energy consumption is given in Tab. 2.6. The percent difference is calculated as

$$\% \text{ energy difference} = \frac{\max(E_1) - \max(E_2)}{\max(E_1)} \times 100\% \tag{2.30}$$

where E_1 is the energy consumed during flight by Grouping 1 and E_2 is the energy consumed during flight by Grouping 2. For all motions, the different rotor groupings produce a 4-5% difference in energy consumed.

Before any conclusions are made, the cause of the energy difference is explored. The difference in the trajectories and the difference in the energies can be compared to determine if the energy difference comes from a difference in trajectory or the rotor groupings. Figure 2.19 shows how the energy and trajectory vary over the entire simulation. The variation in the trajectory is calculated over time as follows for each time step.

$$\% \text{ trajectory difference}(t) = \frac{x_2(t) - x_1(t)}{x_1(t)} \times 100\% \tag{2.31}$$

Similarly, the difference in energy over time is calculated as follows for each time step.

$$\% \text{ energy difference}(t) = \frac{E_2(t) - E_1(t)}{E_1(t)} \times 100\% \tag{2.32}$$

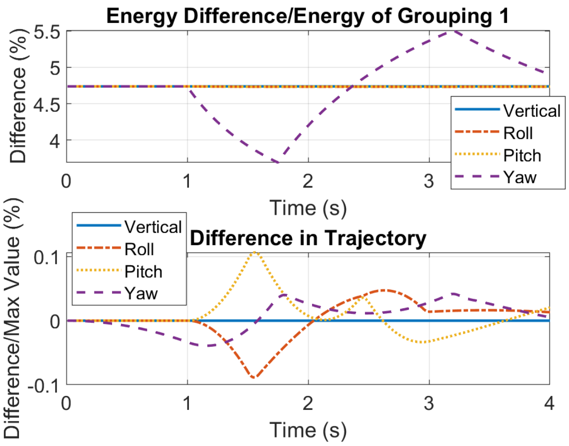


Figure 2.19: Plots showing the energy difference (top) and trajectory difference (bottom) between Grouping 1 and Grouping 2

In general for these small motions, the percentage differences in the trajectory do not exceed 0.1% while the energy difference maintains a fairly constant 4.73% difference except in the case of yaw which oscillates between 3.69% and 5.51%. This confirms that the difference in energy does not arise due to a difference in the trajectory, but the rotor groupings.

In addition to comparing the energy and trajectory differences, the different rotor speed inputs are also compared. As seen in Fig. 2.18, the different groupings have very different rotor speeds. The different rotor speeds mean that the total power used by each rotor will be different (see Eqn. (2.24)). Thus, the energy difference likely is due to the different rotor speeds since the rotor speeds affects the energy calculated for each grouping and trajectory. In addition, all simulations done were completed for the open loop drone system meaning that there is no additional energy consumption for either grouping that is due to corrections by a control system.

From this it can be concluded that the rotor grouping has an effect on the total energy required to fly the same trajectory. By adjusting which combination of rotors are used for each motion, the total energy consumption can be increased or decreased. This implies that each type of rotor grouping has an inherent base energy cost. This suggests that when doing

Table 2.6: Summary of difference in energy consumption between groupings 1 and 2

Trajectory	Roll	Pitch	Yaw
Difference without Efficiency Curve (%)	4.73	4.73	4.90

trajectory planning for minimum energy consumption, the rotor groupings should also be optimized for energy consumption. Further discussion of the implications of this will be given after examining the effects of rotor grouping on a complex trajectory.

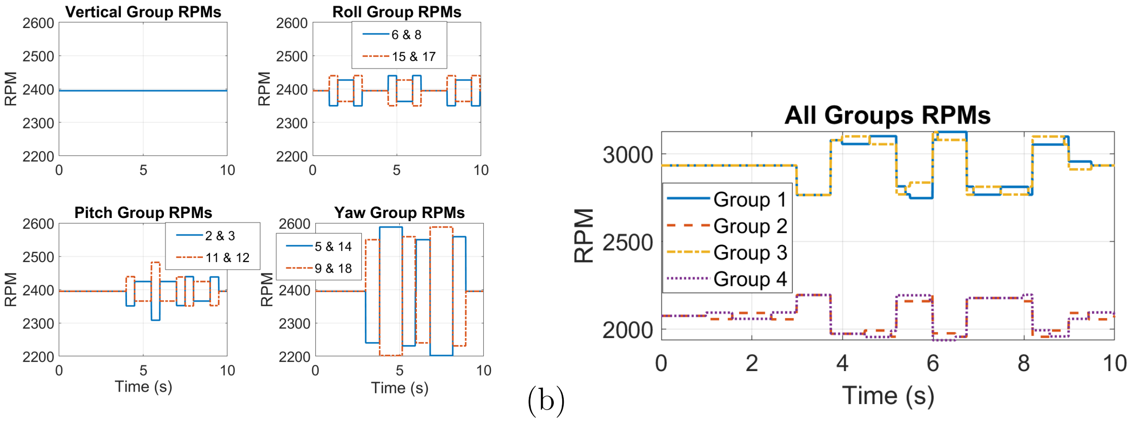
2.5.3 Energy Comparison for Complex Trajectories

In Sec. 2.5.2, the energy required to fly three simple trajectories for two different rotor groupings are examined. These simple trajectories ignore possible compounding effects that could occur during more complex trajectories. Therefore, a complex trajectory must be analyzed in order to determine if the energy difference between the two groupings compounds with complexity. The chosen trajectory incorporates all of the basic motions while flying upward. Like with the simple trajectories, the total energy is calculated using Eqn. (2.30).

Results

The complex trajectory is achieved by summing together multiple instances of the simple motions found in Fig. 2.18 starting at various times during the simulation. The rotor inputs to achieve the complex trajectory are given in Fig. 2.20. For Grouping 1, the two different inputs for pairs of rotors in each group (see Fig. 2.16) are shown in separate plots in Fig. 2.20(a). For Grouping 2, the different inputs to each group of rotors are shown in different colors in Fig. 2.20(b).

The final trajectory achieved for each rotor grouping is shown in Fig. 2.21. For the majority of the simulation time, the two groupings are able to maintain very similar tra-



(a) (b)

Figure 2.20: Plots showing inputs for (a) Grouping 1 and (b) Grouping 2

jectories. Some deviation comes at the end due to an accumulation of differences in the basic motions between the groupings as can be seen in Fig. 2.18. Plots of the individual translational and angular positions are given in Fig. 2.22. The plots here show that the two groupings maintain very close agreement throughout the entire simulation time. The final percent difference in energy consumption between the two groupings is calculated as 9.26% according to Eqn. (2.30). This is almost double the difference in energy seen between the two groupings for basic motions. This implies that there could be energy compounding effects present in some complicated trajectories since the complex trajectory shows a 95.8% greater energy difference between groups than the basic trajectories did. Like in Sec. 2.5.2, the simulations were completed as open loop simulations so there are no other factors such as a control system present.

To determine if the energy compounding is due to the combination of basic motions or the differences between the final trajectories of each grouping, the difference between the two trajectories and the difference in the energies is plotted in Fig. 2.23. Equation (2.32) is used to calculate the percent energy difference plotted. To calculate the total difference in trajectory, the magnitude of the difference between the positions is divided by the magnitude of the position for grouping 1 as shown in the equation below.

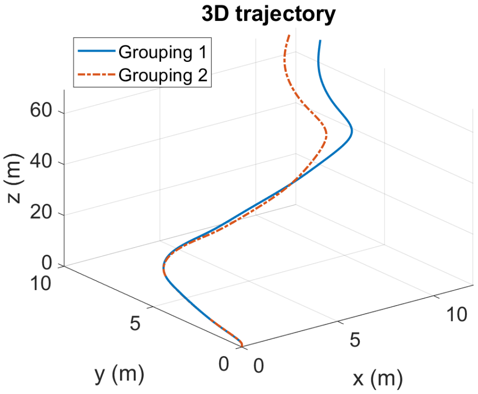


Figure 2.21: Plot showing a complex trajectory as flown by a drone with two different rotor groupings

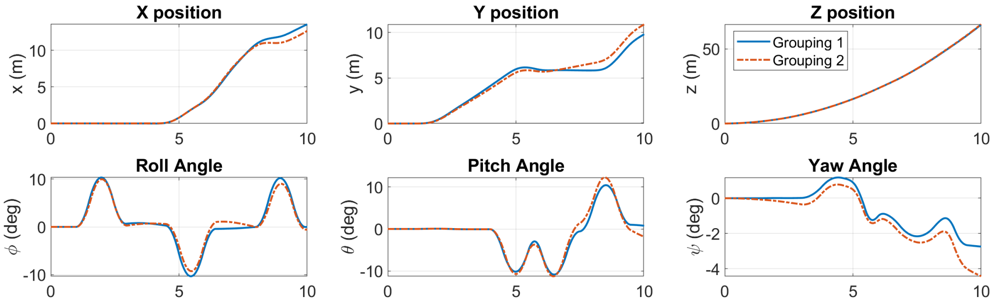


Figure 2.22: Plots showing the components of position (top) and orientation (bottom) of the drone during the complex trajectory for both groupings

$$\%\ \text{diff} \ = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \times 100\% \tag{2.33}$$

As seen in Fig. 2.23, the energy difference between the groups starts at 4.73% and increases over the course of the trajectory to 9.26%. The energy difference appears to increase after the trajectory difference reaches its maximum value of 2.7% at 4 seconds. The slope of the energy difference also increases again after the second increase in trajectory difference at 8.5 seconds. This implies that the increased energy difference is due to the increased differences in the trajectory. This is corroborated by examining the trajectory in Fig. 2.22

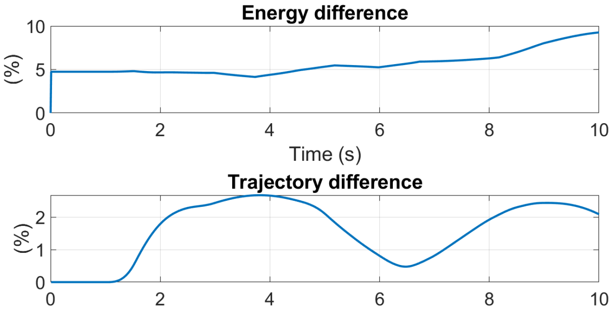


Figure 2.23: Plots showing the energy difference (top) and trajectory difference (bottom) between Grouping 1 and Grouping 2 for a complex trajectory

where the trajectory deviates the most after about 8 seconds. This larger deviation in the trajectory is correlated to the increased slope of the energy difference leading to the 9.26% final energy difference. This means that the increased energy difference of the complex trajectory is likely due to the difference in the final trajectories and not energy compounding effects.

2.5.4 Discussions

The effect of rotor groupings on drones with more than four rotors was examined. The results of this study show that different rotor groupings require different amounts of energy to fly the same trajectories. Similar to the comparison of the aerodynamic models, simulations of the roll, pitch, and yaw motions are used to show that one grouping is more energy efficient than the other. Grouping 1 requires 4.7-4.9% less energy than Grouping 2 to fly almost the same trajectory. Examining how the energy and trajectory differences change over the course of the simulations in Fig. 2.19 shows that the energy difference is not due to the difference in the trajectories. A quick examination of the rotor speeds for each grouping reveals that the two groupings have vastly different rotor speeds for each rotor. Since rotor speeds affect the total power calculation, then the energy difference likely comes from the difference in rotor speeds that different rotor groupings may have.

Then, the effect of rotor groupings on complex trajectories was examined. Grouping 1 required 9.26% less energy to fly a similar complex trajectory to Grouping 2. Further investigation revealed that the energy difference is likely due to the larger differences in the trajectory. The increases in energy difference correlate to the increases in trajectory difference as seen in Fig. 2.23. This means that there is likely not energy compounding effects. Instead the increased energy difference is due to a difference in trajectories. Even though it cannot be concluded that energy compounding effects exist from the specific trajectory studied in this aim, there may exist other trajectories for which there are energy compounding effects. Even though energy compounding effects cannot be shown to exist in this aim, it is still important to choose rotor groupings that have a more optimal energy consumption when energy consumption needs to be optimized.

The difference in energy between the rotor groupings can also be exacerbated when the loop is closed with a controller. This is because control strategies inherently have some minimum energy cost due to how they correct deviations in the trajectory. This energy cost is not usually considered in trajectory planning because the energy cost of the optimal trajectory is usually calculated without accounting for possible deviations. By having the rotor grouping energy cost optimized, one fewer aspect of the drone system needs to be optimized. In addition, for Grouping 1, each group can be controlled separately since each of the four groups controls one specific motion. This means that each group can be controlled by its own control laws that are tailored specifically to the motion they are intended to do.

Another place where rotor grouping may make a difference is near stall conditions or during more extreme motions. For example, when a large gust hits the drone, a very fast correction to the trajectory must be made. Extreme corrections like this cost a lot of energy, so having an optimized rotor grouping that can accomplish this correction with minimal energy consumption will make a significant difference when flying in extreme conditions. Thus, the rotor grouping can make a difference by reducing the control cost to perform an extreme correction.

2.5.5 Conclusions

This section examines the role of rotor grouping in reducing energy consumption of multi-rotor drones. It has been demonstrated through open-loop simulations that rotor grouping affects the total energy consumption for both basic and complex trajectories. From studying the basic trajectories it was concluded that the different rotor speed inputs are likely the cause of the difference in energy between the groupings. In addition, the complex trajectory examined in this section does not show that energy compounding effects exist when the basic trajectories are combined to produce a complex trajectory. However, this does not eliminate the possibility of a trajectory existing that does display energy compounding effects. Thus, it is important when working with overactuated drones, such as the 18-rotor drone analyzed in this section, to consider ways in which the rotors can be grouped in order to minimize the overall energy consumption of the drone and increase its endurance.

Chapter 3

AIM 2: QUANTIFYING THE ENERGY COSTS OF CONTROL SYSTEMS AND DEVELOPING A FRAMEWORK TO DESIGN ENERGY-EFFICIENT CONTROLS

In Aim 2, I quantify the difference in energy costs between an open-loop and closed-loop system. The quantification of this difference is used to establish a framework within which the energy costs of different controllers can be compared. Using the framework to compare energy costs of different controllers will lend itself to designing controllers for energy efficiency by understanding what control system features cause higher energy usage. In this framework, the energy cost of the open-loop system is used as a benchmark against which to compare the closed-loop system energy cost so that comparisons between controllers will be objective.

Using this framework, Aim 2a demonstrates that the closed-loop systems created with a PID controller uses more energy than the open-loop system. Aim 2b also uses this framework to show that the integral control can result in excessive energy usage in certain applications by comparing the energy cost of the PID controller from Aim 2a to that of a PD controller.

The work done on Aim 2a and part of Aim 2b are published in [31].

3.1 Prior Work

A lot of research has been done on controls and trajectory planning for quadcopters carrying suspended payloads [5, 12, 22, 37, 40, 42, 43, 49]. Mainly quadcopters are studied because they are commercially available. However, these quadcopters cannot carry heavy payloads over extended distances [1, 13, 22, 27]. Because of this there is growing interest in developing and studying multi-rotor drones that are able to carry heavy payloads over long distances, characteristics that are required for more involved tasks such as utilities maintenance in difficult terrain, agriculture [53,67], and search and rescue [59]. Currently the largest multi-rotor drones being studied are octocopters [69], but even octocopters may not be powerful enough or provide enough redundancy. Therefore, there is a need for larger, high-performance multi-rotor drones to perform these difficult tasks while minimizing the amount of energy required to perform them [30].

In order to perform these tasks, a large amount of energy is required. Thus, it is important to ensure that the multi-rotor drone is as energy efficient as possible. Previously, work has been done in [30] on determining the effect of modeling choices for a multi-rotor drone using different rotor blade models and rotor groupings in order to make the open-loop system energy efficient. This work uses a much higher fidelity model by using blade element theory to model the rotor blades and shows that the blade element theory produces a significantly different result than the lumped blade model that is usually used in multi-rotor drone models to simplify calculations. There are also opportunities to reduce energy costs in the closed-loop system by making the controller energy-efficient.

Previously, the research that has been completed on controllers for multi-rotor drones is mainly trajectory-based. These controllers use trajectory-based methods such as deviation from the main trajectory [22,43], being able to pass through specified way points [12], minimization of flight time [49], obstacle avoidance [5], and minimizing payload swing [12,37] to adjust the controller input. In these controllers, the energy consumption, if it is considered at all, is taken into account via a cost function planning optimal trajectories [43]. While this accounts for the nominal energy expenditure to fly a trajectory, it does not take into account the energy cost of other disturbance events that could occur during flight or the energy cost of flying autonomously using a controller as compared to flying the drone as an open-loop system with specified inputs.

In addition, there has been minimal discussion of energy costs of such controllers. Most energy optimizations occur when planning the trajectory for the drone [2,45,53]. In [26], two energy-optimal controllers are developed, but the actual energy used to fly the drone is never discussed let alone compared to the open-loop energy required to fly the same trajectory.

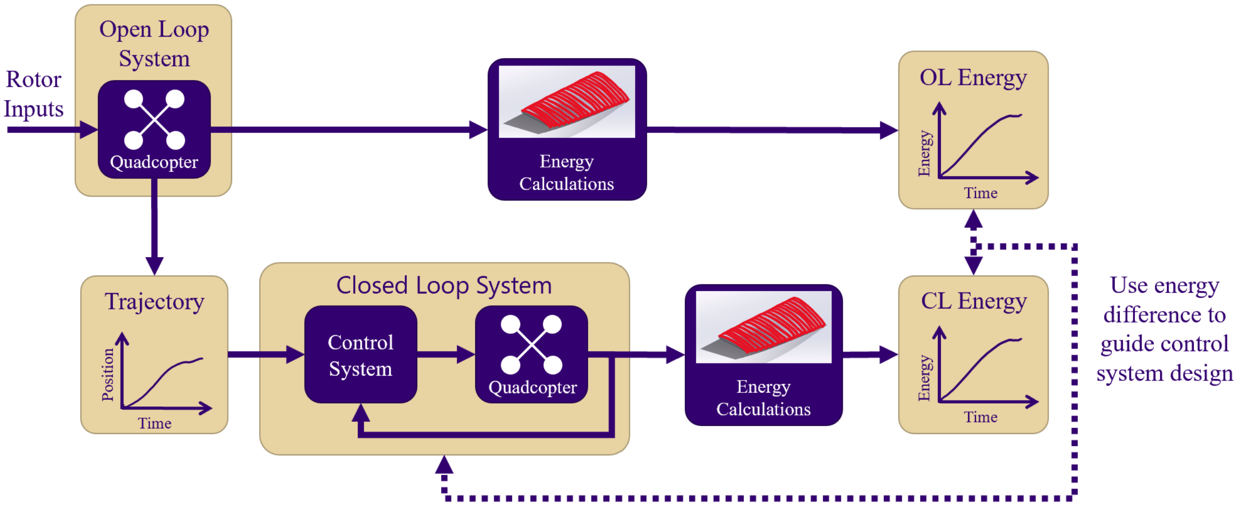


Figure 3.1: Schematic of framework for calculating the energy cost of a control system

The main focus was on the effectiveness of these controllers at stabilization and minimizing trajectory error. Thus, even in the research on developing controllers that are optimized for energy consumption, there is no distinction between the closed-loop and open-loop energy costs.

3.2 Proposed Framework

The difference between the open- and closed-loop energy costs is important to quantify because then the true cost of a control system can be understood. The relative increase in energy costs for increasingly extreme trajectories can then be calculated which increases understanding of how the energy efficiency of different controllers varies with the accuracy of the tracking as trajectories become more extreme. Understanding these aspects of different controllers will enable the designer to choose a better controller for any given situation.

This framework shown in Fig. 3.1 allows the designer to compare the performance of various controllers as well as various flight and disturbance conditions. Using the open-loop system energy consumption as a baseline gives an objective view of controller performance. Therefore, the actual energy cost of a controller can be found and compared to other controllers. In addition, if the open-loop energy consumption is known, then the energy cost of the correction following a disturbance can also be quantified. For example, if the drone will fly in areas known to have sudden gusts, the energy usage for the open-loop system to fly a control trajectory can be compared to the closed-loop system's energy usage to correct for a chosen gust disturbance. This process can be repeated for multiple controllers to compare the performance of the various controllers. Thus, this formulation can be used to find the true optimal controller for a given flight situation.

