Department of Mechanical Engineering

DESIGN OF A SCALE-MODEL HUMAN-POWERED HELICOPTER

A Major Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Andrew Camann

Kevin Arruda

David J. Olinger, Advisor

April 28, 2011

Abstract

Over the past several decades, several attempts have been made at designing and constructing human-powered helicopters (HPH). The goal of this MQP is to design an 8.25:1 scale model of an HPH that incorporates the most functional aspects of past designs and addresses their shortcomings. The design is based on the rules as defined by the Igor I. Sikorsky Human-Powered Helicopter Competition. Dynamic similarity was used to determine the essential characteristics of the model, such as weight, rotor angular velocity, and power required. A counter-rotating blade design is used to promote stability and minimize structural weight. Rotor blades are designed from a high-lift Epler series airfoil obtained from a low Reynolds number airfoil database. Using a wind tunnel and a custom-designed and built test stand, original lift and drag data has been collected for a Reynolds number of 10,000 on a test wing. This raw data is used to obtain regression curves for lift and drag, from which ideal wing geometries can be determined. A simple blade element theory code was developed using MATLAB. Using this code various planforms are considered, including a linear and elliptic taper on the trailing edge. Geometric twist is also incorporated into the design of the wing sections to provide the most efficient lift and drag distributions along the span of the rotor. An effort was made to construct the model using materials similar to those that likely would be used to build the prototype. Preliminary tests determined that the power required to produce the required lift is below the threshold set by the dynamic similarity analysis.

Contents

List of Figures

List of Tables

1　Introduction

The goal of this project is to design and construct a properly scaled model of a human powered rotary aircraft. The idea of human powered flight has been around for ages, and over the years many different aircraft designs have been devised and tested. Human powered flight was first achieved in August 1935. The HV-1 Mufli was designed and built by Helmut Haessler and Franz Villinger. The design largely replicated proven glider designs of the era. The flight lasted for over 200 meters, but a catapult was used to assist take off. Over the next 20 years, there was an extreme interest in human powered flight, and the number of successful attempts increased. In November of 1959, the Kremer prize was announced for the amount of 5000 pounds for the first HPA (human powered aircraft) to take off unassisted and fly a figure eight around two points separated by one half mile. The prize was open to any British designed and built HPA that was flown in the British Commonwealth that met the regulations of the competition. The prize would later be doubled to 10,000 pounds and opened to all nationalities [1]. In November of 1961 that the first HPA capable of unassisted take off was constructed. On August 23, 1977 Dr. Paul MacCready and his team from AeroVironment were awarded the first Kremer prize when their aircraft the Gossamer Condor 2 was the first to complete the figure eight course. MacCready and his team then went on to win the second Kremer prize for designing and building the Gossamer Albatross which was the first HPA to successfully cross the English Channel [2].

The Sikorsky prize in the amount of 20,000 dollars was announced in 1980 and will be awarded to the first HPH (human powered helicopter) to achieve flight for 60 seconds,while reaching a height of 3 meters, and remaining within a 10 me-

ter square during the duration of the flight. When asked about the possibility of designing and constructing a HPH MacCready replied "The reality of a human-powered helicopter is that, with the simple formula for static thrust of a rotor, you quickly find you need a disk diameter of 150 feet or so for serious human-powered helicopter flight. It can be done, but the task is huge, and the dollar prize not worth the time expenditure. There are many more exciting, never-been-done-before challenges that can be accomplished with much less work." Since MacCreadys statement there have been many attempts at designing and constructing HPHs. The two most notable attempts thus far were the Da Vinci III developed by Dr. William Patterson and his team from the California Polytechnic Institute, and the Yuri 1 developed by Dr. Akira Naito and students at the Hihon University [3].

The Da Vinci III pictured below in Figure 1 was the first HPH to achieve flight in December 1989 [4]. The flight lasted for duration of 8 seconds and reached a height of 0.2 meters. The design was powered by rotors located at the wing tips, and incorporated a flywheel to smooth out the power delivered by the pilot. Because of the instabilities associated with the design two assistants were required to stabilize the center of the craft during the flight. It was ruled that the assistants were allowed to help so long as they did not contribute any upward force to the craft [3].

Figure 1: The Da Vinci III human powered helicopter.

The Yuri I pictured below in Figure 2 is the most successful HPH ever constructed. In 1994, it achieved flight for 19.46 seconds and an altitude of 0.2 m. Later, it unofficially achieved flight for 24 seconds and an altitude of 0.7 m. The Yuri I had a truss system that spanned 20 meters with 4 rotors each consisting of two 5-meter blades. There has not been a sucesssfull HPH flight since 1994.
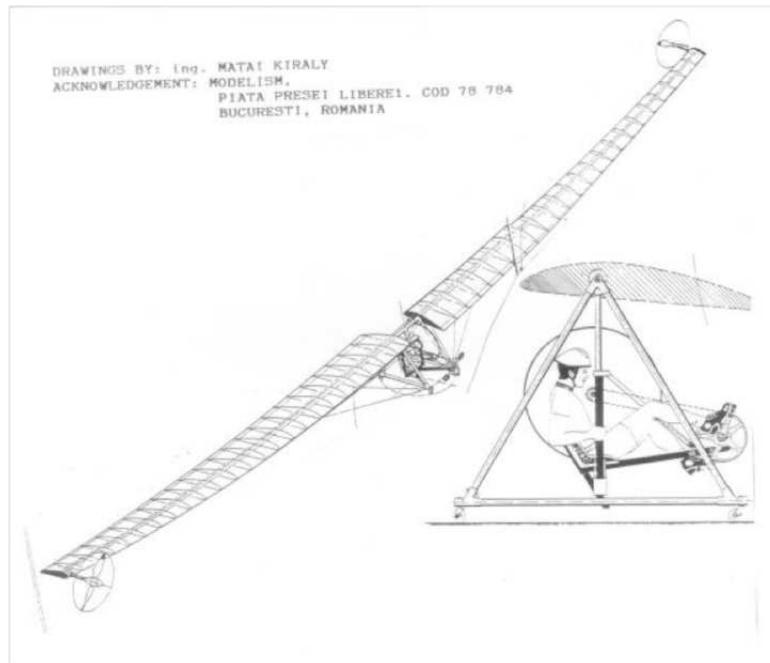
Figure 2: The Yuri I human powered helicopter.

1.1 Background

A background of related literature is provided that encompasses the relevant information regarding the design of the scaled model of the human powered helicopter. First, a review of dimensionless groups and dynamic similarity is presented, followed by blade element theory.

1.1.1 Dynamic Similarity

One of the major design goals of this project is to scale down a full-size prototype to a small model that can operate off of a lab bench. To do this, the Buckingham Pi Theorem, as presented in [5] is applied to obtain a set of Π products that yield dimensionless products.

Theorem 1.1 (Buckingham Pi Theorem [5])**.** Let K equal the number of fundamental dimensions required to describe the physical variables (in mechanics, $K = 3$: mass, length, and time). Let $P_1, P_2, \ldots, P_N$ represent N physical variables in the physical relation

$$f_1(P_1, P_2, \ldots, P_N) = 0 \tag{1}$$

Then, the physical relation Equation (1) may be expressed as a relation of $(N - K)$ dimensionless products (called Π products),

$$f_2(\Pi_1, \Pi_2, \ldots, \Pi_{N-K}) = 0 \tag{2}$$

where each Π product is a dimensionless product of a set of K physical variables plus one other physical variable. Let $P_1, P_2, \ldots, P_K$ be the selected set of K physical

5

variables. Then

$$\Pi_1 = f_3(P_1, P_2, \ldots, P_K, P_{K+1})$$
$$\Pi_2 = f_4(P_1, P_2, \ldots, P_K, P_{K+2}) \qquad (3)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\Pi_{N-K} = f_5(P_1, P_2, \ldots, P_K, P_N)$$

The choice of repeating variables, $P_1, P_2, \ldots, P_K$ should be such that they include all the K dimensions used in the problem. Also, the dependent variable should appear in only one of the Π groups.

For the human-powered flight application, the parameter of interest is the required power. The power that the motor provides to the model must represent what a human could provide to the full-size prototype and lift off the ground. This will help establish the feasibility of the human-powered helicopter.

Dynamic similarity will establish baseline power requirements for the model. Despite the fact that the model and prototype will be of different physical sizes, certain parameters may be constrained to be the same for both. By using the same airfoil geometries on the model and prototype, and enforcing that the resulting Π groups are the same, dynamic similarity will be achieved. A derivation of the Π groups and their application is given in Section 2.1.

1.1.2 Blade Element Theory

Blade element theory is the method by which aerodynamic forces may be calculated at various sections along the span of a blade. Much of the application of the theory to this project is discussed in Section 2.4. One aspect of blade element theory

6

worth mentioning here is how to determine the angle of attack across the span of the blade. The model that minimizes power required is as follows [6]:

$$\theta(r) = \frac{\theta_t R}{r} \tag{4}$$

where θ_t is the angle of twist at the tip, R is the radius of the blade, and r is the distance along the blade from root to tip. Although this method is not used in the design of the model for this project, it is worth considering in possible future designs as an alternative to a linear twist along the blade span.

1.2 Project Goals

The main problem posed by this MQP is to determine the feasibility of a low-Reynolds number human-powered helicopter. As there will not be enough resources or time to build and test a full-scale prototype, a scale model must be built and tested, the results of which will establish the practicality of such a helicopter. In the interest of future work, a flightless test stand will be developed that will be capable of measuring the lift generated by any airfoils that are connected to it. Optimization methods are relevant to this problem, since the power input from a human is limited. The main optimization parameters can be translated into figures of merit. The three figures of merit are as follows:

Table 1: Figures of Merit

Figure of Merit	Range or Limit	Desired Outcome
Total Mass	Influenced by structure and materials	As low as possible
Lift Generated	Influenced by airfoil geometry	As high as possible
Power Required	Influenced by wing geometry	As low as possible

Successfully optimizing these parameters will yield a working scale model. The goals of the project are listed below, the methods that will be used to accomplish these goals are detailed in Section 2.

1. Use the principles of dynamic similarity to scale down a human-powered helicopter to a smaller scale model.

   (a) Scale model weight.

   (b) Scale model power.

2. Optimize the airfoil geometry along the span of the blades using Blade Element Theory.

   (a) Select airfoils that show the best lift to drag coefficient data.

   (b) Determine the best angle of attack distribution over the span of the blade using drag data.

3. Re-design the WPI force balance for use in collecting lift and drag data for the airfoil.

   (a) Construct an adjustable table to balance the new test stand.

   (b) Determine a procedure for calibrating the test stand for measuring lift and drag forces.

4. Construct the scale model of the airfoils and counter-rotation transmission, along with a test stand capable of measuring the lift generated by the airfoils and the power required to do so. Test results will determine the practicality of using each designed airfoil for the construction of a a full scale prototype.

(a) Determine if model can lift its scaled weight.

(b) Determine the feasibility of constructing a prototype helicopter based on measured aerodynamic forces generated by the model.

2 Methodology

This chapter describes the methods used to design the model helicopter. Since most of the design is driven by the theoretical weight of the model, the process of dynamically scaling the estimated prototype parameters to the model parameters will be presented first. Knowing the model parameters will allow the use of blade element theory to optimize the lift and drag characteristics of the rotor blades and determine which blade designs will function best. Finally the design of the testing apparatus and the construction of the model will be presented.

2.1 Scaling the Model

After minimizing the weight of the model, the rest of the dimensional analysis may proceed. This process is described in detail in both [5] and [7]. First, list the variables involved in the scaling as follows:

Table 2: Dimensions

P	W	D_r	ω	ρ	μ	c	L	D
ML^2t^{-3}	MLt^{-2}	L	t^{-1}	ML^{-3}	$ML^{-1}t^{-1}$	L	MLt^{-2}	MLt^{-2}

From this, three repeating variables must be chosen that do not form a dimensionless group. For this scaling, $\omega, \mu,$ and D_r will repeat in the Π groups. Since there are $n = 9$ variables and $j = 3$ dimensions, there will be $n - j = 6$ Π groups. The procedure will be outlined for the first group, then it will be understood that the rest may be derived the same way.

$$\Pi_1 = \omega^a \mu^b D_r^c P^1 = (t^{-1})^a (ML^{-1}t^{-1})^b (L)^c (ML^3 t^{-3})^1 = M^0 L^0 t^0 \qquad (5)$$

Equate the exponents for each basic dimension:

$$\text{Mass:} \qquad M^b M = M^0$$

$$\text{Length:} \quad L^{-b} L^c L^2 = L^0$$

$$\text{Time:} \qquad t^{-a} t^{-b} t^{-3} = t^0$$

$$b + 1 = 0$$

$$-b + c + 2 = 0$$

$$-a - b - 3 = 0$$

Solving these simultaneously for $a, b,$ and c yields

$$a = -2$$

$$b = -1$$

$$c = -3$$

Substituting these into Equation 5 yields

$$\Pi_1 = \frac{P}{\omega^2 \mu D_r^3} \tag{6}$$

Proceeding in a similar fashion for the other groups, obtain

$$\Pi_2 = \frac{\rho \omega D_r^2}{\mu}$$

$$\Pi_3 = \frac{C}{D_r}$$

$$\Pi_4 = \frac{W}{\omega \mu D_r^2}$$

$$\Pi_5 = \frac{L}{\omega \mu D_r^2}$$

$$\Pi_6 = \frac{D}{\omega \mu D_r^2}$$

According to [9], it can be shown that not all Π groups can be satisfied for a model less than the full-scale size. However, certain parameters should still maintain their full-scale values to maintain dynamic similarity. For the purpose of sizing the rotor blade, Π_3 will be used as follows: set the ratio of chord length to rotor diameter to be equal for the prototype and model.

$$\frac{C_p}{D_{r_p}} = \frac{C_m}{D_{r_m}}$$

$$\frac{0.915 \text{ m}}{15.25 \text{ m}} = \frac{C_m}{D_{r_m}}$$

$$0.06 = \frac{C_m}{D_{r_m}}$$

This means the ratio of model chord length to rotor diameter must equal 0.06. A diameter of 4 feet (1.22 meters) was chosen for size considerations, which means the model chord length must be 0.24 feet (0.073 meters).

As is stated in [10], two other scaling factors, the Mach number and Froude number, may be used to obtain design characteristics. It is difficult to satisfy both

parameters together, so the choice depends on the application. Since the model will be operating at low speeds, Froude scaling will be favored. The Froude number is a dimensionless characteristic velocity, given by:

$$\text{Fr} = \frac{V^2}{gc} = \frac{(\omega D_r)^2}{gc} \tag{7}$$

where $V = \omega D_r$, $g = 9.81$ m/s^2 is the gravitational constant, and c is the chord length. Setting the Froude number of the prototype equal to that of the model, one may solve for the required angular velocity of the model.

$$\text{Fr}_p = \text{Fr}_m$$
$$\frac{(\omega_p D_{r_p})^2}{g_p c_p} = \frac{(\omega_m D_{r_m})^2}{g_m c_m}$$
$$\omega_m = \omega_p \left(\frac{D_{r_p}}{D_{r_m}}\right) \sqrt{\frac{c_m}{c_p}} \tag{8}$$

Let a scaling factor λ be defined as $\frac{\text{model length}}{\text{prototype length}}$, such that for this project, $\lambda = 0.08$. Equation 8 may be rewritten as

$$\omega_m = \omega_p \left(\frac{1}{\lambda}\right) \sqrt{\lambda} \tag{9}$$

Knowledge of the prototype angular velocity provides the model angular velocity. At about 10 rpm for the full-scale helicopter, the model rotors should rotate at

$$\omega_m = \frac{10}{\lambda} \sqrt{\lambda} = \frac{10\sqrt{0.08}}{0.08} = 35.355 \text{ rpm} \tag{10}$$

13

To properly calculate the model blade dimensions, the scaling factor λ must be implemented. Set the total area of all the model blades to be equal to the scaled area of all the prototype blades, modeled in the equation below:

$$n_m A_m = n_p \lambda^2 A_p$$

where n_m is the number of blades for the model, A_m is the area of each blade, n_p is the number of prototype blades, and A_p is the area of each prototype blade. For the Da Vinci III, which had two blades, $n_p = 2$. Solving for A_m yields

$$A_m = \frac{n_p \lambda^2 A_p}{n_m} \tag{11}$$

Maintaining a constant aspect ratio between the model and prototype blades, the model span and chord may now be found

$$b_m = \sqrt{AR \cdot A_m} \tag{12}$$

$$c_m = A_m/b_m \tag{13}$$

While these equations are implemented in the blade element code, in the interest of time, the actual blades were simply assigned dimensions for root/tip chord and blade span. With a straight leading edge and elliptically tapered trailing edge, the shape and area of the actual wing was determined independently of this analysis, though the actual area was found to be close to that predicted by the scaling method described above.

