5-2012

Humanitarian Response Unmanned Aircraft System (HR-UAS)

Justin T. Knott

David P. Brundage

John S. Campbell

D. Austin Eldridge

Shaun B. Hooker

See next page for additional authors

Author

Justin T. Knott, David P. Brundage, John S. Campbell, D. Austin Eldridge, Shaun B. Hooker, Jake R. Mashburn, and Jacob L. Philpott

Humanitarian Response Unmanned Aircraft System (HR-UAS)

David P. Brundage, John S. Campbell, D. Austin Eldridge,
Shaun B. Hooker, Justin T. Knott, Jake R. Mashburn, and Jacob L. Philpott *

Robert E. Bond †

Department of Mechanical, Aerospace, and Biomedical Engineering
The University of Tennessee at Knoxville

The design for a Humanitarian Response Unmanned Aircraft System (HR-UAS) is presented. This vehicle is designed to be an affordable, autonomous aircraft that can deliver 1,800 lbs of relief supplies contained on two pallets to unimproved runways of less than 500 feet in length wherever supplies are needed. It is also designed to fit inside a C-130J-30 for transport and staging deployment into remote regions. A typical mission consisting of transporting a full payload 300nm and then returning to the operating base is analyzed. The overall design of the aircraft, its systems, structures, aerodynamics, and flight performance is also presented.

Nomenclature

α_t	Lift Curve Slope Of Tail Airfoil
α_w	Lift Curve Slope Of Wing Airfoil
ϵ_α	Downwash Angle
η_m	Motor Efficiency
η_p	Propeller Efficiency
η_t	Dynamic Pressure Ratio Between Wing And Tail
AGL	Above Ground Level
AR	Aspect Ratio
b	Wingspan
$C.G.$	Center Of Gravity
C_{D0}	Profile Drag
C_{fe}	Skin Friction Coefficient
C_l	Airfoil Section Lift Coefficient
C_L	Wing Lift Coefficient
$C_{l_{max}}$	Maximum Airfoil Section Lift Coefficient
C_r	Chord At The Wing Root
C_t	Chord At The Wingtip
h_l	Tail Aerodynamic Center (Located At Root Quarter-Chord)
h_n	Airplane Neutral Point/Aerodynamic Center
h_{nw}	Wing Aerodynamic Center (Located At Root Quarter-Chord)
h_t	Vertical Distance From Wing Chord To Tail Chord

*Undergraduate Student, 414 Dougherty Building. The University of Tennessee, Knoxville, TN USA 37996.
†Associate Professor and corresponding author, 414 Dougherty Building. The University of Tennessee, Knoxville, TN USA 37996.

J	Advance Ratio
K	Constant Multiplied By ${C_l}^2$ To Obtain The Induced Drag
l_{ac}	Length From h_{nw} To h_l
P_A	Power Available
P_S	Power Delivered To The Shaft
RC	Remote Control
S	Planform Area Of Wing
S_t	Planform Area Of Tail
S_{wet}	Wetted Surface Area Of The Plane
V_{max}	Maximum Velocity
V_y	Velocity For Fastest Rate Of Climb

I. Introduction

The HR-UAS was designed in response to the AIAA's request for proposal for a Humanitarian Response Unmanned Aircraft System (HR-UAS). In light of recent natural disasters such as the earthquakes in Japan and Haiti, a need has arisen for cargo supply humanitarian relief missions. The HR-UAS was designed as an aircraft system to provide this aid, wherever it is needed.

A. Design Objectives and Background

Whenever a natural disaster strikes, logistical challenges hampering response efforts usually follow. Earthquakes, tsunamis, hurricanes, etc. almost always damage critical infrastructure cutting off relief supplies. Precision, unmanned, cargo supply could alleviate many of the challenges these calamities cause. The HR-UAS was designed as a practical and easy to operate system that provides critical supplies to remote, unimproved areas.

The AIAA RFP includes several major requirements:

1. Create an unmanned aircraft system capable of autonomous flight using GPS, including terminal operations. It must also support control via a ground station backpack weighing less than 50 pounds.
2. The aircraft must be able takeoff and land with a 500' ground roll on unimproved runways.
3. The aircraft must have a useful load of 3,000 lbs. This includes 1,800 lbs for two 36" x 36" x 42" pallets containing humanitarian relief supplies and 1,200 lbs of fuel.
4. The pallets must be able to be loaded and unloaded in 30 minutes by personnel at a remote resupply area.
5. The aircraft must have a cruise true airspeed greater than or equal to 140 knots.
6. The aircraft must have a service ceiling of at least 15,000' MSL.
7. The aircraft must have a 600nm range, including 2 takeoff and landings, with its full 1,800 lb payload.
8. The aircraft must be able to be shipped and fit completely within the cargo bay of a C-130J-30 (10' x 9' x 55'). It must then be easily reassembled to a flying configuration.

A team of 7 Senior Aerospace Engineering students from the University of Tennessee sought to meet these requirements with guidance from an advising professor.

B. Research Aircraft

Before beginning the design of a new aircraft, the designers sought to determine if a suitable aircraft already existed to fulfill these requirements. Several different airplanes were researched, each designed for a different mission.

1. Short Takeoff and Landing

The Zenith STOL CH series aircraft are lightweight kit planes designed for "off-airport" short take-off and landing. These aircraft were of similar scale and form to the designer's initial vision of the HR-UAS. Zenith's wing design heavily influenced that of the HR-UAS. Utilizing permanent leading edge slats and full length "junker style" flaperons (both ailerons and flaps), the Zenith STOL aircraft demonstrate remarkable take-off and landing capabilities. It was this short field take-off and landing capability that made the Zenith planes stand out. Even though the HR-UAS was expected to be much heavier than the Zenith STOL aircraft, the insight they provided into wing design and high lift devices was invaluable.

2. Freight/Cargo Aircraft

Heavy lift, awkwardly shaped aircraft such as the Airbus Beluga and the Aero Spacelines Super Guppy were researched for ideas. Aircraft that serve a transport function were important to the project because the HR-UAS was designed to serve that purpose as well. The awkward shape of the Beluga allows for oversized loads to be transported, including parts of planes under construction. The aircraft has a limited weight capacity, but the awkward shape allows the Beluga to accommodate large cargo that most other planes would be unable to carry. The Super Guppy is very similar to the Beluga, but an older model that is propeller driven instead of jet propelled. In the end, these aircraft were not very influential to the final design simply because of their large, size and awkward shape. However, other general cargo aircraft were researched simply to achieve an understanding of cargo areas and how most function. The functionality of cargo flooring as well as loading and unloading systems were investigated and ideas for the design of the aircraft were obtained.

3. Crop-dusters

Ayers Corporation makes the Thrush agricultural aircraft. The Thrush demonstrated very desirable flight characteristics like relatively short take-off and landing, adequate speed for the purposes of HR-UAS and, high loading capability. The Ayers was of similar scale and form to the initial vision of the HR-UAS but the Thrush had a low wing design. Considered because it is much heavier than the Zenith STOL aircraft, the Thrush had a larger engine, similar to that on the HR-UAS, and was still capable of relatively short field take-off and landing. The Thrush uses a side hatch for loading and unloading. A similar design was experimented with for the HR-UAS. First pass estimates of engine and flight performance were taken from picking components of the Thrush and other aircraft to determine what components would yield desirable characteristics.

4. Military

The M28 Skytruck was heavily relied upon for this project. The Skytruck is a STOL aircraft, and it is designed for light cargo and passenger transportation. The plane is a perfect description of the capabilities that is needed for the HR-UAS. It was designed with STOL capabilities, unimproved runway capabilities, and twin PT6A-65B turboprops producing 1100 shp each. The twin turboprops are used to produce enough horsepower to lift off the ground in a short distance. The aircraft has a high wing design to protect the wings, engines, and propellers from damage from to unimproved runways. The M28 Skytruck is currently used by the Air Force Special Operations for their missions. The STOL capability and unimproved runway capability make the Skytruck perfect for their line of work. They need to be able to land and take off in a short distance in remote places of the world. The HR-UAS needed to be scaled down from the Skytruck in order to meet the requirements for this project. Other military aircraft that were influential for the design were the Global Hawk and the C-130. The Global Hawk, as a UAV, was of importance because the HR-UAS is also unmanned. Unmanned aircraft typically have unique designs compared to conventional airframes, especially with respect to the wing area and placement. The C-130 was referenced for its cargo capabilities. The loading and unloading systems are focused out of the tail of the aircraft with cargo floors and systems for accommodating heavy loads. The HR-UAS design team took characteristics from the Skytruck, Global Hawk, C-130J, and previous planes to develop a preliminary design.

C. Societal Impact

In 2011, according to the United Nations International strategy for disaster reduction, 363 billion dollars were lost due to damage, 162 million people were impacted, and 32,816 people lost their lives in natural disasters across the globe. This paints a clear picture as to how detrimental natural disasters are to people's lives as well as the economy. Although, the preemptive mitigation of impact is important, it is virtually impossible to predict and prevent all damage. Therefore, humanitarian relief is necessary. In the time of these disasters the logistics behind humanitarian efforts is vital to the overall effectiveness of the relief. Often logistical problems arise in the mere transportation of supplies causing a surplus to build up in one area while other areas experience devastating shortages.[1]

The use of an UAV in these efforts addresses some of the transportation issues. Three significant factors that govern transportation logistics after a natural disaster are flexibility, speed, and cost. The use of a short takeoff and landing aircraft for the delivery of the humanitarian relief opens up otherwise hard to reach locations and runways. The location of drop of points can be chosen based on ideal locations for distribution and not governed by the country's current state of infrastructure. The use of aircraft decreases delivery time by eliminating slow and difficult ground routes. An unmanned aircraft will also decrease the expense of humanitarian operations. By eliminating the existence of a crew on board, many costly, safety regulations can be relaxed. Also, the crew changes from a three to four man onboard crew to one operator on ground who can control multiple vehicles. This decreases the overhead which, in terms of relief, means more money can be used on the food and supplies.

D. Management Summary

In order to minimize errors, all design decisions were made as a team, typically with one person as the lead in a specific area. The leader of an area performed necessary calculations, researched different alternatives, and reported the things they learned back to the team. If the task was large enough and additional help could speed up the process, primary assistants for an area helped the leader before reporting back to the rest of the team. Designing an aircraft is a very iterative process and a decision or constraint in one area can very easily affect and limit what is possible in another area. Strong communication and flexibility were critical in order to successfully design the HR-UAS.

1. Working Plan

A general timeline was set out at the beginning of the project which was adapted over time. At each step, different objectives were changed, thrown out, or accomplished. The table below features the general steps of the project and the goals that were achieved at different points.

Table 1. General Timeline of Work

Week	Task
11/01/11	Grouping and Brainstorming
11/08/11	Research Aircraft and General Ideas
11/15/11	Landing Gear/Loading Decisions
11/22/11	Basic Initial Design
11/29/11	Initial Presentation and Future Steps
Christmas Break	
01/08/12	Plan and Organize
01/15/12	Preliminary C.A.D. Sketch
01/22/12	Timeline Finalized
01/29/12	Landing Gear Finalized
02/05/12	Avionics
02/12/12	Wing Structure and Material
02/19/12	Airfoils
02/26/12	Air Frame
03/04/12	Loading/Unloading
03/11/12	Propulsion
03/18/12	Abstract
03/25/12	Mid Presentation (3/30/2012)
04/01/12	Combine Structures
04/08/12	Cut Models
04/15/12	Test Models
04/22/12	Take-off/Landing
04/29/12	Complete Rough Draft
05/06/12	Final Presentation, Submit Report

II. Mission Requirements

The AIAA RFP includes an example mission to evaluate the aircraft's performance. The aircraft was to be evaluated for zero wind conditions and a standard day atmosphere flying the following mission:

1. Warm-up and taxi (if needed) at idle for 5 minutes
2. Takeoff from a forward operating base (FOB) at sea level with 1,800 lbs payload
3. Climb to 8,000 feet MSL with 1,800 lbs payload
4. Cruise at 8,000 ft MSL for 300 nm with 1,800 lbs payload (not including distance to climb)
5. Descend to sea level and land in less than or equal to a 500 ft ground roll at a remote resupply area (RRA) with 1,800 lb payload
6. Cargo unloading/loading time of 30 minutes (engines off) followed by warm-up and taxi (if needed) at idle for 5 minutes
7. Takeoff at sea level in less than 500 ft ground roll from an RRA without refueling and with 1,800 lb payload retained
8. Climb to 8,000 feet MSL with 1,800 lbs payload
9. Cruise at 8,000 ft MSL for 300 nm with 1,800 lbs payload (not including distance to climb)

10. Descend to sea level and land at a FOB with 1,800 lbs payload, taxi (if needed) and shutdown

The aircraft must also have fuel reserves for 20 minutes of flight at 2,000 ft MSL, should not receive a range credit for descents, and should assume the RAA has an improvised gravel or grass landing strip. The mission profile can be seen in Figure 1.

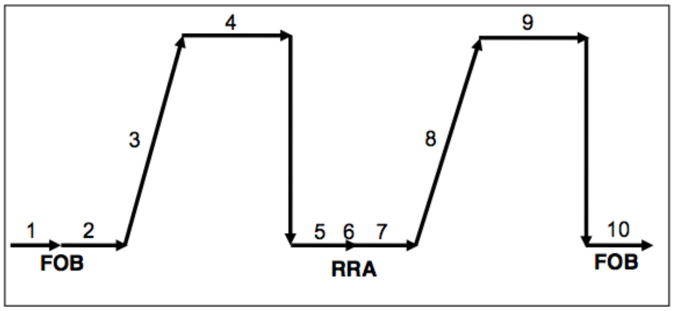

Figure 1. HR-UAS Mission Profile

III. Preliminary Design

A. Design and Analysis Methodology

1. Wing Sizing and Control Surfaces

The design team attacked the problems of wing size, placement, and control surfaces very early on. For obvious reasons, the wing size and control surfaces are extremely important components of the aircraft. These components are going to primarily determine how the HR-UAS aircraft will perform. The team knew that a larger wing planform area would yield more lift so a long wing span and fairly wide chord were initially chosen. It was possible to minimize induced drag by choosing a particular taper ratio (λ). As one can see in Figure 2, the minimum induced drag occurs around $\lambda = 0.4$. The team's initial design had a $55ft$ wing span, $8ft$ root chord, and $3ft$ tip chord, with no quarter chord sweep for structural simplicity. It wasn't until the structure of the wing was further analyzed that the team determined the wing span was too long to be structurally sound. At this point the span was cut down to $40ft$ with a root chord of $8ft$ and tip chord of $4.37ft$. This new configuration has a taper ratio of $\lambda = 0.55$ which is still very close to the minimum induced drag point in Figure 2.

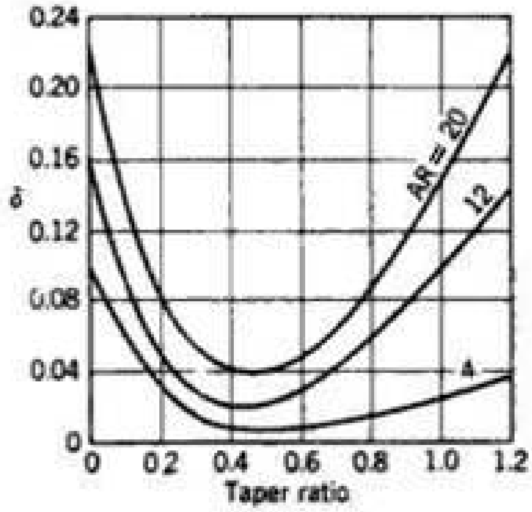

Figure 2. Induced Drag vs. Taper Ratio

Aileron sizing was based on historical guidelines presented in Raymer's textbook, *Aircraft Design: A Conceptual Approach*.[2] An aileron span to wing span ratio of 1.0 and aileron length of $8ft$ was chosen to leave adequate room for flaps to be added and from the guidelines below, an aileron chord of 11% of the wing chord was chosen. Figure 6 shows the final planform layout of the wing, ailerons, and flaps.

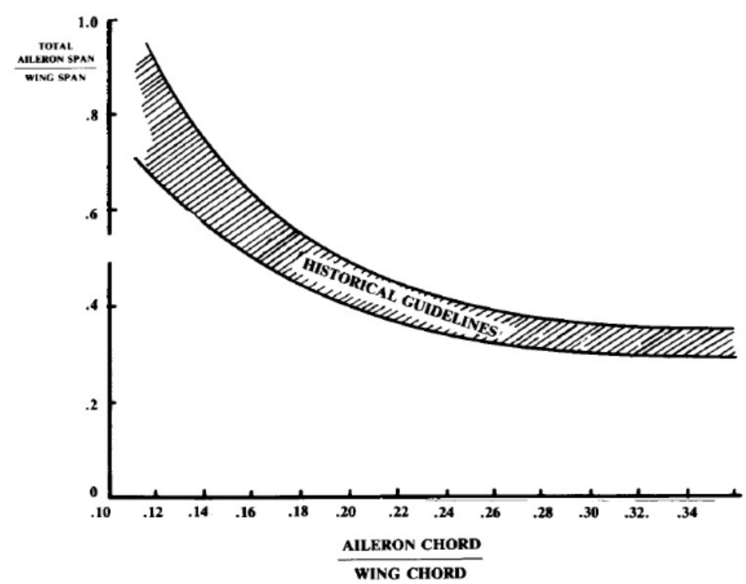

Figure 3. Aileron Sizing Guidelines

Wing placement was addressed from a structural perspective and from the view of those who will do the loading and unloading. By using a high wing, the main spar along the unswept quarter chord was taken as one straight piece only broken at the folding joints. A high wing will also make maneuvering around the aircraft on the ground, like during loading and unloading cargo, much easier. Mid and low wing configurations would be cumbersome in this process. Another benefit of a high wing is that it will be safe from flying debris when taking off and landing on unimproved runways. Any gravel, dirt, or sand kicked up by the propeller is much less likely to impact a higher wing.

2. High Lift Devices

One of the defining design requirements was short field take-off and landing on unimproved runways. The maximum take-off and landing ground roll distance was specified as $500ft$ with no explicit obstacle clearing requirements. This meant the HR-UAS would need significant amounts of lift in order to satisfy these requirements when fully loaded.

Two main configurations were identified early on for consideration. The first was a traditional flap and slat system inspired by the Zenith STOL aircraft. A more exotic flow control configuration called the Co-Flow Jet (CFJ) airfoil was a second option. A preliminary trade study, shown in Tables 4 and 5, was conducted comparing the two configurations. It was originally determined that CFJ would be the better option but only by a slight margin. The main draw back to the CFJ system was that it required a large amount of power to run while flaps and slats did not. Since CFJ came out on top in the trade study, flaps and slats were put on hold at this point.

Table 2. Top Level Criteria Weights for High Lift Configuration. Scores: 1 - Equal Importance, 3 - Moderate Importance, 5 - Strong Importance, 7 - Very Strong Importance, 9 - Extreme Importance

	Ease of Production	Stall	Lift	Drag	Reliability	Weight	Systems Impact	Power	Row Tot.	Criteria Weight
Ease of Production	**1.00**	0.20	0.20	0.33	0.33	0.33	3.00	0.20	5.60	**0.05**
Stall	5.00	**1.00**	1.00	1.00	7.00	3.00	3.00	3.00	24.00	**0.20**
Lift	5.00	1.00	**1.00**	1.00	7.00	5.00	5.00	5.00	30.00	**0.25**
Drag	3.00	1.00	1.00	**1.00**	5.00	3.00	3.00	3.00	20.00	**0.17**
Reliability	3.00	0.14	0.14	0.20	**1.00**	0.33	3.00	3.00	10.82	**0.09**
Weight	3.00	0.33	0.20	0.33	3.00	**1.00**	0.33	0.20	8.40	**0.07**
Systems Impact	0.33	0.33	0.20	0.33	0.33	3.00	**1.00**	0.33	5.87	**0.05**
Power	5.00	0.33	0.20	0.33	0.33	5.00	3.00	**1.00**	15.20	**0.13**

Table 3. Weighted Scores for High Lift Configurations. Scores: 100 - Excellent, 60 - Acceptable, 30 - Marginal, 0 - Not Addressed

	Ease of Production	Stall	Lift	Drag	Reliability	Weight	Systems Impact	Power	Score
CFJ	50	100	100	100	60	60	50	10	**77.39**
Flaps and Slats	60	70	60	60	100	60	70	100	**71.17**

A Co-Flow Jet airfoil injects a high energy jet of air into the flow over the wing near the leading edge and subsequently sucks air in near the trailing edge. Whatever mass flow is injected is in turn pulled back in through the suction slot. This high energy jet mixes with the flow around the airfoil transferring energy to over come the adverse pressure gradient and therefore can put off stall in some cases to nearly 40°angle of attack. CFJ also dramatically improves the coefficients of lift and drag to the point of potentially generating negative drag, or thrust. After going through quite a few calculations to estimate the flight performance of the HR-UAS utilizing the CFJ system, it was determined the system would be impractical. Despite promising some exceptional flight characteristics, the system required too much power. Equations 1, 2, and 3 were used to calculate jet momentum coefficient, power required and power coefficient.[3]

$$C_\mu = \frac{\dot{m}V_{jet}}{\frac{1}{2}\rho_\infty V_\infty^2 S} \tag{1}$$

$$P = \frac{\dot{m} C_P T_1}{\eta_p} (\Gamma^{\frac{\gamma - 1}{\gamma}} - 1) \quad (2)$$

$$P_C = \frac{P}{\frac{1}{2} \rho_\infty V_\infty^3 S} \quad (3)$$

The design team analyzed two CFJ set ups, an open slotted design and a higher performance partially obstructed slot design (DCFJ 2/3). In Table 6 the results from this analysis are shown. Required shaft power was calculated with three momentum coefficients for both set ups. For these calculations, a pump efficiency of 80% was assumed to get shaft horse power. As one can easily see, the CFJ system would require obscene amounts of power to get any significant gains in lift. Another engine, even bigger than the HR-UAS main engine, would be required to run this system and achieve any real performance boost over flaps and slats.