This chapter seeks to address the gap of understanding in how to develop an energy-efficient controller for multi-rotor drones. To do the preliminary work, several simple open-loop trajectories are chosen as the desired trajectories for a closed-loop system. The closed-loop systems will be created using two different controllers in order to compare the energy costs of both controllers with that of the open-loop system and each other. Comparing the energy costs will reveal what controller features are "expensive" in regards to energy usage. In addition, the total energy consumption will also be compared to the trajectory errors for each controller. This comparison provides information on the effectiveness of the controller at reaching the desired trajectory as it is related to the total energy usage. This knowledge can then be used to choose the most energy efficient controller while maintaining trajectory-tracking performance.

3.3 Modeling

In order to complete the study described, accurate models for the drone dynamics and the aerodynamics must be used. The dynamic model for the multi-rotor drone is described in Sec. 2.2.1, the aerodynamic rotor blade model is described in Sec. 2.2.2, and the rotor grouping used in the following work is described in 2.5.1.

3.4 Control Systems

In this section, two control systems that will be used to show the energy cost of flying a multi-rotor drone will be discussed. In addition, a cap on the rotor speed is used so that the rotor speed never increases above its max speed or decreases below 0 RPM.

3.4.1 PID Controller

A PID controller is developed to use as a proof of concept. This PID controller acts independently on each of the four basic vertical, roll, pitch, and yaw motions. The control constants are chosen so that motions up to 30 degree rotations in the roll, pitch, and yaw directions are still stable for the closed-loop system. The control system is given below in Eq. 3.1.

$$u = -K_P(\mathbf{r}_d - \mathbf{r}) - K_I \int (\mathbf{r}_d - \mathbf{r})dt - K_D(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) \tag{3.1}$$

where K_P, K_I, and K_D are matrices with constant gains in the appropriate elements, r_d is the desired trajectory position and angle, and r is the actual trajectory position and angle. The resulting value u will be added onto the open loop rotor speeds in order to get the total rotor speed inputs for the drone. The actual entries for r and r_d are given below in Eq. 3.2.

$$r = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}, \text{ and } r_d = \begin{bmatrix} x_d \\ y_d \\ z_d \\ \phi_d \\ \theta_d \\ \psi_d \end{bmatrix} \tag{3.2}$$

The gain matrices are chosen such that entries corresponding to each angle and position are only nonzero for the appropriate rotors. For example, rotors 2, 3, 11, and 12 will only have entries corresponding to deviations θ since they are in the pitch group, and entries corresponding to deviations in x and y. This is because x and y movements An example of the K_p matrix is given below.

$$\begin{array}{cccccccc}
 & x & y & z & \phi & \theta & \psi & rotor \\
\left[\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right.\!\!\!\! &
\begin{array}{c} 0 \\ K_{x_2} \\ K_{x_3} \\ 0 \\ 0 \\ K_{x_6} \\ 0 \\ K_{x_8} \\ 0 \\ 0 \\ K_{x_{11}} \\ K_{x_{12}} \\ 0 \\ 0 \\ -K_{x_{15}} \\ 0 \\ -K_{x_{17}} \\ 0 \end{array} &
\begin{array}{c} 0 \\ K_{y_2} \\ K_{y_3} \\ 0 \\ 0 \\ K_{y_6} \\ 0 \\ K_{y_8} \\ 0 \\ 0 \\ K_{y_{11}} \\ K_{y_{12}} \\ 0 \\ 0 \\ -K_{y_{15}} \\ 0 \\ -K_{y_{17}} \\ 0 \end{array} &
\begin{array}{c} K_z \\ 0 \\ 0 \\ K_z \\ 0 \\ 0 \\ K_z \\ 0 \\ 0 \\ K_z \\ 0 \\ 0 \\ K_z \\ 0 \\ 0 \\ K_z \\ 0 \\ 0 \end{array} &
\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ K_\phi \\ 0 \\ K_\phi \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -K_\phi \\ 0 \\ -K_\phi \\ 0 \end{array} &
\begin{array}{c} 0 \\ -K_\theta \\ -K_\theta \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ K_\theta \\ K_\theta \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} &
\left.\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ K_\psi \\ 0 \\ 0 \\ 0 \\ K_\psi \\ 0 \\ 0 \\ 0 \\ 0 \\ K_\psi \\ 0 \\ 0 \\ 0 \\ -K_\psi \end{array}\right] &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{array}
\end{array} \tag{3.3}$$

In Eq. 3.3, K_{xi} and K_{yi} are linear combinations of K_ϕ and K_θ since x and y are seen as motions found by combining the gains in the roll and pitch directions using a sensitivity that relates the motion in the x and y direction to the motion in the roll and pitch directions. Further discussion of how to find this sensitivity is given in the following section. Using these results, the equation to find K_{xi} and K_{yi} are given as follows

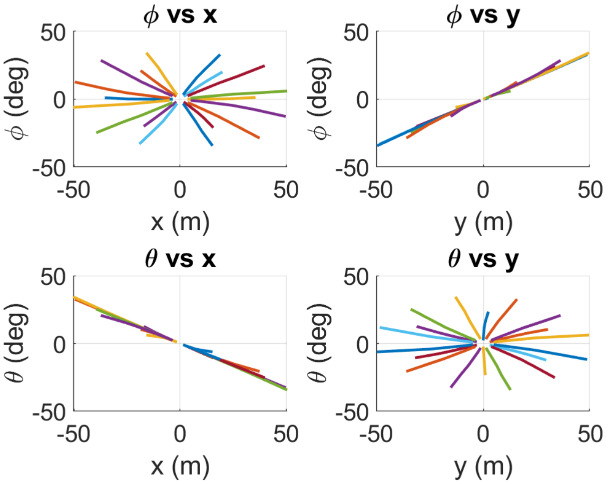


Figure 3.2: Plots showing the sensitivities are linear within a small range of rotor speed variations

$$\begin{bmatrix} K_{\phi,i} \\ K_{\theta,i} \end{bmatrix} = \begin{bmatrix} \left[\frac{d\phi}{dx}\right]_i & \left[\frac{d\theta}{dx}\right]_i \\ \left[\frac{d\phi}{dy}\right]_i & \left[\frac{d\theta}{dy}\right]_i \end{bmatrix} \begin{bmatrix} K_{xi} \\ K_{yi} \end{bmatrix} \tag{3.4}$$

where $K_{\phi,i}$ and $K_{\theta,i}$ are the entries in the gain matrix related to the i-th rotor and the appropriate angle, and $\left[\frac{d\phi}{dx}\right]_i$, $\left[\frac{d\phi}{dy}\right]_i$, $\left[\frac{d\theta}{dx}\right]_i$, and $\left[\frac{d\theta}{dy}\right]_i$ are the sensitivities of the i-th rotor. In this way, the controls on the horizontal motion are dependent on roll and pitch which are much more intuitive to control.

Sensitivity Study for Horizontal Motion

In order to get the sensitivities used in Eq. 3.4, the open loop system was simulated with rotor speed inputs varied for each rotor by small amounts ranging from 10 RPM to 100 RPM. From these simulations the roll, pitch, x, and y motions when changing a single rotor input are measured. In order to get the sensitivities, the roll or pitch angle is compared to the deflection in the x or y directions (Fig. 3.2). Each of the lines in the plot for Fig. 3.2 represent the relationship between the angle and position when a single rotor's speed is

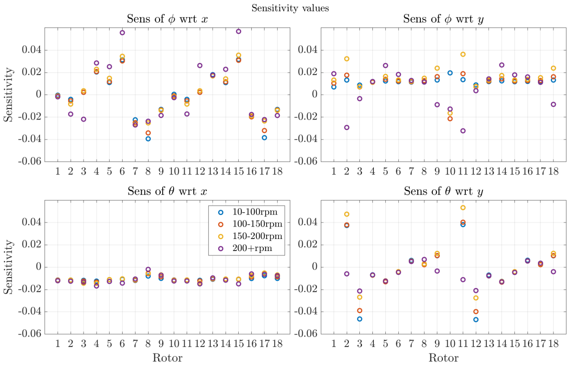


Figure 3.3: Plots showing the sensitivities for each range in the piecewise sensitivity

varied by a range of rotor speeds. The lines for the i-th rotor for each relevant comparison can be represented using the equation shown in Eq. 3.5

$$\Phi_i = \left[\frac{d\Phi}{dX}\right]_i X_i \tag{3.5}$$

where Φ_i is the angle and X_i is the position during the flight when varying the i-th rotor speed. The sensitivity, or slope of the line, can then be found using least-squares regression.

A potential area of concern is that the sensitivities used to create the control gain matrices will only apply to a very small range of rotor speed variations making these results inaccurate. One way to address this issue is to create a piecewise sensitivity function. The sensitivities will depend on how large the rotor speed variation is since larger rotor speed variations will result in the sensitivity relationship becoming more nonlinear. The bounds for the piecewise sensitivity function are chosen based on which rotor speed regions produce an trend that appears linear like in Fig. 3.2. The regions of rotor speed variation chosen are (1) 0-100 rpm, (2) 100-150 rpm, (3) 150-200 rpm, and (4) greater than 200 rpm. Figure 3.3 shows the values of the sensitivities in each range of rotor speeds. It is clear from the plots in Fig. 3.3 that the sensitivities in general do not vary a lot in each range of rotor speeds.

Simulations were completed to determine if these sensitivities were necessary. The open loop trajectories were used to determine if the rotor speeds exceeded the 100 rpm range for the linear sensitivities or not. Based on these simulations, the rotor speeds rarely exceeded

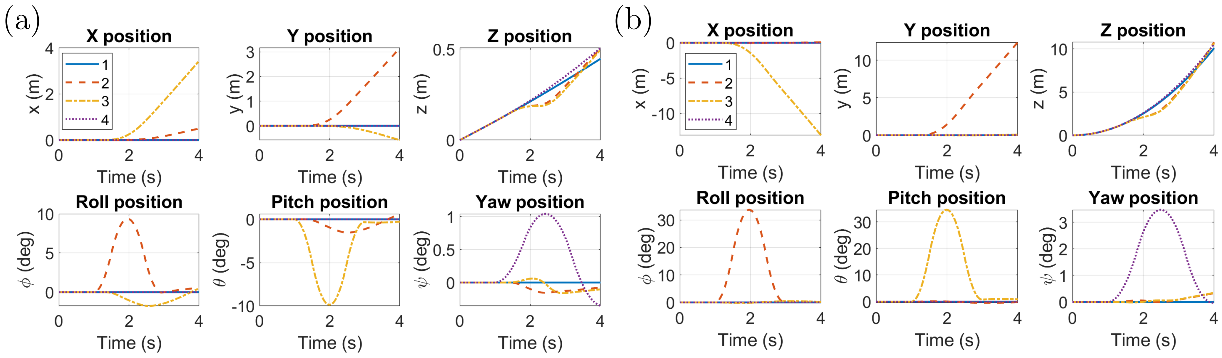


Figure 3.4: Plots of the four (a) normal and (b) extreme open-loop trajectories

the 100 rpm range resulting in no significant differences in the resulting trajectories when the piecewise function for sensitivities was used as compared to when a constant value was used for the sensitivities. Thus, it was reasonable to use a constant value for the sensitivity in all simulations in this study.

3.4.2 PD Controller

As a comparison, a PD controller is also developed. This PD controller, like the PID controller in the previous section, controls each of the four basic motions of vertical, roll, pitch, and yaw independently. The output of the PD control system is shown in Eq. 3.6.

$$u = -K_P(\mathbf{r}_d - \mathbf{r}) - K_D(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) \tag{3.6}$$

The same control constant matrices for K_P and K_D from the PID controller are used in the PD controller so that the two controllers are directly comparable.

3.5 Aim 2a: Evaluate the Difference between Open- and Closed-Loop Energy Consumption

To characterize the energy costs of controllers and how that compares to the energy costs of controlling a system with open-loop inputs, several open-loop trajectories will be chosen as the input to the controllers described in the previous section. These trajectories are the four basic trajectories of vertical, roll, pitch, and yaw motion. These trajectories are pictured in Fig. 3.4(a). The nonlinear closed-loop system will then be simulated with these inputs and then the energy required to track the open-loop trajectories will be compared to the open-loop trajectory energy. In addition the difference in the trajectories will also be compared to ensure than any energy differences are not due only to the differences between the open-loop and closed-loop trajectories flown.

An additional four closed-loop simulations will be completed with a more extreme version of the trajectories shown in Fig. 3.4(a). These extreme trajectories are shown in Fig. 3.4(b). The main difference between these trajectories and the trajectories in Fig. 3.4(a) is that the roll, pitch, and yaw motion will increase to approximately 30 degrees (3 degrees for yaw motion) as opposed to the approximately 10 degrees (1 degree for yaw motion) for the basic trajectories. The reason the yaw motion is so much smaller is that because a very large change in the rotor speed is required to achieve even a small rotation in the yaw direction.

For all eight of these simulations, the closed-loop system will have a perturbed initial condition of 0.01 m/s for the x, y, and z velocities. This is used to represent a sudden disturbance in the system, such as a sudden, impulsive gust which applies a perturbed initial condition, that pushes the drone.

3.5.1 Results

The trajectories of the closed loop system and the error with the desired trajectory are plotted by their individual components in position (x, y, and z) and angle (ϕ, θ, and ψ) in Fig. 3.5. The error between the two trajectories plotted in Fig. 3.5 is calculated as follows

$$\%\text{ error} = x_{CL} - x_d \tag{3.7}$$

where x_{CL} is the relevant state variable for the closed-loop simulation and x_d is the desired trajectory of the relevant state variable. For example, for the vertical trajectory $x_{CL} = z_{CL}$ and $x_d = z_d$ and for the roll trajectory $x_{CL} = \phi_{CL}$ and $x_d = \phi_d$.

In addition to examining the trajectories and errors on a component basis, the error is

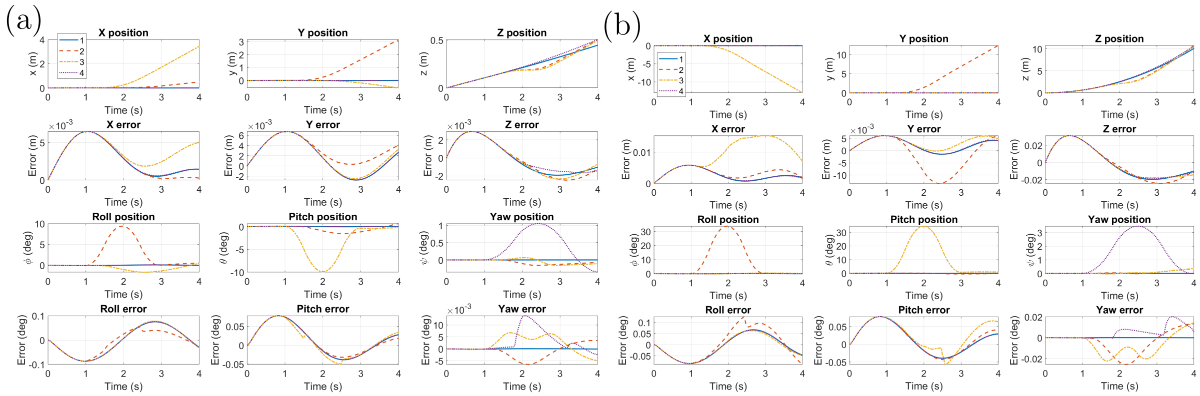


Figure 3.5: Plots of the four closed-loop trajectories and the errors for the (a) normal and (b) extreme trajectories for the PID controller

also examined cumulatively by finding the quadrature sum of the error in the position. This is calculated using the following equation

$$\% \text{ error} = \frac{\sqrt{(x_{CL} - x_d)^2 + (y_{CL} - y_d)^2 + (z_{CL} - z_d)^2}}{\sqrt{x_d^2 + y_d^2 + z_d^2}} \times 100\% \tag{3.8}$$

where x_{CL}, y_{CL}, and z_{CL} are the actual positions of the drone and x_d, y_d, and z_d are the desired positions of the drone. The difference in energies is calculated similar to Eq. 3.7 as shown in the equation below

$$\% \text{ difference} = \frac{E_{CL} - E_{OL}}{E_{OL}} \times 100\% \tag{3.9}$$

where E_{CL} is the total energy required for the closed-loop system to track the trajectory and E_{OL} is the total energy required for the open-loop system to fly the trajectory.

Using Eq. 3.8 and Eq. 3.9, Tab. 3.1 is created to help understand what the main causes are for the differences in power between the open-loop system and the closed-loop systems. The average trajectory error shown in Tab. 3.1 is found by calculating the mean of the vector resulting from Eq. 3.8. A positive energy difference means that the closed-loop flight used more energy, and a negative energy difference means that the open-loop flight used more energy. It is clear from the table that the PID controller uses significantly more energy than the open-loop system for more extreme trajectories. The difference in energy increases

Table 3.1: Summary of results for energy difference and trajectory error for PID controller

	Energy Difference		Trajectory Error	
Trajectory Direction	Normal	Extreme	Normal	Extreme
Vertical	7.09%	62.22%	5.11%	19.291%
Roll	10.28%	73.27%	4.79%	19.287%
Pitch	9.43%	72.80%	4.81%	19.313%
Yaw	12.49%	71.68%	5.05%	19.281%

significantly with the extreme trajectory as opposed to the normal trajectory. The average trajectory error also increases by about four times. This leads to the question of where the energy difference is arising from since an average trajectory error and final energy difference do not give a full picture of the results. To determine where the energy differences are coming from, the controller rotor speed input, trajectory error, and energy difference as a function of time are examined in Fig. 3.6.

As seen in Fig. 3.6, for all trajectories the closed-loop system starts with using less energy than the open-loop system, as the simulation progresses, then the energy of the closed-loop system increases above that of the open loop system. Figure 3.6(a) shows that the closed-loop system starts with using less energy than the open-loop system before increasing to using about 7-13% more energy than the open-loop system. In addition, the trajectory error appears to have no effect on the energy difference since no significant change in energy difference is seen as the trajectory error decreases. Conversely, the rate at which the energy difference increases appears to increase with increasing rotor speed inputs. The larger the rotor speed increase, the faster the energy difference increases. This is corroborated in Fig. 3.6(b) where the rotor speed increases until the saturation limit is first reached at about 2.5 seconds for all four trajectories. After this saturation limit is reached, the energy difference

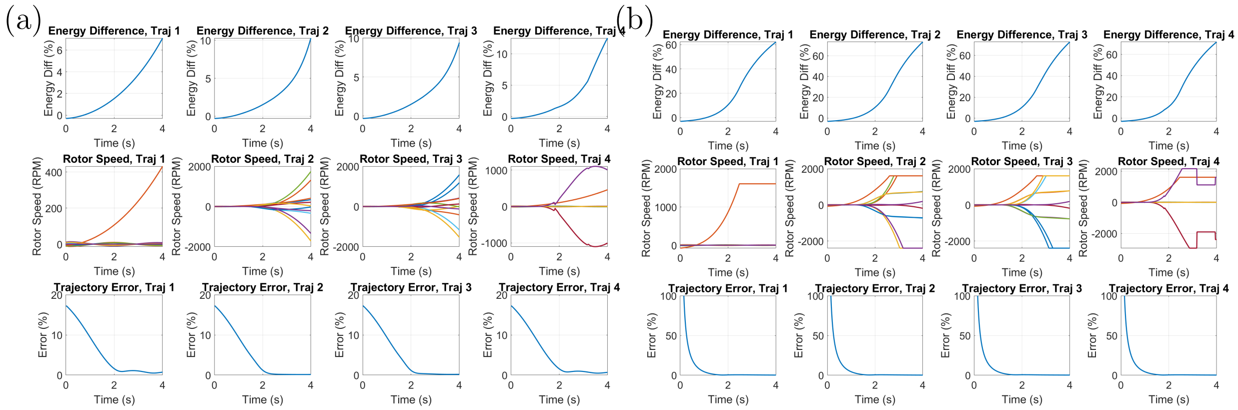


Figure 3.6: Plots of the relevant energy differences, controller inputs (u), and trajectory error as a function of time for the (a) normal and (b) extreme trajectories of the PID controller

shows an inflection point showing that the rate at which the energy difference increases is due to the increasing rotor speed.