A summary of prototype and model parameters are presented in the table below:

Table 3: Summary of Key Design Parameters

Parameter	Prototype	Model
Power (watts)	500	13.8
Weight (N)	1020	2.75
D_r (m)	15.25	1.85
ω (rpm)	10	35.4
ρ (kg/m^3)	1.23	1.23
μ (kg/m $\cdot$ s)	1.81×10^{-5}	1.81×10^{-5}
Chord (m)	0.915	0.134
Lift (N)	1100	7.9
Drag (N)	300	1.13

2.2   Helicopter Design

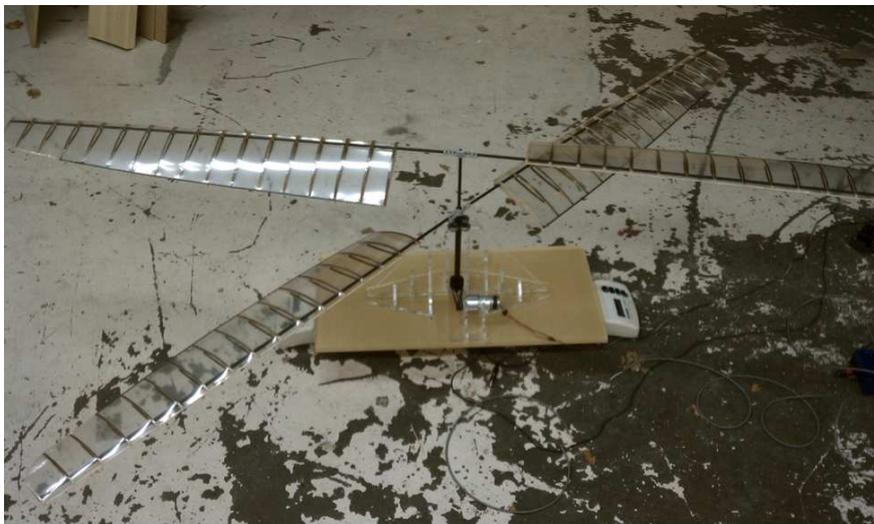Below is the final constructed model of the HPH for this MQP.

Figure 3: Constructed helicopter constrained to the base. When operating, the sum of the scale readings is the lift generated by the model.

Based on the initial research performed on HPH designed in the past, there were several design paths that could be taken. The obvious solution was to look at the most successful prototypes constructed in the past. However it seems that the most successful designs constructed in the past each had major inherent issues. The Yuri I HPH required a massive truss work to support its rider and its blades. It seems that devoting a significant portion of the weight of the machine to a truss system is wasteful. The next most successful HPH ever designed was the Da Vinci III. The design was inherently unstable to the point that the machine had to be supported by humans throughout its entire flight. Also, locating tip drive rotors at the wing tips is not as efficient as a purely mechanical drive system. After much research and deliberation it was decided that a counter-rotating design was the best for this application. A counter-rotating design would be inherently stable, and require limited weight to be allocated to a base structure.

If a 150-pound rider weight, and a 100-pound helicopter weight are assumed, the total weight of the system will be about 250 pounds. For this project, the theoretical scaled weight should be about 0.38 pounds (described in Section 2.1). Rather than attempting to construct a full model that weighs less than 0.38 pounds a test stand was developed that is capable to measuring the lift produced by the rotor blades, and power consumed. The model will however be designed using a counter rotating shaft design, and twisted timing belt transmission driven by an electric motor. Incorporating these design aspects into the model will prove their worth on the full scale prototype.

The counter rotating shaft assembly incorporates two pultruded carbon fiber tubes. This means that the tubes are made up of long strands of carbon fiber all aligned along the major axis of the tube. The inner shaft will be longer and of

a smaller diameter than the outer shaft. The inner shaft will have a 0.158 in ID (inner diameter), and a 0.254 in OD (outer diameter), and the outer shaft will have a 0.400 in ID and a 0.500 in OD, these carbon fiber tubes are available through Dragon Plate. Ball bearings will be purchased that will fit in-between the shafts such that the shafts spin freely and support each other. Ball bearings with a 0.25 in ID, and 0.375 in OD are available through Stock Drive Products.

An interference fit can be achieved in between the OD of the inner shaft and the ID of the bearing. On the other hand, in order to achieve a fit between the OD of the bearing and the ID of the outer shaft bearing seats will need to be designed and machined. The inner shaft is longer than the outer shaft to create a separation between the two rotor disks (one connected at the upper end of the outer shaft, and one connected at the upper end of the inner shaft). This separation is intended to minimize the aerodynamic interference that each rotor will have on one another. The inner shaft also protrudes bellow the bottom of the outer shaft to provide a mounting surface for the inner shaft drive pulley, and a point to mount the entire shaft assembly to the base structure. A SolidWorks model and a picture of the finished part are included below.

In the interest of managing time, the rest of the base test stand was designed before the design of the blades was finalized. The motors and pulleys were selected based on ω_m (model angular velocity) predicted by the dynamic scaling process. The base test stand consists of two large arbors that position four bearings that support the outer shaft just below the lower rotor disk. All of the parts of the base were laser cut out of acrylic, and assembled using acrylic cement. A single bearing on the bottom of the inner shaft supports the shaft assembly at its bottom. This rigid mounting system is designed to limit the vibrations that may affect data

(a) CAD model.   (b) Finished assembly.

Figure 4: Counter-rotating shaft.

collection. The base structure also incorporates an adjustable motor mount that will be capable of moving to tension the two belts. The motor that was selected to use is a DC gear motor that has a shaft output of about 3.75 rpm/v. To optimize this motor shaft rotational speed form 0-30 volts two 0.08 in pitch 26 groove timing pulleys were modified such that one double groove triple flange pulley assembly was created and fixed to the motor shaft as pictured below. The double groove motor pulley is connected to two 0.08 in pitch 40 groove timing pulleys, one affixed to the inner shaft and one to the outer shaft, with two 1/8 in wide 8.64 in long

grooved timing belts. A SolidWorks model of the base structure, motor and pulleys is included below.
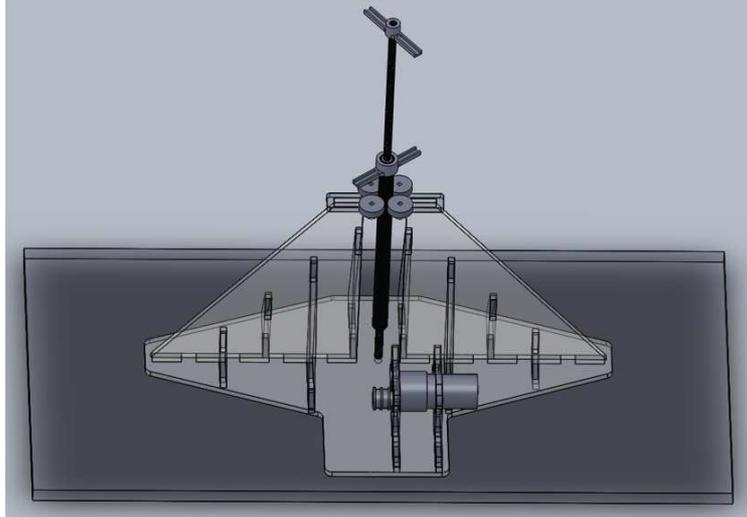

Figure 5: CAD model of the helicopter base (the counter-rotating shaft is included).

Figure 6: A close-up of the pulleys shows how to produce the counter-rotation.

2.3   Design of a New Test Stand

The next challenge was to design a user friendly test stand that was capable of measuring approximately one gram of drag force, and approximately ten grams of lift force. This was necessary because the existing test stand was not capable of measuring such small drag forces. The test stand was designed with manufacturability, and future usability in mind. The first improvement that was made was to replace the stack of cinder blocks that was used to support the original test stand underneath the wind tunnel with amore sturdy table. The table was constructed with a large flat surface, and four legs that each contain a 12 inch threaded rod connected to padded feet. The table also features a permanently fixed bubble level on each axis so that the table top surface can quickly and easily be leveled. The tabletop surface has eight holes bored in it that serve as points for the feet of two electronic scales to rest in. The stand will eventually sit on top of these scales and the scale readings provide the data.

The holes were bored so that the scales would sit in the same place on the table, during each test. They were also placed so that the scales would be held at a convenient center to center distance of 10 inches apart. This 10-inch distance would allow future users to use either the original test stand or the new test stand without changing the scale configuration. It is also a convenient number to be factored into the static equilibrium equations that will be discussed in Appendix A. A picture of the table with the scales placed in their locating holes is pictured below.

Figure 7: The test table with two scales mounted in the bored holes. The test stand will sit on top of the scales so that lift can be measured.

The test stand was designed to support the wing-stand interface with a four bar linkage, similar to the original test stand. The major difference is that the new test stand features a built in indexing slide that with the twist of a wing nut allows the user to adjust the angle of attack of the wing inside of the wind tunnel to a known value between -15 and 45 degrees. The scale is in increments of full degrees but the user is not constrained to a minimum increment of adjustment. This feature makes testing faster, easier, more accurate, and more repeatable. A picture of the SolidWorks model and the actual test stand are included below.

Figure 8: CAD model of the test stand.
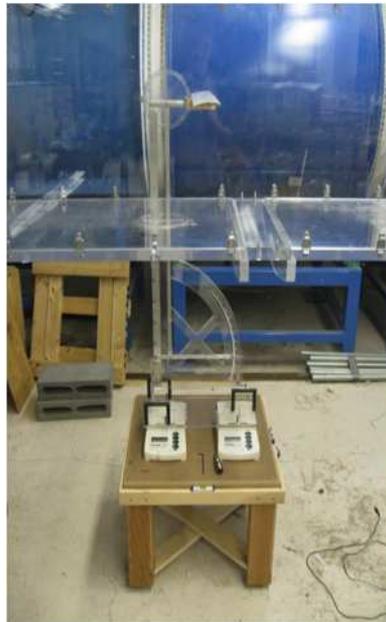
Figure 9: The test stand configured for lift data collection.
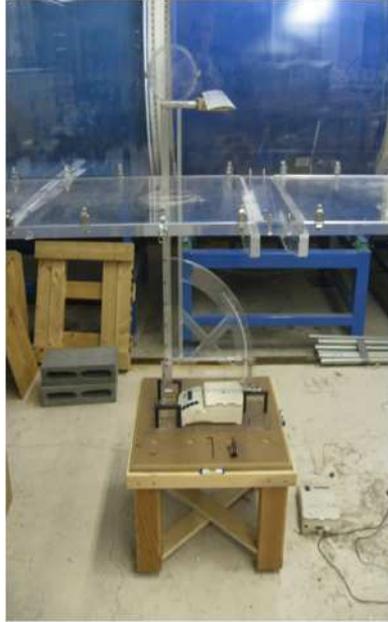
Figure 10: The test stand configured for drag data collection.

2.4   Lift and Drag Testing

The first step in designing the rotor blades was researching low speed airfoil cross sections. Data from [8] was used as the major source of air foil shapes and data. The data collected for most of the airfoils in the book only went down to a Reynolds number of about 60,000. The Reynolds number that the model experiences is about 1000.

$$\mathrm{Re} = \frac{\rho V c}{\mu} = \frac{(1.23)(0.2)(0.0762)}{1.81 \times 10^{-5}} = 1000 \tag{14}$$

The process of airfoil selection was to examine all of the coefficient of lift versus coefficient of drag curves for each airfoil at a Reynolds number of 60,000. An

assumption was made that the airfoil that had the highest C_L to C_D at 60,000 would perform the best at even lower Reynolds numbers. Due to time and financial constraints it was decided that a single airfoil would be used along the entire length of the rotor blade. After examining the data for each airfoil, the highest Cl to Cd ratio was exhibited by the E214 airfoil. The C_L versus C_D curve, the C_L versus angle of attack, and the points defining the surface of the E214 airfoil are included below. The curve data is for the E214C-PT, unfortunately the surface data was only available for the E214 parent airfoil.
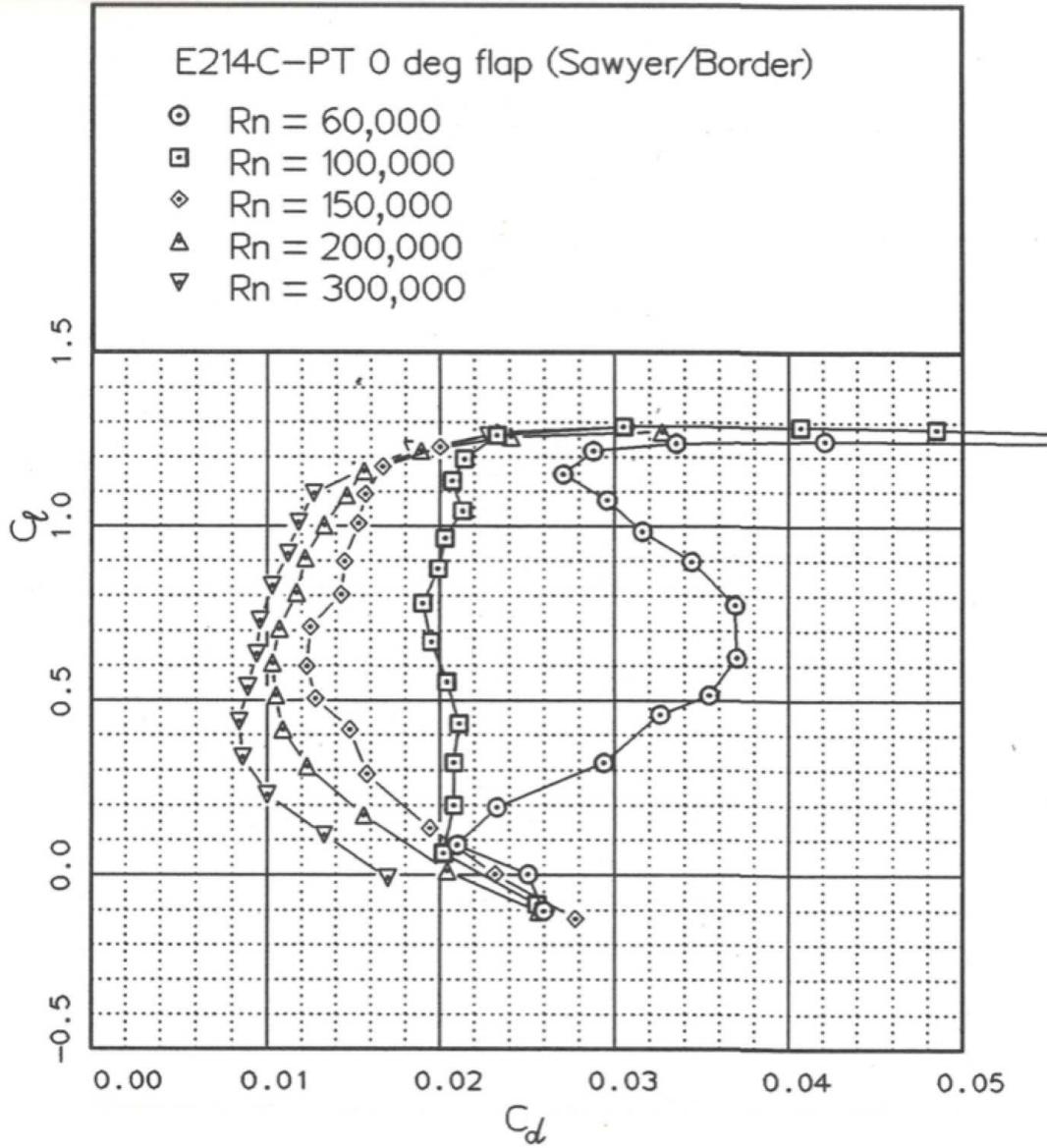
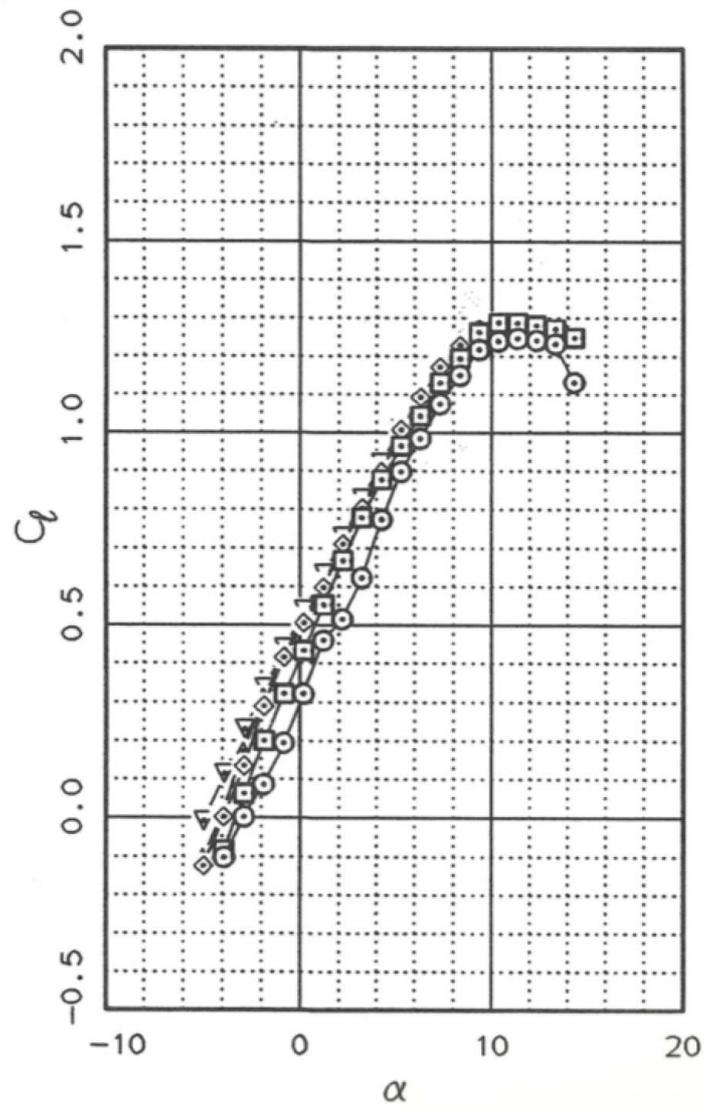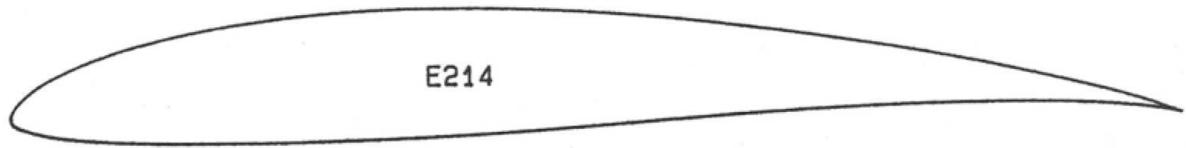Figure 11: Lift versus drag coefficient at various Reynolds numbers [8].

