Optimization of a Micro Aerial Vehicle Planform Using Genetic Algorithms

by

Andrew Hunter Day

A Thesis

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Mechanical Engineering

by

_____

May 2007

Approved:

_____

Professor David J. Olinger, Thesis Advisor

_____

Professor Nikolaos A. Gatsonis, Committee Member

_____

Professor Zhikun Hou, Committee Member

_____

Professor Mark W. Richman, Graduate Committee Representative

Abstract

Micro aerial vehicles (MAV) are small remotely piloted or autonomous aircraft. Wingspans of MAVs can be as small as six inches to allow MAV's to avoid detection during reconnaissance missions. Improving the aerodynamic efficiency of MAV's by increasing the lift to drag ratio could lead to increased MAV range and endurance or future decreases in aircraft size. In this project, biologically inspired flight is used as a framework to improve MAV performance since MAV's operate in a similar flight regime to birds. A novel wind tunnel apparatus was constructed that allows the planform shape of a MAV wing to be easily altered. The scale-model wing mimics a bird wing by using variable feather lengths to vary the wing planform shape. Genetic algorithms that use natural selection as an optimization process were applied to establish successive populations of candidate wing shapes. These wing shapes were tested in the wind tunnel where wings with higher fitness values were allowed to 'breed' and create a next generation of wings. After numerous generations were tested an acceptably strong solution was found that yielded a lift to drag ratio of 3.28. This planform was a non conventional planform that further emphasized the ability of a genetic algorithm to find a novel solution to a complex problem. Performance of the best planform was compared to previously published data for conventional MAV planform shapes. Results of this comparison show that while the highest lift to drag ratio found from the genetic algorithm is lower than published data, inabilities of the test wing to accurately represent a flat plate Zimmerman planform and limitations of the test setup can account for these discrepancies.

Acknowledgements

There are many other people that deserve recognition for this research paper. First and foremost I would like to thank Professor David Olinger. His idea of using genetic algorithms as a means of micro aerial vehicle planform optimization was both novel and inspiring. His continued support through the both my undergraduate and graduate career has made working on this project a valuable and enjoyable experience.

I would also like to thank Neil Whitehouse in his continued support in helping fabricate the test wing and the wing fixture. The logistics of creating a test wing with a large variation of planforms proved a formidable task. Modifying the test fixture in order to be sensitive enough to accurately obtain lift and drag forces of such small magnitudes was also difficult. In this regard I would like to thank Scott Blanchard, David DeFusco, and Christ Donoghue, for pioneering the force balance setup used in this paper, so rather than reinvent the wheel I could alter a preexisting setup.

I would like to thank the Mechanical Engineering Department for giving me the opportunity to pursue my graduate career, and the professors who made my growth and at this institution possible. Additionally, I would like to thank my thesis committee for their time and effort editing my paper and attending my thesis defense. I would also like to thank the secretarial staff in the department, specifically Barbara Edilberti for helping to keep me on track and providing much needed humor along the way.

Last but surely not least I would like to thank my friends and family for their unrelenting support in my education.

Table of Contents

List of Figures

List of Tables

Nomenclature

AR =aspect ratio
b =wingspan
C_D =drag coefficient
C_{Di} =induced drag coefficient
C_{D0} =parasitic drag coefficient
C_l =lift coefficient
c =wing chord
c_{root} =root chordlength
D =drag force
E =aircraft endurance
e_0 = Oswald efficiency factor
$f(X_i)$ =fitness value (fitness function evaluated for chromosome X_i)
g =generation size
h =gene number
K_0 = C_D vs. C_l^2 slope
k_m =breeding crossover point m
L =lift force
$\frac{L}{D}$ =lift to drag ratio
$\frac{L}{D}_{high}$ =high $\frac{L}{D}$ for error bars
$\frac{L}{D}_{low}$ =low $\frac{L}{D}$ for error bars
l =number of parents
N =random number threshold
n =generation number
$P(X_i)$ =parents selection probability
R =aircraft range
Re =Reynolds number
S =planform area
s =drag slope
V =aircraft cruising velocity
V_f =voltage at test speed
V_i =voltage at zero flow
W_f =aircraft final weight
W_i =aircraft takeoff weight

X_i = chromosome i

X_{ij} = gene j of chromosome i

$X_i^{Fitness}$ =fitness value of chromosome

x_j = gene j

z =spanwise distance

α =angle of attack

θ_1, θ_2 =drag calibration angles

σ_I =standard deviation of instrumentation error

σ_R =standard deviation of repeatability error

σ_{Total} =standard deviation of total error

Chapter 1: Introduction

1.1 Genetic Algorithms

1.1.1 Introduction

In recent years genetic algorithms have emerged as exciting new optimization tools. Genetic algorithms use evolutionary theory to solve engineering design problems. In this research, genetic algorithms will be used to optimize the planform shape of a micro aerial vehicle wing. Before continuing, an explanation of the genetic algorithm and its application are presented.

1.1.2 Evolution

Natural evolution is the process by which a biological organism changes over time. Usually these changes adapt the organism to its surroundings. Charles Darwin, considered the father of evolutionary theory, stated "In the struggle for survival, the fittest win out at the expense of their rivals because they succeed in adapting themselves best to their environment."[1] These advantageous traits come about through random mutations in subsequent generations. Similarly, disadvantageous traits may also arise. Those with the advantageous traits are more likely to survive and pass these traits on to future generations. Those with the disadvantageous traits will find it more difficult to survive and are less likely to pass their traits on to future generations. Over time, the species evolves as more and more advantageous traits are passed on[1].

1.1.3 History of Genetic Algorithms

Although evolutionary theory was developed in the mid nineteenth century it provided little practical application in engineering for about one hundred years. In the late 1950's biologists first attempted to mimic natural evolution using computer simulation. In

the mid 1960's John H. Holland proposed using evolutionary theory to solve various types of optimization problems[2]. Over the next decade Holland would work on this technique, which he called “genetic algorithms”, before he would publish Reference 3 in 1975. In Reference 3 a number of different methods using evolutionary based computer programming to find optimal solutions to complex problems are explored. In short, a genetic algorithm takes an initial population of random possible solutions and determines which of these possible solutions are the strongest of the population using a fitness function. These individuals are “bred” and produce offspring that are again evaluated to distinguish strong from weak solutions. In each successive generation, new parents are selected and offspring are produced. This process continues until the process converges on an acceptably strong solution[4]. The steps of the genetic algorithm are depicted in Figure 1.

The genetic algorithm, when used properly, virtually guarantees finding the global maximum or minimum. Ironically, an automated genetic algorithm optimization program could take mere minutes to perform whereas natural evolution’s process has been stretched out over thousands or millions of years. Rao[5], accurately describes genetic algorithms as “evolution at warp speed”. In an article written bearing this same title, Rao describes the difficult task of maintaining appropriate inventory for a large car company. Due to the varying demands of the public and certain constraints of the manufacturing plant, genetic algorithms were incorporated to bring order and efficiency to a scheduling process that was before based on experienced intuition.

A more detailed discussion of how genetic algorithms function follows next.

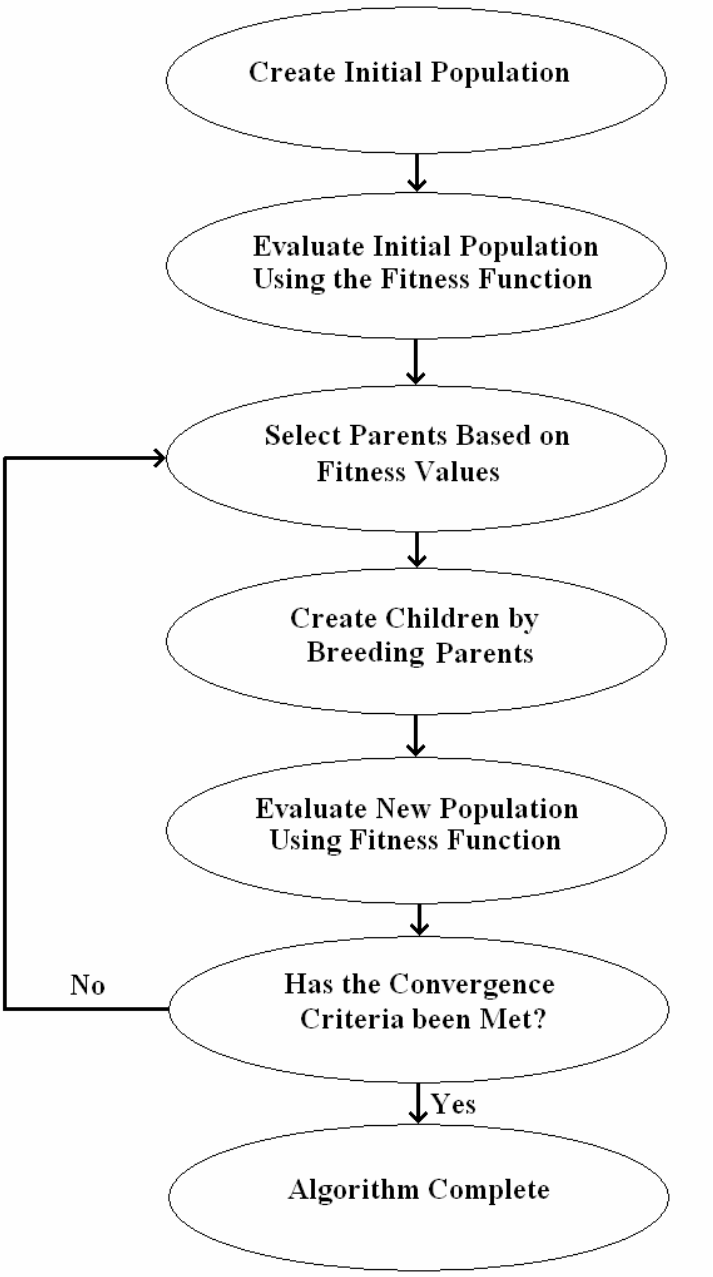

Figure 1. Genetic Algorithm Flowchart

1.1.4 The Chromosome and the Fitness Function

The basic genetic algorithm uses a string of numbers that is analogous to a biological chromosome. Each of the positions in the chromosome is termed a gene. In the basic model of genetic algorithms these genes are binary, either a one or a zero, representing a characteristic that is either present or not. However, the genes are not

always binary. There could also be many possible values for a particular gene, each called an allele[4].

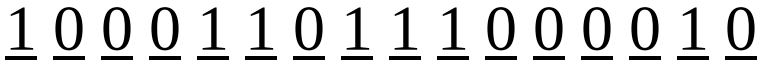

Figure 2. Sample Binary Chromosome

The sixteen genes in the sample binary chromosome in Figure 2 represent a unique individual. Each individual is different, and when asked to perform a task they do so with varying results. For the purposes of this research this task will be the lift to drag ratio of a MAV wing. However for other design problem the task may be different. For most engineering applications the task is increasing efficiency to reduce overall costs. This could mean maximizing fuel efficiency, minimizing the amount of materials used, minimizing labor and operating costs, maximizing a vehicle range, minimizing noise production or one of many things. Regardless of the task, there are potential solutions. The effectiveness of each solution may vary. The measure of how well a potential solution accomplishes its intended task is termed the fitness function[4]. This function can be explicit, written in terms variables (genes). Often if this is the case, the genetic algorithm is run computationally relatively quickly. The function can also be laboratory based. In this case the physics are not well understood enough to devise a representative equation in order to maximize it. The fitness function is therefore experimentally measured and then entered back into the genetic algorithm. In the present work the lift to drag ratio of an MAV wing with a specified planform shape serves as the fitness function.

In mathematical terms let the vector X_i represent any chromosome i with h number of x_j genes, where j simply denotes any specific gene in the chromosome.

$$X_i = [x_1, x_2, ..., x_h] \tag{1}$$

Then, the fitness value of the chromosome is simply the evaluation of that chromosome in the fitness function f .

$$X_i^{Fitness} = f\left(X_i\right) \tag{2}$$

If you were to evaluate every possible solution in order to find the optimal solution you would have 2^{16} (over 65,000) possible solutions for the 16 bit chromosome shown in Figure 2. The number of possible solutions is called the solution space or the search space[2]. Some more complex problems could have hundreds, or potentially thousands of bits. One can see that evaluating every solution in the search space may prove to be prohibitively exhaustive, in both time and money spent. The advantage of the genetic algorithms is that it only requires the evaluation of a small fraction of the entire solution space to find optimal solution[4].

1.1.5 The Initial Population

To initiate a genetic algorithm an initial population of individuals, much smaller than the size of the solutions space, is randomly created. The size of the initial population depends on the size of the solution space. Larger solution spaces require larger initial populations. If the initial population is too large, the additional time necessary to evaluate the extra individuals in each population may be inefficient. However, if the initial population is too small, the global optimum may be overlooked, providing a major setback to the optimization process[4].

To illustrate, a plot of Rastrigin's function is shown in Figure 3 as a sample solution space. One can see the presence of many local maxima and minima. Because of this, Rastrigin's function is widely used as a benchmark solution space when testing optimization programs due to the presence of many local maxima and minima[6]. However,

this function has one global minimum, which occurs at the (0, 0) position. If one seeks to identify the global minimum using a genetic algorithm the size of the initial population must be carefully specified. If the initial population is too small, the individuals may be dispersed in such a way that there are no individuals near the global minimum. Therefore, as the genetic algorithm progresses it is possible that a local minimum is converged upon rather than the global minimum.

$$R(x, y) = 20 + x^2 + y^2 - 10\left(\cos(2\pi x) + \cos(2\pi y)\right) \qquad (3)$$

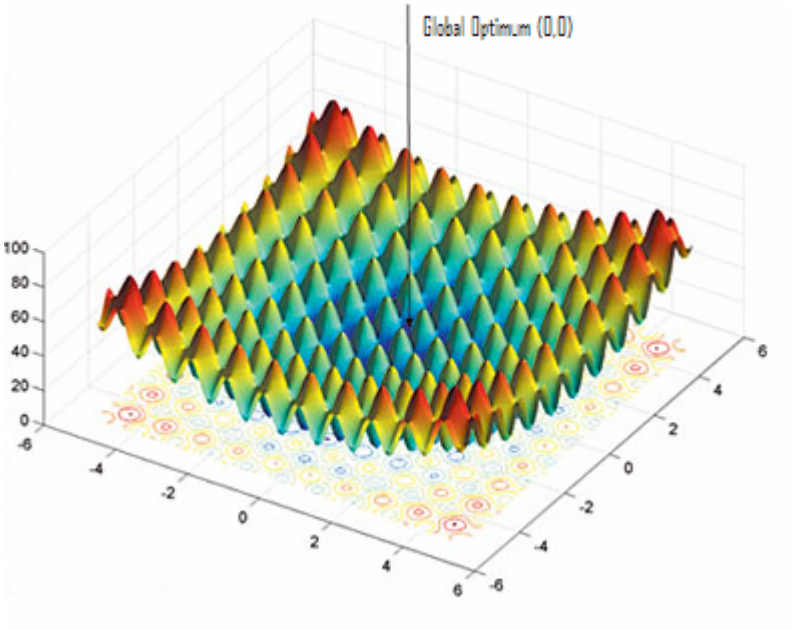

Figure 3. Rastrigin's Function[6]

One method of avoiding this risk is to create an initial population that is significantly larger than future generations. This ensures that a large enough sample has been taken to confidently survey the solution space for the global optimum[4]. The global optimum cannot be guaranteed, but in general it is better to take a larger than necessary

sample and take longer to find the better solution than to take too small a sample and prematurely converge on a local optimum instead of the global optimum.

There are no strict guidelines as to how large your initial population should be, or the size of the subsequent generations, but a general guideline says the following generations should have at least as many individuals as there are genes in a binary chromosome[2]. So, for the example in Figure 2, the population would have a minimum of sixteen individuals. The initial population may have two or three times that. In a later section, the selection of initial and subsequent generation size for the micro aerial vehicle planform optimization will be discussed.

1.1.6 Parent Selection

Once the initial population is created, its individuals are evaluated using the fitness function.

$$X_i^{Fitness} = f\left(X_i\right) \tag{4}$$

Next it is necessary to discard part of the population to allow for vacant chromosome locations in the next generation. The individuals that are not discarded become the parents of the next generation. Typically, from one quarter to half of the population are selected as parents, such that $0.25g \le l \le 0.50g$, where l is the number of parents and g is the number of individuals in the population. In the initial population however, the fraction of parents to available individuals may be smaller. These parents may or may not be duplicated to appear in the next generation. Although not strictly representative of biological reproduction, the logic behind copying the parents into the next generation is simply the reluctance to dispose of a good solution, especially if the

children produced do not show any improvement over their parents[4]. This technique will be used in the current work.

There are various methods of selecting the parents of the next generation. The first and most simple of the parent selection processes is elitism, which takes the l strongest individuals to use as the parents. This technique converges comparatively quickly, but may be slightly less likely to find a global optimum in certain circumstances. The reason for this is that sometimes an individual may have a portion of genes that are very beneficial, but overall the solution yields a low fitness value. Not selecting such an individual as a parent would eliminate the chance that these beneficial genes combining with other potentially beneficial genes to produce an otherwise overlooked strong solution[4].

There are several methods that attempt to remedy this. The roulette method assigns selection probabilities in proportion to the fitness level of an individual much like on an assumed roulette wheel. Individuals with higher fitness values get proportionally larger slices while individuals with lower fitness scores get proportionally smaller slices[4]. It is then possible, although less likely, for a weaker individual to be selected as a parent. Table 1 contains ten individuals with artificially created fitness values. The selection probability is given by:

$$P(X_i) = \frac{f(X_i)}{\sum_{i=1}^{n} f(X_i)} \tag{5}$$

These selection probabilities are used to construct a sample roulette wheel found in Figure 4. Clearly the first chromosome has the highest chance (18%) of being selected whereas the tenth chromosome has the least chance (2%). Due to its more random nature

and ease of application the roulette method was chosen as the parent selection method in this work.

Table 1. Roulette Wheel Example Population

Chromosome	1	2	3	4	5	6	7	8	9	10	sum
Fitness Value	2	1.8	1.6	1.4	1.2	1	0.8	0.6	0.4	0.2	11
Selection Probability	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.05	0.04	0.02	1

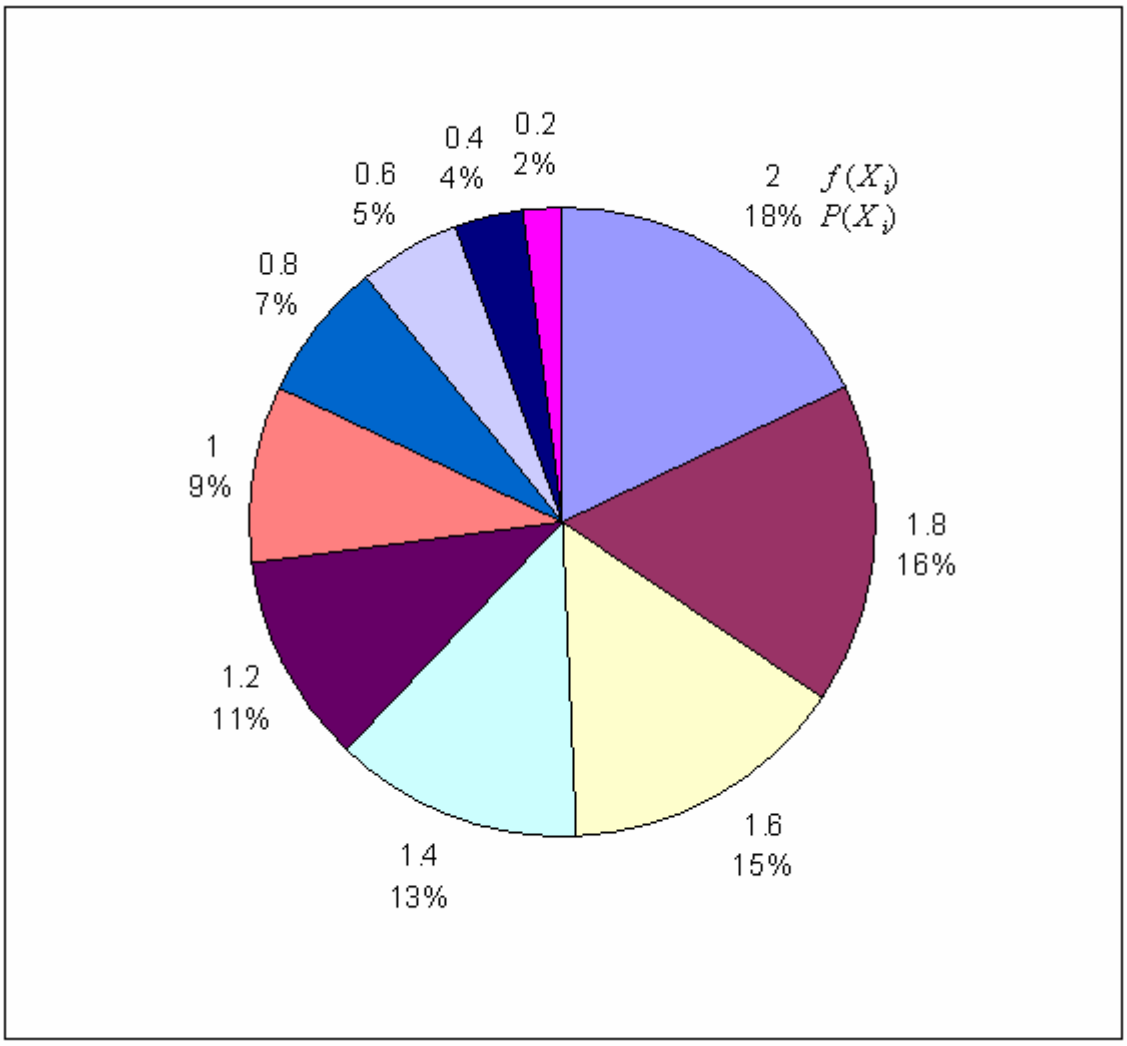

Figure 4. Sample Roulette Wheel

Stochastic uniform sampling is similar to the roulette method in that stronger individuals have a higher chance of being selected than weaker individuals. To establish this method individuals are spread along a line in order from most fit to least fit as shown

in Figure 5. The stronger individuals have proportionally bigger line segments. A random number is generated that falls in the first individual's segment. Then, the total length of the line, which can be normalized to one as it is in this example, is divided by the desired number of parents l to find the incremental length ($1/l$) between pointers that are established. If one of these pointers falls in an individual's line segment it is selected as a parent[4]. A representation of this procedure is also shown below in Figure 5 in which parents 1, 2, 3, 4, 6, and 8, are chosen.

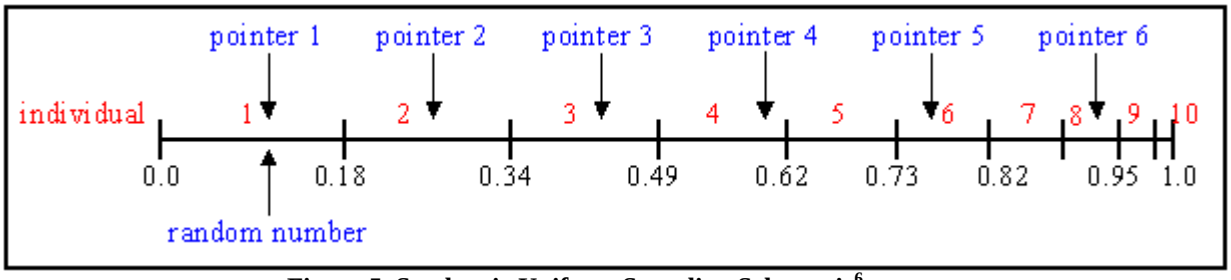

Figure 5. Stochastic Uniform Sampling Schematic[6]

This method guarantees the individual with the highest fitness function is selected. Due to the less random nature, stochastic uniform sampling was not used in the current work.

A less common method is a tournament style selection process in which subgroups are randomly constructed. These subgroups then compete amongst themselves to determine which individual from each subgroup is selected as a parent. The individual with the highest fitness value wins the tournament for each subgroup and is therefore selected as a parent for the next generation[4]. Due to the additional complexity of the subgrouping this method was not selected.

1.1.7 Pairing

Once the parents have been selected, it is necessary to couple the parents before any offspring can be produced. There are several methods to accomplish this pairing. The

first method is sequential pairing, where the first and second parents mate, the third and fourth mate and so forth[4]. Although this is the simplest technique to implement, its predictive structure may bias the results depending on the parent selection method chosen.

Another method is random pairing, where the parents of each pair are chosen at random[4]. Although this method may prevent the bias that may be present in the sequential pairing method, it also brings the possibility that a selected parent will not be paired and will therefore not mate. This parent's information will not necessarily be lost because it will remain present in the next population, as will all the individuals that are selected as parents. However, there is no guarantee that the individual will be selected as a parent in subsequent generations.

In this work sequential pairing is used. A potential bias is avoided by having the parents listed in random order. Listing the parents in random order allows one to sequentially pair the parents and still have a random pairing. Randomly pairing a list of randomly paired numbers would be redundant, so it is not done. Additionally, every individual that was selected as a parent mates using this method. This prevents each individual from having to pass two selection processes in order to mate and also keeps a more diversified population.

1.1.8 Mating

Once the parents are paired they must mate in order to form the next generation of chromosomes. In this mating process, genes are swapped between the parents in order to form new individuals. There are several different mating methods, the most basic of which is the single point crossover shown in Figure 6. This technique divides two

chromosomes at the same randomly chosen gene location k. Then genes x_1 - x_k are selected from the first parent while genes x_{k+1} - x_n are taken from the second parent. Combining these two sets of genes yields the child chromosome[4]. Another child, using the remaining two sections of genes, may also be formed.