Table 4. Power Required and Performance Gain Analysis of CFJ System

	Open Slot			**DCFJ 2/3**		
Cmu	0.08	0.16	0.3	0.08	0.16	0.3
Pc	0.07	0.19	0.47	0.70	2.40	5.70
Shaft HP	**88.7**	**240.9**	**595.8**	**887.4**	**3042.5**	**7225.8**
ΔCL	0.7	1.05	1.45	1.25	1.8	2.6

With the CFJ system now deemed impractical, the more traditional configuration of flaps and slats was back under consideration. Heavily influenced by the Zenith STOL aircraft, the HR-UAS was given full length permanent leading edge slats. The slats were designed to be permanently extended for ease of manufacture, use in flight, and to save weight on an already heavily loaded aircraft. During steady level flight the effects on lift and drag of the deployed slats is fairly insignificant[4] as is shown in Figure 5. When the wing is at low angles of attack, the influence of slats is small. Fowler flaps were chosen for the trailing edge due to their relatively high lift performance capabilities. The Fowler flaps were sized such that they generate adequate improvements to the lift while leaving enough room for effective ailerons. It was determined that the total flap span should cover 35% of the wing span to yield a $\Delta C_{Lmax} = 0.48$. A planform drawing of the flap and slat configuration can be seen in Figure 6. The dark shaded region represents the flapped planform area of the wing. Combining both flaps and slats, the maximum lift coefficient of the wing becomes $C_{Lmax} = 2.32$.

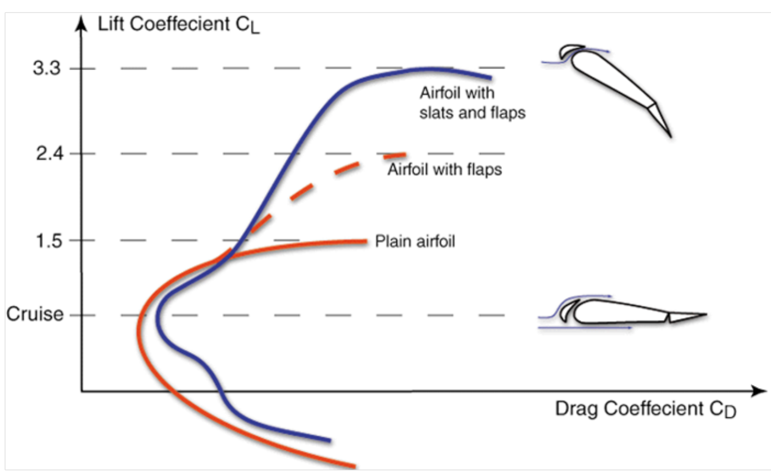

Figure 4. Example Drag Polar with Flaps and Slats

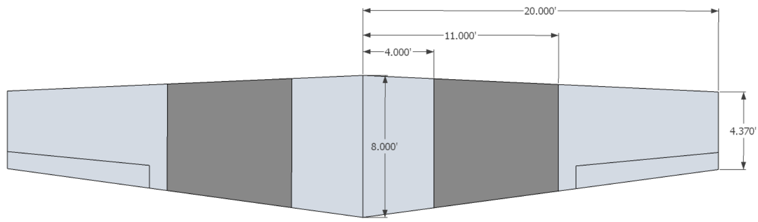

Figure 5. Planform Layout of Flaps and Ailerons

3. Engines

In the preliminary design, engine selection was important to determine how much power the aircraft needed. Based on preliminary calculations, roughly 1000shp was needed to accomplish STOL capabilities, but STOL capabilities depends on power and wing design. The Skytruck was an example of these capabilities as it had two turboprop engines with a smaller wingspan. The HR-UAS could not support two engines due to the constraints of the folding wings. Thus, with a reduction of power, the HR-UAS needed a larger wingspan but one engine. The next decision was between piston engines, turboprop, turbofan, and turbojets. Turbofans and turbojets provide too much power, too much weight, and a high fuel consumption to be considered as the main power for a light unmanned cargo aircraft. Piston engines provided light weight with small power around 400 shp, but the team needed an engine fair on weight and that produced a great amount of power. The turboprop engine provides anywhere from 700 shp to 1600 shp in the PT6 class series produced by Pratt and Whitney. Another deciding factor against piston engines is that turboprops had reverse thrust capabilities, which is crucial for STOL. Table 7 contains the specifications for that engine.

Table 5. PT6A-60A Engine Specifications

Equivalent Shaft HP	1113	hp
Shaft HP	1050	hp
Jet Thrust	157	lbs
Output RPM	1700	RPM
Gas generator RPM	39000	RPM
Maximum Reverse	900	hp
Oil Tank Capacity	2.5	gal
Usable Oil Tank Capacity	1.5	gal
Length	72.09	in
Nominal Diameter	18.29	in
Maximum Radius	12.84	in
Weight	487	lbs

A difficult task was the selection of the engine. A trade study was analyzed to determine how ease of manufacture, power requirements, maximum reverse, weight, reliability/maintenance, size of the engine, and the impact on other systems compared to each other. From there, three turboprop engines were compared to each other to determine which engine would be best suited for the aircraft. The three engines are the PT6A-50, PT6A-60A, and PT6A-65AR. The trade study is located in Tables 8 and 9, and the higher the number, the engine performs better in that particular field. The PT6A-50 is a turboprop engine that favors maximum reverse, the PT6A-60A favors a low weight compared to the others, and the PT-65AR favors maximum power of the three engines. The PT6A-60A was chosen because of the low weight, and it offers an average of power and maximum reverse between the other two engines.

Table 6. Top Level Criteria Weights for Engine Selection. Scores: 1 - Equal Importance, 3 - Moderate Importance, 5 - Strong Importance, 7 - Very Strong Importance, 9 - Extreme Importance

	Ease of Production	Power	Maximum Reverse	Weight	Reliability	Size	Systems Impact	Row Totals	Weight
Ease of Production	1.00	0.14	0.14	0.20	0.20	0.33	1.00	3.02	**2.62%**
Power	7.00	1.00	0.33	5.00	5.00	7.00	7.00	32.33	**28.04%**
Maximum Reverse	7.00	3.00	1.00	5.00	5.00	5.00	7.00	33.00	**28.62%**
Weight	5.00	0.20	0.20	1.00	0.33	3.00	5.00	14.73	**12.78%**
Reliability	5.00	0.20	0.20	3.00	1.00	3.00	5.00	17.40	**15.09%**
Size	3.00	0.14	0.20	0.33	0.33	1.00	7.00	12.01	**10.41%**
Systems Impact	1.00	0.14	0.14	0.20	0.20	0.14	1.00	2.83	**2.45%**

Table 7. Weighted Scores for Engine Selection. Scores: 100 - Excellent, 60 - Acceptable, 30 - Marginal, 0 - Not Addressed

	Ease of Production	Power	Maximum Reverse	Weight	Reliability	Size	Systems Impact	Total
PT6A-50	70	60	100	60	60	40	60	**69.62507226**
PT6A-60A	70	80	80	100	60	60	60	**79.01478239**
PT6A-65AR	70	100	80	80	60	60	60	**77.86522421**

4. Propellers

Once the PT6A-60A was chosen, a propeller had to be chosen to harness the power the engine was producing. After some research, a Hartzell 4 bladed propeller with a 105 inch diameter was used for that particular engine for a Beech aircraft. The 105 inch diameter could not be used due to the C-130 constraint; thus, Equation 4 was used to reduce the diameter without losing any power, but it increased the number of blades. The propeller was changed to a 70 inch, 6 blade propeller. The specifications and tip speed are located in Table 10. The tip speed is important because the propeller should not be spinning at or above the speed of sound. This causes increased drag, decreased propeller efficiency, and structural damage. The advantage of having this type of propeller is that it will be a variable pitch propeller allowing the operator of the aircraft to decide on the desirable pitch of the blades, and they can choose which propeller efficiency is needed. The propeller efficiency plotted as a function of advance ratio is in Figure 7. The advance ratio is the distance the propeller moves forward during one revolution given propeller diameter. The disadvantage is that it will reduce the propeller efficiency due to the smaller diameter.

$$D_2 = D_1 \left(\frac{\#ofpropellerblades_1}{\#ofpropellerblades_2} \right) \quad (4)$$

Table 8. Hartzell Propeller Specifications

Diameter	70	inches
	5.83	ft
RPM	1700	RPM
	28.33	rev/s
Pitch	123.69	inches
Vtip	519.24	ft/s
Vtip helical (100 ft/s)	528.78	ft/s

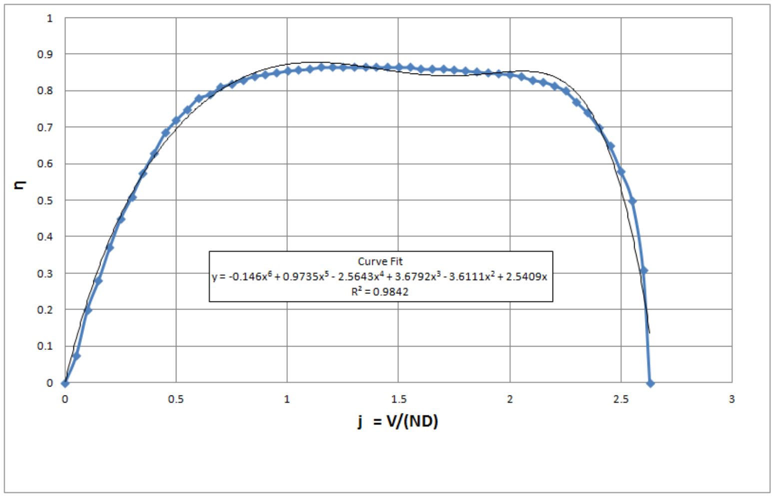

Figure 6. Variable Pitch Propeller Efficiency Curve

5. Loading and Unloading

The loading and unloading system used in this humanitarian relief vehicle was required to meet certain speed and performance criteria as well as be simple in operation. As always, simplicity in design and manufacturing was important as well. The user must be able to unload or load two pallets, each weighing 900 lbs, in less than 30 minutes. This process must take place at a forward operating base with unimproved runways. Finally, the engine must be powered down during unloading and loading.

From the initial brainstorming down to the final concept the entire design process was iterative. Early on, many concepts were explored, such as a Beluga style front loader, a traditional tail ramp, and something less orthodox such as a suspended floor. The front loader was ruled out primarily due to its impact on electrical and hydraulic systems. Any cables running the length of the plane would need to be cut for the front end to swing open. Also, expensive manufacturing processes result from the structural requirements of this type of system that are not present in other designs.

Next, the suspended floor was analyzed. In this system, the floor where the pallets were located would drop to ground level by assistance of a winch system. This system was particularly attractive due to the ease of operation. Unfortunately, as the rest of the vehicle was being designed, many issues arose in the placement and structure of the landing gear. Because this vehicle would be used for short takeoff and landing on unimproved runways, the landing gear would be experiencing substantial forces meaning it had to be structurally sound. Also, this system would require a great deal of extra structure within the fuselage.

After lengthy research and a trade study, found in Tables 2 and 3, the much more traditional style rear loading system was chosen. The tail section of the plane would fold down to become a ramp. This design has very little impact on other systems while staying simple in design.

Table 9. Top Level Criteria Weights for Loading and Unloading Configuration. Scores: 1 - Equal Importance, 3 - Moderate Importance, 5 - Strong Importance, 7 - Very Strong Importance, 9 - Extreme Importance

	Ease of Production	Safety	Ease of Use	Reliability	Weight	Systems Impact	Power	Row Total	Criteria Weight
Ease of Production	1.00	0.14	0.50	0.50	0.17	0.14	0.33	2.79	**3.41%**
Safety	7.00	1.00	3.00	2.00	3.00	4.00	4.00	24.00	**29.34%**
Ease of Use	2.00	0.33	1.00	0.33	0.50	0.33	2.00	6.50	**7.95%**
Reliability	2.00	0.50	3.00	1.00	2.00	2.00	0.33	10.83	**13.25%**
Weight	6.00	0.33	2.00	0.50	1.00	2.00	3.00	14.83	**18.14%**
Systems Impact	7.00	0.25	3.00	0.50	0.50	1.00	0.50	12.75	**15.59%**
Power	3.00	0.25	0.50	3.00	0.33	2.00	1.00	10.08	**12.33%**

Table 10. Top Level Criteria Weights for Loading and Unloading Configuration. Scores: 1 - Equal Importance, 3 - Moderate Importance, 5 - Strong Importance, 7 - Very Strong Importance, 9 - Extreme Importance

	Ease of Production	Safety	Ease of Use	Reliability	Weight	Systems Impact	Power	Score
Suspended Floor	40	50	90	60	50	20	40	**48.25**
Rear Tail Ramp	60	70	50	70	60	70	60	**65.02**
Front Unloading	40	60	50	70	70	50	60	**60.10**

A schematic of the loading system can be seen in Figure 4. The rear of the aircraft folds down and has channels for the tires of an all terrain pallet jack to travel in. A winch located behind the firewall of the airplane provides assistance for loading the heavy pallets.

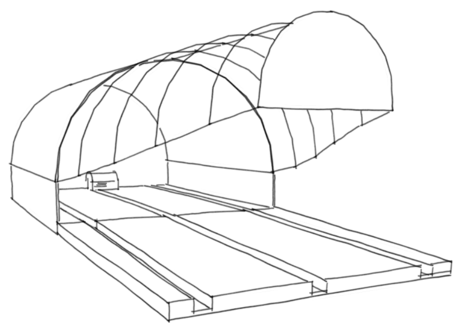

Figure 7. Loading System Schematic

6. Landing Gear Systems

According to the AIAA flight requirements, the aircraft should be expected to land and take off from rough terrain. The first landing gear design started with a tail dragger configuration. This plan was thought best since the impact on the tail wheel is lower than most other designs. However, a tail dragger arrangement can complicate the loading and unloading of the aircraft cargo. Trade studies were performed to analyze what aspects of a landing gear system were more important and which type of configuration would better suit those needs. Tables 11 and 12 feature the results of these studies showing that the tricycle gear was better suited for landing on rough terrain. Given the complications of the tail dragger design, and the results of the trade study, a nose wheel is used so that the aircraft is level after landing. The landing gear is fixed. This adds drag but increases the simplicity and strength of the design, which in turn usually is a cost saver. Figure 8 shows the nose wheel design.

Table 11. Top Level Criteria Weights for Landing Gear Configurations. Scores: 1 - Equal Importance, 3 - Moderate Importance, 5 - Strong Importance, 7 - Very Strong Importance, 9 - Extreme Importance

	Rough Terrain Capabilities	**Ease of Production**	**Weight**	**Reliability**	**Systems Impact**	**Total**	
Rough Terrain Capabilities	1.00	6.00	3.00	3.00	3.00	16.00	**43.44%**
Ease of Production	0.16	1.00	1.00	1.00	3.00	6.16	**16.74%**
Weight	0.33	1.00	1.00	2.00	2.00	6.33	**17.19%**
Reliability	0.33	1.00	0.50	1.00	0.33	3.16	**8.60%**
Systems Impact	0.33	0.33	0.50	3.00	1.00	5.16	**14.03%**

Table 12. Weighted Scores for Landing Gear Configurations. Scores: 100 - Excellent, 60 - Acceptable, 30 - Marginal, 0 - Not Addressed

	Rough Terrain Capabilities	**Ease of Production**	**Weight**	**Reliability**	**Systems Impact**	**Total**
Tricycle Gear	80	50	50	60	80	**68.10**
Tail Dragger	30	60	60	60	30	**42.76**

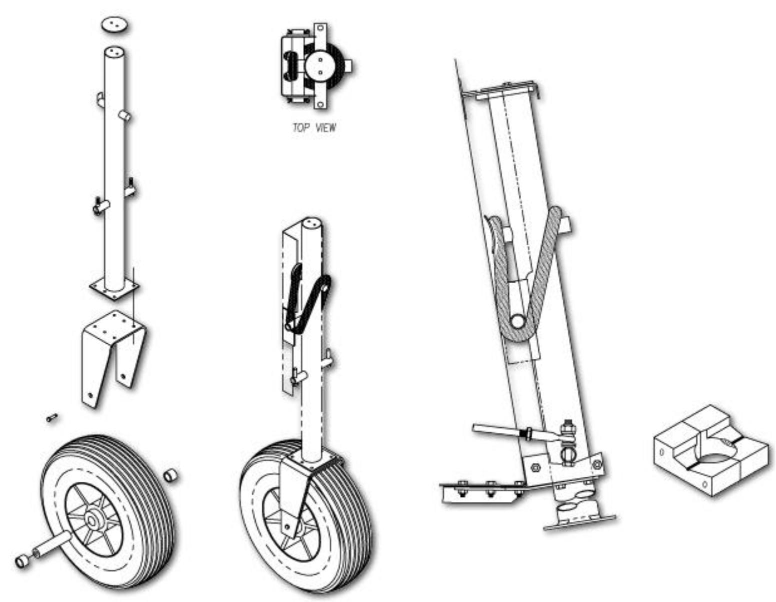

Figure 8. Front Nose Landing Gear System

The main landing gear must be strong to withstand sudden impacts when landing on short rough runways. The strength comes from a solid beam that runs through the aircraft from one main wheel to the other. Other than this feature, all components was assumed standard. Figure 9 shows a M-28 Skytruck with the approximate main landing gear that were used in the UAV design.

Figure 9. Main Landing Gear with Main Landing Gear Brace Structure

One can see from the figure above how the solid rigid structure would be able to withstand rougher landings that might occur on short unimproved runways.

7. Fuselage Structure

The structure used for the UAV is a very common style known as semi-monocoque. Figure 10 is a picture depicting the two most common structure types.

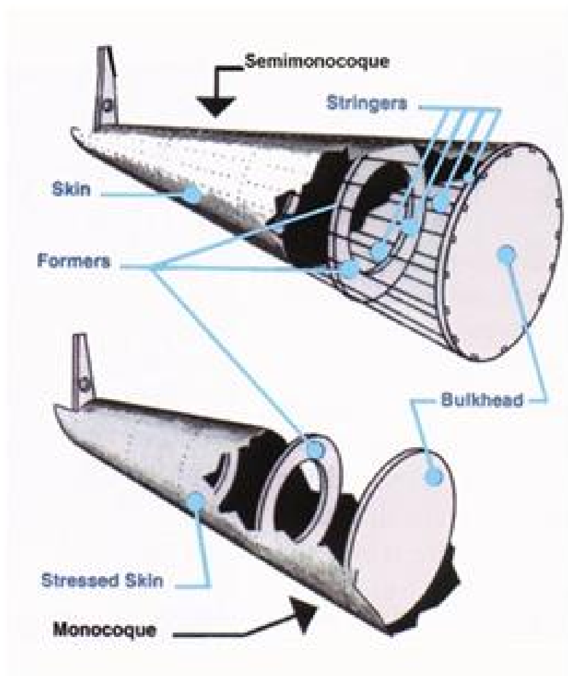

Figure 10. Semi-Monocoque and Monocoque Fuselage Structures

The difference between the semi-monocoque and the monocoque is that the latter does not use any interior structure but only the outer skin for all applied loads. The semi-monocoque style uses both stringers and longerons to form a strong interior structure that works with a strong outer shell structure. This of course added more weight but also added to the overall strength of the structure. This is needed since the aircraft will be flying with a heavy cargo load.

8. Wing Structure

Throughout the design process, numerous changes to the wing were made to better deal with the heavy loads the wing had to support. Initially, the wing had a fifty five foot span to easily accommodate the short take off and landing requirement. However, this proved to be difficult to properly support. The wingspan was shortened to forty feet, which greatly reduced the moments acting at the wing root. Preliminary designs of the aircraft had a cantilevered wing. As the structural requirements for this were closely examined, it was determined that the size of spar necessary to support the wing in this configuration were so large that the spar would not fit in the wing itself. To remedy this, it was decided that the aircraft should have a braced wing rather than a cantilevered one. The addition of the brace was effective in reducing the moments about the wing root to a magnitude with which the wing could be supported by spars made of inexpensive materials of reasonable size.

The brace was designed to run from the fuselage of the aircraft to a point thirteen feet from the root of the wing. This brace was to be constructed of Aluminum 2014-T6, a common material in aircraft structures. Using results from the vortex lattice method applied to the wing, the necessary force to be carried by the brace was determined to be nearly 20 kips. After applying a factor of safety of 1.8 to this value, the brace was designed to be capable of carrying a load of nearly 35 kips. This required a brace diameter of 0.92 inches.

Since the wing was no longer cantilevered, it was beneficial to have the wing be pinned at the root such that there was no resistance to moments at the pin. This allowed easier calculations of internal loads and bending resistances. The maximum internal shear load found in the wing when the load factor is three is 5,560 pounds. The maximum internal moment under these conditions is 32,600 feet-pounds. The internal support of the wing was designed using these maximum values as criteria which must be met. Shown below are plots of internal shear and internal moments.