Figure 12: Lift coefficient versus angle of attack [8].

26

E214

1	1.00000	0.00000	17	0.47142	0.09093	33	0.00000	0.00000	49	0.52450	-.00678
2	0.99669	0.00104	18	0.42346	0.09281	34	0.00005	-.00086	50	0.57932	-.00245
3	0.98737	0.00422	19	0.37645	0.09332	35	0.00360	-.00632	51	0.63400	0.00155
4	0.97312	0.00916	20	0.33076	0.09241	36	0.01326	-.01087	52	0.68770	0.00495
5	0.95431	0.01501	21	0.28674	0.09008	37	0.02830	-.01475	53	0.73959	0.00756
6	0.93081	0.02139	22	0.24474	0.08639	38	0.04858	-.01784	54	0.78883	0.00923
7	0.90279	0.02833	23	0.20510	0.08142	39	0.07390	-.02011	55	0.83461	0.00994
8	0.87072	0.03576	24	0.16816	0.07532	40	0.10406	-.02161	56	0.87612	0.00970
9	0.83508	0.04344	25	0.13424	0.06822	41	0.13874	-.02236	57	0.91265	0.00862
10	0.79626	0.05102	26	0.10365	0.06028	42	0.17759	-.02245	58	0.94352	0.00684
11	0.75457	0.05841	27	0.07665	0.05168	43	0.22017	-.02193	59	0.96809	0.00461
12	0.71040	0.06544	28	0.05349	0.04258	44	0.26599	-.02086	60	0.98582	0.00235
13	0.66430	0.07207	29	0.03434	0.03321	45	0.31449	-.01928	61	0.99646	0.00065
14	0.61682	0.07813	30	0.01934	0.02379	46	0.36508	-.01721	62	1.00000	0.00000
15	0.56852	0.08344	31	0.00856	0.01465	47	0.41714	-.01453			
16	0.51991	0.08776	32	0.00210	0.00619	48	0.47030	-.01100			

Figure 13: Geometry of the E214 airfoil [8].

After selecting the E214 as the airfoil cross section for the rotor blades the next step was to gather actual lift data at a Reynolds number closer to the actual operating Reynolds number. To make a model of the airfoil, the surface point data tabulated above was used to draw a cross section of the E214 in SolidWorks. A square hole for indexing the angle of attack on a square wing spar was added along with a second round indexing hole used to structurally tie the test wing together. An image of the SolidWorks drawing including the indexing holes is included below.
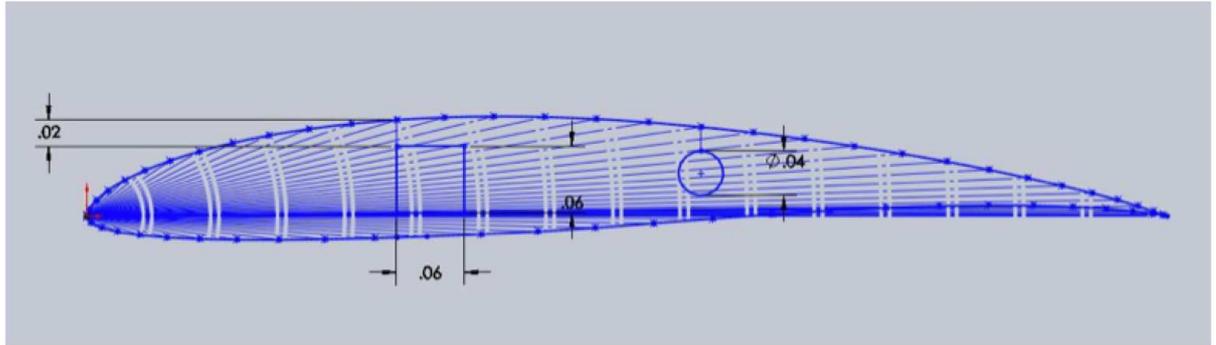
Figure 14: SolidWorks implementation of the E214 airfoil.

A wind tunnel test stand was designed that made taking accurate and repeatable lift and drag measurements at low Reynolds numbers possible. This test stand was designed to easily interface with the two-foot test section continuous loop wind tunnel in the basement of Higgins Labs.

In the wind tunnel experiments, two separate tests were performed to collect raw left and drag data. Data was collected from an angle of attack ranging from -5 to 20 degrees at a Reynolds number of 10,000. Calibrations were performed on each test set up, and actual left and drag data were calculated. A complete instruction manual for setting up and using the wind tunnel test stand is included in Appendix A. A Matlab code is used to calculate the lift coefficient at each angle of attack. The resulting lift and drag coefficients from the testing are shown in Tables 4 and 5. The raw data is in Appendix B.

Table 4: Lift Coefficients: New Test Stand

AoA (degrees)	C_L: Test 1	C_L: Test 2	Percent Difference (%)
-5.0	0.5273	0.4554	15.7895
-2.5	0.4264	0.3909	9.0909
0.0	0.4630	0.4283	8.1081
2.5	0.5567	0.4975	11.9048
5.0	0.5633	0.5393	4.4444
7.5	0.6066	0.6982	-13.1148
10.0	0.7669	0.7898	-2.8986
12.5	0.8602	0.8602	0.0000
15.0	1.0639	1.0187	4.4444
17.5	1.2437	1.2319	0.9615
20.0	1.3859	1.4332	-3.3058

The next table shows the calculated drag coefficients from the drag testing.

Table 5: Drag Coefficients: New Test Stand

AoA (degrees)	C_D: Test 1	C_D: Test 2	Percent Difference (%)
-5.0	-0.2510	-0.2605	-3.6535
-2.5	-0.1523	-0.1523	0.0000
0.0	-0.0749	-0.0650	15.0728
2.5	-0.0372	-0.0272	0.0000
5.0	0.0599	0.0599	0.0000
7.5	0.0667	0.0765	-12.9001
10.0	0.1027	0.1027	0.0000
12.5	0.1193	0.1193	0.0000
15.0	0.0863	0.1159	-25.5567
17.5	0.0886	0.0886	0.0000
20.0	0.1126	0.1126	0.0000

The negative coefficients are most likely due to a slight error in the calibration. One of the assumptions in collecting the lift and drag data using the new test stand is that lift is measured independently, i.e., drag is not coupled into the lift data. This was found to be true for the most part, however, it may be slightly off, in which case, the lift forces being measured overcompensate in the drag equation (39). As a result, future work may need to be done in refining the calibration of

the new test stand.

The data from Tables 4 and 5 is passed through a linear regression function in Matlab and the best fit line equations are displayed. For each test (using the new test stand), the results are as follows:

$$C_{L_1} = 0.0035x + 0.0468 \tag{15}$$

$$C_{L_2} = 0.0038x + 0.0437 \tag{16}$$

$$C_{D_1} = 0.0013x - 0.0084 \tag{17}$$

$$C_{D_2} = 0.0013x - 0.0083 \tag{18}$$

It is clear that the line equations for lift are consistent, which is also indicated by the percent error column in Table 4. The actual data points and corresponding best fit lines are displayed in the figures below.

Figure 15: Initial test results of lift versus angle of attack (using the new test stand). The red dots are the measured values and the blue line is the linear regression.