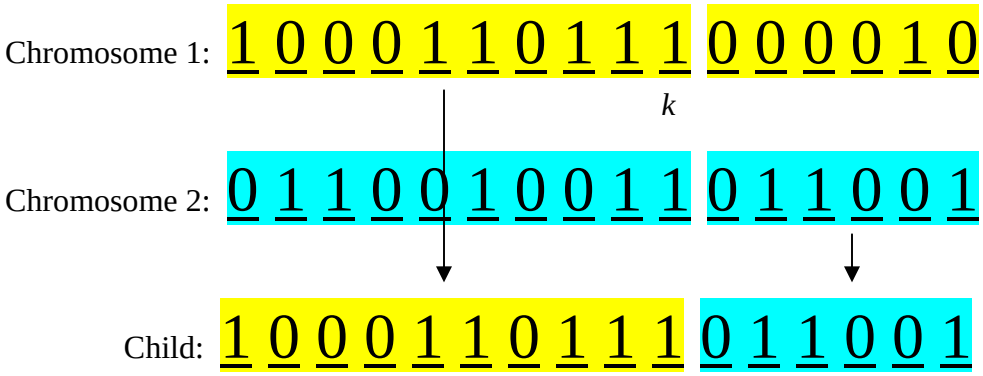

Figure 6. Single Point Crossover

Mathematically, let X_{ij} represent the gene j of individual i. If X_{1j} and X_{2j} represent the first and second chromosomes in Figure 6 respectively then the single point crossover of two individuals at gene k can be represented as:

$$X_{3j} = X_{1j} \text{ if } \quad j \le k$$

$$X_{3j} = X_{2j} \text{ if } \quad j > k$$

Where X_{3j} represents the child created from the two parents

Multipoint crossover is an extension of the simpler single point crossover. In this technique there can be any number of crossover points ($k_1, k_2 \ldots k_m$)[4]. This multipoint crossover can give more variation to future generations. An example of double point crossover where $m = 2$ is shown in Figure 7.

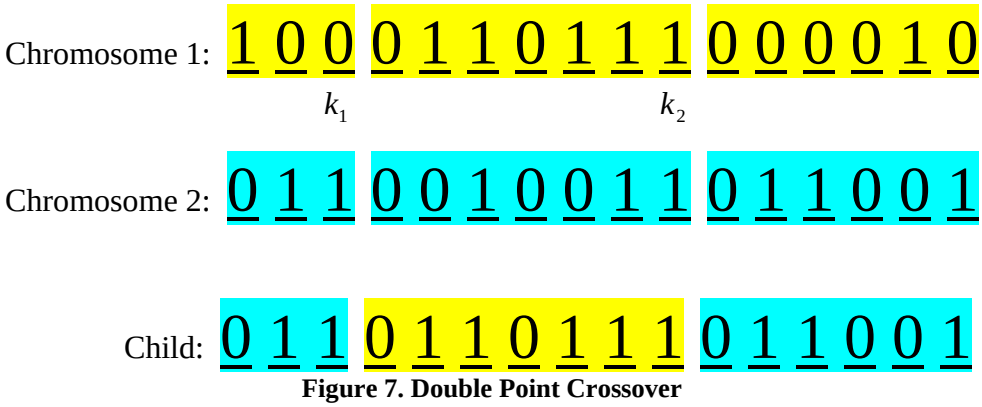

Figure 7. Double Point Crossover

This double point crossover process can be depicted mathematically according to the following:

$$X_{3j} = X_{2j} \text{ if } \quad j \le k_1 \text{ or if } \quad j > k_3$$

$$X_{3j} = X_{1j} \text{ if } \quad k_2 < j \le k_3$$

Unfortunately, this simple crossover method is not always optimum. One can see that in single point crossover the genes on the right side of the parent chromosomes and the genes on the left side of the parent chromosomes are usually separated. This results in a child where the left side of the chromosome and the right side are almost always from different parents. In double point crossover they are almost always from the same parent.

A different approach to mating solves this sectionalizing bias that occurs with point crossovers. This technique randomly selects which parent will provide each of the genes by creating a random number N associated for each gene. If the number is above a certain threshold then one parent is selected to provide the information for that particular gene. If not, the other parent is selected[4]. If random numbers between 0 to 1 are chosen, the obvious threshold is $N = 1/2$.

In Table 2 random numbers have been generated for each position to determine which parent will contribute each gene.

Table 2. Random Parent Selection for Each Gene

Gene	Random Number	Parent Chosen
1	0.5733	2
2	0.7710	2
3	0.6426	2
4	0.7719	2
5	0.1687	1
6	0.2563	1
7	0.7987	2
8	0.8203	2
9	0.2665	1
10	0.3790	1
11	0.6755	2
12	0.4759	1
13	0.4677	1
14	0.5871	2
15	0.7414	2
16	0.0466	1

Chromosome 1: 1 0 0 0 1 1 0 1 1 1 0 0 0 0 1 0

Chromosome 2: 0 1 1 0 0 1 0 0 1 1 0 1 1 0 0 1

Child: 0 1 1 0 1 1 0 0 1 1 0 0 0 0 0 0

Figure 8. Resulting Child from Mating of Random Parent Selection for Each Gene

Figure 8 shows the resultant child from the random mating method. This method of mating is more representative of biological reproduction and provides an opportunity for additional variation in the subsequent generations of chromosomes and was therefore the chosen method of mating in this work.

1.1.9 Mutation

Another mating technique is to have a child born with a characteristic unlike either of its parents through mutation. The same crossover detailed in Figure 6 is shown

in Figure 9, however this time a mutation in the gene x_7 is shown is red. With crossover alone, this child would not have been possible. However, with mutation, a key component in natural evolution, a unique child is born that could potentially provide an otherwise overlooked viable solution.

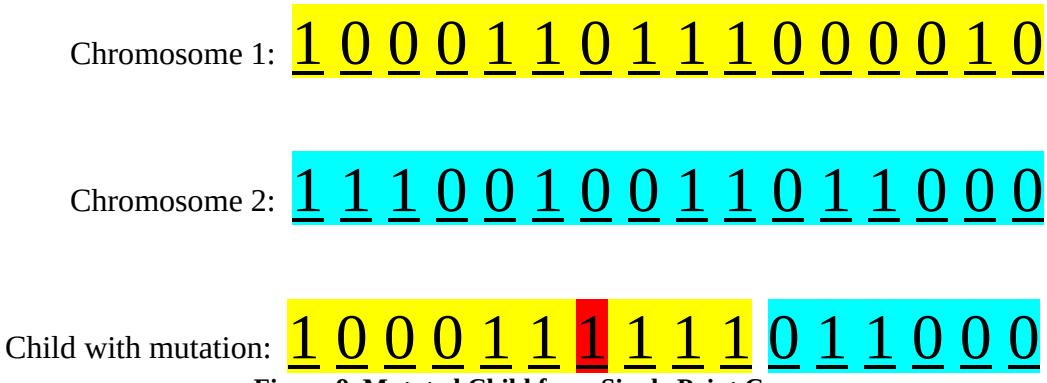

Figure 9. Mutated Child from Single Point Crossover

Mutation occurs at random. One method of determining the mutation is to specify the number of mutations per generation and use a random number generator to determine which gene of which child gets mutated. Another method is to give each gene the same chance of mutation. This means that the number of mutations per generation varies, whereas the first method keeps the same number of mutations per generation[4]. Because of its more random nature, the second method is generally preferred and will be the method of choice in this work.

As the solution begins to converge, the population begins to look more and more uniform. Because the final goal is to find an optimal solution, the existence of mutation at this point can often prevent the final convergence on a uniform optimal solution. Therefore, in practice it is sometimes common to set the mutation rate to zero for the last few generations[4].

1.1.10 Termination

The genetic algorithm is complete when the population converges on a maximized solution. A converging population means that the individuals X_i are all beginning to show the similar genes x_j and are achieving similar fitness values $f(X_i)$. It is possible, but generally not likely, that multiple distinct global optimums could occur. In this case the fitness values would be very similar but the individual's genes are different. Therefore in the final generation the individuals conform to one of multiple optimal solutions.

Caution is necessary however, because prematurely terminating a genetic algorithm could result in a poor convergence on the global optimum or perhaps convergence on a local optimum instead. This could significantly reduce the benefits obtained from the genetic algorithm.

Conversely, letting the algorithm run too long may result in costly computational or experimental time. In general the final goal of a genetic algorithm is to save money during the optimization process. If the money one invests into a genetic algorithm optimization is more than the money spent using a more conventional optimization method, then the investment is a poor one. In this regard, convergence becomes extremely important. Unnecessary populations lead to wasted time and resources.

1.1.11 Alternative Optimization Techniques

There are many optimization techniques that have been used in engineering. As mentioned above, the exhaustive search method evaluates every possible solution in an attempt to find the optimal solution. A slight modification is the progressive exhaustive search, where a coarser search grid is made to more quickly find where the optimal

solution might be found. This region of the solution space is then evaluated with a finer grid. This procedure can be used multiple times until the optimal solution is found[4]. Although this method may be more efficient than the original exhaustive search, the progressive exhaustive search does carry a risk of overlooking the global optimum whereas a purely exhaustive solution would guarantee that the optimal solution over a discretized search space is found.

If the solution space can be represented mathematically, more sophisticated optimization techniques can be implemented. Most of these are derived from fundamental calculus. In the simplest of these methods, peaks are found by evaluating the roots of the derivative of the fitness function. Those roots indicate the presence of a local minimum or maximum. However, no insight is gained as to which of the roots are a global minimum or maximum. An evaluation of all the roots must be conducted. Consequentially, a complex solution space is a large burden for the calculus based methods. In addition, to evaluate the derivatives, a continuous fitness function is necessary. This prohibits the use of discrete and experimentally obtained data as a fitness function as was necessary in the present work where lift to drag ratios from micro aerial vehicle wings are used as a fitness function.

Lagrange multipliers have also been introduced to incorporate constraints on the solution space before finding the roots. With simple derivative based methods the roots have to be evaluated to make sure they satisfied the appropriate criteria. Lagrange multipliers eliminate this problem by incorporating the constraints and the search space into one function, so that the roots obtained already meet the appropriate criteria and no extraneous work is needed to eliminate unrealistic solutions[4].

Hill climbing is a simple technique in which the solution space is analogous to a hill, in which a blind hiker is trying to find the peak of the hill. The hiker, starting in a randomly chosen spot on the hill, takes one step in a random direction. If there is a gain in altitude then another step is taken. If there is a drop in altitude, then the hiker returns to the previous point and takes a step in a different direction. If every direction results in a drop in altitude, the summit has been reached[4]. The hiker could use a similar procedure to find the lowest point of a terrain. Although simple, the technique does not always work if multiple local maxima and minima are present. A simple solution to this problem may include introducing more hikers to improve the chances of finding the global minimum or maximum.

Simulated annealing mimics the annealing process in metallurgy, in which the metal is methodically heated and cooled in order to maximize the crystallinity of the structure and minimize the defects, thus increasing the strength of the metal. By heating the metal and gradually cooling it, the particles have a chance to naturally orient themselves in a lower energy state[4]. In the optimization analogy, nearby solutions of a search space are explored for lower costs. Initially larger jumps in the solution space are evaluated. As time progresses these jumps decrease in size. This method is similar to the hill climbing/descending method with the difference being that simulated annealing allows the possibility of climbing in the uphill direction to avoid converging on local minima and maxima. Other optimization techniques include neural networks, particle swarm optimization, and colony optimization[4].

1.1.12 Previous Applications of the Genetic Algorithm

Although still a relatively young field, genetic algorithms have proven their worth in many different fields, as discussed in this section. Their speed and adaptability have made them an ideal tool for complex design problems.

Genetic algorithms are currently used in many branches of engineering, including the aerospace industry. Many studies have combined computational fluid dynamics and genetic algorithms in order to optimize wing shapes and airfoils. Zhang[7] used genetic algorithms and CFD software to optimize the airfoil of the wing at certain points along the wingspan. Quagliarella[8] and Hacioglu[9] also used genetic algorithms and CFD to optimize a transonic airfoil. Similar to airfoils, helicopter blades may also be optimized using genetic algorithms. Jones[10] used genetic algorithms to develop rotorcraft airfoils that could potentially address aerodynamic and acoustic concerns.

Applications of genetic algorithms in the aerospace industry also include design of aircraft configurations. Marta[11] used aircraft parameters such as the fuselage length and diameter, cruise altitude, take-off weight, wingspan, wing chord, angle of attack, wing location, tail location, and a number of other aircraft characteristics as genes in a chromosome. The range of the aircraft was used as the fitness function. The genetic algorithm was run for search space criteria to mimic that of a small regional jet. Similar studies were done by Liu[12], whose genetic algorithm was tailored towards optimizing an existing F-16, and Blasi[13], whose fitness function was the direct operating cost.

Genetic algorithms are also used in the electronics and telecommunications industry. Cox[14] used genetic algorithms to reduce costs by optimizing the layout of telephone networks. Noren[15] used genetic algorithms to design electronic circuits. Altshuler[16] used genetic algorithms to design a wire antenna.

Application is by no means limited engineering fields. Lucasias[17] used genetic algorithms to determine the structure of a sample of DNA using spectrometric data. Levin[18] used genetic algorithms to identify protein signal sequences in important cellular functions. There has also been much effort put into protein structure prediction, and genetic algorithms have proved to be promising in that field. Accurately predicting the structure of a protein would help make new medicines. Brinkman[19] used genetic algorithms to predict the secondary and tertiary structures of a protein molecule given the primary structure. Deerman[20] and Gates[21] used different genetic algorithms in order to predict the primary structure of a protein given the atomic sequence of the molecule.

Bauer[22] and Chen[23] both showed how the investment industry could use genetic algorithms to trade stocks and bonds more profitably, devise stronger portfolio management solutions, and solve other complex financial problems for themselves and their clients.

Genetic algorithms are also being used in code breaking, game theory, software development, lean manufacturing, and a wealth of other diverse fields. As the use of genetic algorithms becomes increasingly popular the potential of genetic algorithms will be seen in a number of new fields.

1.2 Micro Aerial Vehicles

Micro aerial vehicles (MAV's) are small remotely piloted or autonomous aircraft. There are several types of MAV's which include fixed-wing aircraft, helicopters, and ornithopters (flapping wing aircraft). This thesis will pertain to fixed wing MAV's only. The initial development of MAV's was for use in a reconnaissance based mission. Therefore, small size was critical in the MAV's stealth characteristics. Current MAV's

can have a wingspan of less than six inches. Also, their aspect ratio is lower than most conventional aircraft. The aspect ratio is defined as:

$$AR = b / c \qquad \text{for rectangular wings} \qquad (6)$$

$$AR = b^2 / S \qquad \text{for non rectangular wings} \qquad (7)$$

Equations 6 and 7 show the aspect ratio for each type of wing, where AR is an abbreviation for the aspect ratio, b represents the wingspan, c is the chord length, and S is the planform area. The planform is the vertical orthographic projection of the wing. For a rectangular wing the aspect ratio is the wingspan divided by the chord length. For nonrectangular wings the aspect ratio is the squared wingspan divided by the planform area.

In nature, one can see that the wing planforms of many birds are not the same as the simple shapes of most fabricated wings. Figure 10 shows a number of different species of birds wings: the White-tailed Ptarmigan (upper left), the Sage Grouse (upper right), the California Quail (lower right), and the Wild Turkey (lower right). The scale marker placed underneath each wing (the upper marker denoting inches and the lower marker denoting centimeters) clearly shows the large difference in wing size. One can see that each wing contains a small notch in its trailing edge. Evolutionary theory would lead one to believe that over a long period of time those notches have given the bird an advantageous trait that has enabled the bird to survive and therefore pass this trait on to future generations. Through experimentation in a water tunnel, Drovetski[24] observed that the trailing edge notches increase the maximum lift to drag ratio. It is entirely possible that in finding the optimal shape for a MAV wing, a similar notched wing could be found.

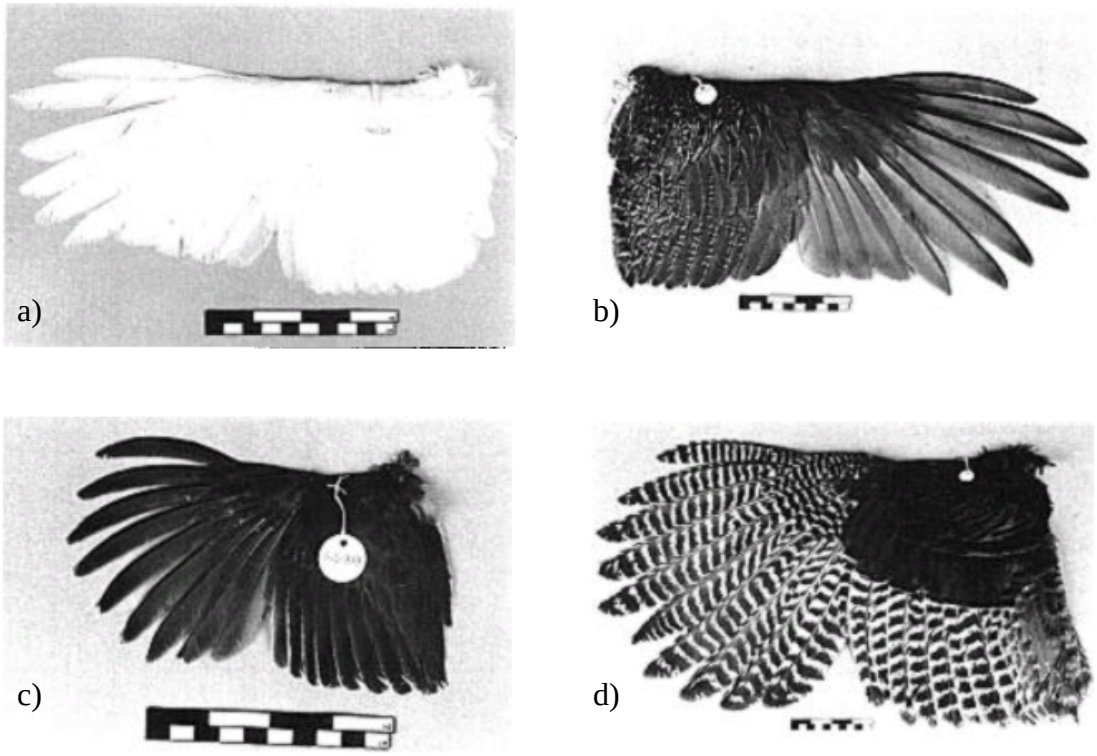

Figure 10. Trailing Edge Notches on Wings of Various Birds[24]

Next we will review other organisms in nature where interesting physical features appear on bird wings or fins of aquatic animals. Each of these features could be candidate features that may evolve from genetic algorithms applied to MAV planforms.

Humpback whales have protuberances on the leading edge of their pectoral flippers. Levshin[25] demonstrates that these protuberances are in fact beneficial to the whale. Conventional airfoils would stall abruptly at a large angle of attack, drastically losing lift. These protuberances prevent this violent stall and the loss of lift is much more gradual.

Bushnell[26] also looks at naturally occurring methods of drag reduction. He too notes the protuberances on the humpback whale. In addition, he discusses serrated trailing edges on sharks, whales, and some birds, as well as the serrated head of a

hammerhead shark shown in Figure 11. Although the physics of the drag reduction is not well understood Bushnell speculates it is a reduction in drag due to lift, as opposed to pressure or viscous drag forces.

Figure 11. Serrations on a Hammerhead Shark

Serrations on the leading and trailing edges of aircraft wings have also shown advantages. Vijgen[27] showed a reduction of drag at low angles of attack. In addition, serrations in wings have also been shown to reduce the noise created by the wing. Scwind[28] observed that an owl's flight is unusually quiet and hypothesized the wing's leading edge serrations are responsible for this. With tests conducted in a wind tunnel it was shown that indeed the presence of fine serrations on the leading edge of a wing can reduce the noise of wing. Dassen[29] showed that serrated trailing edges are effective in reducing noise as well. Burnett[30] described how this technology is being used to reduce the engine noise on commercial airliners by using a serrated engine exhaust nozzle. Additionally, slightly serrated turbine fan blades helped reduce the amount of engine noise created.

Gurney flaps are short flat plates that are attached to the trailing edge of a wing, perpendicular to the chordline. These flaps are shown to increase the lift substantially while creating a small additional drag. Li[31] and Vijgen[27] both demonstrate that using serrated gurney flaps can increase the lift coefficient substantially with less of a drag penalty.

MAV flight is a relatively new field of study in aerodynamics. To date, there has been little effort into optimizing the MAV planform. In order to decrease the wingspan while maintaining a high planform area, many MAVs have adopted a near circular planform. The arbitrariness of these basic planforms leaves room for improvement.

Over millions of years, evolution has changed bird wings into more efficient and more capable instruments of flight. It is my hope that upon completion of this study, the MAV will also evolve, providing a strong planform on which further research can be based. In addition, I hope to show that genetic algorithms are a viable technique to use in wind tunnel experiments. Lastly, it will be interesting to see how my optimum solution compares with existing MAV wings and also existing characteristics of bird wings.

1.3 Project Objectives

- While there has been an increasing effort in studies involving genetic algorithms to optimize airfoil shapes and wing planforms there has been little effort in optimizing the planform of a MAV. It is the goal of this research that through the use of genetic algorithms, a more efficient planform can be found for this type of aircraft.
- Additionally, while it is true that most studies using genetic algorithms are purely computational, there are some that obtain their data through experimental means,

as is the case in this paper. One objective of the paper is to not only help solidify the use of genetic algorithms in general, but specifically those with experimentally obtained data. I plan to show experimentally obtaining data for the use of genetic algorithms is a viable and efficient means of optimization. To this end a novel wind tunnel apparatus, the feather-wing concept, that can be used to easily adapt an MAV planform shape will be developed

- In nature some flying animals have trailing edge notches, serrated wings, and various other wing features. Evolution has taught us that these features serve a purpose even though their precise function may not be well understood. Similarly I hope to show that a less conventional wing planform may indeed be beneficial when creating a micro aerial vehicle. The genetic algorithm may yield one of these features, or a less conventional wing that has not been previously considered as a viable design solution. Finding a more efficient, less conventional wing would further justify the use of genetic algorithms in aerodynamics and potentially lead to exciting advancements in micro aerial flight.

Chapter 2: Methods

2.1 Test Setup

2.1.1 Wind Tunnel

A closed circuit wind tunnel located in Higgins Laboratories at WPI was used to experimentally obtain the lift and drag for the test wings. This wind tunnel can achieve a maximum airspeed of approximately 55 m/s with a freestream turbulence level of 0.73% at lower speeds and 0.5% at maximum airspeed[32]. The test section has a length of 2.4 meters, width of 0.61 meters, and a height of 0.61 meters. The contraction ratio is 6:1.

2.1.2 Test Wing

The test wing used in the genetic algorithm study needed to satisfy the following criteria:

- The wing would need the ability to change its planform shape
- The wing would need a wide variation in attainable planform shapes
- The wing planform would need to be easily altered
- The test wing would need to be small to reduce the wind tunnel blockage
- The range in Reynolds number due to the change in chordlength would need to be kept to a minimum
- The setup would need to utilize the side access hole in the wind tunnel

In order to change the planform of the wing a feather-like concept shown in Figure 12 was adopted in which tabs could be pulled to shorten or lengthen each feather to create a different planform. This setup closely represents a bird wing where the number and configuration of a bird's feathers determines its wing planform. Because the

feathers needed to be lengthened and shortened it was necessary to only use half a planform (one wing). This way the wing was placed through the side porthole of the wind tunnel as shown in Figure 13. The tabs that altered the feather lengths were then readily accessible from outside the tunnel as shown in Figure 14. The boundary layer thickness resulting from the boundary layer development on the side wall of the wind tunnel was estimated to be approximately 1.3 centimeters.

Figure 12. Feather Wing Concept

The wing had seven test feathers and a half wingspan of approximately 13 centimeters. The feathers were compressed between a lexan housing in order to keep the feathers close together and guide their retraction. The width of these feathers decreased toward the lexan housing to prevent crowding when the feathers retracted.

Figure 13. Test Wing Setup through Side Porthole

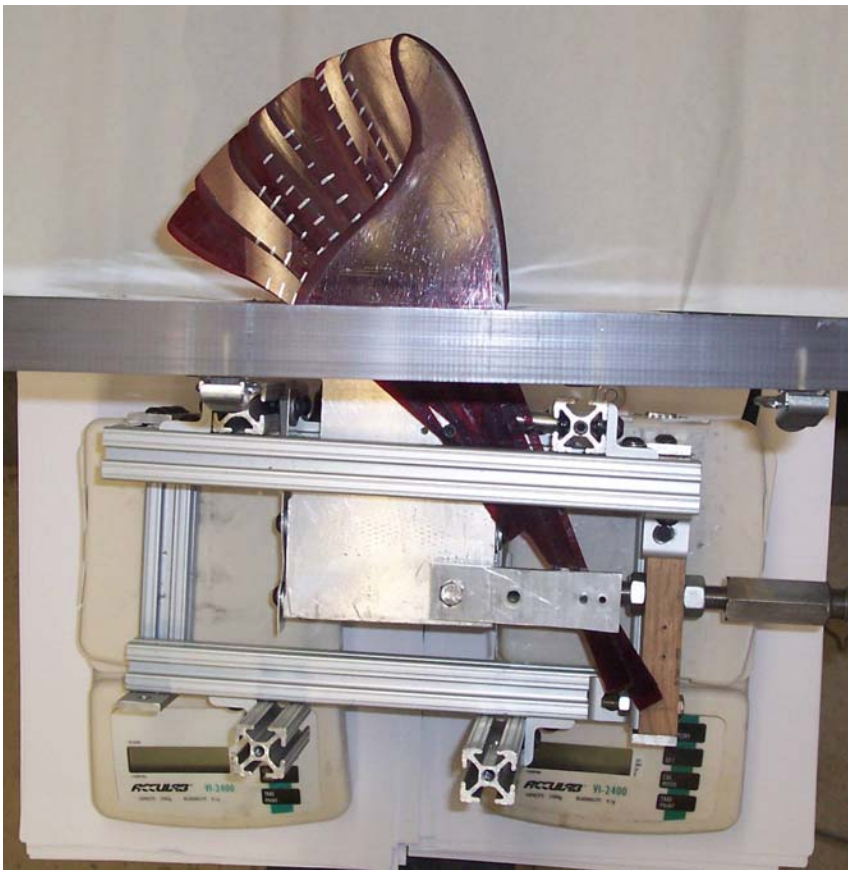

Figure 14. Top View of Test Setup

Because each feather varied in length they also varied in how far they could be retracted. Longer feathers generally could retract more and therefore had more alleles for that particular gene. Table 3 shows the breakdown of approximately how far each feather could retract and how many alleles this resulted in for each feather. Each length was divided by 6mm to find the approximate number of alleles, with one additional allele being the feather in its fully retracted position. The markers shown on the feathers in Figure 12 are used to properly position the feathers. For example Figure 12 represents the individual (7 7 5 5 4 3 2). The genes correspond to the spaces visible between the visible feather markers. This breakdown of alleles led to a solution space of approximately 140,000 (the product of the number of alleles listed in Table 3).

