8-1-2013

Dynamics and control of a single-line maneuverable kite

Christopher Donnelly

Dynamics and control of a single-line maneuverable kite

by

Christopher Joseph Donnelly

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science
in Mechanical Engineering

Supervised by

Assistant Professor Dr. Mario W. Gomes
Department of Mechanical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
August 2013

Approved by:

Dr. Mario W. Gomes, Assistant Professor
Thesis Advisor, Department of Mechanical Engineering

Dr. Agamemnon Crassidis, Associate Professor
Committee Member, Department of Mechanical Engineering

Dr. Jason Kolodziej, Assistant Professor
Committee Member, Department of Mechanical Engineering

Dr. Wayne Walter, Professor
Department Representative, Department of Mechanical Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Dynamics and control of a single-line maneuverable kite

I, Christopher Joseph Donnelly, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Christopher Joseph Donnelly

Date

Acknowledgments

Before I begin, there are several people whose help and guidance I appreciate more than I can possibly express. You have always been there to help me through my problems and help me grow. I could not have done this without you.

First of all; Dr. Gomes, has been a joy to work with you. Thank you for this opportunity and somehow always making time and having the patience to talk things over with me. You are a great professor and your concern for your students' success and comprehension is truly admirable. I hadn't really even considered teaching until working with you. I still plan on holding you to your offer of introducing/recommending me to a few people for PhD programs (definitely a few years down the road though). To my committee members, Dr. Kolodziej and Dr. Crassidis; thank you for putting up with me and Dr. Gomes and our crazy ideas, multiple reschedulings, and often late/last minute distribution of papers. Also, thank you Professor Hanzlik, for always making me think about everything I hadn't considered.

Rob Kraynik, Jan Maneti, and Dave Hathaway have been invaluable in all things related to the experimental construction and testing. I have learned a great deal from you, and you were always there to answer my questions, however stupid they were. Thank you for all of your help.

I'm sure KGCOE has the best staff of any college at RIT. Jill, Diane, Diedra, Bill; a huge thank you is due for all of your help in scheduling, room reservation, academic advising, computer troubles, and just always brightening my day when I see you in the hall. Thank you to the RIT Honors Program, I've gained great experiences through Honors and really appreciated the financial assistance for conference attendance.

To Coach Jim, the entire RIT Crew, the Pep Band, and all of my friends at RIT; thank you for all of your time and friendship. I honestly don't know what I would have done without you; thank you for making me spend time outside of the lab on a regular basis.

Kelsey, Shriya, Konrad, Saloni, Ashwin, and the various co-ops, independent studies, and other part time lab members; thank you for making this experience even better. Someone was almost always there for me to complain and vent to, lend a hand in construction (thank you Ashwin), share each others' problems, and offer new perspectives. I'm leaving the lab in more than capable hands and I look forward to hearing how your research progresses. You can always ask me questions. (Also, don't try to do Matlab for more than 15 hours straight, it just doesn't often turn out well.)

And lastly, Danielle, thank you for distracting me just the right amount and making sure I do the work I need to do when I need to do it. You have been an incredible voice of reason and perspective for me, and I can never thank you enough.

Again, thank you all for everything you have done for me. Your support really helped me through. I wouldn't be here, at the completion of my thesis, without you.

Abstract

Dynamics and control of a single-line maneuverable kite

Christopher Joseph Donnelly

Supervising Professor: Dr. Mario W. Gomes

Through simulation, an automated control system for a single-line maneuverable kite is developed for application in kite wind energy production. The kite used in this study is a small, tension-controlled, single-line kite, commonly known as a fighter kite. These kites have a simple design, but flying them requires complex control of line tension and visual input. At low tether tension, the kite is unstable; spinning about the tether. Increasing tension in the tether causes the kite to deform and fly in the direction it was facing. Experienced fliers can produce intricate maneuvers and often participate in competitions with other fliers. A simplified physical and behavioral numeric simulation of the kite's dynamics was created and shown to closely approximate the actual kite's flight characteristics. This model was used to develop successful control algorithms for autonomous flight. Information of the kite's state and orientation used by the controller was gradually reduced to that which is physically measurable from the ground. An experimental test rig was designed and constructed for future testing in real wind conditions.

Contents

List of Tables

List of Figures

Nomenclature

$\hat{\imath}$	Inertial coordinate frame, X-axis
$\hat{\jmath}$	Inertial coordinate frame, Y-axis
$\hat{k}$	Inertial coordinate frame, Z-axis
$\hat{e}_\theta$	θ direction, local coordinate frame
$\hat{e}_\phi$	ϕ direction, local coordinate frame
$\hat{e}_r$	Tether direction, local coordinate frame
[T]	Transformation matrix between inertial and local frames
$\hat{k}_s$	Kite spar direction, orientation coordinate frame
$\hat{k}_f$	Kite facing direction, orientation coordinate frame
$\hat{k}_n$	Kite normal direction, orientation coordinate frame
[K]	Transformation matrix between local and orientation frames
$\vec{\omega}$	Angular velocity vector
$\dot{\vec{\omega}}$	Angular acceleration vector
$\vec{V}_k$	Kite velocity
AR	Aspect ratio
A	Kite area
m	Mass of kite
$P_{crit\ l}$	Spinning mode transition tension
$P_{crit\ h}$	Tracking mode transition tension
r	Tether length
θ	Kite position zenith angle
ϕ	Kite position azimuth angle
ψ	Rotation angle about tether
β	Bridle angle
γ	Roll angle
$\vec{W}$	Wind velocity
$\vec{W}_{app}$	Apparent wind velocity
α	Angle of attack
$\vec{L}$	Lift force
$\hat{l}$	Lift direction
$\vec{D}$	Drag force
$\vec{P}$	Tether tension
C_D	3-D Drag coefficient
C_L	3-D Lift coefficient
C_d	2-D Drag coefficient
C_l	2-D Lift coefficient
ρ	Air density
g	Gravity

Chapter 1

Background Information

1.1 Introduction

The main attraction of kite wind power is that it can be used to harness power from higher altitudes (above 200 m) than conventional wind turbines on towers. Higher altitudes tend to have faster, more consistent winds, and thus more available energy. High-altitude winds around the world contain approximately 100 times the energy consumption of the entire human population [2]. Wind towers can not reach these winds because the size and strength of the required tower would not be economically or in some cases even physically possible. As seen in Fig. 1.1, the aerodynamic forces acting on the turbine and tower produces a large bending moment, M_y, which is approximately equal to the force from the oncoming wind times the tower height. Conversely, the ground station of a kite-based system only has to withstand a changing tension vector.

Harnessing this energy is the challenge. Kite power systems use the lift and/or drag forces from wind to mechanically or electrically transmit energy from the kite to the ground. Mechanically coupled systems are controlled to alternate between a high tension power phase and low tension recovery phase. On the power phase, the tension is used to pull line off a spool and turn a generator. During the recovery phase, a motor winds tether back onto the spool. When properly controlled, this cycle produces a

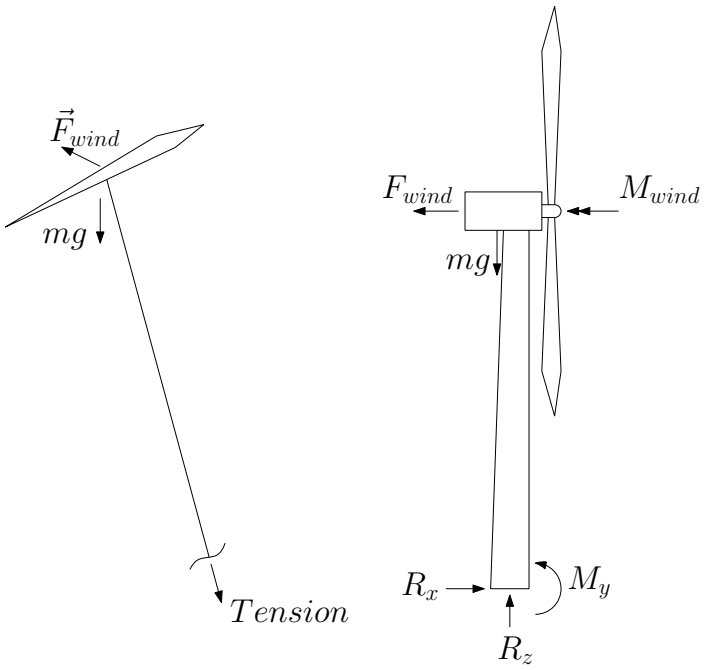


Fig. 1.1: Simple loading comparison of kite systems and tower systems. The bending moment grows with tower height and wind speed.

net positive power. Alternatively, electrical transmission systems use an array of small turbines on the kite to generate electricity, which is then transmitted along a conductive tether to the ground. Figure 1.2 (a) and (b) show general schematics of mechanical and electrical transmission systems, respectively. There are advantages to both methods in terms of control, efficiency, and continual production. The work described in this thesis is the beginning of a novel control method for a mechanical transmission system.

The tension control steering method of the kite studied in this work has advantages and disadvantages in comparison to other kite controls. Many kites used in power systems are re-purposed kite-boarding kites, which are controlled by multiple tethers. Kite power systems are being designed to fly with hundreds of meters of release tether. The tether can contribute a significant amount of drag as it flies through the air, slowing the kite and decreasing power extraction. Single-line kites produce significantly less drag than multiple line kites, increasing power production

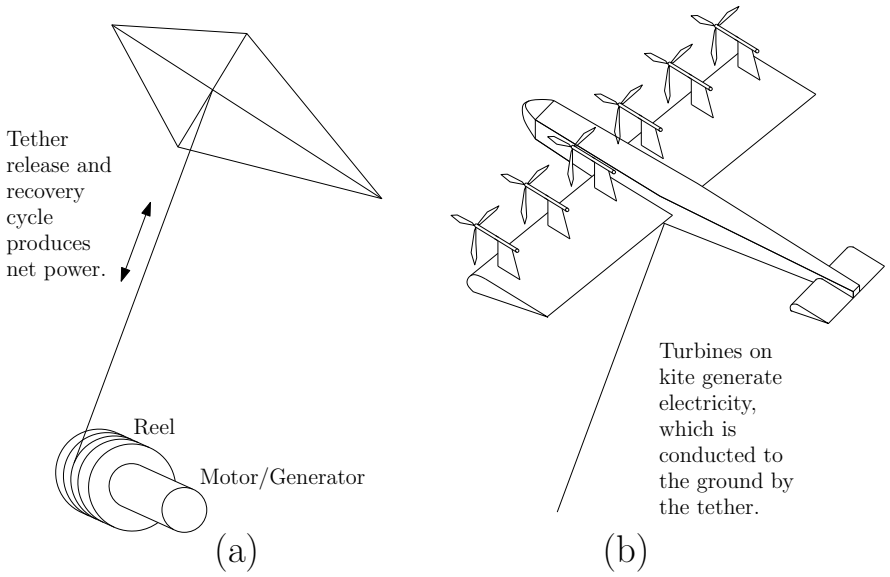


Fig. 1.2: Basic schematics of (a) mechanically coupled kite "pumping" system and (b) electrically coupled systems. For (a) power is generated on the ground and tether length oscillates with each cycle. For (b) power is generated on the kite and conducted along the tether.

potential. There are other systems that use single-line kites for power production, but their control systems rely on control surfaces and/or adjusting bridles to steer the kite. A combination of GPS, rate gyros, and accelerometers and/or other on-kite sensors are generally used to provide attitude and position data, wind speed, and any other information the controller needs to determine the kite's current behavior and how to fly the kite. The single-line tension control method described in this thesis could offer a simpler solution. A fighter kite's only control method is changing tether tension. By controlling tension and using it flex the kite and control its aerodynamic properties, on-kite measurement and actuation systems could be greatly reduced, if not eliminated. This makes the kite nearly expendable, easier to fabricate, transport, and replace. These benefits come with a cost. By removing on-kite instrumentation, all information on the kite's orientation and flight path must be measured or approximated from knowledge of the tension and the direction of the tether. These approximations reduce overall knowledge of position and control of the kite. Long

tether lengths and tether sag introduce more error in ground based position estimation, and may reduce responsiveness to tension control adjustments. There are no existing high lift to drag ratio designs for single-line tension control kites, providing another hurdle for exploration of this actuation method. Further work will determine if the benefits make up for these losses.

1.2 Motivation

Despite their long history and popularity, there has been no academic study of fighter kite dynamics. The work described in this thesis has improved our understanding of fighter kite dynamics and the extent of control possible without on-kite actuation or measurement systems for single-line kites. The knowledge gained provides a base for future research in single-line, mode-switching kite design and power production. The anticipated reduction in cost, tether drag, and complexity could offer many advantages. It could lead to a cheap, portable power system for use in remote locations, developing countries, or disaster relief situations, as well as more permanent generation sites.

1.3 Literature Review

Development of renewable energies is needed to offset the societal and environmental costs associated with acquiring and burning fossil fuels. High-altitude kite wind power is one of the more promising alternative energy sources currently available. Interest in kite wind power systems was sparked by a 1980 publication in the Journal of Energy. Miles Loyd analyzed the potential power extraction for a kite flown across the wind to generate high apparent velocity. Loyd conducted theoretical and numerical analyses, showing that a tethered C-5A airframe could generate as much as 6.7MW, exceeding the conventional wind tower turbines of the time. Crosswind

flight is perpendicular to the oncoming wind, and can result in apparent speeds several times that of the oncoming wind [16]. This greatly increases the aerodynamic forces that can be harnessed to generate electricity or do useful work on a system. Loyd's vector analysis of crosswind flight is shown in Fig. 1.3.

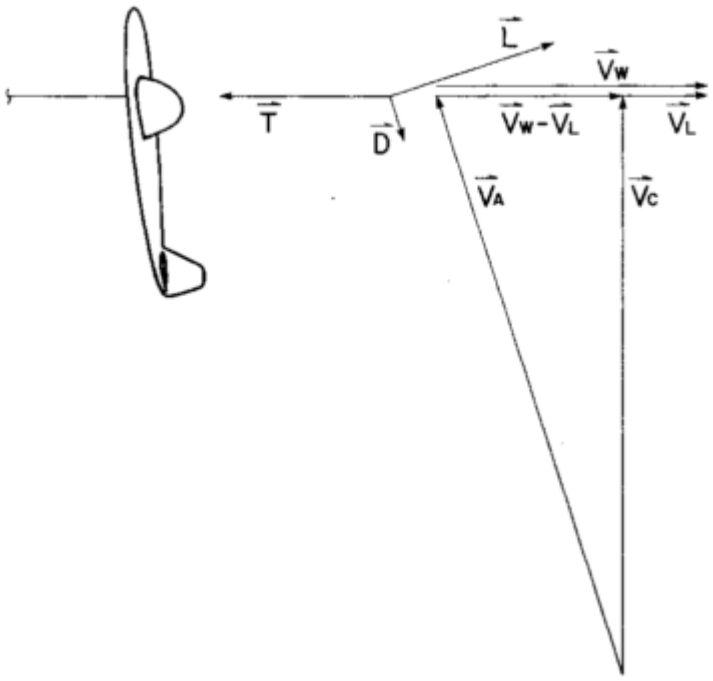


Fig. 1.3: Vector analysis of crosswind kite flight, showing the large increase in the apparent wind, $\vec{V}_A$, compared to the ambient wind speed, $\vec{V}_W$. Image taken from [16].

At low altitude, wind is slowed by friction with the earth and is also highly variable. Several hundred meters above the earth, shear effects are minimal and the wind's power density and consistency increase. Wind towers are limited to the slower, more variable winds at lower altitude because the physical forces on the tower are too great to build higher. Fagiano, Milanese, and Piga analyzed the capacity factor (portion of rated power that a system is able to produce at a specific site given a set of conditions) of kite systems and tower systems for several sites across Europe. They found that the more consistent wind speed at the higher operating range of kite systems nearly doubled capacity factor at every location [9]. This means that a 1MW kite system

would produce as much energy over a year as a 2MW wind tower. Higher capacity factors make it feasible to implement kite wind energy systems at a wide range of locations.

Unlike wind towers, which are only economical in limited areas, high-altitude wind is accessible over large geographical areas. Archer and Caldeira investigated the potential wind resource between altitudes of 500 m and 12,000 m above ground, showing that near 500 m, energy densities greater than 0.5 kW/m^2 are available for 95% of the year in wide areas of northern Africa, North America, and several offshore sites around the world [2]. The speed increase of wind with altitude is a significant factor in energy production. The power available in a fluid scales by its speed cubed, as given in Eqn. 1.1, where A_s is the swept area of the system.

$$P_{fluid} = \frac{1}{2}\rho_{fluid} A_s V_{fluid}^3 \tag{1.1}$$

At the proper latitudes, kites could encounter the high-speed, consistent wind of the jet streams, reaching power densities of 20 kW/m^2 [18]. Kites can harness wind energy and transmit it to the ground electrically or mechanically. Regardless of the conversion and transmission method, crosswind flight paths are generally used to generate high speeds and increase power output.

The simplest power extraction method is to use the tether tension to pull an object. Fagiano *et al.* conducted simulation and experimental testing to propel and steer a small ship [10]. Using manual flight control, they were able to generate a boat speed of 1.2m/s from a 2m/s wind, with only a 10m^2 kite. The experimental flight system is shown in Fig. 1.4. Precise trajectory control proved to be a large challenge in variable wind conditions. Williams *et al.* have also investigated kite towing systems for tracked (fixed trajectory) and free moving vehicles [22]. Path optimization was calculated for both power production and path tracking.

Electrically coupled systems incorporate turbines onto the tethered aircraft to

Fig. 1.4: Experimental flight tests of a manually operated kite system used for ship steering and propulsion. Kite is controlled by adjusting relative lengths of the two tethers. Image taken from [10].

generate power. The turbines are often powered and used to generate lift and fly the structure to the desired height and initiate its flight path for power production. The turbines are then used to generate electricity, which is conducted to the ground by a tether. Roberts *et al.* analyzed a four turbine flying electric generator that acts as a simple rotor-craft to gain altitude, and then can maintain elevation by adjusting the pitch and speed of the turbines [18]. The main advantage of electric systems is their ability for controlled flight for takeoff, landing, and in low or zero wind conditions. When producing electricity they generate a near constant positive power, as opposed to the cyclic production/consumption phases of the mechanical systems discussed below. A constant signal is advantageous for grid connection or storage. Their main drawbacks are conduction losses through the tether, and speed reduction of the kite due to the increased drag from the turbines. A slower kite does not generate as large

aerodynamic forces to do useful work.

Mechanical systems use the tether tension to generate electricity at the kite's ground station. The tether is wound around a drum or spool on the ground, which is coupled to a generator. In the high tension power phase, tether is released and electricity is generated as the reel unwinds. A lower tension phase is then used to recover the tether length, using less energy than was generated in the tension power phase. This cycle repeats in a periodic "pumping" motion as long as there is sufficient wind. The scope of this thesis applies to the mechanical pumping systems.

Modeling of kite systems poses some very difficult problems. Most kites are not rigid, resulting in coupled solutions of the aerodynamics and the mechanics of kite deflection. The tether model can vary in complexity from a mass-less, drag-less, straight rod, to multiple point masses connected by flexible links. Motion of the kite and tether can build to very complex dynamics problems quite rapidly. The wind velocity model can also range from simple uniform flow to an elevation profile with random fluctuations, direction change, and gusts. Given these complexities, some type of simplification is often needed to be able to produce a solution and understand the kite's flight characteristics.

Williams *et al.* created a very complete dynamic model of kite flight including tether sag and elasticity which they used to determine optimal trajectories for a vehicle towing system. Analysis of the system moving in a track and moving freely with a desired direction has shown interesting results. The nature of the optimal paths were highly dependent on wind speed, shifting from horizontal to vertical figure eights under faster winds [22]. They were also able to show that there is an optimal tether length, beyond which drag on the tether reduces gains from higher altitude, or longer high tension passes.

The tether system has a significant effect on the kite's performance. One of the

interesting conclusions from Argatov and Silvennoinen was the following:

$$\frac{{P_M}^{(n)}}{{P_M}^{(1)}} = \frac{(1+\chi_*)^2}{(1+\sqrt{n}\chi_*)^2} \tag{1.2}$$

where ${P_m}^{(n)}$ is the mechanical power of a kite system with n tethers and χ is a geometric relationship of the kite area and tether dimensions [4]. For significantly large values of χ, using L'Hopital's rule this becomes:

$$\frac{{P_M}^{(n)}}{{P_M}^{(1)}} = \frac{1}{n} \tag{1.3}$$

and shows that if fewer tethers are used, higher potential power is available. The power loss is due to drag on hundreds of meters of tether moving through the air and slowing down the kite. Argatov *et al.* showed how important the cable characteristics are in modeling. Beginning with a simple tether model, they increased complexity until the model accounted for the sag, drag, weight, and flexibility of the tether that most authors assumed out of their calculations. They also note the effects of tether sag on the angle of attack and show through numerical analysis that their model is accurate to within 6% [3]. Tether sag effectively decreases the angle of attack for systems with a fixed angle to their tether, in some cases by up to five degrees.

Modeling the kite is necessary to develop a control system that will keep the kite on an optimal flight path and produce the most energy. Control of kites requires some knowledge of the kite's current condition and the ability to change that condition. Sanchez and Christoforou employed two-dimensional models to develop station-keeping angular elevation controls for a kite. Canale *et al.* and Ilzhfer *et al.* analyzed kite systems to develop nonlinear model predictive control (NMPC) solutions. These solutions analyze the behavior of the system and predict what actuation will be needed at the next time step based on a set of inputs and a mathematical approximation of an optimization problem. Williams also used a nonlinear solution.

Houska and Deihl went a different direction, defining stable loops that would not require feedback.

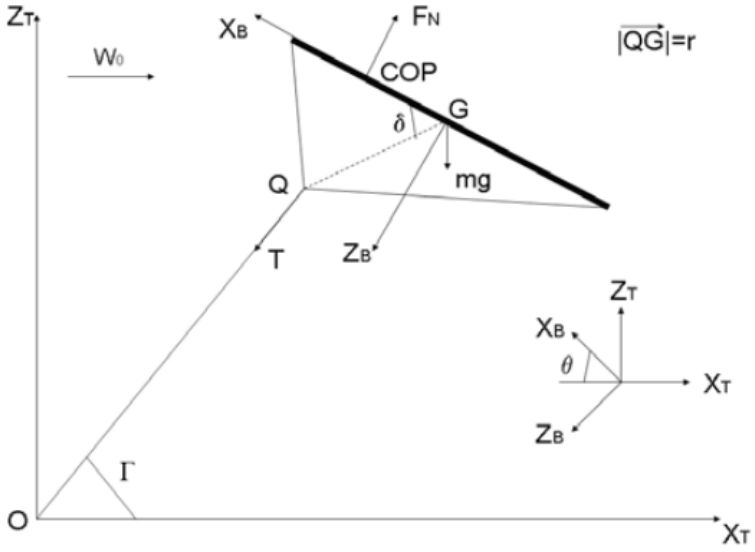


Fig. 1.5: 2-D model of kite employed by Sanchez for elevation control. Image taken from [19].

Sanchez modeled a flat plate with an shifting bridle. The connection point between the tether and kite was able to shift along a string connected to the leading and trailing edges of the kite. Under the assumption that the tether does zero work, he used Lagrangian equations to remove its mass and derive the below equilibrium equations.

$$\cos(\delta - \theta) + \beta(\sigma - \cos\delta)C_N(\theta) = 0 \tag{1.4}$$

$$\Gamma = \arctan\left(\frac{\beta C_N(\theta)\cos\theta - 1}{\beta C_N(\theta)\sin\theta}\right) \tag{1.5}$$

$$\beta = \frac{\rho A {W_o}^2}{2mg}, \sigma = \frac{x_{CP}}{r} \tag{1.6}$$

Where d and r are bridle geometry terms, x_{CP} is the location of the center of pressure acting on the kite to produce lift, and C_N is the normal force coefficient (normal to the kite) [19]. From these equations a control law was developed that will maintain the system at a set elevation, rejecting disturbances from change in wind speed. The

system worked well for small changes, but was susceptible to large gusts.

Christoforou conducted similar two-dimensional, station-keeping kite modeling and expanded his work to include experimental testing. A stable box kite was used, and bridle angle was changed with a motor and spool system. The motor rotates the spool and changes the relative lengths of the bridle on either side to change the angle of attack [8]. The experimental system showed smooth performance in angular elevation control.

There are several design solutions for mechanical pumping power kite systems. The three main designs are the laddermill of Delft University of Technology, the "yo-yo" or pumping systems (widely studied by various authors), and the carousel configuration of Torino Polytechnic University, Italy. Each is controlled to follow an optimized path based on either a predefined reference or a constantly optimized path.

The original laddermill concept was designed by Ockels and consists of a series of kites connected to a single tether wrapped around a drum on the ground, as seen in Fig. 1.6. The wind pulls out cable, generating electricity. At peak line extension the kites change in some way to reduce lift and the cable is drawn back in. This essentially produces a conveyor belt of kites, and generates a constant power [17]. Acting as a loop, the system poses several difficulties and can not take advantage of the high tensions associated with crosswind motion. The laddermill has since progressed to the mechanical pumping kite system.

Most pumping kite systems use a recreational power kite or surf-kite with two control tethers to fly optimal power paths. Each tether is wound around a drum and the kite is steered by adjusting the relative lengths of the tethers. This provides roll control. The kite is controlled to follow an optimal path; reeling out during the high tension of fast crosswind motion, and reeling in during the low tension phase. The traction system discussed in [10] for propelling ships follows the same design, but with a different optimization scheme.

Fig. 1.6: The original laddermill power generation system design. Image taken from [17].

The carousel configuration is actually a combination of pumping and towing power systems purposed and studied by Canale, Fagiano, and Milanese. Similar to the yo-yo configuration, a power kite with two control tethers is used to fly optimal loops by adjusting the relative lengths of the tethers, expect multiple kites are used simultaneously. The ground station is a large "carousel" set on a circular track. The optimal paths that the kites are controlled to fly along act in unison to rotate the entire structure. The combination of multiple pumping kites and pulling around the track will produce a continuous net positive power [7]. A schematic is shown in Fig. 1.7. They have conducted testing of manually controlled single kite pumping configurations and shown good correlation with their models, generating up to 1.5kW. Several issues stand in the way of the carousel configuration, including overall diameter (600m in their simulation), mechanical losses from multiple generator/gear systems, and potential entanglement of kites. Despite the large size, the authors expect much higher energy production per land area used than a traditional wind farm due to the wide spacing needed for wind turbines to avoid each others' wind shadows.