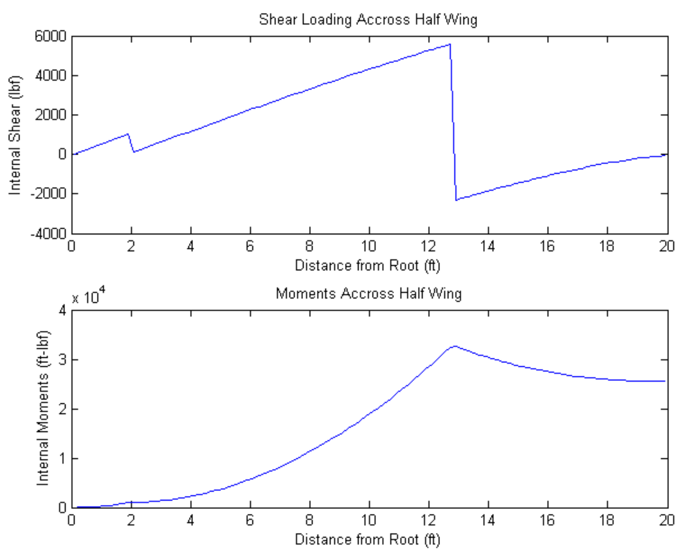

Figure 11. Shear and Moment Distribution

The structure of the wing consists of two spars, located at distances of twenty-five and seventy percent of the chord from the leading edge; 5 stringers, one inch in width; and aluminum skin. The area moment of inertia required to withstand the maximum internal bending was calculated to be 53.2 in^4. Sixty percent of this required moment of inertia was to be provided by the main spar at one quarter of the chord length from the leading edge. Fifteen percent was provided by the rear spar and an additional fifteen percent was provided by the stringers. This left the remaining ten percent to be provided by the aluminum skin. The front and rear spars were designed as I-beams made of Aluminum 2014-T6. The stringers were designed from the same material and extend across the span of the wing. Dimensions of the spars are shown below.

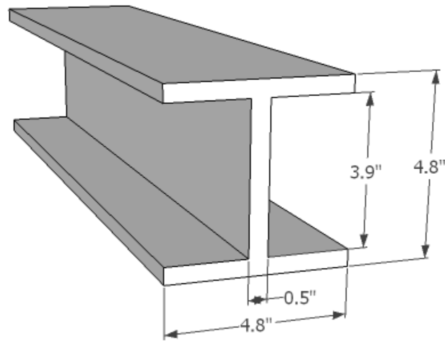

Figure 12. Main Spar

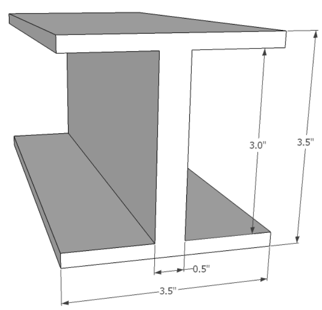

Figure 13. Rear Spar

9. Materials

The material selection for the UAV pertains to only the main structural components. The outer skin material will be made of a thermoplastic composite. This will reduce weight to the overall design significantly. Also, the thermoplastic composite is made to withstand sudden impacts from debris which will most likely be encountered on any unimproved runway. The loading and unloading door and the floor of the aircraft was taken to be made of Aluminum 7075, which has a higher strength than the more common aircraft Aluminum 2024. All other structural components (stringers and longerons) will be made of Aluminum 2024.

IV. Detailed Design

A. Basic Parameters

The basic parameters of the aircraft went through many changes during the design process. The wingspan was increased then decreased while the total length of the plane was shortened. The idea of a rectangular wing was adapted then changed to a wing with sweep, but the tail remained rectangular. The overall general characteristics of the aircraft are listed in Table 13.

Table 13. Basic Parameters

		Wing			Tail
b =	40	ft	**bt =**	9.47	ft
ct =	8	ft	**ctt =**	2.83	ft
cr =	4.37	ft	**crt =**	2.83	ft
le =	2.6	deg	**St =**	26.80	ft2
	0.0454	rad			
S =	247.4	ft2			**Fueselage**
AR =	6.5		**l =**	27.5	ft
Taper =	0.546		**w =**	5	ft
			h =	8.825	ft

B. Weight and Balance

The overall weight and balance of the aircraft was initially estimated at simple values that seemed very reasonable and translated well into what we felt would be good performance for the plane. However, our initial design created somewhat of an awkward looking aircraft, which then led to unconventional placements of items within the aircraft. The original placement of the wings and the shape of the tail created a few problems with the center of gravity calculations, and the cargo system, as well as the cargo itself, created problems that had to be adjusted from their original estimations. Not only were placement approximations changed, but weight estimations were corrected through the application of detailed weight calculations, forming more specific totals for both a cargo/transport aircraft and a general aviation plane. Since our aircraft is somewhat of a hybrid of the two types of planes, we decided to use both totals and average the component weights together to obtain the specific weights of our plane. Components such as the wing, fuselage, tail sections, cargo, and fuel were analyzed through the weight calculator for both general aviation and cargo planes, averaging the values together to obtain our aircrafts weight as shown in Tables 14 and 15.

Table 14. **Cargo and Transport Weight Calculations (lbs)**

Wing	492.582
Horizontal tail	35.152
Vertical tail	15.442
Fuselage	958.657
Main landing gear	154.094
Nose landing gear	54.575
Nacelle group	0.000
Engine controls	8.200
Starter (pneumatic)	33.330
Fuel system	39.297
Flight controls	232.966
APU installed	0.000
Instruments	22.876
Hydraulics	69.187
Electrical	165.865
Avionics	80.934
Furnishings	93.303
Anti-ice	11.460
Handling gear	1.719
Cargo handling system	153.600
Cargo weight	2100.000
Fuel	1200.000
Total Weight	5923.240

Table 15. **General Aviation Weight Calculations (lbs)**

Wing	545.330
Horizontal Tail	34.035
Vertical Tail	11.269
Fuselage	475.940
Main Landing Gear	216.723
Nose Landing Gear	59.354
Installed Engine (Total)	773.889
Fuel System	98.533
Flight Controls	54.593
Hydraulics	0.041
Electrical	177.622
Avionics	81.444
Furnishings	268.486
Anti-Ice	0.000
Cargo Weight	2100.000
Fuel	1450.000
Total Weight	6313.226

Once the weight components were determined and averaged to create the values for our aircraft, the center of gravity was the next obstacle to tackle. Analysis of six different scenarios was performed in order to evaluate the performance of the aircraft in different stages of flight. Scenarios of full fuel, half fuel, and empty fuel were analyzed for both fully loaded cargo and no cargo as can be seen in Table 16.

Table 16. Load Scenarios

Case 1	Full fuel w/ pallets
Case 2	Half fuel w/ pallets
Case 3	Empty fuel w/ pallets
Case 4	Full fuel w/o pallets
Case 5	Half fuel w/o pallets
Case 6	Empty fuel w/o pallets

With the original placement of the wings and cargo, our aircraft was not very stable. Using the dimensions of the finalized wing, the aerodynamic center was determined to be 11.51 ft from the nose of the plane. With the aerodynamic center at this location, the center of gravity had to be located in front of this value. All of the original placements and determined weights gave us a center of gravity well in front of the aerodynamic center; however, it was too far in front. For a good stable aircraft, the aerodynamic center and the center of gravity should only be 4-7 percent apart and the original calculations resulted in a 12-15 percent difference between the two values. The main change performed was moving the wings almost a full four feet forward from their original location on the aircraft. We experimented with adding extra dead weight to the aircraft but instead found this a very wasteful solution to the problem. After analyzing several possible solutions, the decision was made to move the avionics and flight control systems to the rear of the plane. Although very unorthodox, the benefits of making such a move far outweighed the cost of a few extra design components because the center of gravity values were moved to between 4-6 percent of the aerodynamic center for each scenario. The component locations, weights, CG locations, and CG placements are located in Tables 17 and 18.

Table 17. Component Locations and Weights

Type	Distance (ft)	Weight (lb)	Moment (ft.lb)
Fuselage	10.3	680	7004
Wings	11	615	6765
Tail	25.1	40	1004
Motor w/propeller	2.75	650	1787.5
Cargo	10.75	2250	24187.5
Fuel	11	1450	15950
Half Fuel	11	725	7975
Landing Gear (Main)	12	185	2220
Landing Gear (Nose)	3.5	50	175
Avionics/Systems	23	350	8050
Flight Controls	23	115	2645
Loading System	10.75	150	1612.5
Extra Weight	0	0	0

Table 18. Scenario Center of Gravity Locations

	Weight (lbs)	Total Moment (ft.lb)	CG (ft)	% Difference Of AC and CG
Case 1	6385	69788	10.930	5.044
Case 2	5660	61813	10.921	5.122
Case 3	4935	53838	10.909	5.222
Case 4	4285	47213	11.018	4.277
Case 5	3560	39238	11.022	4.245
Case 6	2835	31263	11.028	4.197
		Avg CG=	10.971	

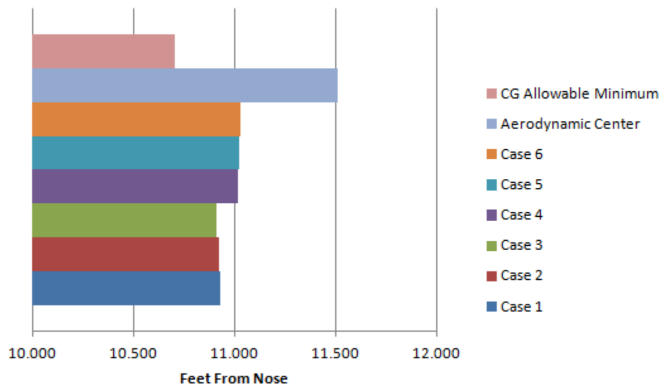

Figure 14. CG Placement

After repositioning items in the original design of the aircraft, the stability of the plane both in the air and on the ground was solidified. The resulting balance can perform exceptionally for the mission at hand without compromising the simplicity of design.

C. Avionics

The avionics for the HR-UAS include both onboard systems and a ground unit. Onboard systems are composed of servo motors, GPS, imaging systems, and guidance systems which together allow for fully autonomous flight and terminal area operations such as take off, landing, and taxiing. The ground component had to meet the specific requirements of automated control and have an option for pilot-in-the-loop control. The ground station equipment had to be transportable by backpack and weight less than 50 lbs. The control surfaces were controlled using servos. All other parts were selected the from Cloud Cap Technology catalogue and can be found in Appendix D. The onboard avionics are located within the tail of the vehicle.

D. Loading and Unloading

Resulting from research and a trade study which can be found in Tables 2 and 3 the much more traditional style rear loading system was finalized. The tail section of the plane would fold down to become a ramp. This design had very little impact on other systems while staying simple in design. The uniqueness of the design was in the ease of operation. To alleviate the requirement of equipment at the forward operating base an all-terrain pallet jack was included in the design. Also, to help in the movement of the 900 lb pallets up the ramp a battery operated winch was placed at the front of the cargo area and was attached to the pallet jack using cables. The wheels of the pallet jack were guided using tracks built into the floor of the cargo area and ramp.

The final design specifications are listed in Table 19 below. A commercially available winch and pallet jack were selected to show the feasibility of this design based on already available technology. The technology was available, however small modifications to the parts were needed. A C1000 Crane Winch by Superwinch was selected for its 1000 lbf of line pull and its light weight, and it was modified to facilitate two cables. The Vestil All-T-2 pallet jack was chosen for its 2000 lb load capacity and all-terrain capabilities; it was modified to fit the dimensions of the loading area as well as be able to attach to the winch system.

Table 19. Loading and Unloading Specifications

Parameter	Specification
Ramp Height	2.04 ft
Ramp Length	6.71 ft
Required Winch Force	275.6 lbf
Cable Tension	138 lbf
Pulley Bending Moment	19.5 lbf-ft
Pulley Shear Stress	194.8 lbf

E. Folding Wing

In order reach the climb and cruise requirements, the wingspan was set at 40 feet. Unfortunately, with a 40 foot wingspan, the vehicle would not fit in the cargo area of a C-130. Two options were under consideration to bring the overall width to under 10 feet. The first option was to detach the wings. This provides ease in design and manufacturing but makes the assembly process longer and more difficult. A second option considered was to have the wings fold back. This requires a greater level of design and manufacturing than the detachable wings, but it allows the plane to be easily assembled even by one person prior to operation. In the circumstances of natural disaster and with the use of many volunteers, the skill level and number of the ground crew could be limited. In this case, the huge increase in ease of operation outweighed the slight increase in manufacturing cost.

Once the folding concept was settled upon, the specifics of that fold were explored. The components of the fold had to support the bending moment and shear forces that would occur during flight, during the folding process, and during storage. The fold also needed to have connections for the servos that control the flaps and ailerons. Finally, the folding process ideally should be simple enough for one person to operate.

To determine the required material and dimensions, the case with a load factor of three was studied. This case occurs during flight. Using the vortex lattice method found in Appendix C, the bending moment and shear force were calculated at four feet from the root chord during flight. This was the chosen location for the folding joint as it allowed one foot of clearance on either side of the C-130 cargo bay. The bending moment was determined to be 2130.3 ft-lbf and the shear stress was 1113.3 lbf. By using Aluminum alloy 2024 which has a yield strength of 289 MPa and applying a factor of safety of 1.5 a two inch diameter rod was required to make the connection. This material was chosen due to its ability to resist corrosion and its high yield strength. This is a very common material used in aircraft components.

For ease of operation the process is comprised of two simple steps. First the leading edge of the wing rotates up. After this is complete, the wing rotates back. The strut is connected to the wing in line with the pivot axis. It is connected to the wing and to the fuselage using hinges with two degrees of freedom. These connections rotate allowing the wing to be rotated back without any tools. During rotation the connection to the power supply have to be broken. These connections were made using pins that connect on assembly.

F. Drawings and 3-D Models

As in most aircraft design procedures, a scale model is used to accurately test flight performance for the full size aircraft. The obvious reason for this is to reduce total cost of designing the aircraft. Cost reduction would come from the ability to run multiple tests and to make design changes all on a smaller scale. Although certain parameters such as skin friction coefficients, Reynolds number, and moving mechanical systems (engine) are not represented precisely on the model, the overall shape of the exterior of the plane is conveyed. The exterior shape of the model will be essential given that wind tunnel testing is performed in order to derive accurate flight characteristics for the full size aircraft. The design of the model used for this project started by using the 3D computer aided drafting (CAD) program Catia version 5. Figure 15 is an isometric view of the model inside the CAD program Catia.

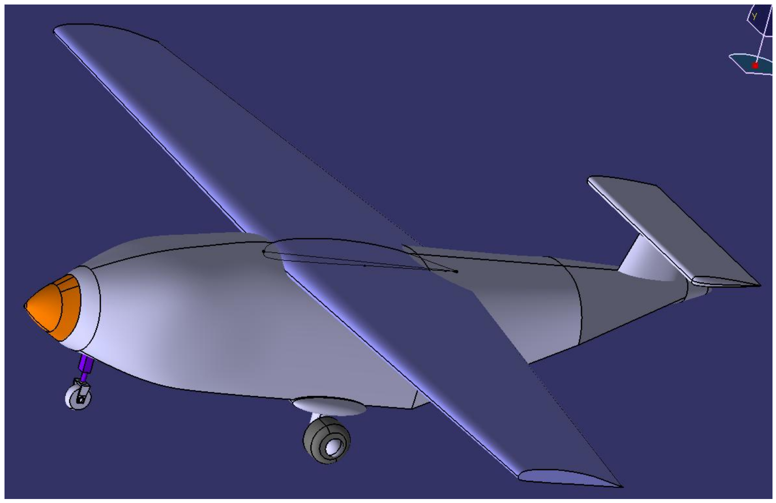

Figure 15. Catia Version 5 Isometric view of the Model Aircraft

Initially the design started with a basic aircraft shape which came from hand sketched drawings. The design is constantly updated whenever features of the aircraft change. CAD designing is used in order to obtain a 3D model for testing, but it is also useful when any measurements of the aircraft are needed. Surface area of the wing, tail, and fuselage are just some of the difficult measurements that are needed when designing an aircraft. These measurements are easily obtained as well as others by using the CAD program Catia. Figure 16 is a 2D drawing obtained from the CAD program showing a simple to scale layout of the aircraft.

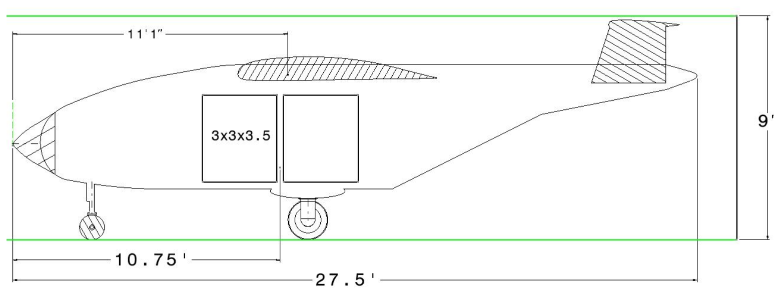

Figure 16. Catia Version 5 Cross Sectional Cut View Showing Height Restriction and Key Dimension Values

The aircraft must fit inside a zone of nine feet tall, 10 feet wide, and 50 feet long. The drawing shows that the plane fits inside the maximum height of nine feet. Excess space is both in front and behind the plane. The folding wing design will allow the aircraft to fit inside the 10 foot width zone. To obtain an actual model for testing, Mr. Jeffrey D. Wilkinson of the University of Tennessee-Knoxville's Art and Architecture building was contacted for his services. Mr. Wilkinson used a ZCorp ZPrinter 310 plus which builds the model in four pieces using the materials of ZP 131 powder and ZB60 binder. Once finished, the model is very fragile, and is coated in Elmer's Rotted Wood Repair which gives the model a higher strength factor. Once coated and assembled the model is ready for testing in the wind tunnel. In order to accurately measure the forces on the model during testing, a fixture is needed which fits both the wind tunnel force gage and the model itself. This fixture was made out of aluminum in the University of Tennessee Dougherty building machine shop.

V. Performance Results

A. Cruise Speed

The design team originally wanted the airplane to cruise at $V_{TR_{min}}$ =150 knots. This was to conserve fuel and have the aircraft cruise as economically as possible. However, since a turboprop engine was used, the gas generator section must be run at 62% N1. This is the flight idle speed and is required for the engine to continue running. This gives a shaft horsepower of 651 HP and a fuel burn of 358 $\frac{lb}{hr}$. In order to calculate the cruise speed of the airplane at this power output the thrust was compared with the drag.

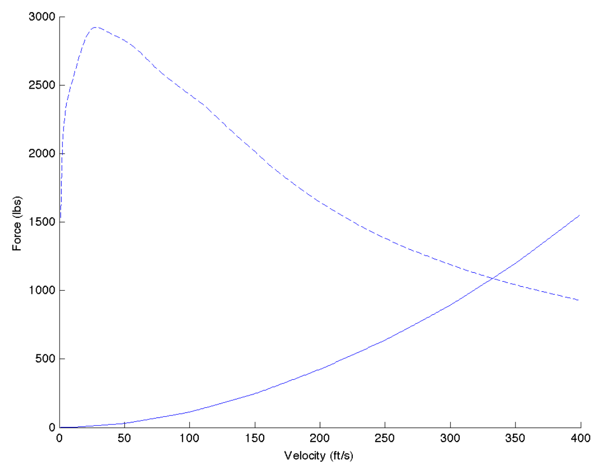

Figure 17. Cruise Speed Determination

As can be seen in Figure 17, the forces are balanced at 332.3 ft/s or 197 knots. This is the point where the Newton's Second Law is balanced for the flight direction, making this the minimum speed the plane can cruise at in steady level flight with the engine turned on and propeller properly pitched. Since the fuel must be spent to keep the engine running, this is the slowest the HR-UAS will cruise.

B. Takeoff and Landing

The required takeoff and landing distance of 500' on unimproved runways was one of the primary drivers for the overall aircraft design. The design was going to be heavy compared to most STOL aircraft and 500' is not a very large distance so an accurate means to calculate takeoff and landing was needed. In the early parts of the design process the methods of Anderson[5] were used. This method is not highly accurate but provided a good starting point and helped the design team to determine some guidelines for the range of critical parameters such as weight and C_L. Based on the estimates provided by Anderson's method, It became clear to the team that the larger constraint was related to takeoff and not landing. Landing an aircraft is more difficult to perform consistently and varies significantly between pilots, conditions, and even landing attempts making it more difficult to compute in a highly accurate manner. Since the airplane will land autonomously, it can be assumed it will fly the most efficient technique and with the reverse thrust provided by the the turboprop, high friction from landing on unimproved surfaces, large drag from fully extended flaps, and low stall speed it became clear that if the plane could takeoff in 500', it could land in 500'.

The takeoff constraint contributed to much of the aircraft design including flaps, slats, and the engine choice.

As the design progressed, the team began to use the takeoff code developed by Lynn and MacMillin.[6] This code required more inputs but also generated more accurate takeoff distances and helped to verify the team's design decisions. A numerically integrated takeoff code was eventually developed by the team once the design had progressed far enough to have meaningful and accurate data for lift, drag, thrust, and weight.

The numerical takeoff code, seen in Appendix A, considers forces and acceleration in the horizontal direction of travel and also the direction perpendicular to the ground. By summing the forces of drag, lift, thrust, and weight, the total force in both directions can be determined and acceleration is calculated by using the known aircraft mass and Newton's Second Law:

$$\sum \bar{F} = m\bar{a} \tag{5}$$

The code starts with the airplane at rest at the beginning of the runway with the engine at full power. The aircraft releases its brakes and the code iterates through time until the plane lifts off ground. The code makes several assumptions including standard day sea level conditions with no wind. It also ignores moments imposed on the airplane, forces in the yaw direction such as P-factor which would be counteracted by the rudder and cause more drag, and makes other simplifying assumptions. The code was broken into different helper functions that calculate individual forces and do the integration work in order to simplify updates as the design progressed and changes were made. Forces are mainly calculated as curve fits to data calculated through various means by the designers working in different areas.

Table 20. Numerically Integrated Takeoff Distance

Distance (ft)	Velocity (ft/s)	Time (sec)
430.96	112.26	7.10

Table 20 shows the distance, velocity, and time before the airplane leaves the ground. This takeoff is within the design limit of 500 ft. Table 21 shows the time and distance to clear a $50ft$ obstacle. It is likely that the distance to clear $50ft$ will be shorter for the actual HR-UAS since ground effects were neglected in the team's analysis.

