5-2013

Design of Economical Upper Stage Hybrid Rocket Engine

Christopher R. Potter
University of Tennessee - Knoxville, cpotter5@utk.edu

AE429 Design Project
Upper Stage Economical Hybrid Rocket Motor Design

Brian Hampton, Stephanie Pollock, Chris Potter, Rachel Preston, and Ben Whitehead
University of Tennessee-Knoxville, MABE

This project was initiated to design the third stage of a hybrid rocket to launch nanosatellites into orbit. A hybrid rocket was chosen due to its throttling ability and the harmlessness of its byproducts when using certain fuels. To begin the design process, an analysis was conducted using MATLAB and CEA to determine the theoretical thrust output from a variety of rocket configurations. An experiment was designed to use air as the oxidizer in a head-end injection system with a fuel of ABS plastic. A water-cooled nozzle was included to analyze the heatflux experienced by the nozzle.

Nomenclature

A	=	area
A^*	=	nozzle throat area
A_e	=	nozzle exit area
a	=	sonic velocity
C^*	=	characteristic velocity
C_T	=	thrust coefficient
c_p	=	specific heat
D_H	=	hydraulic diameter
G_{ox}	=	mass flux
g_e	=	earth's gravitational acceleration
h	=	height of fuel grain, heat transfer coefficient
I_{SP}	=	specific impulse
k	=	thermal conductivity
$\dot{m}_{fuel}$	=	fuel mass flow rate
$\dot{m}_{oxy}$	=	oxidizer mass flow rate
Nu	=	Nusselt number
n	=	heating or cooling coefficient
$\dot{r}$	=	regression rate
r_1	=	inner grain radius before one second of burning
r_2	=	inner grain radius after one second of burning
P_0	=	total pressure
P_a	=	ambient pressure
P_e	=	exit pressure
Pr	=	Prandtl number
Re	=	Reynolds number
u_{eq}	=	equivalent exhaust velocity
V	=	velocity
γ	=	specific heat ratio
μ	=	dynamic viscosity
ρ	=	density

I. Introduction

HYBRID rockets have traditionally been used as a hybrid between solid fuel rockets and liquid fuel rockets. Since, as the name suggests, solid fuel rockets use a completely solid grain with the oxidizer included in the fuel, they burn quickly and in an uncontrolled fashion. Once the fuel is lit solid rockets burn until there is no more fuel left. This makes throttling any solid fuel rocket impossible. On the other hand, liquid fuel rockets are easily

throttled, with a controllable level of thrust dependent on controlling the flow of liquid propellent entering the combustion chamber. With higher specific impulses than solid and hybrid rockets, liquid rockets initially seem to be the best option to propel a spacecraft into orbit. But the large amount of complex plumbing required to maintain a liquid rocket is a disadvantage. This plumbing can include a series of cryogenic tanks to keep the propellant in a liquid state, and many pipes and valves to route the fuel to the combustion chamber. Hybrid rockets are an efficient combination of both solid and liquid fuel rockets. They use a solid fuel grain that reacts with a gaseous or liquid oxidizer to generate thrust. The oxidizer is provided from a reservoir that requires much less plumbing than traditional liquid rockets. The burn rate of the grain can also be controlled or completely stopped by regulating the amount of oxidizer entering the combustion chamber. This gives a rocket that combines aspects of both solid and liquid fuel rockets.

Hybrid rockets were briefly studied in the 1930's when groups of German, Soviet, and American scientists conducted independent experiments with hybrid rockets using oxygen and various fuel grains. The topic was largely left untouched until the 1980's and 1990's when some American universities and aerospace companies developed and tested some midsize hybrid sounding rockets using LOX and HTPB as fuel. These experiments culminated in Scaled Composites and SpaceDev developing SpaceShipOne, a sub-orbital manned vehicle using a Nitrous Oxide/HTPB hybrid motor. SpaceShipOne is one of the most successful hybrid rockets to date, joining a select group of hybrid sounding rockets in its success. [1]

Hybrid rockets have many advantages over traditional solid and liquid fueled rockets, namely that they are much safer. Solid rockets are a high explosion hazard since both of the ingredients for combustion are included in the fuel grain. The slightest spark or source of ignition can ignite the whole assembly with catastrophic results virtually guaranteed. The exhaust of a hybrid rocket can also be relatively harmless. Non-toxic oxidizers like LOX and nitrous oxide can be used to reduce the harmful nature of rocket byproducts. Another benefit of hybrid rockets is that they have a higher theoretical specific impulse than solid rockets and have the added benefit of controlling the level of thrust output. This specific impulse can then be increased further by adding metal additives to the fuel. These additives tend to be reactive metals, such as aluminum, that when incorporated in the fuel grain increase the specific impulse of the rocket. When compared to liquid fuel rockts, hybrid rockets are much simpler. They only have a single gaseous or liquid propellent so the amount of plumbing required is much less. Using a solid fuel grain also reduces the volume of fuel in a hybrid rocket since solid fuel is more dense than liquids or gases. Overall, hybrid rockets represent a common ground utilizing the best facets of solid and liquid fuel rockets. [2]