Table 3. Breakdown of Alleles for Each Feather

Feather #	Retraction Length (mm)	Resulting Alleles	Gene Values
1	67	8	0-7
2	67	8	0-7
3	52	6	0-5
4	49	6	0-5
5	37	5	0-4
6	31	4	0-3
7	18	3	0-2

The leading edge of the wing was rounded to reduce the drag on the lexan housing. The seven feathers and the trailing edge of the housing were also streamlined in order to reduce the drag and reduce the effect of ‘stepping’ experienced by the flow over one feather to the next.

The root chordlength of the wing was approximately 11.5 centimeters. With the feathers extended, individual (7 7 5 5 4 3 2) in Figure 15, the area of the wing was 157cm^2 with a mean chordlength of 13 cm. The aspect ratio was 2.22. With a test velocity of 18m/s this corresponded to a Reynolds number of 182,000.

Figure 15. Test Wing with Feathers Fully Extended for Zimmerman Representation

Figure 16. Test Wing in with Feathers Fully Retracted

With the feathers fully retracted, representing individual (0 0 0 0 0 0) shown in Figure 16, the area of the wing was 105cm^2 with a mean chordlength of 11.5 cm. The aspect ratio was 3.33. With a test velocity of 18m/s this corresponded to a Reynolds number of 161,000.

2.1.3 Force Balance

In order to measure the lift and drag experienced by the wing, a force balance system recently developed by a group of WPI students as part of their Major Qualifying Project was used[33]. Modifications were made to the force balance setup in order to accommodate the needs of the test wing. The original design had a sting passing through the test section bottom wall into the wind tunnel. This allowed the test wing to be located in the middle of the wind tunnel flow. Because only a half span wing is used in the current work, the sting arm of the previous force balance was removed and the wing was attached directly to the base fixture using flexible sheet metal shims. Due to a much lower drag anticipated on the wing, it was necessary to use only two shims of a more flexible aluminum. This would make the wing more sensitive to drag forces incurred. A photograph of the wing in the test fixture is shown in Figure 17.

Using an iron saw horse placed on cinder blocks, the fixture was raised to the appropriate height in order to fit the half wing through the preexisting access hole in the side of the wind tunnel (Figure 18). A Plexiglas insert was milled in order for the wing to protrude into the wind tunnel while minimizing the amount of air escaping through the hole in the side of the tunnel. This was further corrected by placing a transparency sheet over the remaining gaps.

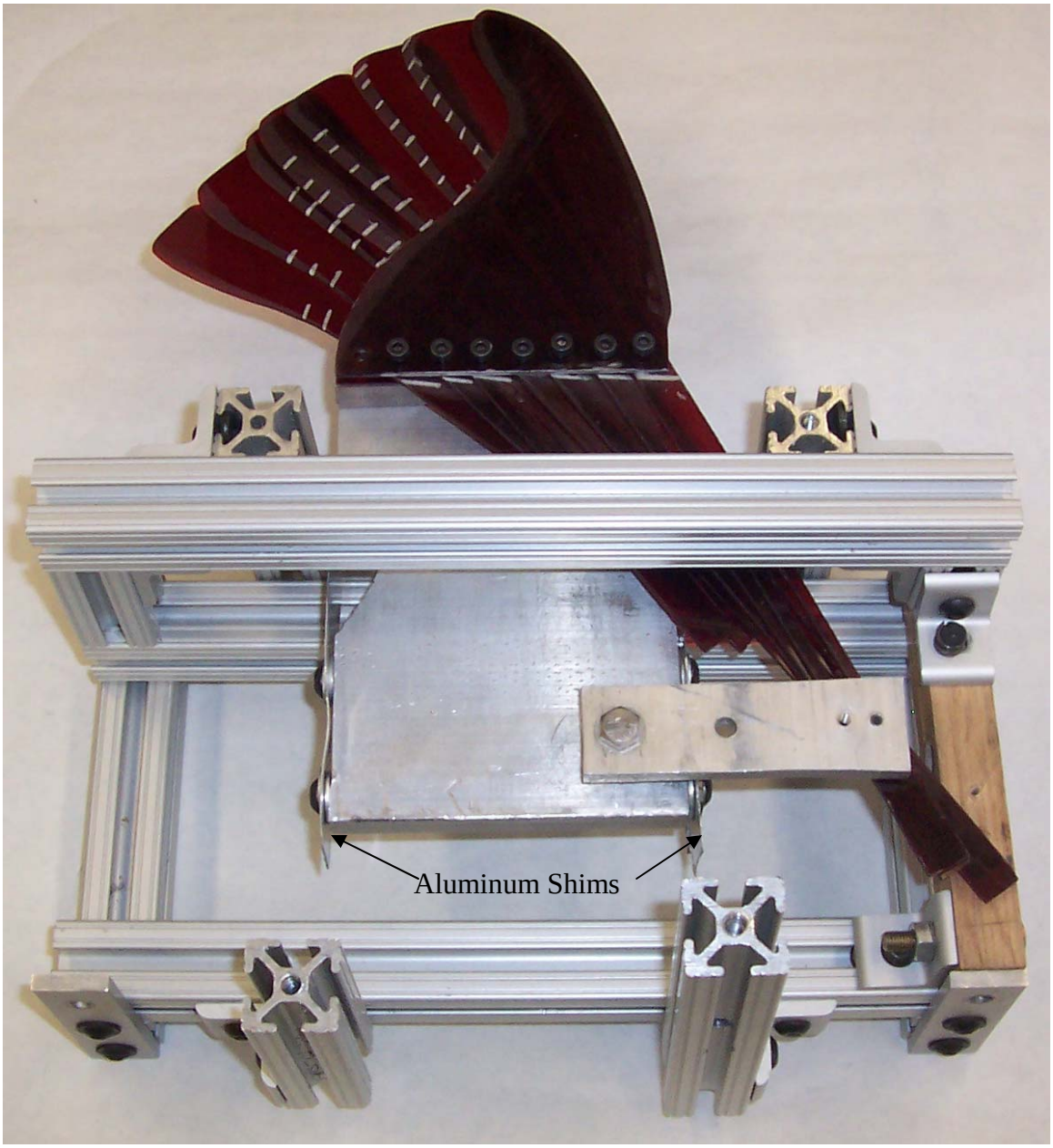

Figure 17. Wing Test Fixture

It was important however that neither the Plexiglas insert nor the transparency touched the test fixture. Because the two objects were attached firmly to the wind tunnel the friction caused by their contact with the test fixture would introduce inaccuracies in the lift and drag measurements.

Figure 18. Test Setup on Raised Iron Horse

The test apparatus rested on two separate Acculab VI-2400 digital scales, each with an accuracy of ± 0.1 grams, in order to calculate the lift on the wing. Once a wing was ready to be tested, the scales were zeroed. When the tunnel was at the desired airspeed of 18 m/s the scales would each read a negative value. The absolute value of the sum was the lift on the wing. The angle of attack of the wing could be altered by raising

the height of one of the digital scales. The angle of attack during the genetic algorithm testing procedure was 4.6 degrees.

As previously stated, the test wing was attached to the base fixture using flexible metal shims. This essentially connected the wing to a spring mass system in the direction of the wing's chordline, such that when the wing experienced a drag force the shims would deflect. An Indikon AP1297-2 eddy current proximity probe shown in Figure 19 translated this displacement into a voltage. The difference in displacement between a steel target on the fixture and the probe shown in Figure 20 created a small difference in the magnetic flux density field. The probe generated an output voltage that was linearly proportional to the displacement. This output voltage was amplified and measured using a standard Hewlett Packard 3478A multimeter.

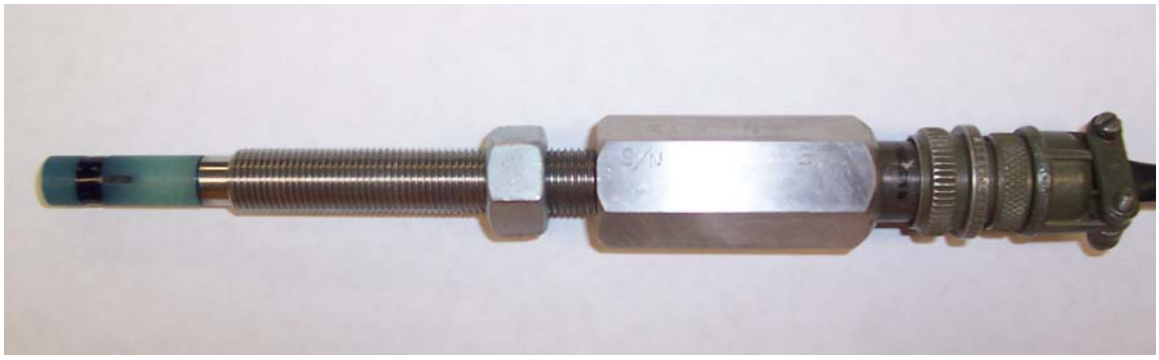

Figure 19. Indikon AP1297-2 Eddy Current Proximity Probe

To find this constant of proportionality a fishing line was attached to the test wing and also to the top of the wind tunnel. By suspending a known mass from the string and measuring the angles the string made with the horizon, an equivalent drag force on the wing was calculated. A schematic of the calibration setup is shown in Figure 21.

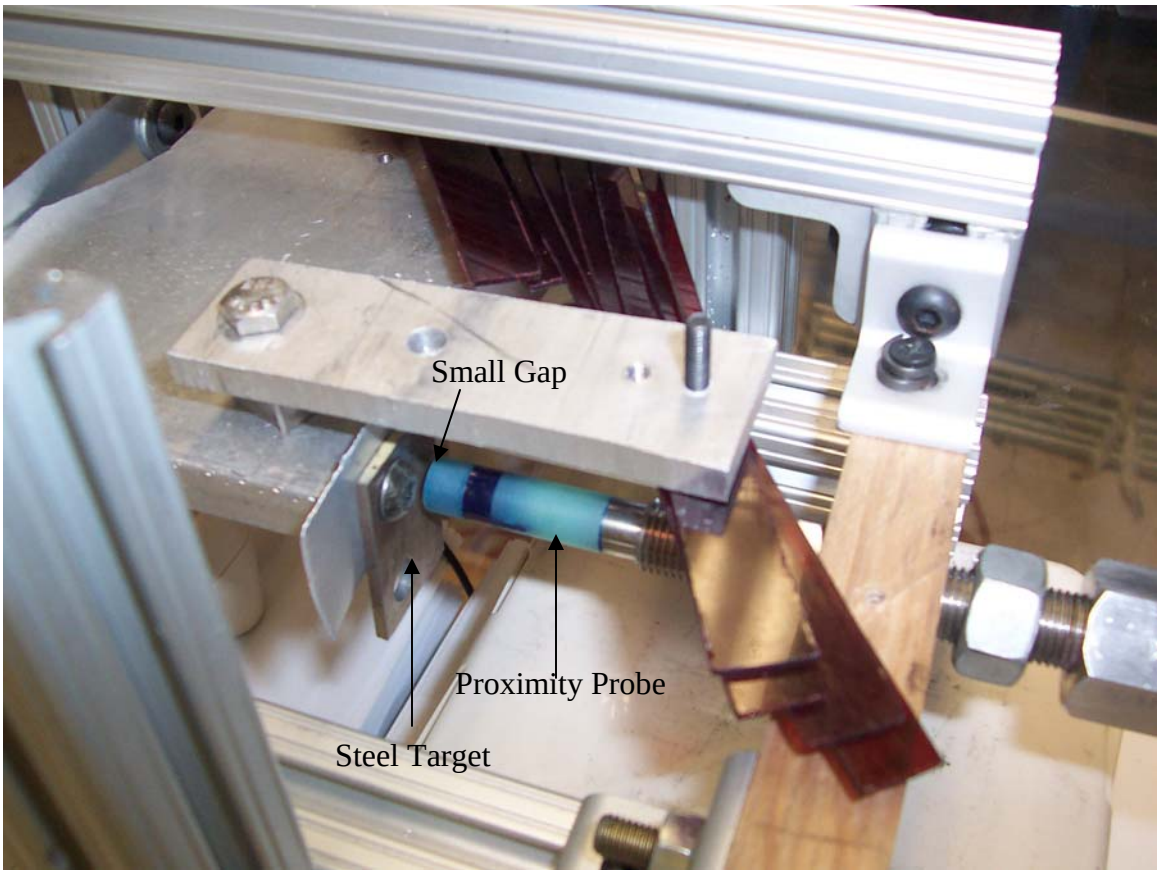

Figure 20. Proximity Probe in Test Setup

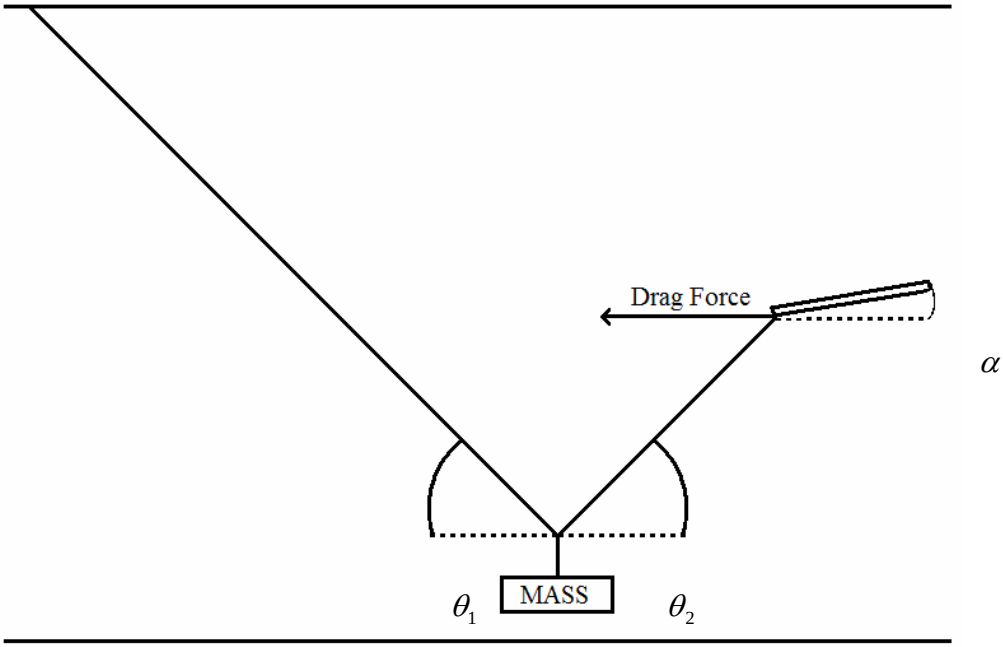

Figure 21. Drag Calibration Schematic

Because the wing is connected to a spring mass in which force and distance are also directly proportional, the relationship between the voltage read from the proximity probe and the drag experienced by the wing was therefore directly proportional. The drag for various masses was plotted with the resultant voltage read from the multimeter. A linear regression was used in order to determine the drag as a function of the voltage. Drag calibrations were performed before each generation to ensure that the voltages read yielded accurate drag forces. A sample drag calibration plot is shown in Figure 22 where a linear relationship between measured drag force and the output voltage is confirmed.

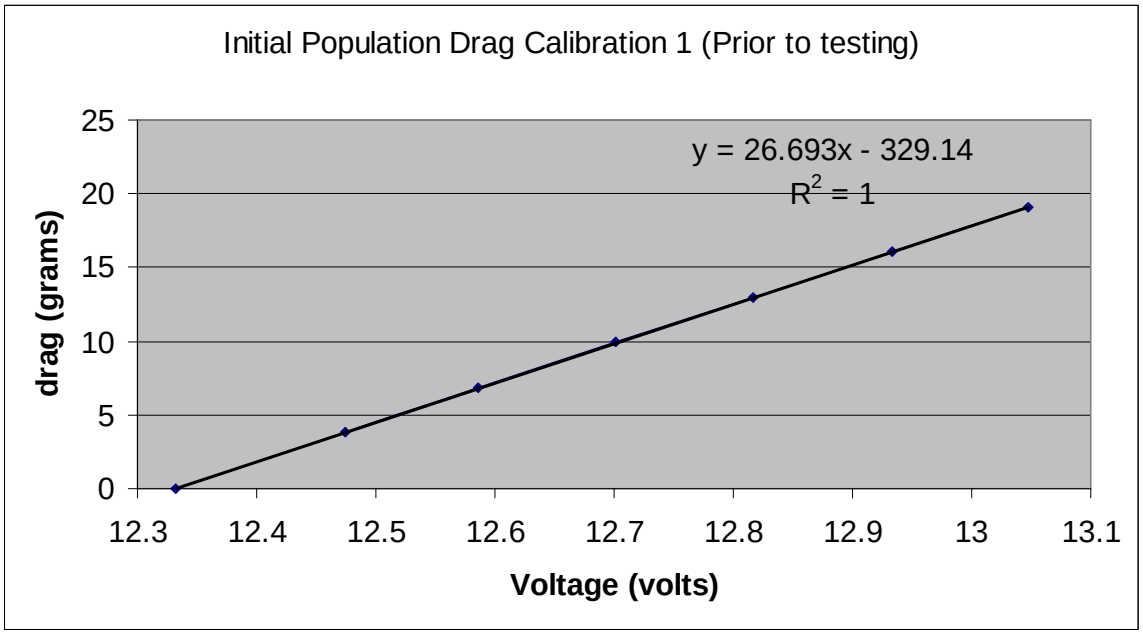

Figure 22. Sample Drag Calibration Curve

It is important to realize that the wing deflects in the direction of the chordline of the wing, and not the direction of the flow. The difference between these two is the angle of attack α of the wing. Figure 23 shows a vector diagram of the test wing. The drag force is parallel to the flow and the lift is perpendicular to the drag. The axial force is parallel to the chordlength and the normal force is perpendicular to the axial force.

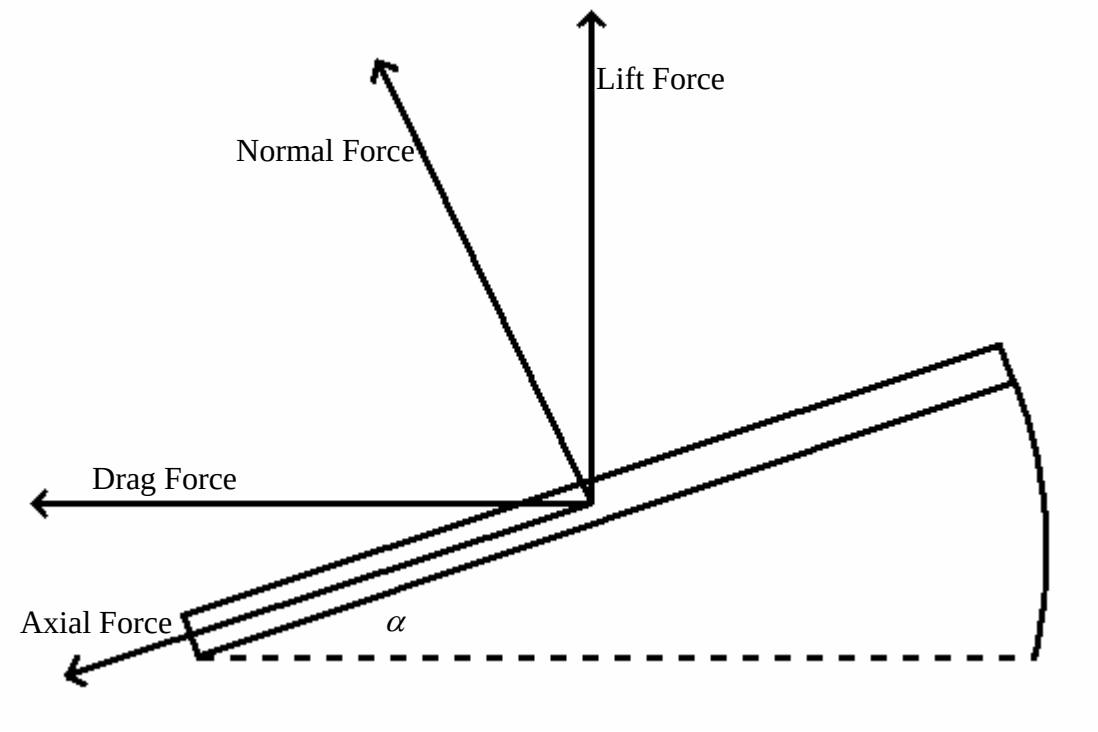

Figure 23. Vector Diagram of Wing

The proximity probe is in the axial force direction. Therefore, while the drag experienced moves the target away from the probe, the lift moves the target slightly towards the probe. When calculating the drag it is necessary to compensate for this apparent loss of drag through:

$$D = (V_f - V_i)s + L\sin(\alpha) \tag{8}$$

In this equation, V_i is the voltage read at zero flow, and V_f is the voltage read at the testing speed. The difference is multiplied by the drag slope s with a range of values from 25-27 grams per volt.

2.1.4 Matlab Genetic Algorithm

MatLab was used in order to create a program that could incorporate the fitness values taken from the wind tunnel and perform the parent selection, pairing, and mating processes to ultimately output the next generation of wing planforms to be tested. MatLab

provides a Genetic Algorithm and Direct Search Toolbox that is commercially available, however the software was not able to suit the needs of this project. The MatLab software was developed for continuous functions, rather than experimentally obtained results. Because of this an entire new program was created with help from a few open source files from the existing MatLab toolbox. Based on the individuals and their fitness values read from the experimentally obtained data on a spreadsheet, the program would give the appropriate next generation to be tested. The MatLab code and a more in depth discussion of how the program functions are presented in Appendix A.

2.1.5 Test Procedure

A MatLab script was written to make an initial population of 72 individuals. As a rule of thumb, the initial population should be at least twice as large as the subsequent generations. These generations should be at least as large as the number of chromosomes if the system was binary. Since it was not, it was easy to convert to binary by taking log base two of the solution space. This gave a number of approximately 17 individuals for the future generations; however a slightly larger population size of 24 was chosen for generations $n > 1$. The initial population was three times larger than that. In our experiments $n = 12$ generations were used (Section 3). As a result, with a population of 72 individuals for $n = 1$ and 24 individuals for $2 \le n \le 12$, a total of 336 wing planform shapes were studied. Time constraints limited the study to a single run $1 \le n \le 12$ of the genetic algorithm process. Although it was time consuming to test every one of these 336 soluitons, it helped guarantee a representative portion of the solution space had been sampled.

The initial population was tested in the wind tunnel. Measuring the lift and drag of the wing started with correct placement of the wing feathers. Once this was accomplished a plastic spacer was used in order to ensure the wing was protruding into the wind tunnel the same amount for each wing. Then, the spring mass was put into a small harmonic motion and monitored to ensure there was no interference with the transparency sheet. Had there been interference the motion would have ceased in a short time. Once it was verified that there was no interference the scales were zeroed. The baseline voltage was taken after the spring mass system stopped moving. The wind tunnel was turned up to an airspeed of 18m/s. The lift was read using the scales, and the voltage was taken. As mentioned before, the lift was simply the absolute value of the sum of the scales. The drag was calculated using Equation 8.

The fitness function for each individual was the lift to drag ratio minus a constant number. The lift to drag ratio is an aerodynamic parameter that indicates how efficiently a wing produces lift. The constant subtracted from the lift to drag ratio was just slightly lower than the lowest lift to drag value experienced in the current population of wings. This accentuated the difference between the better performing and worse performing wings when it came to parent selection.

Once the fitness values were calculated in the spreadsheet, another MatLab script took the population and their fitness functions and exported the new generation of chromosomes to the same spreadsheet. The method of parent selection was the roulette wheel. Half of the population was selected as parents. The parents that were selected were randomly paired (the result of sequentially pairing a random list of parents). Once the pairs were determined the mating process was also random producing two children

per pair in order to maintain the same population size. Mutation was also present at a rate of 20%.

Figure 24 shows an example of the mating process that took place between two parents selected from the initial population. One can clearly see which parent contributed which gene to the children. The mutations are shown in red.

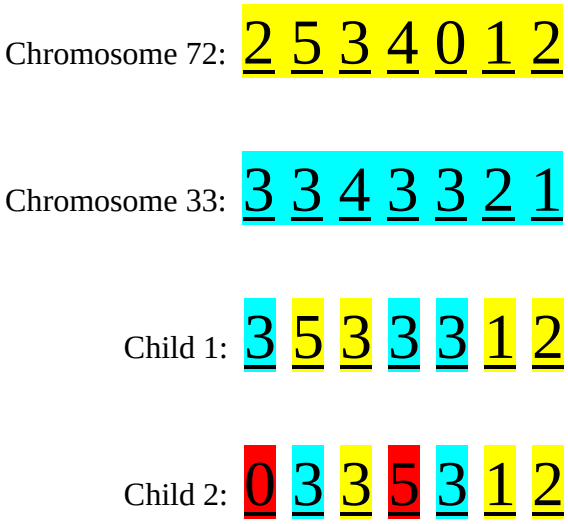

Figure 24. Resulting Children from Selected Parents in Initial Population

The next generation was tested in the wind tunnel to find the fitness values. These were then inserted back into the MatLab program to find the next generation of chromosomes. This process was repeated until the proper convergence criteria had been met. These convergence criteria will be discussed in the results section.