Table 21. Takeoff Distance to Clear an Obstacle

Distance Along Ground(ft)	Airplane Altitude (ft)	Velocity (ft/s)	Time (sec)
1041.81	50.23	130.01	12.12

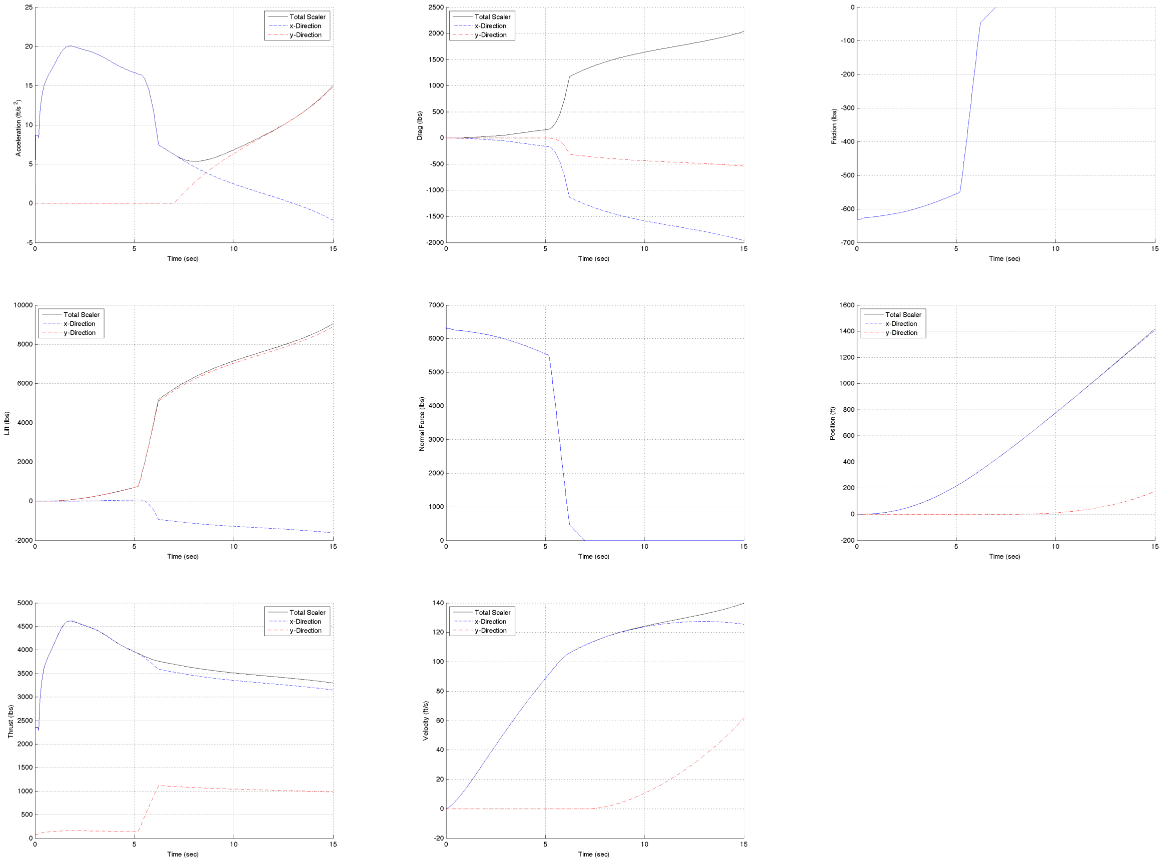

Figure 18. Calculated Takeoff Parameters

Figure 18 shows different values calculated by the takeoff code as the aircraft progresses with time. The sudden change in slope of many of the values are due to the plane rotating for takeoff.

A similar method was used for the landing integration. The landing code in Appendix B starts at the moment the main wheels contact the unimproved runway surface. The plane rolls freely until the nose wheel touches down and then brakes are applied until the aircraft comes to a stop. The results of the integration can be seen in Table 22 and Figure 19 respectively.

Table 22. Numerically Integrated Landing Distance

Distance (ft)	Time (sec)
441.36	6.40

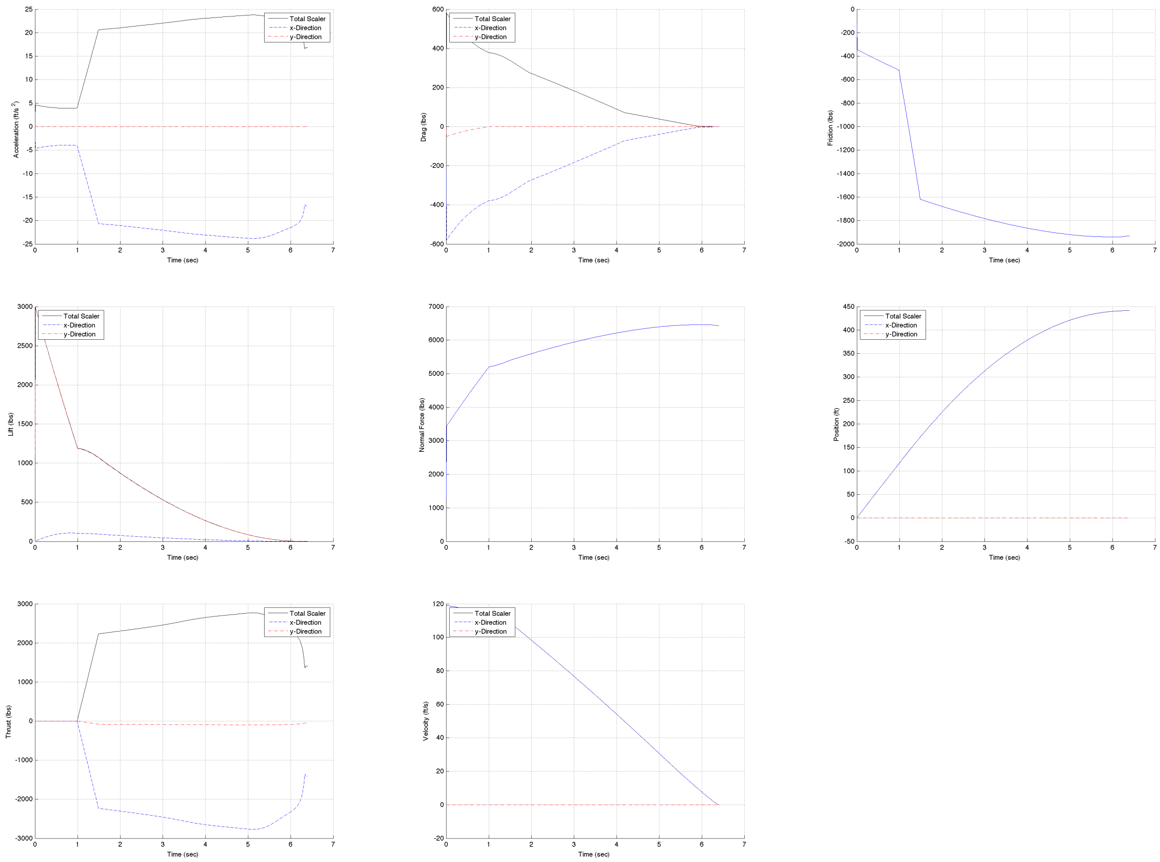

Figure 19. Calculated Landing Parameters

C. Aerodynamic Analysis

1. Lift

A vital performance characteristic of any plane is its ability to generate lift. To predict the amount of lift generated by the HR-UAS, the design team used the vortex lattice method to solve for the vortex strength across the wingspan. From the known vortex strength, the lift per unit span across the wing was found. The magnitudes of the lift per unit span are dependent upon the angle of attack, velocity, and air density. If one plots this lift distribution, however, the plot will maintain its shape regardless of the relative magnitudes. A generic plot of lift distribution is shown below to demonstrate this shape.

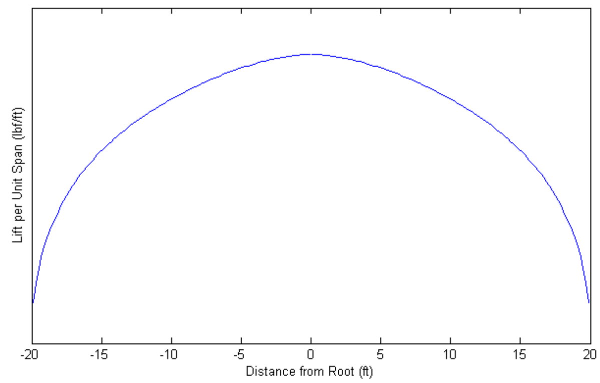

Figure 20. Lift Distribution Across Wing

From the known lift distribution, the derivative of the lift coefficient with respect to angle of attack was determined. Since the vortex lattice method assumes inviscid flow, it does not account for the presence of a boundary layer and is therefore not valid in the stall regime. The plot shown below is of lift coefficient versus angle of attack in the range of operation at which the HR-UAS is designed.

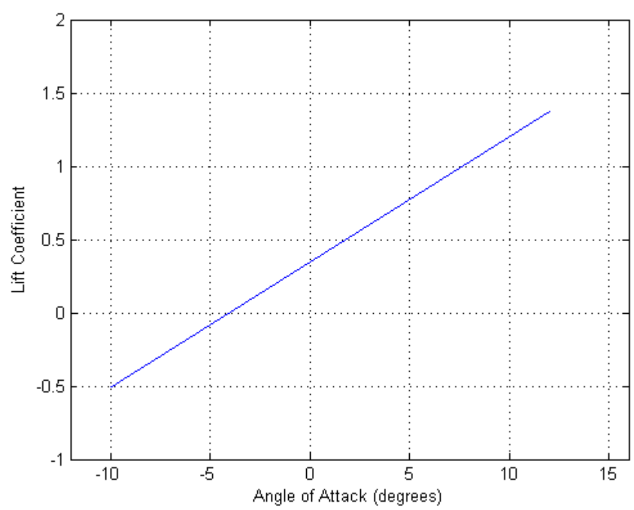

Figure 21. Lift Coefficient as Function of Angle of Attack

2. Drag

Drag is an important part of the analysis to determine aerodynamic characteristics of the aircraft, and it is measured two ways: analytically and experimentally. The analytical analysis involved using the aircrafts dimensions to determine the drag. The experimental analysis involved placing the model in a wind tunnel to calculate drag. Drag was calculated by analyzing the different components of the plane including the wing, fuselage, empennage, landing gear, and high-lift devices. Drag is essential to determine cruise speed, top speed, take-off distance, etc. For each component, zero lift drag and lift induced drag were calculated. Every component has zero lift drag and lift induced drag, but the lift induced drag of the landing gear is much smaller than the wing. The zero lift drag includes skin friction, viscous pressure drag, and fuselage upsweep drag. The lift induced drag includes vortex drag produced by the wingtips, and lift-dependent viscous drag. Wave drag was ignored since the HR-UAS is not going to be flying in the transonic or supersonic region.

The analytical analysis of drag assumes standard day conditions at sea level and at 8000 ft. The zero lift coefficient for the aircraft decreases due to the decreasing skin friction coefficient. The skin friction coefficient decreases due to the increasing speed of the plane. The drag coefficient due to lift is analyzed for angles of attack from -6 to 16 degrees. The lift induced drag highly depends on the square of the coefficient of lift; thus, as lift increases, so does drag. The total coefficient of drag is analyzed for velocities from 0 ft/s to 400 ft/s at 50 ft/s intervals at each angle of attack.

Tables 23 and 24 display the coefficient of drag for a specific angle of attack and velocity at sea level and 8000ft. The coefficient of drag remains the same as velocity is increased for a specific angle of attack, which it deviates a small amount due to human error of interpolating values off of a plot. The coefficient of drag decreases then increases from angles of attack of -6 degrees to 16 degrees. This is valid because it follows the same trend as the coefficient of lift as at low angles of attack, the coefficient of lift is small, but at higher angles of attack, there is an increase in coefficient of lift. Fig. 22 and Fig. 23 plot the drag polar for the aircraft at sea level and 8000ft.

Table 23. Coefficient of Drag at Sea Level

α	CL	10	50	100	150	200	250	300	350	400
(deg)	-	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)
-6	-0.1506	0.062	0.055	0.053	0.051	0.049	0.047	0.046	0.046	0.046
-4	0.0098	0.049	0.042	0.039	0.037	0.035	0.033	0.032	0.032	0.032
-2	0.1702	0.043	0.036	0.033	0.031	0.030	0.027	0.027	0.026	0.026
0	0.3306	0.045	0.038	0.036	0.034	0.032	0.030	0.029	0.029	0.029
2	0.491	0.056	0.049	0.046	0.044	0.042	0.040	0.039	0.039	0.039
4	0.6514	0.074	0.067	0.064	0.062	0.060	0.058	0.057	0.057	0.057
6	0.8118	0.100	0.093	0.090	0.088	0.087	0.084	0.084	0.083	0.083
8	0.9722	0.134	0.127	0.125	0.123	0.121	0.119	0.118	0.118	0.118
10	1.1326	0.177	0.170	0.167	0.165	0.164	0.161	0.161	0.160	0.160
12	1.293	0.228	0.221	0.218	0.216	0.214	0.212	0.211	0.211	0.211
14	1.4534	0.287	0.280	0.278	0.275	0.274	0.272	0.271	0.270	0.271
16	1.6138	0.355	0.348	0.345	0.343	0.342	0.339	0.339	0.338	0.338

Table 24. Coefficient of Drag at 8000ft

α	CL	10	50	100	150	200	250	300	350	400
(deg)	-	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)
-6	-0.151	0.066	0.057	0.053	0.052	0.050	0.049	0.047	0.047	0.047
-4	0.010	0.052	0.043	0.040	0.038	0.037	0.035	0.034	0.033	0.033
-2	0.170	0.046	0.037	0.034	0.033	0.031	0.029	0.028	0.027	0.027
0	0.331	0.049	0.040	0.036	0.035	0.033	0.032	0.030	0.030	0.030
2	0.491	0.059	0.050	0.046	0.045	0.044	0.042	0.041	0.040	0.040
4	0.651	0.077	0.068	0.064	0.063	0.062	0.060	0.059	0.058	0.058
6	0.812	0.103	0.094	0.090	0.089	0.088	0.086	0.085	0.084	0.084
8	0.972	0.137	0.128	0.124	0.123	0.122	0.120	0.119	0.118	0.118
10	1.133	0.179	0.170	0.167	0.165	0.164	0.162	0.161	0.160	0.160
12	1.293	0.230	0.221	0.217	0.216	0.214	0.213	0.211	0.211	0.211
14	1.453	0.288	0.280	0.276	0.275	0.273	0.271	0.270	0.270	0.269
16	1.614	0.356	0.347	0.343	0.342	0.341	0.339	0.338	0.337	0.337

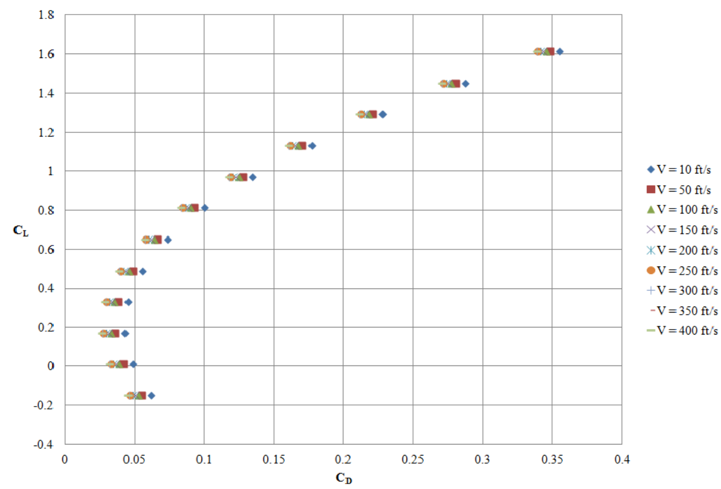

Figure 22. Drag Polar at Sea Level

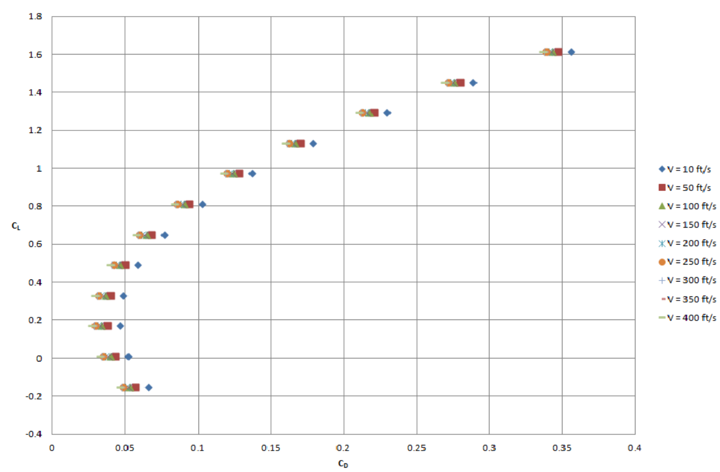

Figure 23. Drag Polar at 8000ft

The next step is to analyze the effects of flaps and slats on the aircraft. Profile drag, lift induced drag, and interference drag are included in the drag calculations for the flaps and slats. The increase of drag for the flaps and slats are located in Table 25. The next step is to determine how much drag force is affecting the aircraft at a certain velocity and angle of attack. Tables 26 and 27 display that information for sea level and at 8000 ft, and Tables 28 and 29 display the drag with 15 degrees and 30 degrees of flaps and slats at sea level. Fig. 24 plots drag as a function of velocity with no flaps, 15 degrees of flaps, and 30 degrees of flaps for sea level. Fig. 25 plots drag as a function of velocity with no flaps. The increase in coefficient of lift results in higher coefficients of drag, which is a result of implementing high lift devices such as flaps and slats. The increase in coefficient of drag increases the drag force shown in Fig. 24.

Table 25. CD of High Lift Devices

ΔCd_{flaps}	0.0464
ΔCd_{slats}	0.0128

Table 26. Drag (lbs) at Sea Level (No Flaps)

α (deg)	10 (ft/s)	50 (ft/s)	100 (ft/s)	150 (ft/s)	200 (ft/s)	250 (ft/s)	300 (ft/s)	350 (ft/s)	400 (ft/s)
-6	1.832	40.687	154.954	334.140	574.524	858.159	1213.672	1639.172	2146.537
-4	1.431	30.662	114.854	243.916	414.125	607.537	852.776	1147.953	1504.944
-2	1.266	26.546	98.393	206.878	348.281	504.654	704.626	946.303	1241.565
0	1.335	28.275	105.306	222.432	375.932	547.860	766.842	1030.986	1352.171
2	1.636	35.802	135.417	290.182	496.375	736.052	1037.839	1399.843	1833.944
4	2.171	49.174	188.902	410.523	710.315	1070.333	1519.203	2055.033	2689.703
6	2.942	68.454	266.025	584.050	1018.808	1552.353	2213.312	2999.793	3923.675
8	3.952	93.711	367.051	811.358	1422.911	2183.765	3122.545	4237.360	5540.088
10	5.204	125.009	492.244	1093.042	1923.682	2966.219	4249.280	5770.971	7543.172
12	6.701	162.415	641.868	1429.696	2522.178	3901.368	5595.894	7603.863	9937.154
14	8.444	205.995	816.187	1821.914	3219.455	4990.864	7164.768	9739.274	12726.262
16	10.437	255.814	1015.466	2270.291	4016.570	6236.357	8958.278	12180.440	15914.724

Table 27. Drag (lbs) at 8000ft (No Flaps)

α (deg)	10 (ft/s)	50 (ft/s)	100 (ft/s)	150 (ft/s)	200 (ft/s)	250 (ft/s)	300 (ft/s)	350 (ft/s)	400 (ft/s)
-6	1.515	32.774	122.974	269.358	465.890	701.985	985.159	1323.299	1720.182
-4	1.201	24.908	91.509	198.562	340.032	505.331	701.977	937.857	1216.748
-2	1.071	21.658	78.511	169.316	288.039	424.092	584.993	778.629	1008.777
0	1.123	22.973	83.772	181.152	309.081	456.970	632.337	843.069	1092.944
2	1.357	28.818	107.152	233.759	402.603	603.099	842.763	1129.483	1467.035
4	1.773	39.228	148.792	327.448	569.161	863.345	1217.517	1639.565	2133.264
6	2.375	54.255	208.897	462.686	809.584	1239.006	1758.469	2375.860	3094.957
8	3.162	73.950	287.677	639.940	1124.704	1731.381	2467.489	3340.915	4355.436
10	4.139	98.365	385.339	859.679	1515.351	2341.767	3346.445	4537.272	5918.025
12	5.306	127.553	502.091	1122.370	1982.357	3071.464	4397.208	5967.477	7786.048
14	6.667	161.565	638.139	1428.480	2526.552	3921.769	5621.648	7634.075	9962.829
16	8.222	200.454	793.693	1778.476	3148.768	4893.981	7021.633	9539.611	12451.692

Table 28. Drag (lbs) at Sea Level (15°Flaps)