Previously, Canale *et al.* had studied optimal control of single kite systems. They used a pair of NMPC problems to control kite flight; one for the *traction phase* and

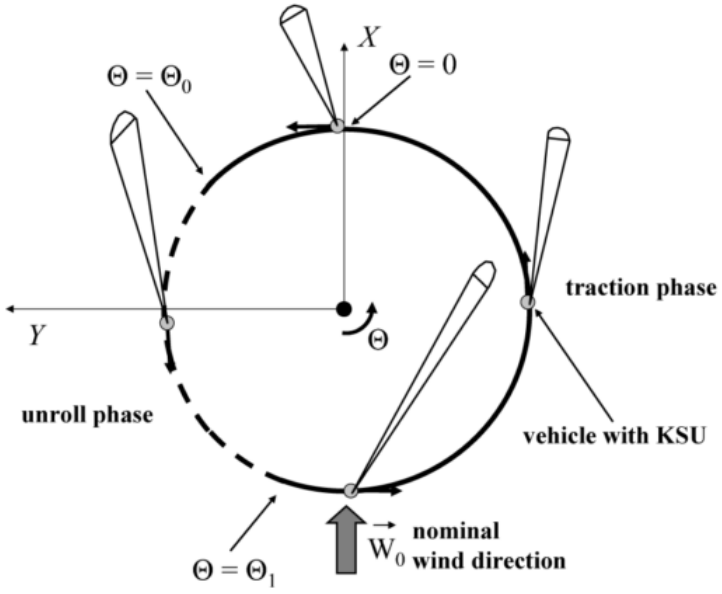


Fig. 1.7: Carousel configuration kite power system schematic. The individual pumping kites are optimized to both generate power individually and work cooperatively to rotate the entire system, which also generates electricity. This combination produces a net positive power at all times. Image taken from [7].

one for the *passive phase*. By splitting the problem in two, the authors were able to optimize each solution for maximum total power production [6]. Due to the need for a small sampling time, the NMPC was changed to a faster MPC, which is not as accurate, but allows for more path corrections than an NMPC. The problem directly optimized the flight path instead of using reference values, and was tested for various wind disturbances. Path results are shown in Fig. 1.8.

Ilzhfer *et al.* computed an optimization control solution for a patented kite design. A set of predetermined flight paths based on wind speed were created and an NMPC solution based on instrument input acted to maintain flight on the current path or shift according to changes in wind speed. The authors also modeled the response to changes in wind speed and direction and found that the system would reach an optimal loop within 40 seconds. One slight problem occurs with unsustained gusts. Small, random, and temporary changes in wind speed can trigger excessive shifting between different optimal paths. If the changes are only short term gusts the kite

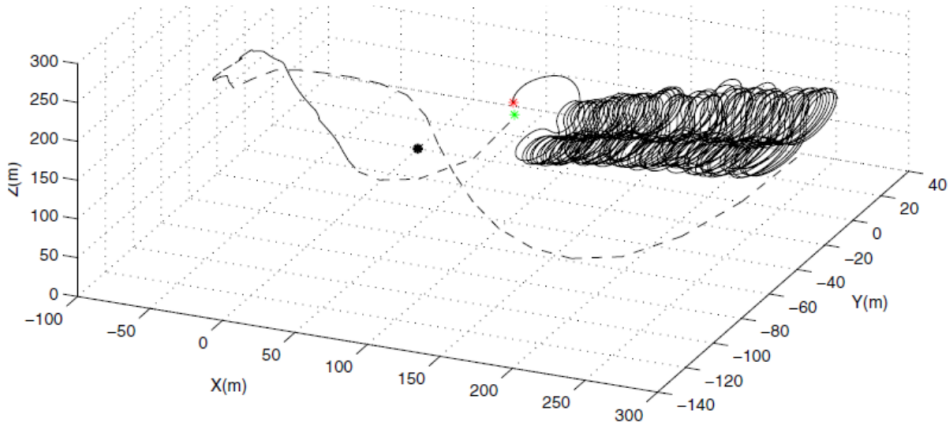


Fig. 1.8: Kite trajectory, showing 3-D motion of optimal power production path. Image taken from [6].

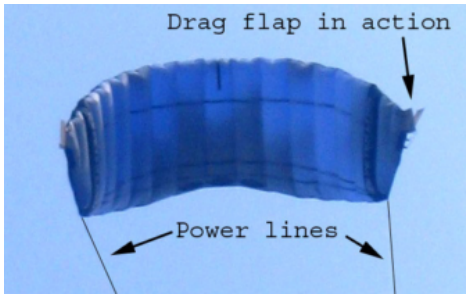
loses efficiency from switching between multiple paths instead of running on a slightly less optimal path [12].

Williams *et al.* solved the controls problem with a pair of nested loops. The outer loop follows the optimal flight path. The inner loop is used to reduce disturbances such as wind gusts and maintain flight in case the outer loop fails to solve or times out. The flight path is optimized for a predetermined set of reference speeds [21]. Having preset optimal paths greatly reduces in flight computational costs, but can't account for all speeds and so efficiency is reduced when the wind speed is between values of the reference set.

Houska and Deihl used their simulation to solve the optimal loops for power production, but found them to be unstable. By searching the model's behavior instead for stable loops, they were able to find flight paths that do not require any feedback. To account for wind variation they added a robustness factor in their calculations. Their final conclusions led to 23% power loss compared to the optimal loop, but gained stability and therefore simplified controls [11].

Lansdorp *et al.* continued with experimental work on kite steering mechanisms for pumping systems. They used a highly curved power kite such as those used for kite-surfing. Control mechanisms tested include a pair of small servomotors that

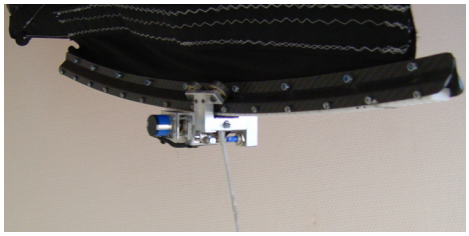
rotated flaps on the kite for differential drag adjustments, a pair of servos that pulled on the opposite corners of the kite to deform it and change the angle of attack, and a track and slider assembly on the edges of the kite with a motor that changed the tether attachment position (later replaced by a translating motor for greater travel distance). Figure 1.9 shows the various experimental flight actuators tested. From flight testing it was found that the track system (Fig. 1.9d) was most effective at overall steering and orientation control. By moving the motors together or separately it allows changes in angle of attack and left-right steering [15, 14].



(a) Drag flaps for lateral steering.

(b) Servomotor to deform opposite corner of kite for yaw control.

(c) Motor driven sliding attachment to shift angle of attack.



(d) Motorized track for shifting angle of attack.

Fig. 1.9: Kite actuation methods investigated by Lansdorp *et al.* Images taken from [14].

Most of these investigations use currently available kites for their analysis that were designed for something else, and thus have superfluous functionality that leaves room for optimization. One of the few projects working on designing the actual kite (besides those exploring electrically coupled systems) is Tensairity. Breuer and Luchsinger are developing an inflatable kite comprised of simple elements (cables,

links, and gas bags) that will all be in tension [5]. This approach would be very light weight and could be filled with helium to provide extra lift so that the kite doesn't have to come down during low wind conditions and deployment would be very simple. Another large gap in the literature is simple performance studies and information for various kite designs and configurations.

1.4 Fighter Kite Overview

Fighter kites have been used recreationally for centuries throughout the Middle East, India, and other parts of Asia. They are very popular and come in a wide variety of shapes and sizes designed to provide different performance characteristics. The object of traditional fighter kite competitions is to cut the line of your opponent. The kite tether is coated in a mixture of glue and powered glass. Highly skilled fliers maneuver their kite so this abrasive tether will "saw" across the opponent's kite string, cutting it. On festival days, hundreds of kites can be in the air at a time over a city, all involved in the competition. The kites are flown with hundreds or even thousands feet of tether in the air. The work done in this thesis focuses on North American style fighter kites, which are a more recent development. The main difference between traditional and North American fighter kites is the objective of the kite competition. North American fighter kites don't use cutting line and are used to play line touch games or complete obstacle/challenge maneuvers. Consequently, the kite design has been altered to better perform these type of tasks. North American style fighter kites tend to be smaller, more maneuverable, fly and turn faster, rarely have tails, and are flown with shorter tether lengths [13]. Their greater maneuverability and speed are more advantageous for power production.

Despite their differences, nearly all fighter kites are flown by switching between two distinct modes: spinning and tracking. In the spin mode the kite is essentially a flat disc. It is aerodynamically unstable and spins about the tether. The tracking

mode occurs when the kite is flexed into a U-shaped profile. It becomes more stable and flies in a straight path in the direction it was facing at the time. The transition between modes is based on the mechanical loads on the kite. High tether tension pulls the kite into the wind, and causes it to flex into the U-shape. Under lower tension the kite is stiff enough to remain flat and spins. We think that the switch between spinning and tracking modes does not occur at a single defined tension value. Rather, we believe there is a range of tension where the kite could be in either mode. Its behavior remains the same until it leaves this transitional zone. Spin direction can be clockwise or counterclockwise. It depends primarily on bridle tuning, mass distribution, and the kite's orientation and tether tension during the transition from tracking to spinning. Other behaviors are possible, but not considered for this study. Through precise small reductions in tension the kite can be made to hover or "cobra", so called because it sways from side to side like a cobra in a snake charming act.

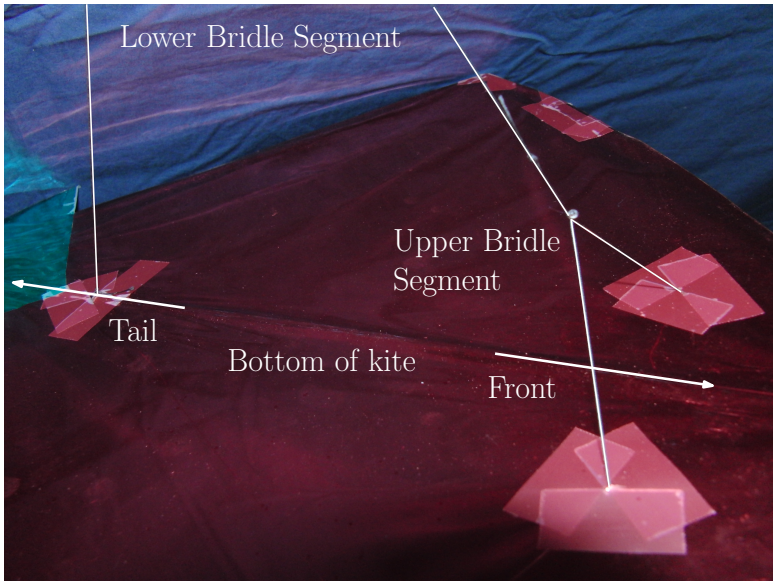


Fig. 1.10: Bridle geometry for a three point, adjustable bridle.

The kite's performance and stability depends on weight distribution, size, shape, and the bridle geometry. Two, three, four, and five point bridles can be used to connect the kite to the tether. The bridle lengths can be adjusted to tune the kite's

flight characteristics. The kite used in this study has a three-bridle. The three-point bridle consists of upper and lower segments. The upper segment is attached to the flexing cross spar at points approximately two inches from the spine. The lower segment is attached to the spine, approximately one third of the spine's total length from the tail. The other end of the lower segment is tied to the middle of the upper segment with an adjustable knot. Figure 1.10 shows the attachment points and labels the different segments. The tether is tied to an adjustable point along the lower segment of the bridle. The tether used in this work is waxed cotton string. The wax reduces drag in the air and helps to prevent twisting and tangling.

Top and bottom views of a typical North American fighter kite are shown in Figs. 1.11 and 1.12. The top view shows the cross-spar and spine of the kite. The cross-spar is a bamboo splint that bows across the entire width of the kite. The stiffness of the cross-spar determines how much force is needed to flex the kite, and effectively what range of wind speeds the kite can be flown in. The spine is also made of bamboo, but has a larger cross-section and is much stiffer than the cross-spar. Fighter kites can use different types of wood, carbon fiber, and/or plastic for the spine and cross-bar [13]. For traditional fighter kites, tissue paper is used for the sail since it is light and relatively strong. Many kites now use thin sheets of plastic, such as Mylar for the sail.

Fig. 1.11: Top of kite, showing flexing crossbar and spine structure.

Fig. 1.12: Bottom of kite, showing bridle attachment points.

Chapter 2

Simulation: Kite Dynamics

2.1 Kite Model Description

In order to investigate the kite's behavior and rapidly test control algorithms, a simplified kite model was developed. The model integrates the kite's equations of motion through time to simulate kite flight. The model itself is built from several simplifying assumptions.

The kite is modeled as a point mass on the end of a variable length tether moving in three-dimensional space. The tether is inelastic, straight, has no mass, and is unaffected by aerodynamic drag. Only the primary tracking and spinning behaviors of fighter kites are portrayed in the model. During the spin mode the kite is assumed to rotate with a positive angular velocity about an axis along the tether. All segments of the bridle remain in tension and we have neglected aerodynamic moments acting in the plane of the kite. Since these moments are neglected, it is assumed that the angle between the tether and the spine of the kite is fixed. The bridle angle, β is the complement of this angle. To maintain all bridle segments in tension, the roll angle, γ, is limited to plus or minus forty degrees. Roll angle is assumed to occur about the spine of the kite, along the facing direction, $\hat{k}_f$. The small mass and inertia of the kite makes its roll resistance insignificant, therefore we assume that the kite instantaneously rolls to be perpendicular to the oncoming wind (or as close as possible

given the limited range). To enforce physical limitations, the simulation produces an error and exits if the tension becomes negative (the tether is in compression, pushing the kite) or the zenith angle exceeds ninety degrees (the kite passes through the ground). The geometry of the kite is chosen as a square as opposed to a diamond. This simplifies the aerodynamic analysis by having a constant chord length.

The position of the kite is defined in spherical coordinates by tether length r, azimuth angle ϕ, and zenith angle θ. Three more angles are used to fully define the kite's orientation: spin angle ψ, bridle angle β, and roll angle γ. From an inertial coordinate frame, $[\hat{i}\hat{j}\hat{k}]$, a ZYZ Euler transformation is used to define the local coordinate frame, $[\hat{e}_\theta\ \hat{e}_\phi\ \hat{e}_r]$. The local coordinate frame is the moving frame from which kite orientation is defined. Bridle angle and roll angle are used to relate the local system to the kite orientation frame, $[\hat{k}_s\hat{k}_f\hat{k}_n]$. The kite orientation frame is fixed to the kite and its directions align with features of the kite. The facing direction, $\hat{k}_f$, runs along the spine of the kite from the tail to the front. When facing the bottom of the kite, the spar direction, $\hat{k}_s$, is rotated ninety degrees counter clockwise from the facing direction. It runs from the center of the kite to the edge of the wing. The kite's normal direction, $\hat{k}_n$, is perpendicular to the plane of the kite. It points out of the top side of the kite, whereas the tether attaches to the bottom of the kite. Figure 2.1 details the transformation from the inertial frame to the local frame. Figure 2.2 depicts the kite, as viewed downed the spar direction, when $\gamma = 0$ to better illustrate the bridle angle β's relation to the local and orientation coordinate systems. If there is no roll angle, the spar direction and $\hat{e}_\theta$ are aligned. Figure 2.3 shows the kite viewed along the facing direction with $\beta = 0$ to more clearly define the roll angle γ. If there is no bridle angle ($\beta = 0$), the facing direction and $\hat{e}_\phi$ are aligned.

Transformation matrices [T] and [K] are used to convert vectors between the inertial and local, and local and orientation coordinate systems, respectively. [T] is a full ZYZ transformation built from three discrete rotations; first by ϕ about the

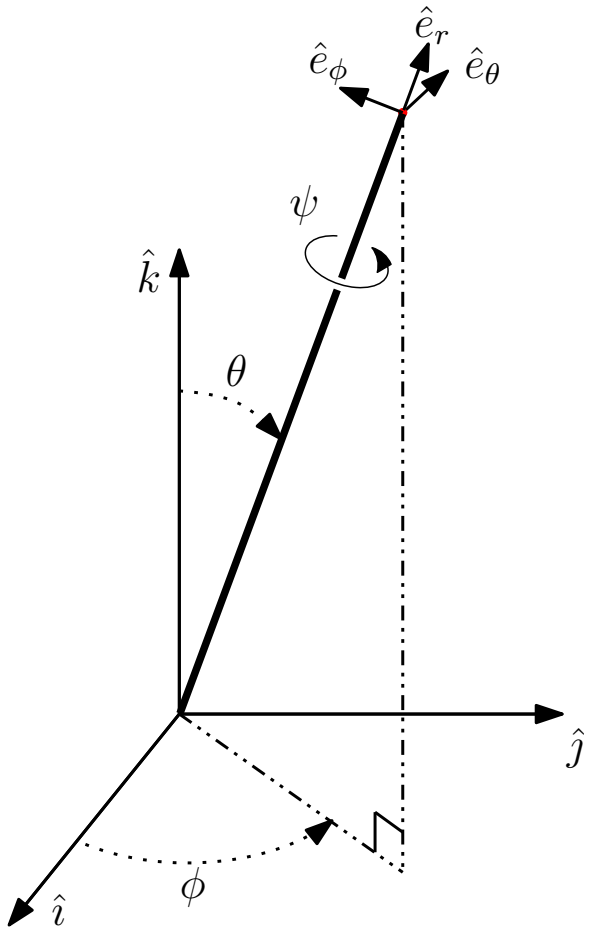


Fig. 2.1: Kite coordinate systems diagram, showing ϕ, θ, and ψ rotations to go between inertial and local systems.

Z-axis, then by θ about an intermediate $\hat{\jmath}'$ direction (the Y'-axis). The third rotation is the spin angle, ψ about the tether, the $\hat{e}_r$ direction. The transpose of [T] is used to convert from the position frame to the inertial frame. The bridle and roll angles are used to define [K]. The first rotation is β about $\hat{e}_\theta$, followed by γ about $\hat{k}_f$, the

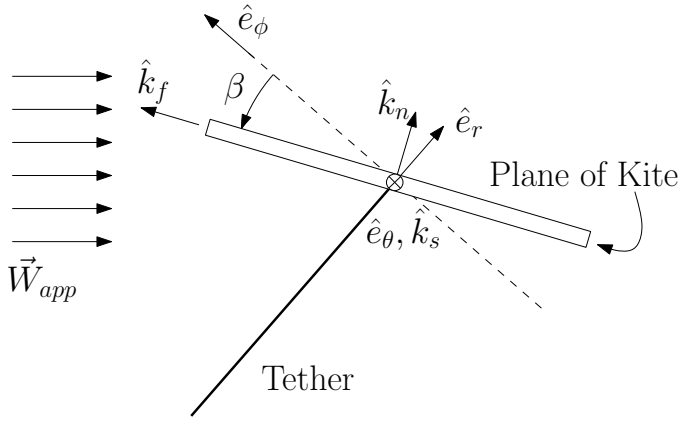


Fig. 2.2: Visualization of β angle relating facing direction to tether. Roll angle, γ, is zero.

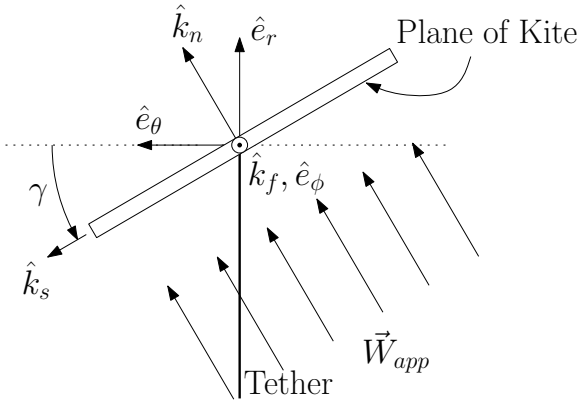


Fig. 2.3: Visualization of roll angle, γ, relating kite orientation to apparent wind velocity. Bridle angle, β, is zero.

spine of the kite.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}_{local} = T \begin{bmatrix} A \\ B \\ C \end{bmatrix}_{inertial}$$

where $T =$

$$\begin{bmatrix} \cos\psi\cos\theta\cos\phi - \sin\psi\sin\phi & \cos\psi\cos\theta\sin\phi + \sin\psi\cos\phi & -\cos\psi\sin\theta \\ -\sin\psi\cos\theta\cos\phi - \cos\psi\sin\phi & -\sin\psi\cos\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta \\ \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{bmatrix} \tag{2.1}$$

$$\begin{bmatrix} d \\ e \\ f \end{bmatrix}_{orientation} = K \begin{bmatrix} D \\ E \\ F \end{bmatrix}_{local} \quad \text{where } K = \begin{bmatrix} \cos\gamma & \sin\gamma\sin\beta & -\sin\gamma\cos\beta \\ 0 & \cos\beta & \sin\beta \\ \sin\gamma & -\cos\gamma\sin\beta & \cos\gamma\cos\beta \end{bmatrix} \tag{2.2}$$

Mode change is based on high and low transition tensions with a hysteresis gap between them. This gap prevents the kite from switching between spinning and tracking due to small oscillations about a single transition point. Figure 2.4 displays the simulation's method for switching between the tracking and spinning modes.

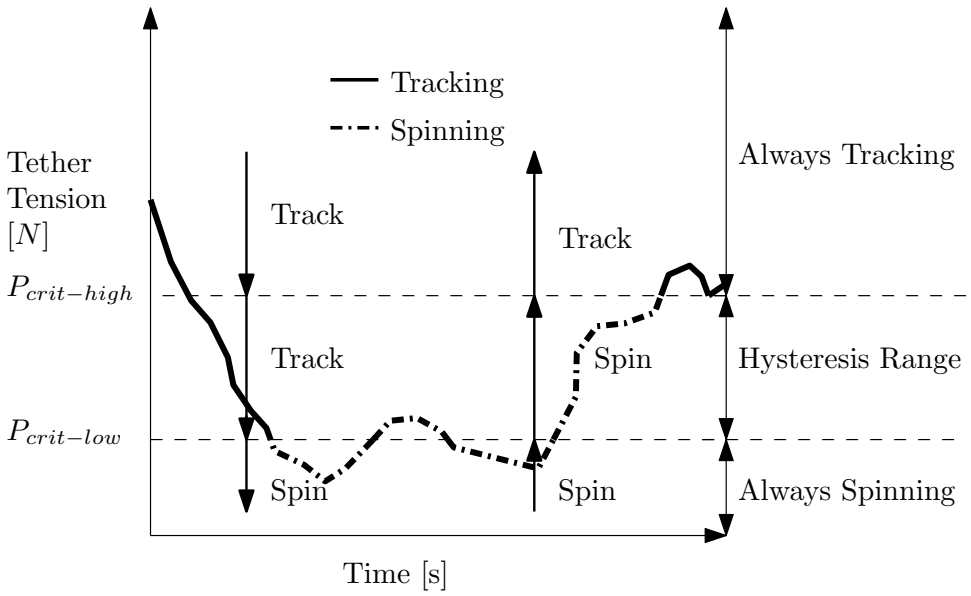


Fig. 2.4: Method for applying hysteresis between spinning and tracking modes. Tension must pass the lower bound to switch from tracking to spinning, or the upper bound to switch from spinning to tracking.

The kite's equations of motion are derived below with Eqns. 2.3-2.9. The $\dot{r}$ and $\ddot{r}$ terms arise because the tether changes length during flight. In the local coordinate frame, the kite's position is simply a distance r along the tether, as seen in Eqn. 2.3.

$$\vec{r} = r\hat{e}_r \tag{2.3}$$

Its derivatives follow, giving the velocity and acceleration terms in Eqns. 2.4 and 2.5.

$$\vec{v} = \dot{r}\hat{e}_r + \vec{\omega} \times r\hat{e}_r \tag{2.4}$$

$$\vec{a} = \ddot{r}\hat{e}_r + \vec{\omega} \times \dot{r}\hat{e}_r + 2\dot{\vec{\omega}} \times r\hat{e}_r + \vec{\omega} \times (\vec{\omega} \times r\hat{e}_r) \tag{2.5}$$

The derivatives of the finite rotations of the system define the angular velocity, $\vec{\omega}$, and angular acceleration, $\dot{\vec{\omega}}$ of the local coordinate frame, as seen in Eqns. 2.6-2.8. As the kite is a point mass, only rotations θ, ϕ, and ψ effect the equations of motion. The unit vector $\hat{\jmath}'$ is in the $\hat{\imath} - \hat{\jmath}$ plane. This intermediate axis is defined by rotating the inertial $\hat{\jmath}$ direction (the Y-axis) by ϕ degrees about the inertial $\hat{k}$ direction (the Z-axis), as seen in Fig. 2.5.

$$\omega = \dot{\phi}\hat{k} + \dot{\theta}\hat{\jmath}' + \dot{\psi}\hat{e}_r \tag{2.6}$$

Equation 2.7 gives the rotations rewritten in the local frame. Unless otherwise noted,

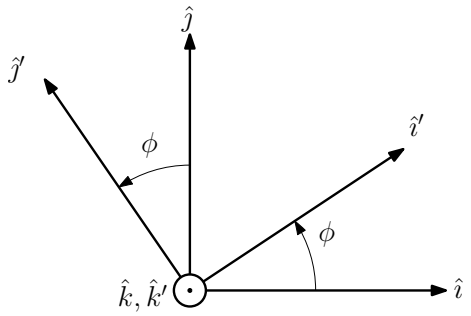


Fig. 2.5: Intermediate step in the ZYZ Euler coordinate transformation, identifying $\hat{\jmath}'$.

all equations are written in the local coordinate system, as it is the most convenient for analysis.

$$\vec{\omega} = \begin{bmatrix} \dot{\theta}\sin\psi - \dot{\phi}\sin\psi\sin\theta \\ \dot{\theta}\cos\psi + \dot{\phi}\sin\psi\sin\theta \\ \dot{\psi} + \dot{\phi}\cos\theta \end{bmatrix} \tag{2.7}$$

$$\dot{\vec{\omega}} = \begin{bmatrix} \ddot{\theta}\sin\psi - \ddot{\phi}\cos\psi\sin\theta + p\dot{h}i\dot{\theta}\cos\psi - \dot{\phi}\dot{\theta}\cos\psi\cos\theta + \dot{\phi}\dot{\psi}\sin\psi\sin\theta \\ \ddot{\theta}\sin\psi + \ddot{\phi}\sin\psi\sin\theta - p\dot{h}i\dot{\theta}\sin\psi + \dot{\phi}\dot{\theta}\sin\psi\cos\theta + \dot{\phi}\dot{\psi}\cos\psi\sin\theta \\ \ddot{\psi} + \ddot{\phi}\cos\theta - \dot{\theta}\dot{\phi}\sin\theta \end{bmatrix} \tag{2.8}$$

A free body diagram of the kite and forces acting on it; lift, drag, kite weight, and tether tension is shown in Fig. 2.6. Equation 2.9 relates the kite's acceleration to the forces acting on it using simple Newtonian dynamics, $\Sigma\vec{F} = m\vec{a}$.