2.1.6 Error Analysis

The error was categorized into instrumentation error and repeatability error. The instrumentation error was the error incorporated with reading the scales for the lift force and the multimeter for the drag force. The voltage taken from the digital multimeter was not constant and therefore introduced a small level of inaccuracy. The voltage usually varied within ± 5 mv. With a drag slop of approximately 26.93 from Figure 22 this corresponded to an error in the drag measurement of ± 0.135 grams. The error in the lift

measurements was also small. The resolution of the scales was 0.1 grams. The lift was usually quite stable, within ± 0.1 gram.

There was also the repeatability error from generation to generation. This repeatability error was found to be larger than the instrumentation error and was due to the inability to recreate the exact same conditions for identical wings. Drag calibrations were taken often in order to have an accurate drag slope; however these calibration curves were also subject to error. While there was little error in the voltages read from the hanging of the calibration weights, small errors in the angle measurements could lead to small errors in the drag calibration curves and therefore the drag numbers from generations to generation could vary.

There was also error in the inability to recreate the exact same wing shape using the feathers. Although the feather increments were clearly marked it is possible the feather placement when making an individual created slight error. Also, despite the spacer used, it is possible that the lateral distance the wing protrudes into the wind tunnel varied ever so slightly. Error bars that incorporate these sources of error will be presented in the Results section.

Chapter 3: Results

3.1 General Results

In this section the general results of the genetic algorithm will be discussed. In Section 3.2 the dynamics of the wing evolution resulting in the best planform solutions will be discussed. In Section 3.3 the test wing Zimmerman representation will be compared to published Zimmerman data. Local chordlength distributions along the wingspan will be presented and analyzed. Lastly in Section 3.4 a discussion of the wing aspect ratio, the Reynolds number variation, and other possible effects on the test wing will be discussed.

Figure 25 shows the average L/D for each generation, the average L/D of the parents of that generation, and the high and low L/D values of the generation. Total error bars are presented on the parent L/D averages because these individuals appeared in several previous generations before. Therefore when taking lift and drag numbers, there is already data from the previous generation so that a standard deviation of the repeatability error, σ_R, can be calculated. The instrumentation error σ_I is calculated by taking the high and low values of the digital scales and the multimeter. The sum of the standard deviations of these two errors, $\sigma_{Total} = \sigma_I + \sigma_R$, is used to calculate the error bars seen in Figure 25.

$$\frac{L}{D}_{high} = \frac{L}{D} + \sigma_{Total} \tag{9}$$

$$\frac{L}{D}_{low} = \frac{L}{D} - \sigma_{Total} \tag{10}$$

The total number of generations to study was not determined before testing began. Because of the complex nature of the genetic algorithm and time limitations for the

number of possible generations run, it was decided to start the genetic algorithm and observe how the genetic algorithm progressed before convergence criteria were set. In the end, the algorithm was terminated at the 12^{th} generation when 9 out of 12 of the parents were identical to the strongest known wing shown later in Figure 30. The other three wings were slight variations of this wing. Due to the overall conformity of the generation and the fact that the total error bars on the average parent L/D values encompassed both the high L/D value and the average L/D values (see Figure 25, $n = 12$) it was determined that further testing with a higher n generations was not likely to yield a stronger solution. While there is no guarantee that the best wing planform found is indeed the optimal configuration, it is, with several other similar wings, a very strong solution. In this regard, the time available was also a factor in the termination of the genetic algorithm. If more time were available more generations could have been tested to increase the level of confidence that the best solution had been found.

One can see clearly that maximum, minimum, and average L/D values all increase in Figure 25 with increasing generation number n, indicating that the genetic algorithm was functioning correctly.

- Average L/D value increased by 12.6%
- Average parent L/D value increased by 10.7%
- Maximum L/D value increased by 3.7%
- Minimum L/D value increased by 16.6%

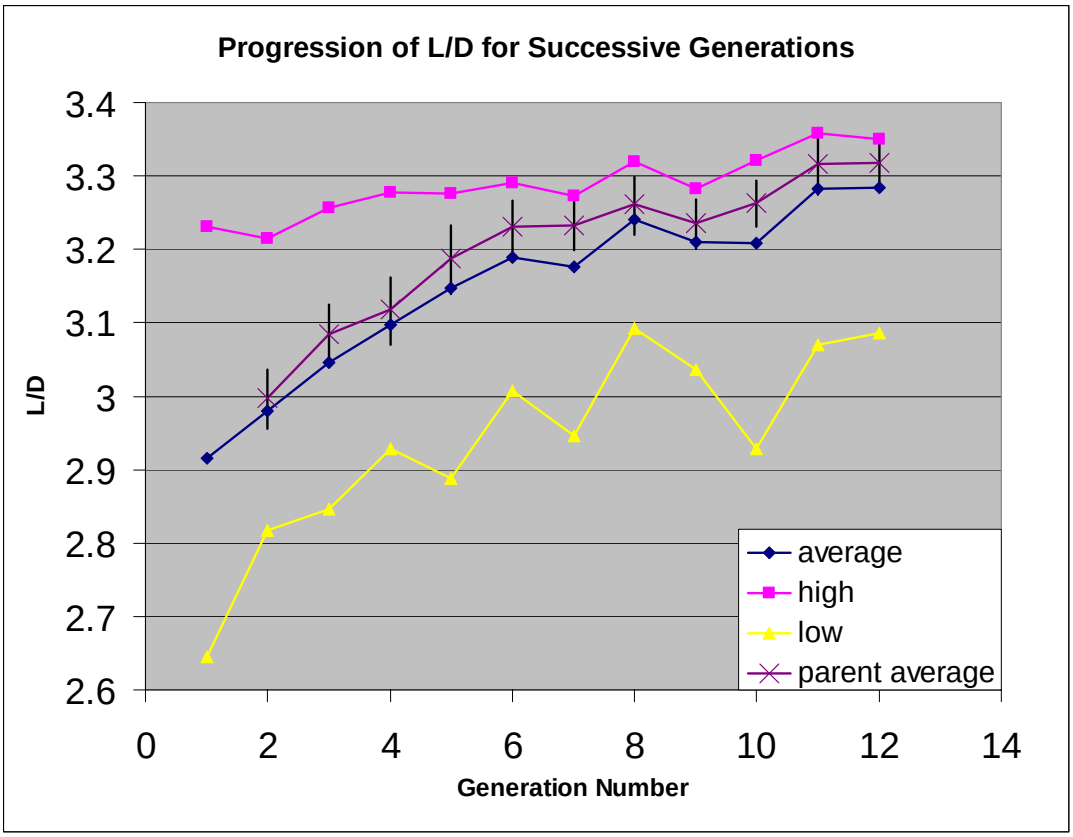

Figure 25. L/D Trends in Successive Generations

The increase in the maximum L/D value is comparatively small. This is due to a fairly efficient wing (L/D=3.23, individual (7 7 3 3 2 0 0)) that was created in the initial population, $n=1$. Even small increases in L/D however can lead to considerable increases in performance in range and endurance. The range R and the endurance E of an aircraft are given by equations 11 and 12.

$$R = \frac{L}{D}\frac{V}{c}\ln\left(\frac{W_i}{W_f}\right) \tag{11}$$

$$E = \frac{L}{D}\frac{1}{c}\ln\left(\frac{W_i}{W_f}\right) \tag{12}$$

In these equations V is the cruising velocity, c is the wing chordlength, W_i is the initial takeoff weight of the aircraft, and W_f is the final weight of the aircraft once its fuel has been expended. One can see that both the range and endurance equations are directly proportional to L/D. Therefore an increase in L/D of 10%, seen from the full Zimmerman representation to the best wing produced from the genetic algorithm, corresponds to an increase in range and endurance of 10%. This is a significant improvement for MAV design.

It is important to note however, that not every generation had a higher average L/D than the generation before it. Generations 7 and 9 both saw small decreases in the average L/D values. This meant that in these generations the parents on average bred in such a way that the children created had lower L/D values. A similar argument can be said about the average L/D values for the parents selected in generation 9. In this generation, on average, weaker parents were chosen. This apparent anomaly is expected to occur because the algorithm is ultimately based on random chance.

Figure 26 adds photos of individual planform shapes to the results of Figure 25. Wings with the highest L/D values for the first, ninth, eleventh, and twelfth generations and lowest L/D values for the first, fifth, ninth, and twelfth generations are shown.

The shapes of Figure 26 a-d are very similar and will be discussed later, but one can attribute these small changes to mutations that occurred in the mating process. The lowest L/D planforms shown in Figure 26 e-h have a more dramatic variation in planform shape showing crossover effects that were at work. One could also note large trailing edge serrations (discussed as a possible candidate feature in Section 1) are present for example in Figure 26 g) and h).

The lowest L/D wings may be useful for MAV design if the possibility of 'morphing' wing shapes during MAV flight at different flight conditions is considered. Consider a descent/landing condition where one wants to descend as quickly as possible. To accomplish this one would want to lower L/D to minimum values. The wings in Figure 26 e-h may be suited for this application, where one could 'morph' the wing planform shape from the high L/D shapes to the low L/D shapes in flight at the start of the descent.

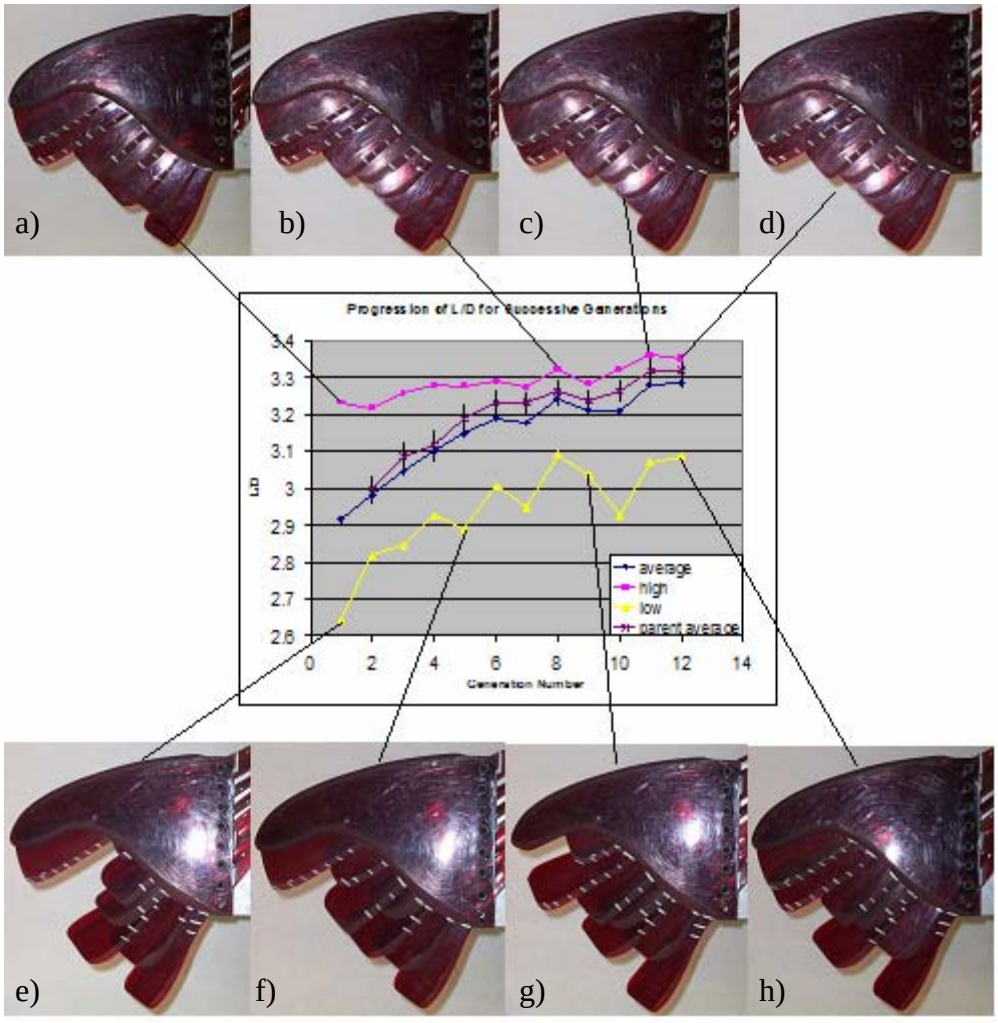

Figure 26. L/D Trends with Certain High and Low Individuals Shown

3.2 Wing Evolution

The first generation of 72 individuals was a very diverse population encompassing a large portion of the solution space. Figure 27 shows a sampling of eight individuals from this generation. One can see the large variation in the planform shapes. For example Figure 27 a) shows a large notch in the planform at the third gene. Figure 27 b) shows a large span near the wing leading edge and then monotonically increasing feather lengths. Figure 27 c) and d) show large thin notches as well as smaller thicker notches. Figure 27 e) and f) might be characterized has having large amplitude serrations. Figure 27 g) shows a wing with a small but wide trailing notch. Figure 27 h) shows a very large trailing edge gap near the leading edge in addition to a small notch near the trailing edge.

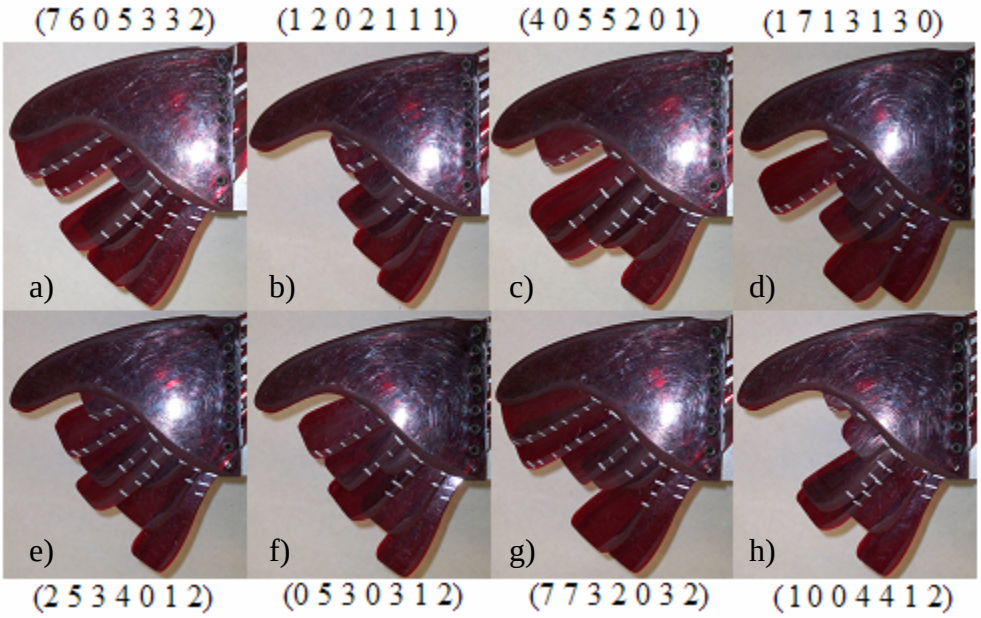

Figure 27. Selected Individuals from First Generation

As previously stated, the first generation yielded a comparatively strong individual. This individual, whose seven genes are (7 7 3 2 2 0 0), is pictured in Figure 28.

Figure 28. Individual 7 7 3 2 2 0 0

This individual was chosen as a parent for generation 2. In successive generations 3-6 this individual became selected as a parent more and more often at 2, 3, 4, and 6 times for generation 3, 4, 5 and 6 respectively. For reference, data for each generation is

available in Appendix B. In the seventh generation ($n = 7$) a slightly stronger wing was created as a child. This occurred when the strong wing from Figure 28 bred with an identical individual. Without mutation the child from these identical parents would have been identical to the parents. A slight mutation of the 4th gene yielded an individual with genes (7 7 3 3 2 0 0) shown in Figure 29 that had a slight advantage in the lift to drag ratio.

Figure 29. Individual 7 7 3 3 2 0 0

In the 8th generation, the individual in Figure 28 again bred with an identical individual to create an even stronger individual (7 7 3 2 1 0 0) through a random mutation of the 5th gene (see Figure 30). All three of these wings are very similar in shape and only slightly different in performance.

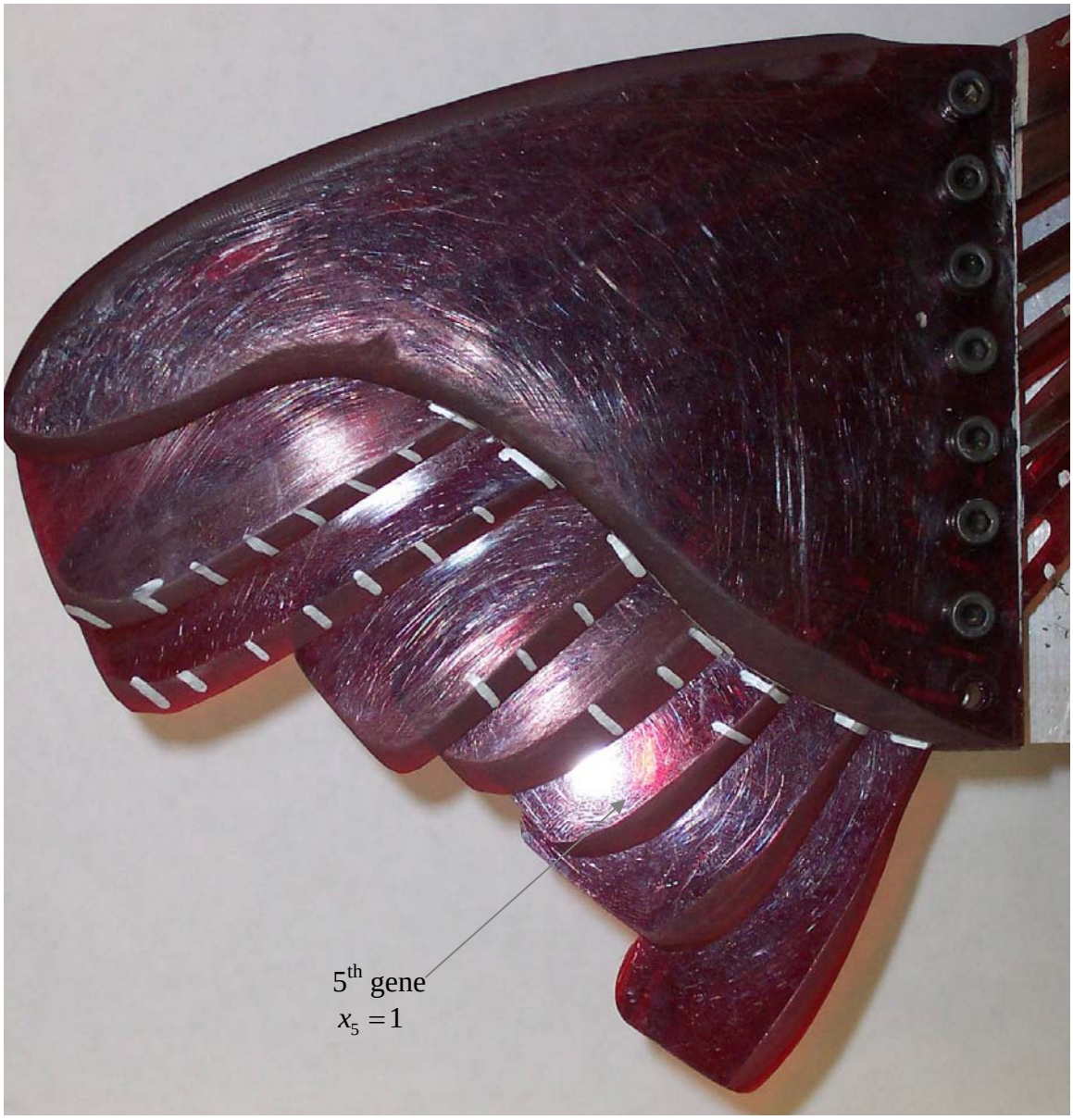

Figure 30. Individual 7 7 3 2 1 0 0

Generations 9-12 experienced a similar dynamic found in generations 3-6, where the best performing wing was selected more and more frequently as a parent. In

generations 9, 10, 11 and 12 individual (7 7 3 2 1 0 0) was selected 3, 6, 9, and 9 times respectively.

3.3 Comparison to Zimmerman Shaped Wings

In order to validate our experimental L/D data, we must present results comparing one of our wing shapes to existing literature. A Zimmerman planform has been used as a MAV wing shape fairly extensively. Torres and Mueller investigate the characteristics of low aspect ratio wings at low Reynolds numbers[34]. One of the wings investigated was a Zimmerman planform with an aspect ratio of 2 and a root chord of 8 inches. The wing was tested at a Reynolds number of 70,000 and 100,000. In this section, we will present results from our work, where our feather wing test model is set up in order to closely approximate a Zimmerman planform. L/D and other characteristics are then compared to the published data.

The test wing in its full Zimmerman representation is shown in Figure 31 (all its feathers fully extended). The Zimmerman model uses two half ellipses pieced together at the quarter chord point to form a continuous planform shown as the shaded region in Figure 32. L/D data was collected from the Zimmerman test wing and compared to published data on the Zimmerman planform[34].

Figure 33 shows that the published L/D values for the Zimmerman planform are significantly higher than the Zimmerman representation achieved with the test wing over a wide range of angles of attack. There are a few key differences between the wings in the two studies that might account for the lower L/D values.

Figure 31. Test Wing Zimmerman Representation

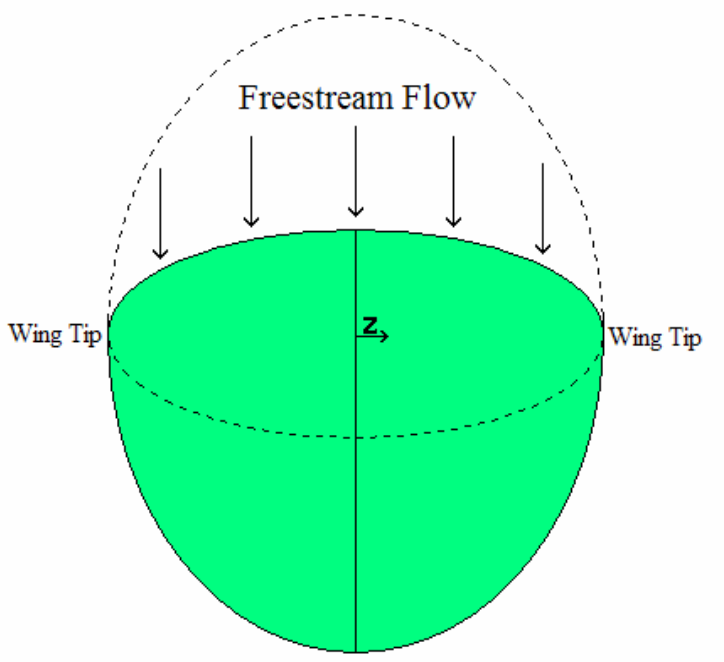

Figure 32. Zimmerman Planform

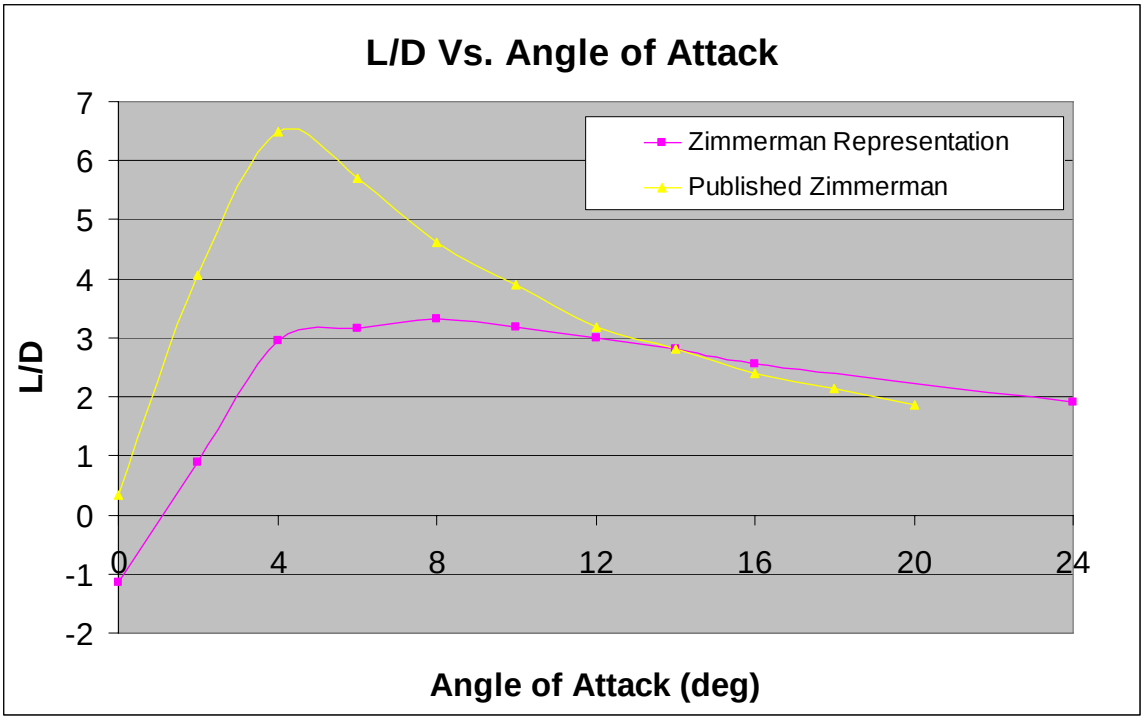

Figure 33. Published L/D Values for Zimmerman Representation and Published Zimmerman Data

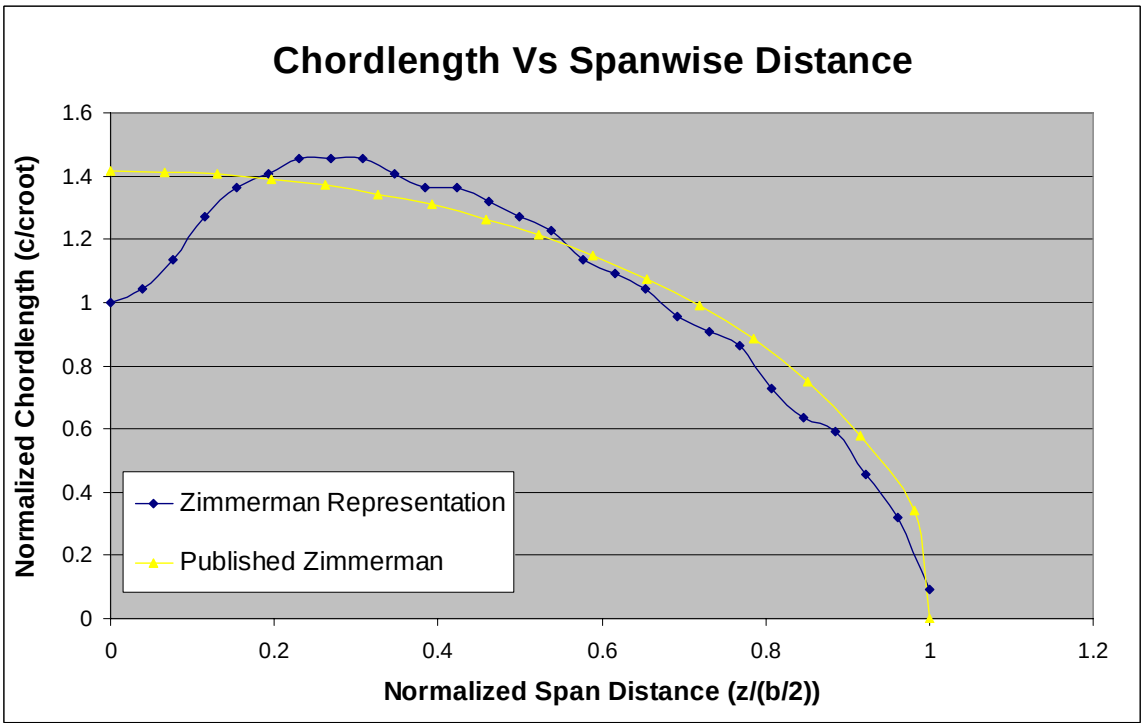

Figure 34. Local Chordlength versus Normalized Span Distance

To illustrate the difference in planform areas, Figure 34 shows the chordlength versus the spanwise distance for the test wing Zimmerman representation and the

published Zimmerman wing. The chordlength is the distance from the leading edge to the trailing edge of the wing. The spanwise distance is perpendicular to the chordlength and is measured from the root of the wing. For comparison, the chordlength and the spanwise distance are normalized with the root chordlength and the half span value respectively. Because of the different root chordlengths of the test wing and the published Zimmerman data, the published Zimmerman wing was normalized in order to give the published Zimmerman and the Zimmerman representation the same planform area. From Figure 34 one can see the discrepancy between the published Zimmerman and the Zimmerman representation near the root of the wing. This area is critical in lift production. A plot of the lift coefficients in Figure 35 shows that the published data for the Zimmerman is significantly higher than that of the test wing representation for all angles of attack.