Despite their advantages, hybrid rockets still come with their own unique disadvantages. The oxidizer to fuel ratio changes as the fuel grain regresses. This makes it difficult to regulate the thrust output of the rocket. Hybrid rockets typically have lower regression rates than solid fuel rockets. This can result in lower specific impulses in certain instances. To combat this a few tricks have been proposed and documented to increase the regression rate. One is the use of vortex flow in the combustion chamber. Using aft-end injection, the oxidizer is injected tangential to the fuel grain. This causes the oxidizer to spiral up the inside of the grain in a vortex. As the oxidizer reaches the top of the grain it turns and plunges down the center of the combustion chamber, reacting and combusting with the fuel as it leaves the combustion chamber. Using vortex flow, regression rates can be increased by as much as 6 times than when it is not used. [3] Another way to increase the regression rate of hybrid rockets is to use different grain geometries. Using a grain geometry with more than one hole can increase the combustible surface area in the hybrid engine, thereby increasing the regression rate of the rocket. [2]

II. Theoretical Analysis

A. Theoretical Thrust Analysis

Before conducting experiments with the thrust stand, an initial theoretical performance analysis must be conducted. A Matlab program was written utilizing the known information of the target combustion pressure, the oxidizer mass flow rate, the throat area, the nozzle area ratio, and the grain dimensions. With these values, the goal of the thrust program was to obtain data on the thrust, ISP, mass flow rates, regression rates, and O/F ratios of the rocket. Using these theoretical values, the expected performance of the experimental rocket can be predicted. To begin the theoretical calculation, Gox, or the mass flux out of the engine was determined.

$$G_{ox} = \frac{\dot{m}_{oxy}}{Area} \tag{1}$$

Since the mass flow rate of the oxidizer is one of the assumed values in this problem and the dimensions of the fuel grain are known, then the mass flux is easily calculated using Eq (1). With this value a preliminary estimate of the regression rate is easily calculated.

$$\dot{r} = a(G_{ox})^n \tag{2}$$

The regression rate determined in Eq (2) is the distance that the grain burns outward in a specific amount of time. This is a function of the mass flux and will change as the mass flux changes. Since the mass flux is a function of the cross-sectional area of the fuel grain, which changes as the fuel burns outwards, the regression rate also changes with the changing cross-sectional area. The values of a=0.2 and n=0.54 are experimental values obtained from previous experiments conducted in this area. [4] Now since the regression rate and the dimensions of the fuel grain are known, the mass flow rate of the fuel can be calculated. When paired with the mass flow rate of the oxidizer the oxidizer to fuel ratio is calculated. This is a valuable input that CEA needs in order to analyze the combustion characteristics in the chamber. Eq's (3) and (4) show these calculations.

$$\dot{m}_{fuel} = ({r_2}^2 - {r_1}^2)\pi h \rho_{fuel} \tag{3}$$

$$\frac{O}{F} Ratio = \frac{\dot{m}_{oxy}}{\dot{m}_{fuel}} \tag{4}$$

In addition to the O/F Ratio, the area ratio of the nozzle is known from confirmed nozzle geometry. These two values, when combined with the total combustion pressure (P_0), make up the inputs received by CEA. Once the oxidizer and fuel materials are specified, CEA calculates the combustion characteristics taking place in the combustion chamber. This yields values for the characteristic velocity (C*), the exit pressure, and specific heat ratio.[4] Knowledge of all of these values allows the thrust coefficient to be calculated.

$$C_\tau = \sqrt{\frac{2\gamma^2}{\gamma-1}\left(\frac{2}{\gamma+1}\right)^{\frac{(\gamma+1)}{(\gamma-1)}}\left[1 - \frac{P_e}{P_0}^{\frac{(\gamma-1)}{\gamma}}\right]} + \frac{P_e - P_a}{P_0}\frac{A_e}{A^*} \tag{5}$$

Now since the total mass flow rate, characteristic velocity, and thrust coefficient are known, the theoretical thrust of the rocket can be calculated.

$$T = \dot{m} C^* C_\tau \tag{6}$$

The I_{SP} is calculated using the definition of specific impulse as defined in The Mechanics and Thermodynamics of Propulsion. [4] Since the thrust and total mass flow rate are known, the equivalent exhaust velocity can be calculated. When used in conjunction with gravity for earth, the specific impulse is easily calculated.

$$u_{eq} = \frac{T}{\dot{m}} \tag{7}$$

$$I_{SP} = \frac{u_{eq}}{g_e} \tag{8}$$

The output of this program gives data on the specific impulse, thrust, mass flow rates, oxidizer to fuel ratios, and regression rates for a rocket. Knowledge of these values allows a model for the performance of a rocket to be built. Trajectory analysis can then be conducted with knowledge of the thrust of the rocket.