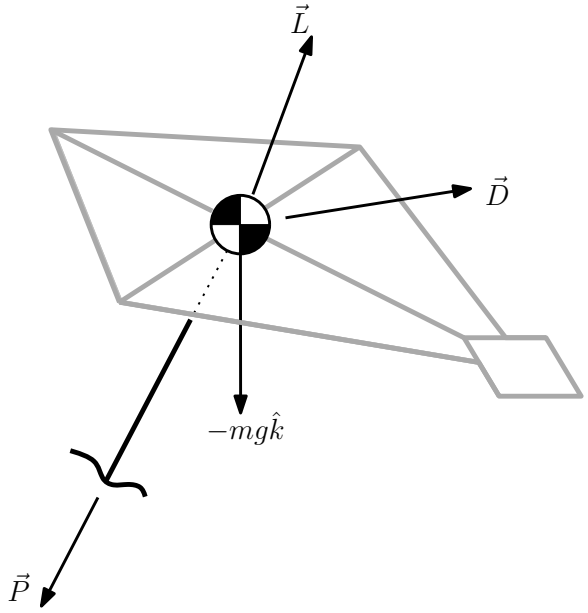


Fig. 2.6: Free body diagram of simplified kite model. Tether tension, P, is along the tether.

$$\vec{L} + \vec{D} - mg\hat{k} - P\hat{e}_r = m\vec{a} \tag{2.9}$$

The directions of the drag and lift vectors are determined by the apparent wind velocity, $\vec{W}_{app}$, and kite orientation. The apparent wind is the wind velocity as measured from the kite. This relative velocity is therefore the actual wind, $\vec{W}$, minus the velocity of the kite, $\vec{V}_k$, as shown in Eqn. 2.10. The drag vector is parallel to the apparent wind. The lift direction is more complex. Lift is perpendicular to $\vec{W}_{app}$ and the cross-spar of the kite. Based on the model assumption that the kite rolls

instantaneously to eliminate the roll moment, the spar of the kite ($\hat{k}_s$) is perpendicular to the apparent wind. So the lift vector is then perpendicular to $\hat{k}_s$ and $\vec{W}_{app}$. The equations for drag and lift, Eqns. 2.11 and 2.12, use steady-state lift and drag coefficients, C_L and C_D, to determine their magnitudes. The aerodynamic properties of the kite are assumed to be comparable to NACA airfoil 0015. NACA 0015 is a symmetric airfoil that is relatively flat. Sheldahl and Klimas experimentally measured 2-D lift and drag coefficients through 360° of attack angles for several airfoils. This provides coefficients for the necessary range that the simulation may encounter [20]. Conversion of coefficients from 2-D to 3-D is detailed in App. B.2.

$$\vec{W}_{app} = \vec{W} - \vec{V}_k \tag{2.10}$$

$$\vec{D} = \frac{1}{2}\rho A C_D |\vec{W}_{app}| \vec{W}_{app} \tag{2.11}$$

$$\vec{L} = \frac{1}{2}\rho A C_L |\vec{W}_{app}|^2 \hat{l} \tag{2.12}$$

$$\hat{l} = \left(\hat{k}_f \times \frac{\vec{W}_{app}}{|\vec{W}_{app}|} \right) \times \frac{\vec{W}_{app}}{|\vec{W}_{app}|} \tag{2.13}$$

The equations of motion result in four unknowns: $\ddot{\theta}$, $\ddot{\phi}$, $\ddot{r}$, and P (the tether tension). To simulate control of the kite, the length of the tether and its derivatives are defined by the control algorithm, and thus $\ddot{r}$ is considered known. This brings the system to three equations and three unknowns, which can be solved for each of the remaining unknowns. This solution can be numerically stepped forward in time to simulate kite flight. The state vector is given in Eqn. 2.14.

$$\begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \\ \psi \\ \dot{\psi} \\ r \\ \dot{r} \end{bmatrix} \tag{2.14}$$

Having generated the equations of motion, kite properties and other simulation parameters must be chosen. The magnitude of the spin rate was determined from experimental testing. Video footage of manual kite flight was recorded, and a hand-held anemometer was used to determine wind speed. Frame by frame analysis from several flights showed that spin rate had little dependence on wind speed and primarily ranged between ten and fifteen radians per second. We selected a constant value of 12.5 rad/s for the simulation as a median between the extremes. Tension transition values were determined by probing the system's behavior and experimental insight. It was difficult for simple experiments to accurately measure tether tension during the transitions between tracking and spinning. We did note when the kite transitioned from one mode to another, it stayed in that mode until largely perturbed from it. Initial simulation using a single transition value showed multiple rapid changes between spinning and tracking modes that did not match the kite's behavior. A transitional zone was used to better simulate the kite's tendency to stay in the current mode. Values ranging from 1N to 6N were tested in the simulation to adjust the kite behavior to more closely follow the actual kite. If the gap between was too small the kite would perform undesirable transitions between spinning and tracking. If the gap was too large significant tether speed changes would be needed that did not correlate

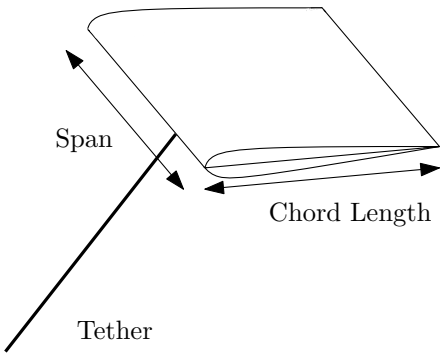


Fig. 2.7: Labeled square kite geometry used to simplify aerodynamic properties by having a constant chord length.

well to manual flight. A hysteresis range of 1.5N was eventually chosen, starting at 4N for the transition to spinning and 5.5N to transition to tracking. The span, chord, and kite mass used in simulation are direct measurements of the fighter kite shown in Figs. 1.10-1.12. The chord and span are shown in Fig. 2.7. Air density was set to 1.205 kg/m^3. This is the density of air at normal temperature and pressure; 1 atmosphere and 20°C. Table 2.1 lists the simulation parameters, values, and units.

Table 2.1: Table of Simulation Parameters

Parameter	Value	Units
Time Step	0.001	s
Kite Mass	0.016	kg
Gravity	9.81	m/s^2
Chord Length	0.41	m
Span	0.41	m
Air Density	1.205	kg/m^3
Upper Critical Tension	5.5	N
Lower Critical Tension	4	N
Kite Spin Rate	12.5	rad/s

A fixed-step size, fourth-order Runge-Kutta scheme was used to simulate the kite's flight. A numerical convergence study was conducted to determine an appropriate time step. Final values of zenith, azimuth, and roll angles, were recorded for three second simulations at times steps from 0.04 to 0.000005 seconds. The percent relative error, defined in Eqn. 2.15 and plotted in Fig. 2.8, compared the relative accuracy of

the simulation for each step size. The term x_i denotes the final value of the simulation for a step size and x_{i+1} is the final value for the simulation run for the next smaller step size. As expected with a fourth order integration routine, the slope on the log-log graph is approximately four. Integration and truncation error accumulated for time steps smaller than 0.0001, which accounts for the increase in error shown in Fig. 2.15. A time step of 0.001 seconds was chosen to minimize error and still allow for reasonable simulation run-times.

Initial error analysis did not show fourth order error reduction with time step. Instead, a slope of one was seen, which would be expected of a first-order integration routine, such as Euler's Method. This arose because the state vector did not originally contain the tether length or rotation angle and their derivatives: r, $\dot{r}$, ψ, and $\dot{\psi}$. Even though each could be exactly calculated, they introduced error into the Runge-Kutta integration. The fourth-order Runge-Kutta routine calculates the state vector for partial steps between each major time step to provide a better approximation. The terms not included in the state vector were not appropriately updated in the partial steps and so made the other state variables less accurate. This was solved by simply extending the state vector to include all of the terms that are now in Eqn. 2.14.

$$\text{relative error} = \left|\frac{x_i - x_{i+1}}{x_i}\right| \tag{2.15}$$

The full simulation code is available in App. A.1.

2.2 Uncontrolled Simulation Results

The simulation appears to capture the desired behavior of the kite's flight. Conservation of energy was the first test used to check the validity of our equations of motion. The aerodynamic forces were eliminated (as they do work on the system) and the

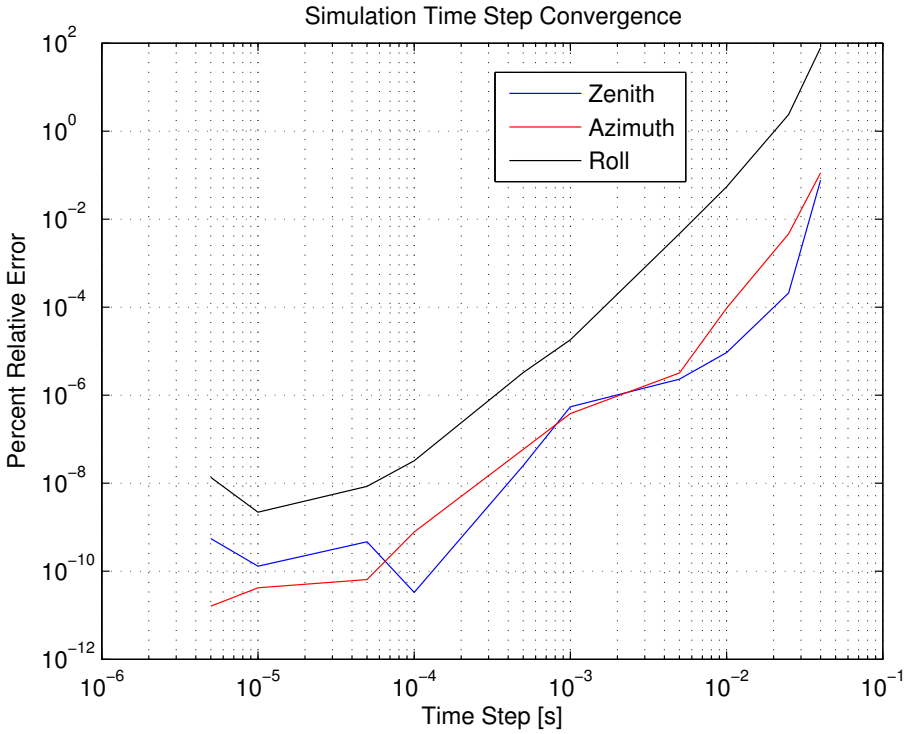


Fig. 2.8: Reduction of relative error as time step decreases. The slope is near four decades error reduction for each decadal reduction in time step, as expected for a fourth order integration routine.

simulation run from various initial conditions. For all cases energy was conserved; within the tolerances of our numerical integration scheme. System energy during a three second simulation is plotted in Fig. 2.9.

We were also able to compare the simulation with observed and logically expected kite behavior. For all figures of kite path, the tether originates at the origin, [0, 0, 0]. Figures 2.10 and 2.11 show the kite path and tension for increasing wind speed. The wind is directed along the X-axis for these figures. For low wind, minimal tension is generated, and therefore the kite falls, spinning as it does so. As the wind increases, it produces more lift on the kite, and the kite moves faster. The faster moving kite, while still in spin mode, cannot make as tight of loops as a slower kite. For high enough wind (the light blue path) the spinning kite will gather more speed and thus increase the aerodynamic forces enough to switch to tracking mode. For the highest

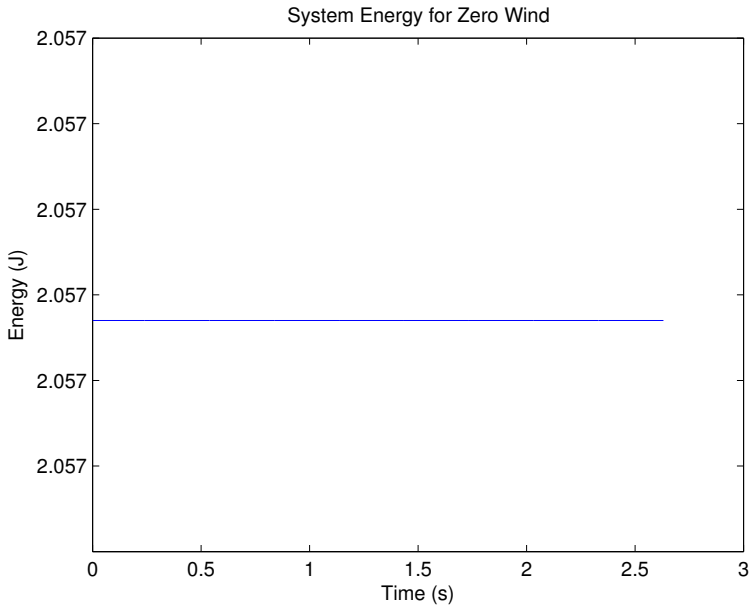


Fig. 2.9: Plot of energy conservation for kite simulation.

wind speeds (magenta and black paths) the kite doesn't spin, but tracks to a steady position. The periodic fluctuation in tether tension seen in Fig. 2.11 occurs as the kite spins and changes its orientation to the wind. A tension of zero indicates that the kite has crashed into the ground. It is interesting to note that the tension fluctuates at the same frequency as the kite's spin rate.

Kite orientation to wind has a large effect on behavior. Keeping all other conditions the same, initial zenith angle was changed to start the kite at high and low elevations, zenith angles of 40° and 85°. As seen in Fig. 2.12, at a wind speed of five m/s the lower elevation case (left side) had enough tension to make the kite initially track up, whereas at higher elevation (right side) the kite began spinning. This is because of the bridle angle β. A small beta angle means the kite is almost perpendicular to the tether, so at low elevation the angle of attack is near 90°. This produces a large drag force and therefore tension. At higher elevations the spine of the kite is almost in line with the wind, producing very little drag.

Figures 2.13 and 2.14 show how the drag force acts to move the kite downwind for

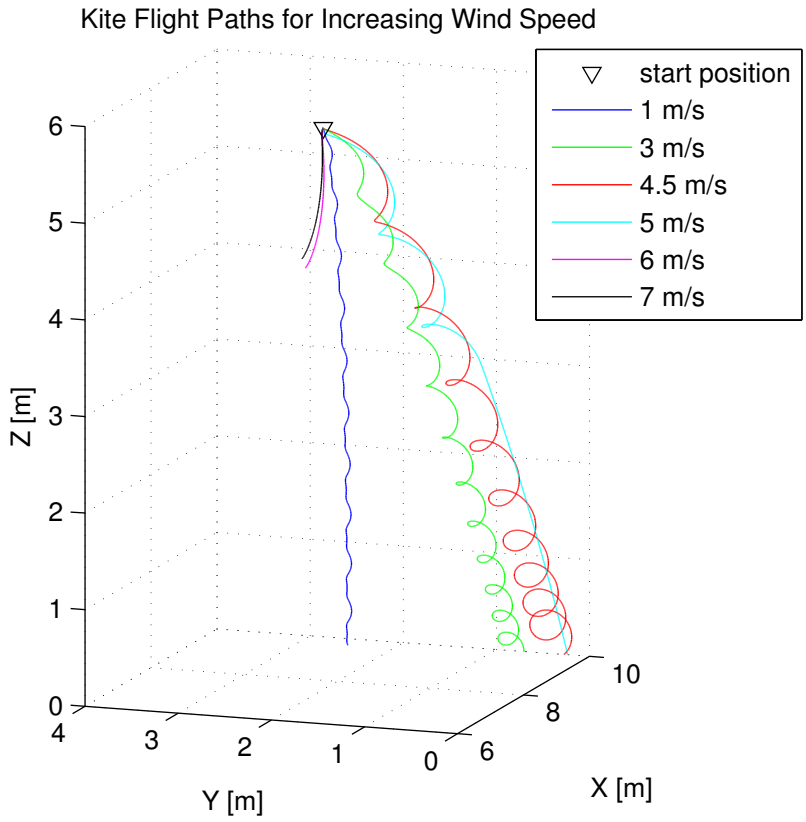


Fig. 2.10: Plot of kite paths for increasing wind speed. Wind is directed along the X-axis. All other conditions held constant. Initial conditions: $r_0 = 10m$, $\theta_0 = 55°$, $\phi_0 = 15°$.

both high and low wind speeds, respectively. This behavior is expected, as regardless of the direction the kite is facing and tracking toward, the drag force will always try to align the tether with the apparent wind. Wind speeds are seven m/s for Fig. 2.13 and five m/s for Fig. 2.14. The wind is directed at 45° from the X axis. A fixed 10m tether is used with various initial release points. The stable point in Fig. 2.13 occurs because of the high wind speed. Stable points aren't desirable, because there is no chance for power extraction. To effectively fly the kite at higher wind speeds would require a stiffer cross-spar.

Control of the kite will be simulated by changing the speed of tether release/intake. Reeling in the kite will increase tension, whereas releasing line will lower it. For a given wind speed, tension can be controlled to make the kite track or spin as desired.

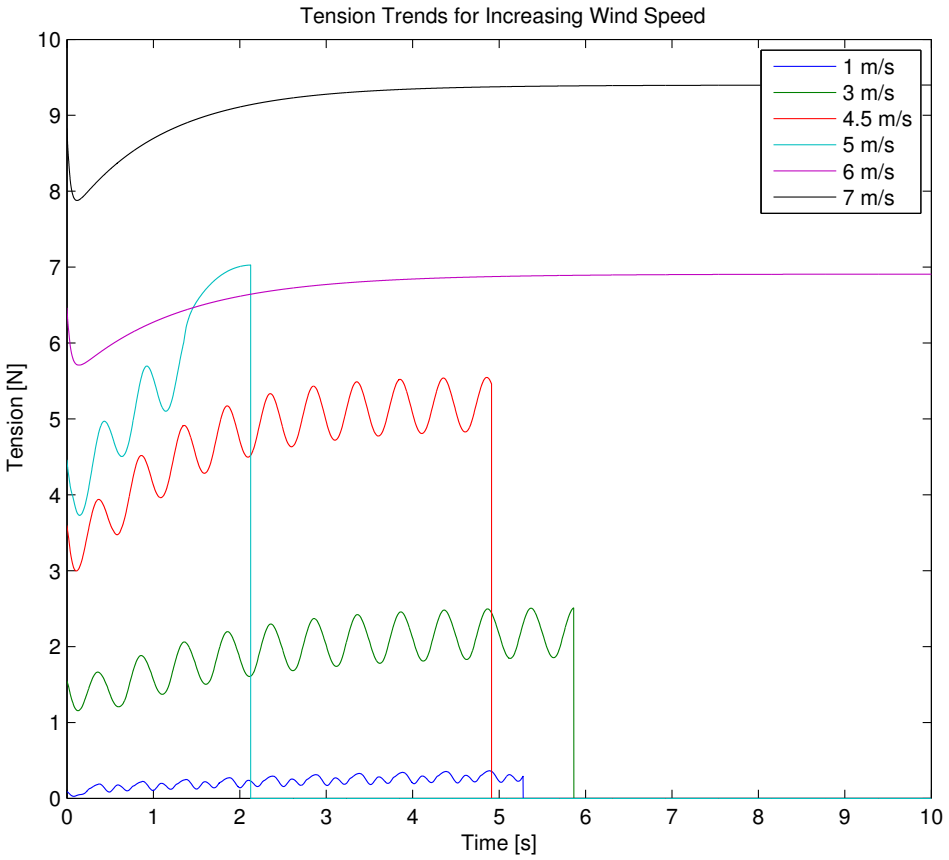


Fig. 2.11: Magnitude of tether tension for increasing wind speed, given the same initial conditions as Fig. 2.10.

Figure 2.15 shows the results of tests run with different tether speeds and the same initial conditions: $r_0 = 10m$, $\theta_0 = 85°$, $\phi_0 = 45°$, $V_\infty = 3m/s$. As expected, when reeling in (blue path) the tension increases and the kite tracks upward until its angle of attack becomes too small to provide enough lift and drag to maintain enough tension for tracking mode. For zero reel speed (black path) the kite spins and falls. When releasing tether (green path) the kite spins and falls while moving downwind. If the release speed is too high, the calculated tension can become negative. This happens because the model assumes a straight tether, so the kite is essentially being "pushed" by the tether. As stated before, if this occurs our overall model is invalid and the simulation exits.

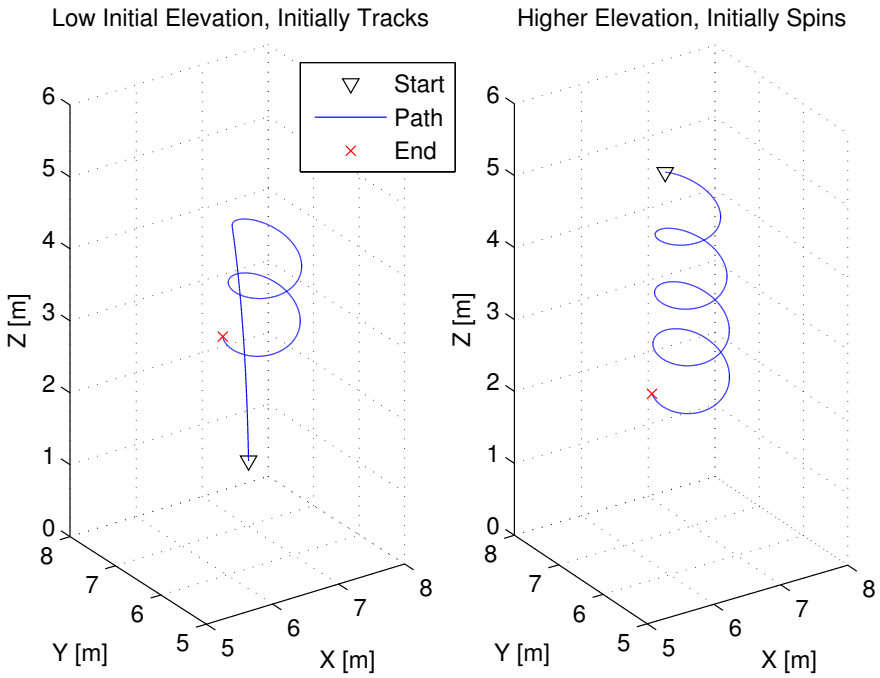


Fig. 2.12: Differences in kite response based on change in initial position. Initial conditions: $r_0 = 10m$, $\theta_{0\ left} = 85°$, $\theta_{0\ right} = 40°$, $\phi_0 = 45°$, $V_\infty = 5m/s$.

2.3 Simulation Conclusions

A numeric simulation of the fighter kite's behavior was created that appears to provide an accurate prediction of the kite's behavior. Several tests have shown that the kite behaves as seen in manual flights. This work has gained insight into the overall dynamics and behavior of these kites, how they respond to changes in tether tension and different wind speeds. Satisfied that this model captures the overall behavior of the specified kite, our next step was to investigate autonomous flight control.

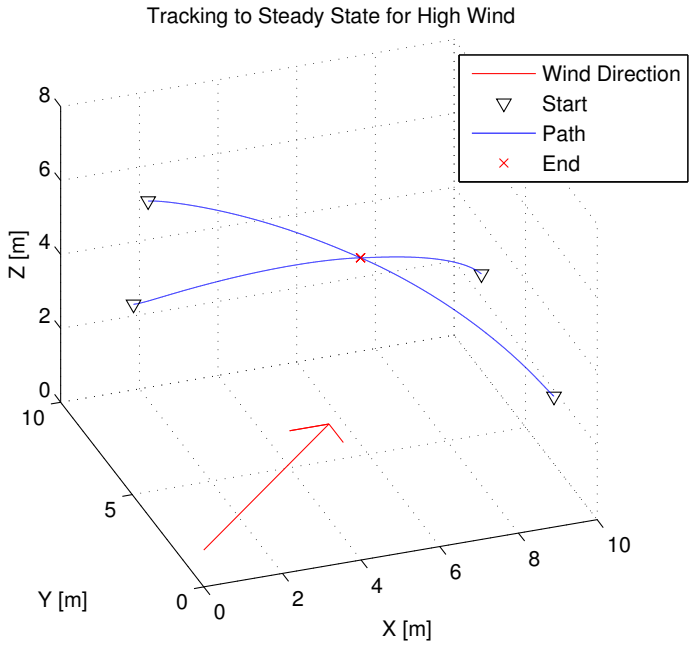


Fig. 2.13: Kite in stable tracking mode being blown downwind. Tether length and wind speed are constant, 10m and 7m/s.

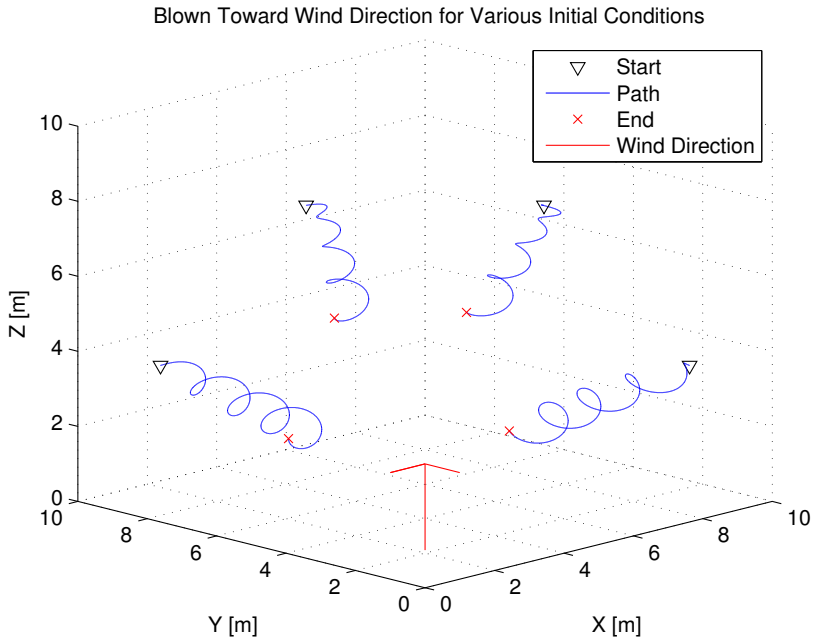


Fig. 2.14: Kite in spin mode being blow downwind as it falls. Tether length and wind speed are constant, 10m and 5m/s.