Figure 16: Results of the second test of lift versus angle of attack (using the new test stand). The red dots are the measured values and the blue line is the linear regression.
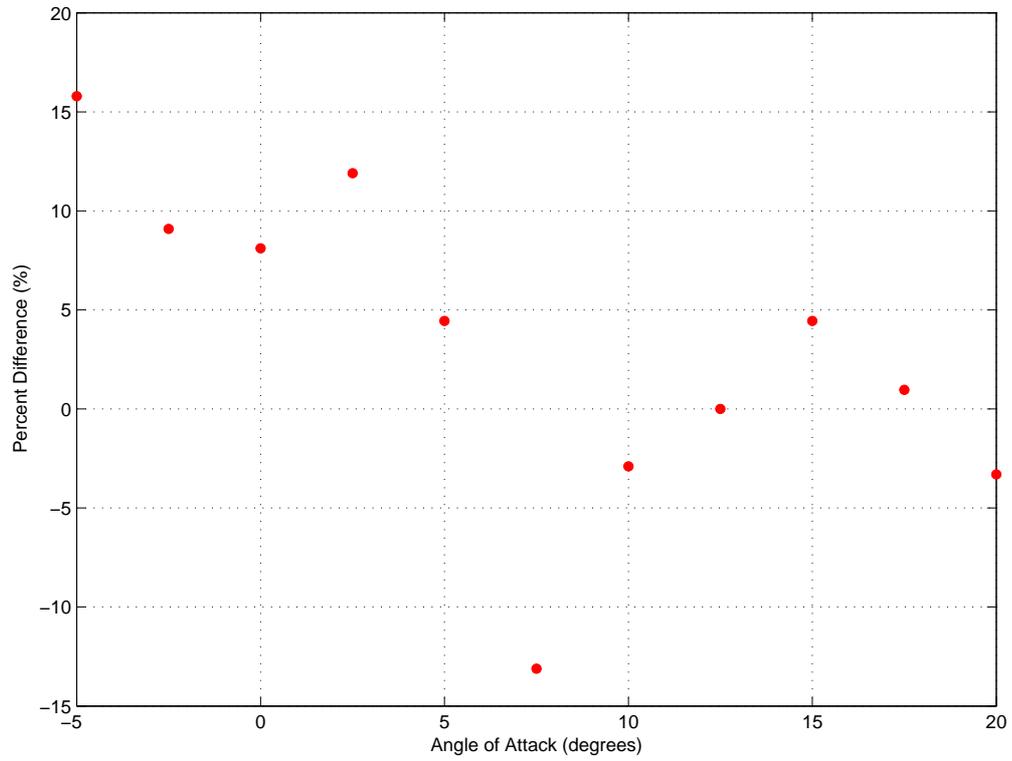
Figure 17: For clarity, the percent difference between the two tests (using the new test stand) is presented.
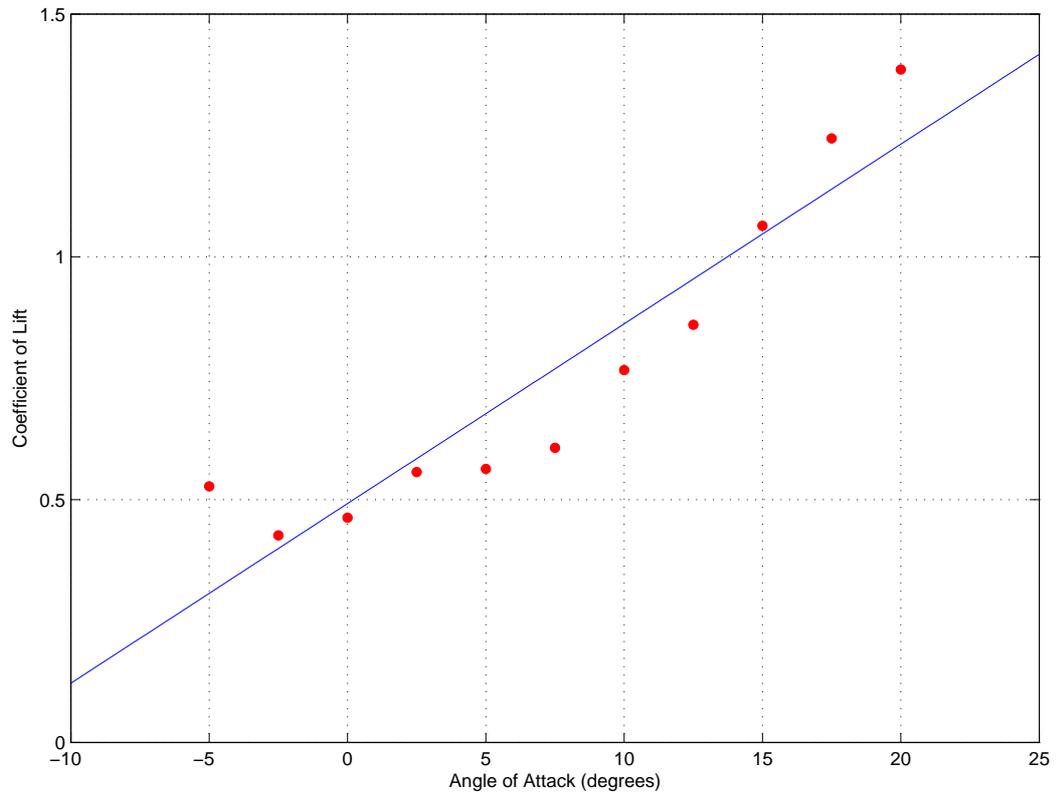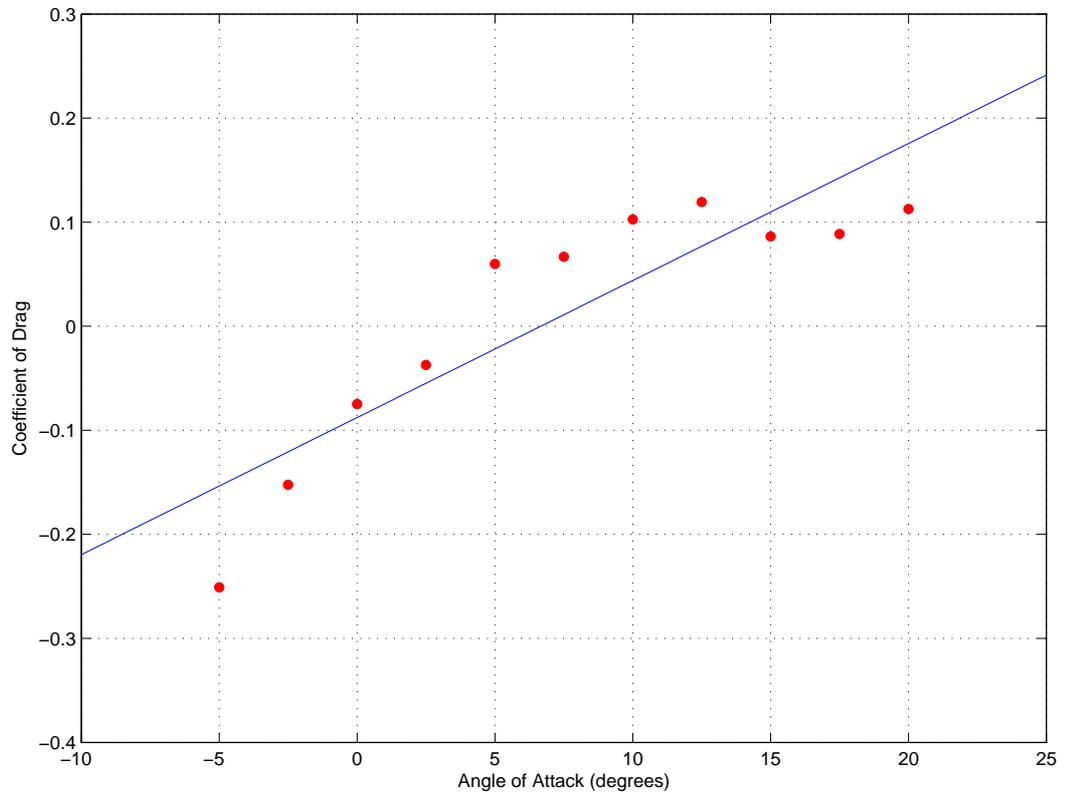
Figure 18: Initial test results of drag versus angle of attack (using the new test stand). The red dots are the measured values and the blue line is the linear regression.
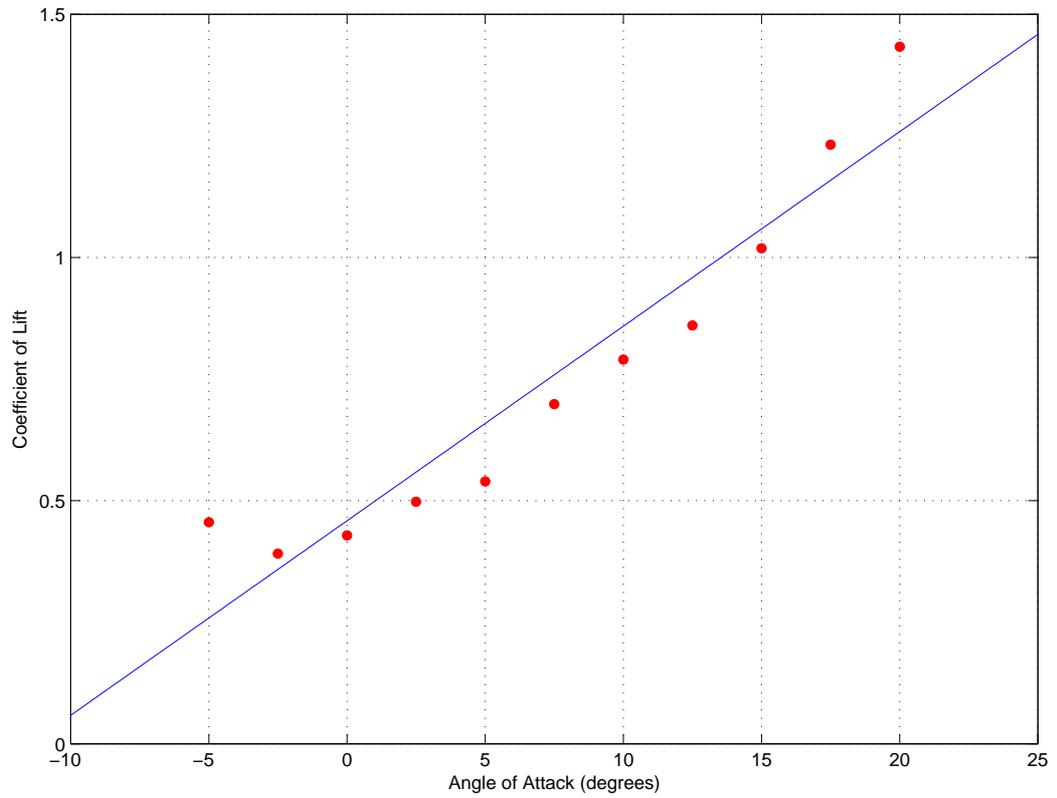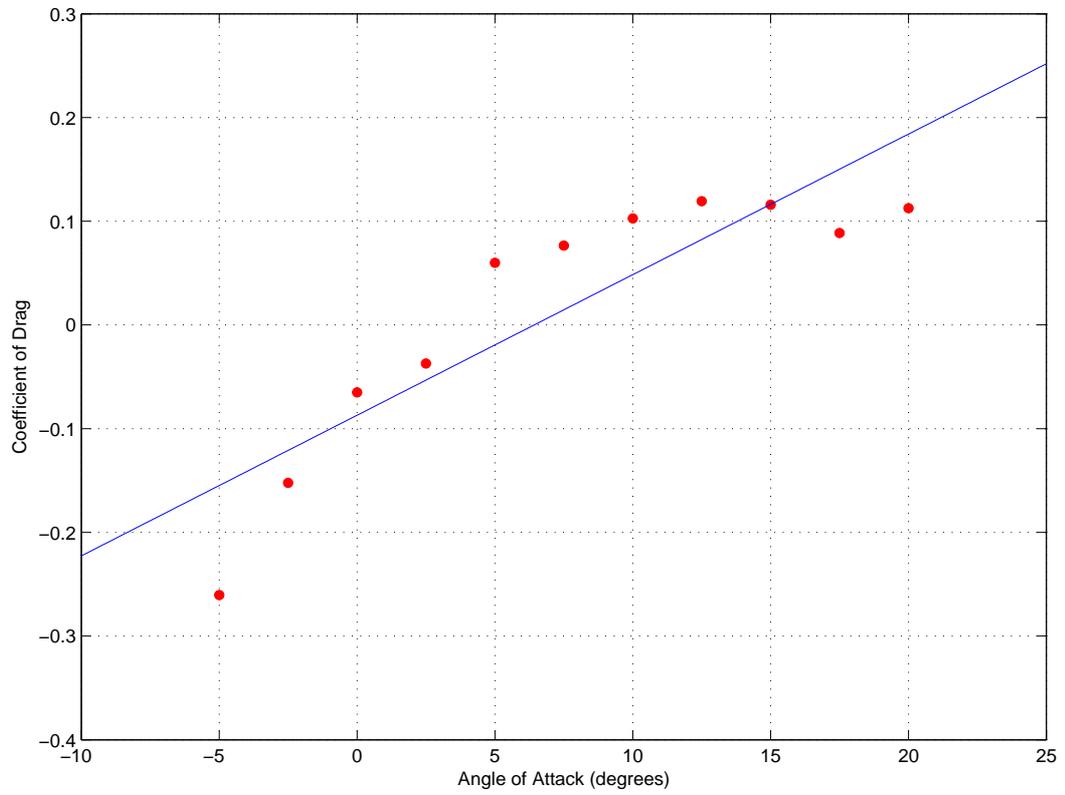
Figure 19: Results of the second test of drag versus angle of attack (using the new test stand). The red dots are the measured values and the blue line is the linear regression.
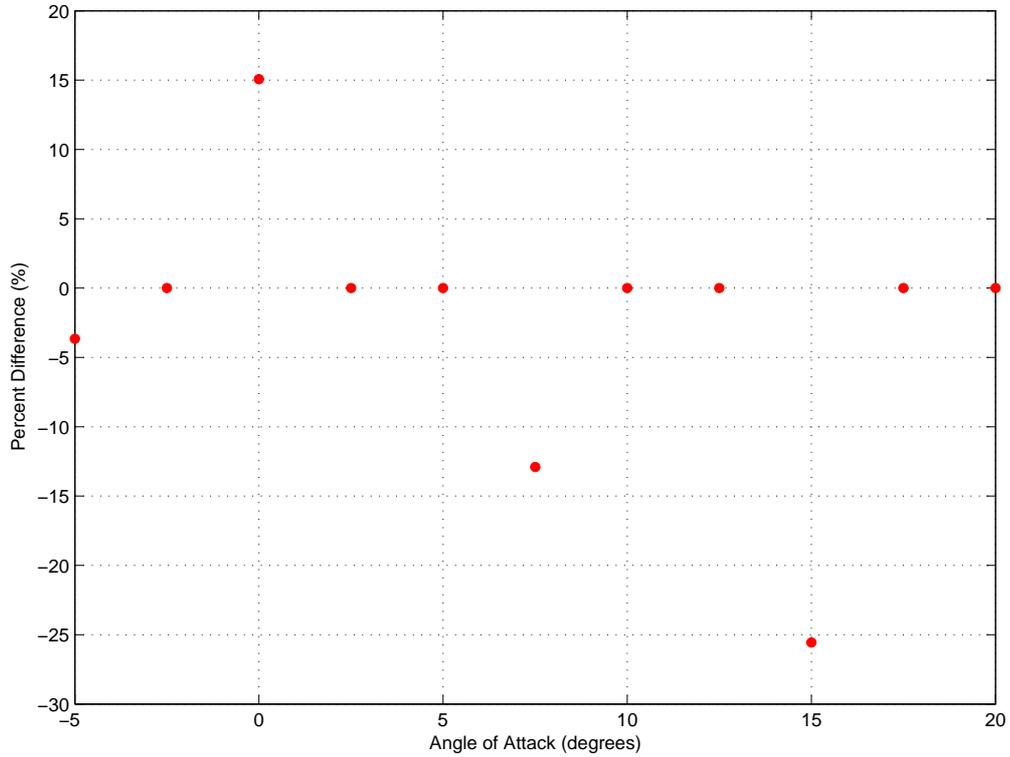
Figure 20: For clarity, the percent error between the two tests (using the new test stand) is presented.

2.5 Construction of the Blades and Blade Shaft Interface

The blade-shaft interface was designed to support two blades on each of the rotor shafts. The blade-shaft interface was designed in SolidWorks and manufactured using a Mini-mill in Washburn shops. The interface is pictured below and designed to accept a square wing spar that is affixed with three wire ties. Hollow square carbon fiber tubes of 0.188 inches were used as continuous wing spars. These carbon fiber tubes are available in 4-foot sections through Dragon Plate. Wing ribs were cut out of 3/32-inch balsa wood sheet using the laser cutter in Washburn

36

Shops. It is important to note that the factory settings for cutting balsa wood did not work well. The best results were obtained by manually setting the power to 100%, the speed to 35%, and the PPI to the maximum setting. The Matlab code was used to find the chord length and angle of attack at each two-inch interval along the span of the wing. The procedure for drawing each wing rib consisted of scaling the original E-214 SolidWorks drawing to the correct chord length, and placing a notch directly at the leading edge for the leading edge stringer. Each spar hole is located 0.65 inches from the leading edge and rotated to the proper angle of attack. Lightening holes were added to make the wing ribs as light as possible. An example of a drawing of a wing rib is included below.
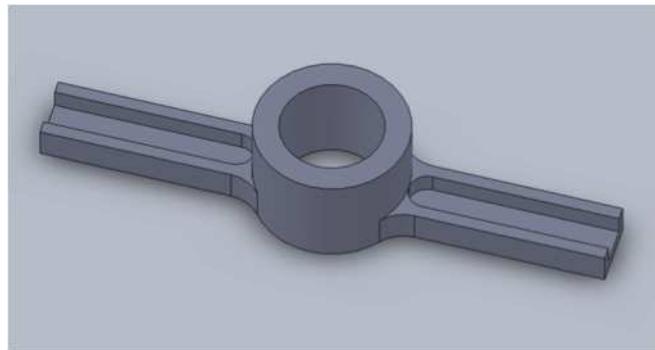
Figure 21: The blade-shaft interface, which attaches to the counter-rotating shaft. The wing spars attach to the protrusions from the central ring.
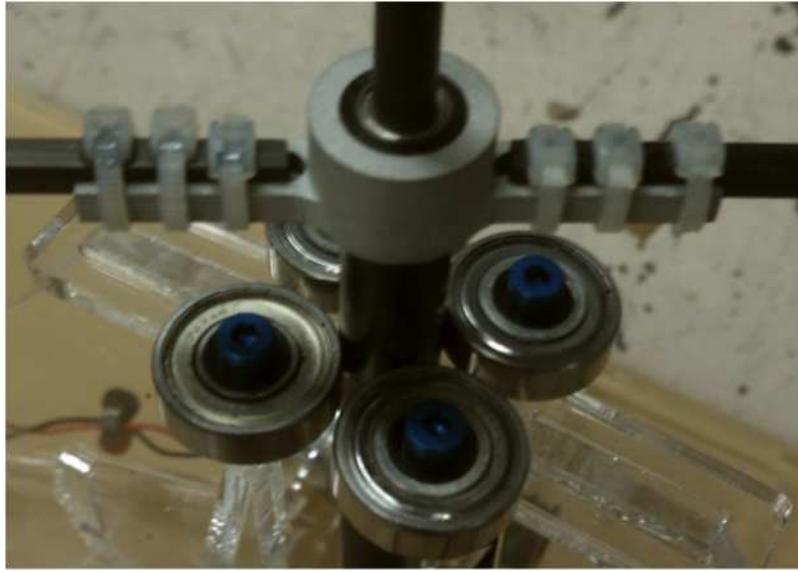
Figure 22: The constructed blade-shaft interface, with wing spars secured to the assembly.
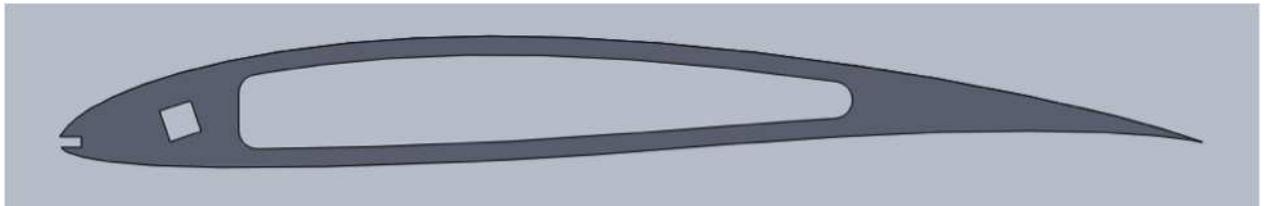
Figure 23: CAD model of a wing rib, showing three main components: the leading edge notch, the angled spar hole, and the lightening hole, which occupies a considerable area of the rib.

Knowing that the blade length is 32 inches and the spar length is 34.5 inches (input in the Matlab code); all four spars were cut to length at the same time. Each of the 17 stations were slid onto the spar one at a time and placed two inches apart. The leading and trailing edge stringers were attached, and all the parts were glued in place. A picture of an uncovered wing is provided below.

Figure 24: Wing spar with the cut ribs attached. This constitutes one of the four rotor blades.

All surfaces of the blades were sanded and cleaned off with compressed air. A very thin layer of a mixture of 3M spray adhesive and lighter fluid was applied to the upper and lower surfaces of each wing using a roller. A single and separate piece of Glad plastic wrap was then used to cover the upper and lower surfaces of the wing. This covering process is extremely delicate and extreme care must be taken to make sure that the covering goes on with as few initial wrinkles as possible. After the initial application the covering is trimmed with an Exacto knife and allowed to dry overnight. The plastic covering can then be shrunk to

its finished form with a hair dryer. Keep in mind that excessive heat pointed at a single point for too long will cause the covering to melt. If the hole is small enough, it can be patched with a thin piece of packing tape. When making the blades, great care was taken to make sure that each blade was made in exactly the same way. A picture of a finished wing is included below.
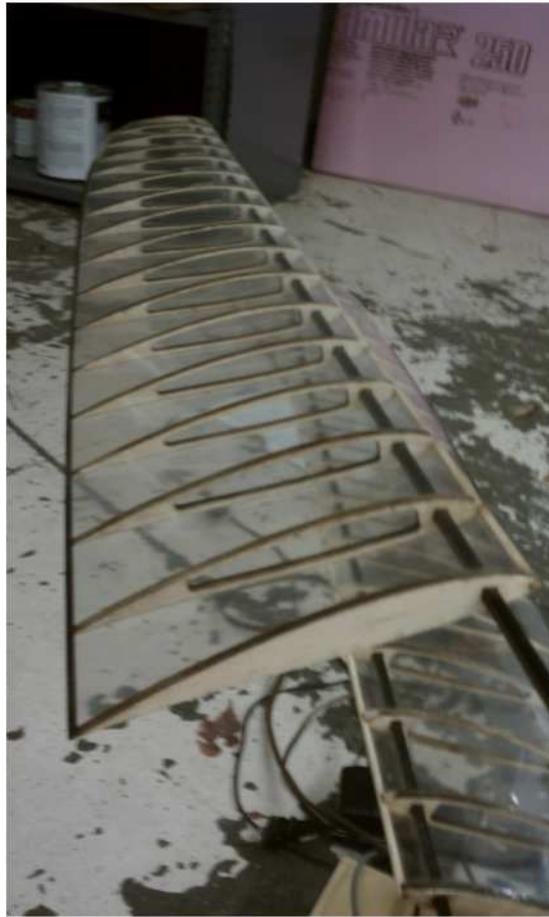
Figure 25: The finished wing is shown in its final position on the model. The wing taper and geometric twist are evident in this figure.

A control box was designed and constructed that is capable of both controlling the rotational speed of the DC gear motor and monitoring in real time the power

that it is consuming. The internal meter displays volts which correlate to motor speed, and the amps that the motor draws in real time. It is capable of outputting between 0 and 28 volts, and providing about 0.5 amps of current. The control box is labeled for future user convenience. A picture of the control box is provided below.

Figure 26: The power supply box with power switch and voltage controller.