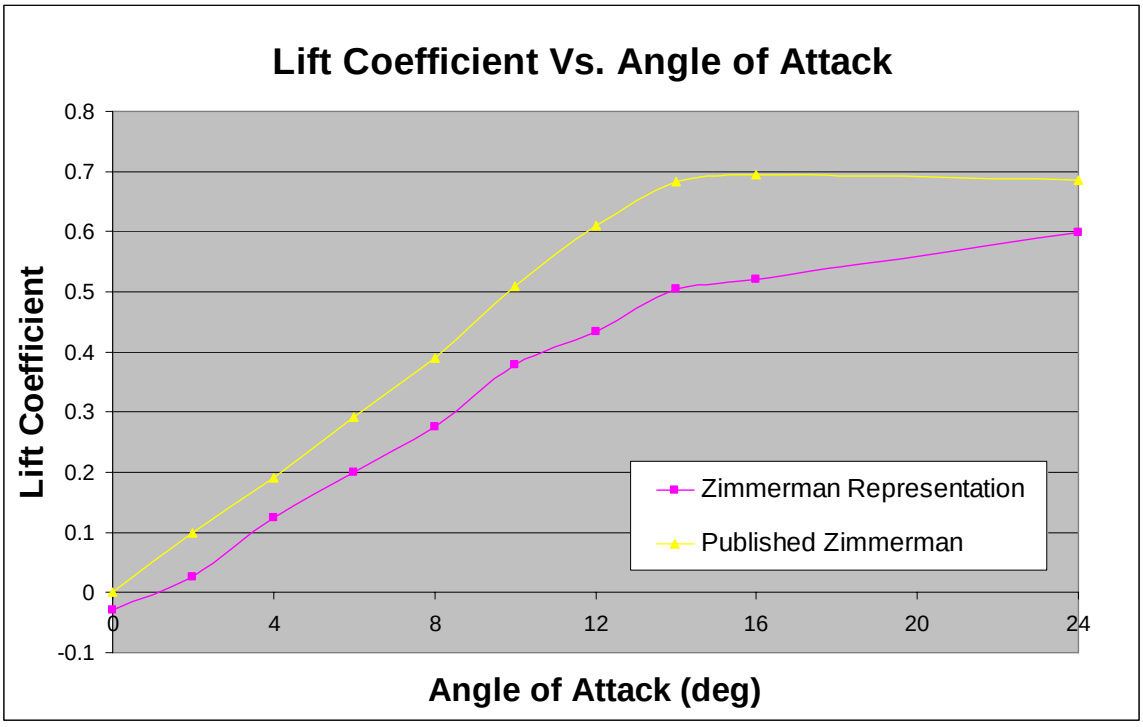

Figure 35. Published Lift Coefficients for Zimmerman Planform and Test Wing Representation

Although the wing was created with the Zimmerman planform in mind, certain design challenges prohibited the full planform from being realized. Geometric limitations made it extremely difficult to add a variable length feather in the design to create this added wing root area. This is because this feather would have needed to have been essentially aligned with the flow direction and therefore would have made it difficult to pass this feather outside the test section so it could be adjusted, etc. Future design work might incorporate a fixed feather that eliminates this area difference. Figure 36 shows the test wing with the additional area shown. Had it been possible to construct a wing to accomplish this difference, lift values might have been significantly increased.

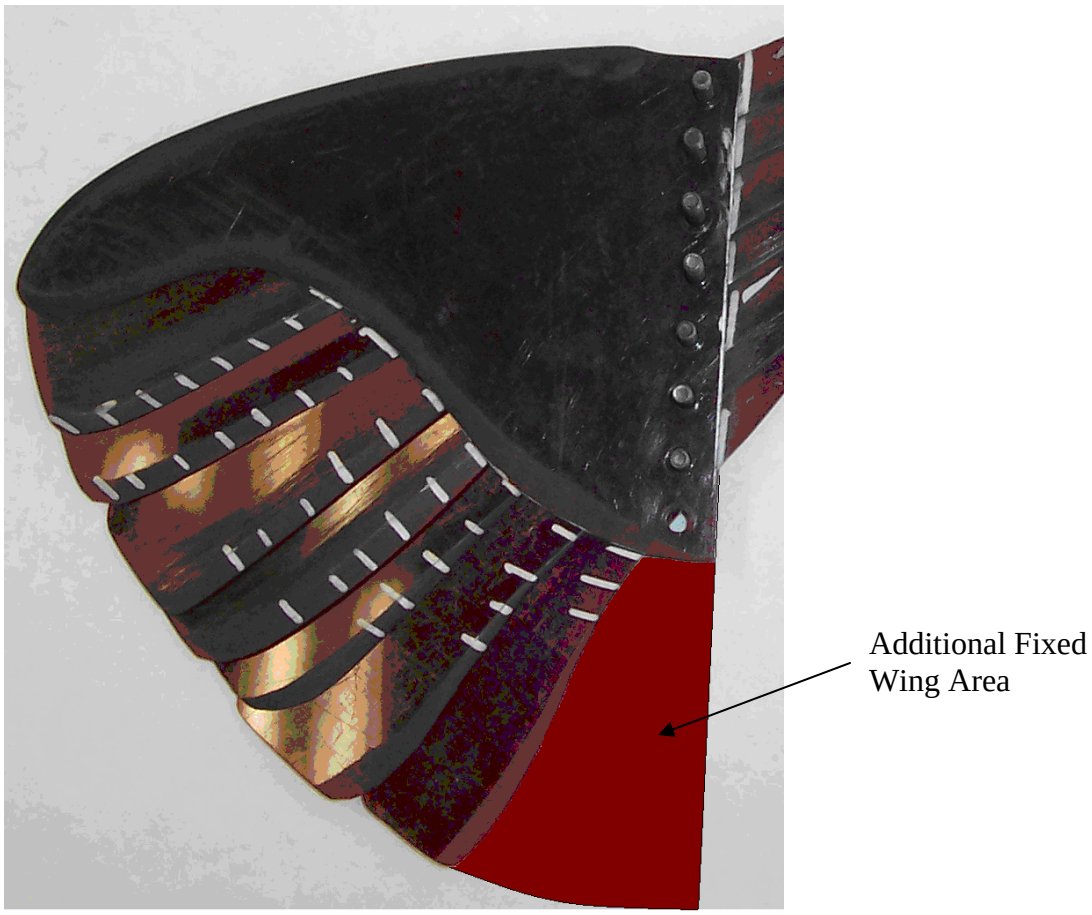

Figure 36. Test Wing with Additional Area

However, differences in wing area and root wing chordlength cannot completely account for the L/D discrepancy. Additionally, the test wing was not an exact flat plate as was the case in the published data. This was due to the overlapping lexan feathers that fit into the feather's housing. This configuration also increased the thickness to chord ratio of the wing in excess of 3.6% as opposed to the published 1.96% in Torres and Mueller[34], an 84% increase. A thicker wing will yield a higher drag coefficient and consequently a lower L/D value. Even streamlining the feathers so that no blunt edges or steps from feather to feather were present, there was no way to avoid these increases in drag. To confirm the drag increase a comparison of drag coefficients of the test wing's Zimmerman representation with published Zimmerman data is shown in Figure 37.

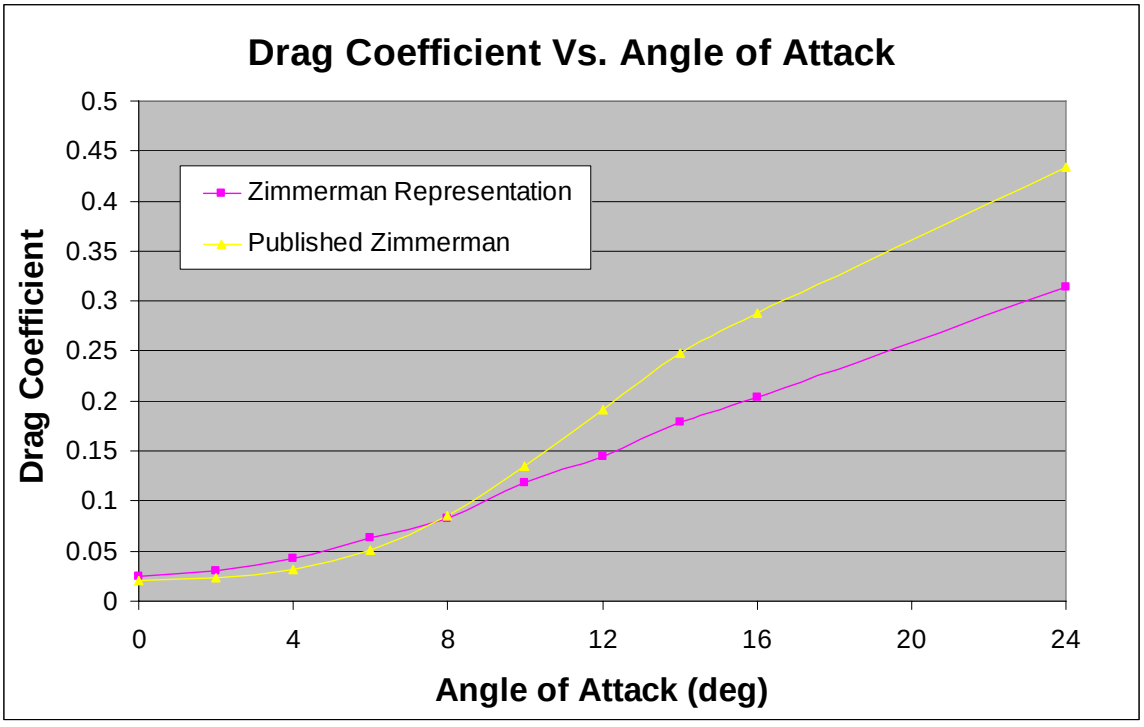

Figure 37. Published Drag Coefficients for Zimmerman Planform and Test Wing Representation

One can see the slight increase in the drag coefficient at low angles of attack. This slight difference is actually comparatively large when L/D is calculated. At 4 degrees there is a measured 32% increase in the drag coefficient for the Zimmerman representation. This angle of attack is very close to the test angle of attack of 4.6 degrees. In addition Figure 37 shows that at larger angles of attack, the published Zimmerman wing has a higher drag coefficient. One can imagine that the small area missing from the test wing representation may have decreased the amount of drag experienced by the test wing.

The published data in Torres and Mueller[34] was taken at a Reynolds number of 100,000 whereas this study occurred at Reynolds numbers ranging from approximately 160,000 to 180,000. A difference in Reynolds number could create slightly different L/D values for the same wing. Published data on the Zimmerman planform shows that in general there is a small increase in L/D when decreasing the Reynolds number from 100,000 to 70,000. Perhaps lowering the Reynolds number from the current range to approximately 100,000 would increase the L/D values of the test wings.

Because the test wing is a half wing there is also the possibility of some interaction between the wing tip vortices and the wall. The wing tip vortices are created at the wing tip due to the difference in pressure on the upper and lower surfaces of the wing. If these wing tip vortices were large enough, there might be a slight difference in the L/D values had the wing been a full wing placed in the center of the wind tunnel in order to eliminate this potential effect. However, other wind tunnel investigations have also used half-span MAV wings near test section walls.

3.4 Comparison of Best Wing to Zimmerman Representation

Next we will compare the best wing to the Zimmerman representation as an appropriate reference point. One can clearly see the improved performance over a wide range of angles of attack achieved by the best solution (Figure 30). Comparing the drag coefficients in Figure 39 one can see the two are extremely similar. In Figure 40 one sees that the best solution has a higher lift coefficient, accounting for the slight increase in L/D.

Also present in Figure 38 is L/D data for a recent MAV developed at the University of Florida[35]. Similar to the published data on the Zimmerman planform, data from the University of Florida shows significantly higher L/D values over large ranges of angles of attack. However, this planform is cambered to provide additional lift and uses wingtip endplates to diminish the effects of induced drag. Furthermore, the missing root wing area shown in Figure 36 and thickness effects increase the discrepancy in L/D.

In future work the test wing might be modified to incorporate camber. Once an optimal wing is found it should be constructed as a true cambered plate as opposed to the feather approach used in the work. These cambered plates could be full-span wings placed in the center of the wind tunnel attached to a separate force balance (Blanchard, et al.[25]) to eliminate any potential interaction from the wall. L/D data from these wings could then be compared fairly to the data presented from the University of Florida.

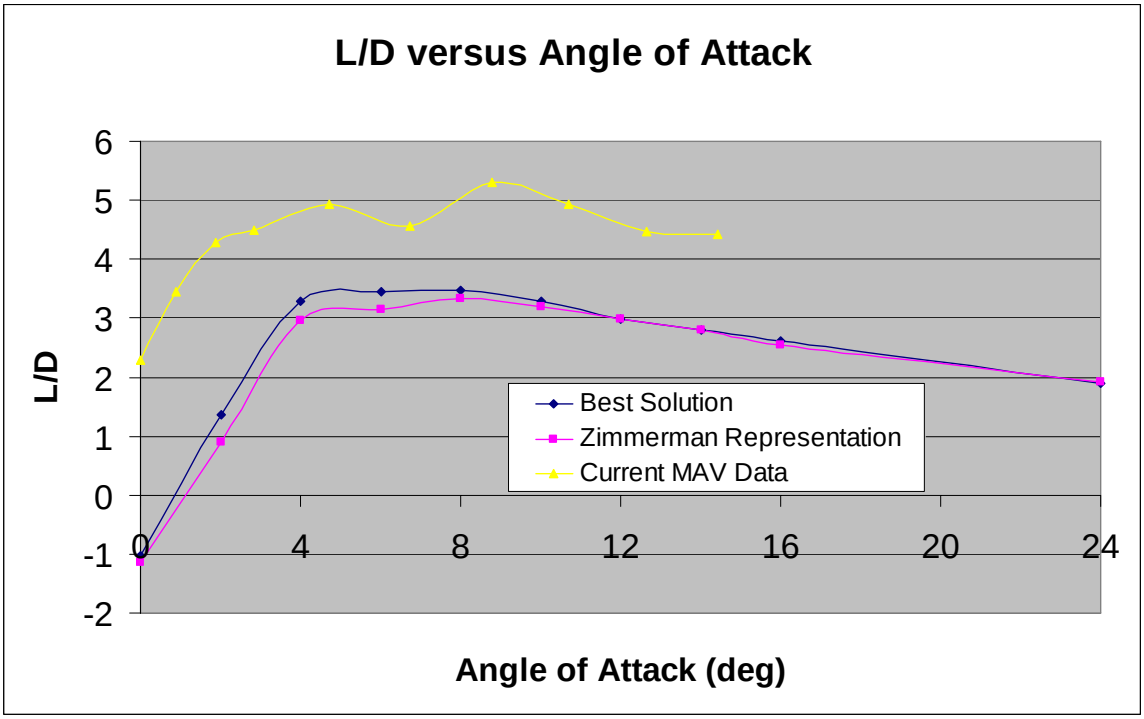

Figure 38. L/D versus Angle of Attack for the Best Solution and the Zimmerman Representation

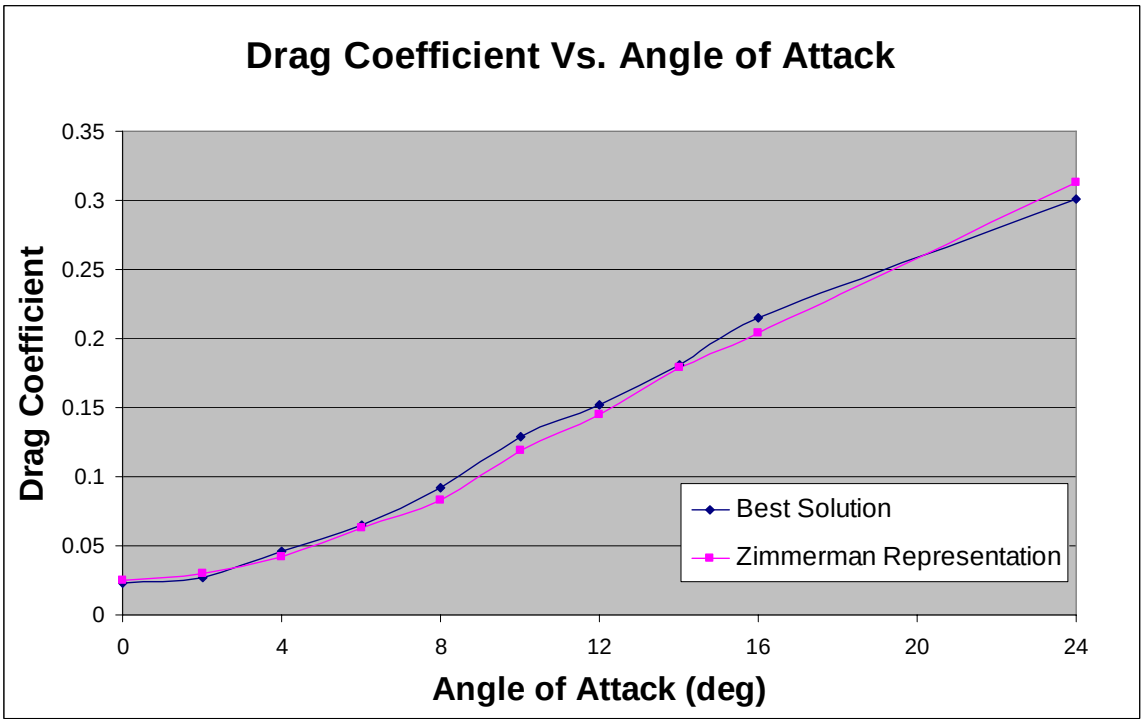

Figure 39. Drag Coefficient versus Angle of Attack for the Best Solution and the Zimmerman Representation

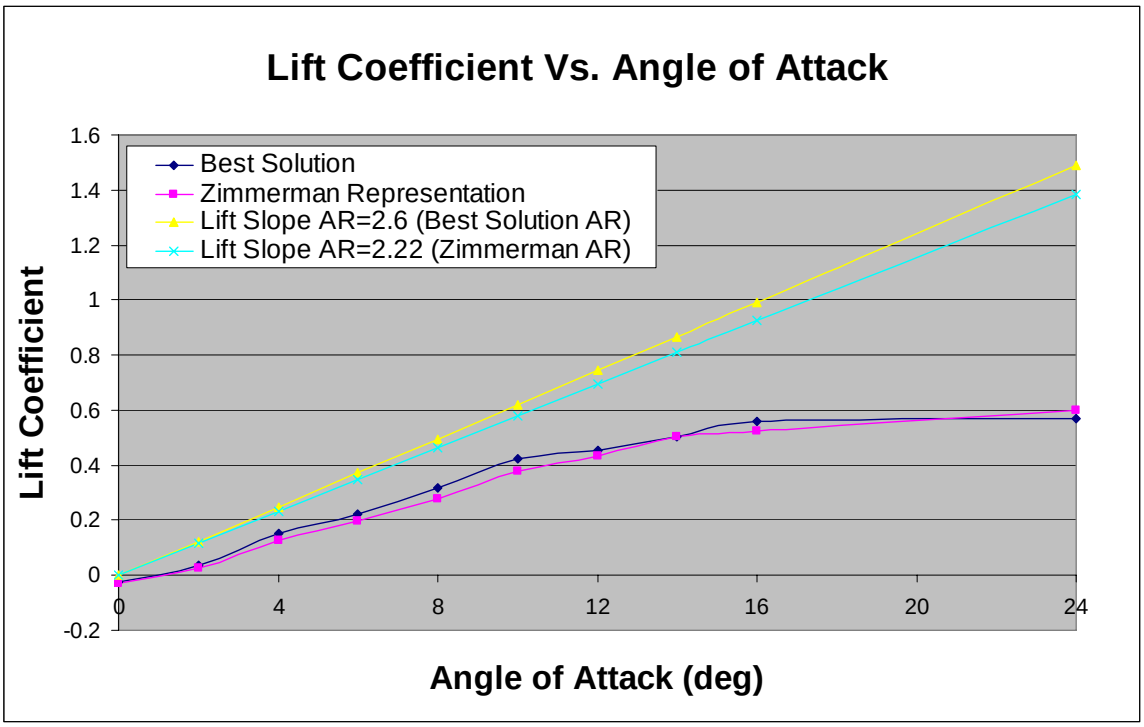

Figure 40. Lift Coefficient versus Angle of Attack for the Best Solution and the Zimmerman Representation

Figure 40 also shows two lines representing the lift coefficients of two different aspect ratio wings according to flat plate finite wing theory. Flat plate finite wing theory predicts both lift coefficients over all ranges of angles of attack that are higher than those measured with the test wing. Because flat plate theory uses an elliptically shaped wing, which yields the maximum amount of lift, it is not surprising that the best solution and Zimmerman representation yield lower lift coefficient values.

Additionally, the large discrepancy between the theoretical lift coefficients and the test lift coefficients brings into question the validity of the theoretical lift coefficient equations at low Reynolds numbers. As discussed earlier, low Reynolds numbers can have an impact on the lift and drag coefficients. It is possible that although the theory

claims to encompass all Reynolds numbers, perhaps it is actually invalid at the low Reynolds numbers experienced in this work.

A standard lift coefficient versus drag coefficient curve is shown in Figure 41. One can see the parasitic drag, the portion of drag created at no lift, which is approximately .0233 for the best solution. In addition a plot of L/D of 3.28 for the best wing shows that from approximately 4-10 degrees, L/D remains nearly constant.