α	10	50	100	150	200	250	300	350	400
(deg)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)
-6	2.703	62.483	242.138	530.304	923.259	1403.058	1998.327	2707.175	3541.479
-4	2.302	52.458	202.038	440.080	762.861	1152.436	1637.431	2215.955	2899.886
-2	2.138	48.342	185.577	403.042	697.016	1049.553	1489.281	2014.306	2636.507
0	2.207	50.071	192.490	418.596	724.668	1092.759	1551.497	2098.989	2747.113
2	2.508	57.598	222.601	486.346	845.111	1280.951	1822.494	2467.846	3228.886
4	3.043	70.970	276.086	606.687	1059.051	1615.232	2303.858	3123.036	4084.645
6	3.814	90.250	353.209	780.214	1367.544	2097.252	2997.967	4067.796	5318.617
8	4.824	115.507	454.235	1007.522	1771.647	2728.664	3907.200	5305.363	6935.031
10	6.076	146.805	579.428	1289.206	2272.418	3511.119	5033.935	6838.974	8938.114
12	7.572	184.211	729.051	1625.859	2870.913	4446.268	6380.549	8671.866	11332.096
14	9.316	227.791	903.371	2018.078	3568.190	5535.763	7949.423	10807.277	14121.204
16	11.308	277.610	1102.650	2466.455	4365.306	6781.256	9742.933	13248.443	17309.666

Table 29. Drag (lbs) at Sea Level (30°Flaps)

	10	50	100	150	200	250	300	350	400
(deg)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)	(ft/s)
-6	3.58	84.28	329.32	726.47	1271.99	1947.96	2782.98	3775.18	4936.42
-4	3.17	74.25	289.22	636.24	1111.60	1697.34	2422.09	3283.96	4294.83
-2	3.01	70.14	272.76	599.21	1045.75	1594.45	2273.94	3082.31	4031.45
0	3.08	71.87	279.67	614.76	1073.40	1637.66	2336.15	3166.99	4142.06
2	3.38	79.39	309.78	682.51	1193.85	1825.85	2607.15	3535.85	4623.83
4	3.91	92.77	363.27	802.85	1407.79	2160.13	3088.51	4191.04	5479.59
6	4.69	112.05	440.39	976.38	1716.28	2642.15	3782.62	5135.80	6713.56
8	5.70	137.30	541.42	1203.69	2120.38	3273.56	4691.86	6373.37	8329.97
10	6.95	168.60	666.61	1485.37	2621.15	4056.02	5818.59	7906.98	10333.06
12	8.44	206.01	816.24	1822.02	3219.65	4991.17	7165.20	9739.87	12727.04
14	10.19	249.59	990.55	2214.24	3916.93	6080.66	8734.08	11875.28	15516.15
16	12.18	299.41	1189.83	2662.62	4714.04	7326.16	10527.59	14316.45	18704.61

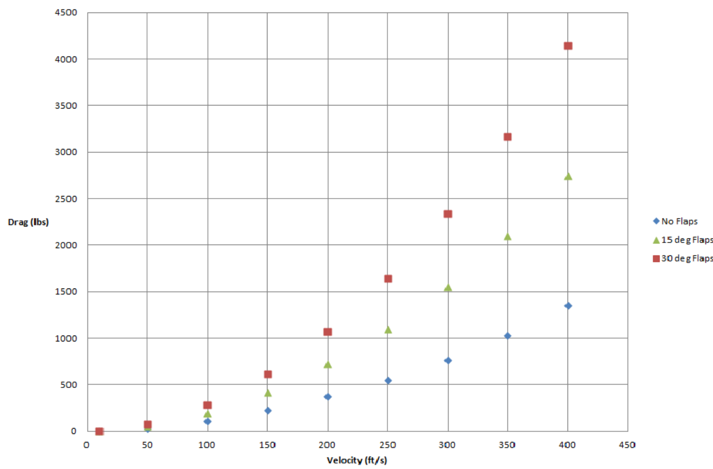

Figure 24. Drag vs Velocity at Sea Level

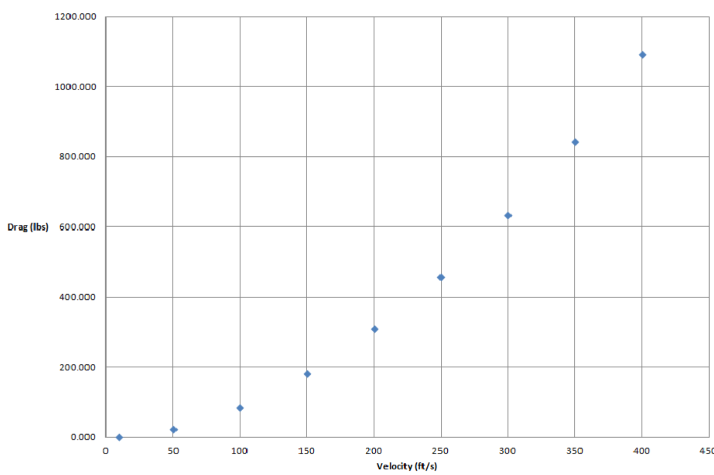

Figure 25. Drag vs Velocity at 8000ft

3. Wind Tunnel

After fabrication of the model was finished, it was then time to move into the testing phase. The model was scaled and to be tested in the large subsonic wind tunnel in the Dougherty engineering building because the computerized system allowed for easier calculations and lower chance of human error. Testing conditions, found in Table 30, were noted and used in the calculations. The test section in the wind tunnel was calibrated first to observe what drag was present without any model on the test stand. This calibration data is shown in Figure 27 Using the data from this calibration, the drag data was corrected in order to account for the testing aperture and to give more accurate drag forces from the model alone.

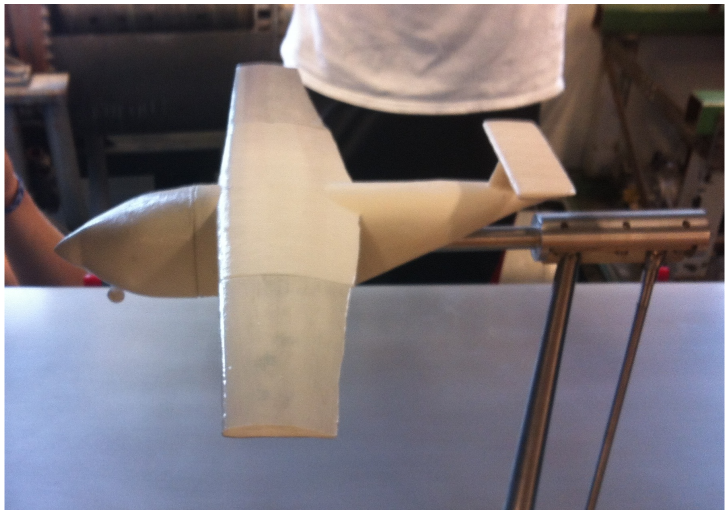

Figure 26. Wind Tunnel Model

Table 30. Test Conditions

Temperature	70.5	F
	530	R
Pressure	980	mbars
	14.214	psi
	2046.8	psf
Density	0.0022	slug/ft3
Splane	247.4	ft2
Smodel	0.2533	ft2

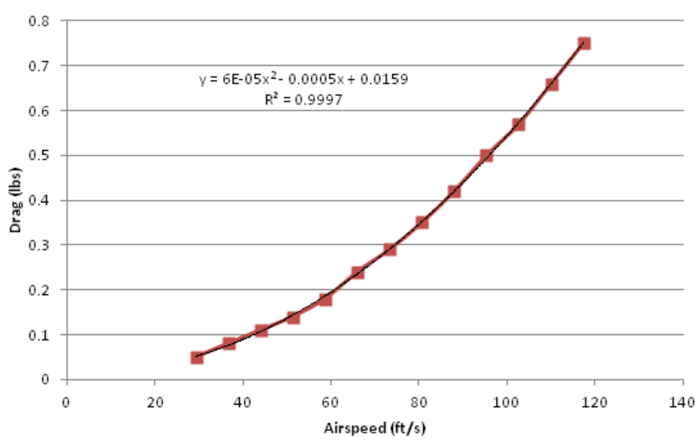

Figure 27. Drag vs. Velocity Calibration

The testing began with the model stationed at a zero angle of attack while the airspeed was varied. This allowed for a simulation of steady level flight at a cruising altitude. Lift and drag data was recorded as the airspeed was varied in increments of 5 mph. Drag readings were then corrected based on the drag calibration equation that was obtained earlier. The coefficients of lift and drag were calculated using the atmospheric conditions of the testing area as well as the general characteristics of the model. These coefficients of lift and drag are not only representations for the model aircraft, but can be translated to be used for the actual full size aircraft by simply using the planes wing area instead of the model's value. The coefficient of drag values were also corrected using the equation $CD_{corrected} = CD + 0.02CL^2$ and these values were recognized as the true data for zero angle of attack flight both for the model and the full scale aircraft. These values can be found in Table 31. The coefficient of drag values varied unusually at lower speeds but as the airspeed increased they became more consistent with each other and began to appear as a generally flat line. The airspeed was not pushed too high in order to maintain the integrity of the model and avoid breaking it before other testing was accomplished.

Table 31. Zero Angle of Attack Testing Data

Velocity MPH	Velocity ft/s	Lift lbf	Drag lbf	Corrected Drag lbf	CL	CD	Corrected CD
20	29.333	0.050	0.070	0.017	0.204	0.070	0.071
25	36.667	0.070	0.110	0.032	0.183	0.083	0.084
30	44.000	0.120	0.150	0.040	0.217	0.072	0.073
35	51.333	0.180	0.220	0.072	0.240	0.095	0.097
40	58.667	0.260	0.270	0.077	0.265	0.078	0.080
45	66.000	0.400	0.350	0.106	0.322	0.085	0.087
50	73.333	0.530	0.410	0.108	0.346	0.071	0.073
55	80.667	0.690	0.500	0.134	0.372	0.072	0.075
60	88.000	0.840	0.600	0.163	0.381	0.074	0.077
65	95.333	1.010	0.700	0.186	0.390	0.072	0.075
70	102.667	1.190	0.810	0.213	0.396	0.071	0.074
75	110.000	1.420	0.920	0.233	0.412	0.068	0.071
80	117.333	1.650	1.050	0.267	0.421	0.068	0.072

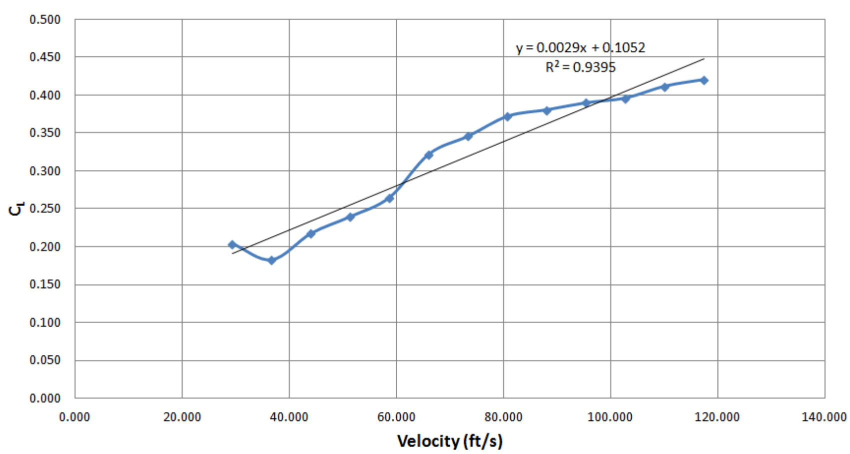

Figure 28. Zero Angle of Attack CL

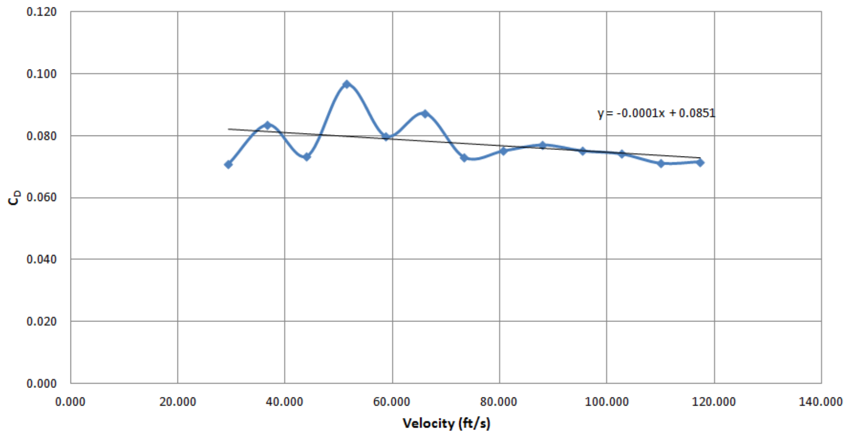

Figure 29. Zero Angle of Attack CD

To evaluate stall angles and the general performance of the aircraft, the model was then tested at varying angles of attack at different air speeds. For these tests, each airspeed was held constant while the angle of attack was changed, observing lift and drag values along the way, refer to Tables 32, 33, and 34 and Figures 30 - 35. Speeds of 35 mph (51.333 ft/s), 45 mph (66 ft/s), and 55 mph (80.667 ft/s) were used for testing variation of lift and drag as functions of AoA. Higher speeds were not tested due to concerns of pushing the model to the point of failure because of weakness in construction. The actual aircraft could easily perform at higher speeds, but the weaknesses of the model only allowed for testing at these lower speeds. However, the coefficients of lift and drag that were determined at each angle of attack, and derived from the lift and drag forces observed in testing, applied to the full scale model. These values were used to determine the lift and drag that would be present on the airplane at a certain airspeed and angle of attack. For a list of these values refer to Table 35.

Table 32. 35 mph Test Data

AoA (deg)	Corrected AoA (deg)	CL	CD	Corrected CD
-1.986	-1.933	0.046	0.079	0.079
-0.011	0.205	0.188	0.088	0.089
1.976	2.527	0.479	0.068	0.072
3.986	4.843	0.745	0.081	0.092
5.997	7.021	0.891	0.084	0.100
7.995	9.147	1.001	0.099	0.119
9.972	11.131	1.008	0.116	0.136
12.005	13.033	0.895	0.254	0.270

Table 33. 45 mph Test Data

AoA (deg)	Corrected AoA (deg)	CL	CD	Corrected CD
-3.982	-4.085	-0.090	0.071	0.071
-2.008	-1.917	0.079	0.065	0.065
0.000	0.316	0.275	0.061	0.063
1.988	2.664	0.588	0.062	0.069
3.998	4.876	0.764	0.053	0.065
5.997	7.042	0.909	0.065	0.082
7.984	9.136	1.002	0.072	0.092
10.006	11.277	1.106	0.100	0.125
12.005	13.086	0.941	0.226	0.244
13.992	14.977	0.857	0.284	0.299

Table 34. 55 mph Test Data

AoA (deg)	Corrected AoA (deg)	CL	CD	Corrected CD
-4.004	-4.031	-0.023	0.079	0.079
-2.995	-2.937	0.051	0.064	0.064
-2.008	-1.842	0.144	0.063	0.063
-1.009	-0.728	0.244	0.057	0.058
-0.011	0.394	0.352	0.055	0.058
0.847	1.418	0.497	0.054	0.059
1.988	2.688	0.609	0.054	0.062
3.004	3.803	0.695	0.062	0.072
3.986	4.878	0.775	0.060	0.072
4.992	5.972	0.853	0.065	0.079
5.997	7.029	0.898	0.068	0.084
7.013	8.136	0.976	0.073	0.092
8.018	9.255	1.076	0.092	0.115
9.001	10.287	1.118	0.097	0.122
9.994	11.321	1.154	0.112	0.139
10.988	12.204	1.057	0.180	0.203
12.005	13.098	0.950	0.229	0.247
13.010	14.061	0.914	0.263	0.280
14.015	15.069	0.917	0.291	0.308

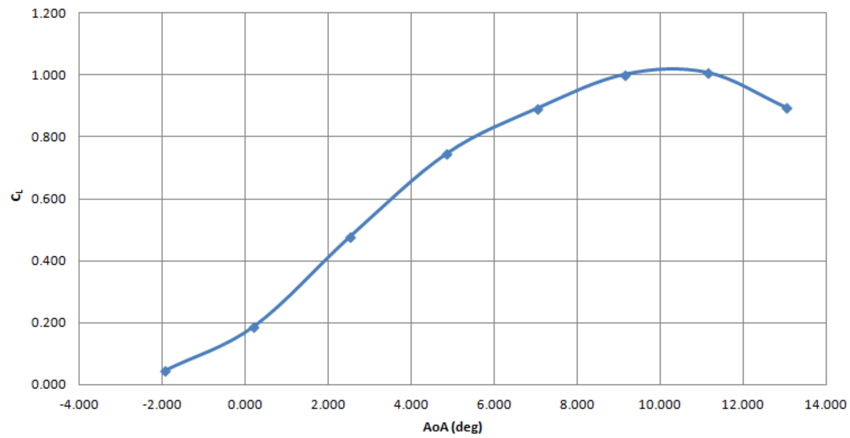

Figure 30. 35 mph CL vs. AoA

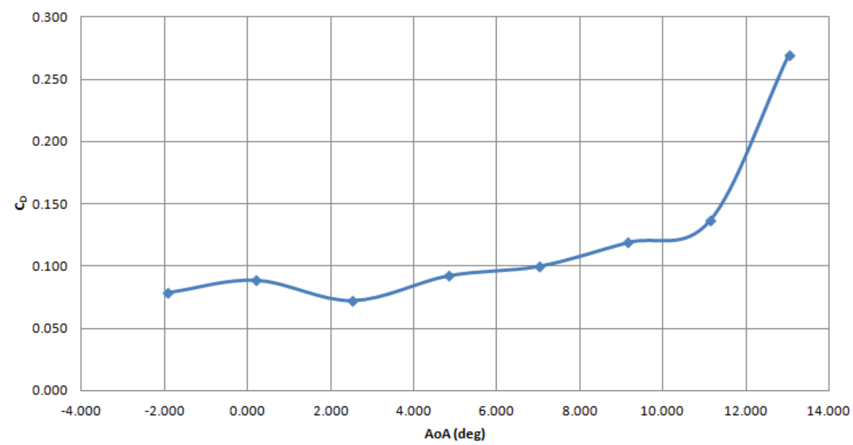

Figure 31. 35 mph CD vs. AoA

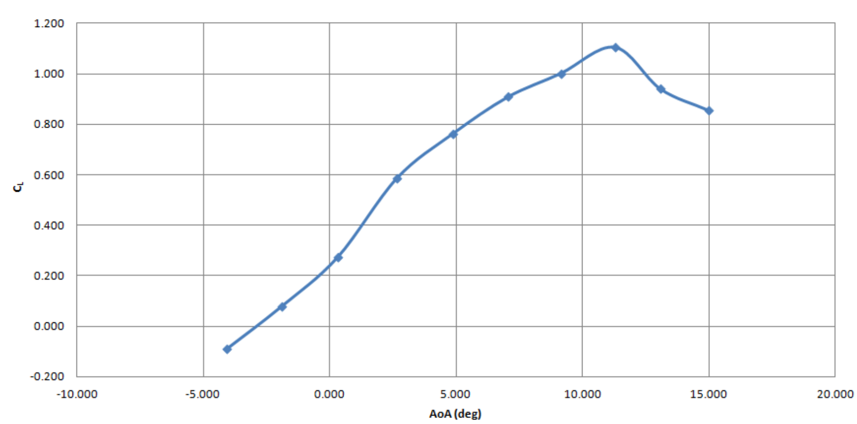

Figure 32. 45 mph CL vs. AoA

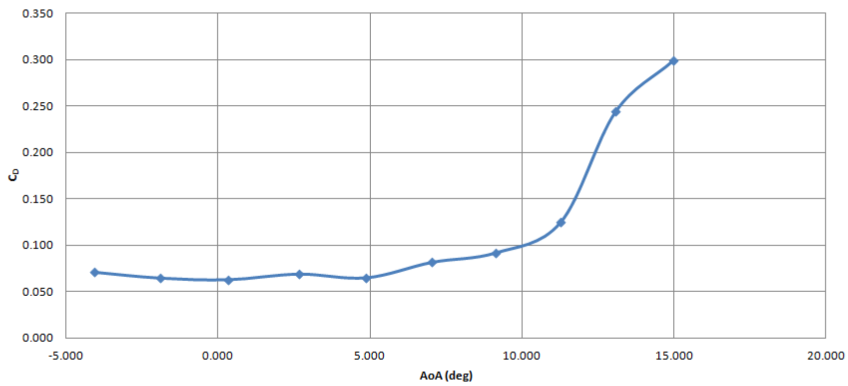

Figure 33. 45 mph CD vs. AoA

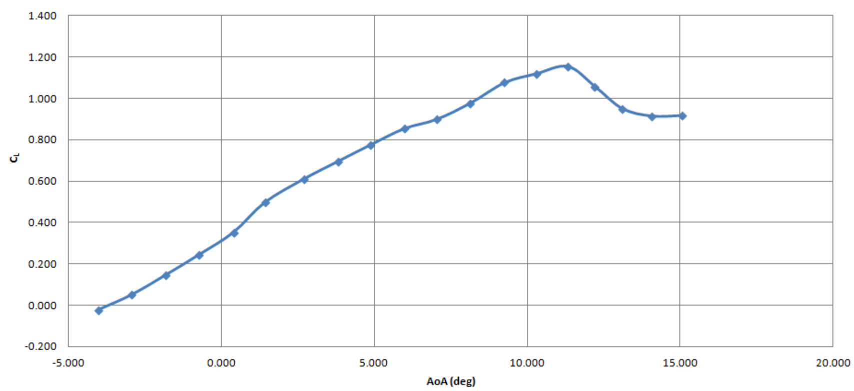

Figure 34. 55 mph CL vs. AoA

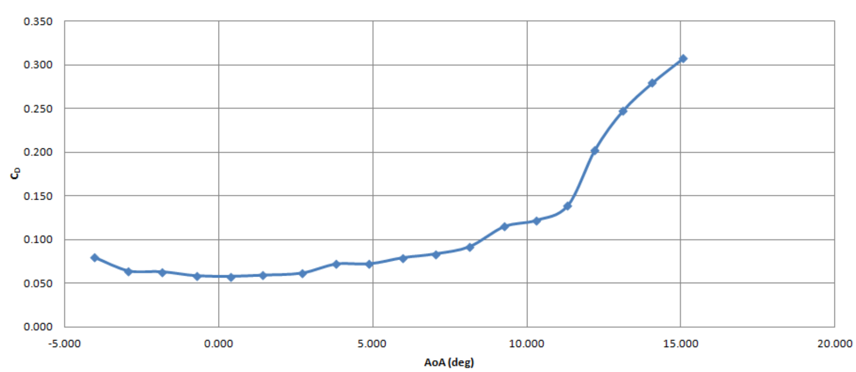

Figure 35. 55 mph CD vs. AoA

Table 35. Full Scale Lift and Drag

	35 mph		45 mph		55 mph	
AoA	**Full Scale Drag (lbs)**	**Full Scale Lift (lbs)**	**Full Scale Drag (lbs)**	**Full Scale Lift (lbs)**	**Full Scale Drag (lbs)**	**Full Scale Lift (lbs)**
-4			84.698	-106.717	143.370	-40.146
-3					115.830	93.101
-2	59.850	59.819	76.184	92.732	113.971	263.189
-1					105.489	451.454
0	64.180	63.669	72.594	317.201	104.485	648.118
1					106.802	913.113
2	54.394	50.956	81.230	692.045	111.123	1106.645
3					129.916	1251.123
4	68.124	59.936	75.855	892.116	130.247	1424.263
5					143.020	1531.520
6	72.993	61.402	95.070	1057.660	151.034	1630.630
7					166.001	1810.002
8	88.005	73.171	110.011	1200.189	207.462	1920.792
9					220.253	2050.130
10	103.341	87.962	147.422	1307.478	251.067	2078.026
11					366.289	1908.893
12	193.696	182.206	284.444	1096.518	446.395	1742.733
13					504.749	1662.769
14			353.390	1012.021	555.859	1636.271

From the data at each airspeed, the stall angle was observed to occur between 10 and 11 degrees for each of the scenarios. The limitations of the wind tunnel computer prevented the collection of data in a more controlled and concentrated range to find a more specific value for the stall angle. The coefficient of lift data was transferred into the takeoff calculations and checked against the already existing values; the drag data was compared to the drag calculations derived from the code. The wind tunnel data of lift and drag greatly enhanced the team's ability to understand how a full scale version of the aircraft would perform under different scenarios.