2.6 Blade Element Code

A blade element code was written in Matlab to calculate lift, drag, torque, and power requirements over each blade for the model. The code is flexible and allows chord length and angle of attack to change along the span. All the required scripts and functions are presented in Appendix C.

The code uses a number of inputs to generate the required outputs. The main

inputs include: desired blade root/tip chords, span, minimum/maximum angles of attack, number of blade elements to consider, lift and drag data (wind tunnel testing using the *new* test stand), known and estimated prototype characteristics, and estimated model characteristics. The main output values are the total lift and power required for takeoff. The code automatically checks if the lift is sufficient for takeoff by comparing with the estimated model weight. The code may be modified such that a range of minimum and maximum angles of attack can be considered. In this case, for each combination, the power required is output in a table. This allows the user to determine the most efficient choice of root and tip angles of attack. For blade rib construction purposes, the code also outputs the chord length and angle of attack at each section.

The code is set so a straight leading edge and elliptically tapered trailing edge is applied to the blade geometry. The three inputs that determine the chord length at each section according to this scheme are: root chord, tip chord, and blade span. In the following equations, TE refers to trailing edge and LE refers to leading edge.

$$\text{TE}_i = \frac{c_r}{2}\sqrt{1 - \frac{(x_i - R)^2}{b_m^2}} + \frac{c_r}{2} \tag{19}$$

$$\text{LE}_i = \frac{\left(\frac{c_r}{2} - c_t\right)x_i}{b_m} \tag{20}$$

where c_r is the root chord, c_t is the tip chord, R is the distance from the center of rotation to the blade root, b_m is the blade span, and x_i is the x-coordinate of the

ith blade element along the span. This method will ensure that the leading edge is straight, and preserves the desired root and tip chords, as well as span. With coordinates of the leading and trailing edges obtained, the chord length at the ith blade element is just the difference between TE and LE:

$$c_i = \text{TE}_i - \text{LE}_i \tag{21}$$

For the project, the inputs to the blade element code are set as follows (dimensions in inches):

$$c_r = 6$$
$$c_t = 3$$
$$b_m = 32$$
$$R = 4.5$$

A total of seventeen sections are taken along the span of the blade, such that each section is exactly two inches apart. The resulting top view of the blade is generated by the code:
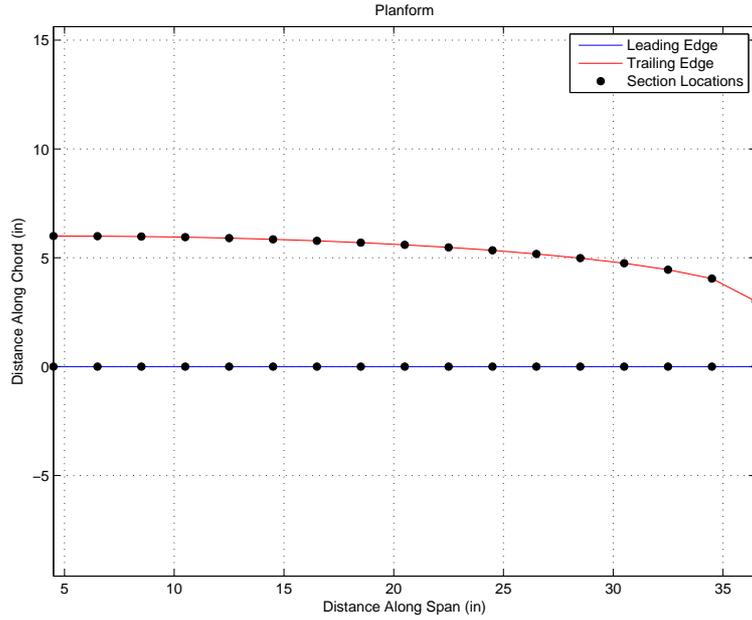
Figure 27: Top view of the blade geometry. Each black dot corresponds to the locations of the sections on the leading and trailing edges where the aerodynamic forces are calculated.

It is apparent that the root of the blade does not start at the center of rotation. This is physically impossible because the structure of the blade-shaft interface prevents this from happening. This translates into $R \neq 0$ for the code. In addition, the angle of attack at the root and tip are set as follows:

$$\alpha_{\text{root}} = 20°$$

$$\alpha_{\text{tip}} = 5°$$

To obtain the lift and drag distribution along the blade, the following equations are implemented in the code:

$$L_i = \frac{1}{2}\rho V_i^2 S_i C_{L_i} \tag{22}$$

$$D_i = \frac{1}{2}\rho V_i^2 S_i C_{D_i} \tag{23}$$

where

$$S_i = (x_{i+1} - x_i)c_i \tag{24}$$

is the area of each blade element. Each one experiences a different velocity, has a different area (due to the chord taper), and has a different lift and drag coefficient (due to varying angle of attack). With these calculated, the distribution may be plotted, and is shown below.
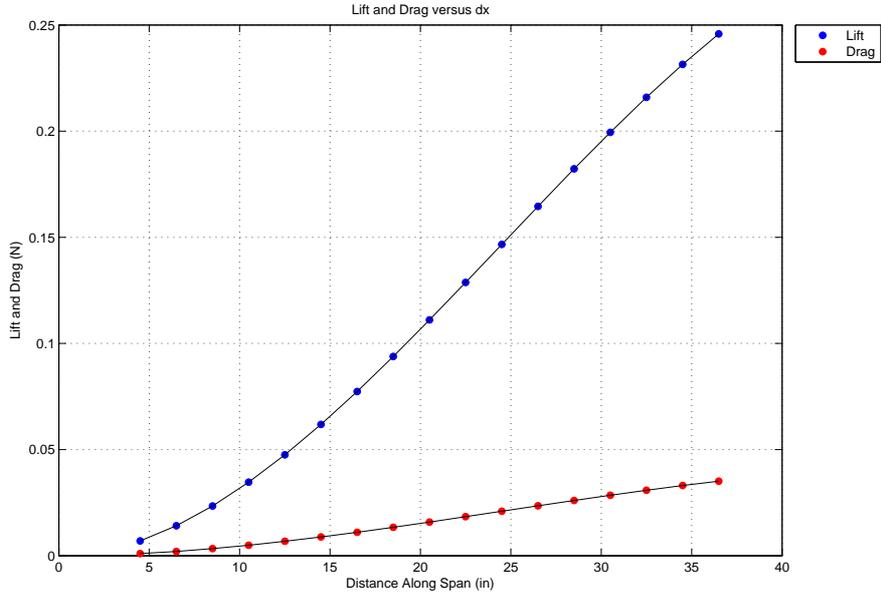
Figure 28: Lift and drag forces as a function of distance from the root of the rotor. To calculate the total lift, sum the lift at each section.

Note that this plot is the lift at each section for just one blade. The total lift is obtained by multiplying by the number of blades used, and is found as follows:

$$L_{\text{total}} = n \cdot \sum_{i=1}^{N} L_i \tag{25}$$

where n is the number of blades, and L_i is the lift at the ith section of the a single blade, and the summation is carried out for all N finite blade sections.

Once the lift and drag distributions have been found, torque and power must be computed. The drag force at each section applies a moment to the HPH, which can be modeled as:

46

$$M_i = x_i D_i \tag{26}$$

$$M_{\text{tot}} = \sum_{i=1}^{N} M_i \tag{27}$$

$$P_{\text{req}} = M_{\text{tot}} \omega_m \tag{28}$$

This calculated P_{req} is the required power for takeoff, assuming a scaled blade angular velocity. The goal is to minimize the power required, and the best way to do that is to design the most efficient blade planform possible such that the drag at each section is minimized. This minimizes the moment, and in turn, power. Assessing whether or not the design is efficient enough is addressed in the next section.

3   Scale Model HPH Testing and Results

Testing the lift generated by the model required placing the entire model test stand assembly on top of two digital scales as pictured below. With the motor connected to the shafts with the timing belts the motor control box was then turned on and plugged in. Starting at 10 volts, voltage and amperage data was collected up to 30 volts in increments of one volt. Because the motion of the rotors caused a cyclic vibration of the system, each scale would cycle through a range of values. At each voltage increment the range from maximum to minimum on both scales was recorded. It was assumed that the total lift generated by the system was the sum of the average of the values collected from each scale. The power required to generate the lift at each voltage interval is the voltage at that interval times the amperage that the control box read out. Rotor RPM data was also collected in increments of 5 volts from 10 volts to 30 volts. This was done by counting the number of full revolutions for a time of one minute. Rotor RPM for the other voltage intervals was interpolated; the Matlab code used to generate the interpolated values are provided in Appendix C.

Because of the inherent inefficiencies associated with running any electric gear motor, the power required to run the motor unloaded at each voltage interval was also collected. To collect the free operating power data the two screws that hold the motor in place and the timing belts under tension were loosened and the belts were removed from the system. The power required to run the motor alone was collected by recording the voltage and amperage required to run the motor from 10 volts to 30 volts in increments of one volt. The voltage and amperage at each increment is then multiplied for operating the motor alone and this power

is deducted from the total power required to operate the model at each interval. Subtracting these two powers gives the power required to operate the system based on drag and bearing friction alone.

The data collected from the testing is presented in Table 13 in Appendix B. From this data, three plots were generated using the Matlab code:
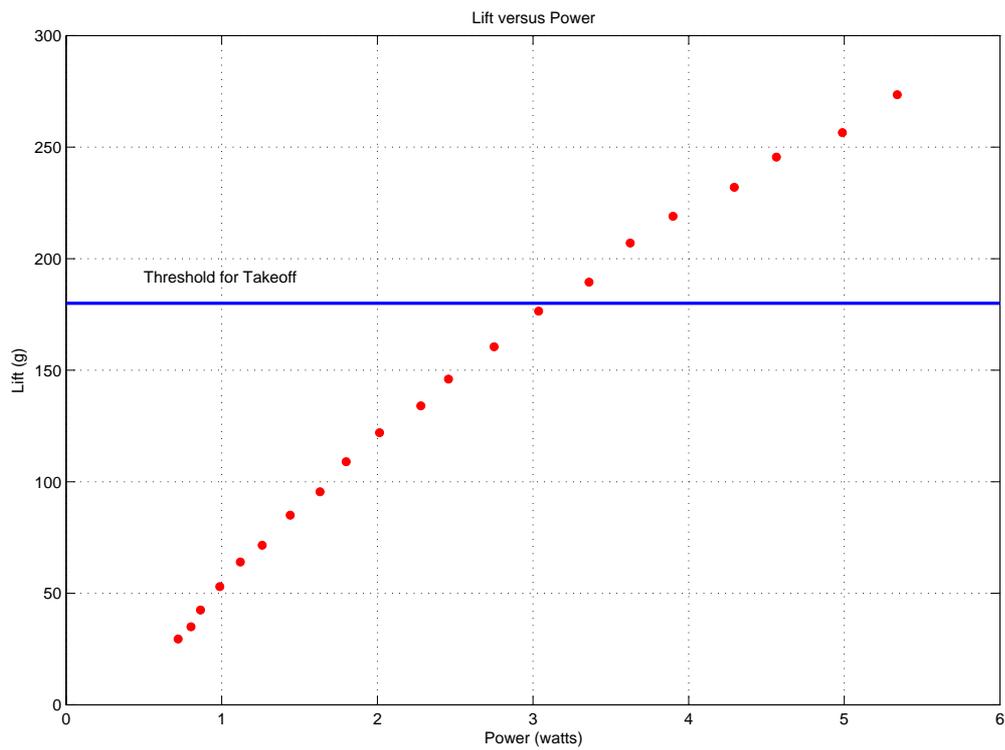
Figure 29: Model lift versus power.
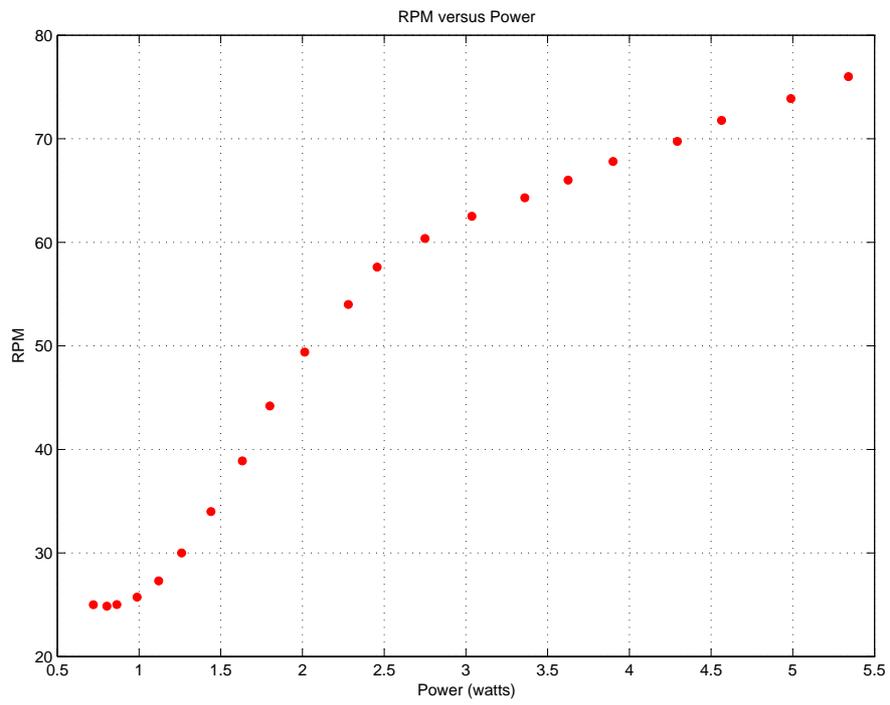
Figure 30: Model blade RPM versus power.
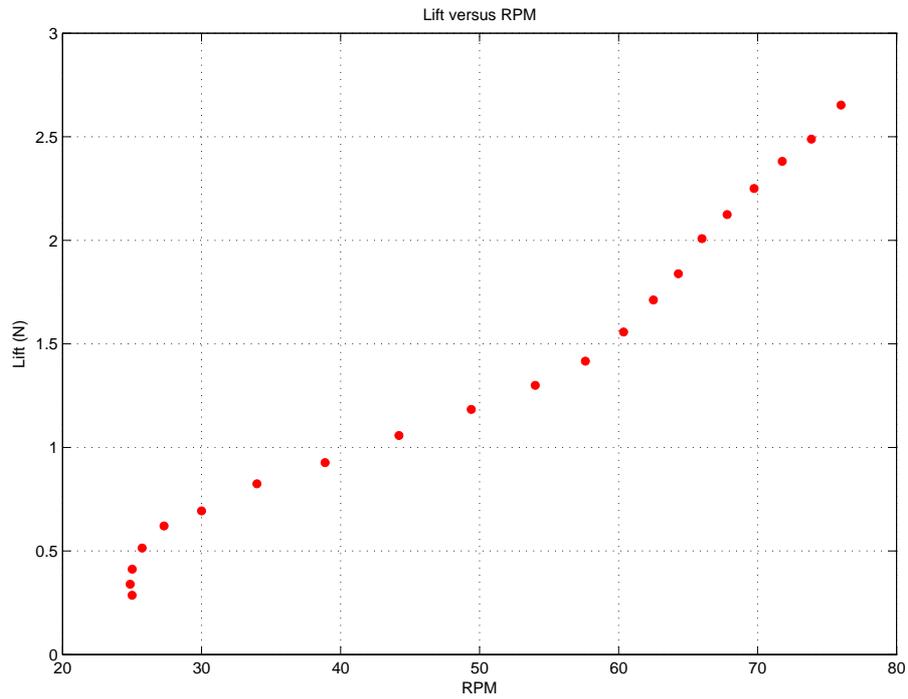
Figure 31: Model lift versus rpm.

Note that the scheme for interpolating RPM values is nonlinear in nature, which is the cause of the generally nonlinear plots involving RPM. From Figure 29, it is clear that the model continues to generate more than the required lift for takeoff far below the maximum power threshold as determined by the dynamic similarity analysis.
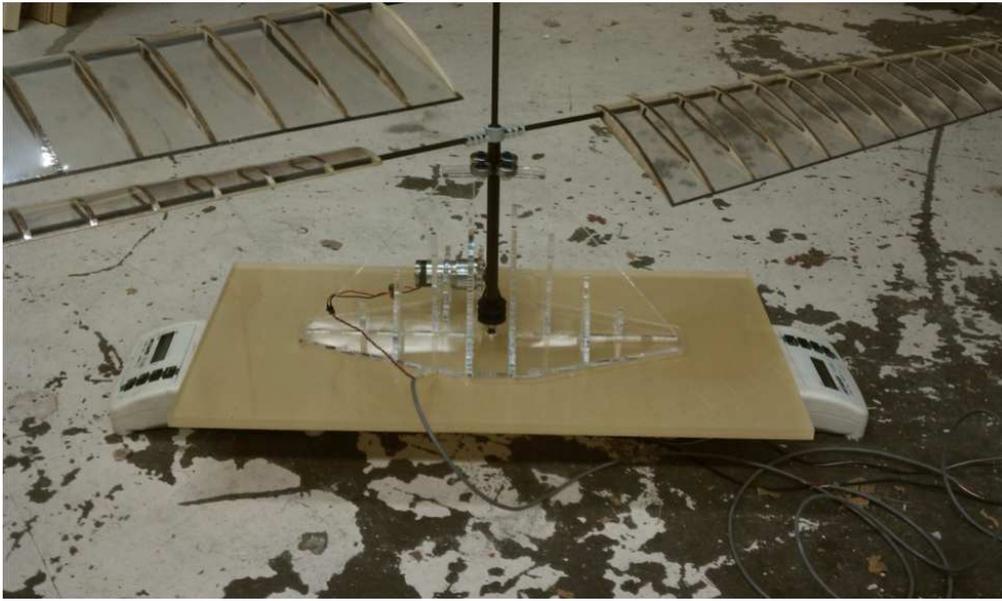
Figure 32: Another view of the finished model HPH sitting atop the two scales.