The next question is why the controller inputs appear to increase exponentially until the saturation limit is reached despite the trajectory error approaching zero. It is hypothesized that the continuous increase in the rotor speed input is likely due to the integral term in the controller since the integral portion will accumulate error for the entirety of the flight. To check this hypothesis, a PD controller is used to simulate a closed-loop flight of the same eight trajectories considered for the PID controller.

3.6 Aim 2b: Comparing the Energy Costs of Two Controllers

In order to determine if the integral controller is the reason for the increase in energy difference, a PD controller (described previously) will be considered. The closed loop system with the PD controller is simulated with the same eight desired trajectories (see Fig. 3.4) as the PID controller, and the resulting energy calculations are compared to determine if the integral control is what is responsible for the difference in energy. In addition, the magnitude of the integral control is varied to fully characterize the trade-off relationship between trajectory tracking and energy consumption for integral control for the multi-rotor drone.

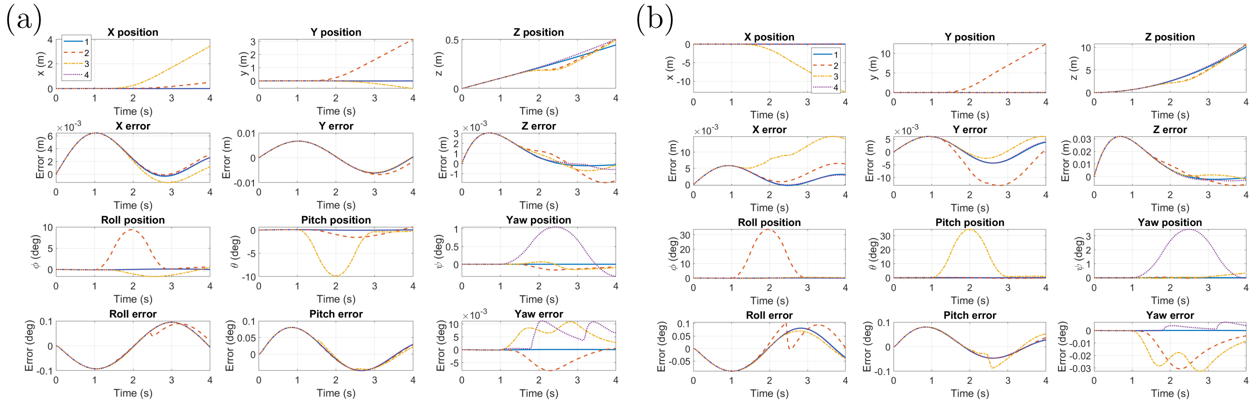


Figure 3.7: Plots of the four closed-loop trajectories and the errors for the (a) normal and (b) extreme trajectories of the PD controller

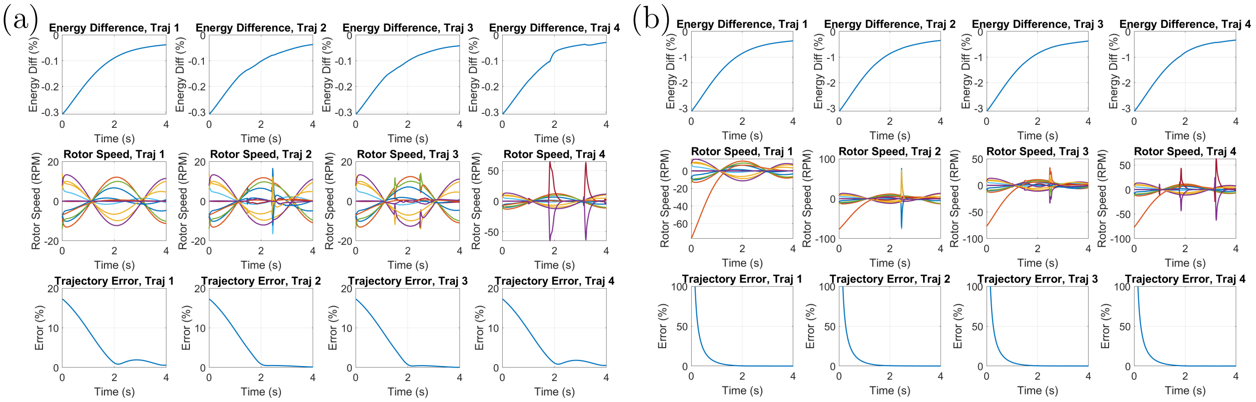


Figure 3.8: Plots of the relevant energy differences, controller inputs (u), and trajectory error as a function of time for the (a) normal and (b) extreme trajectories of the PD controller

3.6.1 PD Controller Energy Comparison

The results of the eight simulations are shown in Fig. 3.7; the error between the actual and desired trajectories plotted in Fig. 3.7 is calculated using Eq. 3.7. In addition, the energy difference, controller inputs (u), and trajectory error are plotted in Fig. 3.8. Note that the trajectory errors in Fig. 3.8(b) start above 100% because the desired trajectories start very close to zero leading to a deceivingly large error percentage (see Eq. 3.8). The error in each position and angular position component is shown in Fig. 3.7 for additional context.

Like with the PID controller system, Fig. 3.8 shows that the trajectory error does not have an effect on the energy difference since the drop in trajectory error near the beginning of the simulation does not cause any significant changes in the energy difference. The rotor speeds appear to have the most effect on the energy difference plots. This can especially be seen in the energy difference plot for trajectory 4 in Fig. 3.8(a) where the energy difference spikes just before 2 seconds which corresponds with the first spike in one of the controller inputs. In general, however, it is difficult to see the exact effect of the controller inputs on the energy difference. This is because the spikes in controller inputs are of the same or smaller magnitude as the oscillating inputs, except for in trajectory 4, making the effects of these spikes difficult to see.

The most notable feature is that the PD controller system always uses less energy than the open-loop system. The energy difference approaches zero as the simulation progresses showing that the closed-loop system energy does approach the open-loop system energy. In addition, flying the extreme trajectory uses less energy than flying the normal trajectory. This can be partially explained by considering the controller inputs. The larger trajectory error in the extreme trajectory implies that there is more room to find a lower operating rotor speed to follow the trajectory. This can be seen since one of the inputs for all of the extreme trajectories starts at a much lower value for the extreme trajectory (Fig. 3.8(b)) compared to the normal trajectory. This lower rotor speed may be responsible for the extreme trajectory requiring less energy than the normal trajectory since power is proportional to rotor speed as seen in Eq. 2.24. Other possible factors are how the torque changes over time. In addition, it may not be fair to compare the normal and extreme trajectory energy differences in this manner because the error in the trajectories is very different between the two trajectory types.

Discussions Comparing Two Controllers

The average trajectory error and final energy difference of the PD controller is listed in Tab. 3.2 and can be directly compared to the results listed in Tab. 3.1. Like before, the

Table 3.2: Summary of results for final energy difference and average trajectory error for PD controller

	Energy Difference		Trajectory Error	
Trajectory Direction	Normal	Extreme	Normal	Extreme
Vertical	-0.0376%	-0.367%	5.29%	19.32%
Roll	-0.0368%	-0.352%	4.81%	19.38%
Pitch	-0.0417%	-0.374%	4.76%	19.37%
Yaw	-0.0286%	-0.337%	5.24%	19.32%

mean quadrature sum error of the trajectory is calculated using Eq. 3.8 and the final energy difference is calculated using Eq. 3.9.

These results clearly show that the PD controller uses much less energy overall compared to the PID controller while maintaining a comparable average error in the trajectory for both the normal and extreme trajectories. Tables 3.2 and 3.1 show that for both trajectories, the PD controller uses slightly less energy than the open-loop system while the PID controller uses significantly more energy than the open-loop system. In contrast, the average trajectory error is very similar between the two different controllers despite the large difference in energy. Further understanding can be gained by examining the controller inputs to the rotors and trajectory errors.

Examining the controller inputs from both controllers in Figs. 3.6 and 3.8 reveals that the PID controller produces rotor inputs on the order of $10^2 - 10^3$ as compared to the PD controller which produces rotor inputs which are on the order of $10^1 - 10^2$. Both controllers start with the same energy differences, but have very different final energy differences. Since the same proportional and derivative control matrices are used in both controllers, then this means that the difference in the controller inputs is due to the integral term. This makes sense since at the start, the integral error will be zero so initial rotor speed inputs from both controllers is only dependent on the proportional and derivative errors (see Eq. 3.1). Further insight can be found by examining the trend in the energy differences.

Examining Figs. 3.6 and 3.8 for the trend in energy differences reveals that the PID controller results in an exponential increase in relative energy usage while the PD controller has an asymptotic increase in relative energy usage. This trend can be partially explained using the controller inputs. For the PID controller, at least one of the rotor inputs from the controller shows an exponential increase in rotor input until the rotor speed maximum is reached. This contrasts starkly with the trend for the PD controller where the rotor inputs oscillate with a relatively small amplitude that is decreasing over time. The exponential increase in one or more inputs for the PID controller explains why the energy difference shows an exponential increase for the normal trajectories. By comparison, the asymptotic behavior in the energy difference for the PD controller is explained by the decreasing amplitude of the oscillating rotor speed inputs. This difference in trends implies that the PD controller is more energy efficient than the PID controller because the integral term accumulates the error over the entire flight time. Therefore any errors at the start of the trajectory will continue to affect the control inputs later in the trajectory. Thus, for extreme trajectories where large errors can occur, integral control will result in large controller inputs leading to higher energy consumption.

Finally, examining both the average trajectory (Tabs. 3.2 and 3.1) and the trajectory error throughout the flight (Figs. 3.6 and 3.8) reveals that both controllers perform very similarly in terms of reducing trajectory error as the trajectory progressed. Thus, it can be concluded that the PD controller is a significantly more energy efficient way to track a trajectory for a large, payload-carrying multi-rotor drone.

There are several observations made above that need to be further explored. First, examining the rotor speeds for the PID controller reveals that they deviate significantly from the nominal rotor speeds. This is possibly because the K_I values chosen are too large significantly increasing the importance of accumulated error. The large deviation in rotor inputs also may make the results for the PID controller less reliable because the PID controller is a linear controller meaning that it is intended to function close to the nominal values. In addition, the increasing inputs may indicate that the controller is unstable for this particular initial condition or desired trajectory. While this may not be the case for all initial conditions or situations, further investigation into the linearity of the sensitivity plots shown in Fig. 3.2 has revealed that in general, the controllers simulated here do not generally exceed the ±100rpm limit on the linear sensitivities. The only rotors exceeding this limit for linearity is the rotors in the PID controller. Second, additional work must be done to study the effect of smaller values of K_I to confirm the effects of the integral control. Varying the magnitude of K_I will also give further insight into the trade-off relationship between energy consumption and trajectory tracking. This study and its results are described in the following section.

3.6.2 Variable Integral Control Study

For this study, simulations using the four small-amplitude trajectories shown in Fig. 3.4(a) are run with the same initial conditions. The integral control parameter n is varied between 0 (PD control from Sec. 3.4.2) and 1 (PID control from Sec. 3.4.1) for all four small-amplitude trajectories to demonstrate the effects that varying integral control will have on the energy difference and steady state trajectory difference.

In addition, results from Sec. 3.6.1 also motivate studying the trade-off relationship for large-amplitude trajectories since the large-amplitude steady-state trajectory errors appeared to exhibit different behavior compared to the small-amplitude trajectory steady state errors. Therefore the variable integral control study will also be done using the large-amplitude trajectories to determine if trends different for more extreme trajectories.

Variable Integral Control

In order to determine the trade-off relationship for integral control, the magnitude of the integral control will be varied using a parameter n as shown in the control law below.

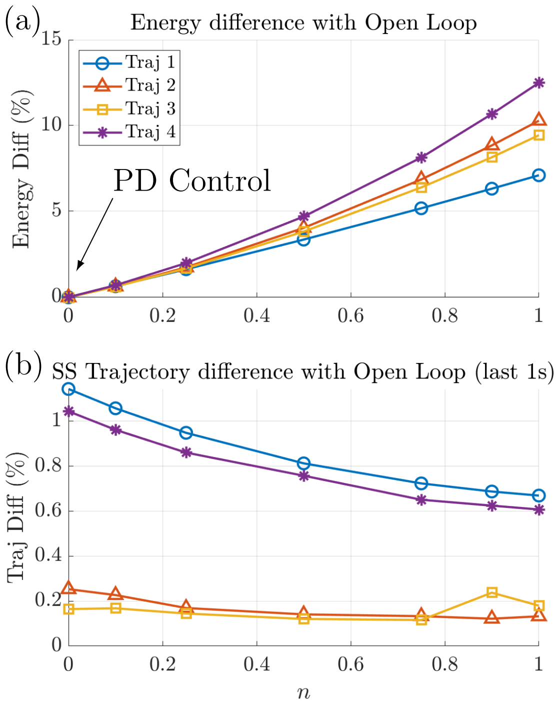


Figure 3.9: Plots of (a) energy difference and (b) steady state trajectory difference as $n\mathbf{K}_I$ is varied for small-amplitude trajectories; $n = 0$ results in the PD control system

$$u = -\mathbf{K}_P(\mathbf{r}_d - \mathbf{r}) - n\mathbf{K}_I \int (\mathbf{r}_d - \mathbf{r})dt - \mathbf{K}_D(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) \tag{3.10}$$

where $\mathbf{r}$ and $\mathbf{r}_d$ are defined as in Eq. 3.2.

Results

The results of the simulations are collected in the plots shown in Fig. 3.9. The resulting trajectories are not plotted here as they look very similar to the trajectories shown in Fig. 3.5. The different marker styles indicate the four different trajectories that were tracked using the nK_I controller. For all four trajectories, the same general trend is seen: as the magnitude of the integral control (n) increases, the steady state trajectory error decreases and the energy difference increases.

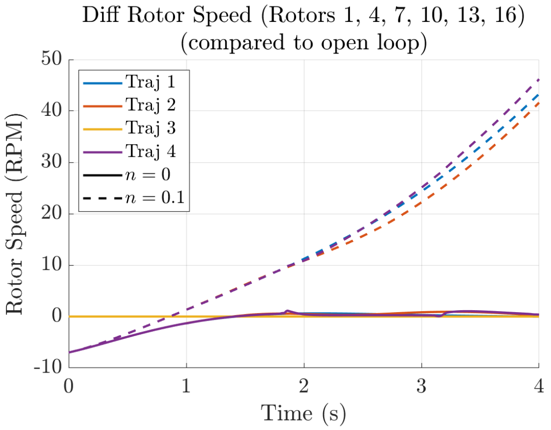


Figure 3.10: Plot of relevant rotor speed inputs for the PD control system and the PID control system with $n = 0.1$. The plot shows the difference in rotor speeds between the nominal, open-loop system rotor speed and the closed-loop system rotor speed

These results demonstrate that even a small amount of integral control will cause an increase in the energy difference. The reason for this energy increase can be found by plotting the rotor speeds for the PD controller system and the $n = 0.1$ controller system as shown in Fig. 3.10. In the case when there is no integral control, the rotor speeds approach the open-loop rotor speed. However, when there is a small amount of integral control ($n = 0.1$), this behavior changes and the rotor speed appears to grow without bound away from the open-loop rotor speed. This is true for all values of $n \neq 0$ and the rate of growth increases with the magnitude of n. These results demonstrate the trade-off relationship between energy usage and trajectory tracking since, as the integral control increases, the energy usage increases and steady-state trajectory error decreases. The relationships also appear to vary with motion type as the trade-off is much less pronounced when examining Trajectories 2 and 3 (roll and pitch motions). The results in Fig. 3.9 implies that when designing a control system it is important to identify what metric is more important, energy usage or close trajectory tracking and then perform an analysis like the one shown here since there will always be a trade-off between close trajectory tracking and energy consumption.

As stated previously, large-amplitude trajectories seem to show a different trade-off com-

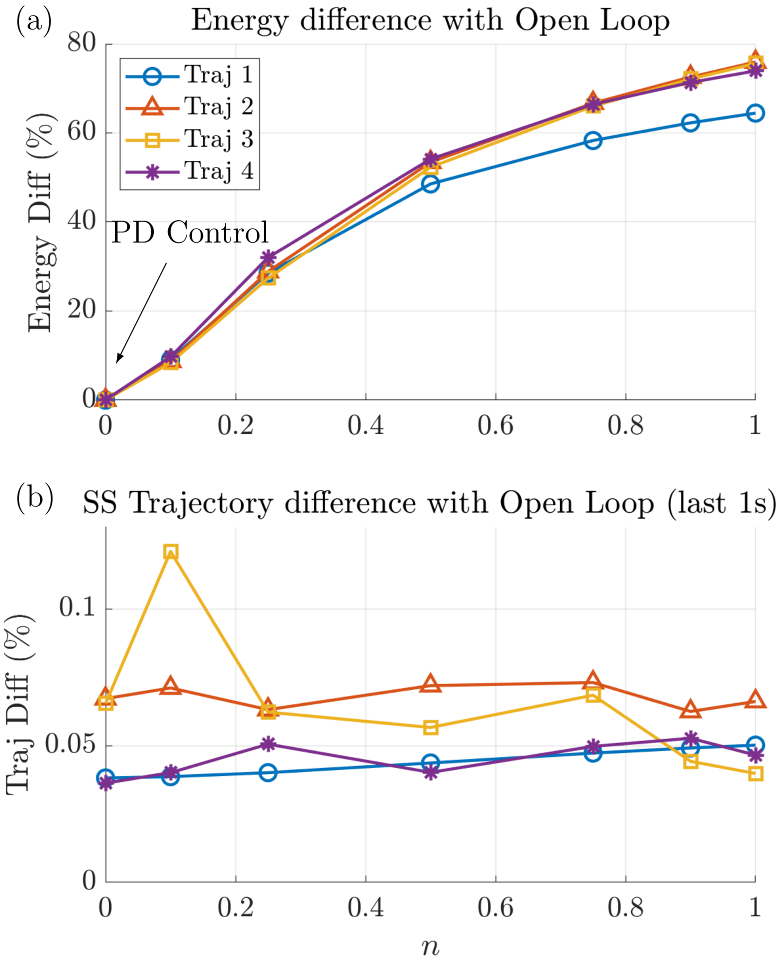


Figure 3.11: Plots of (a) energy difference and (b) steady state trajectory difference as nK_I is varied for large-amplitude trajectories; $n = 0$ results in the PD control system

pared to the small-amplitude trajectories. Additional simulations were completed for this study using the large-amplitude trajectories as the desired trajectories. The results are shown in Fig. 3.11. The trade-off relationship shown here is not the same as the relationship for the small-amplitude trajectories since the decrease in the steady state trajectory error appears to only show a decreasing trend for trajectories 2 and 3 while trajectories 1 and 4 show a slightly increasing trend. The increasing trend is likely due to the integral control being too large resulting in the drone over-correcting for the error leading to the steady-state error being larger for the larger integral control gains. Another potential reason is that the simulations are too short to allow the drone to reach a true steady state position. Further analysis of the actual trajectories for each integral control variation is needed to fully understand this. The main takeaway from these results is that it highlights the importance of checking the trade-off relationship for multiple trajectory amplitudes. There may be different trade-off relationships depending on the anticipated type of trajectory and anticipated initial conditions as well as potential disturbances during the flight. The work done here is intended as a demonstration that these considerations are important when choosing a control system for a particular application.

Discussions

The results presented here demonstrate that the choice of control system is important because there is a trade-off between better trajectory tracking and energy optimization. If energy usage is the most important metric, then the designer should chose a controller that minimizes the energy usage while maintaining acceptable trajectory performance. Conversely, if trajectory tracking is more important, then the designer should choose a controller that tracks the trajectory within the desired margins while maintaining a reasonable energy consumption.