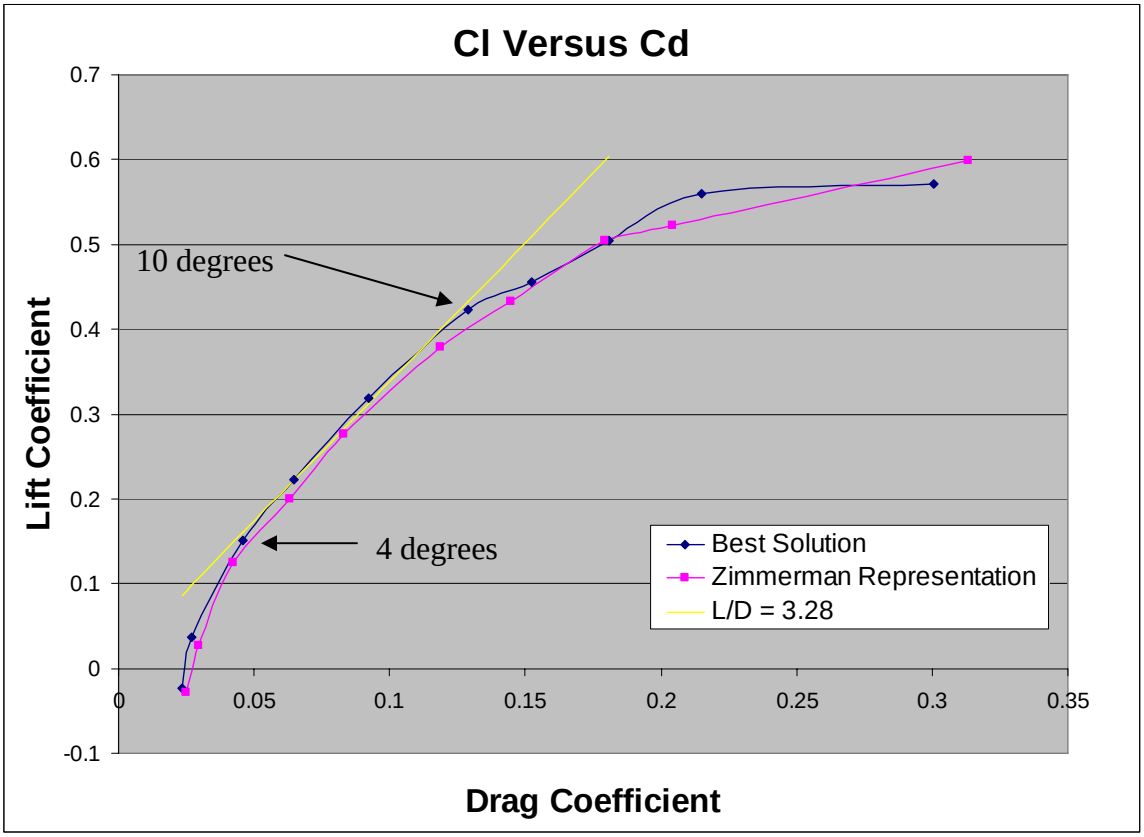

Figure 41. Lift Coefficient versus Drag Coefficient for the Best Solution and the Zimmerman Representation

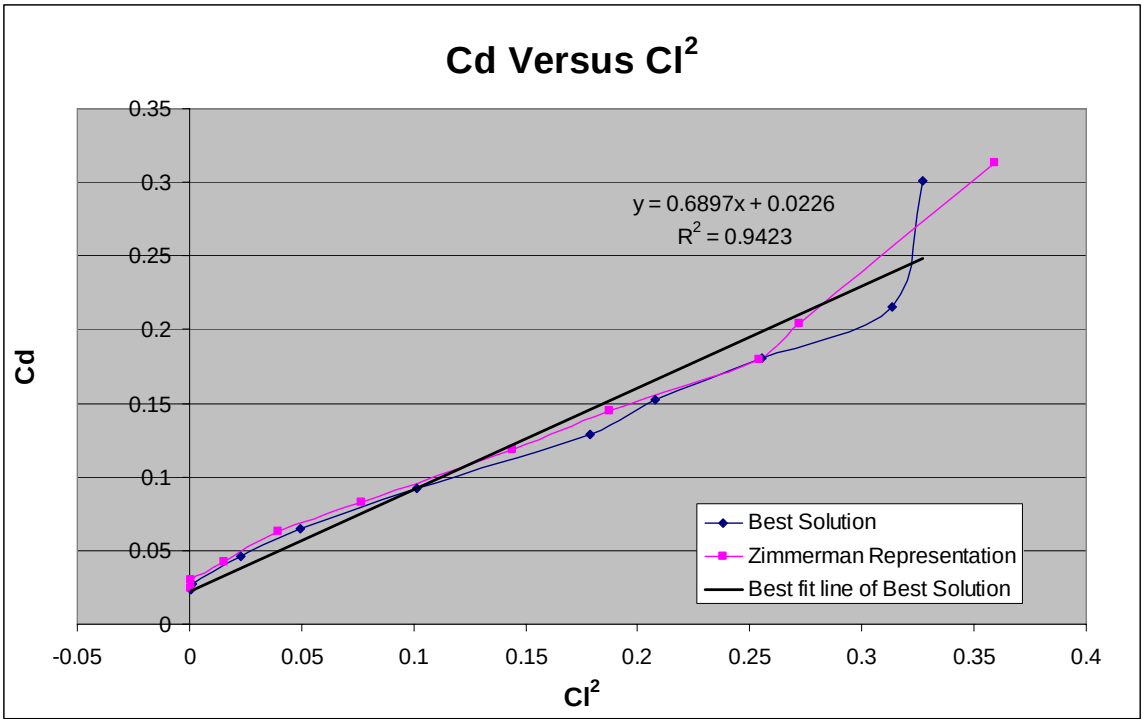

Figure 42. Cl^2 versus Cd for Best Wing and Zimmerman Representation

Figure 42 shows the lift coefficient squared versus the drag coefficient. This plot is useful in determining the Oswald efficiency factor. The slope, taken as $K_0 = 0.6897$ from a linear regression of the best solution, can be used in the following relation between the slope and the Oswald efficiency factor:

$$K_0 = \frac{1}{e_0 \pi AR} \tag{13}$$

In this equation e_0 is the Oswald efficiency factor. With and aspect ratio of 2.6 the Oswald efficiency factor is 0.177. This value is comparatively lower than standard values of between 0.5 and 1.0. Due to the lower lift coefficients achieved by the test wing in Figure 40 one would expect the slope of the curve in Figure 42 to increase. From equation 13 as the slope K_0 increases the Oswald efficiency factor e_0 decreases possibly accounting for the lower than expected value of e_0. In Torres and Mueller an aspect ratio

of 2 yields an Oswald efficiency number of 0.43, a reasonable value for MAV wings[34]. If the lift coefficients in Figure 40 were larger as theory predicted, perhaps the Oswald efficiency factor would more closely resemble the published value.

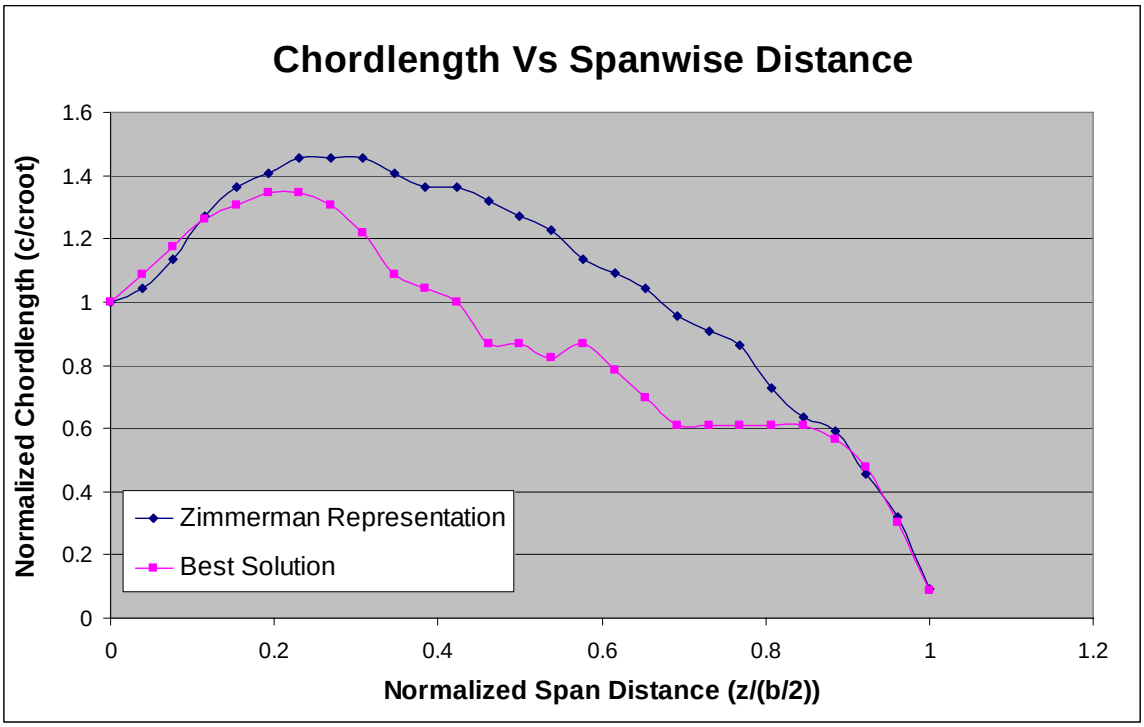

Figure 43. Local Chordlength versus Normalized Span Distance for Best Wing Planform

Figure 43 shows the local chordlength versus the normalized span for the best wing planform found from the genetic algorithm as compared to the Zimmerman representation. One can see the large difference in wing shape. While the best wing shape closely follows the shape of the Zimmerman near the wing tip and the wing root, it diverges in the middle. There is also an interesting region from $0.7 < \frac{z}{b/2} < 0.85$ where the chordlength remains constant. Although not as pronounced, there is a smaller constant region from $0.45 < \frac{z}{b/2} < 0.6$. A possible result of this study may be that wings with

constant chordlength regions near the wingtip yield higher L/D values, but this requires further study.

3.5 Discussion

Next we will conclude the results section with a general discussion of issues encountered that may have had an effect on the wing performance. In this section possible changes in measured L/D values due to changes in aspect ratio and Reynolds number will be estimated through calculations based on simple aerodynamic theory and previously published work. Numerical values for important aerodynamic parameters, such as parasitic drag coefficient, will be extracted from the measured data in the present study when necessary. Additionally, the effect of small gaps in the feather lengths will be investigated.

The aspect ratio of the wing changed with every individual tested. It is possible that this made a large enough impact such that the differences in L/D values attained from the changing planform shapes were merely a result of the changing aspect ratio. From Figure 40 flat plate theory tells us that there should be an increase of approximately 7% when increasing the aspect ratio from 2.22 to 2.6 for all angles of attack. However, the difference in the lift coefficient between the best wing (AR=2.6) and the Zimmerman representation (AR=2.22) at 4 degrees is 21%, three times higher than the theoretical change due to the aspect ratio. This may imply the changes in lift coefficient values are not solely due to aspect ratio effects.

The aspect ratio also has an effect on the induced drag. The induced drag is given by equation 14.

$$C_{Di} = \frac{Cl^2}{e_0 \pi AR} \tag{14}$$

The total drag, which is a combination of the induced drag and the parasitic drag C_{D0} (the drag experienced when there is no lift) is given be equation 15. The value for C_{D0} was 0.0233 for the best wing (see Figure 41).

$$C_D = C_{D0} + \frac{Cl^2}{e_0 \pi AR} \tag{15}$$

Due to the dependence of these drag equations on the lift coefficient the aspect ratio had a varying effect on the calculated drag coefficients at different angles of attack. For angles of attack of 0-10 degrees this change remained less than 4%.

At and angle of attack of 4 degrees with a 7% calculated increase in lift coefficient and a 2% calculated increase in drag coefficient, there is a calculated 5% increase in L/D, approximately half of the difference obtained in the wind tunnel experiment between the Zimmerman representation and the best wing solution and 40% of the 12.6% increase in the average L/D values over the 12 generations in Figure 25. These estimates suggest that changes in L/D during the genetic algorithm study are largely due to planform shape variations and not due to variations in aspect ratio.

As previously mentioned in Section 3.3 there is a difference in Reynolds number between the best solution (Re=182,000) and the Zimmerman representation (Re=161,000) due to the change in the chordlength. This difference (11.5%) is much smaller than the difference in Reynolds number studied in Torres and Mueller[34], where a 30% decrease in Reynolds number was studied. Specifically, at 4 and 5 degrees there was an increase in L/D of 3% and a decrease 2% respectively in Torres and Mueller[34]. Because of the small magnitude of these changes and the comparatively smaller variation

of Reynolds number studied in this work, one would not expect to find a significant effect from the change in Reynolds number in this work.

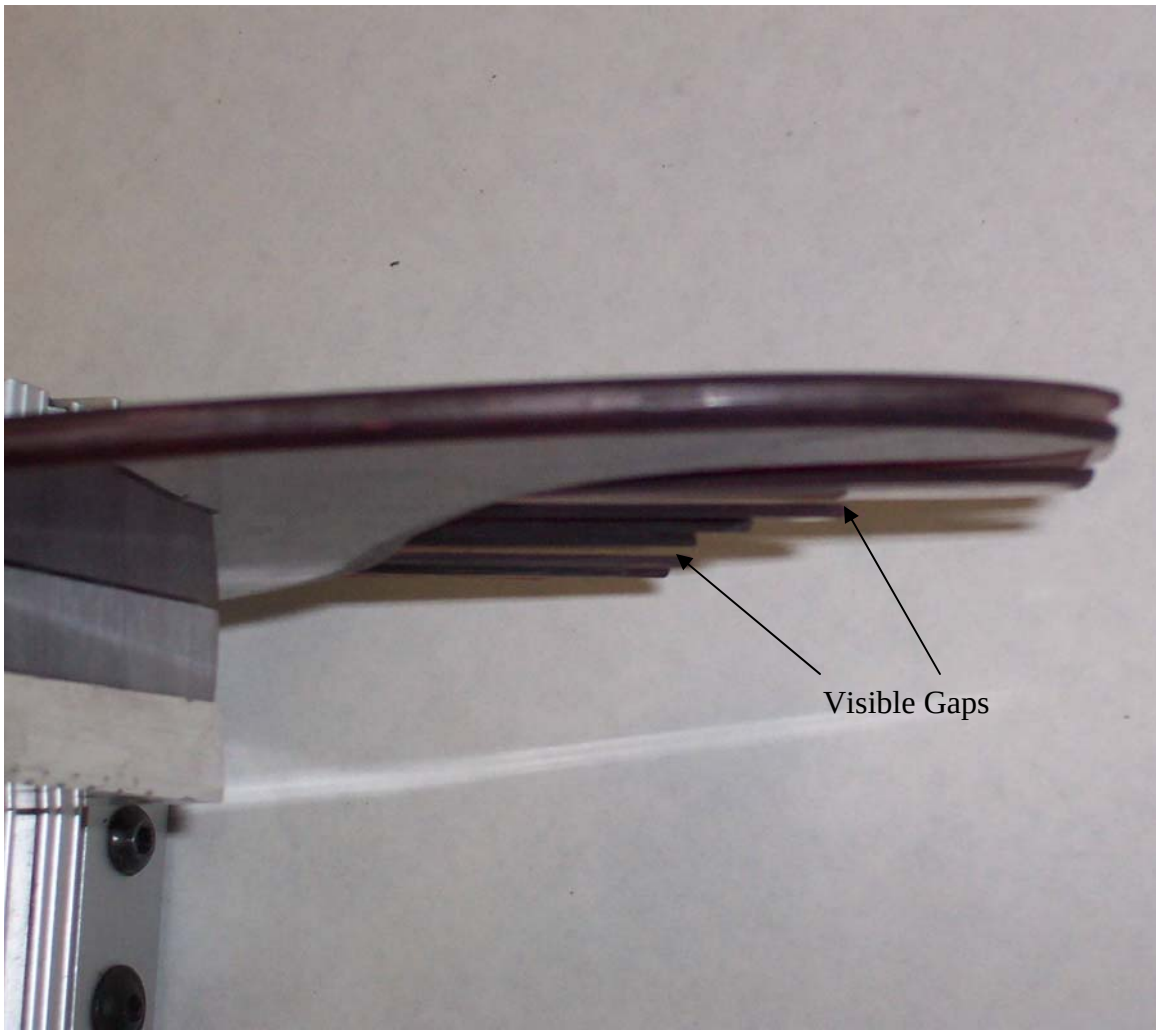

Figure 44. Visible Feather Gaps

Upon close observation of the test wing one can see that when looking in the direction of the flow there is a small vertical gap between some of the feathers shown in Figure 44, through which air could possibly flow. References 36 and 37 have studied these effects as they pertain to drag reduction in birds; however these studies incorporate much larger feather gap distances. There is a possible question of whether or not the small gaps seen in Figure 44 have an effect on the lift to drag ratio experienced. In order

to investigate this, small pieces of tape were used to close the gaps in between the feathers. The wing was tested five times in the wind tunnel under these conditions and compared to five tests without the tape.

From Table 4 there is a small increase in the average lift to drag of approximately 1.4%. One can see that both the lift and the drag are reduced once the tape is added. Overall however, the drag decreases enough to increase the lift to drag ratio. The amount of the increase is small and a within total standard deviation error bars as shown in Figure 45.

Table 4. Best Solution with and without Tape

	Without Tape	With Tape
Average L/D	3.281	3.314
Average Lift	47.980	45.960
Average Drag	14.684	13.939

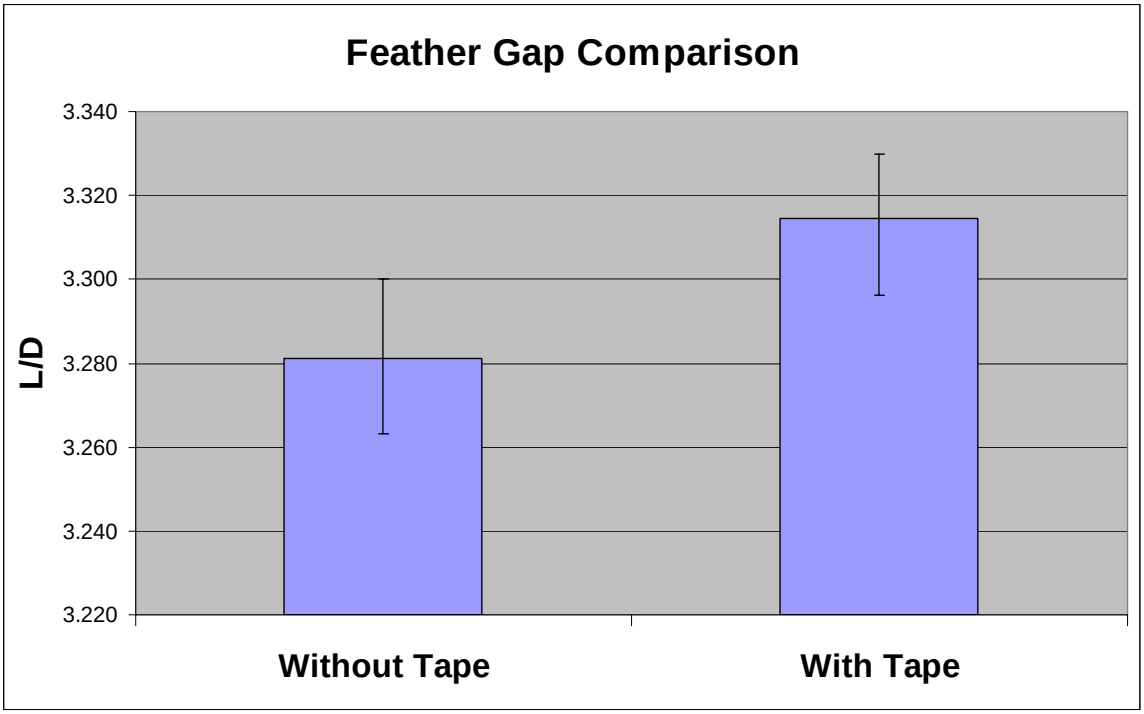

Figure 45. Comparion of Best Wing with and without Tape Covering Feather Gaps

Chapter 4: Conclusions

The main results of the study are summarized as follows:

- A wind tunnel study using genetic algorithms was conducted to optimize wing planforms shapes for MAV's.
- A novel 'feather wing' concept was conceived, fabricated and implemented. This feather wing used variable feather lengths to vary wing planform shapes on a half-span MAV wing.
- A single run of a developed genetic algorithm involving 12 generations and comprising of over 300 wing shape solutions that were experimentally tested in the wind tunnel was conducted. The genetic algorithm used the roulette wheel method, a modified random pairing method, random crossover, and a random mutation of 20% as a means of parent selection, pairing, mating, and child mutation rate.
- The genetic algorithm yielded a best wing with a peak L/D value of 3.28 at 4.6 degrees angle of attack. Typical L/D values for MAV's range from 4.0-7.0. The best individual resulted from minor mutations of a very strong individual present in the initial population. However, restrictions in the test wing fabrication made it difficult to fairly compare the best wing planform with previously published data. Specifically an area near the trailing edge root of the wing was not present. This region is responsible for a sizeable amount of lift. Additionally, the test wing was unable to represent a true flat plate due to the feather concept adopted.
- The genetic algorithm proved to be a strong optimization tool for experimentally obtained results. The increase in the average L/D values in successive generations

and the plateau reached showed that the wing was improving until the best planform was overwhelmingly dominant throughout the population.

- Due to the large number of wings tested, had a computational fluid dynamics approach been used, the setup and test time would have been much longer. Once testing began the progression of the genetic algorithm was relatively quick and the experimentally obtained results proved to be efficient, further justifying the use of experimentally obtained results while using a genetic algorithm.
- Important aerodynamic parameters of the wing were also calculated for the basis of comparison. The parasitic drag coefficient C_0 was 0.0233. The Oswald efficiency factor e_0 was 0.177.
- The best solution, although not a trailing edge notch, was an unprecedented shape that is currently not in use, containing a constant chordlength region near the wing tip. This underlines one of the strong points in the use of genetic algorithms in that it is not confined to any predisposed shape that design engineers may be accustomed to.
- Low L/D planform were also found. These planforms had large trailing edge gaps and large amplitude serrations. These planforms could be used in morphing wing technology. With micro-servos these morphing wings could employ a feather-wing concept to change planforms in flight.
- Due to the discrepancy between the published Zimmerman data and the test wing representation of the Zimmerman planform, further research is recommended in comparing the results found in the research to published data. One might make a true flat plate representation of a few of the strongest solutions found in this

research and tediously match the conditions in published work. This would eliminate the discrepancies and allow us to determine if indeed a stronger micro aerial vehicle planform has been achieved.

- More future work might include additional runs of the genetic algorithm with different random initial populations to determine if the best wing is indeed the global optimum and if any other interesting solutions appear.
- Studies of a multivariable fitness function should also be made. Here other aerodynamic performance parameters such as lift coefficient, quarter chord moment, etc could be combined with L/D to establish a more complex fitness function.
- Additionally, CFD studies may be conducted in the future to study the flow physics of the best wing solutions. Perhaps insight on how the constant chord region found near the wingtip affects flow dynamics leading to higher L/D values can be found.

References

1. Darwin, C., (2004). The Origin of Species, Barnes and Noble Books.

2. Mithchell, M., (1996). An Introduction to Genetic Algorithms, Cambridge: MIT Press.

3. Holland, J., (1992) Adaptation in Natural and Artificial Systems, Cambridge: MIT Press.

4. Haupt, R., and Haupt, S., (1998). Practical Genetic Algorithms, New York: John Wiley and Sons, INC.

5. Rao, S., (1998, January) Evolution at Warp Speed, *Forbes Magazine.*

6. Mathworks Matlab Genetic Algorithm and Direct Search Toolbox

7. Zhang, F., Chen, S., and Khalid, M., (2002) "Multi-Point Optimization of Transonic Wing by Real-Coded Genetic Algorithm," Institute for Aerospace Research.

8. Quagliarella, D., and Cioppa, A., "Genetic Algorithms Applied to the Aerodynamic Design of Transonic Airfoils," *Journal of Aircraft*, 15, 889-890.

9. Hacioglu, A., and Ozkol, I., "Transonic Airfoil Design and Optimisation by Using Vibration Genetic Algorithm," *Aircraft Engineering and Aerospace Technology*, 75, 350-357.

10. Jones, B., Crossley, W., and Lyrintzis, A., "Aerodynamic and Aeroacoustic Optimization of Rotorcraft Airfoils via a Parallel Genetic Algorithm," *Journal of Aircraft*, 37, 1088-1096.

11. Marta, A., "Parametric Study of a Genetic Algorithm Using a Aircraft Design Optimization Problem," Stanford University, Department of Aeronautics and Astronautics.

12. Liu, J., "Intelligent Genetic Algorithm and Its Application to Aerodynamic Optimization of Airplanes," *AIAA Journal*, 43, 530-538.

13. Blasi, L., "Conceptual Aircraft Design Based on a Multiconstraint Genetic Optimizer," Journal of Aircraft, 37, 351-354.

14. Cox, L.A., Jr., Davis, L. & Qiu, Y., (1991) "Dynamic Anticipatory Routing in Circuit-Switched Telecommunication Networks," *Handbook of Genetic Algorithms*, 124-143.

15. Noren, K., "Analog Circuit Design Using Genetic Algorithms," University of Idaho.

16. Altshuler, E., & Linden, D., (1997) “Wire-Antenna Designs Using Genetic Algorithms,” *IEEE Antennas Propagation Magazine*, 39 33-43.

17. Lucasius, C., (1990) “Conformational Analysis of DNA Using Genetic Algorithms,” *Lecture Notes in Computer Science*, 496 90-97.

18. Levin, M., (1993) “Application of Genetic Algorithms to Molecular Biology: Locating Putative Protein Signal Sequences,” Harvard Medical School.

19. Brinkman, D., (1993) “Genetic Algorithms and Their Application to the Protein Folding Problem,” Air Force Institute of Technology.

20. Deerman, K., (1999) “Protein Structure Prediction Using Parallel Linkage Investigating Genetic Algorithms,” Air Force Institute of Technology.

21. Gates, G., (1994) “Predicting Protein Structure Using Parallel Genetic Algorithms,” Air Force Institute of Technology.

22. Bauer, R., (1994). Genetic Algorithms and Investment Strategies, John Wiley & Sons.

23. Chen, S., (2002) Genetic Algorithms and Genetic Programming in Computational Financing. Massachusetts, Kluwer Academic Publishers.

24. Drovetski, S., (1996). “Influence of the Trailing-Edge Notch on Flight Performance of Galliforms,” *The AUK- Journal of American Ornithologists’ Union*, 113(4), 802-810.

25. Levshin, A., & Custodio, D., “Effects of Leading Edge Protuberances on Airfoil Performance,” AIAA Conference Paper.

26. Bushnell, D., (1991) “Drag Reduction in Nature,” *Annual Review of Fluid Mechanics* 23, 65-79.

27. Vijgen, P., Van Dam, C., Holmes, B., & Oward, F., “Wind Tunnel Investigations of Wings with Serrated Sharp Trailing Edges,” *Conference on Low Reynolds Number Aerodynamics*, University of Notre Dame, 295-313.