4 Conclusions and Future Work

This project has focused on building a dynamically scaled model of a human-powered helicopter to test the efficiency of a wing designed to generate lift at least equal to the scaled weight. Initial testing conducted on the model shows that there is enough lift to lift off a theoretical 0.38 lbs model helicopter, which is the weight that would be expected from the dynamic scaling process outlined in Section 2.1. From the scaling process it is also known that the maximum allotted power to be considered a success while lifting 0.38 lbs (173 g) is 11.16 watts. Some of the major results and accomplishments are listed below:

- Dynamic similarity was successfully used to determine model characteristics and requirements. Several iterations in determining these values were done before settling on the final design to be built.

- A new test stand was designed and built to collect lift and drag data. This test stand may be used in future design projects for collecting data at any Reynolds number, however, it was especially useful in this project for the low-Reynolds numbers the model was expected to experience.

- A blade element code was developed in Matlab to use the data collected with the new test stand to design an efficient model rotor blade geometry.

- A testing base and power supply was designed and built to test the lift generated by the model at various power outputs. The blade-shaft interface allows future projects to test newer designs by removing the current blades and attaching new ones.

- The data collected for the model shows that at a voltage of 23 V and an amperage draw of (0.21 A - 0.078 A), the power required was 3.036 watts, which provided the model with the theoretical lift of 0.38 lbs − well under the maximum allotted 11.16 watts.

- Therefore, the wing designed for this project is efficient enough to be considered for use in a full-scale prototype.

There remains some additional work to be done that can result in a better overall design. These include:

1. Design, make, and test more blade designs, incorporating the following features:

   - Aerodynamic twist.
   - Various taper geometries.

2. Design a test apparatus that is capable of measuring the effect that distance between the two rotor disks has on the total lift generated by the model.

3. Design a test apparatus that is capable of measuring the effect that the distance of the lower rotor disk to the ground has on the lift generated by the model.

4. Perform tests on the E-214 airfoil to confirm the data that is presented by the Airfoils at low speeds database.

5. Develop a model testing apparatus that addresses the issues with scale reading oscillations encountered in the final data collection.

6. Make considerations about the effects that the vertical bars of the test stand, and the wooden blade test stand interface has on the total drag of the wing.

7. Determine the amount of lift generated at the maximum threshold of 11.16 watts, and assess the feasibility of possible forward flight.

8. Conduct preliminary research on the feasibility of using composite materials on a full-scale prototype.

References

[1] *Henry Kremer and the Kremer Prizes*, 2009.
    http://www.raes.org.uk/cmspage.asp?cmsitemid=SG_hum_pow_kremer

[2] Curry, Marty. *Gossamer Albatross*. Dryden Flight Research Center, 2002.
    http://www.dfrc.nasa.gov/gallery/photo/Albatross/HTML/ECN-12665.html

[3] Lehoux, Gilles. *Dr. Paul MacCready*. HumanPoweredHelicopters.org, 2002.
    http://www.humanpoweredhelicopters.org/people/index.htm#Dr_Paul_MacCready

[4] Patterson, W. *The First Certified Human Powered Helicopter*. California
    Institute of Technology, 2002. http://www.calpoly.edu/∼wpatters/helo1.htm

[5] J. D. Anderson, *Fundamentals of Aerodynamics*, McGraw Hill, 2007.

[6] L. N. Sankar, *Blade Element Theory for Rotor in Hover and Climb*, 2008.
    http://soliton.ae.gatech.edu/people/lsankar/AE6070.Spring2008/Blade.Element.Theory.doc

[7] F. M. White, *Fluid Mechanics*, McGraw Hill, 2008.

[8] M. S. Selig, *Airfoils at Low Speeds*, H. A. Stokely, 1989.

[9] R. Cansdale, *An Aeroelastic Model Helicopter Rotor*, RAE Technical Report
    73042, 1974.

[10] P. P. Friedmann, *Aeroelastic Scaling for Rotary-Wing Aircraft with
     Applications*, Journal of Fluids and Structures, March 2004.

A   Calibrating and Collecting Data with the New Test Stand

This section provides a detailed description of how to calibrate and collect data using the custom built test stand. The first step is to calibrate for lift.

Figure 33: Basic test stand setup for collecting lift data.

1. Center the wooden table approximately below the opening on the bottom surface of the wind tunnel. Level the wooden table by adjusting the feet beneath each leg. The easiest way to do this is to level the edge marked in red below using the bubble level circled in red using only legs l and ll. Next level the edge marked in blue using the bubble level circled in blue using only leg lll. Extend the fourth leg (unmarked) until the table is stable. This procedure will ensure that the table is close to level.

57

Figure 34: Leveling the test stand table.

2. To measure lift, two scales are placed on the leveled table in the scale feet holes as pictured.

Figure 35: Scales in place for measuring lift.

3. The top of the test stand is then inserted through the bottom of the wind tunnel without the airfoil shaft interface attached. The base of the test stand is then placed on the two scales and the six indexing pins are extended about 0.25

below the base of the test stand. The test stand is then pushed to the right until the indexing pins contact the upper surface of the scales. The indexing pins ensure that the test stand sits in the same position on the scales every time. A picture below shows the test stand sitting on the scales with the indexing pins highlighted in red.

Figure 36: The red circles indicate screw locations to hold the test stand stationary for measuring lift.

4. Ensure that the blue screw highlighted in the image below is securely fastened.
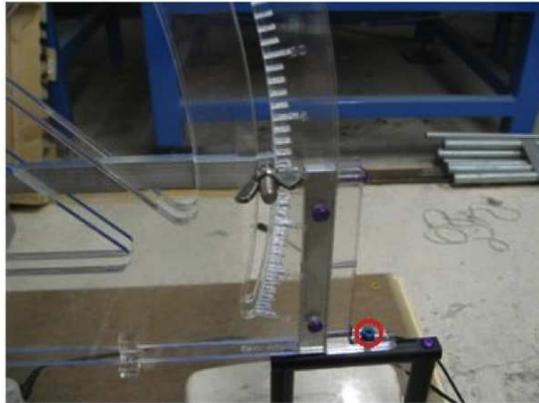
Figure 37: The blue screw secures the test stand and prevents it from tilting backward.

5. Attach the test airfoil to the airfoil upright interface. For low speed applications a 0.249 hole should be drilled in the airfoil or an adaptor (adaptor was made from a piece of pine for this project), and the airfoil should be pressed onto the aluminum airfoil test stand interface as shown below (press in as far as possible).

Figure 38: The test wing used in the experiment.

6. Attach the airfoil and the upright interface to the uprights using two 1/4-20 bolts as pictured below. Ensure that the span of the airfoil is parallel to the

bottom of the wing tunnel.

Figure 39: Close-up of the test wing fixed to the stand in the wind tunnel.

7. Move the table and stand as one unit until the uprights of the stand are visually centered through the access hole in the bottom of the wind tunnel as pictured below.
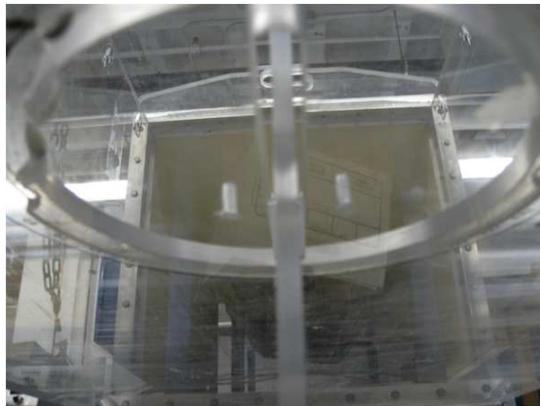
Figure 40: Looking up at the test stand going into the wind tunnel. It should not be in contact with either edge of the access hole.

8. Once the vertical arms of the stand are centered in both dimensions of the opening, the table should be re-leveled using the bubble level on the back of the

scale as the reference. Keep in mind that leveling the table may cause the vertical arms of the test stand to move out of center. This will require again moving the entire assembly and then re-leveling. Each iteration will bring the stand and the table closer to their optimal positions. The system is set properly when the table is level using the bubble level on the back of the scale as reference, and the vertical arms of the test stand are centered in the opening at the bottom of the wind tunnel. A picture of the bubble level on the back of the scale is included below.

Figure 41: Leveling device on the scale.

A.1 Calibration Data

9. Place the calibration ring stand and pulley in the position shown below.

Figure 42: Lift calibration setup with a 10g mass.

10. Once the calibration stand is set in the proper position disconnect the mass from the test stand. Turn on and tare the scales. Then re-connect the hanging mass to the test stand as pictured above, being careful not to move the calibration stand or bump the test stand itself. Record the readings on both scales (the reading on the left hand scale should be positive, and the readings on the right hand scale should be negative) along with the mass used, and the angle of attack.

11. Repeat steps 9 and 10 for each angle of attack that lift and drag data will be collected for.

As always with any experimental testing it is best to take each data point several times and take a numerical average. Numerical averages should be taken both

63

when collecting the actual data for each desired angle of attack and when collecting the calibration data.

12. Turn on the main power stitch to the wind tunnel and set the control box to the desired speed setting. For this testing, the wind tunnel was set to operate at 2.3 Hz, which correlated to a Reynolds number of about 10000. A picture of the control box set to the setting used throughout this experimentation is included below.

Figure 43: Control panel for setting wind tunnel speed.

13. Set the test stand to the desired angle of attack, and tare the scales (at this point both of the scales should read zero). Turn on the wind tunnel and wait for the readings on both scale to settle (this will take longer at higher angles of attack, if the number does not settle take a visual average). Record the data being sure to mark the scale (right or left) and the sign of the value.

14. Repeat step 13 for all of the desired angles of attack.

As always with any experimental testing it is best to take each data point several times and take a numerical average. Numerical averages should be taken both when collecting the actual data for each desired angle of attack and when collecting the calibration data.

15. Now both the calibration data, and the test data must be used together to calculate the actual lift forces experienced by the system. The equations below were derived using statics, and should be used to calculate the actual lift forces for each angle of attack.
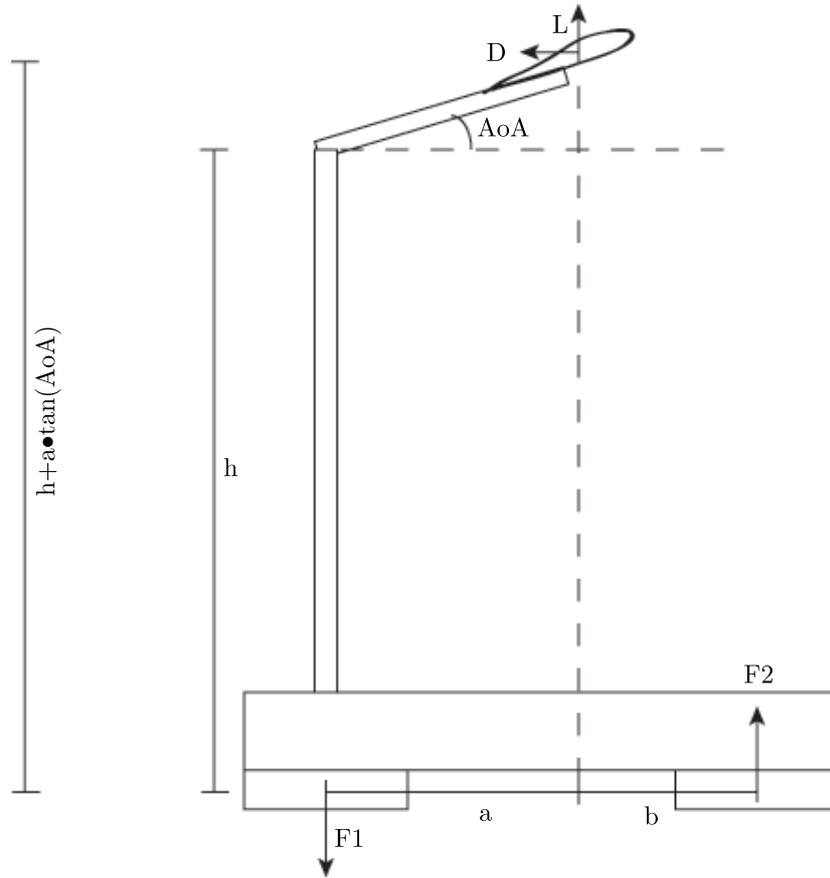
Figure 44: Schematic of forces acting on the test stand for measuring lift. L represents lift, D is drag, and F_1 and F_2 are the scale readings.

Before finding the lift forces, the parameters a and b must be found. These vary with angle of attack because they represent the location of the center of pressure on the wing, which is a function of angle of attack. Perform a moment balance around the center of pressure to obtain expressions for a and b:

$$\sum M_{x_{cp}} = F_1 a + F_2 b + D(h + a \tan \alpha) = 0 \tag{29}$$

If a new parameter C is defined such that $C = a + b$ (representing the distance between the two scales, which is always constant), then the above equation may be written as:

$$F_1 a + F_2(C - a) + D(h + a \tan \alpha) = 0 \qquad (30)$$

Solving for a, obtain:

$$a = \frac{-F_2 C - Dh}{F_1 - F_2 + D \tan \alpha} \qquad (31)$$

Performing a force balance in the x direction and a moment balance about point A, obtain the following equations:

$$\sum F_y = F_2 - F_1 + L = 0 \qquad (32)$$

$$\sum M_A = F_2 C + D(h + a \tan \alpha) + La \qquad (33)$$

Solving each equation for L, obtain:

$$L = F_1 - F_2 \qquad (34)$$

$$L = \frac{-F_2 C - D(h + a \tan \alpha)}{a} \qquad (35)$$

This suggests that there are two methods of calculating the lift force, however,

using the raw data in Appendix B, it was shown that each equation gives equivalent lift forces. Therefore, despite the fact that drag does affect the scale readings, it does so in such a way that its effect cancels out, and as a result, lift is measured independently of drag.

Upon collecting scale readings F_1 and F_2 and calibration data for a range of angles of attack, one can convert the raw data to lift by applying the calibration data:

$$L = \frac{F_1 - F_2}{\text{calibration factor}} \cdot \frac{4.4}{453.592} \tag{36}$$

The second fraction contains conversion factors that result in a lift force in newtons.

A.2 Measuring Drag

Figure 45: Basic test stand setup for collecting drag data.

16. Back out all of the indexing pins marked below in red until they no longer protrude from the base of the test stand. Entirely remove the indexing pin marked in blue (leave it on the table so as not to lose it).

Figure 46: The screws to be backed out are indicated in red and the one to be removed entirely is in blue.

17. Being careful not to move the table, remove the scales from below the test stand, and place the base of the test stand directly on the upper surface of the table.

18. Center the vertical arms of the test stand in both dimensions of the opening at the bottom of the wind tunnel as pictured below (this time the table is already level, so only the test stand should be moved to achieve centering).

Figure 47: As with lift measurements, make sure the test stand is not in contact with the edges of the access hole.

19. Remove the blue screw from the rear post of the test stand, as pictured below.

Figure 48: The test stand ready for a scale to be placed underneath.

20. Now that the blue screw is removed, the test stand is free to pivot on two bearings located on the forward posts. Holding the base of the test stand in position, pivot the upper section of the test stand and place a single scale under the upper section of the test stand as pictured below. Make sure that the scale

is centered lengthwise on the test stand base, and the back edge of the scale is pushed into the rear upright. (These conditions should allow the contact T to contact the center of the scale force plate).

Figure 49: A single scale is place under the test stand.

A.3 Calibration Data

21. Place the calibration ring stand and pulley in the position shown below.

Figure 50: Setup for drag calibration data with a 10g mass.

The calibration stand should be set such that the force imparted on the test wing by the hanging mass is horizontal with the flow of the tunnel, and in line with the two vertical members of the test stand. For these tests a mass of 10g was used.