In this study, PID and PD controllers were used to demonstrate how the integral control may be a more energy expensive feature to include in the control system. Varying the strength of the integral control term allows characterization of the energy cost and trajectory error trade-off for integral control. From the results, it is clear that the advantage to including integral control is that the trajectory can be tracked more closely. However, a direct comparison of the PID and PD controller trajectories shows that for the controllers specified here, the PID controller does not perform significantly better than the PD controller over the course of the desired trajectories. Therefore, if a trajectory does not have narrow margins, then an energy saving approach with less or no integral control may be more appropriate so that less energy is used to fly a specific trajectory.

The results also show the importance of knowing the type of trajectories the drone will experience. The results in Figs. 3.9 and 3.11 demonstrate that it is important to check the trade-off relationships between different controllers for different trajectory amplitudes. This can also be used to infer that different types of trajectories may also have a differing trade-off relationship, especially for more complex trajectories since the trajectories seen here are all simple one-degree-of-freedom motions.

However, these results may be skewed by the fact that only a single initial condition was used in all of the simulations. Additional simulations have been done to examine the effects of initial conditions. The initial conditions for all of the simulations shown previously were zero except for velocity which had an initial condition of $v_i = [0.01, 0.01, 0.11]$ m/s is used for all the trajectories. This initial condition corresponds to a perturbation of 0.01 m/s in the x, y, and z velocities. Additional simulations of the four small-amplitude trajectories using different initial conditions have been completed. The additional initial conditions included a perturbation of -0.01 m/s, 0.1 m/s, and -0.1 m/s in the x, y, and z velocities. The results trajectories are not pictured here since they do not significantly differ from the simulation results already shown. The results indicate that the initial conditions have no effect on the resulting energy differences. The only difference in results is that the larger perturbations of ± 0.1 m/s in the initial velocities result in a larger steady-state trajectory error due to the drone taking more time to approach the desired position from the initial condition. The general conclusions, however, do not change which implies that the choice of initial conditions does not affect the results presented here.

3.7 Conclusions

The purpose of this chapter is to determine the energy cost of control systems. First, a PID controller is used to track four low-amplitude trajectories and four high-amplitude trajectories. It is found that for the normal trajectories, the PID controller is about 10% less energy efficient than the open-loop system. This contrasts with the extreme trajectory results which show that the PID controller uses significantly more energy (60-70% more energy) than the open-loop system. Based on these results, it was hypothesized that the main cause for the extreme trajectory resulting in such a large energy difference is the integral control. To verify this, the same control system is examined without the integral component. The PD controller was used to track the same eight trajectories, and the results show that the PD controller is more energy efficient than the PID controller while maintaining similar trajectory errors. Examining the actual energy difference and control inputs over the simulated flight reveals that the large energy difference is likely due to the PID controller saturating the rotor speed inputs. As stated previously, the possible instability of the PID controller makes it difficult to guarantee these results for all situations and initial conditions. In order to fully characterize the energy cost of the integral control term, additional simulations were completed where the magnitude of the integral control was varied.

The additional simulations where the magnitude of the integral control was varied were completed on PID systems. The magnitude of the integral control was varied by a parameter n which ranged from 0 (PD control) to 1 (PID control from Sec. 3.4). The results of these additional simulations showed that there is a trade-off between energy consumption and trajectory tracking. How closely the trajectory is tracked is measured by calculating the steady-state trajectory error, the average trajectory error in the last one second of the simulation. The PID controller, in general, tracks the trajectory closely at the cost of more energy usage. By comparison, the PD controller uses less energy while tracking the trajectory less closely than the PID controller. Additional simulations using the large-amplitude trajectories showed that the trajectory amplitude has a significant effect on the trade-off relationship as well. This implies that finding the trade-off relationship requires using appropriate desired trajectories for the results and conclusions to be useful. In addition, more simulations were completed that ruled out the effect of initial conditions. Therefore, it can be concluded that it is important to understand the a trade-off between energy usage and trajectory tracking ability when choosing a controller. This trade-off needs to be characterized and understood in order to make an educated choice of the best controller for a given situation and set of requirements.

This study demonstrates that there is a trade-off between energy usage and trajectory tracking ability and that this trade-off is highly dependent on the types of trajectories the drone will need to fly as well as the initial conditions for those trajectories. PID and PD controllers are used to demonstrate this trade-off relationship. Although PID control produces a smaller position error, using it risks spending a lot more energy because some rotors may need to significantly increase their speed to reduce the position error. In contrast, the PD controller may not achieve the same level of position error as the PID control. This is because the rotor speeds simply fluctuate around their nominal values, spending roughly the same energy as in the open-loop flight. This analysis shows the importance of understanding the trade-off between different objectives when designing the control system for a multi-rotor drone. If the trade-off can be fully characterized, then a more educated choice of controller can be made. If the controller is not analyzed in this way, then the most effective controller may not be found and the desired margins for trajectory tracking and energy consumption will not be met.

Chapter 4

AIM 3: DEMONSTRATE THAT MODELING GUSTS AS A STOCHASTIC PROCESS IS NECESSARY

Like in the previous two aims, the focus of Aim 3 is to understand if current approaches for modeling wind disturbances, all of which are deterministic, are sufficient to capture real gust behavior or if adapting a stochastic model normally used in wind turbine applications will improve simulation results and energy cost estimates. There are certain checks that need to be done before applying a stochastic model to a multi-rotor drone. One of these checks include determining how sensitive the model is to different gust parameters. If the drone response is sensitive to different input parameters when the gust is deterministic, then the drone response will become very unpredictable when the gust is modeled stochastically. This check is important to perform as stochastic simulations, such as Monte Carlo simulations, can require a lot of computing time to run [4].

In Aim 3a, it is demonstrated that simulated flight trajectories are very sensitive to various parameters of deterministic gust disturbances. Complex physical behavior, such as aerodynamic stalls, further complicates straightforward prediction of the flight trajectories. As a result, there is no trend in the system's response to these disturbances. In Aim 3b, the effect of variations in the control system is explored to demonstrate that the drone's response is also very sensitive to variations in the control gain. Finally, in Aim 3c, the resulting energy calculations from these simulations are used to demonstrate that energy is dependent on a variety of factors including rotor speed and angle of attack. These three sub-aims will demonstrate that using stochastic simulations is necessary to account for every possible outcome when the multi-rotor drone is subjected to a stochastic gust disturbance since the system is very sensitive to certain wind input parameters. In this case, Monte Carlo simulations will be very useful to fully characterize the drone's response to a stochastic gust disturbance.

4.1 Prior Work

Within research on multi-rotor drones, external disturbances are considered in multiple capacities. The majority of the research completed on drones currently either focuses on other disturbance types (such as no-fly zones [39] or obstacles [8]), or treats external disturbances as a force-based disturbance [35, 52]. For papers considering gusts as a disturbance, they often model the disturbance using deterministic models [18,23,70,74]. This greatly contrasts with how wind is modeled in other applications as a stochastic process. For example, the Weibull distribution function is often used when calculating energy production of wind turbines [7, 61]. This is so that the stochastic nature of gusts is captured in the final energy estimates which would not be the case if deterministic models were used.

In literature on gust disturbance for multi-rotor drones, deterministic gust disturbance models are used without sufficient evidence provided to demonstrate that these models adequately capture drone behavior when responding to sudden gusts. In addition, there is a lack of understanding of how of gust disturbances affect energy consumption for multi-rotor drones. Using a stochastic gust model such as the Weibull distribution can give better insight into how multi-rotor drones will respond to sudden gusts. The Weibull distribution has not been applied to multi-rotor drones to represent the wind speed distribution in this way before. By using the Weibull distribution to model wind disturbances, the stochastic nature of gusts is taken into account when performing simulations and computing energy usage. This will give a better estimate of the total energy required to follow a trajectory through a gusty environment. The stochastic model may also capture phenomenon that is not seen when modeling the gust as a deterministic disturbance.

4.2 Simulation Inputs

The inputs for each of the simulations performed in this chapter are described in this section.

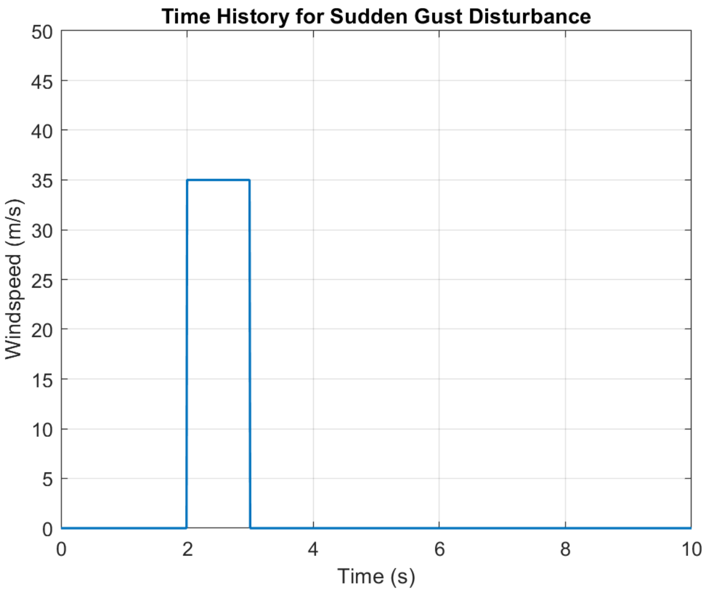


Figure 4.1: Time history of the wind magnitude for all simulations

4.2.1 Gust Disturbance

To demonstrate that drone response is highly sensitive to gust disturbance input parameters, a simple gust disturbance with variable parameters was chosen as the disturbance for the system. Sudden, short-duration gusts in calm conditions are likely to cause the largest response to the multi-rotor drone system. To mimic and simplify the time history of such a gust, the wind time history shown in Fig. 4.1 is chosen. This wind time history has no wind disturbance for the entire duration of the simulation except for a one-second interval starting at two seconds. The following parameters about the wind disturbance are varied:

- Wind velocity magnitude
- Wind angle above/below the horizontal plane
- Which rotors are hit by the wind

In methodically varying these parameters, simulations can be completed for the following parameter sweeps.

- Velocity magnitude: 5 m/s, 10 m/s, 20 m/s, 35 m/s, 50 m/s
- Angle above/below horizontal: 5°, 10°, 20°, 30°, 45°, 60°
- Rotors affected by wind: all rotors, half of the rotors

In addition to these variations, it is also be noted that the multi-rotor drone being simulated has its rotors grouped to fly using different rotor groups to control vertical motion and roll, pitch, and yaw rotation. The wind direction in the horizontal plane can also be varied. This parameter will not be varied as the changes in trajectory when wind direction in the horizontal direction are not as significant as when the wind angle above/below the horizontal plane is varied. For all simulations, the wind direction pointed 45 degrees relative to the positive x-axis of the drone so that all four rotor groups must respond to the wind disturbance. Finally, each of these parameters will be held constant for the duration of the gust disturbance.

It is also important to acknowledge that this parameterization of the wind is not realistic, but the simplicity of it will make the results easier to interpret. If a small, singular change made to the wind disturbance parameters results in a major difference in the resulting simulation results, then the simulation results are highly sensitive to the input parameter varied. If so, when the input gust is modeled as a stochastic process, the high sensitivity of the system dynamics will make drone response (and its probability distribution) very difficult to predict without using complex and expensive simulation tools, such as Monte Carlo methods.

4.2.2 Desired Trajectory

For all simulations, the multi-rotor drone will follow a simple hovering trajectory with zero initial conditions. This hovering trajectory holds the drone at the zero position without rotation or translation. This is done so that all simulation results can be compared to each other directly and the desired trajectory does not add unnecessary complications to the analysis by making it difficult to determine whether the differences arise from the complex desired trajectory or from the system's response to the wind disturbance.

4.3 Modeling

Since there is interest in determining if the power used while flying the trajectories is a parameter of interest, an alternative method for representing the power calculation could provide some insight into why certain gust disturbances cost more energy to respond to than others. Like with thrust coefficients, the power coefficient for an airfoil can be represented as C_P in the following equation.

$$P = \frac{1}{2}\rho C_P A V^3 \tag{4.1}$$

Based on Eqs. 2.18a, 2.18b, 2.23b, and 2.24, C_P for one element on the rotor can be found as a function of the coefficients of lift and drag (C_L and C_D, respectively) as follows.

$$C_P = C_D \cos(\phi) + C_L \sin(\phi) \tag{4.2}$$

Then power from one element in the rotor can be calculated as follows

$$P = \frac{1}{2}\rho C_P A r V_{rel}^2 \omega \tag{4.3}$$

where r is the distance of the element from the rotational axis of the rotor, V_{rel} is the relative velocity that the element is experiencing, and ω is the rotational speed of the rotor. Equation 4.3 is equivalent to the method for calculating power seen previously in Sec. 2.2.2. The advantage to adjusting the method for calculating power to this form is that now C_P can now be examined to determine if the angle of attack (AOA; α in Fig. 2.3) will play a large roll in the power required to fly the drone.

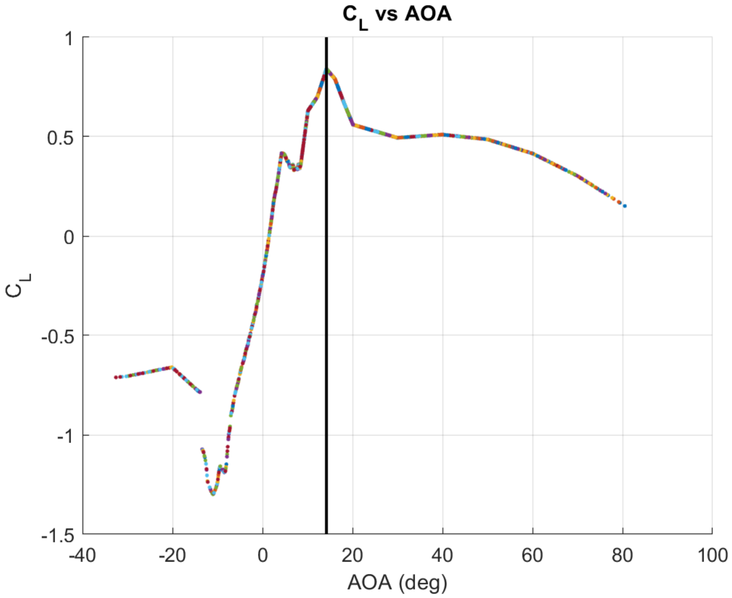


Figure 4.2: Plot of Coefficient of Lift vs. Angle of Attack, α, to find the critical angle of attack (marked with a black line)

4.3.1 Stall

The AOA is especially important to consider as the AOA determines if the rotor is in stall. Stall is a phenomenon where there is a sudden loss of lift and the critical AOA is the angle above which this sudden loss of lift occurs [16]. Usually stall is discussed as a phenomenon that occurs for a rotor, but stall can occur for portions of the rotor as well. This is where using the BET model has an advantage over the lumped blade model. With the BET model, the portion of the rotor blade experiencing stall can be distinguished from the portion of the blade that is not experiencing stall.

The coefficient C_L is plotted against the AOA to determine the critical angle of attack as shown in Fig. 4.2. The critical AOA, α_0, is 14.21° since at this angle the coefficient of lift suddenly drops in value. The angle α_0 can then be shown on plots for C_D and C_P in Fig. 4.3. How these three coefficients behave around α_0 may play a role in the drone's response to a gust disturbance since the gust disturbance will change α of the effected rotors. The

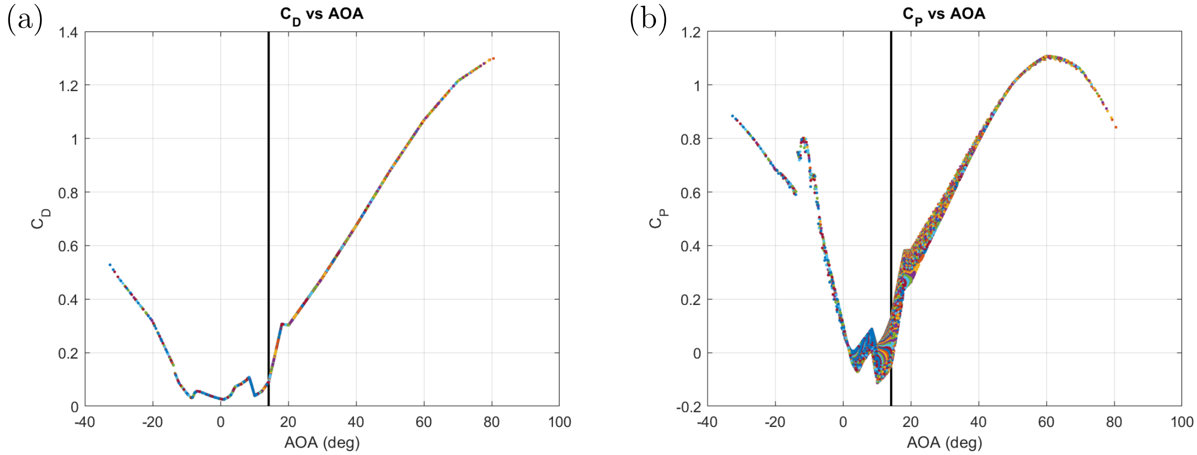


Figure 4.3: (a) Plot of Coefficient of Drag and (b) plot of Coefficient of Power vs. Angle of Attack, α, with the critical angle of attack marked with a black line

role of these coefficients and how they vary with respect to the critical angle of attack will be discussed further in the next sections.

4.4 Aim 3a: Demonstrate that simulations are sensitive to gust disturbance parameters

To determine if the gust disturbances give widely varying simulations, each parameter listed above will be varied individually and the results will be examined in this section. If small changes in any of the parameters result in significantly different simulation results, then it is not realistic to use low-level statistic tools to capture the full behavior of the drone's response when the gust disturbance is stochastic.

4.4.1 Varying Wind Magnitude

First, the wind magnitude was varied while keeping the wind pointed in a constant direction of 45° above the horizontal plane so that the wind appears to come from above the drone. The simulations were completed with wind magnitudes of 5, 10, 20, 35, and 50 m/s. The results of these simulations are shown in Fig. 4.4(a). While there seems to be a trend that

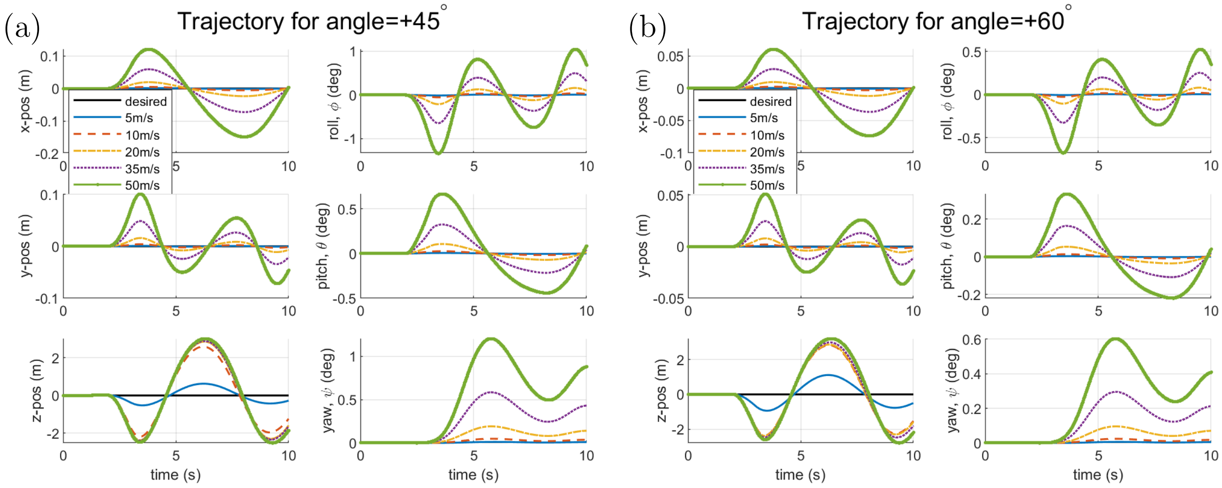


Figure 4.4: Plots showing the x, y, z, roll (ϕ), pitch (θ), and yaw (ψ) trajectories when a multi-rotor drone is hit by wind with varying magnitudes at (a) 45° and (b) 60° above the horizontal plane.

the amplitude of the oscillations in the trajectory grow with wind magnitude, the increase in the amplitude is not consistent through each increase in wind magnitude. For example, consider what happens to the maximum altitude that the drone reaches when the windspeed is increased from 5 m/s to 10 m/s and when the windspeed is increased from 10 m/s to 20 m/s. These values are displayed in Tab. 4.1. Even though the first increase is only by 5 m/s, the maximum altitude of the drone in the simulated trajectory increases from around 0.6 m to about 2.6 m. This can be compared to the second increase in windspeed from 10 m/s to 20 m/s. Despite this being an increase of 10 m/s (twice the amount of the first increase), the maximum altitude of the drone in the simulated trajectory only increases from 2.59 m to 2.85 m. The increase in windspeed thus does not give an easily scalable increase in the resulting trajectory for the altitude. This particular component of the trajectory is important to understand as it could result in the drone hitting the ground if it has not flown sufficiently far enough from the ground to avoid the wind causing it to drop down and crash into the ground.