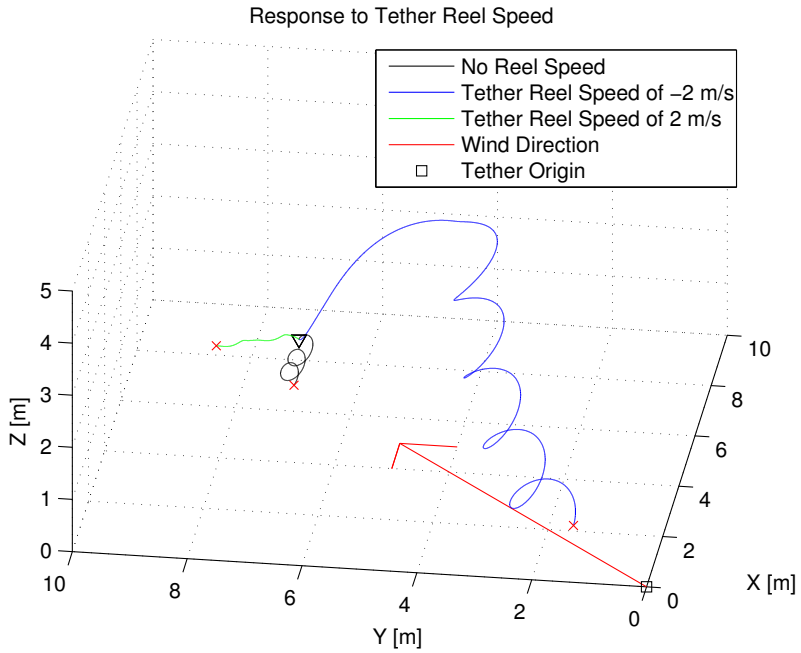


Fig. 2.15: Kite flight path for varying tether intake/release speeds. Initial conditions and wind velocity are the same for each path.

Chapter 3

Simulation: Kite Control

3.1 Initial Controller, with Full Knowledge of Kite

Without control, the kite will either track to a stable position (high wind speeds) or crash into the ground (lower wind speeds). Neither of which are exceptionally interesting or useful. Conversely, a well controlled kite can be useful in numerous applications. Heuristic control algorithms were developed with the goal of indefinite autonomous flight. For development of the initial controller, it was assumed that all information of the kite's position, state vector, current mode, and tether is available for use as control inputs. This is similar to a person flying the kite; both seeing the position and orientation data and feeling the tether in their hand. This provided a reasonable starting point for development of our controller.

The mode-switching kite behavior resulted in the development of situational control algorithms. From the position and orientation of the kite, the algorithm determines what behavior is needed to avoid a crash, then applies a specific feedback loop that will give that behavior. Each control algorithm has multiple feedback loops that it may apply, with goals such as switching from tracking to spinning and maintaining a desired tether length. This basic architecture is shown in Fig. 3.1, where the dashed box can contain various positive or negative feedbacks and gains.

The controller's goal at this time is to keep the kite from crashing into the ground.

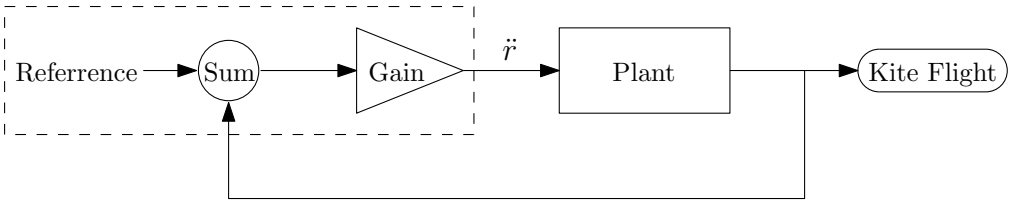


Fig. 3.1: Basic block diagram of control method. The dashed box will change based on which behavior the controller is trying to apply.

Path tracking is not investigated in this work. Since the goal of this work is to maintain altitude, the most important input to the controller is the zenith angle. For small values of the zenith angle, θ, it is relatively unimportant what the kite is doing, but as the zenith angle increases past a critical value the controller must act to turn the kite around and fly it back up or it will continue to fall and crash. To act appropriately the controller must know if the kite is spinning or tracking, and what direction it is facing. The initial, full knowledge algorithm is described below and given in App. A.2.

The first part of the controller checks kite position. If it is above the critical zenith angle, $\theta = 55°$, and therefore in no danger of crashing, the controller recovers or releases tether to maintain a desired reference tether length, r_{ref}. Negative proportional feedback from the tether length is used to determine the desired tether speed. The feedback is scaled by the magnitude of the tension to avoid negative tension. This condition was added because initial results either ran the tether to zero or extended it beyond several hundred meters within a few minutes. Neither of which allows sufficient control in real conditions or shows any promise of a periodic motion.

When in danger of crashing, the controller must check mode (spin or track) and facing direction before acting. If the kite is spinning, the controller waits until it is facing upward ($0 < \psi < \psi_{up}$), then rapidly pulls in tether to increase tension. It uses the positive proportional feedback loop in Eqn. 3.1 to ensure a swift transition into tracking mode, with C as a constant scaling factor. The range of "upward facing" ψ is limited from zero to ψ_{up} because tension fluctuates during the spin mode. As ψ

reaches 180° the kite begins to accelerate downward, increasing tension. If tether is accelerated until the rotation angle passes 180° but is still spinning, the increase in tension on the downward part of the spin could be enough to transition to tracking and start the kite flying straight down. Also, it is preferable to track upward when ψ is near 90° to gain the most altitude. For these reasons, ψ_{up} is generally set at 120°, but can range based on desired path directions.

$$\ddot{r} = -C(P + P_{\text{crit h}}) \tag{3.1}$$

If the kite is tracking and facing downward ($\psi > 179°$), the kite will fly straight into the ground. Tether must be released relatively quickly to switch to tracking mode, but a negative tension has to be avoided. The solution is to simply set $\ddot{r}$ to a scaled value of the tension, as in Eqn. 3.2. So it will quickly release tether, but at a slowing rate as tension decreases. D is a scaling factor to allow more tuning capabilities.

$$\ddot{r} = DP \tag{3.2}$$

Even with the tether length adjustments when the kite is not in danger of crashing, these cases often consumed all of the initial tether length. As tether length decreases, the kite becomes more difficult to control, since a shorter tether means less altitude, and less time for corrective action to prevent crashing. The solution was to slowly reduce tension after the kite switches to tracking mode. If the kite is below the critical zenith angle, tracking, and facing upward the acceleration is defined by Eqn. 3.3.

$$\ddot{r} = E(r_{ref} - r)(P - P_{\text{crit l}}) \tag{3.3}$$

This scaled feedback avoids negative tension and works to optimize tether length to the desired r_{ref} value. E is a scaling factor. A flow chart detailing the full knowledge

controller is shown in Fig. 3.2.

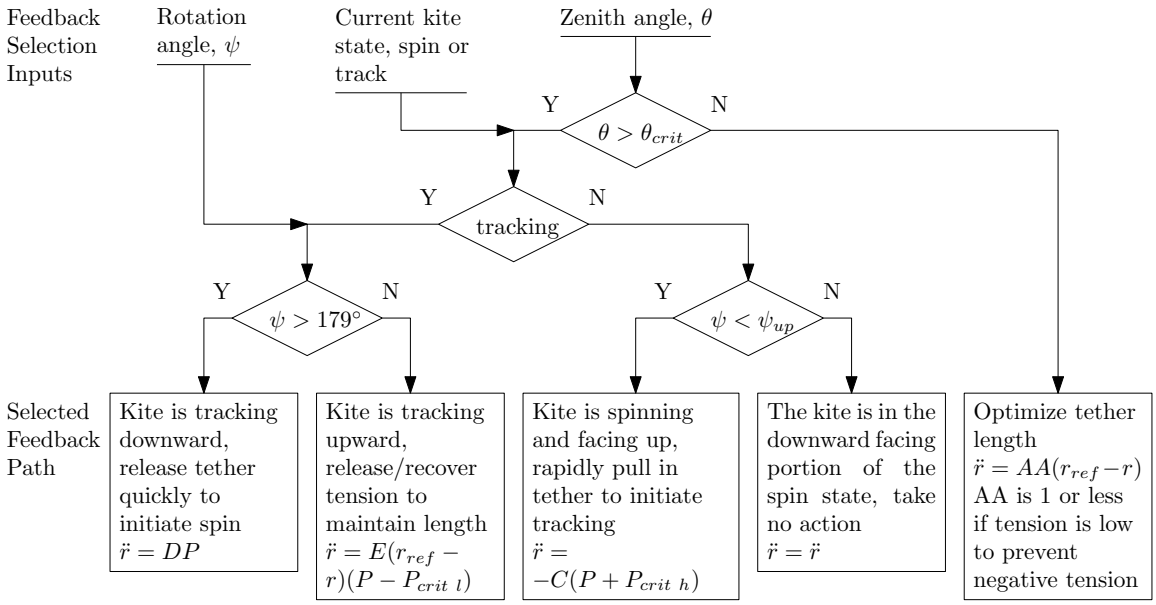


Fig. 3.2: Flow chart of feedback loop selection for the full knowledge controller.

With the use of full state vector information, this control solution was highly successful and stable for a range of the scaling factors C, D, and E and various initial conditions and wind speeds. In constant wind conditions, the controller has been successful for simulated flight tests of over five hours. One such test is shown in Fig. 3.3.

The shape of the periodic motion that the kite settles into depends largely on the values of the scaling factors used in the feedback loops and the wind speed. Changing the scaling values will alter the shape of the path by changing how quickly the kite switches between modes. Figure 3.4 shows the change in path shape with a change in scaling factors relative to Fig. 3.3, as well as the large basin of attraction and apparent stability of the controller.

The controller sets values of the tether acceleration, which are then used to determine tether speed. This was done in the simulation to avoid instantaneous jumps in velocity. Instantaneous changes in velocity are essentially infinite acceleration, and

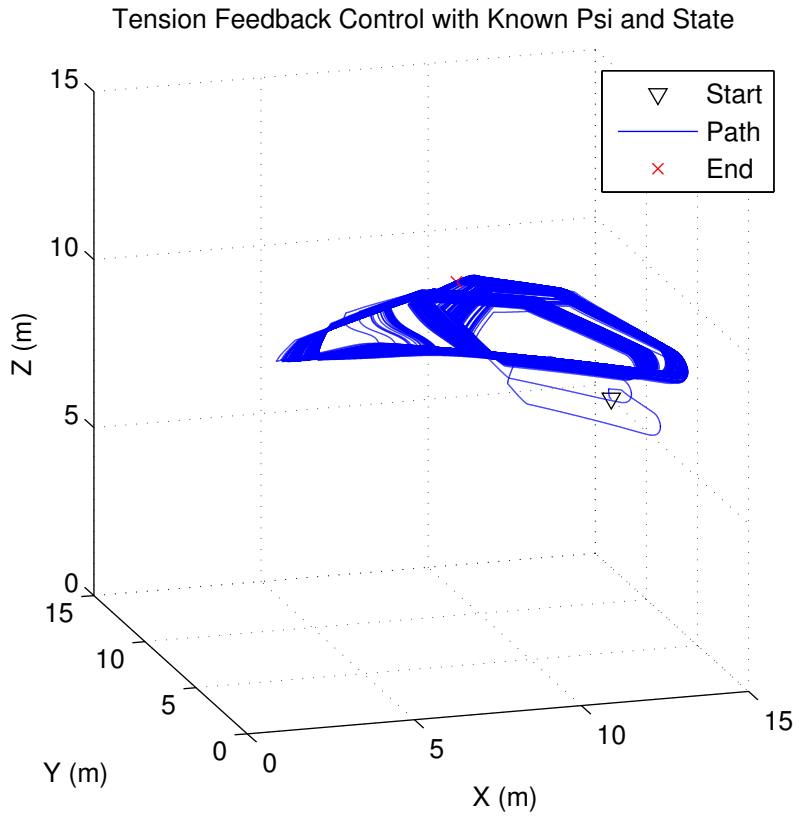


Fig. 3.3: Kite flight path of initial full knowledge controller using tension feedback. Exact information allows quick changes in mode to form a stable loop. $C = 1.8,\ D = 1.1,\ E = 3.0,\ V_\infty = 4.5m/s$.

because tether tension is determined from the tether acceleration, this would produce erroneous infinite tensions. The second reason for setting an acceleration based on tension is that it makes the solution more robust. If each case gave a set value of tether speed or acceleration, the controller is only effective in a limited range of wind speeds. By basing the magnitude of the acceleration on the tension (the driving force between mode switching) the controller can be applied to a wider range of wind speeds.

Autonomous control is relatively simple to implement with complete information of the kite's position and orientation. To experimentally gather this data, a system would require multiple accelerometers, GPS, and rate gyros, some combination

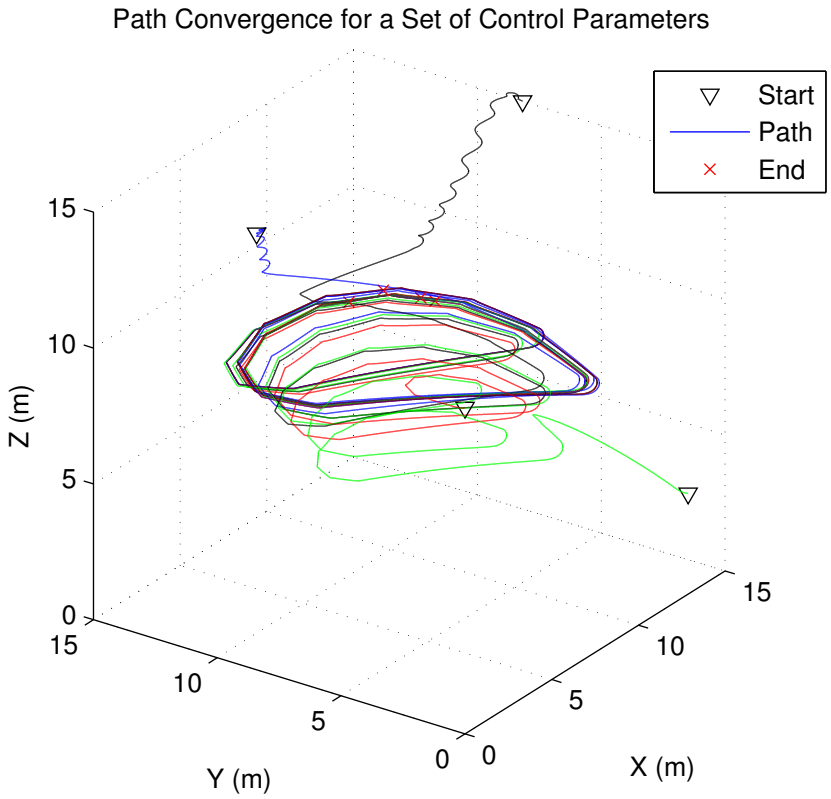


Fig. 3.4: Path convergence for various initial conditions using the full knowledge controller with adjusted feedback loop scaling factors. $C = 2.2,\ D = 2.9,\ E = 0.5,\ V_\infty = 4.5m/s$.

thereof, or a computer vision system. The remainder of this thesis focuses on measurement and control of the kite, with the limitation of only using information from ground-based sensors.

3.2 Reduced Knowledge Controller

Without on-kite instrumentation or an advanced vision system, the full knowledge controller can not be implemented. The input knowledge to the controller was reduced to that which is easily measurable from the ground: zenith angle, azimuth angle, tether tension, and tether length. As the kite flies through the air, the tension vector changes in magnitude and direction. Using the three-axis load cell system that we designed, built, and instrumented (described in Chp. 4), this tether information can

be determined. The exact orientation, velocity and current flight behavior (spinning or tracking) are unknown. This is analogous to a person flying the kite with their eyes closed, just being able to feel and control the tether. Approximations can be made from the kite's position history to provide a rough estimate of some of these values. The full code for each controller discussed is given in App. A.2.

A very simple algorithm was attempted that does not require very much information of the kite's current orientation. It uses the sign of $\dot{\theta}$, the derivative of the zenith angle, to determine the desired rate of tether length, $\dot{r}$. The derivative of the zenith angle is approximated by the second order backwards finite difference method in Eqn. 3.4.

$$f' \approx \frac{f_{i-2} - 4f_{i-1} + 3f_i}{2\delta} \tag{3.4}$$

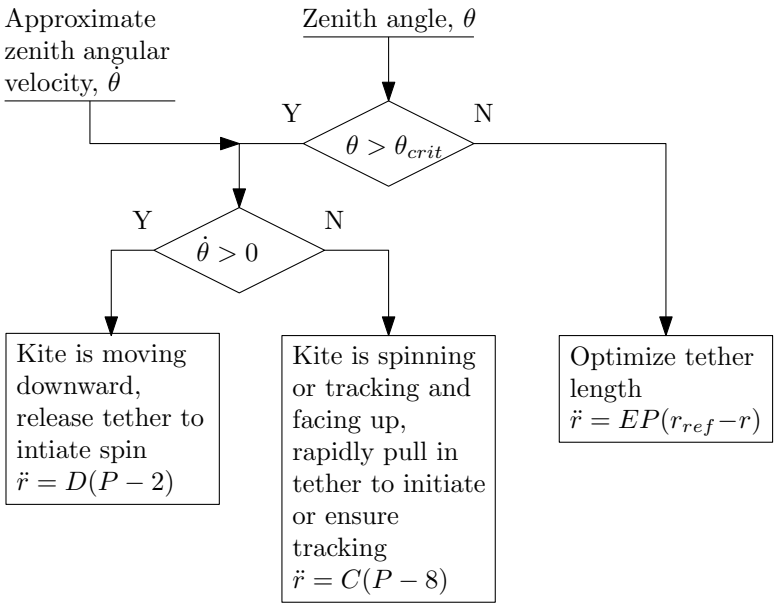


Fig. 3.5: Flow chart describing the feedback loop selection of the simple zenith angular velocity-based controller.

When the kite is in spin mode the zenith angular velocity will fluctuate between positive and negative values. As the kite swoops downward it gains speed. This speed and the spinning behavior of the kite result in a periodic increase and decrease

of zenith angle. The resulting control solution is simple. For values of θ less than $\theta_{crit} = 65°$ (no danger of crashing), the controller attempts to optimize tether length. For zenith angles greater than 65° the sign of $\dot{\theta}$ is considered. If zenith angle is increasing, $\dot{\theta}$ is positive and the kite is heading toward a crash. Tether is released to make the kite spin. If zenith angle is decreasing, $\dot{\theta}$ is negative and the kite is, at least at the moment, moving upward. Whether it is tracking upward or in the upward portion of a spin, the kite is facing up. Therefore, tether is reeled in to ensure that the kite is tracking. A flow chart illustrating the controller is shown in Fig. 3.5. Figure 3.6 shows a successful flight of the zenith angular velocity-based controller.

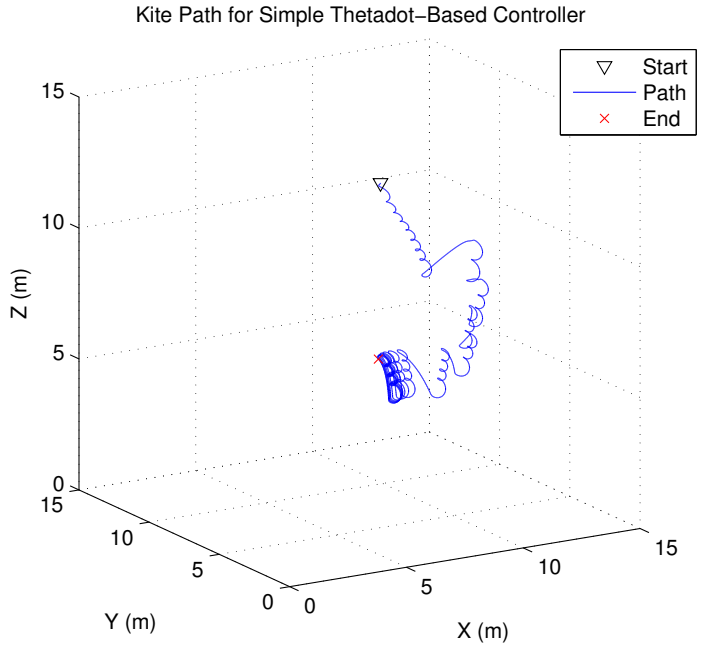


Fig. 3.6: Flight path for controller based on zenith angular velocity. $C = 3.3$, $D = 1.0$, $E = 0.5$, $V_\infty = 4.3m/s$.

This very simplified controller was moderately successful, but lacks the robust stability seen in the full knowledge controller. The final path is stable, but as seen in Fig. 3.7, only with the proper initial conditions will the kite reach the stable path. This is particularly evident from the dark green and light blue paths. Each had the same initial position, with only a small difference in the rotation angle ψ. Though it

appears to produce a stable periodic motion, the controller's success is limited. The main problem is seen in both the light and dark green paths of Fig. 3.7. If the kite begins tracking with ψ near zero or 180°, it won't gain altitude and the controller will eventually reel in the kite. Any disturbance from that stable path could crash the kite.

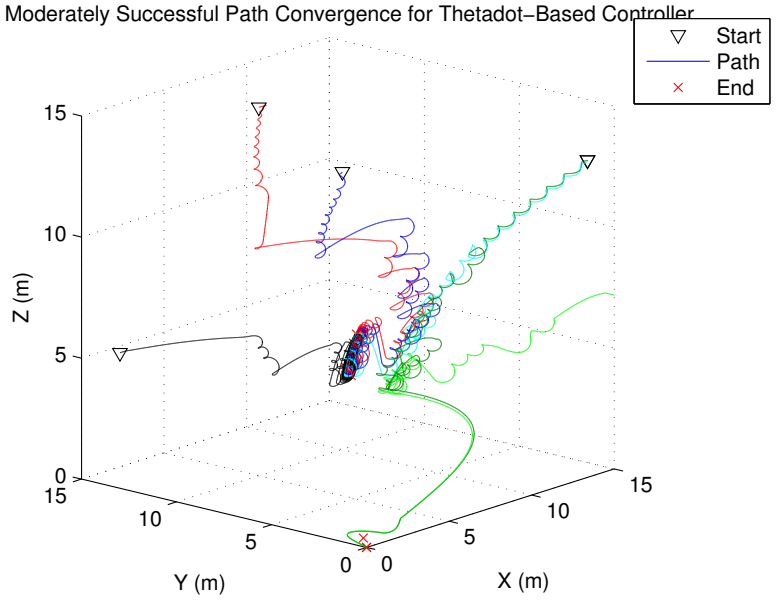


Fig. 3.7: Path of kite under the simple $\dot{\theta}$-based control for various initial release points. Only some initial conditions will produce stable motion. Conditions identical to Fig. 3.6.

The problem arises because the controller does not know if the kite is spinning or tracking or what direction it is moving in. Another difficulty with this simple controller is that the functional range is small, and it often ends up crashing from running out of tether. During the spin mode the controller is cycling between releasing and recovering tether. The recovery must be quick enough to ensure that the kite begins tracking, whereas the release has to be slower to prevent negative tension. During tracking mode the controller doesn't know that the kite is tracking, so it continues to recover tether faster than necessary. During spin mode the kite doesn't know it is spinning, only that $\dot{\theta}$ is positive and that it needs to let out tether to ensure that the kite is spinning. The limited success of this controller showed the importance

of knowing the kite's mode.

From early simulation results, it is shown that the magnitude of the tension and zenith angle fluctuate at the same frequency as the spin rate of the kite. During controlled flight, however, this fluctuation is less prominent due to the acceleration of the tether. Recognizing the signal fluctuation and shape quickly and reliably is also difficult and requires the kite to stay in the spin mode for at least a full rotation. An alternative mode approximation was found by stepping back to look at the kite's overall behavior. The most significant and recognizable difference between the spinning and tracking modes is the magnitude of $\dot{\psi}$, the spin rate of the kite.

To approximate the spin rate, the spin angle was approximated by using the position history of the kite to approximate the velocity by finite difference methods. The kite flies generally in the direction it is facing. The velocity vector gives the direction the kite is moving in, so when projected into the plane perpendicular to the tether, it shows the facing direction, and thus the rotation angle, ψ_{approx}. The derivative of the rotation angle was then approximated using Eqn. 3.4. We then assumed that when the magnitude of $\dot{\psi}_{approx}$ is greater than two rad/s the kite is spinning, otherwise it is tracking. A value of two radians rather than zero was used to avoid errors caused by slight changes in the kite's direction that occur while tracking due to the forces acting on the kite. ψ_{approx} is compared to the actual value of ψ in Fig. 3.8. The value of two rad/s was determined from several simulation runs to see what magnitude was most effective at neglecting the small changes of tracking mode, while still noticing the more significant change of the spin mode. Of note is the poor approximation of ψ for the initial entrance into spin mode. This is caused by the transition of the kite from tracking to spinning mode. The zenith angular velocity, $\dot{\theta}$, doesn't change sign until the spin becomes fast enough to increase elevation at the very top of the spin.

Supplied with knowledge of the kite's facing direction and mode, an attempt was

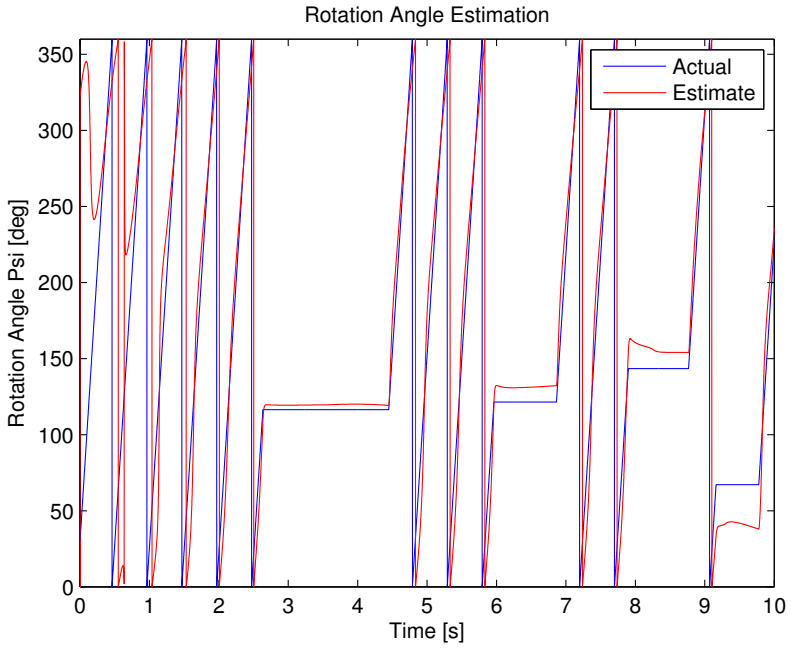


Fig. 3.8: Comparison of actual and approximated rotation angle ψ. The approximation improves with kite speed.

made to simply apply the full knowledge control algorithm and replace the previously known values with the approximations. Unfortunately, the full knowledge controller proved too aggressive; it operates by quickly changing the kite's mode between tracking and spinning. The high error apparent in the facing direction when entering the spin mode led to mode switching at the wrong orientation and often crashed the kite before it could recover.