22. Once the calibration stand is set in the proper position disconnect the mass from the test stand. Turn on and tare the scales. Then re-connect the hanging mass to the test stand as pictured above, being careful not to move the calibration stand or bump the test stand itself. Record the readings on the scale (the reading on the scale should be negative) along with the mass used, and the angle of attack.

23. Repeat steps 21 and 22 for each angle of attack that lift and drag data will be collected for.

24. Place the calibration ring stand and pulley in the position shown below.

Figure 51: Setup for additional lift calibration data.

The calibration stand should be set such that the force imparted on the test wing by the hanging mass is vertical. For these tests a mass of 10g was used.

25. Once the calibration stand is set in the proper position disconnect the mass from the test stand. Turn on and tare the scales. Then re-connect the hanging mass to the test stand as pictured above, being careful not to move the calibration stand or bump the test stand itself. Record the readings on both scales (the reading on the scale should be negative) along with the mass used, and the angle of attack.

26. Repeat steps 24 and 25 for each angle of attack that lift and drag data will be

collected for.

As always with any experimental testing it is best to take each data point several times and take a numerical average. Numerical averages should be taken both when collecting the actual data for each desired angle of attack and when collecting the calibration data.

27. Turn on the main power stitch to the wind tunnel and set the control box to the desired speed setting. For this testing, the wind tunnel was set to operate at 2.3 Hz, which correlated to a Reynolds number of about 10000. A picture of the control box set to the setting used throughout this experimentation is included below.

Figure 52: Control panel for setting the wind tunnel speed.

28. Set the test stand to the desired angle of attack, and tare the scales (at this point the scales should read zero). Turn on the wind tunnel and wait for the readings on the scale to settle (this will take longer at higher angles of attack, if the number does not settle take a visual average). Record the data being sure to

mark the sign of the value.

29. Repeat step 28 for all of the desired angles of attack.

As always with any experimental testing it is best to take each data point several times and take a numerical average. Numerical averages should be taken both when collecting the actual data for each desired angle of attack and when collecting the calibration data.

30. Now the calibration data, the test data, and the lift data must be used together to calculate the actual drag forces experienced by the system. The equations below were derived using statics, and should be used to calculate the actual drag forces for each angle of attack.
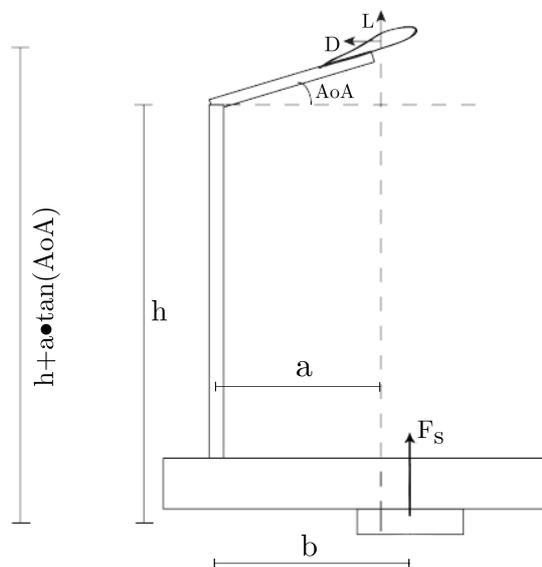
Figure 53: Schematic of forces acting on the test stand for measuring drag. L represents lift, D is drag, and F_s is the scale reading.

To obtain the drag force from the raw scale readings, perform a moment balance about point A:

$$\sum M_A = F_D(h \pm a \tan \alpha) + F_s(b) + L_{\text{cal}}(a) = 0 \tag{37}$$

where F_D is the uncorrected drag force, F_s is the scale reading, and L_{cal} is the calibrated lift force, and is found from

$$L_{\text{cal}} = \frac{L}{\text{calibration factor}} \tag{38}$$

Solving for F_D, obtain:

$$F_D = \frac{-F_s(b) - L_{\text{cal}}(a)}{h \pm a \tan \alpha} \tag{39}$$

To obtain the drag force in newtons, use the following equation:

$$F_{D_N} = (\text{drag calibration factor}) \cdot F_D \frac{4.4}{453.592} \tag{40}$$

B   Raw Lift Data

Below is a table of the raw lift data obtained from both scales for each initial test (using the original test stand) This data was not used in determining the final blade geometries, but is presented for completeness. Drag tests were not performed using the old test stand because it was not sensitive enough to read drag forces.

Table 6: First Lift Test: Original Test Stand

AoA (degrees)	Right Scale (g)	Left Scale (g)	Lift (g)
-5.0	0.4	0.0	0.4
-2.5	2.1	-2.3	-0.2
0.0	2.5	-3.9	4.4
2.5	3.0	-5.1	8.1
5.0	3.7	-6.5	10.2
7.5	4.7	-7.8	12.5
10.0	5.3	-8.8	14.1
12.5	6.4	-10.0	16.4
15.0	7.1	-11.0	18.1
17.5	8.0	-11.7	19.7
20.0	9.0	-12.7	21.7
22.5	10.2	-14.4	24.6
25.0	11.7	-16.3	28.0
27.5	11.9	-15.8	27.7
30.0	14.5	-19.0	33.5

Table 7: Second Lift Test: Original Test Stand

AoA (degrees)	Right Scale (g)	Left Scale (g)	Lift (g)
-5.0	1.7	-1.4	0.3
-2.5	2.1	-2.6	-0.5
0.0	2.5	-3.7	-1.2
2.5	3.2	-5.3	-2.1
5.0	3.4	-6.1	-2.7
7.5	4.2	-7.2	-3.0
10.0	5.0	-8.3	-3.3
12.5	6.1	-9.4	-3.3
15.0	6.7	-10.3	-3.6
17.5	7.3	-11.0	-3.7
20.0	9.2	-12.9	-3.7
22.5	9.8	-13.7	-3.9
25.0	10.5	-14.5	-4.0
27.5	12.9	-17.0	-4.1
30.0	14.8	-19.0	-4.2

The next set of tables shows the raw data collected using the new test stand that was actually used in calculating lift and drag forces for designing the rotor blade geometry. The first table below is the calibration data for lift.

Table 8: Lift Calibration Data for a 10g Mass: New Test Stand

AoA (degrees)	Left Scale (g)	Right Scale (g)	Difference (g)	Calibration Factor
-5.0	-3.6	-4.9	8.5	0.85
-2.5	-3.6	-5.0	8.6	0.86
0.0	-3.7	-5.1	8.8	0.88
2.5	-3.6	-5.0	8.6	0.86
5.0	-3.6	-4.9	8.5	0.85
7.5	-3.7	-5.2	8.9	0.89
10.0	-3.9	-5.0	8.9	0.89
12.5	-4.2	-4.8	9.0	0.90
15.0	-4.4	-4.6	9.0	0.90
17.5	-4.0	-4.6	8.6	0.86
20.0	-4.0	-4.6	8.6	0.86

The next table represents the raw data collected for lift measurements. The last column shows the lift, which is obtained by dividing the difference in scale readings by the calibration factor obtained from the above table for each angle of attack.

Table 9: First Lift Test: New Test Stand

AoA (degrees)	Left Scale (g)	Right Scale (g)	Difference (g)	Lift (g)
-5.0	2.8	-1.6	4.4	5.18
-2.5	2.0	-1.6	3.6	4.19
0.0	1.7	-2.3	4.0	4.55
2.5	1.6	-3.1	4.7	5.46
5.0	1.2	-3.5	4.7	5.53
7.5	1.3	-4.0	5.3	5.96
10.0	1.7	-5.0	6.7	7.53
12.5	2.1	-5.5	7.6	8.44
15.0	2.9	-6.5	9.4	10.44
17.5	3.4	-7.1	10.5	12.21
20.0	4.0	-7.7	11.1	12.91

The next table shows the raw data for the second lift test.

Table 10: Second Lift Test: New Test Stand

AoA (degrees)	Left Scale (g)	Right Scale (g)	Difference (g)	Lift (g)
-5.0	2.5	-1.3	3.8	4.47
-2.5	1.8	-1.5	3.3	3.84
0.0	1.6	-2.1	3.7	4.20
2.5	1.4	-2.8	4.2	4.88
5.0	1.1	-3.4	4.5	5.29
7.5	1.7	-4.4	6.1	6.85
10.0	1.8	-5.1	6.9	7.75
12.5	2.1	-5.5	7.6	8.44
15.0	2.7	-6.3	9.0	10.00
17.5	3.7	-7.0	10.4	12.09
20.0	4.2	-7.9	12.1	14.07

The next table shows the calibration data for drag measurements. Since lift is

coupled with drag in the measurements, the lift forces obtained from the above two tables must be recalibrated to find the correct force.

Table 11: Drag Calibration Data for a 10g Mass: New Test Stand

AoA (degrees)	Drag Calibration (g)	Lift Calibration (g)
-5.0	-38.1	-8.7
-2.5	-38.2	-8.6
0.0	-38.5	-8.6
2.5	-39.4	-8.6
5.0	-39.7	-8.5
7.5	-39.9	-8.5
10.0	-39.9	-8.4
12.5	-40.6	-8.4
15.0	-41.0	-8.3
17.5	-41.2	-8.3
20.0	-41.3	-8.3

The next table shows the raw data collected for drag measurements, according to the setup in Figure 45. They only represent what the scale actually reads, not the actual drag force, i.e., lift is coupled with the measurements. Lift and drag may be decoupled according to Equations 37 through 40.

Table 12: Raw Drag Measurement Data: New Test Stand

AoA (degrees)	Scale Reading: Test 1 (g)	Scale Reading: Test 2 (g)
-5.0	-1.1	-1.0
-2.5	-1.5	-1.5
0.0	-2.6	-2.7
2.5	-3.6	-3.6
5.0	-4.7	-4.7
7.5	-5.5	-5.6
10.0	-6.7	-6.7
12.5	-7.4	-7.4
15.0	-9.2	-8.5
17.5	-9.5	-9.5
20.0	-10.8	-10.8

The final table shows the results from testing the model. The two scale readings fluctuated considerably during operation, so a lower and upper bound is presented for each voltage setting. The RPM was measured at every five voltages, so the rest can be filled in by interpolation.

Table 13: Lift Generated During Model Testing

V	A (full setup)	A (model)	Scale 1 Low	Scale 1 High	Scale 2 Low	Scale 2 High	RPM
10	0.142	0.070	12	22	7	18	25
11	0.143	0.070	14	27	9	20	
12	0.144	0.072	17	33	10	25	
13	0.148	0.072	20	39	14	33	
14	0.154	0.074	25	47	17	39	
15	0.158	0.074	28	53	19	43	30
16	0.164	0.074	34	62	24	50	
17	0.171	0.075	38	70	28	55	
18	0.175	0.075	44	76	34	64	
19	0.181	0.075	52	83	39	70	
20	0.190	0.076	58	91	45	74	54
21	0.194	0.077	66	101	46	79	
22	0.203	0.078	71	109	52	89	
23	0.210	0.078	82	120	57	94	
24	0.219	0.079	87	128	62	102	
25	0.224	0.079	97	140	67	110	66
26	0.229	0.079	102	148	72	116	
27	0.238	0.079	107	155	80	122	
28	0.244	0.081	118	163	85	125	
29	0.253	0.081	124	170	89	130	
30	0.260	0.082	134	179	98	136	76

C Blade Element Code

Below is the Matlab code used to calculate the lift, drag, torque, and power requirements over each blade at each section along its span. All the scripts that initialize the raw data are presented, as well as the function codes for linear regression and interpolation. The first script is the main file to run in Matlab. It outputs the blade geometry, lift, drag, and power.

```matlab
1  clc;
2  close all;
3  clear all;
4
5  lift_data2;
6
7  %% Straight Taper on LE and TE
8
9  n = 4;            % number of blades
10 conv = 39.37;    % inches to meters
11
12 c_r = 6/conv;    % root chord
13 c_t = 3/conv;    % tip chord
14 Dr_m = 32/conv;  % blade radius
15 aoa_min = 5;     % min alpha
16 aoa_max = 20;    % max alpha
17
18 power_m = zeros(length(aoa_min),length(aoa_max));
19
20 for i = 1:length(aoa_min)
21     for j = 1:length(aoa_max)
22         [dx,y_low,y_up,LE,TE,chord_m,power_m(i,j),omega_m,...
23          lift,drag,w_m,vel,L,c_l3,x,root,S] = brute_force(c_r,...
24                 c_t,Dr_m,n,aoa_min(i),aoa_max(j),S_test);
25     end
26 end
27
28 % convert to inches
29
30 dx_in = conv*dx;
31 LE_in = conv*LE;
32 TE_in = conv*TE;
33 chord_m_in = conv*chord_m;
34
```

```matlab
35  if n*sum(lift) < w_m          % check if enough lift
36      disp('not enough lift')
37  else
38      disp('enough lift')
39  end
40
41  for i = 1:length(LE)     % table of blade coordinates
42      coordinates(i,:) = [i*2-2;chord_m_in(i);x(i)]';
43  end
44
45  %% Output
46
47  coordinates
48  spar_length = conv*(root + Dr_m)
49  w_m
50  tot_lift = n*sum(lift)
51  AR = Dr_m^2/S
52
53  row = [0 aoa_max];
54  table = [aoa_min' power_m];
55  table2 = [row;table]
56
57  %% Plotting
58
59  figure(1)
60  plot(aoa,avg_lift_N,'or','MarkerFaceColor','r','MarkerSize',8)
61  hold on
62  plot(aoa,avg_drag_N,'ob','MarkerFaceColor','b','MarkerSize',8)
63  title('Lift and Drag versus \alpha')
64  xlabel('\alpha (degrees)')
65  ylabel('Lift and Drag (N)')
66  legend('Lift','Drag','Location','NorthEastOutside')
67  grid
68
69  figure(2)
70  plot(dx_in,LE_in,'b',dx_in,TE_in,'r')
71  hold on
72  plot(dx_in,LE_in,'ok','MarkerFaceColor','k','MarkerSize',8)
73  plot(dx_in,TE_in,'ok','MarkerFaceColor','k','MarkerSize',8)
74
75  axis equal
76  title('Planform')
77  xlabel('Distance Along Span (in)')
78  ylabel('Distance Along Chord (in)')
79  legend('Leading Edge','Trailing Edge','Section Locations')
80  grid
81
82  figure(3)
83  plot(dx_in,lift,dx_in,drag,'r')
```

```
84  title('Lift and Drag versus dx')
85  xlabel('Distance Along Span (in)')
86  ylabel('Lift and Drag (N)')
87  grid
88
89  % c_r_in = c_r*39.37;
90  % c_r_pr_in = c_r_pr*39.37;
91  %
92  % c_t_in = c_t*39.37;
93  % c_t_pr_in = c_t_pr*39.37;
94  %
95  % % check areas
96  %
97  % A = (c_r + c_t)/2*Dr_m;
98  % A_sc = (c_r_pr + c_t_pr)/2*Dr_m;
```