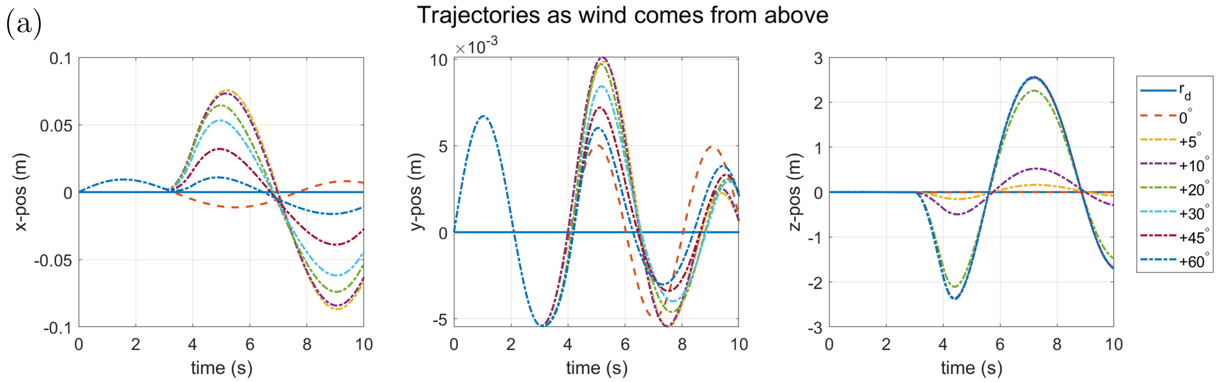
This evidence is corroborated with simulations completed for the same wind speeds but

Table 4.1: Maximum altitude recorded in simulation when the windspeed of the sudden gust disturbance is varied.

	angle = 45°	angle = 60°
5 m/s	0.6247 m	1.109 m
10 m/s	2.587 m	2.813 m
20 m/s	2.854 m	2.851 m
35 m/s	2.887 m	2.977 m
50 m/s	3.014 m	3.192 m

at a constant direction of 60° above the horizontal plane. The results of these additional simulations are shown in Fig. 4.4(b). Like when the wind angle is 45°, the increase in the magnitude of the drone's response displays some sensitivity to the increase in windspeed magnitude. Again, we can consider when the windspeed is increased from 5 m/s to 10 m/s and when it is increased from 10 m/s to 20 m/s. At a windspeed of 5 m/s, the drone has a maximum altitude of 1.1 m. This maximum altitude increases over double to 2.8 m at a windspeed of 10 m/s. By comparison, the drone's maximum altitude reaches 2.85 m when the windspeed is 20 m/s. This is a very small increase despite the windspeed doubling in magnitude. This unpredictability can also be observed in the remaining two increases in windspeed. When the windspeed increases by 15 m/s from 20 m/s to 35 m/s, the maximum altitude increases by 0.126 m. The same amount of increase when the windspeed increases from 35 m/s to 50 m/s gives a maximum altitude increase of 0.215 m. Despite the windspeed increasing by the same amount, the maximum altitude reached varies by a factor of 1.7. This unpredictability shows that the sensitivity to the windspeed is not consistent as windspeed increases.

This same variation in increases between each windspeed magnitude can be seen in the horizontal trajectories and angular rotations as well making the actual trajectories unpredictable. These results provide evidence that the drone's response to gust disturbances is



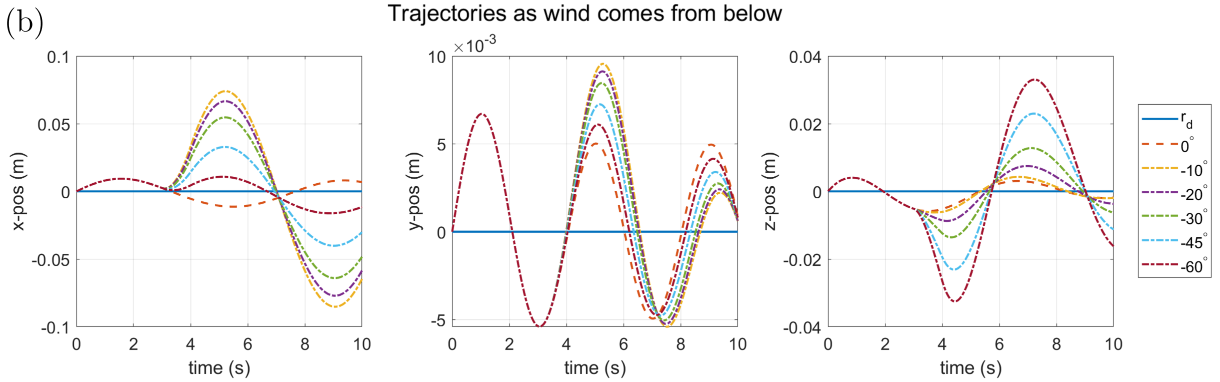


Figure 4.5: Plots showing the x, y, and z trajectories when a multi-rotor drone is hit by wind that is pointed at various angles (a) below and (b) above the horizontal plane.

sensitive to the windspeed of the gust.

4.4.2 Varying Wind Angle

The wind angle was also varied while keeping the windspeed constant at 20 m/s. The wind angle below the horizontal was varied over a large range of angles including 5°, 10°, 20°, 30°, 45°, and 60°. The results of this simulation are shown in Fig. 4.5(a). Based on these simulation results, the drone behaves differently when the wind is purely in the horizontal direction as compared to when the wind is coming at an angle from above the drone. Again there is discrepancy in the magnitude of the response depending on what angle the wind is hitting the drone at. There is a large change in maximum altitude between 10° and 20°, but almost no change above 30°. These results demonstrate that the drone is very sensitive to the wind angle when the wind is coming from above.

A similar result is seen when the same angles above the horizontal are simulated. Since the response is much smaller in magnitude when the wind comes from above the horizontal, no simulation was completed for a wind angle of 5°. The results of this simulation are shown in Fig. 4.5(b). In the horizontal response of the drone, there is a sudden spike in the change in response when the wind is applied horizontally as compared to when the wind is applied at an angle similar to the results seen in Fig. 4.5(a). However, we do not see the same pattern of maximum response magnitudes in the altitude of the drone. There appears to be a trend of increasing vertical oscillation magnitude as the angle becomes steeper. However, this pattern does not match the pattern seen in Fig. 4.5(a). The lack of a consistent trend or pattern again demonstrates that the system's gust response is sensitive to the different wind angles of the gust disturbance.

4.4.3 Rotors Affected

Finally, the effect of the number of rotors hit by the gust must be considered. This corresponds to real flight situations where the drone could be flying in an area with a lot of obstacles that could block the wind from affecting all rotors on the drone equally. This situation is taken to the extreme in this portion of the study by comparing when the wind hits all the rotors with when the wind only hits half the rotors (rotors 1-9 as seen in Fig. 2.4). This scenario imitates when part of the drone could be protected from the wind by an obstacle. This simulation was done with a windspeed of 35 m/s and the wind hitting the drone at 45° from above horizontal.

The results of the simulations for this comparison is shown in Fig. 4.6. The drone is shown to react significantly differently between these two situations. First, it's noted that when the wind only hits half of the rotors, the drone experiences significantly more horizontal motion and about half the vertical motion as when all the rotors are hit by the wind. The drone also experiences significantly more rotation in all three roll, pitch, and yaw directions. These results make sense since when the wind hits only half of the rotors, there will suddenly

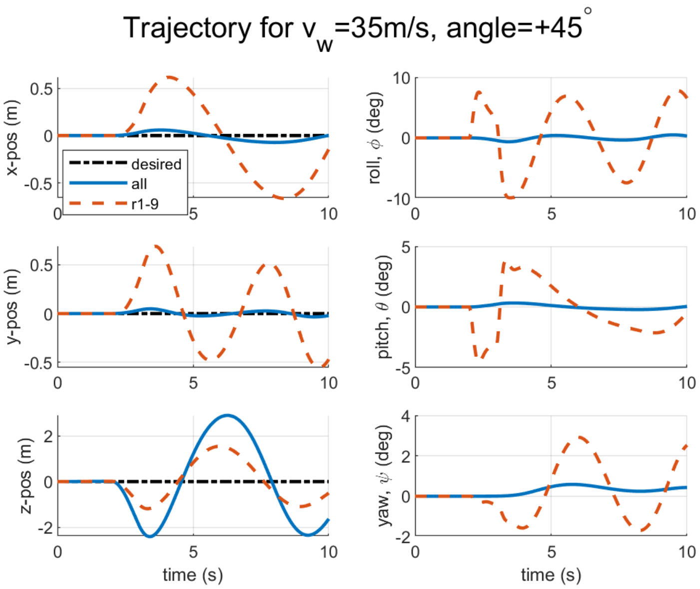


Figure 4.6: Plots showing the x, y, z, roll (ϕ), pitch (θ), and yaw (ψ) trajectories when a 35 m/s wind coming from 45° above the horizontal plane hits the drone at all or only half of the rotors (r1-9).

be an imbalance of forces and torques as the controller takes some time to respond to the gust disturbance. This imbalance of forces and torques leads to the drone rotating more in all directions. This rotation, especially in the roll and pitch direction will lead to more of the drone's thrust contributing to horizontal motion. This results in a larger response in the horizontal direction and a smaller response in the vertical direction as seen in Fig. 4.6. These results demonstrate that the drone's response is very sensitive to a wind disturbance depending on which and how many rotors are affected by the wind.

4.4.4 Discussion

Based on the results detailed in the previous three sections, the drone's response is very sensitive to the gust disturbance parameters tested even with a simple wind disturbance. While the general trends in the drone response seem to exhibit the same qualitative features, the exact response magnitudes varied significantly depending on the exact wind parameters used. This is especially apparently when examining how the drone responds to a gust disturbance acting on all of the rotors as compared to when the gust disturbance acts on only half of the rotors. The response of the drone to these two situations despite having the same gust disturbance is very different. Which of the 18 rotors are hit by the gust also has a significant effect on the drone response. For example, the drone's response when rotors 1-9 are hit by the gust will be different than when rotors 6-14 are affected even though in both cases half of the rotors are hit by the gust.

It is also important to note that this is a very simplified example of how wind could affect the different parts of the drone differently. There are many more complications that could occur depending on factors such as the types of obstacles that are blocking the wind from hitting parts of the drone or potential delays in when the gust hits different parts of the drone. While these differences are relatively easy to explain based on the physics of the system, other scenarios may not be. For example, consider a situation where the drone is attempting to fly around a large wall when a sudden gust hits it from behind the wall. It is not easy to determine how many of the rotors will be hit by the wind as the drone will be mid-maneuver. This makes it difficult to choose a single simulation to study how the drone responds in this one scenario since, as shown, the drone's response does vary significantly depending on which rotors are hit by the wind.

In conclusion, the drone's response is particularly sensitive to this aspect of the wind disturbance which makes it difficult to fully characterize the drone's response using just a handful of deterministic simulations. A better method may involve using Monte Carlo simulations so that many more possible scenarios can be tested and compared to determine the full range of potential responses to the wind.

This analysis based on the trajectories can be improved by examining other potential outputs that could be used to predict the system's response. One potential output is the energy consumption. Energy consumed is highly dependent on both the path flown as well as how the drone achieves that path through its control system. Therefore, a logical next step is to consider how the drone's response to a wind disturbance is affected by its control system as well as other factors that could potentially affect its energy consumption such as the number of rotors in stall. The number of elements in stall depends on both the input to the rotors (which depends on the controls) as well as the wind disturbance.

4.5 Aim 3b: Evaluate the effects of control gain modification when responding to gust disturbances

In addition to considering simulation results when parameters on the gust disturbances are modified, it can also be important to consider when system properties such as the control gains are changed. As discussed in Chap. 3, control gains play a role in how much energy is used to fly a trajectory without any disturbances. This subaim will evaluate the effects of control gain modification on the drone's response to a gust disturbance. If varying the control gains demonstrate trends even with a gust disturbance, then it may be possible to complete simulations for one control system and know what will change about the drone's behavior when modifications are made to that control system.

4.5.1 Variation of Control Gains

There are two ways that the control gains will be varied. First will be similar to in Chap. 3 where only the control gain on the integral control, K_I, will be varied as seen in Eq. 3.10.

The second way the control gains will be varied is by varying all gains (K_P, K_I, and K_D) in the PID controller by the same factor as shown in the equation below.

$$u = -n\left(\mathbf{K}_P(\mathbf{r}_d - \mathbf{r}) - \mathbf{K}_I \int (\mathbf{r}_d - \mathbf{r})\, dt - \mathbf{K}_D(\dot{\mathbf{r}}_d - \dot{\mathbf{r}})\right) \tag{4.4}$$

Both methods of varying the control gain will be tested to determine the drone response's sensitivity to changes in these gains when subjected to a gust disturbance. The following values of n will be considered: $n = (0.25, 0.5, 0.75, 1)$, where $n = 1$ corresponds to the original control system that was designed for this drone.

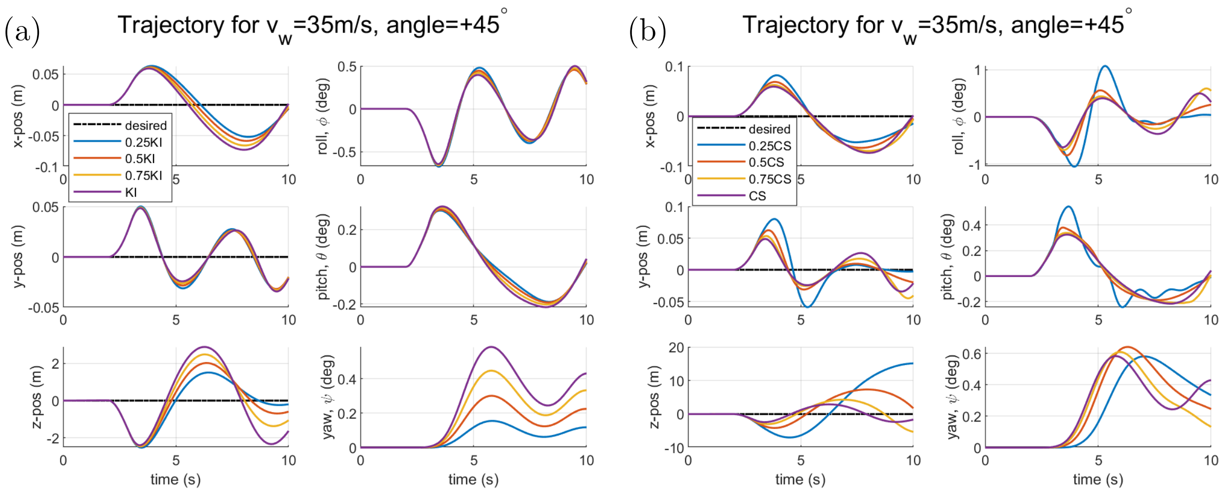


Figure 4.7: Plots showing the x, y, z, roll (ϕ), pitch (θ), and yaw (ψ) trajectories when (a) only K_I and (b) K_P, K_I, and K_D are varied when the wind hits all of the rotors.

4.5.2 Results

In this section, simulations for each variation of the control gains are completed with a gust disturbance with a magnitude of 35 m/s and originating from 45° above the horizontal. The simulations are completed for variations in only K_I and variations in K_P, K_I, and K_D. The simulation results for when the wind hits all of the rotors for all 4 values of n are shown in Fig. 4.7.

First, let's consider when only K_I is varied. The results for these simulations are shown in Fig. 4.7(a). Based on these simulations, there does appear to be a trend in responses when only one of the control gains is varied. The response appears to decrease in magnitude as the integral control decreases. This shows that while there is sensitivity of the variation in the integral control gain due to the wind disturbance, it is not very large.

Now, let's consider when K_P, K_I, and K_D are varied. It is expected that there will be a lot more variation in the responses as K_P, K_I, and K_D are changing. The results for these simulations are shown in Fig. 4.8(b). There is more variation in these responses than when only the integral control was varied. This is likely because as the control gains

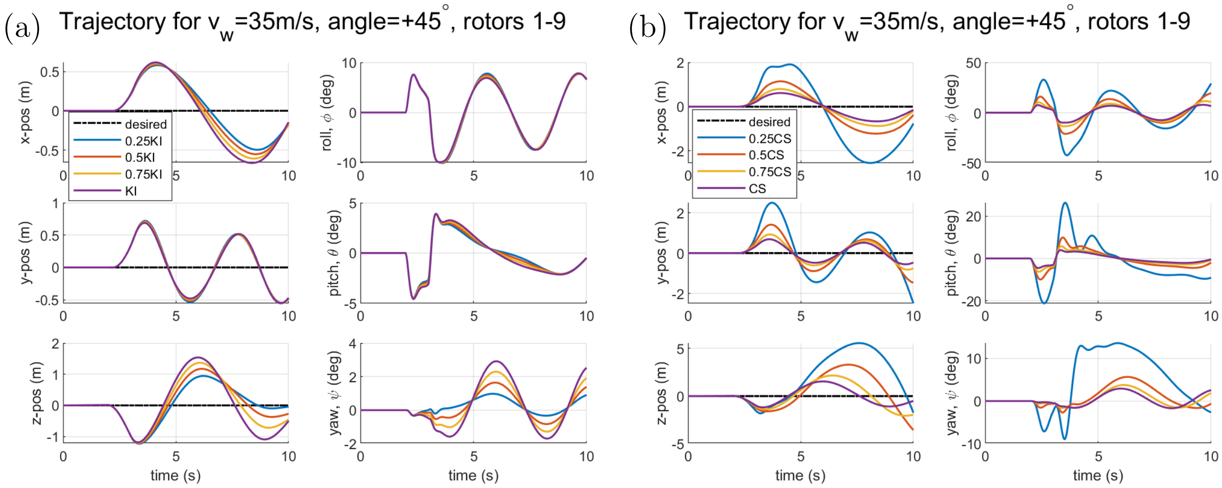


Figure 4.8: Plots showing the x, y, z, roll (ϕ), pitch (θ), and yaw (ψ) trajectories when (a) only K_I and (b) K_P, K_I, and K_D are varied when the wind hits half the rotors.

are reduced, the controller's response to proportional, integral, and derivative errors will change which could result in a very different response. This can especially be seen by the blue line representing when $n = 0.25$. The behavior of the drone when $n = 0.25$ is very different than the other values of n, especially for motion in the y-direction and pitch rotation. This demonstrates that there is a lot of sensitivity to changes in the control system when responding to a gust disturbance. Also, this is a very limited range of different control gains so further simulations are needed to confirm the behavior exhibited by the drone when either just integral control is changed or when all control gains are changed.

In Sec. 4.4.3, it was shown that the drone's response varies wildly when the wind hits all rotors as compared to when the wind hits only half of the rotors. The effect of the wind location could make the drone's response as the control gains are varied display more sensitivity as well. Therefore, simulations where the wind hits half of the rotors for each of the control gain variations considered were completed and the results are shown in Fig. 4.8.

First, consider the results for when only K_I is varied as shown in Fig. 4.8(a). When just examining the x, y, and z trajectories, there is a much larger response in the x- and y-directions and a smaller response in the z-direction compared to the results shown in

Fig. 4.7(a). The roll, pitch, and yaw trajectories also look significantly different again demonstrating that the number of rotors hit by the gust disturbance play a large role in the drone's response. However, the responses as K_I is varied still exhibits a similar trend to the simulations where the wind hits all of the rotors. This could indicate that the system displays a low sensitivity to varying the integral control.