We thought a step back in complexity might provide a better solution, and so returned to further investigate the $\dot{\theta}$-based controller with these new approximations. Approximating the rotation angle ψ provided detailed information, but the error was high. The approximation of $\dot{\theta}$ provides less overall detail of the kite's orientation, but is has little error. The mode approximation was successfully incorporated into the simple controller to correct the tether length issue. A flow chart of the $\dot{\theta}$ and mode approximation based controller is given in Fig. 3.9. The controller works similarly to

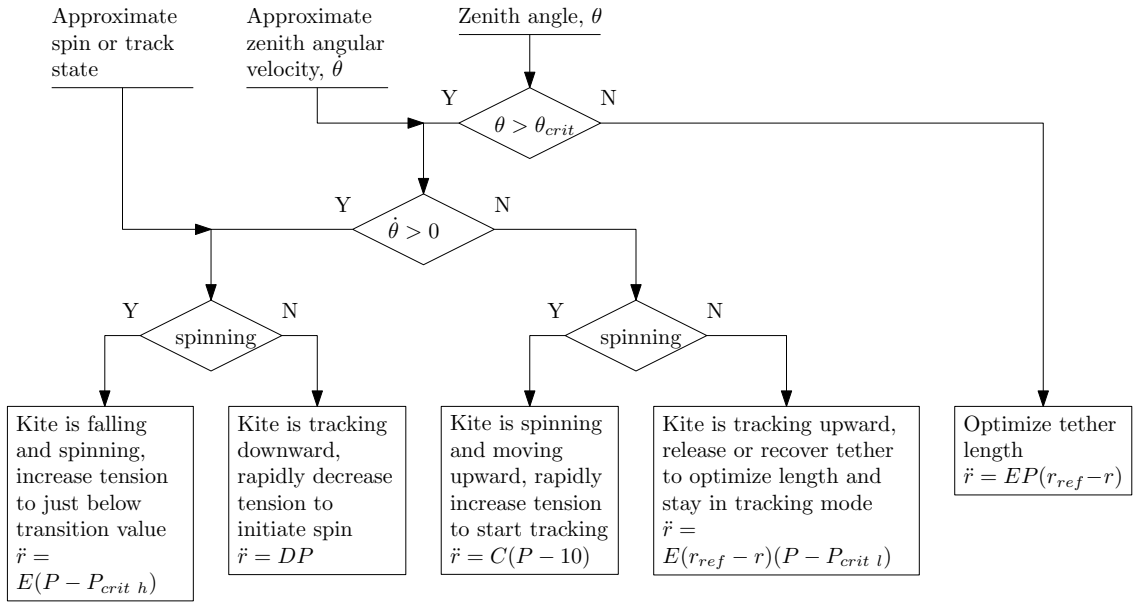


Fig. 3.9: Flow diagram used by the reduced knowledge controller to select feedback path based on $\dot{\theta}$ and an approximation of the current mode.

the full knowledge controller, with the major difference of being based on approximations. If the kite is above θ_{crit} then the feedback is $\ddot{r} = EP(r_{ref} - r)$. Again, this maintains tether length and avoids a negative tension. If θ is below the critical angle then the sign of $\dot{\theta}$ and the spin/track mode of the kite are checked. For positive $\dot{\theta}$ and spin mode, the tension is increased to a value near the tracking transition zone. This makes the transition to tracking occur relatively quickly when desired without a huge change in tether speed. For positive $\dot{\theta}$ and tracking mode the kite is going down and must switch to spin quickly. For negative $\dot{\theta}$ and spin mode the kite is in some upward facing direction and tension must be increased to get the kite to track while it is still facing up. As seen in Fig. 3.9, a negative feedback of tension is used for this case instead of the positive feedback of the full knowledge controller. This makes the controller slower to respond and shift to tracking, but it is more recoverable from if an error is made and tracking starts in the wrong direction. The addition of spin/-track approximation helped the algorithm be more effective and produce the periodic motion seen in Fig. 3.10.

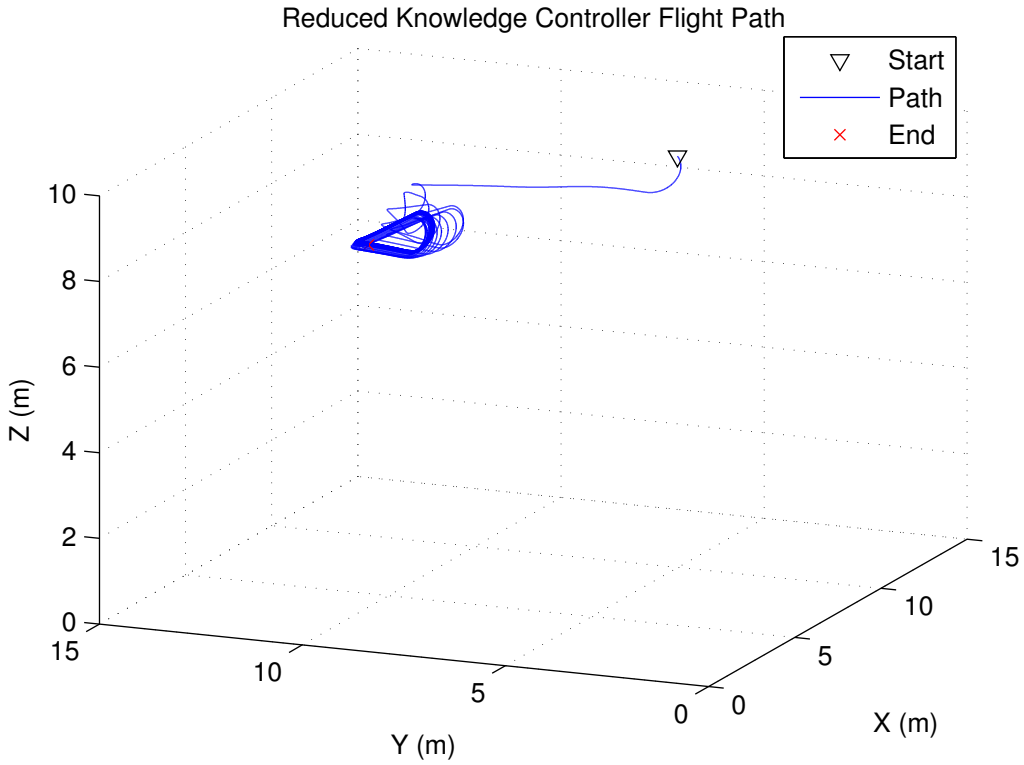


Fig. 3.10: Flight path of reduced knowledge controller that uses position history to approximate if the kite is spinning or tracking. Mode knowledge provides better tuning and control of the kite, producing more stable solutions. $C = 1.5,\ D = 4,\ E = 1.5,\ V_\infty = 4.5m/s$.

Figure 3.11 shows the wide convergence and stability of the reduced knowledge control solution. The gray section in the center is from the overlap of the nearly identical paths. The black kite path takes considerably longer to reach the cyclic path because it starts with a much shorter tether and closer to the ground than the other initial conditions. The final cyclic path is isolated in Fig. 3.12. The path appears to be stable, but not completely periodic. This may be due to the system behavior itself, but is more likely from simulation error. The controller and mode-switching choices are only checked at every time step, but the condition of the kite may have changed at any intermediate time between steps. This slight time error could be the cause of variance from a single repeating path. Even with this error the path appears to remain within an area and does not drift away.

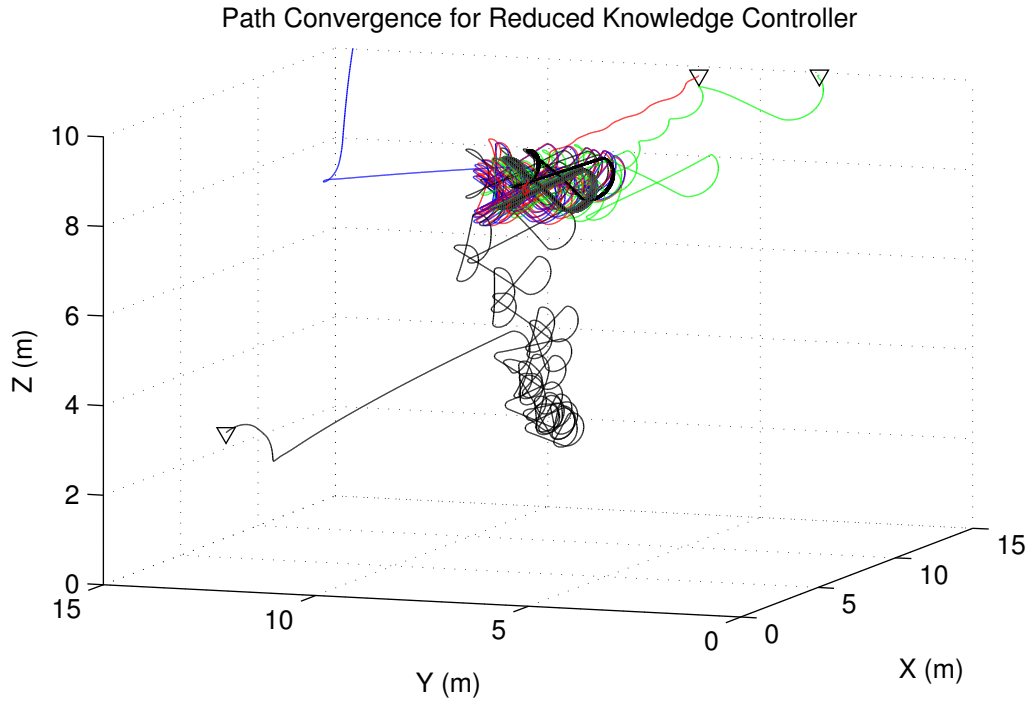


Fig. 3.11: Converging flight paths of the kite released from various initial conditions using the same reduced knowledge controller as Fig. 3.10 with adjusted scaling values. $C = 1,\ D = 4,\ E = 1.5,\ V_\infty = 4.5m/s$.

The success of this controller led to further probing of its capabilities. The full knowledge controller is only truly effective at a small range of wind speeds between 4 and 5m/s. Above 5m/s the full knowledge controller will track to a steady point with a slight oscillation driven by the controller overshooting when trying to optimize tether length. The simple $\dot{\theta}$ controller is still functional at high wind speeds, but encounters the same issue of tracking in horizontal directions, and is not stable. The gains of the more complex reduced knowledge controller were adjusted to find a set that functioned from 4.3m/s to 6m/s. The resulting flight paths at speeds of 4.3, 5, 5.5, and 6m/s are shown in Fig. 3.13.

Though the controller appears to fly the kite well at a range of constant wind speeds, a truer test of the algorithm is to apply more realistic wind conditions, varying the magnitude and direction of the wind. Wind magnitude was set at 4.5 m/s and

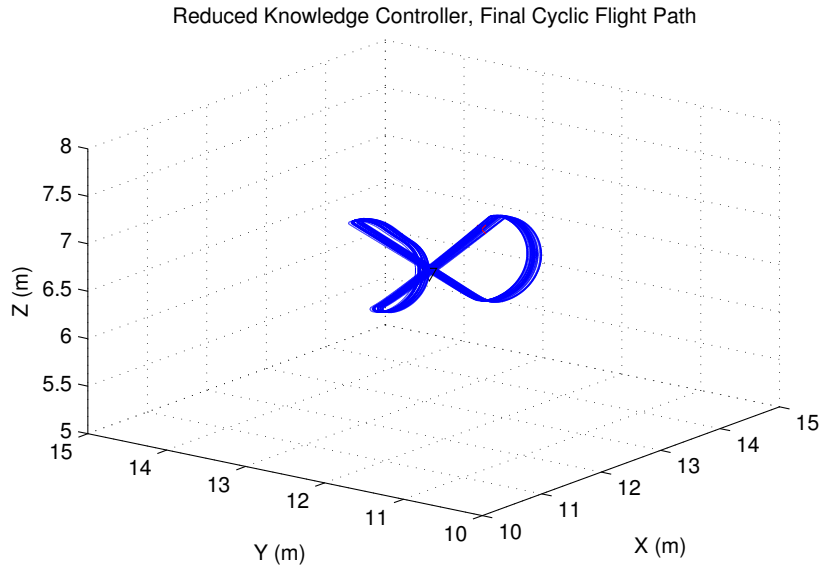


Fig. 3.12: The final cyclic path from Fig. 3.11 is shown with initial conditions taken directly from the final state vector of the blue path in Fig. 3.11.

given random and sinusoidal fluctuations as shown in Eqn. 3.5.

$$|\vec{W}| = 4.5 + .25\cos\left(\frac{time}{10}\right) + rand(0\ to\ 0.5) \tag{3.5}$$

Gradual changes and instantaneous shifts in direction were applied to the kite. Initially, the wind vector was along the X-axis, then it slowly shifts to the Y-axis. The wind vector then rapidly changes direction by 90° back to the X-axis, followed by smaller 45° jumps, and finally settles to a set direction along the X-axis. A small, random vertical disturbance of ±0.1m/s is added to simulate small rising and falling air currents. The magnitude and individual components of the wind are shown in Fig. 3.14. The more complex reduced knowledge controller was used to fly the kite, resulting in the path shown in Fig. 3.15.

Large changes in wind speed can prove difficult to adjust to, but as long as the magnitude remains above the minimum value of 4.3m/s, the controllers have been able to recover. Significant gusting will introduce more error in the mode approximations used by the complex reduced knowledge controller. Small and/or gradual shifts in

wind direction are not highly disruptive to the controller, because it only acts to maintain altitude. Large, rapid changes in wind direction can generate a negative tension if the kite is oriented to another direction when the wind changes.

3.3 Kite Control Conclusions

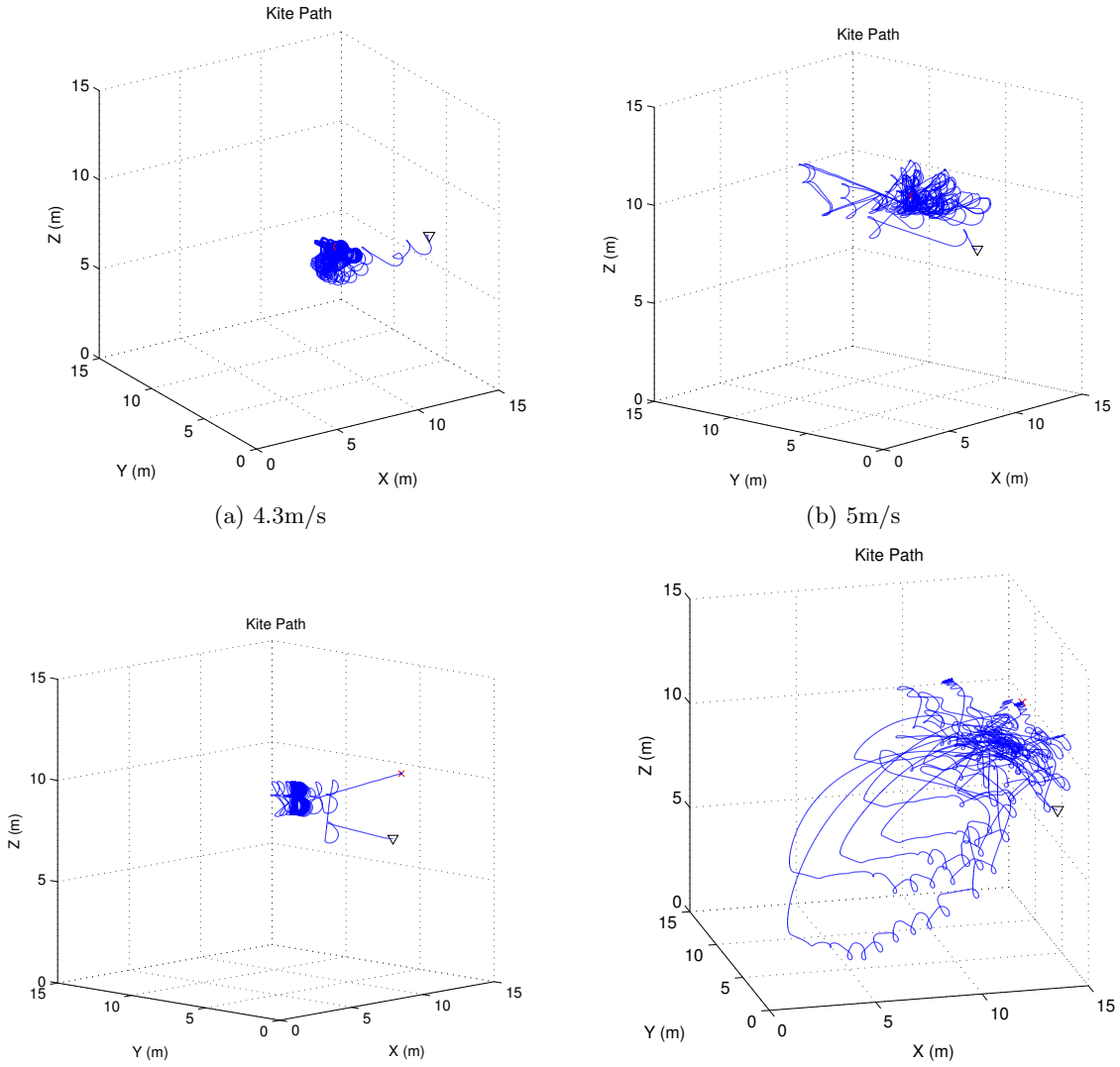
The focus of this work was to investigate the novel control and actuation method that these kites employ and determine the extent of autonomous control possible. Various control algorithms were developed that show potential for sustained autonomous flight. They have been shown to fly the kite in a range of wind conditions and appear to functional well when wind remains above a minimum value.

With complete information of the kite's state vector, highly exact control is feasible. The kite mode can quickly be forced between spinning and tracking to keep the kite in the air and flying along a very cyclic path. By reducing that knowledge to ground measured values, there is a loss in the precision of control over the kite's behavior. Multiple rapid changes are no longer feasible, as the error in the mode and facing direction approximations can easily crash the kite. More relaxed controllers were developed that don't need as precise information to fly the kite. Both reduced controllers (simple $\dot{\theta}$-based and the more complex $\dot{\theta}$ with mode approximation) could prove useful in experimental flight. The more complex controller has produced much more stable periodic motion and appears to function over a wider wind speed range. The large advantage of the simple $\dot{\theta}$ controller over incorporating the mode approximation is that it only uses one differentiation approximation, as opposed to the four steps used to reach the mode approximation. In simulation this is insignificant, as the signal is smooth, but any noise could make the controller difficult to execute.

Variation in wind velocity can be a large problem for autonomous flight, path tracking, and/or power production of any kite system. Balancing the need for keeping the system aloft and power extraction in real wind conditions has many challenges.

For the tension-based control system, this is especially true. Alternating high and low winds can fluctuate the tension and thus the kite mode faster than the controller has to time to correct for.

The removal of instrumentation and actuation from the kite was an additional goal, with the overall intent of reducing system complexity. These results show that a kite can be completely controlled from ground-based sensors and tether tension. Though other work in single-line kites may return actuation and sensing to the kite, this simplification can be applied to various tethered kite systems. While a reduction in sensing and actuation can limit overall control precision (and thus power extraction), it also reduces system weight, cost, and complexity. Simpler systems are easier to implement in real world conditions. These benefits should be weighed against each other in overall system design.



(a) 4.3m/s

(b) 5m/s

(c) 5.5m/s

(d) 6m/s

Fig. 3.13: Controlled kite flight in various wind speeds, using the $\dot{\theta}$ and mode approximation controller. $C = 0.8,\ D = 4,\ E = 1.8,\ \theta_0 = 70°,\ \phi_0 = 25°,\ r_0 = 18m.$

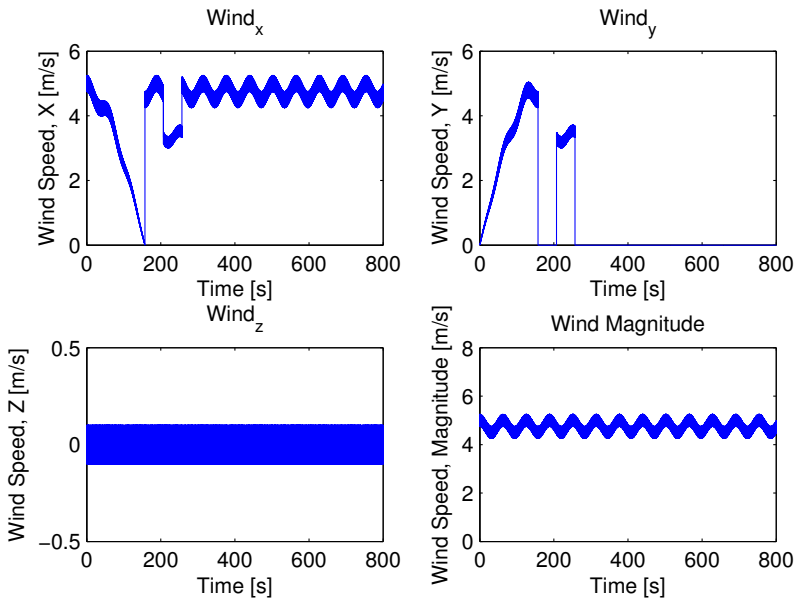


Fig. 3.14: Variable ambient wind velocity used to simulate extreme wind direction shifts. Both smooth transitions and jumps in direction are shown.

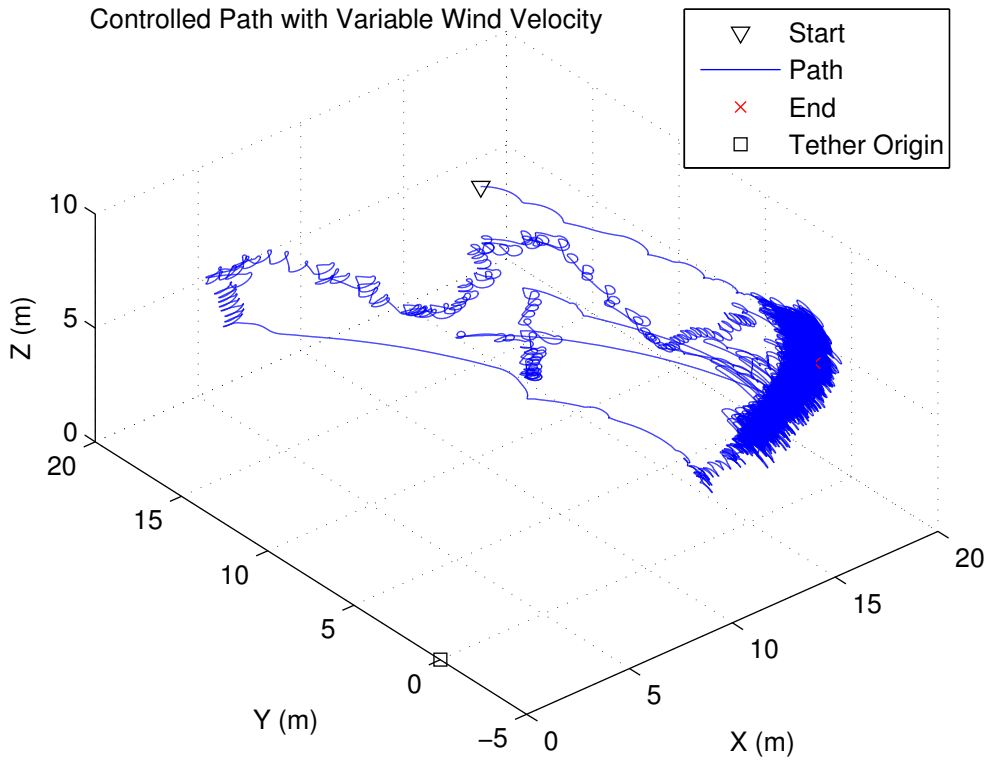


Fig. 3.15: Controlled kite flying in variable winds. The $\dot{\theta}$ and mode-based reduced knowledge controller was used with: $C = 1.5,\ D = 4,\ E = 1.5$.

Chapter 4

Experimental Test-bed Development

4.1 Theoretical Design

We have strong simulation results, but to progress this research accurate experimental flight measurements and control testing are needed in real wind conditions. An experimental measurement and data acquisition system was design and constructed to measure the tension vector and tether length. The initial design was based on experimental work by Lansdorp *et al.* [14] that measured tether tension for a fixed length tether, shown in Fig. 4.1 . This design was adjusted because the kite used in

Fig. 4.1: Fixed tether length load cell used by Lansdorp *et al.* Image taken from [14].

this thesis relies on rapid changes in tether length to change the tension and control the kite's behavior. Three mutually orthogonal strain gauges are aligned to intersect

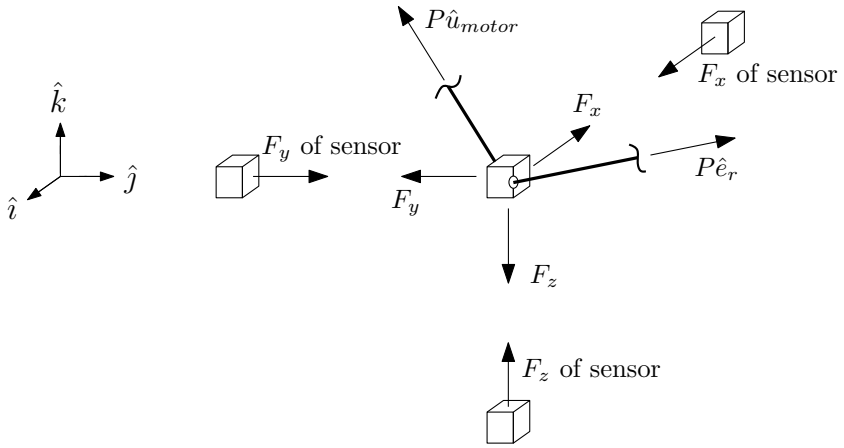


Fig. 4.2: Free body diagram of a tether segment and the pinhole guide block of the measurement system. Because of the large mass difference, any acceleration of the tether is assumed negligible.

at the center of a low friction pinhole guide. The kite's tether passes through this guide to a reel attached to the drive motor. The kite's position is defined in spherical coordinates by the unwound tether length along the unit vector $\hat{e}_r$. The encoder on the drive motor tracks tether length, which can be used to approximate tether speed.