The next function is called by the previous script in calculating lift, drag, power, and blade coordinates.

```
1   function [dx,y_low,y_up,LE,TE,chord_m,power_m,omega_m,lift,...
2             drag,w_m,vel,L,c_l3,x,root,S] = brute_force(c_r,c_t,...
3             Dr_m,n,aoa_min,aoa_max,S_test)
4
5   global C_L1 C_L2 a11 a12 a01 a02
6
7   %{
8       Must run 'lift_data' first to obtain average
9       lift coefficients from the two tests.
10  %}
11
12  %% Parameters
13
14  f       = 17;                  % number of blade elements to take
15  scale   = 0.08;                % scale of model to prototype
16  g       = 9.81;                % gravitational constant, m/s^2
17  rho     = 1.23;                % density, kg/m^3
18  mu      = 1.820755e-5;         % viscosity of air, kg/m-s
19
20  %% Prototype Characteristics
21
22  w_p     = 1023.09;             % weight, N
23  rpm_p   = 10;                  % angular velocity, rpm
24  omega_p = rpm_p*pi/30;         % angular velocity, rad/s
25  Dr_p    = 15.25;               % rotor radius, m
26  chord_p = 0.915;               % rotor chord, m
```

```matlab
27 A_p     = Dr_p*chord_p;        % prototype wing area, m^2
28 AR_p    = Dr_p/chord_p;        % prototype aspect ratio
29
30 %% Model Wing
31
32 A_m   = 2*scale^2*A_p/n;
33 w_m   = 0.28*9.81;          % estimated weight of model, N
34 % Dr_m  = sqrt(AR_p*A_m); % span, m
35
36 root = 4.5/39.37;  % distance from center of rotation to root, m
37 dx   = (0:Dr_m/(f-1):Dr_m)+root;  % section interval, m
38
39 y_up = c_r/2*sqrt(1 - (dx - root).^2/Dr_m^2) + c_r/2;
40 y_low = -c_r/2*sqrt(1 - (dx - root).^2/Dr_m^2) + c_r/2;
41
42 % below LE follows linear regression pattern for elliptic planform
43 % [slope,y_int] = LinearRegression(x,y_low);
44 % LE = slope*x + y_int;
45
46 % below LE is brute forced to provide a given tip chord
47 LE = (c_r/2 - c_t)*dx/Dr_m;
48 TE = y_up;
49 chord_m = TE - LE;
50
51 omega_m = omega_p*sqrt(scale)/scale;    % angular velocity, rad/s
52 % omega_m = 50*pi/30;
53
54 %% Model Characteristics
55
56 c_l = zeros(length(C_L1),1);
57
58 for i = 1:length(C_L1)
59     j = length(C_L1)+1-i;
60     c_l(j) = (C_L1(i) + C_L2(i))/2;
61 end
62
63 vel  = dx*omega_m;          % velocity at each section, m/s
64
65 %% Obtain lift coefficients for lift calculation
66
67 a1 = (a11 + a12)/2;     % average slope from regression curves
68 a0 = (a01 + a02)/2;     % average y-int. from regression curves
69
70 L = @ (x) a1*x + a0;                % lift force
71
72 if aoa_min == aoa_max       % range of aoa's
73     x = aoa_min*ones(1,f);
74 else
75     x = aoa_max:-(aoa_max - aoa_min)/(f - 1):aoa_min;
```

```matlab
76  end
77
78  L = L(x);                  % calculated lift forces along span
79
80  % wing area
81  A = pi*c_r*Dr_m/8;         % elliptical part
82  B = c_t*Dr_m;             % rectangular part
83  C = (c_r/2 - c_t)*Dr_m; % triangular part (≠ 0 only for tapered LE)
84  % S = A + B + C;         % wing area - sum of 3 parts
85  S = Dr_m*(c_t + c_r)/2;
86
87  tunnel = 2;                         % wind tunnel speed
88  c_l3 = (2*L./(rho*tunnel^2*S_test*chord_m))';   % c_l from force
89
90  %% Lift and drag calculations
91
92  lift = zeros(1,length(dx));
93  drag = zeros(1,length(dx));
94
95  for j = 1:length(dx)
96      lift(j) = 0.5*rho*vel(j)^2*... % lift per blade at each dx, N
97              (dx(2) - dx(1))*chord_m(j)*c_l3(j);
98      drag(j) = lift(j)/7;    % drag per blade at each dx, N
99  end
100
101 tot_lift = n*sum(lift);       % total lift over all blades, N
102 tot_drag = n*sum(drag);       % total drag over all blades, N
103
104 %% Torque and power
105
106 torque_m     = drag.*dx;          % torque per blade, N-m
107 tot_torque_m = sum(torque_m)*n;  % total torque, N-m
108 power_m      = tot_torque_m*omega_m; % total power requirement, W
109
110 end
```

Upon running the code, the output is displayed in the command workspace:

$$A_m = 0.0929 \text{ m}^2 \text{ (model blade area)}$$

$$b_m = 0.8128 \text{ m, model blade span}$$

$$c_r = 0.1524 \text{ m, root blade chord}$$

$$c_t = 0.0762 \text{ m, tip blade chord}$$

$$\omega_m = 3.7024 \text{ rad/s, model angular velocity}$$

$$\text{tot\_lift} = 7.9426 \text{ N, total lift}$$

$$\text{tot\_drag} = 1.1347 \text{ N, total drag}$$

$$\text{power\_m} = 2.8787 \text{ watts, power}$$

This next script generates lift coefficient versus angle of attack

```matlab
1  clc;
2  close all;
3  clear all;
4
5  global C_L1 C_L2 C_L5 a11 a12 a13 a14 a01 a02 a03 a04
6
7  %% Data
8
9  rho = 1.23;                        % density, kg/m^3
10 V = 2;                             % wind speed, m/s
11 S_eng = 3*20;                      % wing area in english units
12 S_test = S_eng*2.54^2/100^2;       % wing area, m^2
13
14 % aoa = pi/180*(-5:2.5:30);    % aoa (radians)
15 aoa = -5:2.5:20;
16
17 left_cal1  = [-3.6 -3.6 -3.7 -3.6 -3.6 -3.7 -3.9...
18             -4.2 -4.4 -4.0 -4.0];
19 right_cal1 = [-4.9 -5.0 -5.1 -5.0 -4.9 -5.2 -5.0...
20              -4.8 -4.6 -4.6 -4.6];
21 factor = abs(left_cal1 + right_cal1)/10;
22
```

```matlab
left1  = [ 2.8  2.0  1.7  1.6  1.2  1.3  1.7...
           2.1  2.9  3.4  4.0];
right1 = [-1.6 -1.6 -2.3 -3.1 -3.5 -4.0 -5.0...
            -5.5 -6.5 -7.1 -7.7];

lift1 = (left1 - right1)./factor;

left2  = [ 2.5  1.8  1.6  1.4  1.1  1.7  1.8...
           2.1  2.7  3.4  4.2];
right2 = [-1.3 -1.5 -2.1 -2.8 -3.4 -4.4 -5.1...
            -5.5 -6.3 -7.0 -7.9];

lift2 = (left2 - right2)./factor;

% drag readings

drag_cal = [-38.1 -38.2 -38.5 -39.4 -39.7 -39.9 -39.9...
             -40.6 -41.0 -41.2 -41.3];
lift_cal = [ -8.7  -8.6  -8.6  -8.6  -8.5  -8.5  -8.4...
            -8.4  -8.3  -8.3  -8.3];

scale1 = [-1.1 -1.5 -2.6 -3.6 -4.7 -5.5 -6.7...
          -7.4 -8.2 -9.5 -10.8];
scale2 = [-1.0 -1.5 -2.7 -3.6 -4.7 -5.6 -6.7...
          -7.4 -8.5 -9.5 -10.8];

avg_lift = (lift1 + lift2)/2;
avg_lift_cal = abs(lift_cal/10).*avg_lift;

f_d1 = zeros(length(aoa),1);
f_d2 = zeros(length(aoa),1);

for i = 1:length(aoa)
    if aoa(i) < 0
        f_d1(i) = (-scale1(i)*8.25 - avg_lift_cal(i)*...
                7.375*cosd(aoa(i)))/(33 - 7.375*...
        sind(aoa(i)));
        f_d2(i) = (-scale2(i)*8.25 - avg_lift_cal(i)*...
                7.375*cosd(aoa(i)))/(33 - 7.375*...
        sind(aoa(i)));
    else
        f_d1(i) = (-scale1(i)*8.25 - avg_lift_cal(i)*...
                7.375*cosd(aoa(i)))/(33 + 7.375*...
        sind(aoa(i)));
        f_d2(i) = (-scale2(i)*8.25 - avg_lift_cal(i)*...
                7.375*cosd(aoa(i)))/(33 + 7.375*...
        sind(aoa(i)));
    end
end
```

89

```matlab
72
73  drag1 = -(drag_cal/10).*f_d1';
74  drag2 = -(drag_cal/10).*f_d2';
75  avg_drag = (drag1 + drag2)/2;
76
77  % convert lift and  drag to N
78
79  lift1_N = lift1*4.4/453.592;    % convert from g to N
80  lift2_N = lift2*4.4/453.592;
81  avg_lift_N = avg_lift*4.4/453.592;
82
83  drag1_N = drag1*4.4/453.592;    % convert from g to N
84  drag2_N = drag2*4.4/453.592;
85  avg_drag_N = avg_drag*4.4/453.592;
86
87  %% Linear Regression
88
89  [a11,a01] = LinearRegression(aoa,lift1_N); % lift linear regression
90  [a12,a02] = LinearRegression(aoa,lift2_N);
91
92  [a13,a03] = LinearRegression(aoa,drag1_N); % drag linear regression
93  [a14,a04] = LinearRegression(aoa,drag2_N);
94
95  %% Lift and Drag Coefficients
96
97  C_L1 = (2*lift1_N/(rho*V^2*S_test))';    % find lift coeff.
98  C_L2 = (2*lift2_N/(rho*V^2*S_test))';
99  error_l = 100*(C_L1 - C_L2)./C_L2;       % percent error b/t tests
100
101 lift_table = [aoa' C_L1 C_L2 error_l];  % table of aoa and c_l's
102
103 C_D1 = (2*drag1_N/(rho*V^2*S_test))';    % find drag coeff.
104 C_D2 = (2*drag2_N/(rho*V^2*S_test))';
105 error_d = 100*(C_D1 - C_D2)./C_D2;       % percent error b/t tests
106
107 drag_table = [aoa' C_D1 C_D2 error_d];  % table of aoa and c_d's
108
109 % disp('Lift Coefficients')
110 % disp('      AoA     Test 1    Test 2  Percent Error')
111 % disp(table)
112
113 % x = pi/180*(-10:0.25:30);
114 x = -10:0.25:30;
115 curve1 = @(x) a11*x + a01;      % lin. regression values
116 curve2 = @(x) a12*x + a02;
117 curve3 = curve1(x);
118 curve4 = curve2(x);
119
120 C_L3 = (2*curve3/(rho*V^2*S_test))';
```

```
121  C_L4 = (2*curve4/(rho*V^2*S_test))';
122
123  C_L5 = (C_L3 + C_L4)/2;
124
125  avg_curve = (curve3 + curve4)/2;    % avg lift
126  avg_m = (a11 + a12)/2;  % avg slope of lin. regression
127  avg_b = (a01 + a02)/2;  % avg y-intercept of lin. regression
128
129  %% Aero Calculation
130
131  % angle of zero lift
132  aoa_zero = x(1) - avg_curve(1)/avg_m;
133
134  %% Plotting
135
136  % figure(1)          % plot results of 1st test
137  % plot(aoa,C_L1,'or','MarkerFaceColor','r','MarkerSize',5)
138  % hold on
139  % plot(x,C_L3,'b')
140  % % title('Test 1')
141  % xlabel('Angle of Attack (degrees)')
142  % ylabel('Coefficient of Lift')
143  % grid
144  %
145  % figure(2)          % plot results of 2nd test
146  % plot(aoa,C_L2,'or','MarkerFaceColor','r','MarkerSize',5)
147  % hold on
148  % plot(x,C_L4,'b')
149  % % title('Test 2')
150  % xlabel('Angle of Attack (degrees)')
151  % ylabel('Coefficient of Lift')
152  % grid
153  %
154  % figure(3)          % plot percent error
155  % plot(aoa,error,'or','MarkerFaceColor','r','MarkerSize',5)
156  % % title('Percent Error versus AoA')
157  % xlabel('Angle of Attack (degrees)')
158  % ylabel('Percent Error (%)')
159  % grid
```

This next function contains the raw data from the final model testing and generates graphs of lift versus power and RPM versus power.

```
1  clc;
2  close all;
```

```matlab
3   clear all;
4
5   %% Data
6
7   V = 10:30;
8
9   A_s = [0.142 0.143 0.144 0.148 0.154 0.158 0.164...
10          0.171 0.175 0.181 0.190 0.194 0.2030.210...
11          0.219 0.224 0.229 0.238 0.244 0.253 0.260];
12
13  A_m = [0.070 0.070 0.072 0.072 0.074 0.074 0.074...
14          0.075 0.075 0.075 0.076 0.077 0.078 0.078...
15          0.079 0.079 0.079 0.079 0.081 0.081 0.082];
16
17  power = V.*(A_s - A_m);
18
19  fr_l = [12 14 17 20 25 28 34 38 44 52 58 66 71 82...
20          87 97 102 107 118 124 134];
21  fr_u = [22 27 33 39 47 53 62 70 76 83 91 101 109 120...
22          128 140 148 155 163 170 179];
23  fr_avg = (fr_l + fr_u)/2;
24
25  ba_l = [7 9 10 14 17 19 24 28 34 39 45 46 52 57...
26          62 67 72 80 85 89 98];
27  ba_u = [18 20 25 33 39 43 50 55 64 70 74 79 89 94...
28          102 110 116 122 125 130 136];
29  ba_avg = (ba_l + ba_u)/2;
30
31  lift = (fr_avg + ba_avg);
32  lift_N = (fr_avg + ba_avg)*4.4/453.592;
33
34  %% Obtain Other RPM's
35
36  z = 1:5:21;
37  rpm = [25 30 54 66 76];
38  V_int = V(z);
39
40  rpm_spline = zeros(length(V),1);
41
42  for i = 1:length(V);
43      rpm_spline(i) = Spline(V_int,rpm,V(i));   % RPM using Spline
44  end
45
46  %% Plotting
47
48  x = 0:6;
49  y = 1.80 + 0*x;
50
51  figure(1)
```

```
52  plot(power,lift_N,'or','MarkerFaceColor','r','MarkerSize',5)
53  hold on
54  plot(x,y,'linewidth',2)
55  title('Lift versus Power')
56  xlabel('Power (watts)')
57  ylabel('Lift (g)')
58  text(0.5,1.90,'Threshold for Takeoff')
59  grid
60
61  figure(2)
62  plot(power,rpm_spline,'or','MarkerFaceColor','r','MarkerSize',5)
63  title('RPM versus Power')
64  xlabel('Power (watts)')
65  ylabel('RPM')
66  grid
```

This next function takes in the lift data and finds the regression curve.

```
1   function [a1,a0] = LinearRegression(x,y)
2
3   %{
4       This function uses the linear regression method to find
5       the constants a0 and a1 to perform curve fitting.
6
7       The linear regression curve is of the form y = a0*x + a1
8   %}
9
10  nx = length(x);         % length of x
11  ny = length(y);         % length of y
12
13  if nx ≠ ny              % check if x and y have the same lengths
14      disp('ERROR: The number of elements in x must be the same as in y.')
15      a1 = 'Error';
16      a0 = 'Error';
17  else
18      Sx = sum(x);                    % add x-values
19      Sy = sum(y);                    % add y-values
20      Sxy = sum(x.*y);                % add products of x and y
21      Sxx = sum(x.^2);                % add products of x with itself
22      a1 = (nx*Sxy - Sx*Sy)/(nx*Sxx - Sx^2);  % calculate a1
23      a0 = (Sxx*Sy - Sxy*Sx)/(nx*Sxx - Sx^2); % calculate a0
24  end
25
26  end
```

This next function uses a cubic spline interpolation scheme to calculate the

unmeasured RPM at various voltages.

```matlab
 1  function Y = Spline(x,y,X)
 2
 3  %{
 4      This function uses the natural cubic spline method to
 5          interpolate a set of data points.
 6      Input:
 7          x = vector of independent variable
 8          y = corresponding function values at each x
 9          X = value of independent variable at which
10              function is to perform interpolation
11      Output:
12          Y = function value at X where data was interpolated
13  %}
14
15  n = length(x);                              % length of x
16
17  %% Interval lengths
18
19  h = zeros(1,n-1);                % initialize interval
20
21  for i = 1:n-1
22      h(i) = x(i+1) - x(i);    % length of the ith interval
23  end
24
25  %% Establish system of linear equations
26
27  a = ones(1,n);                       % initialize 'a' vector
28  a(1) = 0;
29  a(n) = 0;
30
31  A = zeros(n-2);      % initialize coefficient matrix to compute 'a'
32  b = zeros(1,n-2);
33
34  for j = 1:n-2    % calculate vector used to find coefficients
35      b(j) = 6*((y(j+2) - y(j+1))/h(j+1)-...
36                  (y(j+1) - y(j))/h(j));
37  end
38
39  for k = 1:n-2
40      if k == 1
41          A(k,:) = [2*(h(k) + h(k+1))*a(k+1) h(k+1)*a(k+2) 0];
42      elseif k == n-2
43          A(k,:) = [0 h(k)*a(k) 2*(h(k) + h(k+1))*a(k+1)];
44      else
45          A(k,:) = [h(k)*a(k) 2*(h(k) + h(k+1))*a(k+1) h(k+1)*a(k+2)];
```

```matlab
46          end
47      end
48
49      c = (A\b')';              % solve for non-zero coefficients
50
51      for m = 2:n-1
52          a(m) = c(m-1);            % reassign a-values of 1 to be c-values
53      end
54
55      %% Function values
56
57      f = zeros(n-1);             % initialize function coefficient matrix
58
59      for p = 1:n-1      % fill in coefficient matrix with known values
60          f(p,:) = [a(p)/(6*h(p)) a(p+1)/(6*h(p)) (y(p)/h(p) - ...
61                   a(p)*h(p)/6) (y(p+1)/h(p) - a(p+1)*h(p)/6)];
62      end
63
64      for q = 2:n
65          if X < x(q)           % determine which interval X lies in
66              break
67          end
68      end
69
70      Y = f(q-1,1)*(x(q)-X)^3 + f(q-1,2)*(X-x(q-1))^3 + ...
71          f(q-1,3)*(x(q)-X)   + f(q-1,4)*(X-x(q-1));      % calculate Y
72
73  end
```