However, when considering the case where all control gains are varied as shown in Fig. 4.8(b), the results indicate that the response of the drone once $n = 0.25$ will be very different from the drone's response for all other values of n. This difference can be seen in particular in the x-, pitch, and yaw trajectories. Similar to when the wind affects all of the rotors, $n = 0.25$ results in a very different response from all other values of n. This would indicate that drone's gust response is particularly sensitive to when all of the control gains are varied.

4.5.3 Discussion

It is important to also consider other windspeeds or wind angles since only one was considered above.the same simulations for a wind angle of 60° are also conducted for when all the control gains are varied to see if the response does drastically change for a small change in wind angle. This particular set of parameters was checked since large angle does tend to result in large response magnitude as demonstrated in Aim 3a. The results are shown in Fig. 4.9.

Comparing the plots in Fig. 4.9 to those in Figs. 4.7 and 4.8, it is clear that the responses and trends look quite different. For the trajectories shown in Fig. 4.9(a) and Fig. 4.7(b), these differences are mainly seen in the pitch trajectories and the trends in the y- and z-trajectories. In particular in the z-trajectories, with this new angle, the original control system (CS) no longer exhibits the smallest response. Similarly for the trajectories shown in Figs. 4.9(b) and 4.8(b), there is a large difference in the responses in all of the trajectories and the trends shown. For example, the yaw trajectories take on an entirely different shape in Fig. 4.9(b) as compared to Fig. 4.8(b).

While the results for the simulations where only the K_I is varied by n shows that there is a trend for how the response changes as n increases or decreases, this trend is not seen

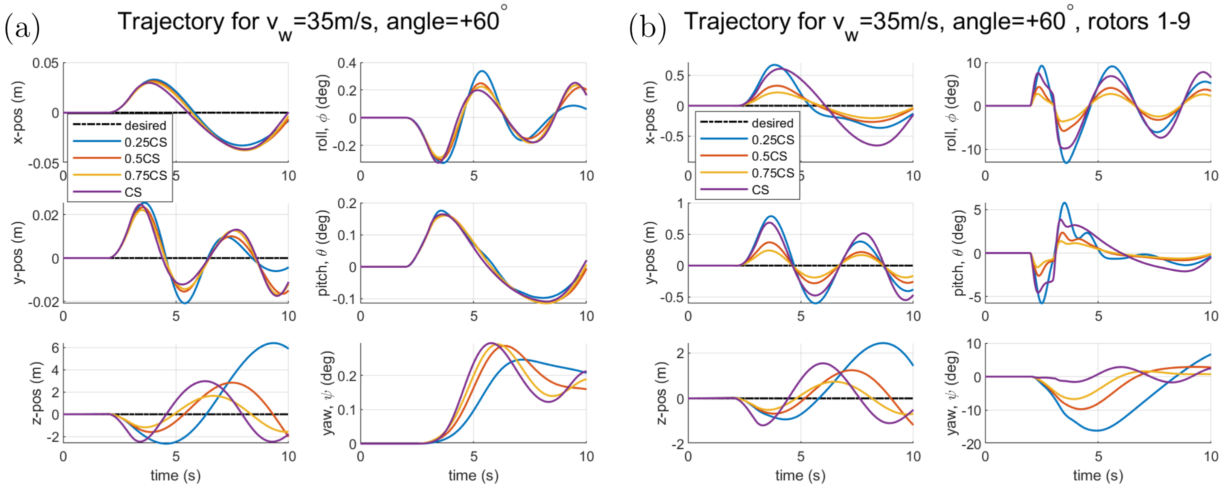


Figure 4.9: Plots showing the x, y, z, roll (ϕ), pitch (θ), and yaw (ψ) trajectories when K_P, K_I, and K_D are varied when the wind hits (a) all and (b) half the rotors.

when the control gains for all three proportional, derivative, and integral controls are varied by the same factor. In addition, the additional simulations at a different angle shows that any potential trends could also depend on the wind angle. This implies that there will not be a pattern in the resulting trajectories that could be used to predict how the drone would respond for different variations on the control system. This makes it difficult to know when would be appropriate to use deterministic gust models to characterize the drone's response to a sudden gust disturbance when considering variations on the control system. Therefore, even though some trends in response were observed and some changes to the control system demonstrated low sensitivity, collectively, these results demonstrate that the system's response is sensitive to system changes as well as being sensitive to the gust disturbance parameters as shown in the previous aim.

These results also indicate that deterministic gust modeling will not be sufficient to fully capture the drone's gust response to compare different control systems. In this section, variations on a PID controller were compared. These variations where shown to result in many different responses to the same gust disturbance, especially for $n = 0.25$ and when the wind hit only half of the rotors as opposed to all of them. This would imply that to compare two very different controllers, such as a PID controller and a model-predictive controller, then deterministic simulations will not be sufficient to fully characterize the differences between the two controllers. In addition to characterizing how the different controllers affect the trajectory, it is also important to consider the energy consumption.

4.6 Aim 3c: Demonstrate sensitivity of total energy consumption to gust disturbances

So far, Aim 3 has focused on determining the sensitivity of the drone's response to various parameters using trajectory since it is very important for large, multi-rotor drones to be able to track a specified trajectory while avoiding obstacles to complete their tasks. One other potential area of concern with large multi-rotor drones is carrying enough energy to complete these tasks. First, it is recognized that energy consumption is nominally affected by several parameters: flight path, disturbances, and control gains. Previously in Chap. 3, a variety of control systems are compared to demonstrate that the energy does depend on the control gains. It was also shown that varying the control gains can drastically change how the drone responds to a gust disturbance in Sec. 4.5. This will also have an effect on the power using during flight and the total energy consumed.

In addition to the controller having an effect on the energy consumed to fly a trajectory, the angle of attack on each element on each rotor will also have an effect since the power required to fly a trajectory is also related to the angle of attack and, consequently, whether or not the rotor is in stall. To add to further complications to this, portions of one rotor blade can be in stall while others are not since the rotor blade does not have a constant pitch angle as seen in Fig. 2.5. Thus, the number of rotor elements in stall will also be considered as a potential way to understand the total energy consumption of the drone.

4.6.1 Predictions Using Rotor Speed

First, the power and total energy consumption will be examined for some of the trajectories considered in Secs. 4.4 and 4.5. First, consider the simulation from Sec. 4.4.3 as this set of

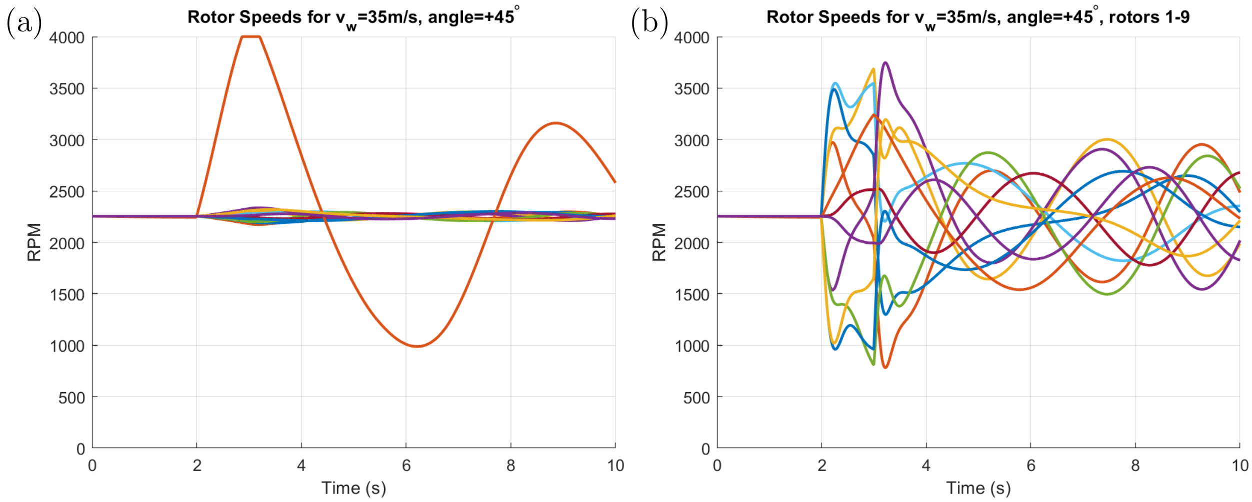


Figure 4.10: Plots showing the rotor speeds of all 18 rotors when (a) all rotors and (b) half of the rotors of a multi-rotor drone are hit by wind at 35 m/s originating from 45° above the horizontal.

simulations displayed the most variation and has the highest likelihood of seeing a difference in energy consumption.

The first step is to ensure that the changes in rotor speed correspond to the drone's trajectory in the different wind scenarios (see Fig. 4.10). In Fig. 4.10(a), the majority of the rotors exhibit very little variation while a subset of the rotors exhibit very large variations in rotor speed. Further inspection reveals that these rotors with large variations all are from the thrust group (as seen in Fig. 2.16(a) in Sec. 2.5.1). This response makes sense since the majority of the drone's deviation occurs in the z-direction when the wind hits all of the rotors. Examining Fig. 4.10(b), all rotors appear to show similar variations in the rotor speed when the drone is hit by the wind at rotors 1-9. Again this makes sense since the drone has motion in all three translational directions as well as all three rotational directions so there should be a response to the disturbance in all four rotor groupings.

Next, the power used by the drone in the two simulations is shown in Fig. 4.11. To compare the power used in each of these situations, a control case is necessary. The control case considered is when there is no wind hitting the drone and the controller is maintaining

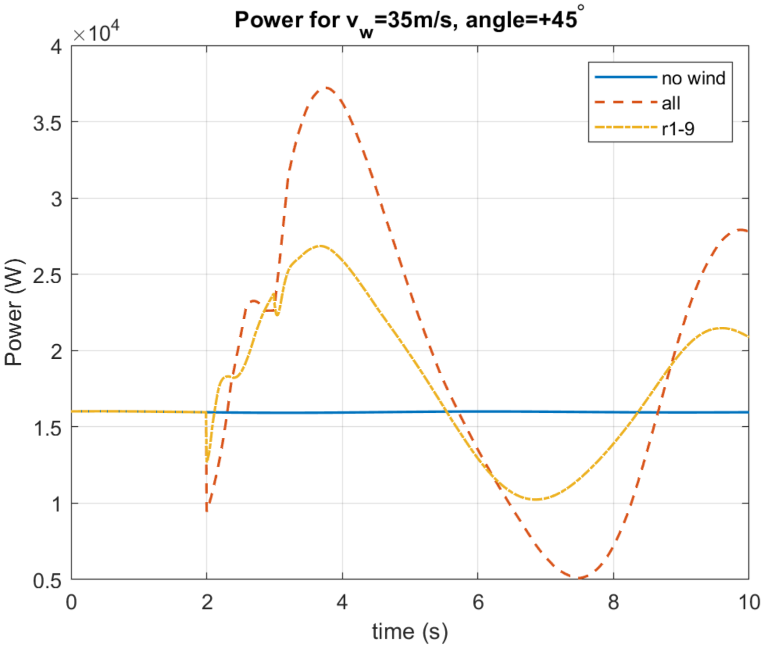


Figure 4.11: Plot showing the power used when all rotors and half of the rotors of a multi-rotor drone are hit by wind at 35 m/s originating from 45° above the horizontal.

a hovering position.

Initially, it was hypothesized that the simulation where only half of the rotors are hit by the wind would use more power since the drone moved in all three x-, y-, and z-directions. As seen in Fig. 4.11, this is not the case. The simulation where all rotors are hit by the wind uses the maximum amount of power between the two simulations. Further analysis requires an examination of Eq. 4.3 for calculating power. In this equation, the power depends on the relative velocity of the rotor element as well as the rotational speed of rotor itself. The relative velocity depends mainly on the rotational speed so $V_{rel} \propto \omega$. Therefore, $P \propto \omega^3$ so any changes in the rotor speed will have a cubed effect on the power. By contrast, other factors such as the C_P have a linearly proportional relationship with power.

Comparing Fig. 4.11 to Fig. 4.10 shows that the drone reaches a much higher max rotor speed when the wind hits all of the rotors. Even though only the six rotors in the thrust group reach this max rotor speed, because $P \propto \omega^3$, then the effect of this difference in rotor

Table 4.2: Percent difference between the energy used to respond to a gust hitting all rotors and only half of the rotors for winds hitting the drone at 45°. Positive indicates that drone uses more energy to fly when there is wind.

Windspeed (m/s)	Energy, all rotors (kJ)	% Diff with no wind case	Energy, r1-9 (kJ)	% Diff with no wind case
0 (no wind)	159.74	–	159.74	–
5	151.68	-5.04%	155.05	-2.94%
10	185.81	16.32%	170.03	6.44%
20	188.24	17.84%	176.64	10.58%
35	188.61	18.07%	176.93	10.76%
50	189.52	18.65%	177.28	10.98%

speed is cubed. This is what leads to the max power level at around 3.6 s in the simulation.

Comparing the energy used to the case when there is no wind disturbance makes it possible to quantify the cost of the gust disturbance. The energy difference is then calculated as follows

$$\%\ \text{Energy Difference} = \frac{E_{sim} - E_{control}}{E_{control}} \times 100\% \tag{4.5}$$

where E_{sim} is the total energy used to fly the drone when the wind hits it either on all or half of the rotors and $E_{control}$ is the total energy used to fly the drone when there is no wind acting on the drone. The energy comparisons for when the wind hits the drone at 45° are shown in Tab. 4.2. The energy comparisons for when the wind hits the drone at 60° are shown in Tab. 4.3.

From the results shown in Tab. 4.2, it appears that in most cases the drone uses more energy to respond to a gust disturbance that affects all of the rotors than to respond to a gust

Table 4.3: Percent difference between the energy used to respond to a gust hitting all rotors and only half of the rotors for winds hitting the drone at 60°. Positive indicates that drone uses more energy to fly when there is wind.

Windspeed (m/s)	Energy, all rotors (kJ)	% Diff with no wind case	Energy, r1-9 (kJ)	% Diff with no wind case
0 (no wind)	159.74	–	159.74	–
5	157.62	-1.33%	156.27	-2.17%
10	188.32	17.89%	175.78	10.04%
20	188.27	17.86%	176.67	10.60%
35	189.21	18.45%	176.97	10.79%
50	190.57	19.30%	177.42	11.07%

disturbance that affects half of the rotors. The percent difference in energy is not consistent across the different windspeeds even for the same load case. For example, when the windspeed is 5 m/s, the drone actually uses less energy to respond to the gust disturbance while for all other windspeeds, the drone uses more energy to respond to the gust. In addition, energy consumption appears to increase only marginally when the windspeed is 10 m/s or higher despite the windspeed changing by 10 m/s or more in each increment. This is not the same as when only half the rotors are hit with wind. In this case, there is a large jump in energy used until 20 m/s when the changes in energy usage are marginal despite the windspeed increasing by 15 m/s in each increment above 20 m/s.

These inconsistencies in the changing energy consumptions are also seen when the drone is hit by gusts at 60° as seen in Tab. 4.3. while for this angle, the percent difference in energy is negative for a windspeed of 5 m/s as well. The energy differences do not show the same trends as seen in Tab. 4.2. In addition, there is no obvious relation between rotor

Table 4.4: Percent difference between the energy used to respond to a gust hitting all rotors and only half of the rotors for variations in K_I. Positive indicates that drone uses more energy to fly when there is a gust.

n	No Wind Case: Energy (kJ)	Energy, all rotors (kJ)	% Diff with no wind case	Energy, r1-9 (kJ)	% Diff with no wind case
1	159.74	188.61	18.07%	176.93	10.76%
0.75	159.75	184.95	15.77%	174.72	9.37%
0.5	159.76	182.77	14.40%	173.90	8.85%
0.25	159.77	182.43	14.18%	173.97	8.88%

speeds and energy despite there being some correlation between rotor speed and power due to Eq. 4.3.

Controller Gain Variations

This same analysis is done to determine if the energy used is sensitive to the variation in the controller as a continuation of the analysis in Sec. 4.5. The summarized results indicating the total energy required to fly a specified trajectory as n is varied on the integral control gain K_I is shown in Tab. 4.4. Like before, the case where there is no wind is used as the control case against which to compare the energy calculations.

The total energy usages as K_I is varied by n are shown in Tab. 4.4. For both wind loading cases, the energy generally decreases as n decreases when half or all of the rotors are hit by the gust disturbance. The only exception is when $n = 0.25$ for the wind loading case when only half of the rotors are hit by the wind. The total energy is slightly larger for $n = 0.25$ than for $n = 0.5$. The same set of simulations are completed for when all control gains are varied by n as shown in Eqn. 4.4. The total energy usages as all control gains

Table 4.5: Percent difference between the energy used to respond to a gust hitting all rotors and only half of the rotors for variations in all control gains. Positive indicates that drone uses more energy to fly when there is a gust.

n	No Wind Case: Energy (kJ)	Energy, all rotors (kJ)	% Diff with no wind case	Energy, r1-9 (kJ)	% Diff with no wind case
1	159.74	188.61	18.07%	176.93	10.76%
0.75	159.76	184.25	15.33%	171.65	7.44%
0.5	159.99	207.27	29.55%	161.09	0.69%
0.25	159.58	291.59	82.73%	168.24	5.43%

are varied by n are shown in Tab. 4.5. As n decreases, the energy used to respond to a gust that hits all rotors increases dramatically. This is exactly the opposite trend that is seen previously when only K_I is varied. In contrast the energy used to respond to a gust on only half of the rotors exhibits no constant trend as n decreases. This unpredictability again demonstrates that energy is very sensitive to variations in the control system. It is also important to keep in mind that these values are also for fairly short simulations. These results could be further amplified for longer simulations.

4.6.2 Predictions Using Stalled Elements

Another potential way to demonstrate the sensitivity of energy to variations in the gust disturbances is to consider the number of elements in stall for each of the 18 rotors since C_P varies with angle of attack and C_P does have an effect on the total power used by the drone while flying. Again, the same trajectories from Sec. 4.4.3 will be considered. The number of rotor elements in stall are plotted as a function of time to compare to trends in power usage.

In Fig. 4.12, the number of elements in stall reach a minimum in the simulation when

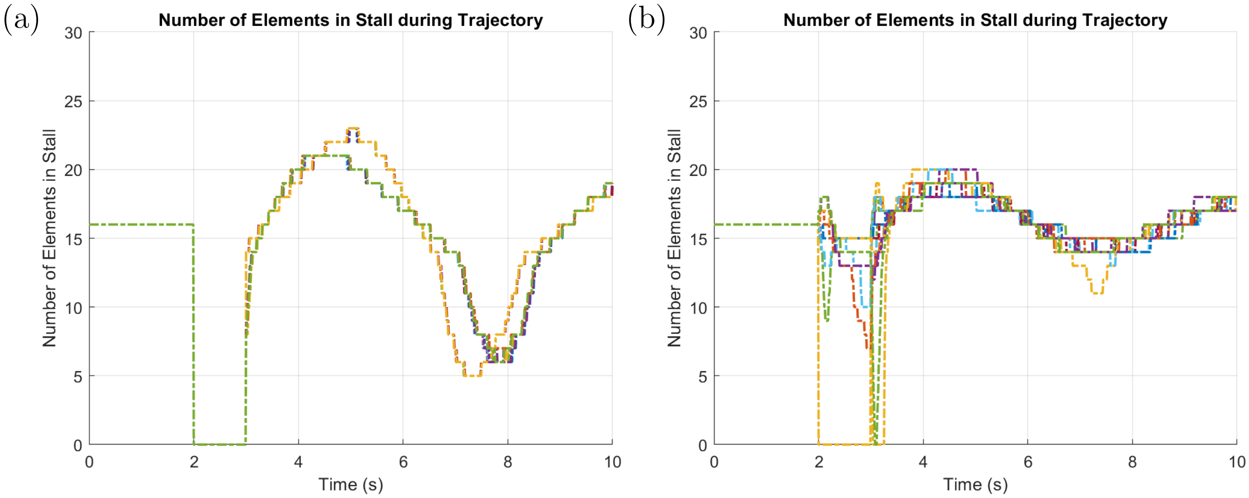


Figure 4.12: Plots showing the number of elements in stall for all 18 rotors when (a) all rotors and (b) half of the rotors of a multi-rotor drone are hit by wind at 35 m/s originating from 45° above the horizontal.

the wind hits the drone. The only difference is that when the wind hits all of the rotors, all of the rotor elements go out of stall as compared to when the wind only hits half of the rotors where some elements remain in stall. Based on Fig. 4.3(b), this would imply that for certain angles of attack below the critical angle of attack, the coefficient of power is much smaller. This would imply that the power consumed at this time in the simulation is less. This is confirmed in Fig. 4.11. When the number of elements in stall increases the total power consumed at that time also increases.