B. CEA Automation Through Matlab

A program developed by NASA, Chemical Equilibrium with Applications (CEA), was needed to perform the combustion analyses detailed in Section A, above. CEA is an executable that reads a specially formatted text file of input parameters and syntax that defines the analysis to be performed and then creates a formatted output file of results. Input and output file syntax is specified by NASA Reference Publication 1311. In addition to creating or

modifying the input file before each run of CEA, CEA requires the user to enter the filename of the input file each time as well. Note that, there is a version of CEA available with a graphical user interface (GUI).

Matlab was the primary programming language used by this group because it is available through the University of Tennessee and has a code library which is useful for engineers. For the purposes of our analyses, it was necessary to make hundreds of runs of CEA. Two Matlab codes were written to interact with CEA. One that automatically writes an input file based on a short list of parameter values sent through a function call and another that reads output files and returns relevant values. Matlab is capable of running external executable files. So, CEA did not need to be manually started for each run. However, CEA still required the user to type in the input file name for every run. Therefore, using Matlab to automatically write an input file, call CEA, and then read the output file after CEA closes (always using the same input and output filenames) still required the user either type in or paste in the name of the input file for each run. Fortunately, NASA made the original code, written in Fortran (The International Business Machines (IBM) Mathematical FORmula TRANslating System), available over the internet and there were free Fortran compilers available for Linux based operating systems and there were free Linux emulators available for Windows operating systems. Also, luckily, it is possible to open Fortran files as text files for modification. This all made it possible to: modify the original Fortran code in notepad, recompile it as an executable in a Linux based Fortran compiler, running in a Linux emulator, running in Windows.

Two modifications were made to the original 6000 line Fortran code. First the line prompting the user for the name of the input file was commented out. Second, the variable which this prompt would have filled with user input text was defined explicitly in code. Please see Figure 1, below. With these modifications, it was possible to write other Matlab codes which could recursively perform CEA analyses with no user input required and at a rate of approximately one run per second.

```
1       IMPLICIT NONE
2       INCLUDE 'cea.inc'
3     C LOCAL VARIABLES
4       CHARACTER*15 ensert(20)
5       CHARACTER*200 infile,ofile
6       CHARACTER*196 prefix
7       LOGICAL caseok,ex,readok
8       INTEGER i,inc,iof,j,ln,n
9       INTEGER INDEX
10      REAL*8 xi,xln
11      REAL*8 DLOG
12      SAVE caseok,ensert,ex,i,inc,infile,iof,j,ln, ↙
      n,ofile,prefix,readok,
13      &  xi,xln
14    C
15      WRITE (*,99001)
16      READ (*,99002) prefix
17      ln = INDEX(prefix,' ') - 1
18      infile = prefix(1:ln)//'.inp'
19      ofile = prefix(1:ln)//'.out'
20      Pfile = prefix(1:ln)//'.plt'
21      INQUIRE (FILE=infile,EXIST=ex)
```

```
1       IMPLICIT NONE
2       INCLUDE 'cea.inc'
3     C LOCAL VARIABLES
4       CHARACTER*15 ensert(20)
5       CHARACTER*200 infile,ofile
6       CHARACTER*196 prefix
7       LOGICAL caseok,ex,readok
8       INTEGER i,inc,iof,j,ln,n
9       INTEGER INDEX
10      REAL*8 xi,xln
11      REAL*8 DLOG
12      SAVE caseok,ensert,ex,i,inc,infile,iof,j,ln, ↙
      n,ofile,prefix,readok,
13      &  xi,xln
14    C
15    C     WRITE (*,99001)
16    C     READ (*,99002) prefix
17    C
18    C  I added this line and commented the 2 ↙
      above so that execution happens without ↙
      querying the user for an input filename
19    C
20      prefix = 'HybRoc'
21    C
22    C
23      ln = INDEX(prefix,' ') - 1
24      infile = prefix(1:ln)//'.inp'
25      ofile = prefix(1:ln)//'.out'
26      Pfile = prefix(1:ln)//'.plt'
27      INQUIRE (FILE=infile,EXIST=ex)
```

Figure 1: CEA Fortran Code Before and After Modification

C. Trajectory Analysis

When designing a rocket it is important to have a system available to be able to approximate the trajectory performance of the vehicle. Stephen Phillips, a University of Tennessee alumnus, created such a program using Matlab. This code consisted of a function file to plot the altitude, relative velocity, flight path angle and relative velocity azimuth throughout the flight of the vehicle. For this project, Stephen's code was taken and modified to correct errors and to make it more user friendly. A PDF of the modified text can be found in Appendix C.