28. Schwind, R., & Allen, H., “The Effects of Leading Edge Serrations on Reducing Flow Unsteadiness about Airfoils,” Nielsen Engineering & Research, Inc.

29. Dassen, T., Parchen, R., Bruggeman, J., & Hagg, F., (1996) “Results of a Wind Tunnel Investigation on the Reduction of Airfoil Noise by the Application of Serrated Trailing Edges.” *European Union Wing Energy Conference and Exhibition*, National Aerospace Laboratory.

30. Burnett, B., (2005) “A Boeing-led Team is working to Make Quiet Jetliners Even Quieter,” *Boeing Frontiers*, 04.

31. Li, Y., Wang, J., Tan, G., & Zhang, P., (2002) “Effects of Gurney Flaps on the Lift Enhancement of a Cropped Nonslender Delta Wing,” *Experiments in Fluids*, 32 99-105.

32. Popp, K. “Experimental Investigation of Reynolds Number and Scale Effects on Parachute Inflation”, M.S. Thesis, Worcester Polytechnic Institute, 2000.

33. Blanchard, S., DeFusco, D., & Donoghue, Chris. (2006) “Experiments on Trailing Edge Notches for Micro Air Vehicle Wings,” Worcester Polytechnic Institute.

34. Torres, G., & Mueller, T., (2001)“Aerodynamic Characteristics of Low Aspect Ratio Wings at Low Reynolds Numbers” Progress in Astronautics and Aeronautics, 194 115-141

35. Vieru, D., Albertani, R., Shyy, W., & Ifju, P., “Effects of Tip Vortex on Wing Aerodynamics of Micro Air Vehicles” Journal of Aircraft, 42 1530-1536

36. Bannasch, R., (2001)“From Soaring and Flapping Bird Flight to Innovative Wing and Propeller Constructions” Progress in Astronautics and Aeronautics, 194 453-471

37. Tucker, V., (1994) “Drag Reduction by Wing Tip Slots in a Gliding Harris’ Hawk, Parabuteo Unicinctus” Journal of Experimental Biology, 198 775-781

Appendices

Appendix A: MatLab Program Description

A Matlab script was created in order to carry out the genetic algorithm. In addition, a small Matlab script was used to create the initial population. The code for each program as well as an explanation is presented in the following section.

InitialPopulationProgram.m

```
popsize=72;
POP=rand(popsize,7);
A=[8, 8, 6, 6, 5, 4, 3];
for i=1:7
    POP(:,i)=A(i)*POP(:,i);
end
InitPopulation=floor(POP);
xlswrite('InitialPopulation.xls', InitPopulation)
```

This is a short program that makes the initial population. The variable “popsize” is the number of individuals in the population. The second line makes a two dimensional array with “popsize” number of rows and seven columns, one for each gene of an individual. The vector A is the total number of increments for each feather that was previously shown in Table 3. The population is comprised of random numbers. By default a random number generator returns a value between zero and one. By multiplying the random number by the appropriate value of vector A, and using the floor function to truncate the value, the correct integer values were obtained for each gene position. Finally, the initial population is exported to an Excel spreadsheet with a portion of the sheet shown in Table 5. On this spreadsheet it is necessary to assign a fitness value in the last column. This is the lift to drag ratio minus a constant. Once these values are added to the spreadsheet, a different program will create the next generation of individuals.

Table 5. Blank Initial Population Spreadsheet

	Position Number							Fitness
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								
41								
42								
43								
44								
45								

46								
47								
48								
49								
50								
51								
52								
53								
54								
55								
56								
57								
58								
59								
60								
61								
62								
63								
64								
65								
66								
67								
68								
69								
70								
71								
72								

GeneticAlgorithm.m

```
close all
clear all
PopNum=24;
nParents=PopNum/2;
A=[8, 8, 6, 6, 5, 4, 3];
numgen=input('Enter the generation worksheet number for you current
population\n')
expectation = xlsread('Populations.xls', numgen, 'I2:I25');%reads
fitness values
parents = selectionroulette(expectation,nParents)

OldPop=xlsread('Populations.xls', numgen, 'B2:H25')
NewPop=zeros(PopNum,7);
for i = 1:nParents
    n=parents(i);
NewPop(i,:)=OldPop(n,:);
end

for k = 1:nParents/2
for i = 1:2
```

```
    for j = 1:7
        r=rand;
        if(r < .5)
            NewPop(nParents+2*(k-1)+i,j) = NewPop(2*(k-1)+1,j);
        else
            NewPop(nParents+2*(k-1)+i,j) = NewPop(2*(k-1)+2,j);
        end
        r=rand;
        if(r<.2)
            r=floor(rand*A(j));
            NewPop(nParents+2*(k-1)+i,j) =r;
        end

    end
end
end
numnewgen=numgen+1
xlswrite('Populations.xls', NewPop, numnewgen, 'B2:H25');
```

This is the main program. First the program asks what generation number you are currently working with. In the spreadsheet each sheet in the workbook is a different generation. The first sheet is the first generation and so on. The program then reads the fitness values and sets them to the array “expectation”. The parents are then selected using a roulette wheel program taken from Matlab’s preexisting Genetic Algorithm and Direct Search Toolbox called “selectionroulette.m”. An explanation of that code is shown later. The roulette wheel program returns the numbers of the chromosomes selected as parents. Using a “for loop” the program then takes the existing population and copies the parents into the next generation, occupying half the vacancies. The next loop takes each pair of parents and through a random number generator determines which parent contributes each gene. After this for loop is another for loop to account for the chance of mutation in every gene. Lastly the parents and their new offspring are exported to the next sheet in the spreadsheet as the next generation.

It is important to note that the random number generator is a deterministic program. Unfortunately, this means that the random numbers generated aren’t entirely random. When starting Matlab to create the next population if I were to only run the

genetic algorithm once, it would give me the same parents and children every time. In order to accommodate for this two dice were rolled to determine how many times to run the genetic algorithm before the next generation given was the generation used.

The selection roulette program mentioned previously is short but very clever.

`Selectionroulette.m`

```
function parents = selectionroulette(expectation,nParents)

wheel = cumsum(expectation)/sum(expectation);
parents = zeros(1,nParents);
for i = 1:nParents
    r = rand;
    for j = 1:length(wheel)
        if(r < wheel(j))
            parents(i) = j;
            break;
        end
    end
end
```

The program first takes the fitness values and calculates the cumulative sums. A cumulative sum takes a series of numbers and adds all the numbers before it in the series to get the cumulative sum. A simple example is shown in Table 6.

Table 6. Cumulative Summation Example

series	cumsum
1	1
2	
3	
4	
5	

series	cumsum
1	1
2	3
3	
4	
5	

series	cumsum
1	1
2	3
3	6
4	
5	

series	cumsum
1	1
2	3
3	6
4	10
5	

series	cumsum
1	1
2	3
3	6
4	10
5	15

The cumulative sum is then normalized by dividing by the sum of the fitness values. Table 7 shows the series after is has been normalized. Additionally, a pie chart shown in Figure 46 represents the roulette wheel created from this example.

Table 7. Normalized Cumulative Summation

Individual	Fitness Value	cumsum
1	1	0.067
2	2	0.2
3	3	0.4
4	4	0.667
5	5	1

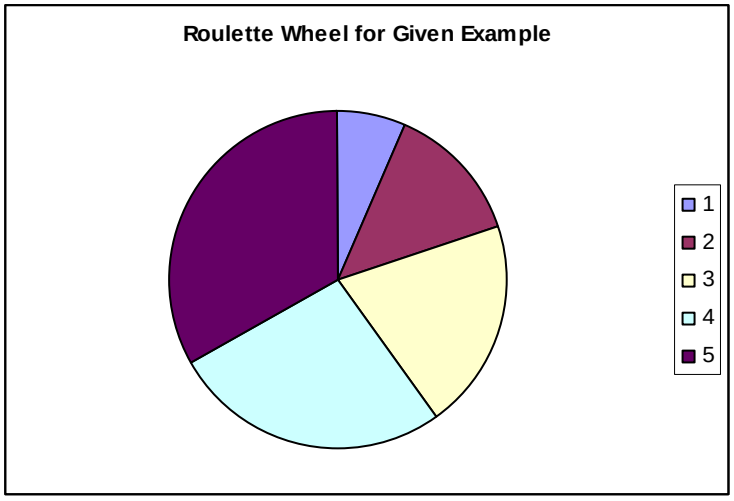

Figure 46. Cumulative Sum Roulette Wheel

A random number is chosen to determine which parent is selected. If the random number falls between the appropriate cumulative sums, that parent is selected. For instance from Table 7, if the random number was between 0.000 and 0.067 then the first parent is selected. If the number is between 0.667 and 1.000 then the fifth parents is chosen. One can see that with the series given, the fifth parent does indeed have the best chance of being chosen, followed by the fourth, etc.

Appendix B: Wing Generation Data

Generation 1

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	7	6	0	5	3	3	2	0.17	2.771	12.772	12.344	14.689	11.8	28.9	40.7
2	1	4	1	3	2	2	2	0.39	2.986	12.815	12.379	15.303	14.4	31.3	45.7
3	4	2	3	5	0	2	2	0.17	2.773	12.788	12.372	14.280	11.6	28.0	39.6
4	3	5	3	0	3	0	1	0.35	2.954	12.797	12.375	14.761	13.3	30.3	43.6
5	7	4	2	0	1	1	0	0.36	2.960	12.766	12.379	13.546	12.2	27.9	40.1
6	6	3	3	4	0	1	1	0.16	2.756	12.771	12.375	13.570	10.7	26.7	37.4
7	3	5	2	2	2	0	2	0.45	3.053	12.808	12.378	15.199	14.9	31.5	46.4
8	0	4	0	5	4	3	2	0.22	2.820	12.825	12.385	15.177	12.8	30.0	42.8
9	6	6	0	4	3	3	0	0.28	2.881	12.799	12.37	14.892	13.3	29.6	42.9
10	3	7	1	4	4	1	2	0.23	2.830	12.812	12.374	15.124	12.3	30.5	42.8
11	4	4	0	2	4	3	1	0.18	2.784	12.796	12.381	14.261	12.7	27.0	39.7
12	6	7	2	0	0	0	2	0.50	3.101	12.774	12.381	13.963	13.6	29.7	43.3
13	7	1	4	0	1	3	2	0.11	2.709	12.775	12.381	13.436	9.9	26.5	36.4
14	5	7	0	1	2	3	0	0.42	3.022	12.795	12.379	14.657	13.7	30.6	44.3
15	1	2	0	2	1	1	1	0.41	3.012	12.803	12.403	14.078	13.0	29.4	42.4
16	3	2	3	5	2	2	0	0.24	2.844	12.799	12.375	14.662	12.5	29.2	41.7
17	7	7	3	2	0	3	2	0.35	2.949	12.786	12.365	14.718	13.1	30.3	43.4
18	7	5	0	4	2	2	1	0.27	2.870	12.777	12.375	13.939	11.8	28.2	40.0
19	3	1	0	2	3	0	0	0.26	2.859	12.806	12.403	13.957	11.4	28.5	39.9
20	7	0	1	2	4	1	2	0.05	2.646	12.775	12.379	13.417	8.9	26.6	35.5
21	0	7	3	2	3	0	2	0.27	2.866	12.819	12.379	15.250	12.9	30.8	43.7
22	2	1	0	2	3	1	0	0.33	2.925	12.803	12.407	13.810	11.8	28.6	40.4
23	6	2	2	5	2	1	0	0.14	2.745	12.784	12.388	13.554	10.3	26.9	37.2
24	0	5	3	0	3	1	2	0.31	2.907	12.824	12.397	14.863	13.1	30.1	43.2
25	1	2	4	1	4	2	0	0.49	3.085	12.777	12.395	13.549	12.3	29.5	41.8
26	1	3	4	0	0	0	2	0.39	2.991	12.803	12.407	13.907	12.6	29.0	41.6
27	1	0	0	4	4	1	2	0.18	2.781	12.826	12.399	14.670	11.5	29.3	40.8
28	4	7	2	3	4	2	2	0.39	2.990	12.815	12.374	15.485	14.1	32.2	46.3
29	2	4	2	5	3	0	0	0.33	2.927	12.821	12.389	15.068	13.2	30.9	44.1
30	1	3	2	3	1	2	1	0.39	2.989	12.824	12.4	14.887	14.3	30.2	44.5
31	0	4	0	2	4	0	0	0.34	2.942	12.801	12.403	13.904	11.6	29.3	40.9
32	5	2	4	1	2	2	2	0.23	2.827	12.804	12.391	14.256	12.1	28.2	40.3
33	3	3	4	3	3	2	1	0.34	2.942	12.816	12.387	14.988	13.6	30.5	44.1
34	7	1	4	4	2	1	0	0.30	2.902	12.798	12.388	14.264	12.6	28.8	41.4
35	3	4	2	2	4	2	1	0.28	2.881	12.818	12.387	14.961	12.6	30.5	43.1
36	3	6	3	0	0	3	2	0.33	2.929	12.805	12.392	14.409	12.7	29.5	42.2
37	6	4	0	2	2	1	1	0.40	3.004	12.787	12.396	13.749	12.8	28.5	41.3
38	4	5	2	4	1	1	0	0.48	3.083	12.811	12.392	14.857	14.6	31.2	45.8
39	1	1	4	4	4	1	0	0.27	2.866	12.848	12.398	15.597	13.4	31.3	44.7
40	5	3	5	5	1	1	0	0.33	2.933	12.803	12.383	14.660	13.6	29.4	43.0
41	6	6	1	5	3	1	1	0.33	2.927	12.814	12.383	15.033	13.4	30.6	44.0
42	0	5	1	2	0	1	0	0.52	3.125	12.817	12.41	14.497	14.7	30.6	45.3
43	5	3	5	3	1	2	1	0.21	2.811	12.799	12.386	14.232	11.5	28.5	40.0
44	3	4	1	4	2	2	0	0.51	3.108	12.803	12.394	14.542	14.6	30.6	45.2
45	6	6	4	1	2	1	2	0.38	2.980	12.801	12.383	14.662	13.5	30.2	43.7
46	4	0	5	5	2	0	1	0.15	2.751	12.832	12.399	14.830	11.7	29.1	40.8
47	5	4	1	3	3	1	2	0.31	2.907	12.818	12.399	14.585	12.7	29.7	42.4
48	3	0	1	3	0	1	0	0.24	2.836	12.815	12.417	13.752	11.4	27.6	39.0
49	2	3	0	1	4	0	1	0.18	2.782	12.819	12.41	14.053	10.6	28.5	39.1
50	1	2	0	3	3	2	2	0.30	2.897	12.831	12.411	14.603	12.7	29.6	42.3
51	1	6	3	5	3	2	0	0.30	2.895	12.846	12.391	15.818	14.0	31.8	45.8
52	5	0	1	2	4	1	2	0.09	2.686	12.802	12.408	13.404	9.1	26.9	36.0
53	2	6	5	3	3	1	0	0.29	2.891	12.805	12.39	14.422	12.4	29.3	41.7
54	4	7	1	2	1	2	0	0.60	3.204	12.818	12.407	14.764	15.7	31.6	47.3
55	1	7	1	3	1	3	0	0.60	3.196	12.807	12.406	14.393	15.2	30.8	46.0
56	5	6	5	4	3	3	2	0.29	2.890	12.837	12.381	15.845	14.1	31.7	45.8
57	3	3	2	2	2	1	0	0.56	3.156	12.809	12.41	14.259	14.8	30.2	45.0
58	6	3	2	2	0	3	0	0.32	2.915	12.787	12.403	13.378	11.7	27.3	39.0
59	6	1	1	3	0	1	2	0.15	2.746	12.799	12.415	13.145	10.0	26.1	36.1
60	4	5	2	5	1	3	0	0.31	2.911	12.831	12.399	15.044	13.1	30.7	43.8
61	3	2	2	2	2	3	1	0.28	2.875	12.819	12.413	14.085	12.0	28.5	40.5
62	7	7	3	2	2	0	0	0.63	3.231	12.813	12.393	15.133	16.7	32.2	48.9
63	6	5	0	3	4	1	2	0.23	2.833	12.825	12.403	14.577	12.3	29.0	41.3
64	5	3	0	3	2	2	2	0.23	2.831	12.817	12.412	13.987	11.6	28.0	39.6
65	6	5	2	4	1	2	2	0.35	2.954	12.819	12.4	14.657	13.3	30.0	43.3
66	5	2	5	3	2	2	1	0.22	2.824	12.829	12.401	14.769	12.6	29.1	41.7
67	2	3	5	4	2	1	0	0.37	2.970	12.847	12.402	15.592	14.8	31.5	46.3
68	2	7	1	2	0	1	0	0.49	3.087	12.832	12.411	14.935	15.1	31.0	46.1
69	2	5	0	1	4	0	0	0.34	2.937	12.811	12.416	13.792	11.6	28.9	40.5
70	4	1	5	4	4	1	0	0.18	2.781	12.848	12.399	15.426	12.5	30.4	42.9
71	5	6	1	5	2	1	2	0.34	2.938	12.818	12.396	14.737	13.0	30.3	43.3
72	2	5	3	4	0	1	2	0.40	2.999	12.832	12.407	14.937	14.1	30.7	44.8

Generation 2

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	3	5	2	2	2	0	2	0.31	3.111	12.828	12.408	14.849	15.0	31.2	46.2
2	0	5	3	0	3	1	2	0.13	2.926	12.835	12.413	14.629	12.9	29.9	42.8
3	2	5	3	4	0	1	2	0.17	2.974	12.822	12.407	14.459	13.0	30.0	43.0
4	3	3	4	3	3	2	1	0.18	2.978	12.855	12.410	15.512	15.1	31.1	46.2
5	3	3	2	2	2	1	0	0.39	3.187	12.811	12.414	14.150	14.9	30.2	45.1
6	6	5	0	3	4	1	2	0.02	2.817	12.813	12.405	13.985	11.0	28.4	39.4
7	0	4	0	5	4	3	2	0.02	2.818	12.859	12.416	15.186	12.8	30.0	42.8
8	1	1	4	4	4	1	0	0.11	2.914	12.855	12.417	15.166	13.4	30.8	44.2
9	7	7	3	2	2	0	0	0.42	3.215	12.816	12.401	14.836	15.7	32.0	47.7
10	4	5	2	4	1	1	0	0.32	3.124	12.828	12.412	14.726	15.1	30.9	46.0
11	6	6	4	1	2	1	2	0.22	3.018	12.814	12.407	14.247	13.2	29.8	43.0
12	7	5	0	4	2	2	1	0.09	2.894	12.803	12.404	13.786	12.0	27.9	39.9
13	5	5	2	4	2	0	2	0.25	3.053	12.822	12.408	14.545	14.0	30.4	44.4
14	0	5	2	0	3	0	2	0.15	2.953	12.839	12.425	14.393	12.8	29.7	42.5
15	3	5	3	3	3	1	2	0.27	3.068	12.836	12.408	15.061	14.7	31.5	46.2
16	0	3	3	5	3	2	1	0.08	2.879	12.848	12.414	14.971	12.8	30.3	43.1
17	3	3	2	3	2	1	1	0.24	3.042	12.832	12.418	14.529	14.1	30.1	44.2
18	3	3	0	2	4	1	0	0.04	2.844	12.821	12.422	13.714	10.8	28.2	39.0
19	1	5	4	4	4	3	0	0.18	2.977	12.853	12.399	15.823	14.8	32.3	47.1
20	1	4	0	4	4	3	2	0.10	2.899	12.852	12.411	15.245	13.7	30.5	44.2
21	7	5	3	4	2	0	0	0.23	3.028	12.809	12.400	14.332	13.6	29.8	43.4
22	4	5	3	4	2	0	2	0.23	3.029	12.826	12.404	14.789	14.0	30.8	44.8
23	7	3	0	1	2	2	1	0.06	2.858	12.783	12.412	12.771	10.6	25.9	36.5
24	7	5	0	5	2	1	1	0.13	2.928	12.805	12.407	13.800	12.1	28.3	40.4

Generation 3

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	0	5	3	0	3	1	2	0.16	2.961	12.832	12.419	14.085	12.5	29.2	41.7
2	4	5	2	4	1	1	0	0.37	3.168	12.812	12.410	14.016	14.0	30.4	44.4
3	7	5	3	4	2	0	0	0.26	3.059	12.808	12.402	13.991	13.4	29.4	42.8
4	7	7	3	2	2	0	0	0.46	3.257	12.828	12.398	15.137	16.8	32.5	49.3
5	3	3	2	2	2	1	0	0.35	3.150	12.826	12.420	14.128	14.6	29.9	44.5
6	7	7	3	2	2	0	0	0.45	3.249	12.829	12.401	15.053	16.5	32.4	48.9
7	3	3	2	3	2	1	1	0.28	3.082	12.821	12.420	13.853	13.4	29.3	42.7
8	3	5	3	3	3	1	2	0.27	3.068	12.839	12.410	14.798	14.3	31.1	45.4
9	4	5	3	4	2	0	2	0.27	3.071	12.826	12.408	14.424	11.8	32.5	44.3
10	1	4	0	4	4	3	2	0.11	2.907	12.842	12.412	14.583	12.7	29.7	42.4
11	4	5	2	4	1	1	0	0.30	3.099	12.825	12.411	14.328	14.1	30.3	44.4
12	0	5	2	0	3	0	2	0.15	2.950	12.836	12.422	14.103	12.4	29.2	41.6
13	0	5	2	4	2	1	2	0.21	3.013	12.836	12.418	14.336	12.4	30.8	43.2
14	0	5	3	0	3	1	2	0.18	2.976	12.838	12.419	14.313	12.9	29.7	42.6
15	3	5	3	4	2	0	0	0.40	3.197	12.817	12.410	14.234	14.5	31.0	45.5
16	7	1	3	4	2	0	0	0.05	2.846	12.801	12.410	13.176	10.5	27.0	37.5
17	3	3	3	2	2	0	0	0.29	3.093	12.824	12.422	13.903	13.5	29.5	43.0
18	3	3	3	3	0	1	0	0.27	3.071	12.808	12.421	13.353	12.7	28.3	41.0
19	3	3	2	3	3	2	1	0.20	2.997	12.836	12.416	14.380	13.4	29.7	43.1
20	3	3	3	3	3	1	1	0.18	2.976	12.835	12.415	14.347	13.1	29.6	42.7
21	4	4	3	4	4	3	2	0.16	2.964	12.847	12.400	15.250	14.2	31.0	45.2
22	1	5	0	4	1	3	1	0.16	2.959	12.832	12.417	14.152	12.6	29.3	41.9
23	0	5	4	4	3	0	2	0.16	2.959	12.862	12.409	15.446	14.2	31.5	45.7
24	4	0	3	0	2	1	0	0.24	3.040	12.804	12.431	12.828	11.9	27.1	39.0

Generation 4

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	3	3	2	3	3	2	1	0.11	3.015	12.842	12.424	14.295	13.5	29.6	43.1
2	3	3	3	2	2	0	0	0.26	3.158	12.807	12.424	13.300	13.3	28.7	42.0
3	0	5	4	4	3	0	2	0.06	2.956	12.855	12.417	14.886	14.0	30.0	44.0
4	0	5	3	0	3	1	2	0.07	2.973	12.834	12.427	13.858	12.4	28.8	41.2
5	3	3	3	2	2	0	0	0.22	3.117	12.812	12.426	13.345	13.1	28.5	41.6
6	7	7	3	2	2	0	0	0.37	3.272	12.815	12.403	14.485	15.9	31.5	47.4
7	4	4	3	4	4	3	2	0.04	2.935	12.836	12.403	14.684	13.1	30.0	43.1
8	7	7	3	2	2	0	0	0.38	3.277	12.826	12.407	14.738	16.6	31.7	48.3
9	4	5	2	4	1	1	0	0.34	3.237	12.806	12.417	13.624	14.4	29.7	44.1
10	3	5	3	4	2	0	0	0.36	3.264	12.819	12.416	14.155	15.4	30.8	46.2
11	0	5	4	4	3	0	2	0.04	2.935	12.853	12.415	14.854	13.1	30.5	43.6
12	7	7	3	2	2	0	0	0.38	3.277	12.820	12.407	14.527	15.9	31.7	47.6
13	3	3	2	2	2	2	1	0.19	3.087	12.832	12.424	14.060	14.3	29.1	43.4
14	3	3	3	2	3	2	1	0.08	2.978	12.838	12.421	14.205	13.0	29.3	42.3
15	0	5	4	4	3	0	1	0.05	2.951	12.861	12.418	15.048	13.6	30.8	44.4
16	0	5	4	2	3	1	2	0.03	2.929	12.831	12.420	13.929	11.8	29.0	40.8
17	1	3	3	3	2	0	0	0.24	3.144	12.827	12.428	13.835	13.8	29.7	43.5
18	3	5	2	2	2	0	0	0.29	3.195	12.848	12.426	14.712	15.6	31.4	47.0
19	4	7	3	1	4	3	0	0.19	3.085	12.839	12.410	14.781	14.4	31.2	45.6
20	3	4	3	2	3	3	2	0.12	3.023	12.852	12.421	14.753	13.6	31.0	44.6
21	4	5	2	4	1	1	0	0.35	3.249	12.806	12.419	13.572	14.3	29.8	44.1
22	3	5	2	4	2	0	0	0.29	3.186	12.812	12.420	13.653	13.8	29.7	43.5
23	0	7	4	2	2	1	0	0.21	3.113	12.841	12.419	14.583	14.8	30.6	45.4
24	0	5	4	2	2	0	2	0.07	2.967	12.846	12.423	14.393	12.8	29.9	42.7