Despite this seeming like a potential trend, there are still certain features that do not appear to correlate between the two plots. For example, consider the simulation between 2 and 3 seconds in Fig. 4.11 and Fig. 4.12(a). The number of elements in stall for each rotor does not change, but the power increases dramatically during this. To understand why this is happening, the angle of attack of each element of one rotor needs to be examined. The angle of attack of each element in rotor 1, which is in the thrust group, is shown in Fig. 4.13.

Between 2 and 3 seconds in Fig. 4.13, the angle of attack is between $-5°$ and $5°$ which on Fig. 4.3(b) shows that the coefficient of power varies anywhere between 0 and 0.3 but with

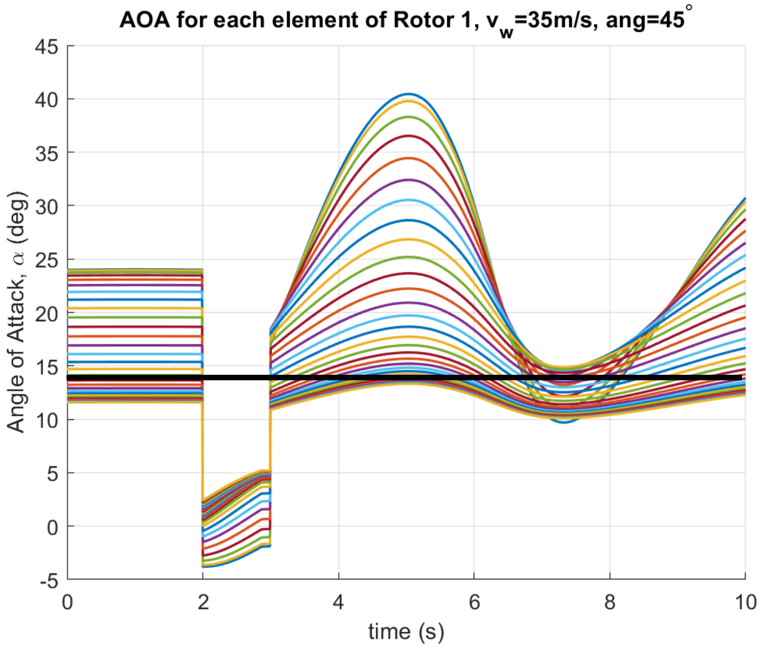


Figure 4.13: Plot showing the angle of attack of each element in rotor 1 when all rotors are hit by wind at 35 m/s originating from 45° above the horizontal. The black line indicates the critical angle of attack.

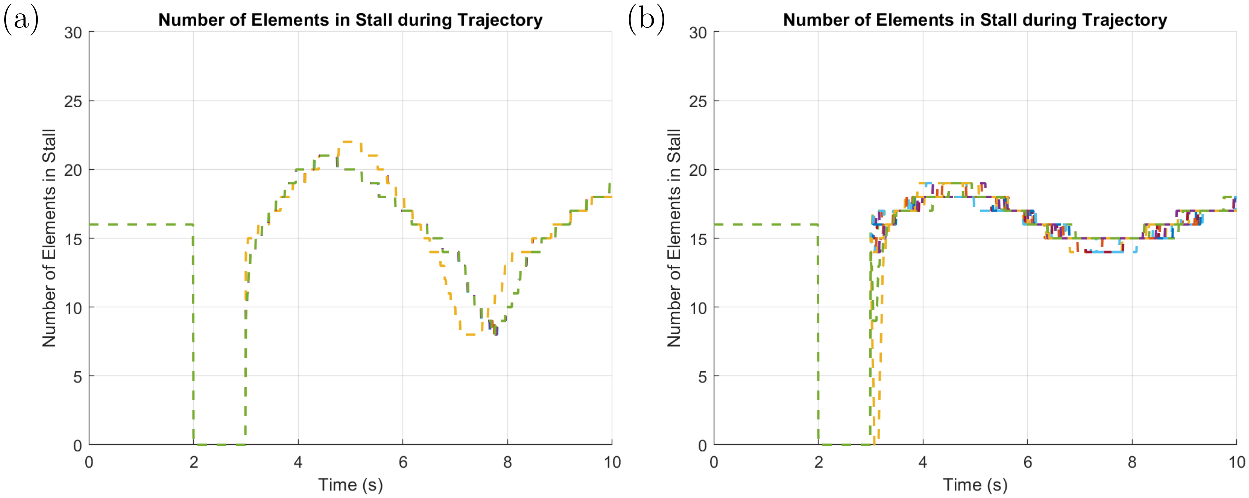


Figure 4.14: Plots showing the number of elements in stall for all 18 rotors when (a) all rotors and (b) half of the rotors of a multi-rotor drone are hit by wind at 10 m/s originating from 45° above the horizontal.

no consistent trend. In addition, it has been shown that rotor speed plays a much larger role in determining the total energy required to fly the drone than the coefficient of lift. It can be confirmed whether or not the number of elements in stall correlates to the total energy consumed by examining two more cases for when the windspeed is 10 m/s at an angle of 45°. The results for this simulation is shown in Fig. 4.14.

As a reminder, in Tab. 4.2, it is noted that when the windspeed is 10 m/s that the drone requires 15 kJ more energy to fly when wind hits all of the rotors as compared to when wind hits half of the rotors. This corresponds to about 10% difference in energy used with respect to the no-wind case. In general the two plots in Fig. 4.14 exhibits similar trends to the plots in Fig. 4.12, but results in a different percent difference in energy consumed. Examining the angles of attack for elements in rotor 1 for this simulation, as seen in Fig. 4.15, reveals a very similar result to Fig. 4.13.

The main difference between these angles of attack and the ones in Fig. 4.13 is that when the wind hits, the angles of attack are reduced to angles between 5° and 12°. Again, when examining Fig. 4.3(b), there is no clear trend in C_P over this range of angles of attack.

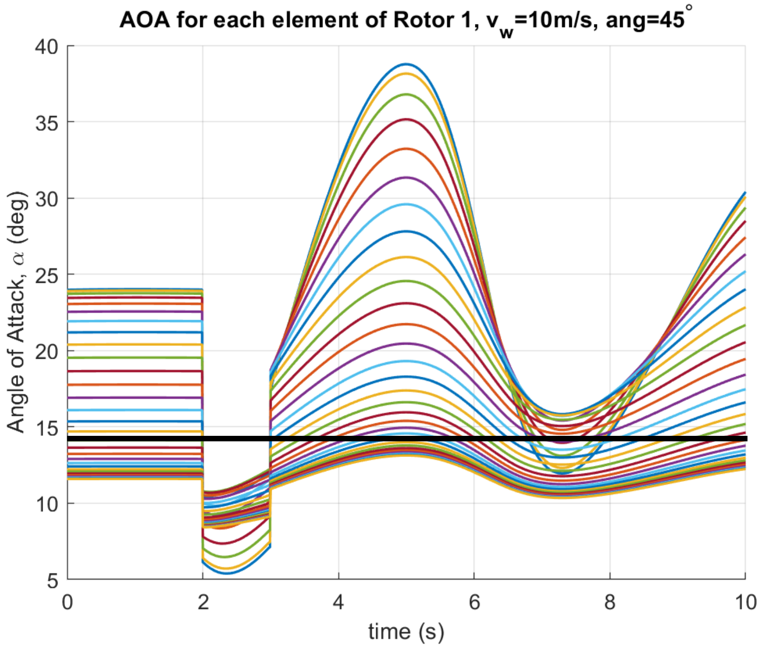


Figure 4.15: Plot showing the angle of attack of each element in rotor 1 when all rotors are hit by wind at 10 m/s originating from 45° above the horizontal. The black line indicates the critical angle of attack.

In addition, this simulation uses 185.81 kJ as opposed to the simulation for a windspeed of 35 m/s which uses slightly more at 188.61 kJ. However, the plots of number of elements in stall look too similar to to make it possible to use those plots to predict which gust disturbance will require the more energy for the drone to correct its trajectory. Even with the plots of the angle of attack of each rotor element, it is very difficult to directly correlate it to how much energy is used with the angle of attack since there are 30 elements for each of the 18 rotors of the drone.

All of these results indicate that energy consumed during flight is very sensitive to variations in the gust disturbance even when the variations are deterministic, and it is difficult to directly correlate the total power and energy consumed during flight based on factors such as angle of attack and rotor speed. This is likely because there are many factors other than just the angle of attack of the rotor blade element that affects the total power and energy consumed including the rotor speed.

4.7 Conclusions

The purpose of this chapter is to demonstrate that it is not possible to fully characterize how a drone will respond to an unknown gust disturbance through deterministic gust modeling simulations because the drone's response is very sensitive to the gust disturbance. In these simulations, a very simple gust disturbance is used where its windspeed, wind angle, and which rotors it affected arere varied. Each of these variations results in responses that are not either quantitatively or qualitatively predictable. In particular, the results demonstrate that which rotors the wind affected results in large differences in the trajectory. This is an especially compelling result since the simple gust disturbance and minor changes in the three parameters resulted in very different flight behaviors for the drone.

The next aspect considered is if the changes in the drone's responses due to changes in the controller are easily predictable for small variations in the control system. The control system used is the same PID controller used in Chap. 3. For this portion of the study, the control gains of the PID controller are varied by a factor n that ranges from 0.25 to 0.75.

The results demonstrate that for different control gains, the simulations produce results that show varying resulting trajectories. Specifically, if only the integral control gain is varied, the resulting trajectories qualitatively look similar to the trajectories from simulations where the integral control gain is not varied. Conversely, this is not true when all of the control gains of the PID system are varied. This is likely due to the fact that when all the control gains of a PID controller are adjusted, then the nature of the controller also changes. This means that in order to compare two actually different controllers (such as a PID controller and a model predictive controller), deterministic simulations are not sufficient to fully characterize the difference in response behavior between the two controllers since the simulation results are also sensitive to variations in the system when a gust disturbance acts on the system.

Finally, energy consumption during a flight trajectory was also considered since sensitivity in the trajectory could also imply sensitivity in the energy calculations, and understanding how energy used changes with the gust disturbance is also important for large multi-rotor drones. Two potential ways to predict the total energy used are considered: rotor speeds and angle of attack. First, it is shown that power is proportional to rotor speed cubed which means that any differences in rotor speed result in a very large difference in power. This is seen in the first example when comparing the power consumption when all the rotors are hit by the wind versus when only half of the rotors are hit by the wind. When the wind hits all of the rotors, the rotor speeds for the thrust group vary significantly while all the other groups have very little variation in their rotor speeds. When the wind only hits half of the rotors, the rotor speeds for all groups vary. Despite more of the rotors exhibiting higher rotor speeds, the drone uses more energy to respond to wind hitting all of the rotors. This demonstrates that the rotor speed, and thus power and energy, are highly sensitive to variations in the gust disturbance. The unpredictability of the energy consumption is also demonstrated by examining the energy required to fly using different control systems. The same control system variations as in Sec. 4.5 are used for comparison. It is demonstrated that when K_I is varied, there is no consistent trend in how the energy varies when the wind hits all of the rotors and when the wind hits half of the rotors. This is confirmed when examining the same energy calculations for when all the control gains are varied.

The number of rotor elements in stall is also considered as a potential way to predict energy usage. In this portion of the study, it was shown that while the number of elements in stall does seem to correlate with the power consumption, it does not appear to correlate with the total energy usage for the entire trajectory. In addition to checking the number of rotor elements in stall, the angle of attack for each rotor element was also considered. While these angles of attack do correlate with the number of elements in stall, they do not provide an easy way to understand the total energy consumption since the angles of attack for when the windspeed is 35 m/s looks similar to the angles of attack for when the windspeed is 10 m/s but the simulations at these two windspeeds result in different energy usages. Therefore, it is difficult to know how much energy is consumed based off of rotor speed or stall because there are 30 elements in each of the 18 rotors of the drone and all of these elements in addition to the wind disturbance will affect the total energy consumed.

In conclusion, the results of this study show that deterministic gust modeling and simulation are not sufficient to characterize how a drone and its controller will respond to any gust disturbance since the drone's response is very sensitive to changes in the gust disturbance. Therefore, to fully understand how a drone responds to a gust disturbance, stochastic simulations such as Monte Carlo simulations are necessary. Future work implementing stochastic gust models will be discussed in the next chapter.

Chapter 5

SUMMARY AND FUTURE WORK

The goal of this research is to evaluate current modeling choices for quadcopters and find ways to improve upon these modeling choices when applied to larger multi-rotor drones. Specifically, modeling choices that could affect the final energy calculations are considered.

5.1 Summary

Aim 1 examines the open-loop drone system and modeling choices that will affect the system dynamics and as a result the calculated energy consumption. I demonstrate in Aim 1a the importance of including a heavy suspended payload in the drone dynamical system when it is present in the real system [28]. This is because both the location and weight of the payload change the dynamics and stability of the system. In Aim 1b, I demonstrate the effect of different rotor blade aerodynamic models on drone dynamics and show the importance of using accurate modeling in both the drone dynamics and the rotor aerodynamics [29, 30]. Finally, in Aim 1c, I demonstrate that grouping rotors presents a new opportunity to optimize for energy consumption as some rotor groupings will inherently have a lower energy cost than others [29, 30].

Aim 2 examines the energy cost of flying a multi-rotor drone using a controller as compared to flying a multi-rotor drone without a controller. I develop a framework for calculating the energy cost of a control system to fly a specified trajectory by comparing the open- and closed-loop energy consumption. This framework is used throughout this aim. In Aim 2a, I demonstrate that for a PID controller, the closed-loop system used significantly more energy than the open-loop system. In Aim 2b, I compare the energy costs of two different controllers. I demonstrate that the PD controller uses significantly less energy than the PID controller [31]. I also completed more simulations by varying the magnitude of the integral control gain to characterize the trade-off relationship between trajectory tracking and energy consumption for the integral control portion of the controller. I found that there is a different trade-off relationship between tracking and energy depending on the type of trajectory that the drone is trying to track. This trade-off relationship can be found between any two controllers, not just a PD and PID controller, and these results demonstrate that this is a step in the design process that can help choose the most appropriate controller for a specific task.

In Aim 3, I demonstrate the importance of modeling gust disturbances as a stochastic disturbance rather than a deterministic disturbance as much of the research on quadcopters has done thus far. In Aim 3a, I consider a simple gust disturbance and demonstrate that even small changes in the windspeed, wind angle, and which rotors the wind hits will result in different simulation results. These differences are especially apparently when the affected rotors are different. As each of these parameters can be represented as random variables, these results suggest that the drone's response is very sensitive to a random gust disturbance. In Aim 3b, I demonstrate that changes in a drone's response to a gust disturbance is also sensitive to variations in control system. This implies that testing variations in control system also may require stochastic simulations in order to fully understand how the drone will respond to a gust due to these changes. Finally, in Aim 3c, I demonstrate that in addition to the resulting trajectory being sensitive to changes in the gust disturbance and control system, total energy consumption is also sensitive to these factors. I demonstrate this by examining the energy consumption for different simulations and analyzing them using rotor speed and stall. Based on all these results, I conclude that in addition to the drone's response trajectory, its energy is also sensitive to the deterministic gust disturbance. These results indicate that stochastic gust modeling is necessary in order to fully characterize how a drone and its control system will respond to a random gust disturbance both trajectory-wise and energy-wise.

5.2 *Future Work*

In the following sections, I will outline some potential future directions to take this work. First, now that I have shown that stochastic gust modeling is necessary to fully characterize drone response to a sudden gust using a simple wind disturbance, a deterministic gust model to test these conclusions made above with more complex gust disturbances. This can be compared to a stochastic gust model built using comparable wind parameters. The results of both sets of simulations can be compared to determine if any new response characteristics appear due to the use of a stochastic gust model instead of a deterministic one. Potential deterministic and stochastic gust models are discussed in the following sections.

In addition, while outside the scope of this work, further examination into the specific phenomenon of how stall occurs under the two different gust models would be interesting to pursue in the future. There is a possibility that the deterministic and stochastic models may cause stall to occur in different situations.

Finally, I will lay out some potential extensions of this work into experimental work and real-life applications in package delivery and education tools.

5.2.1 Trajectory Parameterization

Rather than using an open-loop trajectory as done in Aim 2, new desired trajectories can be generated using a trajectory planning algorithms such as the two-function parameterization method described in [5]. To create new, longer desired trajectories, write the desired xyz positions as a parametric function $\mathbf{R}(\lambda) = [x(\lambda), y(\lambda), z(\lambda)]$ that depends on a monotonically increasing function $\lambda(t)$ which will dictate the time evolution of the trajectory [5]. $\lambda(t)$ can then be adjusted in order to change the trajectory's time dependency, and $\mathbf{R}(\lambda)$ can be adjusted to change the trajectory's shape. An example of this parameterization is shown in Fig. 5.1.

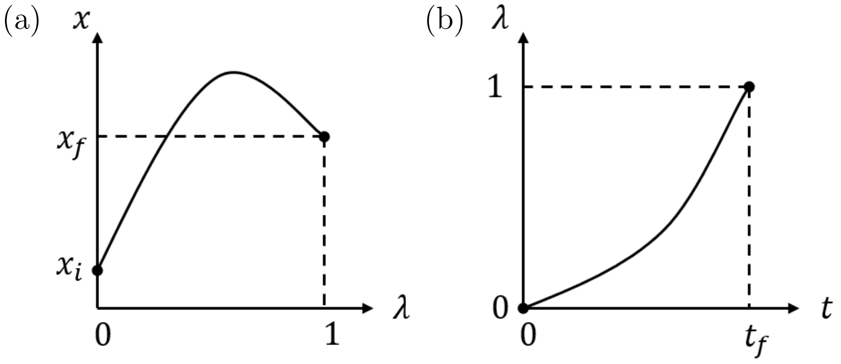


Figure 5.1: Example of how (a) the x-component of trajectory $\mathbf{R}(\lambda)$ and (b) $\lambda(t)$ are parameterized

5.2.2 Deterministic Gust Modeling: Dryden Wind Model

Many different ways to model gusts deterministically have been used in research on multirotor drones [18, 23, 70, 74]. These methods all use slightly different deterministic models. One potential deterministic gust model is the Dryden wind model [70]. The Dryden wind gust model defines the wind velocity as a sum of sinusoids as shown in Eq. 5.1 below

$$v_w(t) = (v_w)_0 + \sum_{i=1}^{n} a_i \sin(\Omega_i t + \phi_i) \tag{5.1}$$

where $(v_w)_0$ is the ambient wind velocity and the sinusoids represent variations in the ambient wind velocity. The frequencies Ω_i and phase shifts ϕ_i are determined randomly, and the magnitude a_i are determined using the power spectral density (PSD) for vertical and horizontal winds. The PSDs for vertical and horizontal winds vary so different equations as outlined in [70] must be used to find a_i.

Simulations can then be conducted using this model to create a gust disturbance to the drone system. The resulting simulations and energy calculations can be used as a benchmark to compare this deterministic model to the stochastic model.

In addition to using these results as a benchmark, some additional simulations can be done to compare the lumped blade and blade element theory (BET) models. The lumped blade model is commonly used in research on the effect of wind disturbances [18, 23, 70]. Therefore it is of interest to see what difference using the BET model for the rotor aerodynamics makes with respect to gust response.

5.2.3 Stochastic Gust Modeling: Weibull Distribution

For stochastic gust modeling, techniques can be borrowed from wind turbine simulation research where the gust disturbance is modeled using the Weibull distribution function as described in [61]. The resulting simulation and energy calculations can be compared against the benchmark developed using the Dryden Gust Model to determine what phenomena in the response are captured by the deterministic model and what phenomena are unique to the stochastic model.