The tether tension and direction are calculated from the summation of forces on the rig based on the free body diagram of the tether segment and pinhole shown in Fig. 4.2. Because the tether mass is much smaller than the structure mass, the acceleration of the tether is assumed to be negligible, thus the tether segment and measurement structure are treated as a single body, with $\vec{a} = 0$ in Eqn. 4.1.

$$\vec{F} = m\vec{a} \tag{4.1}$$

Summation of forces at the guide results in the vector Eqn. 4.2, which (when combined with Eqns. 4.3 and 4.4) has three unknowns: tension magnitude, kite azimuth angle ϕ, and kite zenith angle θ.

$$P\hat{e}_r + P\hat{u}_{\text{motor}} = F_x\hat{\imath} + F_y\hat{\jmath} + F_z\hat{k} \tag{4.2}$$

The unit vector $\hat{e}_r$ is along the tether, from the pinhole toward the kite, given in Eqn. 4.3.

$$\hat{e}_r = \sin\theta\cos\phi\hat{\imath} + \sin\theta\sin\phi\hat{\jmath} + \cos\theta\hat{k} \tag{4.3}$$

Equation 4.4 defines the fixed unit vector $\hat{u}_{motor}$, which goes from the guide to the top of the reel. The angles χ and δ are constant azimuth and zenith angles of -135° and 63°, as illustrated in Fig. 4.3.

$$\hat{u}_{motor} = \sin\delta\cos\chi\hat{\imath} + \sin\delta\sin\chi\hat{\jmath} + \cos\delta\hat{k} \tag{4.4}$$

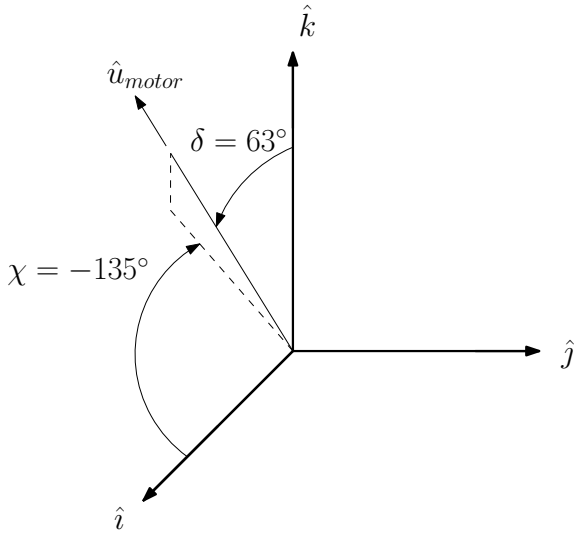


Fig. 4.3: Visual definition of $\hat{u}_{motor}$ unit vector direction from the pinhole to the edge of the motor reel.

With the assumption that tension is the same in both segments, the system of equations was solved to yield the scalar Eqns. 4.5, 4.6, and 4.7.

$$P = \frac{F_x^2 + F_y^2 + F_z^2}{2F_z\cos\delta + 2\sin\delta(F_x\cos\chi + F_y\sin\chi)} \tag{4.5}$$

$$\theta = \arccos\left(\frac{F_z}{P} - \sin\delta\right) \tag{4.6}$$

$$\phi = \arctan\left(\frac{F_y - P\sin\delta\sin\chi}{F_x - P\sin\delta\cos\chi}\right) \tag{4.7}$$

4.2 Construction and Calibration

The drive motor and sensors were mounted on a rigid aluminum frame. Connecting rods from the strain gauges converge to a point and are attached to the guide holder by set screws. The pinhole guide is mounted in the small aluminum holding block to align it with the strain gauges, and allow for replacement of the guide as needed. The full system is shown in Fig. 4.4.

The general ability of the system to fly the kite was been tested manually before the instrumentation was complete. The kite reel was connected to the motor with some initial slack available to launch it into the air. The kite was launched by hand and once the kite was aloft, we dropped the tether from our hands and began adjusting motor speed using the potentiometer shown in Fig. 4.4. Though moderately successful, this proved difficult, as we had no feel for the magnitude of tether tension or release/recovery speed. It showed that the motor and test-bed could physically fly the kite and that the motor transient response can be accounted for.

Calibration of the individual load cells was conducted by orienting the entire system to align the measured axis with the vertical. The pinhole guide and holding block were removed and known masses were hung directly from the connecting rod. Strong linear fits ($R^2 > .9999$) were recorded for each axis. A full description of the calibration and resulting data is given in App. C.2.

With each separate axis calibrated, the performance of the system as a whole was determined. Initial testing showed poor measurement of tension magnitude and direction. The main reason was that the offsets originally determined from the calibration of each load cell were not applicable for the overall system. When the connecting rods are attached to the block holding the guide, they produce a slight pre-load on the

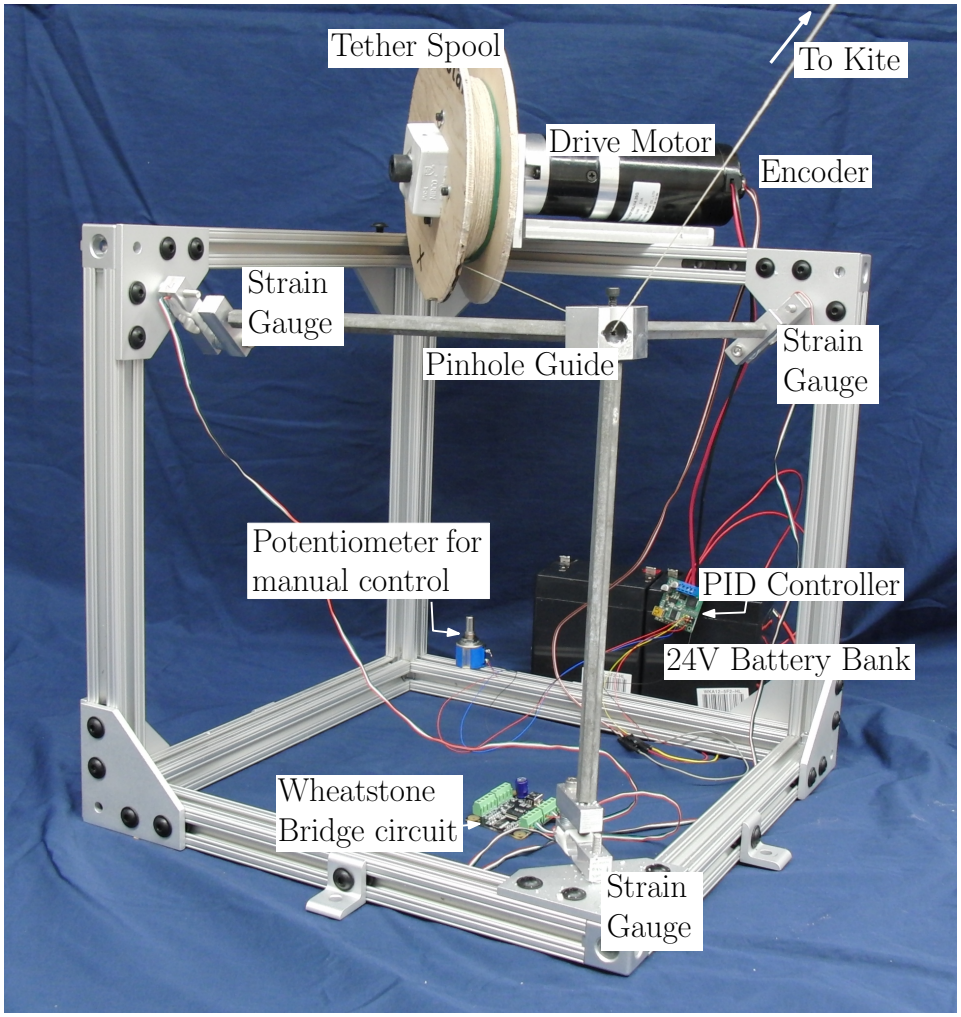


Fig. 4.4: Experimental prototype configured with the motor controller to receive analog voltage inputs from a potentiometer, allowing for manual speed control.

load cell that the former offsets did not account for. This was solved by calculating new offsets at the start of the data acquisition system. The system starts under zero load and takes an average value to be the new offset before entering the main loop. This allowed much more accurate readings.

The system's accuracy was measured using a tripod, low friction bar, and known masses. Positioning the tripod relative to the measurement system set the tether direction. A protractor and plumb bob were used to measure the azimuth and zenith angles to within one degree. The kite tether ran from the motor reel through the

pinhole and over the low friction bar on the tripod. The masses were suspended from this end and ranged from 100g to nearly 2kg. The setup is shown in Fig. 4.5.

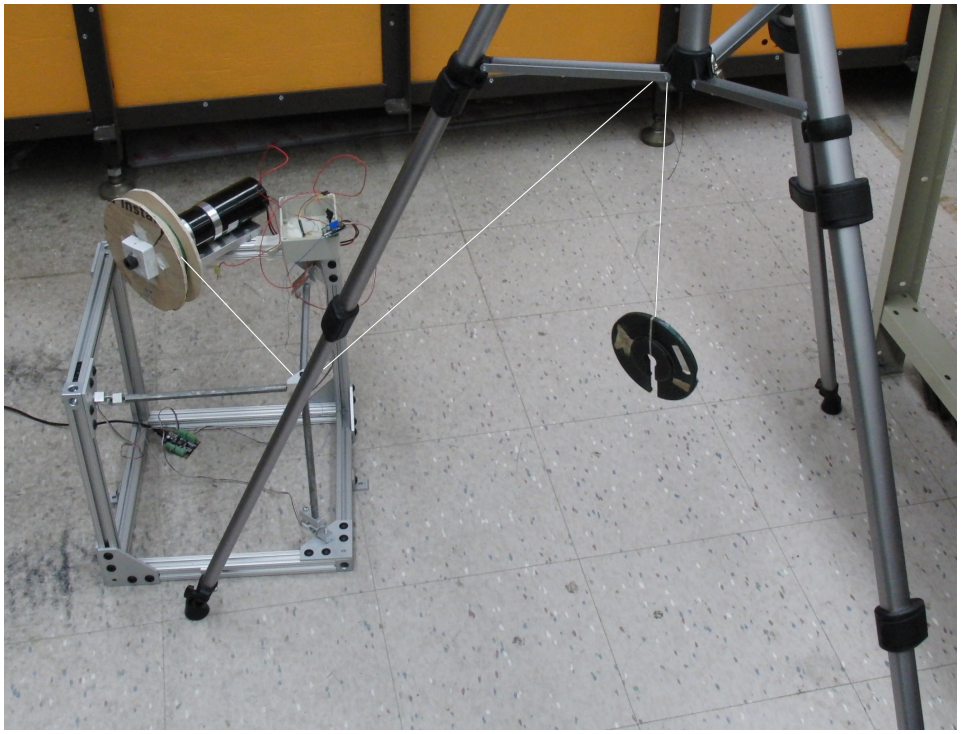


Fig. 4.5: Tripod and hanging mass arrangement used to test measurement system accuracy.

The system was sensitive enough to capture the oscillation in tension of a swinging weight. Final results show that the system is accurate within five degrees of the zenith and azimuth angles. Tension magnitude was also very accurate, ranging from one to ten percent below the actual value. Part of this error is attributed to friction in the hanging system. The other main source of error in the system is cross stresses. The strain gauges of the load cells are most sensitive to shear, but will register some transverse loading. The sensor does not distinguish between axial and cross strains, it just provides a voltage proportional to the total strain on the cell. The mechanical design of the system attempted to minimize this transverse loading. The holes in the ends of the connecting rods were slightly over-drilled for the desired thread size, and are thus compliant by a few degrees in any off-axis motion, but still maintain axial

grip. This "wiggle room" significantly minimizes the off-axis loading seen by each load cell when compared to measurements with the connecting rods tightened snug against the load cell and guide. Overall, the system works quite well and provides a good indication of the kite's position and should prove useful for future experimental research.

Chapter 5

Recommendations for Future Work

This thesis has shown that indefinite autonomous control of a single-line tension controlled kite is possible. As this is a novel actuation method, there is vast potential for following research and it has proven difficult to define a stopping point. Numerous aspects of this work warrant further investigation; path tracking, power production, kite design, robustness, experimental comparison, simulation fidelity.

Sustained flight has been shown possible, though for most applications just keeping the kite in the air is not enough. Accurate control of kite path is needed for effective power production and aerial surveying. Smooth path control is somewhat limited by the behavior of the kite studied in this thesis. There has been significant study of path optimization and robust control of kites for power production systems that can be leveraged to apply to these kites. Also, the simulation is available for use in exploring single-line mode-switching kite design; from simple geometry changes to drastic differences in steering and overall kite behavior. One of the main lacking areas of research is in kite design itself. Of the numerous publications on mechanical pumping systems, only a small handful have designed the kite itself specifically for energy extraction.

The next large step in this research is to conduct flight tests of the control algorithms and the test rig itself. Having built this initial prototype breaks down a significant barrier to future students wishing to conduct experimental kite research.

The main issue that may need to be corrected is system noise. The derivative approximations used in the control algorithms estimate more information from the position history of the kite by using smooth simulation data. Whereas the noise in the actual system may prevent proper implementation of the controller algorithm. Improved filtering or other methods may be needed to produce a useful signal. As stated before, the approximation of kite information introduces error to the control solution. Noise will only act to amplify that error. System latency could be a huge factor as well. It may be desirable to slow the kite down for experimental flights. This can be done by adding a ribbon-type tail to the kite [13].

There is also vast room for increasing the simulation accuracy. The mode-switching behavior of the kite is based on a change in tension, but this change could occur any time between two time steps. Including event detection in the simulation to find the precise time of transitions and updating the integration at that time will improve accuracy. The model bases lift and drag forces from steady-state lift and drag coefficients, though there is significant transient in the system. The tether can be improved to include drag, elasticity, sag, and mass. There has been a wide range of experimental and numerical studies involving loading of cables and kite tethers that could be applied. Other significant changes would be to model the kite as a body rather than a point mass, base the spin and roll behavior on aerodynamic analysis, and include moment coefficients. By treating the kite as a rigid body and including moment coefficients, the bridle angle β could made a variable instead of a constant. The drive motor's transient response could also be included in the simulation, though it is a minimal change. Though significant changes can be made to improve the model, the payoff in accuracy may not be sufficient to warrant the time investment. The most beneficial would be to include tether drag and sag, as these would have direct impacts on the tether tension vector and thus the control implementation.

My final recommendation is to go fly kites.

Bibliography

[1] Anderson, J. *Introduction to Flight*, 7 ed. McGraw-Hill, New York, 2011.

[2] Archer, C. L., and Caldeira, K. Global assessment of high-altitude wind power. *Energies* (2009), 307–319.

[3] Argatov, I., Rautakorpi, P., and Silvennoinen, R. Apparent wind load effects on the tether of a kite power generator. *Journal of Wind Engineering and Industrial Aerodynamics 99*, 10 (Oct. 2011), 1079–1088.

[4] Argatov, I., and Silvennoinen, R. Structural optimization of the pumping kite wind generator. *Structural and Multidisciplinary Optimization* (2010), 585–595.

[5] Breuer, J. C., and Luchsinger, R. H. Inflatable kites using the concept of Tensairity. *Aerospace Science and Technology 14*, 8 (Dec. 2010), 557–563.

[6] Canale, M., Fagiano, L., and Ippolito, M. Control of tethered airfoils for a new class of wind energy generator. *Decision and Control,* (2006), 4020–4026.

[7] Canale, M., Fagiano, L., and Milanese, M. High altitude wind energy generation using controlled power kites. *IEEE Transactions on Control Systems Technology 18*, 2 (2010), 279–293.

[8] Christoforou, E. G. Angular elevation control of robotic kite systems. In *IEEE International Conference on Robotics and Automation* (2010), pp. 614–619.

[9] Fagiano, L., Milanese, M., and Piga, D. High-altitude wind power generation. *Energy Conversion, IEEE 25*, 1 (2010), 168–180.

[10] Fagiano, L., Milanese, M., Razza, V., and Gerlero, I. Control of Power Kites for Naval Propulsion. 4325–4330.

[11] Houska, B., and Diehl, M. Robustness and stability optimization of power generating kite systems in a periodic pumping mode. In *IEEE Multi Conference on Systems and Control* (2010), pp. 2172–2177.

[12] Ilzhoefer, A., Houska, B., and Diehl, M. Nonlinear MPC of kites under varying wind conditions for a new class of largescale wind power generators. *of Robust and Nonlinear Control* (2007), 1–9.

[13] Lambert, B. *North American Fighter Kites*. Small Potatoes Publishing, Yakima, WA, 2003.

[14] Lansdorp, B., Ruiterkamp, R., and Ockels, W. Towards flight testing of remotely controlled surfkites for wind energy generation. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit* (2007), no. August.

[15] Lansdorp, B., and Williams, P. The Laddermill-Innovative Wind Energy from High Altitudes in Holland and Australia. *Windpower 06 Adelaide, Australia* (2006), 1–14.

[16] Loyd, M. L. Crosswind kite power (for large-scale wind power production). *Journal Of Energy 4*, 3 (1980), 106–111.

[17] Ockels, W. Laddermill, a novel concept to exploit the energy in the airspace. *Aircraft design 4* (2001), 81–97.

[18] Roberts, B. W., Shepard, D. H., Caldeira, K., Cannon, M. E., Eccles, D. G., Grenier, A. J., and Freidin, J. F. Harnessing high-altitude wind power. *IEEE Transactions on Energy Conversion 22*, 1 (2007), 136–144.

[19] Sanchez, G. Dynamics and control of single-line kites. *The Aeronautical Journal* (2006), 615–621.

[20] Sheldahl, R. E., and Klimas, P. C. Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines. Tech. rep., Sandia National Laboratories Energy Report, 1981.

[21] Williams, P., Lansdorp, B., and Ockels, W. Nonlinear Control and Estimation of a Tethered Kite in Changing Wind Conditions. *Journal of Guidance, Control, and Dynamics 31*, 3 (May 2008), 793–799.

[22] Williams, P., Lansdorp, B., and Ockels, W. Optimal Crosswind Towing and Power Generation with Tethered Kites. *Journal of Guidance, Control, and Dynamics 31*, 1 (Jan. 2008), 81–93.

Appendix A

Simulation Code

A.1 Kite Dynamics Code

A.1.1 Main Function Loop with gap for Controller

```
% fighter kite flight simulation

clear;
clf;
clc;
clear all
close all
format compact
global first mk g Fsum rdotdot beta vinf clcd_data
global k_area rho tout ten state psidot_rate

first = 1; %used for animation so the lines are only created once
%!!!!!!!!!!!!!!!!!!!!!!!! Timing Information
tstart = 0;    %start time [sec]
tfinal = 1;  %stop time  [sec]
tstep = .001;  %step size  [sec]
tout = (tstart:tstep:tfinal)';

%!!!!!!!!!!!!!!!!! Kite information
mk = .0160;  % mass of the kite [kg]
chord = 0.41;  % chord length [m]
span = 0.41;  % span of the blade, width, length of leading edge [m]
k_area=span*chord;
psidot_rate =12.5; % set spin rate [rad/s]
beta = -10*pi/180; % tether angle

tcrit_h=5.5; % [N] upper critical tension for changing spin/track
```

```
tcrit_l=4.; % [N] lower critical tension, adds hysteresis range
clcd_data=xlsread('LiftDragCoefficientsRe_of_360000','a2:c118');

rho=1.205; %air density [kg/m^3]
vinf = [4/sqrt(2); 4/sqrt(2); 0];     %wind velocity [m/s]
g=9.81; % gravity [m/s^2]

% generate empty arrays to improve run speed
yout=zeros(length(tout),8);
contlog=zeros(length(tout),1); % continuous array of rotation angle
% approximation, used to calculate slope to approximate spin rate

%!!!!!!!!!!!!!!! Initial Conditions
state=0; % 0 is high tension tracking, low tension spinning is state=2
Y0(1) =60*pi/180; %initial zenith angle       (theta)    [rad]
Y0(2) = 0; % initial zenith angular velocity  (thetadot) [rad/s]
Y0(3) = 45*pi/180; %initial azimuth angle     (phi)      [rad]
Y0(4) = 0; %initial azimuthal angular velociy (phidot)   [rad/s]
Y0(5)=30*pi/180; %initial rotation angle      (psi)      [rad]
Y0(6)=0; % inital rotation angular velocity   (psidot)   [rad/s]
Y0(7)=18; % initial tether length            (r)        [m]
Y0(8)=0; % tether reel speed                 (rdot)     [m/s]

yout(1,:)=Y0;
rdotdot=0;
contlog(1)=Y0(5);
%%%%%% Hueristic CONTROL PARAMETERS %%%%%%%%%%%%%%%%%%%%%%
C=1.5; %proporitonal for reel in acceleration to get to tracking
D=4; %proportional for reel out accel to get to spinning
E=1.5; %proportional for tether length feedback loop
rref=20; %referrence tether length

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stateest=state;

for ind = 1:length(tout)-1

    [yout(ind+1,:)] = rk_4_int(yout(ind,:)', tstep);
    theta=yout(ind+1,1);
    thetadot=yout(ind+1,2);
    phi=yout(ind+1,3);
    phidot=yout(ind+1,4);
    r=yout(ind+1,7);
    rdot=yout(ind+1,8);
    if ind==1 % approximate terms from measureable data
        dtheta=(theta-yout(ind,1))/tstep; % thetadot
```

```
    dphi=(phi-yout(ind,3))/tstep; % phidot
    approx=-atan2(dtheta,dphi); % psi, wrapped from 0 to 360
    cont=approx; % psi, continuous
    dpsi=(cont-yout(ind,5))/tstep; % psidot
else % use second order finite diff. method to approximate terms
    dtheta=(3*theta-4*yout(ind,1)+yout(ind-1,1))/(2*tstep); % thetadot
    dphi=(3*phi-4*yout(ind,3)+yout(ind-1,3))/(2*tstep); % phidot
    approx=-atan2(dtheta,dphi)+2*pi; % psi, wrapped from 0 to 360
    cont=-atan2(dtheta,dphi); % psi, continuous
    while contlog(ind)-cont>300*pi/180
        cont=cont+2*pi;
    end
    dpsi=(3*cont-4*contlog(ind)+contlog(ind-1))/(2*tstep); % psidot
end
contlog(ind+1)=cont;
while yout(ind+1,5)>=2*pi % actual psi, wrap into 0 to 360
    yout(ind+1,5)=yout(ind+1,5)-2*pi;
end
psia=yout(ind+1,5);
if approx>2*pi % wrap psi approx into 0 to 360
    approx=approx-2*pi;
end
F1=Fsum(1); % split the load vector into components
F2=Fsum(2);
F3=Fsum(3);
% calculate tension magnitude
ten=F3*cos(theta) - mk*rdotdot + mk*phidot^2*r + mk*r*thetadot^2 +...
    F1*cos(phi)*sin(theta) + F2*sin(phi)*sin(theta) -...
    mk*phidot^2*r*cos(theta)^2;
% check magnitde of tension to see if state switches
if ten>tcrit_h
    state=0; %track state
elseif ten<tcrit_l
    state=2; %spin state
end
%%%%%%%%% ANIMATE %%%%%%%%%%%%%
drawkite(theta,phi,r,psia)
%%%%%%%%% ANIMATE %%%%%%%%%%%%%

% estimate if kite is spinning or tracking from approximate spin rate
if abs(dpsi)>1
    stateest=2;
else
    stateest=0;
end
first=0;
```

```
    % if the tether tension drops below zero the model is invalid also zero
    % tension means that we have no control over the kite, so we avoid it
    % if tension does go negative, the simulation exits
    if ten<0
        fprintf('negative tension \n')
        break
    end
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    % INSERT CONTROLLER HERE

    % END OF CONTROLLER
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    if abs(rdotdot)>15
        rdotdot=15*sign(rdotdot);
    end
    if yout(ind+1,1) >=pi/2
        fprintf('CRASH after %f seconds, \n',tout(ind+1))
        break
    end
end
theta = yout(:,1);
thetadot = yout(:,2);
phi=yout(:,3);
phidot=yout(:,4);
psia=yout(:,5);
psidot=yout(:,6);
rlog=yout(:,7);
rdlog=yout(:,8);
ends=ind+1;
if ten<0
    ends=ind;
end
posi=zeros(ends,3);
for ii=1:ends % calculate position in inertial frame
    T=[cos(theta(ii))*cos(phi(ii)) cos(theta(ii))*sin(phi(ii)) -sin(theta(ii))
        -sin(phi(ii)) cos(phi(ii)) 0
        sin(theta(ii))*cos(phi(ii)) sin(theta(ii))*sin(phi(ii)) cos(theta(ii))];
    posi(ii,:)=(T\[0 0 rlog(ii)]')';
end
% plot kite path
figure
hold on
plot3(posi(1,1),posi(1,2),posi(1,3),'kv')
hold on
plot3(posi(:,1),posi(:,2),posi(:,3))
hold on
```

```
plot3(posi(end,1),posi(end,2),posi(end,3),'rx')
legend('Start','Path','End'),grid on
axis equal,axis ([0 15 0 15 0 15]),title('Kite Path')
xlabel('X (m)'),ylabel('Y (m)'),zlabel('Z (m)')
E=zeros(ind,1);
% Calculate energy of the system for each time step (energy will only be
% conserved if the aerodynamic forces are removed, as they can add and
% remove energy)
for ee=1:ind
    w=[thetadot(ee)*sin(psia(ee))-phidot(ee)*cos(psia(ee))*sin(theta(ee))
        thetadot(ee)*cos(psia(ee))+phidot(ee)*sin(psia(ee))*sin(theta(ee))
        psidot(ee)+phidot(ee)*cos(theta(ee))];
    R=[0 0 rlog(ee)]';
    dR=[0 0 rdlog(ee)]';
    vkite=dR+cross(w,R);
    vel=sqrt(vkite(1)^2+vkite(2)^2+vkite(3)^2);
    E(ee)=.5*vel^2*mk + mk*g*r*cos(theta(ee));
end
figure
plot(tout(1:ind),E),title('System Energy')
xlabel('Time (s)'),ylabel('Energy (J)')
```

A.1.2 Runge-Kutta Integration

```
function [y_out] = rk_4_int(y, step)
%fourth order Runge-Kutta integration
[k1] = kite_derivatives(y);

ytemp = y + k1*(step/2);
[k2] = kite_derivatives(ytemp);

ytemp = y + k2*(step/2);
[k3] = kite_derivatives(ytemp);

ytemp = y + k3*step;
[k4] = kite_derivatives(ytemp);

y = y + step/6*(k1 + 2*k2 + 2*k3+ k4);

y_out = y;
```