4. V-n Diagram

Based on the VLM predicted lift distribution, wing load factors were found as a function of velocity and plotted to generate the V-n diagram shown below in Figure 36. The figure shows a limit load factor of $n_{limit} = 3$. This represents the limit load where structural damage from maneuvering would occur. Accounting for a factor of saftey of 1.5 the ultimate load was taken to be $n_{ult} = 4.5$. The maneuver point, where the stall line intersects the limit load factor, is also shown at $V_{*} = 204 ft/s$. The flight envelope is represented as the area under the stall line and below the limit load factor. To avoid structural damage from maneuvering, the aircraft was designed to opperate within this envelope. Stall and cruise velocities are also illustrated on the figure.

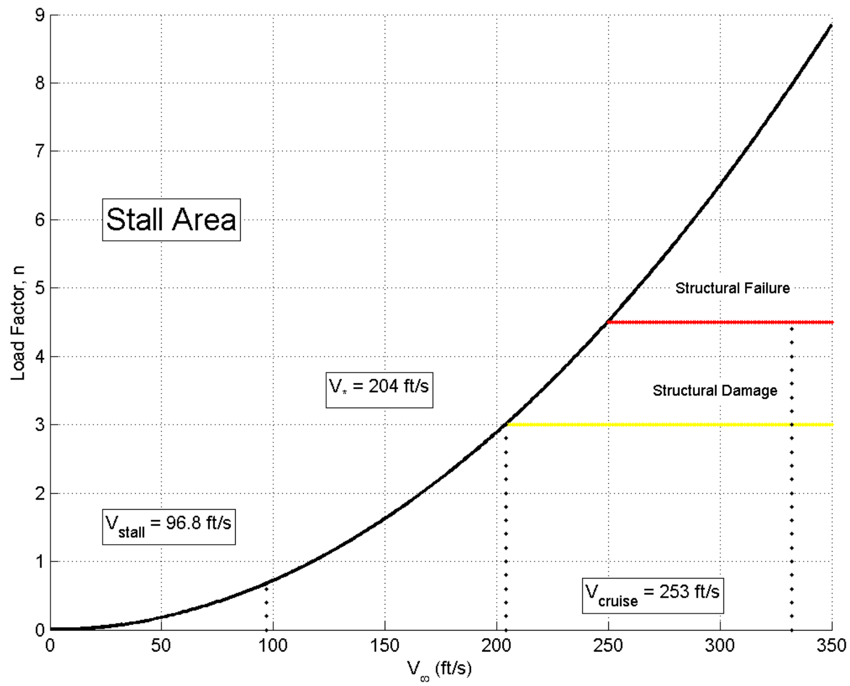

Figure 36. Load Factor vs. Free-stream Velocity

D. Mission Performance

The following gives the results for each step of the mission:

1. The plane is started at its fully loaded maximum gross weight of 6,386 lbs. The Specific Fuel Consumption for the PT6A-65B is .550 lb/SHP/hr. The PT6A-65B is a turboprop engine so, even at idle speeds, it still must operate at ~50% power. This leads to a fuel consumption of ~288 lbs/hr or 43 gal/hr of fuel. After 5 minutes of warm-up and taxi, the plane will burn 23 lbs of fuel.

2. The take-off occurs at full power at sea level. For the mission analysis it is assumed that the FOB is an unimproved runway. Using the take-off code described in Section B, the take-off was found to occur in 431 feet after 7.1 seconds burning 1 lb of fuel.

3. The plane continues under maximum power to climb to 8,000 ft MSL. The rate of climb is limited by the stall angle of the HR-UAS. By using the calculated C_{Lmax} =2.3 and extending the C_L versus α curve, α_{stall} was found to be 19.7°. Climbing at 3900 FPM, a speed of 127 knots, and a climb angle of 18° the plane reaches its cruise altitude in 123 seconds while covering a ground distance of 4.1nm. The aircraft then accelerates to its cruise speed. This process burns 19 lbs of fuel.

4. The aircraft cruises at the 197 knots due to engine constraints. At this speed, flying 300nm takes 1 hour and 31 minutes. The aircraft is at an angle of attack of 0° with the wings mounted at -5° to maintain steady level flight. This section of the flight burns 545 lbs of fuel.

5. The plane then pulls power back to flight idle and begins its decent at a standard angle of 3°. This decent takes 12 minutes and covers 25nm over the ground burning 71 lbs of fuel. The aircraft, having burnt 659 lbs of fuel and weighing 5,727 lbs, then lands on the unimproved runway at the RRA in 423

feet.

6. Cargo unloading/loading in 30 minutes is facilitated by the hoisting system and all terrain pallet jack carried by the aircraft. This will be performed with the engine turned off for safety and to conserve fuel. Once completed and the area and runway are cleared, the HR-UAS will restart its engine and idle for 5 additional minutes once again burning 23 pounds of fuel.

7. The airplane returns to the skies weighing a slightly less 5,704 lbs due to burnt fuel. The take-off covers just a little less ground, only requiring 302 feet after 5.5 seconds and burning 1 lbs of fuel.

8. The climb is also the same but with the reduced fuel load improving the climb rate to 4,300 FPM, at a speed of 136 knots, and a climb angle of 18°. This climb lasts 112 seconds, covering 4nm, and burning 17 lbs of fuel.

9. The cruise is roughly the same speed as before burning 545 lbs of fuel. The angle of attack will be slightly less due to the smaller weight.

10. The decent is performed identically to the first, at an angle of 3°. The aircraft then lands on the unimproved runway at the FOB in 402 feet where the engines are shut down. The final weight of the airplane is 5,070 lbs with 134 lbs of fuel remaining allowing 20 minute of cruise flight.

The HR-UAS meets all of the design requirement and successfully completes its mission. In total the plane flies for 3 hours, covers 658nm over the ground while carrying 1,800lbs of humanitarian supplies. It burns 1,316 lbs of fuel in the process.

E. Cost Estimate

The total cost of our plane is estimated at $2,513,000. The cost breakdown is shown below in Table 36. Items in the category others includes the winch, pallet jack, anti-ice systems, and folding mechanisms.

Table 36. Cost Estimate

Airframe	$800,000
Engine	$1,500,000
Avionics	$150,000
Propellor	$53,000
Other	$10,000
Total	**$2,513,000**

VI. Conclusions

Designing an aircraft is an extremely complicated iterative process. Many hours of computations, coding, plotting, and re-working problems are par for the course. Over the course of the last six months the HR-UAS design team has logged hundreds of hours working to produce the best product possible. From short take-off and landing to transport within a C-130J, the HR-UAS promises to deliver.

If the design process was to continue, the team would go back over nearly all aspects of the aircraft and refine them. Iterating several times in this way is the only way to refine and improve the existing design. Significant improvement and a more in depth analysis could be applied to nearly all aspects of the design.

The avionics and remote control would especially benefit from further analysis. Higher accuracy methods could be used for the high lift components, lift distribution, drag, weight calculations, and take-off and landing. With further revision in the HR-UAS design, another more detailed three dimensional model could be made and tested in the wind tunnel yielding more accurate and realistic results. Despite the potential for improvement an unmanned humanitarian aerial relief system that not only meets but exceeds all AIAA required performance characteristics has been presented.

References

[1]Organization, W. H., "Humanitarian Supply Management and Logistics in the Health Sector," .

[2]Raymer, D. P., *Aircraft Design: A Conceptual Approach*, AIAA, 4th ed., 2006.

[3]Zha, G., Dano, B. P. E., and Castillo, M., "Experimental Study of Co-Flow Jet Airfoil Performance Enhancement Using Discrete Jets," *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2011.

[4]Heintz, C., "Anatomy of a STOL Aircraft," Tech. rep., Zenith Aircraft Company, 2001.

[5]John D. Anderson, J., *Aircraft Performance and Design*, McGraw Hill, 2010.

[6]Lynn, S. and MacMillin, P., "TAKEOFF2.c," 1994.

[7]Bertin, J. J. and Cummings, R. M., *Aerodynamics for Engineers*, Pearson Education, Inc., 5th ed., 2009.

[8]Hill, P. G. and Peterson, C. R., *Mechanics and Thermodynamics of Propulsion*, Addison-Wesley Publishing Company, Inc., 2nd ed., 1992.

[9]Roskam, J., *Ariplane Design*, Darcorporation, 2nd ed., 2003.

Appendices

A. Takeoff Code

```
%% HR-UAS Takeoff/Landing Code
%% %% MAIN FUNCTION %% %%
function Takeoff_Code()
    clear all; close all; clc; format long g;
%%
   %Constants that define how the integration runs
    runTime = 15;
    NumSteps = 500;           %Total number of steps used in the integration
    timestep = runTime/NumSteps;  %Increment between timesteps (secs)

    %Other Constants Controlling the takeoff roll
    g = 32.174; %ft/s^2
    Vstall = 119.1298855; %ft/s^2
    VR = 0.7*1.1*Vstall; %ft/s^2
    ClimbAngle = 15;    %Degrees
    ObstacleHeight = 50;    %ft

    %Values to Compare the integration with:
    [ThrusteqnX, ThrusteqnY] = Thrust(VR, ClimbAngle);
    [DrageqnX, DrageqnY] = Drag(ClimbAngle, VR);
    [LifteqnX, LifteqnY] = Lift(VR, ClimbAngle, 0);

    %AE370Takeoff(W, S, CLmax, g, mu, rho, T, D, L, N)
    dist370 = AE370Takeoff(-Weight(1), 247.4, 2.314655933, 32.2, .3, Density(0), ...
        Total(ThrusteqnX, ThrusteqnY), Total(DrageqnX, DrageqnY), Total(LifteqnX, ...
        LifteqnY), 1)
    %AE370ApproxTakeoff(W, S, CLmax, g, rho, T)
    dist370_2 = AE370ApproxTakeoff(-Weight(1), 247.4, 2.314655933, 32.2, Density(0), ...
        Total(ThrusteqnX, ThrusteqnY))

    %Initialization and book keeping
    %Keep track of all critical integral values
    time = 0:timestep:NumSteps*timestep;%Vector of timestamps events happen
    Ax = zeros(NumSteps+1,1); %Vector of Acceleration in the x-dir (ft/s^2)
    Ay = zeros(NumSteps+1,1);   %Vector of Acceleration in the y-dir (ft/s^2)
    Vx = zeros(NumSteps+1,1);   %Vector of Velocity in the y-dir (ft/s)
    Vy = zeros(NumSteps+1,1);   %Vector of Velocity in the x-dir (ft/s)
```

```
    Sx = zeros(NumSteps+1,1);    %Vector of Position from R/W start in the y-dir (ft)
    Sy = zeros(NumSteps+1,1);    %Vector of Position from R/W start in the x-dir (ft)

    %Keep track of all force values (lbs)
    Lx = zeros(NumSteps+1,1); %Lift
    Ly = zeros(NumSteps+1,1);
    Dx = zeros(NumSteps+1,1); %Drag
    Dy = zeros(NumSteps+1,1);
    Tx = zeros(NumSteps+1,1); %Thrust
    Ty = zeros(NumSteps+1,1);
    W = zeros(NumSteps+1,1);  %Weight
    N = zeros(NumSteps+1,1);  %Normal Force on landing gear
    friction = zeros(NumSteps+1,1);
    Fx = zeros(NumSteps+1,1); %Sum of Forces in the x-dir
    Fy = zeros(NumSteps+1,1); %Sum of Forces in the y-dir


    %Keep up with other important parameters
    theta = zeros(NumSteps+1,1); %Angle between Ground and Flight Path (relative wind of ...
        V_inf)



%     %Plots Thrust versus speed
%     i = 1
%     for V = 1:1:400
%        [Tx(V), Ty(V)] = Thrust(V, 0);
%        T(i) = Total(Tx(V),Ty(V));
%        eta(i) = PropEff(V, 8.75);
%        i = i + 1
%     end
%     V = 1:1:400;
%     plot(V, T);
%     %title('Thrust vs. Speed');
%     xlabel('Velocity (ft/s)');
%     ylabel('Thrust (lbs)');
%     print(printOpts, sprintf('FigThrust'));

%% Integration
%Initial Values
W(1) = Weight(time(1));
N(1) = Normal(W(1), Ly(1), Dy(1), Ty(1));
[Tx(1), Ty(1)] = Thrust(Total(Vx(1),Vy(1)), theta(1));
takenoff = 0;
ClearedObstacle = 0;
    for i = 2:length(time)

        theta(i) = theta(i-1);
        if Total(Vx(i-1), Vy(i-1)) ≥ VR && theta(i-1) < ClimbAngle
           %Assume we can rotate to the ClimbAngle in one second
            theta(i) = theta(i-1) + ClimbAngle.*timestep;
        end


        [Lx(i), Ly(i)] = Lift(Total(Vx(i-1),Vy(i-1)), theta(i), Sy(i-1));

        [Dx(i), Dy(i)] = Drag(theta(i), Total(Vx(i-1),Vy(i-1)));

        [Tx(i), Ty(i)] = Thrust(Total(Vx(i-1),Vy(i-1)), theta(i));

        W(i) = Weight(time(i));

        N(i) = Normal(W(i), Ly(i), Dy(i), Ty(i));

        friction(i) = Friction(N(i), Dx(i) + Lx(i) + Tx(i));

        Fy(i) = N(i) + W(i) + Ly(i) + Dy(i) + Ty(i);

        Fx(i) = Dx(i) + Lx(i) + friction(i) + Tx(i);