Generation 5

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	7	7	3	2	2	0	0	0.37	3.273	12.813	12.390	14.637	15.9	32.0	47.9
2	3	3	3	2	2	0	0	0.19	3.086	12.817	12.410	13.804	13.2	29.4	42.6
3	3	3	2	2	2	2	1	0.23	3.127	12.823	12.409	14.102	14.1	30.0	44.1
4	0	7	4	2	2	1	0	0.20	3.103	12.841	12.400	14.984	15.0	31.5	46.5
5	3	3	3	2	2	0	0	0.20	3.101	12.822	12.410	13.995	13.6	29.8	43.4
6	4	5	2	4	1	1	0	0.31	3.212	12.815	12.402	14.197	14.7	30.9	45.6
7	3	3	3	2	2	0	0	0.26	3.156	12.803	12.407	13.531	13.4	29.3	42.7
8	7	7	3	2	2	0	0	0.35	3.254	12.816	12.388	14.781	16.1	32.0	48.1
9	7	7	3	2	2	0	0	0.38	3.276	12.818	12.393	14.712	16.1	32.1	48.2
10	3	5	2	2	2	0	0	0.35	3.250	12.834	12.415	14.463	17.9	29.1	47.0
11	7	7	3	2	2	0	0	0.37	3.269	12.822	12.394	14.805	16.3	32.1	48.4
12	3	5	2	4	2	0	0	0.24	3.137	12.818	12.403	14.152	13.9	30.5	44.4
13	7	3	3	0	2	0	0	0.00	2.889	12.775	12.404	12.323	9.9	25.7	35.6
14	3	3	3	0	0	0	0	0.26	3.156	12.798	12.415	13.087	13.1	28.2	41.3
15	3	3	4	3	2	1	0	0.23	3.134	12.827	12.402	14.487	14.9	30.5	45.4
16	3	3	4	2	2	3	0	0.11	3.013	12.823	12.399	14.269	13.2	29.8	43.0
17	3	7	3	5	2	0	1	0.16	3.059	12.824	12.390	14.677	13.7	31.2	44.9
18	6	5	4	2	2	1	0	0.21	3.106	12.805	12.393	14.003	13.7	29.8	43.5
19	3	7	3	2	2	0	0	0.31	3.212	12.833	12.401	14.851	15.8	31.9	47.7
20	3	7	3	2	2	0	0	0.31	3.213	12.833	12.402	14.817	15.7	31.9	47.6
21	0	3	3	5	2	0	0	0.05	2.954	12.828	12.410	13.980	12.0	29.3	41.3
22	3	5	2	2	2	0	0	0.35	3.253	12.825	12.403	14.571	15.8	31.6	47.4
23	3	5	3	1	2	0	0	0.25	3.148	12.819	12.405	14.135	14.1	30.4	44.5
24	7	5	3	2	2	0	0	0.24	3.136	12.799	12.396	13.742	13.6	29.5	43.1

Generation 6

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	6	5	4	2	2	1	0	0.12	3.123	12.807	12.397	14.090	14.0	30.0	44.0
2	3	3	4	3	2	1	0	0.20	3.201	12.821	12.409	14.277	15.3	30.4	45.7
3	3	5	2	2	2	0	0	0.27	3.269	12.822	12.411	14.348	15.7	31.2	46.9
4	7	7	3	2	2	0	0	0.28	3.284	12.809	12.391	14.616	16.1	31.9	48.0
5	7	7	3	2	2	0	0	0.29	3.290	12.809	12.392	14.591	16.1	31.9	48.0
6	3	3	3	2	2	0	0	0.06	3.060	12.818	12.415	13.757	13.0	29.1	42.1
7	7	7	3	2	2	0	0	0.28	3.284	12.817	12.392	14.861	16.5	32.3	48.8
8	7	7	3	2	2	0	0	0.28	3.285	12.818	12.394	14.827	16.4	32.3	48.7
9	3	7	3	2	2	0	0	0.21	3.206	12.837	12.407	14.909	15.8	32.0	47.8
10	7	7	3	2	2	0	0	0.26	3.258	12.822	12.396	14.855	16.2	32.2	48.4
11	3	5	2	2	2	0	0	0.25	3.250	12.829	12.413	14.493	15.7	31.4	47.1
12	7	7	3	2	2	0	0	0.25	3.253	12.818	12.392	14.847	16.1	32.2	48.3
13	3	3	3	2	2	1	0	0.10	3.100	12.828	12.413	14.226	14.1	30.0	44.1
14	3	3	5	3	2	1	0	0.06	3.065	12.834	12.406	14.617	14.4	30.4	44.8
15	3	5	2	2	2	0	0	0.28	3.278	12.825	12.412	14.431	15.9	31.4	47.3
16	2	7	2	2	2	0	0	0.22	3.225	12.842	12.410	15.009	16.4	32.0	48.4
17	7	3	3	2	1	0	0	0.01	3.009	12.776	12.406	12.562	10.9	26.9	37.8
18	3	3	3	2	0	0	0	0.01	3.007	12.807	12.417	13.238	11.9	27.9	39.8
19	7	7	3	2	2	0	0	0.29	3.285	12.822	12.392	15.038	17.0	32.4	49.4
20	7	7	3	3	2	0	2	0.24	3.239	12.837	12.396	15.345	16.9	32.8	49.7
21	7	4	3	2	2	0	0	0.05	3.045	12.799	12.403	13.496	12.7	28.4	41.1
22	7	7	3	2	2	0	0	0.24	3.244	12.823	12.398	14.797	15.9	32.1	48.0
23	3	7	3	3	2	0	0	0.23	3.227	12.836	12.399	15.186	16.5	32.5	49.0
24	7	5	3	2	0	0	0	0.05	3.052	12.790	12.401	13.268	12.4	28.1	40.5

Generation 7

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	7	7	3	2	2	0	0	0.27	3.269	12.822	12.394	15.144	16.9	32.6	49.5
2	7	7	3	2	2	0	0	0.27	3.269	12.828	12.401	15.110	16.8	32.6	49.4
3	3	5	2	2	2	0	0	0.25	3.247	12.835	12.418	14.721	16.0	31.8	47.8
4	7	7	3	2	2	0	0	0.24	3.236	12.828	12.402	15.020	16.2	32.4	48.6
5	3	5	2	2	2	0	0	0.24	3.236	12.839	12.421	14.739	15.9	31.8	47.7
6	3	7	3	3	2	0	0	0.22	3.220	12.851	12.408	15.592	17.1	33.1	50.2
7	7	7	3	2	2	0	0	0.23	3.229	12.835	12.407	15.080	16.3	32.4	48.7
8	3	7	3	2	2	0	0	0.16	3.156	12.864	12.429	15.207	15.9	32.1	48.0
9	7	7	3	2	2	0	0	0.26	3.258	12.849	12.420	15.163	16.7	32.7	49.4
10	3	7	3	2	2	0	0	0.18	3.183	12.859	12.427	15.145	16.0	32.2	48.2
11	7	7	3	2	2	0	0	0.24	3.245	12.845	12.418	15.070	16.4	32.5	48.9
12	7	7	3	2	2	0	0	0.25	3.251	12.844	12.419	15.010	16.4	32.4	48.8
13	7	3	3	2	2	0	0	0.00	2.954	12.807	12.432	12.830	11.0	26.9	37.9
14	7	3	3	2	2	0	0	0.00	2.960	12.808	12.432	12.873	11.3	26.8	38.1
15	7	5	3	2	2	0	2	0.05	3.049	12.834	12.426	14.101	13.6	29.4	43.0
16	6	7	2	2	2	0	0	0.27	3.273	12.852	12.424	15.153	16.9	32.7	49.6
17	3	5	1	2	2	0	0	0.20	3.198	12.854	12.440	14.538	15.2	31.3	46.5
18	3	7	0	2	2	0	0	0.12	3.121	12.855	12.440	14.452	14.5	30.6	45.1
19	3	7	0	2	4	0	0	0.00	2.946	12.848	12.435	14.119	11.9	29.7	41.6
20	7	7	3	3	2	0	0	0.27	3.269	12.853	12.419	15.357	17.0	33.2	50.2
21	7	7	3	0	2	0	0	0.17	3.168	12.838	12.425	14.456	14.8	31.0	45.8
22	3	7	3	2	2	3	0	0.10	3.097	12.868	12.420	15.562	15.7	32.5	48.2
23	7	7	3	2	0	0	0	0.18	3.178	12.826	12.421	14.191	14.5	30.6	45.1
24	7	7	3	2	2	0	0	0.24	3.236	12.844	12.419	14.986	16.2	32.3	48.5

Generation 8

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	3	5	1	2	2	0	0	0.21	3.260	12.855	12.442	14.265	15.3	31.2	46.5
2	7	7	3	2	2	0	0	0.25	3.297	12.849	12.424	14.740	16.3	32.3	48.6
3	7	7	3	0	2	0	0	0.17	3.222	12.843	12.428	14.276	14.9	31.1	46.0
4	7	7	3	2	2	0	0	0.23	3.284	12.847	12.424	14.649	16.0	32.1	48.1
5	7	7	3	2	2	0	0	0.23	3.284	12.847	12.424	14.649	16.0	32.1	48.1
6	7	7	3	2	2	0	0	0.22	3.268	12.845	12.421	14.658	15.9	32.0	47.9
7	3	7	3	3	2	0	0	0.16	3.213	12.870	12.429	15.156	16.3	32.4	48.7
8	7	7	3	2	2	0	0	0.21	3.261	12.851	12.424	14.751	16.0	32.1	48.1
9	7	7	3	3	2	0	0	0.25	3.303	12.859	12.422	15.166	17.1	33.0	50.1
10	7	7	3	2	2	0	0	0.22	3.270	12.855	12.427	14.800	16.1	32.3	48.4
11	3	7	3	2	2	0	0	0.15	3.195	12.868	12.431	14.990	15.8	32.1	47.9
12	7	7	3	2	2	0	0	0.22	3.275	12.851	12.422	14.842	16.4	32.2	48.6
13	7	7	3	2	2	0	0	0.23	3.279	12.859	12.429	14.883	16.5	32.3	48.8
14	7	7	5	2	2	0	0	0.18	3.233	12.838	12.423	14.292	14.8	31.4	46.2
15	7	7	3	0	2	0	0	0.19	3.238	12.847	12.433	14.267	15.1	31.1	46.2
16	2	7	3	0	2	0	0	0.07	3.124	12.867	12.442	14.467	14.4	30.8	45.2
17	7	7	3	2	1	0	0	0.26	3.306	12.849	12.423	14.789	16.4	32.5	48.9
18	3	7	3	2	0	0	0	0.13	3.180	12.863	12.437	14.589	15.1	31.3	46.4
19	7	7	3	3	2	0	0	0.27	3.320	12.870	12.429	15.332	17.6	33.3	50.9
20	3	7	3	3	2	0	2	0.14	3.190	12.876	12.427	15.392	16.3	32.8	49.1
21	7	4	3	3	2	0	0	0.04	3.093	12.838	12.432	13.774	13.4	29.2	42.6
22	7	7	3	2	2	0	0	0.22	3.275	12.864	12.428	15.085	16.9	32.5	49.4
23	3	7	3	3	2	0	0	0.18	3.234	12.881	12.433	15.431	16.8	33.1	49.9
24	7	7	3	2	0	0	0	0.12	3.173	12.836	12.428	13.961	14.0	30.3	44.3

Generation 9

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	7	7	3	3	2	0	0	0.13	3.278	12.874	12.444	15.037	16.8	32.5	49.3
2	7	7	3	2	1	0	0	0.10	3.251	12.861	12.449	14.365	15.5	31.2	46.7
3	3	7	3	3	2	0	0	0.05	3.195	12.898	12.460	15.180	16.4	32.1	48.5
4	7	7	5	2	2	0	0	0.00	3.153	12.863	12.442	14.525	14.6	31.2	45.8
5	7	7	3	2	1	0	0	0.12	3.273	12.869	12.452	14.574	16.1	31.6	47.7
6	7	7	3	2	2	0	0	0.09	3.238	12.873	12.449	14.763	16.0	31.8	47.8
7	3	7	3	3	2	0	2	0.01	3.158	12.899	12.455	15.326	16.2	32.2	48.4
8	7	7	3	2	2	0	0	0.11	3.257	12.872	12.447	14.828	16.2	32.1	48.3
9	7	7	3	2	2	0	0	0.11	3.262	12.874	12.448	14.870	16.2	32.3	48.5
10	7	7	3	2	1	0	0	0.13	3.282	12.868	12.450	14.624	16.3	31.7	48.0
11	3	7	3	3	2	0	0	0.08	3.233	12.899	12.459	15.311	16.8	32.7	49.5
12	7	7	3	2	2	0	0	0.10	3.251	12.873	12.454	14.610	15.9	31.6	47.5
13	3	7	3	2	1	0	0	0.06	3.209	12.887	12.465	14.647	15.6	31.4	47.0
14	7	7	3	2	2	0	0	0.09	3.244	12.878	12.455	14.737	16.0	31.8	47.8
15	3	7	3	3	2	0	1	0.01	3.163	12.899	12.456	15.300	16.1	32.3	48.4
16	3	7	3	2	2	0	0	0.03	3.177	12.894	12.460	15.012	15.7	32.0	47.7
17	1	7	3	3	2	0	0	0.01	3.160	12.907	12.468	15.157	15.9	32.0	47.9
18	7	5	2	2	2	0	0	0.04	3.192	12.867	12.461	14.066	14.8	30.1	44.9
19	3	7	3	1	2	0	0	0.02	3.168	12.888	12.466	14.583	15.0	31.2	46.2
20	3	7	3	2	3	0	2	0.00	3.036	12.894	12.461	14.754	14.0	30.8	44.8
21	7	7	3	2	2	0	0	0.06	3.209	12.872	12.451	14.613	15.4	31.5	46.9
22	7	7	3	2	1	0	0	0.11	3.264	12.875	12.453	14.735	16.2	31.9	48.1
23	3	7	3	2	2	1	0	0.06	3.215	12.898	12.460	15.212	16.4	32.5	48.9
24	3	5	3	2	2	0	0	0.02	3.172	12.885	12.470	14.346	14.7	30.8	45.5

Generation 10

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	7	7	3	2	1	0	0	0.12	3.271	12.882	12.462	14.796	16.5	31.9	48.4
2	3	7	3	3	2	0	0	0.07	3.217	12.922	12.480	15.480	17.1	32.7	49.8
3	7	7	3	3	2	0	0	0.14	3.286	12.901	12.462	15.490	17.7	33.2	50.9
4	7	7	3	2	1	0	0	0.13	3.276	12.887	12.467	14.804	16.4	32.1	48.5
5	7	7	3	2	1	0	0	0.13	3.278	12.892	12.474	14.736	16.5	31.8	48.3
6	7	7	3	2	2	0	0	0.10	3.255	12.890	12.468	14.840	16.4	31.9	48.3
7	3	7	3	2	2	1	0	0.08	3.232	12.912	12.478	15.224	16.8	32.4	49.2
8	7	7	3	2	1	0	0	0.10	3.254	12.886	12.470	14.628	16.1	31.5	47.6
9	7	7	3	2	1	0	0	0.11	3.265	12.885	12.470	14.610	16.1	31.6	47.7
10	7	7	3	3	2	0	0	0.14	3.292	12.901	12.470	15.218	17.3	32.8	50.1
11	7	7	3	2	1	0	0	0.15	3.298	12.892	12.474	14.768	16.7	32.0	48.7
12	7	7	3	2	2	0	0	0.09	3.238	12.891	12.474	14.638	15.7	31.7	47.4
13	7	0	3	3	1	0	0	0.00	2.952	12.873	12.487	13.143	11.6	27.2	38.8
14	7	7	3	3	2	3	1	0.01	3.163	12.908	12.464	15.460	16.2	32.7	48.9
15	7	7	0	3	1	0	0	0.00	3.152	12.875	12.480	13.737	13.8	29.5	43.3
16	7	7	3	3	2	0	0	0.17	3.320	12.895	12.469	15.088	17.3	32.8	50.1
17	7	7	3	2	1	0	0	0.12	3.269	12.888	12.472	14.652	16.2	31.7	47.9
18	7	7	5	0	2	3	0	0.00	2.984	12.876	12.463	14.109	12.6	29.5	42.1
19	7	7	3	2	2	0	0	0.15	3.295	12.888	12.473	14.658	16.4	31.9	48.3
20	7	7	3	2	1	2	0	0.00	3.150	12.889	12.470	14.570	14.8	31.1	45.9
21	4	7	3	2	2	0	0	0.08	3.235	12.902	12.480	14.808	16.0	31.9	47.9
22	3	7	3	2	2	2	0	0.02	3.171	12.918	12.480	15.264	16.2	32.2	48.4
23	7	7	3	1	2	0	0	0.06	3.212	12.888	12.470	14.632	15.5	31.5	47.0
24	7	1	3	2	1	0	0	0.00	2.929	12.866	12.484	12.975	11.2	26.8	38.0

Generation 11

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	4	7	3	2	2	0	0	0.08	3.280	12.932	12.507	14.785	16.6	31.9	48.5
2	7	7	3	2	1	0	0	0.10	3.300	12.917	12.511	14.153	15.9	30.8	46.7
3	7	7	3	2	1	0	0	0.11	3.312	12.914	12.510	14.101	15.9	30.8	46.7
4	7	7	3	2	1	0	0	0.11	3.311	12.913	12.508	14.134	15.9	30.9	46.8
5	7	7	3	2	2	0	0	0.09	3.292	12.911	12.502	14.246	15.8	31.1	46.9
6	7	7	3	3	2	0	0	0.12	3.321	12.918	12.500	14.603	16.7	31.8	48.5
7	7	7	3	2	1	0	0	0.13	3.326	12.907	12.501	14.191	16.0	31.2	47.2
8	7	7	3	2	1	0	0	0.13	3.327	12.907	12.502	14.157	16.1	31.0	47.1
9	7	7	3	2	1	0	0	0.13	3.332	12.906	12.501	14.165	16.1	31.1	47.2
10	7	7	3	2	1	0	0	0.13	3.325	12.908	12.501	14.225	16.1	31.2	47.3
11	7	7	3	2	1	0	0	0.13	3.330	12.913	12.506	14.232	16.3	31.1	47.4
12	7	7	3	2	1	0	0	0.13	3.331	12.907	12.501	14.199	16.1	31.2	47.3
13	7	7	3	2	2	0	1	0.06	3.256	12.915	12.500	14.402	15.7	31.2	46.9
14	0	7	3	2	1	0	0	0.00	3.149	12.937	12.518	14.384	14.9	30.4	45.3
15	7	7	3	4	1	0	0	0.07	3.273	12.912	12.499	14.358	15.8	31.2	47.0
16	7	7	3	2	1	0	0	0.10	3.304	12.912	12.505	14.194	15.8	31.1	46.9
17	7	7	3	3	2	0	0	0.16	3.358	12.925	12.503	14.799	16.9	32.8	49.7
18	7	7	5	2	2	3	0	0.00	3.071	12.907	12.492	14.134	13.3	30.1	43.4
19	7	7	3	2	1	0	0	0.11	3.314	12.903	12.502	13.999	15.7	30.7	46.4
20	4	7	3	2	1	0	0	0.08	3.284	12.915	12.509	14.130	15.5	30.9	46.4
21	7	7	3	2	1	0	0	0.10	3.303	12.902	12.500	14.018	15.5	30.8	46.3
22	7	7	0	2	0	0	0	0.00	3.190	12.877	12.506	12.789	13.7	27.1	40.8
23	2	7	2	2	1	0	0	0.06	3.257	12.918	12.511	14.125	15.4	30.6	46.0
24	7	7	3	2	1	0	1	0.04	3.243	12.895	12.491	14.001	15.1	30.3	45.4

Generation 12

	Position Number							Fitness	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
Chromosome #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Function		(volts)	(volts)	(g)	(g)	(g)	(g)
1	7	7	3	2	2	0	0	0.12	3.274	12.906	12.489	14.415	15.6	31.6	47.2
2	4	7	3	2	1	0	0	0.11	3.257	12.926	12.499	14.735	16.0	32.0	48.0
3	7	7	3	2	1	0	0	0.17	3.321	12.916	12.493	14.693	16.6	32.2	48.8
4	7	7	3	2	1	0	0	0.19	3.338	12.916	12.495	14.649	16.7	32.2	48.9
5	7	7	3	2	1	0	1	0.14	3.290	12.927	12.496	14.923	17.0	32.1	49.1
6	7	7	3	2	1	0	0	0.20	3.351	12.918	12.500	14.564	16.7	32.1	48.8
7	7	7	3	2	1	0	0	0.18	3.327	12.923	12.501	14.667	16.8	32.0	48.8
8	7	7	3	2	1	0	0	0.18	3.326	12.923	12.500	14.701	16.8	32.1	48.9
9	7	7	3	2	1	0	0	0.18	3.331	12.924	12.501	14.709	16.8	32.2	49.0
10	7	7	3	2	1	0	0	0.17	3.325	12.925	12.500	14.768	16.9	32.2	49.1
11	7	7	3	2	1	0	0	0.19	3.341	12.925	12.502	14.724	17.0	32.2	49.2
12	7	7	3	2	1	0	0	0.18	3.331	12.925	12.501	14.742	16.9	32.2	49.1
13	7	7	5	2	1	0	0	0.07	3.216	12.909	12.491	14.364	15.0	31.2	46.2
14	4	7	1	2	2	0	0	0.12	3.265	12.931	12.508	14.609	15.9	31.8	47.7
15	2	7	3	2	1	0	0	0.06	3.210	12.940	12.510	14.767	15.8	31.6	47.4
16	7	7	3	5	1	0	0	0.07	3.218	12.917	12.493	14.573	15.3	31.6	46.9
17	7	7	3	2	1	0	0	0.18	3.330	12.911	12.499	14.324	16.1	31.6	47.7
18	7	7	3	2	1	2	0	0.06	3.213	12.921	12.497	14.566	15.3	31.5	46.8
19	7	7	3	3	1	0	0	0.15	3.301	12.928	12.498	14.905	16.9	32.3	49.2
20	7	7	3	2	1	0	0	0.19	3.339	12.914	12.500	14.407	16.3	31.8	48.1
21	7	7	3	2	1	0	0	0.17	3.322	12.916	12.500	14.451	16.2	31.8	48.0
22	7	7	1	2	1	0	2	0.14	3.287	12.924	12.503	14.572	16.6	31.3	47.9
23	7	7	3	2	4	0	2	0.00	3.086	12.923	12.492	14.616	14.0	31.1	45.1
24	7	7	3	2	1	1	1	0.05	3.203	12.920	12.497	14.517	15.2	31.3	46.5

Appendix C: Angle of Attack Data

Best Wing Angle of Attack Data

A0A (degrees)	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
0	-1.024716154	13.525	13.273	6.148044	-6.2	-0.1	-6.3
2	1.350818191	13.936	13.668	7.1808331	1.1	8.6	9.7
4	3.27737432	15.755	15.395	12.143868	13.5	26.3	39.8
6	3.445515075	13.552	13.112	17.094686	21.7	37.2	58.9
8	3.461969187	16.48	16.021	24.292533	34.9	49.2	84.1
10	3.282796444	17.264	16.735	33.995407	46.8	64.8	111.6
12	2.990223505	10.508	9.979	40.231106	52	68.3	120.3
14	2.789889206	11.455	10.921	47.815519	56.2	77.2	133.4
16	2.603847479	12.578	12.019	56.800562	62.1	85.8	147.9
24	1.900919707	12.28	11.587	79.435233	52	99	151

Zimmerman Representation Angle of Attack Data

A0A (degrees)	L/D	voltage	v0	Drag	Lscale	Rscale	Lift
0	-1.152396441	13.568	13.255	7.636261	-8.8	0	-8.8
2	0.894126301	13.996	13.648	9.1709639	-0.9	9.1	8.2
4	2.957722	15.772	15.376	12.982965	11.8	26.6	38.4
6	3.149131582	13.616	13.089	19.529193	22.4	39.1	61.5
8	3.321977044	16.511	16.007	25.707583	35	50.4	85.4
10	3.191504299	17.316	16.724	36.691162	49.2	67.9	117.1
12	2.988155346	10.557	9.968	44.743323	56.9	76.8	133.7
14	2.810804066	11.516	10.907	55.393402	68.7	87	155.7
16	2.552827605	12.663	12.011	63.106494	72.6	88.5	161.1
24	1.911151517	12.4	11.571	96.800279	64	121	185

Appendix D: Drag Calibration Data Summary

Calibration Test #	When it occurred during testing	Drag Slope	R^2 value
1	Initial Population Prior to First Individual	26.693	1.000
2	Initial Population between Individuals 12 and 13	26.423	1.000
3	Initial Population between Individuals 24 and 25	26.524	1.000
4	Initial Population between Individuals 36 and 37	25.717	1.000
5	Initial Population between Individuals 48 and 49	26.616	1.000
6	Initial Population between Individuals 60 and 61	26.509	0.9999
7	Prior to Generation 2	26.532	0.9999
8	Prior to Generation 3	26.007	0.9999
9	Prior to Generation 4	25.930	0.9999
10	Prior to Generation 5	25.521	0.9999
11	Prior to Generation 6	25.758	0.9999
12	Prior to Generation 7	26.058	0.9999
13	Prior to Generation 8	25.511	0.9999
14	Prior to Generation 9	25.776	1.0000
15	Prior to Generation 10	25.978	0.9999
16	Prior to Generation 11	26.034	0.9999
17	Prior to Generation 12	25.885	0.9999
18	Prior to tape test	25.885	0.9999
19	Prior to 0 degree AOA test	24.397	1.0000
20	Prior to 2 degree AOA test	25.531	1.0000
21	Prior to 4 degree AOA test	26.021	0.9999
22	Prior to 6 degree AOA test	24.859	1.0000
23	Prior to 8 degree AOA test	27.425	1.0000
24	Prior to 10 degree AOA test	27.630	1.0000
25	Prior to 12 degree AOA test	28.770	1.0000
26	Prior to 14 degree AOA test	29.107	1.0000
27	Prior to 16 degree AOA test	28.683	0.9999
28	Prior to 24 degree AOA test	26.000	0.9996