The Weibull distribution is a family of curves that are commonly used to represent the wind speed probability density function in wind energy research. The Weibull distribution function is expressed as follows [61]

$$P(v < v_i + dv) = P(V < 0)\left(\frac{k}{c}\right)\left(\frac{v_i}{c}\right)^{k-1}\exp\left[-\left(\frac{v_i}{c}\right)^{k}\right]dv \tag{5.2}$$

where c and k are parameters for the Weibull function, v is the windspeed, v_i is a chosen windspeed, dv is an increment in wind speed, $P(v < v_i + dv)$ is the probability that wind speed is bounded by v and $v + dv$, and $P(v > 0)$ is the probability that the wind speed is nonzero and positive. The parameters can be estimated using time-series data or frequency distribution data on wind speeds in a given region [61].

5.2.4 Applications to Quadcopters

In addition to comparing simulation results between the Weibull and Dryden gust models. It would also be interesting to explore applications of the simulation-based research. First, each of the results in this dissertation need to be verified in experiment by using a quadcopter. Some aspects, such as the rotor grouping cannot be tested using a quadcopter, but others, such as the energy calculation framework developed in Aim 2 can be. In addition, it would be interesting to implement stochastic gust disturbance modeling for a quadcopter drone model and test how well the stochastic simulations predict experimental response in comparable windy environments.

In addition to verifying the work done during my dissertation, it would also be interesting to explore other applications of this work. In particular, if this work, particularly the use of stochastic gust modeling, could be used to develop drones that are able to fly to remote locations in inclement weather. This has many potential applications such as delivering goods and supplies to remote villages, emergency supplies to isolated outdoors-people, and scouting remote regions for conservation work. If using stochastic gust models proves to be successful at characterizing drone response in gusty and hazardous conditions, then these models can be used to improve upon controllers for drones that would be developed for this purpose. This would increase their chance at success in difficult missions such as the ones described previously.

Finally, it would also be interesting to expand this work on multi-rotor drones into education related purposes by developing an inexpensive toolkit that allows schools from low-income neighborhoods to teaching engineering and science in a more hands-on way using small kits and the models developed in this dissertation. These kits would foster excitement and curiosity for science and engineering using small quadcopters and potentially involve students running simulations and then trying to recreate the results in experiment or vice versa.

BIBLIOGRAPHY

[1] Michael C. Achtelik, Jan Stumpf, Daniel Gurdan, and Klaus-Michael Doth. Design of a flexible high performance quadcopter platform breaking the MAV endurance record with laser power beaming. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5166–5172. IEEE, 2011.

[2] Shaimaa Ahmed, Amr Mohamed, Khaled Harras, Mohamed Kholief, and Saleh Mesbah. Energy efficient path planning techniques for UAV-based systems with space discretization. In *IEEE Wireless Communications and Networking Conference, WCNC*, pages 1–6. IEEE, 2016.

[3] Thomas S Alderete. Simulator Aero Model Implementation. Technical report, NASA Ames Research Center, Moffett Field, CA, 1995.

[4] James L Blue, Isabel Beichl, and Francis Sullivan. Faster monte carlo simulations. *Physical Review E*, 51(2):R867, 1995.

[5] Y. Bouktir, M. Haddad, and T. Chettibi. Trajectory planning for a quadrotor helicopter. *2008 Mediterranean Conference on Control and Automation - Conference Proceedings, MED'08*, pages 1258–1263, 2008.

[6] Pierre Jean Bristeau, Philippe Martin, Erwan Salaün, and Nicolas Petit. The role of propeller aerodynamics in the model of a quadrotor UAV. In *Proceedings of the European Control Conference*, pages 683–688, Budapest, 2009.

[7] S. Carlos, A. Sánchez, S. Martorell, and I. Marton. Onshore wind farms maintenance optimization using a stochastic model. *Mathematical and Computer Modelling*, 57(7-8):1884–1890, 2013.

[8] Mo Chen, Somil Bansal, Jaime F Fisac, and Claire J Tomlin. Robust Sequential Trajectory Planning under Disturbances and Adversarial Intruder. *IEEE Transactions on Control Systems Technology*, 27(4):1–17, 2018.

[9] Vishnu S Chipade, Abhishek, Mangal Kothari, and Rushikesh R Chaudhari. Systematic design methodology for development and flight testing of a variable pitch quadrotor biplane VTOL UAV for payload delivery. *Mechatronics*, 55:94–114, 2018.

[10] Dae Hyung Choi, Seong Hwan Kim, and Dan Keun Sung. Energy-efficient maneuvering and communication of a single UAV-based relay. *IEEE Transactions on Aerospace and Electronic Systems*, 50(3):2320–2327, 2014.

[11] Raghvendra V. Cowlagi, Joseph P. Sperry, and Jonathan C. Griffin. Unmanned Aerial Vehicle Trajectory Optimization for Executing Intelligent Tasks. *Journal of Guidance, Control, and Dynamics*, 41(6):1389–1396, 2018.

[12] Patricio J. Cruz and Rafael Fierro. Cable-suspended load lifting by a quadrotor UAV: hybrid model, trajectory generation, and control. *Autonomous Robots*, 41(8):1629–1643, 2017.

[13] Shicong Dai, Taeyoung Lee, and Dennis S Bernstein. Adaptive Control of a Quadrotor UAV Transporting a Cable-Suspended Load with Unknown Mass. In *53rd IEEE Annual Conference on Decision and Control (CDC)*, pages 6149 – 6154, Los Angeles, 2014.

[14] Aleksandra Faust, Ivana Palunko, Patricio Cruz, Rafael Fierro, and Lydia Tapia. Learning Swing-free Trajectories for UAVs with a Suspended Load. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4902–4909, Karlsruhe, 2013.

[15] Aleksandra Faust, Peter Ruymgaart, Molly Salman, Rafael Fierro, and Lydia Tapia. Continuous action reinforcement learning for control-affine systems with unknown dynamics. *IEEE/CAA Journal of Automatica Sinica*, 1(3):323–336, 2014.

[16] Federal Aviation Administration. Aerodynamics of Flight. In *Pilot's handbook of Aeronautical Knowledge*, pages 2–2 to 2–28. Skyhorse Publishing Inc., 2000.

[17] Xian Zhong Gao, Zhong Xi Hou, Zheng Guo, Rong Fei Fan, and Xiao Qian Chen. The equivalence of gravitational potential and rechargeable battery for high-altitude long-endurance solar-powered aircraft on energy storage. *Energy Conversion and Management*, 76:986–995, 2013.

[18] Nikola Gavrilovic, Emmanuel Benard, Philippe Pastor, and Jean-Marc Moschetta. Performance Improvement of Small UAVs Through Energy-Harvesting Within Atmospheric Gusts. In *AIAA Atmospheric Flight Mechanics Conference*, pages 1–18, 2017.

[19] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 57:65–100, 2010.

[20] Farhad A. Goodarzi. Autonomous aerial payload delivery with quadrotor using varying length cable. In *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pages 394–399. IEEE, 2016.

[21] Farhad A. Goodarzi, Daewon Lee, and Taeyoung Lee. Geometric stabilization of a quadrotor UAV with a payload connected by flexible cable. In *Proceedings of the American Control Conference*, pages 4925–4930, Portland, 2014.

[22] Farhad A. Goodarzi, Daewon Lee, and Taeyoung Lee. Geometric control of a quadrotor UAV transporting a payload connected via flexible cable. *International Journal of Control, Automation and Systems*, 13(6):1486–1498, 2015.

[23] Colin M. Greatwood, Rick Thomas, Jim Freer, Thomas S. Richardson, and Euan Nisbet. Automatic Path Generation for Multirotor Descents Through Varying Air Masses above Ascension Island. In *AIAA Atmospheric Flight Mechanics Conference*, pages 1–12, 2016.

[24] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. Garcia. IDA-PBC methodology for a quadrotor UAV transporting a cable-suspended payload. In *International Conference on Unmanned Aircraft Systems*, pages 470–476, Denver, 2015.

[25] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. García. Passivity based control for a quadrotor UAV transporting a cable-suspended payload with minimum swing. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6718–6723, Osaka, 2015.

[26] M. E. Guerrero-Sánchez, H. Abaunza, P. Castillo, R. Lozano, and C. D. García-Beltrán. Quadrotor Energy-Based Control Laws: a Unit-Quaternion Approach. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 88(2):347–377, 2017.

[27] M Eusebia Guerrero-Sánchez, D Alberto Mercado-ravell, Rogelio Lozano, and C Daniel García-beltrán. Swing-attenuation for a quadrotor transporting a cable-suspended payload. *ISA Transactions*, 68:433–449, 2017.

[28] Samantha Hoang, Yifeng Liu, Alberto Aliseda, and I Y Shen. Stability Analysis of High-Performance Drones With Suspended Payloads. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 8, pages 1–10, 2019.

[29] Samantha Hoang, Laurel Marsh, Alberto Aliseda, and I. Y. Shen. Analysis of High Fidelity Modeling of Drone Dynamics and Aerodynamics for Reduced Energy Consumption. In *Volume 7: 32nd Conference on Mechanical Vibration and Noise (VIB)*. American Society of Mechanical Engineers, 2020.

[30] Samantha Hoang, Laurel Marsh, Alberto Aliseda, and I. Y. Shen. Effects of High Fidelity Modeling of Multirotor Drones. *ASME Journal of Autonomous Vehicles and Systems*, 1(1):011007, 2021.

[31] Samantha Hoang and I. Y. Shen. Cost of Controls for Multi-Rotor Drones. volume Volume 8B: 45th Mechanisms and Robotics Conference (MR) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2021. V08BT08A002.

[32] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 1–20, Hilton Head, 2007.

[33] Gabriel M. Hoffmann, Haomiao Huang, Steven L. Waslander, and Claire J. Tomlin. Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control Engineering Practice*, 19(9):1023–1036, 2011.

[34] Nanyaporn Intaratep, William N. Alexander, William J. Devenport, Sheryl M. Grace, and Amanda Dropkin. Experimental Study of Quadcopter Acoustics and Performance at Static Thrust Conditions. In *22nd AIAA/CEAS Aeroacoustics Conference*, pages 1–14, Reston, Virginia, 2016. American Institute of Aeronautics and Astronautics.

[35] Karan P. Jain, Trey Fortmuller, Jaeseung Byun, Simo A. Makiharju, and Mark W. Mueller. Modeling of aerodynamic disturbances for proximity flight of multirotors. In *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, pages 1261–1269. IEEE, 2019.

[36] Kristian Klausen, Thor I. Fossen, and Tor Arne Johansen. Nonlinear control of a multirotor UAV with suspended load. In *International Conference on Unmanned Aircraft Systems*, pages 176–184, Denver, 2015. IEEE.

[37] Kristian Klausen, Thor I. Fossen, and Tor Arne Johansen. Nonlinear Control with Swing Damping of a Multirotor UAV with Suspended Load. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2017.

[38] Endrowednes Kuantama, Dan Craciun, and Radu Tarca. Quadcopter Body Frame Model and Analysis. Technical Report 1, University of Oradea, 2016.

[39] Yoshiaki Kuwata, Tom Schouwenaars, Arthur Richards, and Jonathan How. Robust Constrained Receding Horizon Control for Trajectory Planning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–12, Reston, Virigina, Aug 2005. American Institute of Aeronautics and Astronautics.

[40] Taeyoung Lee. Geometric Control of Multiple Quadrotor UAVs Transporting a Cable-Suspended Rigid Body. *IEEE Transactions on Control Systems Technology*, 26(1):255–264, 2018.

[41] Taeyoung Lee, Koushil Sreenath, and Vijay Kumar. Geometric control of cooperating multiple quadrotor uavs with a suspended payload. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5510–5515, Florence, 2013.

[42] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robotics & Automation Magazine*, 19(3):20–32, 2012.

[43] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2520–2525, Shanghai, 2011.

[44] E.E.L. Mitchell and A.E. Rogers. Quaternion Parameters in the Simulation of a Spinning Rigid Body. *SIMULATION*, 4(6):390–396, 1965.

[45] Fabio Morbidi, Roel Cano, and David Lara. Minimum-energy path generation for a quadrotor UAV. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1492–1498, Stockholm, Sweden, 2016. IEEE.

[46] Mark W. Mueller and Raffaello D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *IEEE International Conference on Robotics and Automation*, pages 45–52, 2014.

[47] Mark W. Mueller and Raffaello D'Andrea. Relaxed hover solutions for multicopters: Application to algorithmic redundancy and novel vehicles. *International Journal of Robotics Research*, 35(8):873–889, 2016.

[48] National Renewable Energy Lab (NREL). Horizontal Axis Rotor Performance Optimization (HARP_Opt), 2014.

[49] Miguel A. Olivares-Mendez, Pascual Campoy, Ignacio Mellado-Bataller, and Luis Mejias. See-and-avoid quadcopter using fuzzy control optimized by cross-entropy. In *IEEE International Conference on Fuzzy Systems*, pages 10–15, 2012.

[50] Oscar Oscarson. *Design , Modeling and Control of an Octocopter*. PhD thesis, KTH Royal Institute of Technology, 2015.

[51] Ivana Palunko, Rafael Fierro, and Patricio Cruz. Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In *IEEE International Conference on Robotics and Automation*, pages 2691–2697, 2012.

[52] Pedro O. Pereira, Manuel Herzog, and Dimos V. Dimarogonas. Slung load transportation with a single aerial vehicle and disturbance removal. In *Mediterranean Conference on Control and Automation*, pages 671–676. IEEE, 2016.

[53] Priyank Pradeep, Sang Gyun Park, and Peng Wei. Trajectory optimization of multirotor agricultural UAVs. In *IEEE Aerospace Conference Proceedings*, pages 1–7. IEEE, 2018.

[54] Stanislaw Radkowski and Przemyslaw Szulim. Analysis of vibration of rotors in unmanned aircraft. In *2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 748–753. IEEE, 2014.

[55] Gene Patrick Rible, Nicolette Ann Arriola, and Manuel Ramos. Modeling and implementation of quadcopter autonomous flight based on alternative methods to determine propeller parameters. *Advances in Science, Technology and Engineering Systems*, 5(5):727–741, 2020.

[56] Charles Richter, Adam Bry, and Nicholas Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. *Springer Tracts in Advanced Robotics*, 114:649–666, 2016.

[57] Vincent Roberge, Mohammed Tarbouchi, and Gilles Labonte. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, 9(1):132–141, 2013.

[58] Hazem Sallouha, Mohammad Mahdi Azari, and Sofie Pollin. Energy-Constrained UAV Trajectory Design for Ground Node Localization. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[59] Víctor San Juan, Matilde Santos, and José Manuel Andújar. Intelligent UAV Map Generation and Discrete Path Planning for Search and Rescue Operations. *Complexity*, 2018, 2018.

[60] John Seddon and Simon Newman. Rotor in Vertical Flight: Momentum Theory and Wake Analysis. In John Seddon and Simon Newman, editors, *Basic Helicopter Aerodynamics*, chapter 2, pages 23–61. John Wiley & Sons, Ltd., Chichester, UK, 3 edition, 2011.

[61] J. V. Seguro and T. W. Lambert. Modern estimation of the parameters of the Weibull wind speed distribution for wind energy analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 85(1):75–84, 2000.

[62] Andrew Short, Zengxi Pan, Nathan Larkin, and Stephen Van Duin. Recent progress on sampling based dynamic motion planning algorithms. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 1305–1311, 2016.

[63] Domen Šoberl, Ivan Bratko, and Jure Žabkar. Learning to Control a Quadcopter Qualitatively. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2020.

[64] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. In *52nd IEEE Conference on Decision and Control*, pages 2269–2274. IEEE, 2013.

[65] Brian L Stevens, Frank L Lewis, and Eric N Johnson. *Aircraft Control and Simulation.* John Wiley & Sons, Ltd., 3 edition, 2016.

[66] Sarah Tang and Vijay Kumar. Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2216–2222, 2015.

[67] Jan Erik Trachte, Luis Felipe Gonzalez Toro, and Aaron McFadyen. Multi-rotor with suspended load: System Dynamics and Control Toolbox. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, 2015.

[68] Abera Tullu, Youngseop Byun, Jae Nam Kim, and Beom Soo Kang. Parameter optimization to avoid propeller-induced structural resonance of quadrotor type Unmanned Aerial Vehicle. *Composite Structures*, 193:63–72, 2018.

[69] Ariel Walter, Michael McKay, Robert j. Niemiec, and Farhan Gandhi. Trim Analysis of a Classical Octocopter After Single-Rotor Failure. In *AIAA/IEEE Electric Aircraft Technologies Symposium*, pages 1–20, 2018.

[70] Steven L. Waslander and Carlos Wang. Wind disturbance estimation and rejection for quadrotor position control. *AIAA Infotech at Aerospace Conference and Exhibit and AIAA Unmanned...Unlimited Conference*, 2009.

[71] Kui Yi, Feng Gu, Liying Yang, Yuqing He, and Jianda Han. Sliding mode control for a quadrotor slung load system. In *Chinese Control Conference*, pages 3697–3703, 2017.

[72] Hyung-Jin Yoon, Venanzio Cichella, and Naira Hovakimyan. Robust Adaptive Control Allocation for an Octocopter under Actuator Faults. In *AIAA Guidance, Navigation and Control Conference*, pages 1–16, 2016.

[73] Daniel Zameroski, Gregory Starr, John Wood, and Ron Lumia. Rapid Swing-Free Transport of Nonlinear Payloads Using Dynamic Programming. *Journal of Dynamic Systems, Measurement, and Control*, 130(4):041001, 2008.

[74] Yu Zhang, Jing Chen, and Lincheng Shen. Real-time trajectory planning for UCAV air-to-surface attack using inverse dynamics optimization method and receding horizon control. *Chinese Journal of Aeronautics*, 26(4):1038–1056, 2013.

Appendix A

NOMENCLATURE FOR DYNAMIC MODELS

The variables used in 2.2 are described in Tabs. A.1 and A.2 below.

Table A.1: List of variables used in dynamic model and their definitions

Variable	Description
$\boxdot^e$	Vector written in the stationary (earth) frame
$\boxdot^b$	Vector written in the rotating (body) frame
$\mathbf{r}_0^b$	Position of drone's geometric center with respect to the system's center of mass (m)
$\mathbf{r}^e$	$\begin{bmatrix} x & y & z \end{bmatrix}$, position of the drone's center of mass (m)
$\mathbf{v}^b$	$\begin{bmatrix} u & v & w \end{bmatrix}$, velocity of the drone's center of mass (m/s)
$\bar{q}$	$\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$, quaternions defining the body frame orientation relative to the stationary frame
Ω	$\begin{bmatrix} p & q & r \end{bmatrix}$, angular velocity of the body frame relative to the stationary frame (radians per second)
M	Total drone mass, including the payload (kg)
I_G	Mass moment of inertia of the drone about the center of mass (kg$\cdot$ m^2)
$\mathbf{F}_B^b,\ \mathbf{F}_E^e$	Body and external forces, respectively, acting on the drone (N)
$\mathbf{T}_B^b,\ \mathbf{T}_E^e$	Body and external moments, respectively, acting on the drone (N$\cdot$m)
$\xi_B^b,\ \xi_E^b$	Location where the body and external forces, respectively, are applied relative to the geometric center (m)

Table A.2: List of variables used in aerodynamic models and their definitions

Variable	Description
ρ	Air density (kg/m^3)
ω	Rate of rotation of the rotors (RPM)
V	Axial velocity through the rotor (m/s)
c	Chord length for propeller blade (m)
R	Propeller blade length (m)
α_t	Pitch angle of the propeller blade (radians)
L	Length of the arm the propellers are on (m)
$C_{L\alpha}$	Lift coefficient
C_{D0}	Parasitic drag coefficient
C_{Di}	Lift-induced drag coefficient
$\bar{u}$, $\bar{v}$, $\bar{w}$	Velocity of the drone relative to wind velocity (m/s)
ε_1, ε_2	Propeller parameters for calculating the lift force
$\mathbf{x}_b$, $\mathbf{y}_b$, $\mathbf{z}_b$	Body frame axes