```

```
        [Ay(i), Vy(i), Sy(i)] =  Integrate(Fy(i), W(i)./g, Ay(i-1), Vy(i-1), Sy(i-1), ...
            timestep);

        [Ax(i), Vx(i), Sx(i)] =  Integrate(Fx(i), W(i)./g, Ax(i-1), Vx(i-1), Sx(i-1), ...
            timestep);

        if N(i) == 0 && takenoff == 0
           Sx(i)
           Vx(i)
           Vy(i)
           Total(Vx(i), Vy(i))
           time(i)
           sprintf('Left the Ground at Sx=%f at time=%d', Sx(i), time(i));
           matrix2latex([Sx(i), Total(Vx(i), Vy(i)), time(i)] , 'tTakeoff.tex', 'label', ...
                'tab:Takeoff', 'columnLabels', {'Distance (ft)','Velocity (ft/s)', 'Time ...
                (sec)'}, 'caption', 'Numerically Integrated Takeoff Distance', 'format', ...
                '%#3.2f','alignment', 'c');
           takenoff = 1;
        end

        if Sy(i) ≥ ObstacleHeight && ClearedObstacle == 0
            [Sy(i) Sx(i)]
            matrix2latex([Sx(i), Sy(i), Total(Vx(i), Vy(i)), time(i)] , ...
                'tTakeoffObstacle.tex', 'label', 'tab:Takeoff', 'columnLabels', {'Distance ...
                Along Ground(ft) ', 'Airplane Altitude (ft)', 'Velocity (ft/s)', 'Time ...
                (sec)'}, 'caption', 'Numerically Integrated Takeoff Distance', 'format', ...
                '%#3.2f','alignment', 'c');
            ClearedObstacle = 1;
        end
    end

    %% Plots
    %Plot Configurations
    lineFormats = {'m*', 'k-', 'b--', 'r-.', 'g:', 'y.', 'cx'}; %First item is used last ...
        since matlab doesn't like zero indexing
    plotFormat = @(index)lineFormats{mod(index,length(lineFormats))+1};
    legstr = {'Total Scaler', 'x-Direction', 'y-Direction'};
    %printOpts = '-depsc';    %Color .eps plots
    printOps = '-dpng';     %Color .png

    figure;
    grid on;
    hold on;
    plot(1:653, PropEff(1:653, 8.75));
    hold off;
    xlabel('Velocity (ft/s)');
    ylabel('\eta - Propeller Efficiency');
    print(printOps, 'FigPropEff');

    figure;
    grid on;
    hold on;
    plot(time, Total(Ax,Ay), plotFormat(1));
    plot(time, Ax, plotFormat(2));
    plot(time, Ay, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthEast')
    ylabel('Acceleration (ft/s^2)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffAccel');

    figure;
    grid on;
    hold on;
    plot(time, Total(Vx,Vy), plotFormat(1));
    plot(time, Vx, plotFormat(2));
    plot(time, Vy, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Velocity (ft/s)')
```

```
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffVelocity');

    figure;
    grid on;
    hold on;
    plot(time, Total(Sx,Sy), plotFormat(1));
    plot(time, Sx, plotFormat(2));
    plot(time, Sy, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Position (ft)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffPos');

    figure;
    grid on;
    plot(time, W);
    hold off;
    ylabel('Weight (lbs)')
    xlabel('Time (sec)')

    figure;
    grid on;
    hold on;
    plot(time, N);
    hold off;
    ylabel('Normal Force (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffNormal');

    figure;
    grid on;
    hold on;
    plot(time, friction);
    hold off;
    ylabel('Friction (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffFriction');

    figure;
    grid on;
    hold on;
    plot(time, Total(Lx,Ly), plotFormat(1));
    plot(time, Lx, plotFormat(2));
    plot(time, Ly, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Lift (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffLift');

    figure;
    grid on;
    hold on;
    plot(time, Total(Dx,Dy), plotFormat(1));
    plot(time, Dx, plotFormat(2));
    plot(time, Dy, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Drag (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffDrag');

    figure;
    grid on;
    hold on;
    plot(time, Total(Tx,Ty), plotFormat(1));
    plot(time, Tx, plotFormat(2));
    plot(time, Ty, plotFormat(3));
```

```
    hold off;
    legend(legstr, 'Location', 'NorthEast')
    ylabel('Thrust (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigTakeoffThrust');

    %% Table of Values
    matrix2latex([time', Ax, Ay, Total(Ax, Ay)] , 'tAccel.tex', 'label', ...
        'tab:Acceleration', 'columnLabels', {'Time','Acceleration-x', 'Acceleration-y', ...
        'Total Acceleration'}, 'caption', 'Acceleration', 'format', '%#3.2f','alignment', ...
        'c');
    matrix2latex([time', Vx, Vy, Total(Vx, Vy)] , 'tVel.tex', 'label', 'tab:Velocity', ...
        'columnLabels', {'Time','Velocity-x', 'Velocity-y', 'Total Velocity'}, 'caption', ...
        'Velocity', 'format', '%#3.2f','alignment', 'c');
    matrix2latex([time', Sx, Sy, Total(Sx, Sy)] , 'tPos.tex', 'label', 'tab:Position', ...
        'columnLabels', {'Time','Position-x', 'Position-y', 'Total Distance'}, 'caption', ...
        'Position', 'format', '%#3.2f','alignment', 'c');
    matrix2latex([time', theta] , 'tTheta.tex', 'label', 'tab:Theta', 'columnLabels', ...
        {'Time','Theta'}, 'caption', 'Theta', 'format', '%#3.2f','alignment', 'c');
    matrix2latex([time', Lx, Ly, Total(Lx, Ly), Dx, Dy, Total(Dx, Dy)] , 'tFAero.tex', ...
        'label', 'tab:FAero', 'columnLabels', {'Time','Lift-x', 'Lift-y', 'Total Lift', ...
        'Drag-x', 'Drag-y', 'Total Drag'}, 'caption', 'Aerodynamic Forces', 'format', ...
        '%#3.2f','alignment', 'c');
    matrix2latex([time', W, N, friction, Tx, Ty, Total(Tx, Ty)] , 'tForces.tex', 'label', ...
        'tab:FAero', 'columnLabels', {'Time','Weight', 'Normal', 'Friction', 'Thrust-x', ...
        'Thrust-y', 'Total Thurst'}, 'caption', 'Other Forces', 'format', ...
        '%#3.2f','alignment', 'c');


end



%% %% FORCE CALCULATIONS %% %%
% All functions should return forces signed correctly for their
% positive/negative direction
%% Lift

% V = Velocity - ft/s
% Theta = Flight Path Angle = Degrees
% altitude = altitude = ft

function [Lx, Ly] = Lift(V, Theta, altitude)
    WingOffsetAngle = -5;
    AlphaL0 = -4;

    %Calculations for ThetaPrime were done with Flaps at 15 degrees
    ThetaPrime = 1.4;
    %CL = 0.0873.*(Theta+WingOffsetAngle-AlphaL0) + .3493;
    CL = 0.0802.*(Theta + WingOffsetAngle-AlphaL0) + .3306;
    CLF = 0.0802.*(Theta +ThetaPrime+ WingOffsetAngle-AlphaL0) + .3306;
    S = 247.4;          %(ft^2)
    SFlapped = 92.9425;
    SFlaps = 2.*4.631;
    g = 32.174; %ft/s^2

    L = 0.5.*Density(altitude)./g.*V.^2.*(S-SFlapped).*CL + ...
        0.5.*Density(altitude)./g.*V.^2.*(SFlapped + SFlaps).*CLF;

    Lx = -L.*sind(Theta+WingOffsetAngle);
    Ly = L.*cosd(Theta+WingOffsetAngle);
    return;
end

%% Drag

%alpha = Flight Path Angle of Plane = Degrees
%V = Velocity - ft/s

function [Dx, Dy] = Drag(alpha, V)
```

```
    %Data from Hooker's Drag Calcs for 15 degrees of flaps
    D_AlphaVals = [-6 -4    -2  0    2    4    6    8    10  12  14  16];
    D_VelocityVals = [0 10  50  100 150 200 250 300 350 400];
    DragVals = [0   2.703   62.483  242.138 530.304 923.259 1403.058    1998.327    ...
        2707.175    3541.479
        0   2.302   52.458  202.038 440.080 762.861 1152.436    1637.431    2215.955    ...
            2899.886
        0   2.138   48.342  185.577 403.042 697.016 1049.553    1489.281    2014.306    ...
            2636.507
        0   2.207   50.071  192.490 418.596 724.668 1092.759    1551.497    2098.989    ...
            2747.113
        0   2.508   57.598  222.601 486.346 845.111 1280.951    1822.494    2467.846    ...
            3228.886
        0   3.043   70.970  276.086 606.687 1059.051    1615.232    2303.858    ...
            3123.036    4084.645
        0   3.814   90.250  353.209 780.214 1367.544    2097.252    2997.967    ...
            4067.796    5318.617
        0   4.824   115.507 454.235 1007.522    1771.647    2728.664    3907.200    ...
            5305.363    6935.031
        0   6.076   146.805 579.428 1289.206    2272.418    3511.119    5033.935    ...
            6838.974    8938.114
        0   7.572   184.211 729.051 1625.859    2870.913    4446.268    6380.549    ...
            8671.866    11332.096
        0   9.316   227.791 903.371 2018.078    3568.190    5535.763    7949.423    ...
            10807.277   14121.204
        0   11.308  277.610 1102.650    2466.455    4365.306    6781.256    9742.933    ...
            13248.443   17309.666];

    if alpha < -6 || alpha > 16
        fprintf('ERROR: alpha = %f (Outside -6 to 16 degrees) is not supported in the Drag ...
            Function - returning nil\n', alpha);
        return;
    elseif V ≥ 400
       fprintf('ERROR: V = %f (Greater than 400) is not supported in the Drag Function - ...
           returning a HUGE value\n', V);
        Dx = -100000;
        Dy = -100000;
        return;
    elseif V == 0
        Dx = 0;
        Dy = 0;
        return;
    else
        col = 1;
        while V > D_VelocityVals(col)
            col = col+1;
        end
        col = col -1;

%         row = 1;
%         while alpha > D_AlphaVals(row)
%             row = row + 1;
%         end
%         row

        d1 = polyfit(D_AlphaVals', DragVals(:,col), 2);
        d2 = polyfit(D_AlphaVals', DragVals(:,col+1), 2);

        drag1 = polyval(d1, alpha);
        drag2 = polyval(d2, alpha);

        V1 = D_VelocityVals(col);
        V2 = D_VelocityVals(col+1);

        drag = drag1 + (V - V1).*((drag2-drag1)./(V2-V1));

        Dy = -drag.*sind(alpha);
        Dx = -drag.*cosd(alpha);

    end
```

```
    return;
end

%% Thrust
%V = Velocity - ft/s

function [Tx, Ty] = Thrust(V, theta)
    epsilon = 2;    %Thrust Deflection Angle
    if V < 1 %|| (1050.*550./V) < 4000
        TProp = 1050.*550;    %Static Thrust
    else
        TProp = 1050.*550./V;
    end
    Tjet = 157;

    Tx = cosd(theta + epsilon).*(TProp.*PropEff(V, 8.75) + Tjet);
    Ty = sind(theta + epsilon).*(TProp.*PropEff(V, 8.75) + Tjet);

    return;
end

function eta = PropEff(V, D)
    J = [0  0.05    0.1 0.15    0.2 0.25    0.3 0.35    0.4 0.45    0.5 0.55    0.6 ...
        0.65    0.7 0.75    0.8 0.85    0.9 0.95    1   1.05    1.1 1.15    1.2 1.25    ...
        1.3 1.35    1.4 1.45    1.5 1.55    1.6 1.65    1.7 1.75    1.8 1.85    1.9 ...
        1.95    2   2.05    2.1 2.15    2.2 2.25    2.3 2.35    2.4 2.45    2.5 2.55    ...
        2.6 2.63];
    eta_pr = [0 0.075   0.2 0.28    0.37    0.45    0.51    0.575   0.63    0.685   ...
        0.72    0.75    0.78    0.79    0.81    0.82    0.83    0.84    0.845   0.85    ...
        0.855   0.8575  0.86    0.865   0.865   0.865   0.865   0.865   0.865   0.865   ...
        0.865   0.865   0.86    0.86    0.86    0.8575  0.855   0.8525  0.85    0.8475  ...
        0.845   0.84    0.83    0.825   0.815   0.8 0.77    0.74    0.7 0.65    0.58    ...
        0.5 0.31    0];

    Prop_eff_curve = fit(J', eta_pr', 'smoothingspline');
    Prop_eff = @(AdvanceRatio)feval(Prop_eff_curve, AdvanceRatio);
    Adv_Rat = @(V, N, D) V./((N./60).*D);
    %PropPDegrees = @(D, P)atand(P./(0.75*D*pi()));

    N = 1700;    %(RPM)

%     figure;
%     hold on
%     plot(Prop_eff_curve)
%     plot(J, eta_pr, 'o');
%     axis([0 .83 0 1])
%     title('Propeller Efficiency');
%     xlabel('J')
%     ylabel('\eta_p_r')
%     legend('hide');
%     hold off;

    eta = Prop_eff(Adv_Rat(V, N, D));

    if eta <=.0038
        eta = .0038;  %prop will never be completely inefficient, This gives a reasonable ...
            value for static thrust at V =0
    end

    return;
end


%% Weight
function W = Weight(time)
    W = -6386;    %(lbs)
    return;
end
```

```
%% Normal Force

%W = Weight in positive y-dir (should be negative number) = lbs
%Ly = Lift in positive y-dir = lbs
%Dy = Drag in positive y-dir
%Ty = Thrust in positive y-dir
function N = Normal(W, Ly, Dy, Ty)
    if ((Ly + Ty) + (W + Dy)) ≥ 0
        N = 0;
    else
        N = -((Ly + Ty) + (W + Dy));    %Equal and opposite force
    end

    return; %(lbs)
end

%% Friction Force

%N = Normal Force = lbs

function f = Friction(N, ForwardForce)
    MUgrd = 0.1;    %rolling friction coefficient for grass runway

    f = -MUgrd.*N;

    if -f > ForwardForce && N ≠ 0
        f = -ForwardForce;
    end
    return; %(lbs)
end

%% %% HELPER FUNCTIONS %% %%
%% Integration

%SumForces = sum of forces along one axis = lbs
%mass = mass of airplane = slugs
%timestep = time between calculations, ∆ time in integration

function [A, V, S] = Integrate(SumForces, mass, oldA, oldV, oldS, timestep)
    A = SumForces./abs(mass);

    V = oldV + 0.5.*(A + oldA).*timestep;%Trapezoid method for integration

    S = oldS + 0.5.*(V + oldV).*timestep;

    return;
end


%% Total - Square Root sum of the squares

%x - quantity in x-dir
%y - quantity in y-dir

function t = Total(x, y)
    t = sqrt(x.^2 + y.^2);
    return;
end
%% Density

%Alt = altitude = ft

function rho = Density(Alt)
if Alt < 0 && Alt > -1e-1
    Alt = 0;    %Stupid Round off error
end
%[Z Z_L Z_U T P rho c g mu nu k n n_sum] = atmo(Alt.*0.0003048,1,2);
[¬, ¬, ¬, ¬, ¬, rho, ¬, ¬, ¬, ¬, ¬, ¬, ¬] = atmo(Alt.*0.0003048,1,2);
%   Input:      alt:        Final Geometric Altitude[km]
%               division:   Reporting points for output arrays[km]
```

```
%                              (.01 km & Divisible by .01 km)
%                units:        1-[Metric]
%                              2-{English}
%   Output:      Each value has a specific region that it is valid in with this model
%                and is only printed out in that region
%                Z:            Total Reporting Altitudes[0≤alt≤1000 km][km]{ft}
%                Z_L:          Lower Atmosphere Reporting Altitudes[0≤alt≤86 km][km]{ft}
%                Z_U:          Upper Atmosphere Reporting Altitudes[86≤alt≤1000 km][km]{ft}
%                T:            Temperature array[0≤alt≤1000 km][K]{R}
%                P:            Pressure array[0≤alt≤1000 km][Pa]{in_Hg}
%                rho:          Density array[0≤alt≤1000 km][kg/m^3]{lb/ft^3}
%                c:            Speed of sound array[0≤alt≤86 km][m/s]{ft/s}
%                g:            Gravity array[0≤alt≤1000 km][m/s^2]{ft/s^2}
%                mu:           Dynamic Viscosity array[0≤alt≤86 km][N*s/m^2]{lb/(ft*s)}
%                nu:           Kinematic Viscosity array[0≤alt≤86 km][m^2/s]{ft^2/s}
%                k:            Coefficient of Thermal Conductivity
%                              array[0≤alt≤86 km][W/(m*K)]{BTU/(ft*s*R)}
%                n:            Number Density of individual gases
%                              (N2 O O2 Ar He H)[86km≤alt≤1000km][1/m^3]{1/ft^3}
%                n_sum:        Number Density of total gases
%                              [86km≤alt≤1000km][1/m^3]{1/ft^3}

    return; %rho in (lb/ft^3)
end


%% AE 370 Takeoff Roll
%Equation 6.94 - page 362



function dist = AE370Takeoff(W, S, CLmax, g, mu, rho, T, D, L, N)
    term1 = (1.21*(W/S))/(rho*CLmax*((T/W)-(D/W)-mu*(1-(L/W))));    %rho is in lbs/ft^3 so ...
        we don't need to multiply g
    term2 = (1.1*N)*sqrt((2/rho./g)*(W/S)*(1/CLmax));
    dist = term1 + term2;
    return;
end

%% AE 370 Approximate Takeoff Roll
%Equation 6.95 - page 362

function dist = AE370ApproxTakeoff(W, S, CLmax, g, rho, T)
    dist = (1.21*(W/S))/(rho.*CLmax*(T/W));
    return;
end
```

B. Landing Code

```
%% HR-UAS Takeoff/Landing Code
%% %% MAIN FUNCTION %% %%
function Landing_Code()
    clear all; close all; clc; format long g;
%%
    %Constants that define how the integration runs
    NumSteps = 300;      %Just a guess for this code but it gets us started
    timestep = 0.01;   %Increment between timesteps (secs)

    %Other Constants Controlling the takeoff roll
    g = 32.174; %ft/s^2
    VStall = 119.1298855; %ft/s^2
    VApproach = 1.1*VStall; %ft/s^2
    DescentAngleOfAttack = 5;     %Degrees
    NoseTouchdownTime = 1.0; %sec
    BrakeApplicationTime = NoseTouchdownTime./2;% sec

```

```
    [ThrusteqnX, ThrusteqnY] = RevThrust(0.7.*VApproach, DescentAngleOfAttack);
    [DrageqnX, DrageqnY] = Drag(DescentAngleOfAttack, 0.7.*VApproach);
    [LifteqnX, LifteqnY] = Lift(0.7.*VApproach, DescentAngleOfAttack, 0);
    %AE370LandingDistance(N, rho, W, S, CLMax, T, D, mu, L)
    AE370Dist = AE370LandingDistance(1,  Density(0), -Weight(1), 247.4, 2.314655933, ...
        Total(ThrusteqnX, ThrusteqnY), Total(DrageqnX, DrageqnY), 0.3, Total(LifteqnX, ...
        LifteqnY))

    %Initialization and book keeping
    %Keep track of all critical integral values
    time = zeros(NumSteps+1,1);%Vector of timestamps events happen
    Ax = zeros(NumSteps+1,1); %Vector of Acceleration in the x-dir (ft/s^2)
    Ay = zeros(NumSteps+1,1);   %Vector of Acceleration in the y-dir (ft/s^2)
    Vx = zeros(NumSteps+1,1);   %Vector of Velocity in the y-dir (ft/s)
    Vy = zeros(NumSteps+1,1);   %Vector of Velocity in the x-dir (ft/s)
    Sx = zeros(NumSteps+1,1);   %Vector of Position from R/W start in the y-dir (ft)
    Sy = zeros(NumSteps+1,1);   %Vector of Position from R/W start in the x-dir (ft)

    %Keep track of all force values (lbs)
    Lx = zeros(NumSteps+1,1); %Lift
    Ly = zeros(NumSteps+1,1);
    Dx = zeros(NumSteps+1,1); %Drag
    Dy = zeros(NumSteps+1,1);
    Tx = zeros(NumSteps+1,1); %Thrust
    Ty = zeros(NumSteps+1,1);
    W = zeros(NumSteps+1,1);  %Weight
    N = zeros(NumSteps+1,1);  %Normal Force on landing gear
    friction = zeros(NumSteps+1,1);
    Fx = zeros(NumSteps+1,1); %Sum of Forces in the x-dir
    Fy = zeros(NumSteps+1,1); %Sum of Forces in the y-dir


    %Keep up with other important parameters
    theta = zeros(NumSteps+1,1); %Angle between Ground and Flight Path (relative wind of ...
        V_inf)


%% Integration
%Initial Values
W(1) = Weight(time(1));
theta(1) = DescentAngleOfAttack;
Vx(1) = VStall;
Vy(1) = 0;
muTouchdown = 0.1;
mu = muTouchdown;
muBraking = 0.3;
RevThrustPercent = 0;
%Integration starts at moment of touchdown
i = 2;
while Vx(i-1) > 0

    time(i) = time(i-1) + timestep;

    if time(i) <= NoseTouchdownTime
        theta(i) = theta(i-1) - DescentAngleOfAttack.*timestep;
    elseif time(i) > NoseTouchdownTime && time(i) <= NoseTouchdownTime+BrakeApplicationTime
        mu = mu + (muBraking-muTouchdown).*timestep.*2;
        theta(i) = 0;
        RevThrustPercent = RevThrustPercent + 1.*timestep.*2;
        [Tx(i), Ty(i)] = RevThrust(Total(Vx(i-1),Vy(i-1)), theta(i));
        Tx(i) = Tx(i).*RevThrustPercent;
        Ty(i) = Ty(i).*RevThrustPercent;
    else
        theta(i) = 0;
        [Tx(i), Ty(i)] = RevThrust(Total(Vx(i-1),Vy(i-1)), theta(i));
    end


    [Lx(i), Ly(i)] = Lift(Total(Vx(i-1),Vy(i-1)), theta(i), Sy(i-1));