A.1.3 Calculate State Vector Derivative

```
function [dy] = kite_derivatives(Y)
```

```
% calculates the derivatives of kite's equations of motion
global mk state beta rdotdot vinf g first Fsum
global k_area rho mg psidot_rate gamma
theta = Y(1);     % [rad]
thetadot = Y(2); % [rad/s]
phi = Y(3);       % [rad]
phidot = Y(4);    % [rad/s]
psia=Y(5);        % [rad]
r=Y(7);           % [m]
rdot=Y(8);        % [m/s]
% transformation matrix, to go from inertial to local coordinates
T=[cos(psia)*cos(theta)*cos(phi)-sin(psia)*sin(phi)...
    cos(psia)*cos(theta)*sin(phi)+sin(psia)*cos(phi) -cos(psia)*sin(theta)
    -sin(psia)*cos(theta)*cos(phi)-cos(psia)*sin(phi)...
    -sin(psia)*cos(theta)*sin(phi)+cos(psia)*cos(phi) sin(psia)*sin(theta)
    sin(theta)*cos(phi) sin(theta)*sin(phi) cos(theta)];
% kf, facing direction in local system
kf=[0
    cos(beta)
    sin(beta)];
vk=[ r*(thetadot*cos(psia) + phidot*sin(psia)*sin(theta))
    -r*(thetadot*sin(psia) - phidot*cos(psia)*sin(theta))
    rdot]; % kite velocity [m/s]
va = T*vinf - vk; % apparent wind velocity
mag_va=(va(1)^2+va(2)^2+va(3)^2)^.5; % apparent wind magnitude
vhat=va/mag_va; % appartent wind direction
% determine initial spar direction
ksval=[kf(2)*vhat(3)-kf(3)*vhat(2)
    kf(3)*vhat(1)-kf(1)*vhat(3)
    kf(1)*vhat(2)-kf(2)*vhat(1)];
alpa=acos(-vhat(2)*cos(beta)-vhat(3)*sin(beta)); % angle of attack
if first==1
    if ksval(1)<0 % prevents kite from starting upside down
        ksval=-ksval;
    end
end
magks=sqrt(ksval(1)^2+ksval(2)^2+ksval(3)^2);
ks=ksval/magks; % determine unit vector for spar direction
gamma=acos(ks(1)); % calculate roll angle
%bring gamma into -180 to 180 range
if gamma>pi
    gamma=gamma-2*pi;
end
if va(1)<0 %if the wind is from the left roll is counter clockwise
    gamma=-gamma;
end
```

```
% limit roll angle magnitude to 40 degrees
if abs(gamma)>40*pi/180
    gamma=40*pi/180*sign(gamma);
end
ks=[cos(gamma)    sin(gamma)*sin(beta)   -sin(gamma)*cos(beta)];
kn=[sin(gamma)    -cos(gamma)*sin(beta)  cos(gamma)*cos(beta)];
% determine if alpha is negative, if the apparent wind is from the above
% the kite angle of attack is negative. if dot product of apparent wind and
% kite normal direction (kn) is negative, angle of attack is negative
a=kn(1)*va(1)+kn(2)*va(2)+kn(3)*va(3);
if a<0
    alpa=-alpa;
end
if alpa==2*pi; % wrap alpa into 0 to 360 range
    alpa=0;
elseif alpa<0
    alpa=alpa+2*pi;
end
direct_l=[va(2)*ks(3)-va(3)*ks(2)
    va(3)*ks(1)-va(1)*ks(3)
    va(1)*ks(2)-va(2)*ks(1)]; % lift force direction
mag_direct_l=(direct_l(1)^2+direct_l(2)^2+direct_l(3)^2)^.5; % unit vector
[cL,cD] = airfoil_data(alpa*180/pi); % look up lift and drag coefficients
Drag=0.5*rho*k_area*mag_va^2*cD*vhat; % drag force in local frame
Lift=0.5*rho*k_area*mag_va^2*cL*direct_l/mag_direct_l; % lift in local frame
mg=[0 0 -g*mk]'; % kite weight in interial frame
Fsum=T'*(Lift+Drag)+mg;
F1=Fsum(1);
F2=Fsum(2);
F3=Fsum(3);
% build derivative of state vector
dy=zeros(4,1);
dy(1)=thetadot;
dy(2)= (F1*cos(phi)*cos(theta) - F3*sin(theta) + F2*cos(theta)*sin(phi))...
    /(mk*r) - (- r*cos(theta)*sin(theta)*phidot^2 + 2*rdot*thetadot)/r;
dy(3)=phidot;
dy(4)= - (2*phidot*rdot)/r - (F1*sin(phi) - F2*cos(phi) + ...
    2*mk*phidot*r*thetadot*cos(theta))/(mk*r*sin(theta));
% set the spin rate
if state<1
    dy(5)=0;
else
    dy(5)=psidot_rate;
end
dy(6)=0;
dy(7)=rdot;
```

```
dy(8)=rdotdot;
```

A.1.4 Lift and Drag Coefficient Interpolation

```
function [cl_i,cd_i] = airfoil_data(a)
% this function interpolates the lift and drag coefficient data for
% angles of attack between the given data points.
global clcd_data
cl_i=interp1q(clcd_data(:,1),clcd_data(:,2),a);
cd_i=interp1q(clcd_data(:,1),clcd_data(:,3),a);
```

A.1.5 Animation of System Position and Orientation

```
function [  ] = drawkite(theta,phi,r,psia)
% draws a pair of subplots, one showing the full kite and tether, the other
% showing kite's orientation
global first gamma beta
T=[cos(psia)*cos(theta)*cos(phi)-sin(psia)*sin(phi)...
    cos(psia)*cos(theta)*sin(phi)+sin(psia)*cos(phi) -cos(psia)*sin(theta)
    -sin(psia)*cos(theta)*cos(phi)-cos(psia)*sin(phi) ...
    -sin(psia)*cos(theta)*sin(phi)+cos(psia)*cos(phi) sin(psia)*sin(theta)
    sin(theta)*cos(phi) sin(theta)*sin(phi) cos(theta)];
K=[cos(gamma)    sin(gamma)*sin(beta)   -sin(gamma)*cos(beta)
    0             cos(beta)              sin(beta)
    sin(gamma)    -cos(gamma)*sin(beta)  cos(gamma)*cos(beta)];

drawing=[0 0 r]'; % position vector in the local frame
ksxyz=T'*K'*[1 0 0]';
kfxyz=T'*K'*[0 1 0]';
knxyz=T'*K'*[0 0 1]';
er=T'*[0 0 1]';

% create handles for each line of the figure so they can be called and
% adjusted at each iteration, rather than recreated
persistent tetherline
persistent kiteline
persistent facing
persistent normal
persistent spar
persistent outline1
persistent outline2
persistent outline3
persistent outline4
persistent tether
```

```
figure(1)
pos=T'*drawing; %postion vector in the inertial frame

%draw the kite string and the lines to represent the kite
if (first == 1)
    first = 0;
    subplot(1,2,1)
    tetherline = line('xdata', [0, pos(1)], 'ydata', [0, pos(2)],...
        'zdata',[0,pos(3)],'color','k','linewidth', 3);
    kiteline = line('xdata',[pos(1),pos(1)+2*kfxyz(1)],'ydata',...
        [pos(2),pos(2)+2*kfxyz(2)],'zdata',[pos(3),pos(3)+2*kfxyz(3)],...
        'color','r','linewidth',2);

    grid on, axis equal, axis([0 15 0 15 0 15]);
    xlabel('X'),ylabel('Y'),zlabel('Z'),title('Kite Motion')

    subplot(1,2,2)
    facing = line('xdata',[0,kfxyz(1)],'ydata',[0,kfxyz(2)],'zdata',...
        [0,kfxyz(3)],'color','r','linewidth',3);
    normal = line('xdata',[0,knxyz(1)],'ydata',[0,knxyz(2)],'zdata',...
        [0,knxyz(3)],'color','b','linewidth',3);
    spar = line('xdata',[0,ksxyz(1)],'ydata',[0,ksxyz(2)],'zdata',...
        [0,ksxyz(3)],'color','g','linewidth',3);
    outline1 = line('xdata',[ksxyz(1),kfxyz(1)],'ydata',[ksxyz(2),...
        kfxyz(2)],'zdata',[ksxyz(3),kfxyz(3)],'color','k','linewidth',1);
    outline2 = line('xdata',[kfxyz(1),-ksxyz(1)],'ydata',[kfxyz(2),...
        -ksxyz(2)],'zdata',[kfxyz(3),-ksxyz(3)],'color','k','linewidth',1);
    outline3 = line('xdata',[-ksxyz(1),-kfxyz(1)],'ydata',[-ksxyz(2),...
        -kfxyz(2)],'zdata',[-ksxyz(3),-kfxyz(3)],'color','k','linewidth',1);
    outline4 = line('xdata',[-kfxyz(1),ksxyz(1)],'ydata',[-kfxyz(2),...
        ksxyz(2)],'zdata',[-kfxyz(3),ksxyz(3)],'color','k','linewidth',1);
    tether=line('xdata',[-er(1),0],'ydata',[-er(2),0],'zdata',...
        [-er(3),0],'color','k','linewidth',2);

    axis equal,axis([-1 1 -1 1 -1 1]),grid on
    xlabel('X'),ylabel('Y'),zlabel('Z'),title('Kite Orientation')
else
    % use the handles to reset the position of each line
    % kite and tether plot
    set(tetherline,'xdata',[0, pos(1)], 'ydata', [0, pos(2)],...
        'zdata', [0,pos(3)]);
    set(kiteline,'xdata',[pos(1),pos(1)+2*kfxyz(1)],'ydata',...
        [pos(2),pos(2)+2*kfxyz(2)],'zdata',[pos(3),pos(3)+2*kfxyz(3)]);
    %orientation plot
    set(facing,'xdata',[0,kfxyz(1)],'ydata',[0,kfxyz(2)],...
```

```
        'zdata',[0,kfxyz(3)]);
    set(normal,'xdata',[0,knxyz(1)],'ydata',[0,knxyz(2)],...
        'zdata',[0,knxyz(3)]);
    set(spar,'xdata',[0,ksxyz(1)],'ydata',[0,ksxyz(2)],...
        'zdata',[0,ksxyz(3)]);
    set(outline1,'xdata',[ksxyz(1),kfxyz(1)],...
        'ydata',[ksxyz(2),kfxyz(2)],'zdata',[ksxyz(3),kfxyz(3)]);
    set(outline2,'xdata',[kfxyz(1),-ksxyz(1)],...
        'ydata',[kfxyz(2),-ksxyz(2)],'zdata',[kfxyz(3),-ksxyz(3)]);
    set(outline3,'xdata',[-ksxyz(1),-kfxyz(1)],...
        'ydata',[-ksxyz(2),-kfxyz(2)],'zdata',[-ksxyz(3),-kfxyz(3)]);
    set(outline4,'xdata',[-kfxyz(1),ksxyz(1)],...
        'ydata',[-kfxyz(2),ksxyz(2)],'zdata',[-kfxyz(3),ksxyz(3)]);
    set(tether,'xdata',[-er(1),0],'ydata',[-er(2),0],'zdata',[-er(3),0]);
end
drawnow
end
```

A.2 Kite Control Algorithms

A.2.1 Full Knowledge Controller

```
% full knowledge controller, knows exact information of kite, entire state
% vector is available
if theta>55*pi/180
    if state==0
        if psia>179*pi/180
            % kite is tracking down, rapidly release line to switch to spin
            rdotdot=(ten)*D;
        else
            % kite is tracking upward, release or recover line to maintain
            % desired length
            rdotdot=(rref-r)*(ten-tcrit_l)*E;
        end
    else
        if psia<150*pi/180
            % kite is spinning, facing up, rapidly pull in linte to track
            rdotdot=-(ten+tcrit_h)*C;
        end
    end
    %the above if statements keep the kite from crashing, this next
    %set tries to keep the tether length in a desirable range
else
    if r>rref
        AA=1;
```

```
    else
        AA=ten-2;
        if ten<2
            AA=0;
        end
    end
    rdotdot=AA*(rref-r);
end
```

A.2.2 Reduced Knowledge Controller: $\dot{\theta}$

```
% reduced knowledge simple controller, motor speed set according to tether
% tension and the sign of the zenith angular velocity
if theta>65*pi/180
    if dtheta>0
        % if zenith angle is increasing, the kite is falling, release
        % tether to make kite spin
        rdotdot=(ten-2)*D;
    elseif dtheta<0
        rdotdot=(ten-8)*C;
        % if zenith angle is decreasing, the kite is rising, recover tether
        % to make the kite track
    end
else
    % kite is safe, maintain desired tether length range
    rdotdot=(rref-r)*ten*E;
end
```

A.2.3 Reduced Knowledge Controller: $\dot{\theta}$ and Spin/Tracking Approximation

```
% reduced knowledge controller, motor speed set according to
% tether tesnion, sign of zenith angular velocity, and an approximation of
% the kite mode (spinning or tracking)
if theta>65*pi/180
    if dtheta>0
        % kite is falling
        if stateest==2
            % kite is spinning, so increase line tension to up edge of
            % spinning zone
            rdotdot=(ten-tcrit_h)*E;
        else
            rdotdot=(ten)*D;
            % kite tracking downward, release tension quickly to make the
```

```
            % kite spin
        end
    elseif dtheta<0
        % kite is moving up
        if stateest==2
            % kite is in the upward portion of a spin, increase tension to
            % make the kite track
            rdotdot=(ten-10)*C;
        else
            rdotdot=(rref-r)*(ten-tcrit_l)*E;
            % kite is tracking, release or recover tether to maintain
            % desired length and stay in tracking mode
        end
    end
else
    % kite is safe, maintain desired tether length
    rdotdot=(rref-r)*ten*E;
end
```

A.2.4 Reduced Knowledge Controller: ψ and Spin/Tracking Approximation

```
% reduced knowledge controller, motor speed set according to tether
% tension, mode approximation, and rotation angle approximation

% Identical to full knowledge controller, with known values replaced by
% approximations
if theta>55*pi/180
    if  approx<150*pi/180 && stateest==2
        %if below 55 degrees, spinning, and pointed up, try to switch to
        %tracking state
        rdotdot=-(ten+tcrit_h)*C;
    elseif stateest==0 && approx>179*pi/180
        %let line out to switch to spin
        rdotdot=(ten)*D;
    elseif stateest==0
        % release or recover line to maintain desired length
        rdotdot=(rref-r)*ten*E;
    end
    %the above if statements take care of getting it
    %spinning/tracking to avoid crashing, this next set tries to keep
    %the tether length from running away, or getting all used up
else
    if r>rref
        AA=1;
```

```
    else
        AA=ten-2;
        if ten<2
            AA=0;
        end
    end
    rdotdot=AA*(rref-r);
end
```

Appendix B

Aerodynamics

B.1 Lift and Drag Coefficient Table

Table B.1: 2-D and 3-D lift and drag coefficients, 2-D coefficients taken from [20].

α	C_l	C_d	C_L	C_D	α	C_l	C_d	C_L	C_D	α	C_l	C_d	C_L	C_D
0	0.0000	0.0091	0.1100	0.0092	85	0.2300	1.8000	0.0900	1.8000	280	-0.3650	1.7800	-0.5000	1.7350
1	0.1100	0.0092	0.2200	0.0094	90	0.0900	1.8000	-0.0500	1.7800	285	-0.5000	1.7350	-0.6300	1.6650
2	0.2200	0.0094	0.3300	0.0098	95	-0.0500	1.7800	-0.1850	1.7500	290	-0.6300	1.6650	-0.7600	1.5750
3	0.3300	0.0098	0.4400	0.0105	100	-0.1850	1.7500	-0.3200	1.7000	295	-0.7600	1.5750	-0.8750	1.4700
4	0.4400	0.0105	0.5500	0.0114	105	-0.3200	1.7000	-0.4500	1.6350	300	-0.8750	1.4700	-0.9550	1.3450
5	0.5500	0.0114	0.6600	0.0126	110	-0.4500	1.6350	-0.5750	1.5550	305	-0.9550	1.3450	-1.0200	1.2150
6	0.6600	0.0126	0.7390	0.0143	115	-0.5750	1.5550	-0.6700	1.4650	310	-1.0200	1.2150	-1.0500	1.0750
7	0.7390	0.0143	0.8240	0.0157	120	-0.6700	1.4650	-0.7600	1.3500	315	-1.0500	1.0750	-1.0350	0.9200
8	0.8240	0.0157	0.8946	0.0173	125	-0.7600	1.3500	-0.8500	1.2250	320	-1.0350	0.9200	-0.9800	0.7450
9	0.8946	0.0173	0.9440	0.0191	130	-0.8500	1.2250	-0.9300	1.0850	325	-0.9800	0.7450	-0.8550	0.5700
10	0.9440	0.0191	0.9572	0.0211	135	-0.9300	1.0850	-0.9800	0.9250	330	-0.8550	0.5700	-0.8788	0.4600
11	0.9572	0.0211	0.9285	0.0233	140	-0.9800	0.9250	-0.9000	0.7550	333	-0.8788	0.4600	-0.8055	0.4320
12	0.9285	0.0233	0.8562	0.0257	145	-0.9000	0.7550	-0.7700	0.5750	334	-0.8055	0.4320	-0.7511	0.4050
13	0.8562	0.0257	0.7483	0.0283	150	-0.7700	0.5750	-0.6700	0.4200	335	-0.7511	0.4050	-0.7015	0.3790
14	0.7483	0.0283	0.6350	0.0312	155	-0.6700	0.4200	-0.6350	0.3200	336	-0.7015	0.3790	-0.6528	0.3540
15	0.6350	0.0312	0.5384	0.1240	160	-0.6350	0.3200	-0.6800	0.2300	337	-0.6528	0.3540	-0.6045	0.3290
16	0.5384	0.1240	0.4851	0.2170	165	-0.6800	0.2300	-0.8500	0.1400	338	-0.6045	0.3290	-0.5616	0.3050
17	0.4851	0.2170	0.4782	0.2380	170	-0.8500	0.1400	-0.6600	0.0550	339	-0.5616	0.3050	-0.5247	0.2820
18	0.4782	0.2380	0.4908	0.2600	175	-0.6600	0.0550	0.0000	0.0250	340	-0.5247	0.2820	-0.4908	0.2600
19	0.4908	0.2600	0.5247	0.2820	180	0.0000	0.0250	0.6600	0.0550	341	-0.4908	0.2600	-0.4782	0.2380
20	0.5247	0.2820	0.5616	0.3050	185	0.6600	0.0550	0.8500	0.1400	342	-0.4782	0.2380	-0.4851	0.2170
21	0.5616	0.3050	0.6045	0.3290	190	0.8500	0.1400	0.6800	0.2300	343	-0.4851	0.2170	-0.5384	0.1240
22	0.6045	0.3290	0.6528	0.3540	195	0.6800	0.2300	0.6350	0.3200	344	-0.5384	0.1240	-0.6350	0.0312
23	0.6528	0.3540	0.7015	0.3790	200	0.6350	0.3200	0.6700	0.4200	345	-0.6350	0.0312	-0.7483	0.0283
24	0.7015	0.3790	0.7511	0.4050	205	0.6700	0.4200	0.7700	0.5750	346	-0.7483	0.0283	-0.8562	0.0257
25	0.7511	0.4050	0.8055	0.4320	210	0.7700	0.5750	0.9000	0.7550	347	-0.8562	0.0257	-0.9285	0.0233
26	0.8055	0.4320	0.8788	0.4600	215	0.9000	0.7550	0.9800	0.9250	348	-0.9285	0.0233	-0.9572	0.0211
27	0.8788	0.4600	0.8550	0.5700	220	0.9800	0.9250	0.9300	1.0850	349	-0.9572	0.0211	-0.9440	0.0191
30	0.8550	0.5700	0.9800	0.7450	225	0.9300	1.0850	0.8500	1.2250	350	-0.9440	0.0191	-0.8946	0.0173
35	0.9800	0.7450	1.0350	0.9200	230	0.8500	1.2250	0.7600	1.3500	351	-0.8946	0.0173	-0.8240	0.0157
40	1.0350	0.9200	1.0500	1.0750	235	0.7600	1.3500	0.6700	1.4650	352	-0.8240	0.0157	-0.7390	0.0143
45	1.0500	1.0750	1.0200	1.2150	240	0.6700	1.4650	0.5750	1.5550	353	-0.7390	0.0143	-0.6600	0.0126
50	1.0200	1.2150	0.9550	1.3450	245	0.5750	1.5550	0.4500	1.6350	354	-0.6600	0.0126	-0.5500	0.0114
55	0.9550	1.3450	0.8750	1.4700	250	0.4500	1.6350	0.3200	1.7000	355	-0.5500	0.0114	-0.4400	0.0105
60	0.8750	1.4700	0.7600	1.5750	255	0.3200	1.7000	0.1850	1.7500	356	-0.4400	0.0105	-0.3300	0.0098
65	0.7600	1.5750	0.6300	1.6650	260	0.1850	1.7500	0.0500	1.7800	357	-0.3300	0.0098	-0.2200	0.0094
70	0.6300	1.6650	0.5000	1.7350	265	0.0500	1.7800	-0.0900	1.8000	358	-0.2200	0.0094	-0.1100	0.0092
75	0.5000	1.7350	0.3650	1.7800	270	-0.0900	1.8000	-0.2300	1.8000	359	-0.1100	0.0092	0.0000	0.0091
80	0.3650	1.7800	0.2300	1.8000	275	-0.2300	1.8000	-0.3650	1.7800	360	0.0000	0.0091	0.0000	0.0000

B.2 Conversion of 2-D Lift and Drag Coefficients to 3-D

Sheldhal experimentally measured lift and drag coefficients for angles of attack from zero to 360 degrees for several Reynolds numbers. The tests used a uniform wing that crossed the entire wind tunnel, generating values for a 2-D wing. The values must be converted to 3-D to account for the edge effects of a finite wing. The coefficients are symbolically changed from C_l and C_d, to C_L and C_D to reflect this conversion. To make the model behave more smoothly, all lift and drag coefficients used correspond to a Reynolds number of 360,000. Initial simulation showed that this was the primary operating range of the kite.

For an infinite wing, air can only flow over or under the wing. For a finite wing, air also flows from under the wing, around the tips toward the low pressure area above the airfoil. This effect reduces lift while increasing drag. To account for this, the lift and drag coefficients are adjusted by using the lift slope method described by Anderson [1].

The initial slope of the 2-D data, a_0 in Eqn. B.1, and airfoil parameters are used to calculate a corrected slope, a in Eqn. B.2. The corrected lift coefficient, Eqn. B.3, is defined by a line of this slope with an offset (zero in this case, as none is needed for a symmetric airfoil) is extended until it intersects the original 2-D data. Beyond this point the airflow has separated and there is little difference between 2-D and 3-D lift coefficients.

$$a_0 = \frac{dC_l}{d\alpha} \tag{B.1}$$

$$a = \frac{a_0}{1 + \frac{a_0}{\pi A R e_1}} \tag{B.2}$$

$$C_L = a\alpha \tag{B.3}$$

The flow around the wing tips generates induced drag, which is added to the 2-D drag coefficient to convert to 3-D. Induced drag is related to the lift coefficient as seen in

Eqn. B.4.

$$C_D = C_d + \frac{C_L^2}{AR\pi e} \tag{B.4}$$

The parameters e and e_1 are efficiency factors that typically range from 0.85 to 0.95, and are assumed to both be 0.9 [1]. AR is the aspect ratio, which for the square kite used in this study is 1. Figures B.1 and B.2 show the 2-D and converted 3-D lift and drag coefficients for all possible angles of attack.

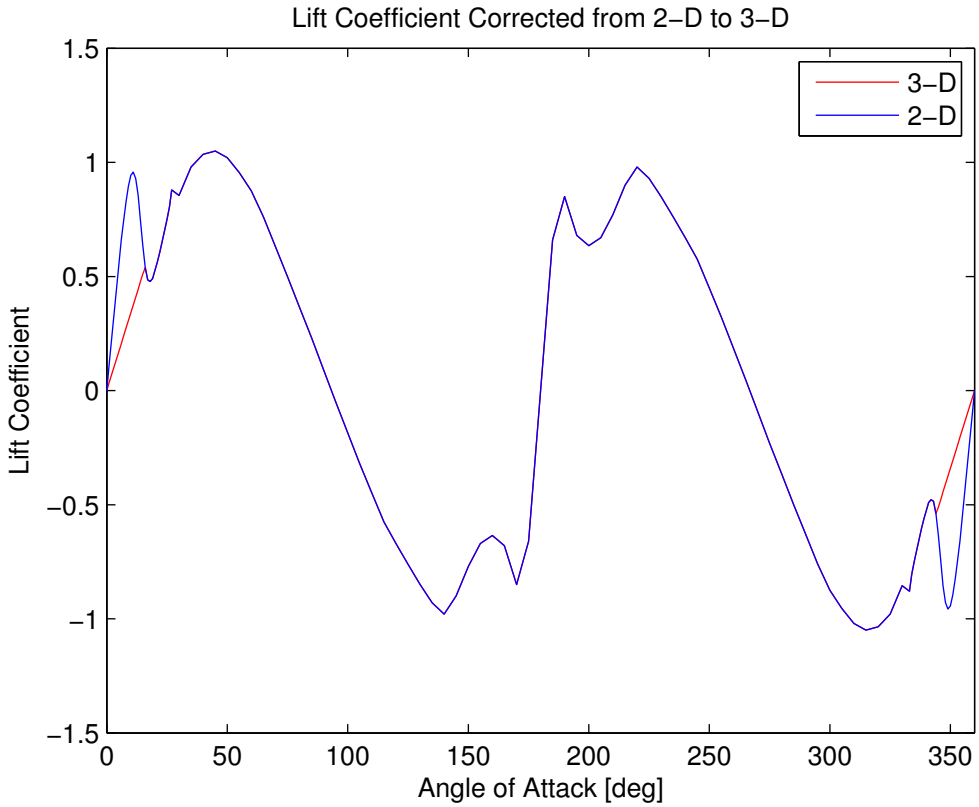


Fig. B.1: Plot of 2-D and 3-D lift coefficients. 2-D coefficients were taken from Sheldhal and Klimas [20] and converted to 3-D following the method described by Anderson [1].