```

```
    [Dx(i), Dy(i)] = Drag(theta(i), Total(Vx(i-1),Vy(i-1)));

    W(i) = Weight(time(i));

    N(i) = Normal(W(i), Ly(i), Dy(i), Ty(i));

    friction(i) = Friction(N(i), mu);

    Fy(i) = N(i) + W(i) + Ly(i) + Dy(i) + Ty(i);
    if Fy(i) >0
        Fy(i) = 0;
    end

    Fx(i) = Dx(i) + Lx(i) + friction(i) + Tx(i);

    [Ay(i), Vy(i), Sy(i)] =  Integrate(Fy(i), W(i)./g, Ay(i-1), Vy(i-1), Sy(i-1), timestep);

    [Ax(i), Vx(i), Sx(i)] =  Integrate(Fx(i), W(i)./g, Ax(i-1), Vx(i-1), Sx(i-1), timestep);

    i = i+1;
end
Sx(i-1)

    %% Plots
    %Plot Configurations
    lineFormats = {'m*', 'k-', 'b--', 'r-.', 'g:', 'y.', 'cx'}; %First item is used last ... since matlab doesn't like zero indexing
    plotFormat = @(index)lineFormats{mod(index,length(lineFormats))+1};
    legstr = {'Total Scaler', 'x-Direction', 'y-Direction'};
    %printOpts = '-depsc';    %Color .eps plots
    printOps = '-dpng';

    figure;
    grid on;
    hold on;
    plot(time, Total(Ax,Ay), plotFormat(1));
    plot(time, Ax, plotFormat(2));
    plot(time, Ay, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthEast')
    ylabel('Acceleration (ft/s^2)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingAccel');

    figure;
    grid on;
    hold on;
    plot(time, Total(Vx,Vy), plotFormat(1));
    plot(time, Vx, plotFormat(2));
    plot(time, Vy, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Velocity (ft/s)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingVelocity');

    figure;
    grid on;
    hold on;
    plot(time, Total(Sx,Sy), plotFormat(1));
    plot(time, Sx, plotFormat(2));
    plot(time, Sy, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Position (ft)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingPos');

    figure;
    grid on;
```

```
    hold on;
    plot(time, W);
    hold off;
    ylabel('Weight (lbs)')
    xlabel('Time (sec)')

    figure;
    grid on;
    hold on;
    plot(time, N);
    hold off;
    ylabel('Normal Force (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingNormal');

    figure;
    grid on;
    hold on;
    plot(time, friction);
    hold off;
    ylabel('Friction (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingFriction');

    figure;
    grid on;
    hold on;
    plot(time, Total(Lx,Ly), plotFormat(1));
    plot(time, Lx, plotFormat(2));
    plot(time, Ly, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Lift (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingLift');

    figure;
    grid on;
    hold on;
    plot(time, Total(Dx,Dy), plotFormat(1));
    plot(time, Dx, plotFormat(2));
    plot(time, Dy, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthWest')
    ylabel('Drag (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingDrag');

    figure;
    grid on;
    hold on;
    plot(time, Total(Tx,Ty), plotFormat(1));
    plot(time, Tx, plotFormat(2));
    plot(time, Ty, plotFormat(3));
    hold off;
    legend(legstr, 'Location', 'NorthEast')
    ylabel('Thrust (lbs)')
    xlabel('Time (sec)')
    print(printOps, 'FigLandingThrust');

%     figure;
%     hold on;
%     plot(time, Total(Lx,Ly), plotFormat(1));
%     plot(time, Total(Dx,Dy), plotFormat(2));
%     plot(time, -friction, plotFormat(3));
%     plot(time, -friction+ Total(Dx,Dy), plotFormat(4));
%     hold off;

    %% Table of Values
```

```matlab
    matrix2latex([time, Ax, Ay, Total(Ax, Ay)] , 'tAccel.tex', 'label', ...
        'tab:Acceleration', 'columnLabels', {'Time','Acceleration-x', 'Acceleration-y', ...
        'Total Acceleration'}, 'caption', 'Acceleration', 'format', '%#3.2f','alignment', ...
        'c');
    matrix2latex([time, Vx, Vy, Total(Vx, Vy)] , 'tVel.tex', 'label', 'tab:Velocity', ...
        'columnLabels', {'Time','Velocity-x', 'Velocity-y', 'Total Velocity'}, 'caption', ...
        'Velocity', 'format', '%#3.2f','alignment', 'c');
    matrix2latex([time, Sx, Sy, Total(Sx, Sy)] , 'tPos.tex', 'label', 'tab:Position', ...
        'columnLabels', {'Time','Position-x', 'Position-y', 'Total Distance'}, 'caption', ...
        'Position', 'format', '%#3.2f','alignment', 'c');
    matrix2latex([time, theta] , 'tTheta.tex', 'label', 'tab:Theta', 'columnLabels', ...
        {'Time','Theta'}, 'caption', 'Theta', 'format', '%#3.2f','alignment', 'c');
    matrix2latex([time, Lx, Ly, Total(Lx, Ly), Dx, Dy, Total(Dx, Dy)] , 'tFAero.tex', ...
        'label', 'tab:FAero', 'columnLabels', {'Time','Lift-x', 'Lift-y', 'Total Lift', ...
        'Drag-x', 'Drag-y', 'Total Drag'}, 'caption', 'Aerodynamic Forces', 'format', ...
        '%#3.2f','alignment', 'c');
    matrix2latex([time, W, N, friction, Tx, Ty, Total(Tx, Ty)] , 'tForces.tex', 'label', ...
        'tab:FAero', 'columnLabels', {'Time','Weight', 'Normal', 'Friction', 'Thrust-x', ...
        'Thrust-y', 'Total Thurst'}, 'caption', 'Other Forces', 'format', ...
        '%#3.2f','alignment', 'c');
    matrix2latex([Sx(i-1), time(i-1)] , 'tLanding.tex', 'label', 'tab:Acceleration', ...
        'columnLabels', {'Landing Distance (ft)','Time (sec)'}, 'caption', 'Landing ...
        Distance', 'format', '%#3.2f','alignment', 'c');


end



%% %% FORCE CALCULATIONS %% %%
% All functions should return forces signed correctly for their
% positive/negative direction
%% Lift

% V = Velocity - ft/s
% Theta = Flight Path Angle = Degrees
% altitude = altitude = ft

function [Lx, Ly] = Lift(V, Theta, altitude)
    WingOffsetAngle = -5;
    AlphaL0 = -4;

    %Calculations for ThetaPrime were done with Flaps at 15 degrees
    ThetaPrime = 1.4;
    %CL = 0.0873.*(Theta+WingOffsetAngle-AlphaL0) + .3493;
    CL = 0.0802.*(Theta + WingOffsetAngle-AlphaL0) + .3306;
    CLF = 0.0802.*(Theta +ThetaPrime+ WingOffsetAngle-AlphaL0) + .3306;
    S = 247.4;          %(ft^2)
    SFlapped = 92.9425;
    SFlaps = 2.*4.631;
    g = 32.174; %ft/s^2

    L = 0.5.*Density(altitude)./g.*V.^2.*(S-SFlapped).*CL + ...
        0.5.*Density(altitude)./g.*V.^2.*(SFlapped + SFlaps).*CLF;

    Lx = -L.*sind(Theta+WingOffsetAngle);
    Ly = L.*cosd(Theta+WingOffsetAngle);
    return;
end

%% Drag

%alpha = Flight Path Angle of Plane = Degrees
%V = Velocity - ft/s

function [Dx, Dy] = Drag(alpha, V)
    %Data from Hooker's Drag Calcs, for 30 degrees of flaps
    D_AlphaVals = [-6 -4    -2  0   2   4   6   8   10  12  14  16];
    D_VelocityVals = [0 10  50  100 150 200 250 300 350 400];
```

```
    DragVals = [0   3.575198098 84.27851494 329.3217572 726.4676724 1271.9947   ...
        1947.957302 2782.982349 3775.177534 4936.421285
                0   3.174202559 74.25362644 289.2222032 636.2436759 1111.596484 ...
                    1697.33509  2422.086363 3283.957998 4294.828421
                0   3.00959053  70.13832573 272.7610004 599.2059695 1045.751673 ...
                    1594.452572 2273.935538 3082.308263 4031.449176
                0   3.078719432 71.86654829 279.6738906 614.7599726 1073.403234 ...
                    1637.658136 2336.15155  3166.991168 4142.05542
                0   3.379827546 79.39425112 309.7847019 682.5092981 1193.846479 ...
                    1825.850707 2607.148852 3535.848607 4623.828401
                0   3.91467659  92.76547723 363.2696064 802.850333  1407.786097 ...
                    2160.13136  3088.512992 4191.038686 5479.586872
                0   4.685909146 112.0462911 440.3928619 976.377658  1716.279119 ...
                    2642.151707 3782.622292 5135.798567 6713.558961
                0   5.696167793 137.3027573 541.4187267 1203.685854 2120.382578 ...
                    3273.563362 4691.855074 6373.36541  8329.972796
                0   6.948095112 168.6009403 666.6114586 1485.369501 2621.153506 ...
                    4056.017936 5818.589662 7906.976376 10333.05651
                0   8.444333684 206.0069046 816.2353158 1822.023179 3219.648935 ...
                    4991.167044 7165.204376 9739.868626 12727.03822
                0   10.18752609 249.5867147 990.5545562 2214.24147  3916.925896 ...
                    6080.662296 8734.07754  11875.27932 15516.14607
                0   12.18031491 299.4064351 1189.833438 2662.618954 4714.041423 ...
                    7326.155307 10527.58748 14316.44562 18704.60818];

    if alpha < -6 || alpha > 16
        fprintf('ERROR: alpha = %f (Outside -6 to 16 degrees) is not supported in the Drag ...
            Function - returning nil\n', alpha);
        return;
    elseif V ≥ 400
       fprintf('ERROR: V = %f (Greater than 400) is not supported in the Drag Function - ...
           returning a HUGE value\n', V);
        Dx = -100000;
        Dy = -100000;
        return;
    elseif V == 0
        Dx = 0;
        Dy = 0;
        return;
    else
        col = 1;
        while V > D_VelocityVals(col)
            col = col+1;
        end
        col = col -1;

%         row = 1;
%         while alpha > D_AlphaVals(row)
%             row = row + 1;
%         end
%         row

        d1 = polyfit(D_AlphaVals', DragVals(:,col), 2);
        d2 = polyfit(D_AlphaVals', DragVals(:,col+1), 2);

        drag1 = polyval(d1, alpha);
        drag2 = polyval(d2, alpha);

        V1 = D_VelocityVals(col);
        V2 = D_VelocityVals(col+1);

        drag = drag1 + (V - V1).*((drag2-drag1)./(V2-V1));

        Dy = -drag.*sind(alpha);
        Dx = -drag.*cosd(alpha);

    end

    return;
end
```

```
%% Thrust
%V = Velocity - ft/s

function [Tx, Ty] = RevThrust(V, theta)
    epsilon = 2;    %Thrust Deflection Angle
    if V < 1 %|| (1050.*550./V) < 4000
        TProp = 1050.*550;    %Static Thrust
    else
        TProp = 1050.*550./V;
    end
    Tjet = 157;

    BetaEfficiency = 0.6;   %60 Percent Reversable Thrust Available

    Tx = -BetaEfficiency.*cosd(theta + epsilon).*(TProp.*PropEff(V, 8.75) + Tjet);
    Ty = -BetaEfficiency.*sind(theta + epsilon).*(TProp.*PropEff(V, 8.75) + Tjet);

    return;
end

function eta = PropEff(V, D)
    J = [0  0.05    0.1 0.15    0.2 0.25    0.3 0.35    0.4 0.45    0.5 0.55    0.6 ...
        0.65    0.7 0.75    0.8 0.85    0.9 0.95    1   1.05    1.1 1.15    1.2 1.25    ...
        1.3 1.35    1.4 1.45    1.5 1.55    1.6 1.65    1.7 1.75    1.8 1.85    1.9 ...
        1.95    2   2.05    2.1 2.15    2.2 2.25    2.3 2.35    2.4 2.45    2.5 2.55    ...
        2.6 2.63];
    eta_pr = [0 0.075   0.2 0.28    0.37    0.45    0.51    0.575   0.63    0.685   ...
        0.72    0.75    0.78    0.79    0.81    0.82    0.83    0.84    0.845   0.85    ...
        0.855   0.8575  0.86    0.865   0.865   0.865   0.865   0.865   0.865   0.865   ...
        0.865   0.865   0.86    0.86    0.86    0.8575  0.855   0.8525  0.85    0.8475  ...
        0.845   0.84    0.83    0.825   0.815   0.8 0.77    0.74    0.7 0.65    0.58    ...
        0.5 0.31    0];

    Prop_eff_curve = fit(J', eta_pr', 'smoothingspline');
    Prop_eff = @(AdvanceRatio)feval(Prop_eff_curve, AdvanceRatio);
    Adv_Rat = @(V, N, D) V./((N./60).*D);
    %PropPDegrees = @(D, P)atand(P./(0.75*D*pi()));

    N = 1700;    %(RPM)

%     figure;
%     hold on
%     plot(Prop_eff_curve)
%     plot(J, eta_pr, 'o');
%     axis([0 .83 0 1])
%     title('Propeller Efficiency');
%     xlabel('J')
%     ylabel('\eta_p_r')
%     legend('hide');
%     hold off;

    eta = Prop_eff(Adv_Rat(V, N, D));

    if eta ≤.0038
        eta = .0038;  %prop will never be completely inefficient, This gives a reasonable ...
            value for static thrust at V =0
    end

    return;
end


%% Weight
function W = Weight(time)
    W = -6386;   %(lbs)
    return;
end

%% Normal Force
```

```
%W = Weight in positive y-dir (should be negative number) = lbs
%Ly = Lift in positive y-dir = lbs
%Dy = Drag in positive y-dir
%Ty = Thrust in positive y-dir
function N = Normal(W, Ly, Dy, Ty)
    if ((Ly + Ty) + (W + Dy)) ≥ 0
        N = 0;
    else
        N = -((Ly + Ty) + (W + Dy));    %Equal and opposite force
    end

    return; %(lbs)
end

%% Friction Force

%N = Normal Force = lbs

function f = Friction(N, Mu)
   %rolling friction coefficient for grass runway

    f = -Mu.*N;

    return; %(lbs)
end

%% %% HELPER FUNCTIONS %% %%
%% Integration

%SumForces = sum of forces along one axis = lbs
%mass = mass of airplane = slugs
%timestep = time between calculations, ∆ time in integration

function [A, V, S] = Integrate(SumForces, mass, oldA, oldV, oldS, timestep)
    A = SumForces./abs(mass);

    V = oldV + 0.5.*(A + oldA).*timestep;%Trapezoid method for integration

    S = oldS + 0.5.*(V + oldV).*timestep;

    return;
end


%% Total - Square Root sum of the squares

%x - quantity in x-dir
%y - quantity in y-dir

function t = Total(x, y)
    t = sqrt(x.^2 + y.^2);
    return;
end
%% Density

%Alt = altitude = ft

function rho = Density(Alt)
if Alt < 0 && Alt > -1e-1
    Alt = 0;    %Stupid Round off error
end
%[Z Z_L Z_U T P rho c g mu nu k n n_sum] = atmo(Alt.*0.0003048,1,2);
[¬, ¬, ¬, ¬, ¬, rho, ¬, ¬, ¬, ¬, ¬, ¬, ¬] = atmo(Alt.*0.0003048,1,2);
%   Input:      alt:        Final Geometric Altitude[km]
%               division:   Reporting points for output arrays[km]
%                           (.01 km & Divisible by .01 km)
%               units:      1-[Metric]
%                           2-{English}
%   Output:     Each value has a specific region that it is valid in with this model
```

```
%                 and is only printed out in that region
%                 Z:          Total Reporting Altitudes[0≤alt≤1000 km][km]{ft}
%                 Z_L:        Lower Atmosphere Reporting Altitudes[0≤alt≤86 km][km]{ft}
%                 Z_U:        Upper Atmosphere Reporting Altitudes[86≤alt≤1000 km][km]{ft}
%                 T:          Temperature array[0≤alt≤1000 km][K]{R}
%                 P:          Pressure array[0≤alt≤1000 km][Pa]{in_Hg}
%                 rho:        Density array[0≤alt≤1000 km][kg/m^3]{lb/ft^3}
%                 c:          Speed of sound array[0≤alt≤86 km][m/s]{ft/s}
%                 g:          Gravity array[0≤alt≤1000 km][m/s^2]{ft/s^2}
%                 mu:         Dynamic Viscosity array[0≤alt≤86 km][N*s/m^2]{lb/(ft*s)}
%                 nu:         Kinematic Viscosity array[0≤alt≤86 km][m^2/s]{ft^2/s}
%                 k:          Coefficient of Thermal Conductivity
%                             array[0≤alt≤86 km][W/(m*K)]{BTU/(ft*s*R)}
%                 n:          Number Density of individual gases
%                             (N2 O O2 Ar He H)[86km≤alt≤1000km][1/m^3]{1/ft^3}
%                 n_sum:      Number Density of total gases
%                             [86km≤alt≤1000km][1/m^3]{1/ft^3}

    return; %rho in (lb/ft^3)
end

function dist = AE370LandingDistance(N, rho, W, S, CLMax, T, D, mu, L)
    term1 = 1.1.*N.*sqrt((2./rho).*(W./S).*(1./CLMax));
    term2 = (1.1.^2.*(W./S))./(rho*CLMax*((T/W)+(D/W)+mu*(1-(L/W))));
    dist = term1 + term2;
    return;
end
```

C. VLM Code

```
clear all; close all; clc;
%% Vortex Lattice Method Applied to UAV Wing

N=100;
%% Geometry

b=40; % feet
lambda=4.37./8;
sweep_quarter=0;
Cr=8;
Ct = Cr.*lambda;
AR=2.*b./(Cr + Ct);
sweep_threequarters=-atand((.75.*Cr - .75.*Ct)/b);
Cr = Cr./b;

%% Locations of Interest(Starboard)
A(1,1)=1;
A(1,2)=(3/4)*Cr+(.5/2/N)*tand(sweep_threequarters);
A(1,3)=.5/2/N;
A(1,4)=1/4*Cr;
A(1,5)=0;
A(1,6)=Cr/4+.5/N*tand(sweep_quarter);
A(1,7)=.5/N;
for i=2:N
    A(i,1)=i;
    A(i,2)=A(i-1,2)+.5/N*tand(sweep_threequarters);
    A(i,3)=A(i-1,3)+.5/N;
    A(i,4)=A(i-1,4)+.5/N*tand(sweep_quarter);
    A(i,5)=A(i-1,5)+.5/N;
    A(i,6)=A(i-1,6)+.5/N*tand(sweep_quarter);
    A(i,7)=A(i-1,7)+.5/N;
end

% Locations of Interest(Port)
Ap=A; Ap(:,3)=-A(:,3); Ap(:,5)=-A(:,5); Ap(:,7)=-A(:,7);

```

```
%% Downwash Velocity

% Starboard
for j=1:N
for i=1:N
    W1s=1/[((A(i,2)-A(j,4))*(A(i,3)-A(j,7))-(A(i,2)-A(j,6))*(A(i,3)-A(j,5)))];
    W2s=[((A(j,6)-A(j,4))*(A(i,2)-A(j,4)))+((A(j,7)-A(j,5))*(A(i,3)-A(j,5)))]/[sqrt((A(i,2)-A(j,4))^2+(A(i,3)-
    W3s=[((A(j,6)-A(j,4))*(A(i,2)-A(j,6)))+((A(j,7)-A(j,5))*(A(i,3)-A(j,7)))]/[sqrt((A(i,2)-A(j,6))^2+(A(i,3)-
    W4s=[1/(A(j,5)-A(i,3))]*[1+(A(i,2)-A(j,4))/sqrt((A(i,2)-A(j,4))^2+(A(i,3)-A(j,5))^2)];
    W5s=[1/(A(j,7)-A(i,3))]*[1+(A(i,2)-A(j,6))/sqrt((A(i,2)-A(j,6))^2+(A(i,3)-A(j,7))^2)];
    ws(i,j)=[W1s*(W2s-W3s)+W4s-W5s];
end
end

% Port
for j=1:N
for i=1:N
    W1p=1/[((A(i,2)-Ap(j,4))*(A(i,3)-Ap(j,7))-(A(i,2)-Ap(j,6))*(A(i,3)-Ap(j,5)))];
    W2p=[((Ap(j,6)-Ap(j,4))*(A(i,2)-Ap(j,4)))+((Ap(j,7)-Ap(j,5))*(A(i,3)-Ap(j,5)))]/[sqrt((A(i,2)-Ap(j,4))^2+
    W3p=[((Ap(j,6)-Ap(j,4))*(A(i,2)-Ap(j,6)))+((Ap(j,7)-Ap(j,5))*(A(i,3)-Ap(j,7)))]/[sqrt((A(i,2)-Ap(j,6))^2+
    W4p=[1/(Ap(j,5)-A(i,3))]*[1+(A(i,2)-Ap(j,4))/sqrt((A(i,2)-Ap(j,4))^2+(A(i,3)-Ap(j,5))^2)];
    W5p=[1/(Ap(j,7)-A(i,3))]*[1+(A(i,2)-Ap(j,6))/sqrt((A(i,2)-Ap(j,6))^2+(A(i,3)-Ap(j,7))^2)];
    wp(i,j)=[W1p*(W2p-W3p)+W4p-W5p];
end
end

% Total
w=ws-wp;  % code is correct but book says (+)

%% Solving for Vortex Strength

for i=1:N
    B(i,1)=-4*pi;
end
V_S=w^-1*B; % times b Uinf alpha

%% Lift Distribution
% POSITIVE LOAD FACTOR

%alpha=15*pi/180; % chosen to give correct load factor n
rho=.00237; % slug/ft^3 @ 8000'
%Uinf=253.17; % ft/s (150kts)
alpha = 19.74.*pi./180;
Uinf = [0:1:350];
Load_Factor = 0;
j = 1;
while j ≤ length(Uinf);
    ∆_y=.5/N;
    LD(1,1)=0+.5*∆_y;
    for i=2:N
        LD(i,1)=LD(i-1,1)+∆_y;
    end
    for i=1:N
        LD(i,2) = V_S(i);
    end
    LD(:,1)=b.*LD(:,1);
    LD(:,2)=alpha.*b.*rho.*Uinf(j).^2.*LD(:,2);
    Load_Factor(j)=2.*trapz(LD(:,1),LD(:,2))./6400;
    j = j + 1;
end
LoadFactor_253 = Load_Factor(253)
x = find(Load_Factor > 3);
xx = find(Load_Factor > 3.*1.5);

figure
grid on
hold on
plot(Uinf,Load_Factor,'k','LineWidth',2)
plot(Uinf(x(1)):Uinf(end), 3,'y.')
plot(Uinf(xx(1)):Uinf(end),3.*1.5,'r.')
```

```
% plot n max, vstar
plot(Uinf(x(1)),0:0.2:3,'k.')
plot(332.3,0:0.2:Load_Factor(253),'k.')
plot(96.8,0:0.2:Load_Factor(97),'k.')
xlabel('V_\infty (ft/s)')
ylabel('Load Factor, n')
text(125,3.5,sprintf('V_* = %3.0f ft/s',Uinf(x(1))),'BackgroundColor',[1 1 ...
    1],'EdgeColor',[0 0 0])
text(25,6,'Stall Area','BackgroundColor',[1 1 1],'EdgeColor',[0 0 0],'FontSize',18)
text(270,3.5,'Structural Damage','FontSize',8,'BackgroundColor',[1 1 1])
text(280,5,'Structural Failure','FontSize',8,'BackgroundColor',[1 1 1])
text(240,0.5,'V_c_r_u_i_s_e = 253 ft/s','BackgroundColor',[1 1 1],'EdgeColor',[0 0 0])
text(25,1.5,'V_s_t_a_l_l = 96.8 ft/s','BackgroundColor',[1 1 1],'EdgeColor',[0 0 0])
print('-f1','-dpng','V_nDiagram')

%% Calculate Lift

Δ_y=.5/N;     % times b
L=2*sum(V_S)*Δ_y;      % times rho, Uinf^2, b^2, alpha

C_L=L*4*6.875/1.375/pi    % times pi, alpha

%% Weight Distribution

y=A(:,3);
W(:,1)=y*b;
W(:,2)=35.989-1.6369.*W(:,1)+.01859.*W(:,1).^2;

%% Shear and Moment

% Shear
Load=LD(:,2)-W(:,2);
Shear(:,1)=W(:,1);
Shear(:,2)=wrev(cumtrapz(Shear(:,1),Load));
Max_Shear=max(Shear(:,2)) % lbf

% Moment
Moment(:,1)=W(:,1);
Moment(:,2)=wrev(cumtrapz(Moment(:,1),Shear(:,2)));
Max_Moment=max(Moment(:,2))


I=1.25*Max_Moment*.12*5.88/58000/2
Ispar = 0.6.*I
```

D. Avionics

5/3/12 Cloud Cap Technology -- Piccolo System Configuration Tool - Print Summary

Print this Page

Cloud Cap TECHNOLOGY

Unmanned Systems
Autopilots Payloads Sensors

«« Back to Summary

Autopilot	Part No.
Piccolo II	
Radio Option: 310-390 MHz Discrete	900-90010-02
Advanced Feature Option: Moving Baseline Capture Software	900-01581-00

* Ground Station	Part No.
Portable Ground Station Kit *with integrated Novatel DGPS	900-90015-42

User Interface	Part No.
Advanced PCC	900-01434-00

* Developers Kit	Part No.
Piccolo Developer's Kit - 310-390/405-425 MHz	900-90003-02

Accessories

Communication Antennas, Ground Planes and Coaxial Cables	Part No.
Antenna Cable, Piccolo II Antenna SMA M to BNC F, 45-inch	500-00312-45

GPS Antennas and Ground Plane	Part No.
Antenna, Ground Plane for Aircraft GPS	620-00562-00

Power	Part No.
Battery Pack, 12V 2700ma 10-Acell NiMH - Piccolo	790-00291-00

Piccolo Flight Harness	Part No.
Cable, Piccolo II, Typical Piccolo Flight Harness	500-01045-00

Piccolo Mounting	Part No.
Mounting Rails, Carbon Fiber Only (need two)	500-00491-00

Deadman Tach	Part No.
Board, Deadman/Tach Engine Interface - Magneto	900-00591-00

Air Data System Kits (tube, reducer fittings, mounting hardware)	Part No.
Air Data Kit, Carbon Fiber, Combined Pitot/Static Tube with 2 port hub	800-00593-00

UAV Transponder	Part No.
Transponder, MicroAir T2000 UAV-S with BNC	500-01231-00

Novatel - RTK GPS Integration Kits	Part No.
Integration Kit, Aircraft, Includes Novatel DGPS, antenna and cables	800-01299-00

www.cloudcaptech.com/piccolo/configure_print.asp?radio=310_390&interface=commandcenter&auto... 1/2

5/3/12 Cloud Cap Technology – Piccolo System Configuration Tool - Print Summary

* The Ground Station Kit and Developers Kit will be configured with a 310-390 MHz Discrete Radio Frequency.

Phone/Fax:
+1.541.387.2120 phone
+1.541.387.2030 fax

Address:
Cloud Cap Technology Inc.
2621 Wasco St. PO Box 1500
Hood River, OR 97031

Exit the Piccolo Configuration Worksheet