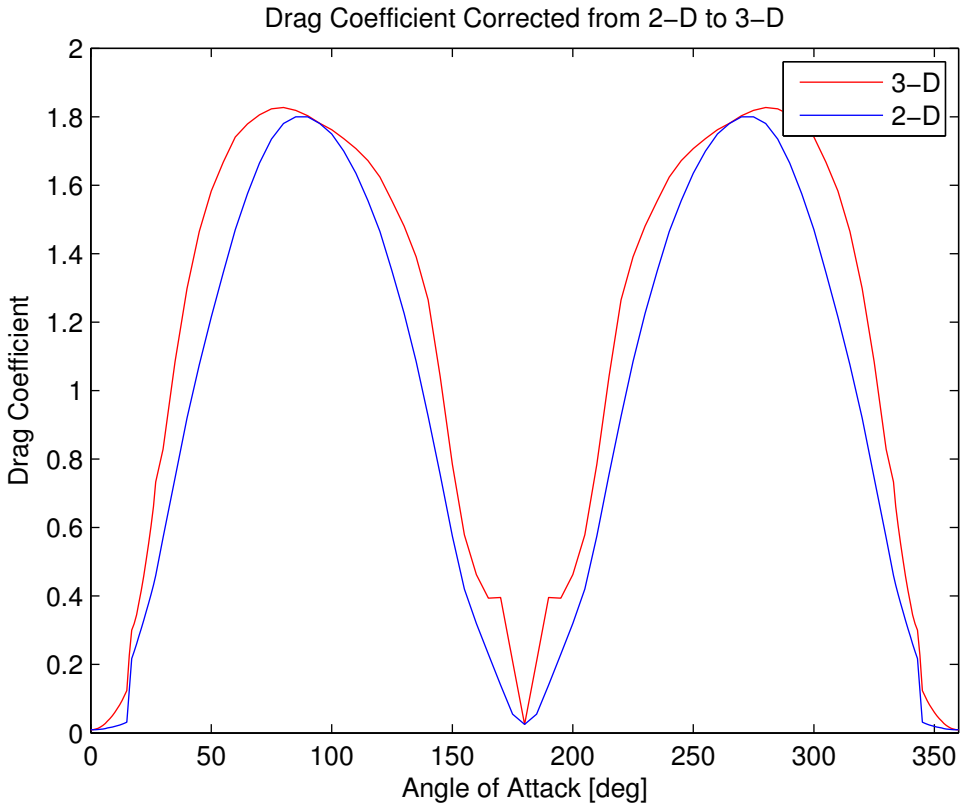


Fig. B.2: Plot of 2-D and 3-D drag coefficients. 2-D coefficients were taken from Sheldhal and Klimas [20] and converted to 3-D following the method described by Anderson [1].

Appendix C

Calibration

C.1 Motor Input

The motor controller receives a serial pulse train from the control algorithm to set the new desired speed. The motor controller recognizes values from zero to 127, for stop to full forward, or reverse if negative. To determine the actual speed of the motor for control purposes, rotation speed was measured using a variable frequency strobe light. A reflective washer was attached to the wheel and the strobe frequency was adjusted until the washer appeared to remained stationary from one strobe to the next. Motor input values from -45 to 45 were measured and are shown in Fig. C.1 along with the linear fit in Eqn. C.1.

$$\text{Speed } [m/s] = .0785 * \text{control input} \tag{C.1}$$

C.2 Load Cells

The load cells are oriented to be mutually orthogonal. Each load cell was calibrated individually. The tether guide was removed and a mass was tied directly to the corresponding connecting rod. The structure was rotated to vertically align the tested axis

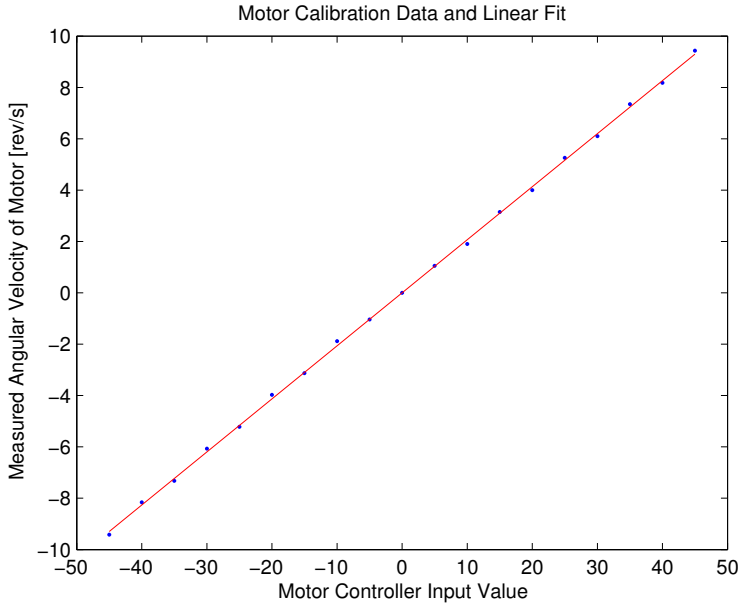


Fig. C.1: Scaling of motor controller input value to generated motor speed.

to produce pure tension loading. The hanging mass ranged from 0.1kg up to 2.8kg. The load cells are rated to 5kg, but for mass over 2.8kg the kite tether used to suspend the masses would break, showing that measurement beyond 2.8kg was unnecessary. The noise averaged sensor value was recorded and a linear fit line produced for each axis. Only the slope of the line is important for experimentation, as the offset will change depending on temperature and any pre-loading from the frame. The measured data and linear fit for each axis are shown in Fig. C.2 and Eqns. C.2-C.4.

$$F_x = 47.342 * sensor + x_{\text{offset}} \tag{C.2}$$

$$F_y = 46.307 * sensor + y_{\text{offset}} \tag{C.3}$$

$$F_z = 47.450 * sensor + z_{\text{offset}} \tag{C.4}$$

As seen in Fig. C.2, each load cell is highly linear and matches the data well across the entire tested range. Off-axis loading arising from bending and torsion is not considered, as the strain gauge cannot distinguish the difference. This will be the

main source of error for tension and position measurements. The design of the tether guide minimizes this by allowing some freedom of movement.

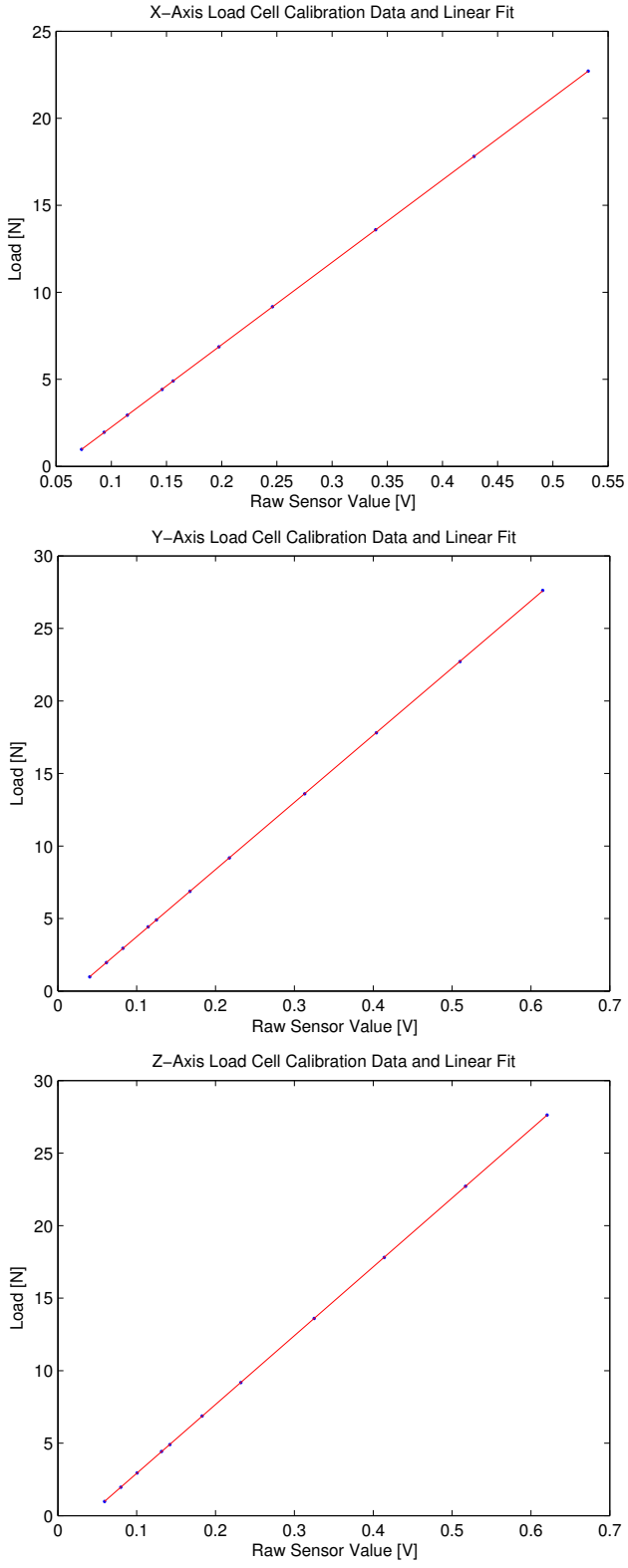


Fig. C.2: Linear fit to the load cell calibration measurements of each axis.

Appendix D

Data Acquisition Code

All automated data acquisition in this study was conducted with National Instruments LabView program. The code gathers data from the load cells and encoder to estimate kite position and orientation and uses that information to determine the appropriate motor speed to maintain flight. A zoomed out image of the full code is given in Fig. D.1. To better document its function, the code was broken down into smaller, functional components for each individual measurement.

D.1 Load Cell VI

The load cell code is given in Fig. D.2. The system uses three orthogonal load cells to measure the tether tension and direction. The program starts by initializing each of the sensors; setting the data rate and gain to be used. The next step is to zero the load cells. The calibration conducted in App. C.2 initially provided linear calibration curves with slope and offset values. The offsets were neglected because the load cells are sensitive to any temperature change or tightening of the overall structure. This could cause changes in the offsets, reducing accuracy. Instead, only the slopes of the lines are used, and a new offset is calculated before the main loop of the program.

The "Zeroing Loop" labeled in Fig. D.2 builds an array of raw sensor values for zero load and averages them. Due to initial communications errors, the sensors occasionally read high (10^6 and above) values. The true/false case structure checks

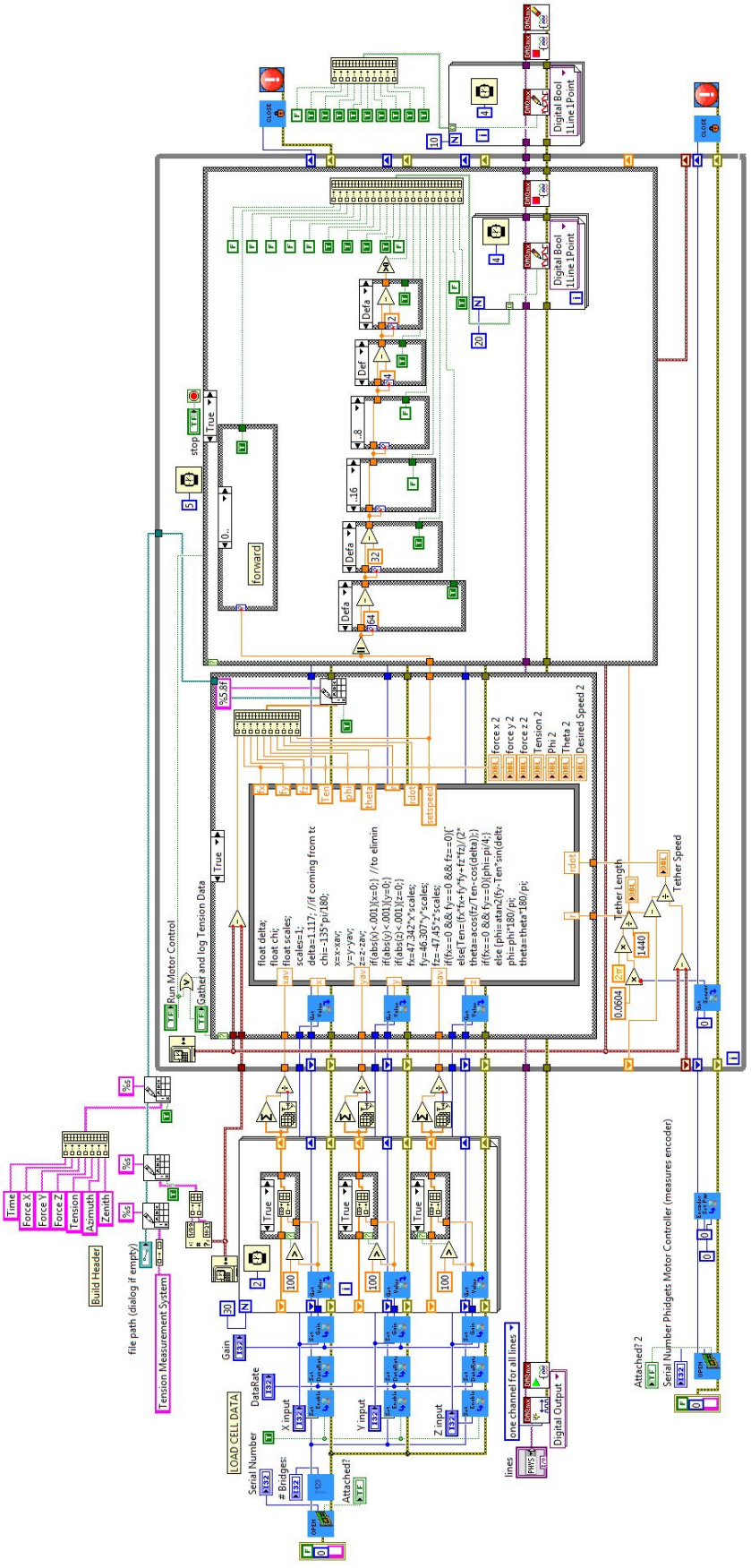


Fig. D.1: View of full LabView code used to measure tether information and apply a desired tether speed.

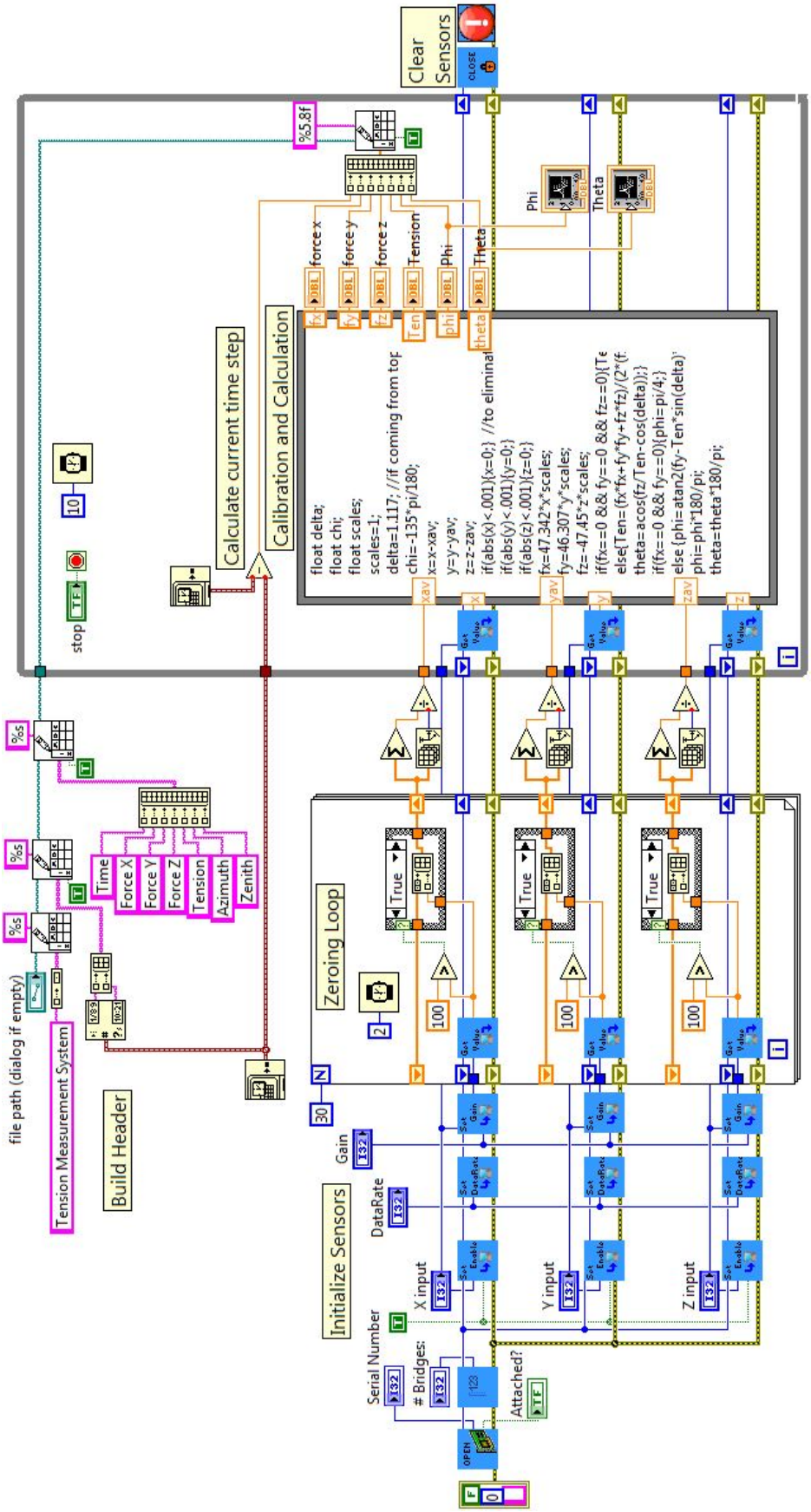


Fig. D.2: Reduced control code showing the load cell value zeroing and measurement loops.

the magnitude before appending to the array to avoid these errors. The values are then averaged and sent into the main while loop where the sensors are polled in each iteration and the offset and calibration slopes are used to calculate the tether tension, zenith angle, and azimuth angle. The values are output to a tab delimited text file. When the user stops the program the while loop exists and the resources are cleared.

D.2 Motor Control VI

The motor code is shown in Fig. D.3. It begins by creating a digital physical channel (an active voltage output that can either be high or low, 5V or 0V) and starting a task. It then enters a while loop that keeps the program running until ended by the user. Within the while loop a series of case structures (boxes with cross-hatched edges) converts the speed into an 8-bit binary array. The pololu motor controller is set to TLL Serial mode to read the value. The full motor signal array consists of 20 bits, each providing different information as shown in Tab. D.1. The motor reads values in least significant bit first format. The motor direction value is 225 for forward or 224 for reverse. Converted to binary and bounded by a low start bit and a high stop bit, this becomes: 0100001111 for forward and 0000001111 for reverse. The motor speed value is similarly converted. The array is fed into a timed for loop that runs twenty times. Each iteration of the for loop draws the corresponding value from the array and sets the output state to either high (true) or low (false). The baud rate of the the motor controller is set to match the timing of the for loop. Every run of the while loop generates a new pulse train. When the user ends the while loop, another for loop sends the stop signal to the motor controller.

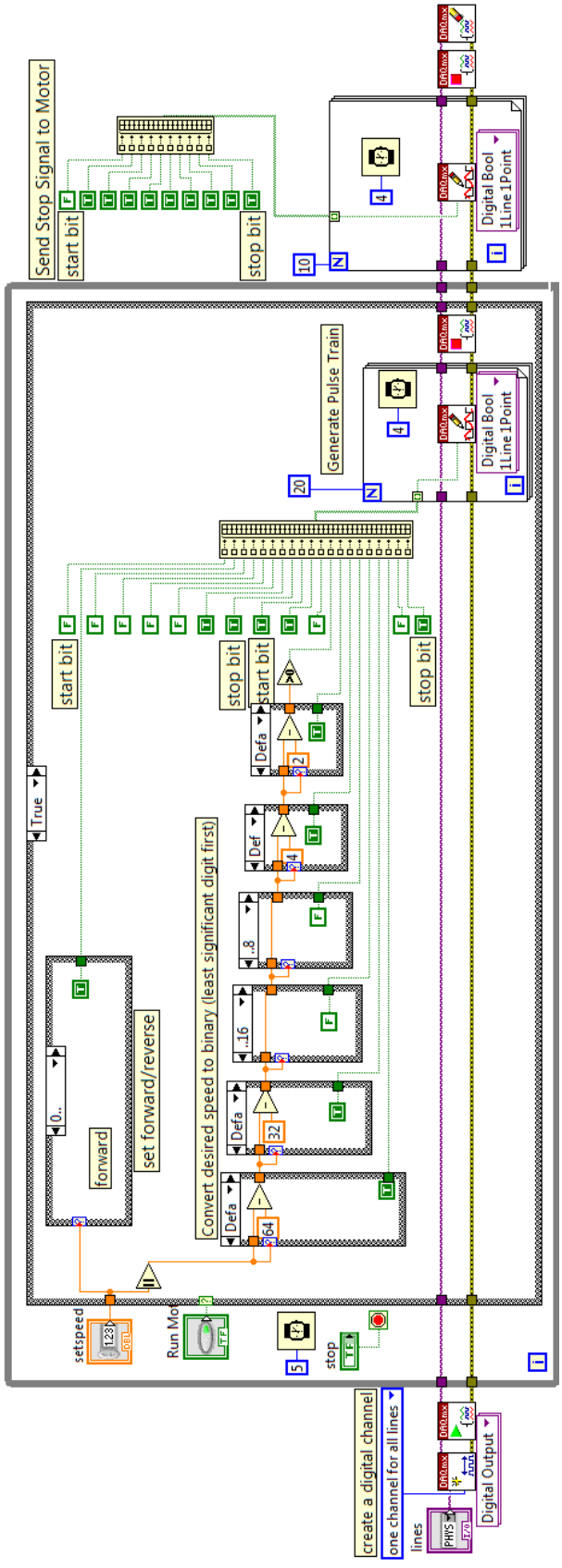


Fig. D.3: Reduced control code showing the communication method between LabView and the PID motor controller.

Table D.1: Definition of Digital Pulse Train Sent to Motor to Control Speed

Bit#	Value
1	START
2	Motor Direction
3	
4	
5	
6	
7	
8	
9	
10	STOP
11	START
12	Motor Speed
13	
14	
15	
16	
17	
18	
19	
20	STOP

D.3 Encoder VI

The encoder is attached to the motor to measure rotation and rotation rate. The encoder measurement code is shown in Fig. D.4. It starts by setting the encoder count to zero. Then it polls the counter to determine how far the motor has rotated. The encoder generates 1440 counts per revolution, or 0.25° per count. Using this value and the radius of the reel (measured to be 0.0604 meters) the tether length can be calculated, as in Eqn. D.1.

$$\text{tether length} = \text{n counts} * \frac{1 \text{ rev}}{1440 \text{ counts}} * \frac{2\pi * \text{radius}}{1 \text{ rev}} \tag{D.1}$$

To find the speed of the tether, simple finite difference is used. Shift registers feed the previous tether length and time stamp into the while loop where they are subtracted from the current value. Equation D.2 is then used to calculate tether speed.

$$\text{tether speed} = \frac{\Delta\text{length}}{\Delta\text{time}} \tag{D.2}$$

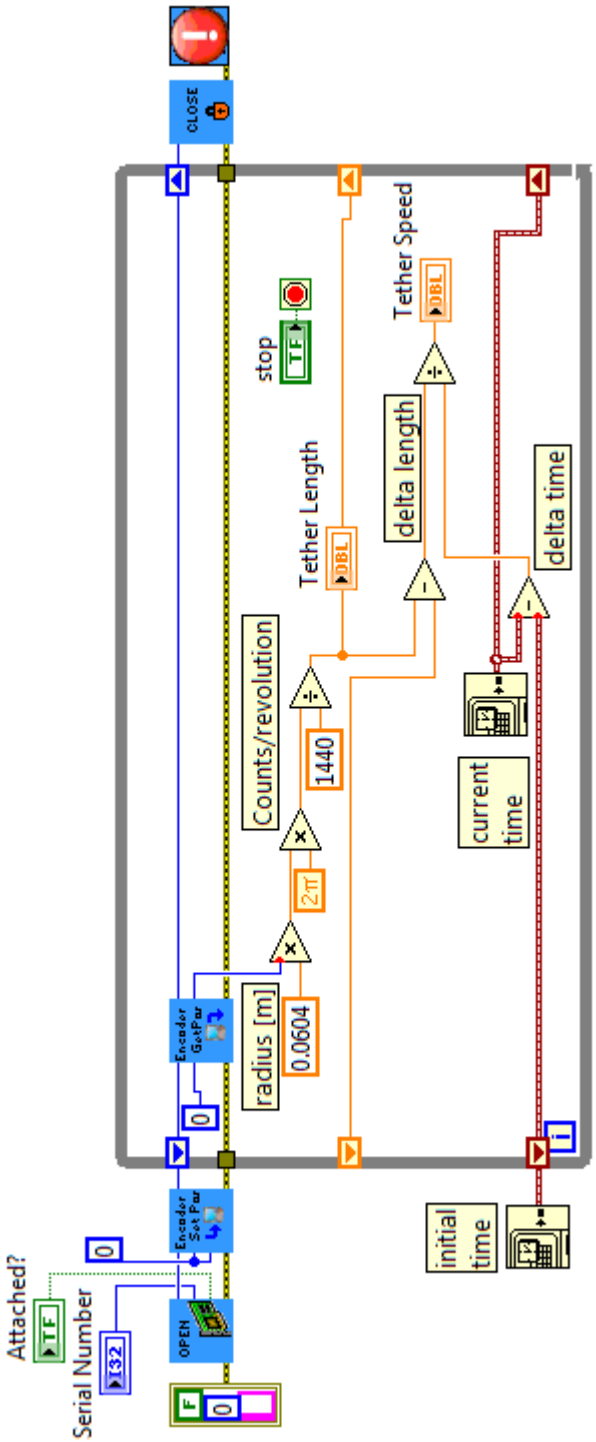


Fig. D.4: Reduced control code showing the measurement and calculation of tether length and speed.

Appendix E

Equipment Specifications

Table E.1: Data Acquisition and Control System Component Specifications

Components	Manufacturer	Part Number	Description
Load Cell	Phidgets	3133_0	0-5 kg capacity shear force load cell
Wheatstone Bridge	Phidgets	1046_0	4-input Wheatstone bridge, 125 sample/s
Motor and Encoder	Phidgets	3269E_1	24V, 4.25:1 gear motor, quadrature encoder, 1440 counts/rev
Motor Controller	Pololu	JRK 21v3	PID controller, receives serial or analog input, variable baud rate
Encoder Counter	Phidgets	1065_0	Motor controller, used for encoder counter
Pulse Train Generator	National Instruments	NI USB-6210	Digital and analog I/O