The main part of the code needing to be modified lay in the mass calculation section. Although the program allowed for variable rocket mass during the burn, all of the mass equations were calculated for a specific rocket design. This did not allow the user to easily incorporate desired burn times for the different stages of a rocket. To modify this, the rocket mass calculation was rewritten utilizing a burn time user input. This allowed for automatic calculations with little to no user effort.

Once the mass was correctly calculated, the code produced plausible rocket trajectories. However, further inspection revealed that there were rocket specifications such as initial mass, drag coefficient, etc, scattered throughout the body of the code. These characteristics are rocket specific so they were included in the user-input section at the beginning of the code to avoid having users search through the code.

This code does not make many simplifying atmospheric assumptions. Instead, it has a gravitational model that takes into account the oblateness of the earth to calculate the correct gravity constant as a function of the distance to the center of the earth. In addition to the gravitational model, the code does not consider pressure, density, and temperature to be constant throughout the flight. Instead, it calculates each of these values for the different altitudes reached by the rocket.

Something else the code takes into account is an optional pitch program for the final stage of a rocket. Though the code always monitors the pitch of the rocket throughout the flight, there is an option to initiate a time based variable pitch function that changes the heading of the rocket a specified number degrees during a specified time interval. To simplify our initial analysis, this pitch function was not used in the preliminary trajectory models.

In an effort to simplify the user interface of the code, a series of graphical user interfaces (GUIs) were created to sequentially appear at the beginning of the code's operation. This system marshaled together the various inputs taken by the code and grouped them in a single place that appears when the code is run. This was done to make the code more accessible to the first time user but can prove tedious when repeatedly running the code in different scenarios. If this becomes tedious, the code allows the GUI inputs to be eliminated while still retaining the correct inputs in an easily accessible list format at the beginning of the code. The code includes a list of instructions about which lines to comment out in order to remove the GUI.

Below are the inputs required to correctly run the trajectory code. They include rocket specifications and initial launch conditions. There is also an option to specify whether the rocket has one or two stages with the default condition set to one stage.

Inputs Required:

Rocket Specifications

- Vehicle takeoff mass in kilograms [kg]
- Drag coefficient
- Rocket frontal area in meters squared [m^2]
- Specific Impulse for each stage in seconds [s]
- Thrust for each stage in newtons [N]
- Burn time for each stage in seconds [s]

Launch Conditions

- Initial launch altitude in meters [m] – altitude vector is perpendicular to the local meridian line
- Initial relative launch velocity in meters per second [m/s] – relative to the rotating atmosphere
- Initial relative flight path angle in degrees [deg] – relative to the plane tangential to the altitude vector
- Initial relative velocity azimuth in degrees [deg] – North [0°], East [90°], South [180°], and West [270°]
- Initial geodetic latitude (geographic latitude) in degrees [deg]
- Initial geodetic longitude (geographic longitude) in degrees [deg]

If you want to incorporate a pitch program, provide the following values:

- Time after launch to begin the pitch program in seconds [s]

- Pitch program duration in seconds [s]
- Pitch rate in degrees per second, must be a positive number [deg/sec]

Once these rocket specifications and initial conditions are input by the user, the code conducts a series of calculations through various functions to calculate the theoretical trajectory of the vehicle. These equations are listed using Stephen Phillips' notes in Appendix D.

The numerical outputs of this code are simply the maximum altitude and velocity that the vehicle reaches. The graphical outputs, however, vary slightly based on whether the pitch program mentioned above is utilized. The code will always output two figure windows. The first window will include four subplots of graphs including: altitude [m] versus time [s], relative speed [m/s] versus time [s], relative flight path angle [deg] versus time [s], and relative velocity azimuth [deg] versus time [s]. The second figure window will appear as a picture of the earth with one of two things on it. If there is no pitch program, the program assumes the rocket is launching straight up and falling back down, so on the figure of the world there will only be a red dot located at the given longitude and latitude. If the user does want to incorporate a pitch program in their analysis, there will be a red line tracing the path of the vehicle across the map figure. The pitch program function was not incorporated in our analysis so the figures we got only have a red dot over Cape Canaveral.

For the sake of demonstration, the following inputs were used to create a nano-satellite launch simulation for a single stage rocket based on our initial thrust analysis from the Matlab thrust program (see Table 1).

Table 1: Rocket Specification

Variable	Value	Units
m0	97	kg
Cd	0.7	
S	0.0082	m^2
ISP1	156	s
T1	3560	N
bt1	10	s

Launching this rocket from rest on the ground (flight path angle = 90°) at Cape Canaveral, 28° latitude and -81° longitude, and headed East. According to the program outputs, the maximum altitude reached is 7,808 meters with a peak velocity of 343.5 m/s. The figure window with the time based graphs of the vehicles flight is located below in Fig. 2.

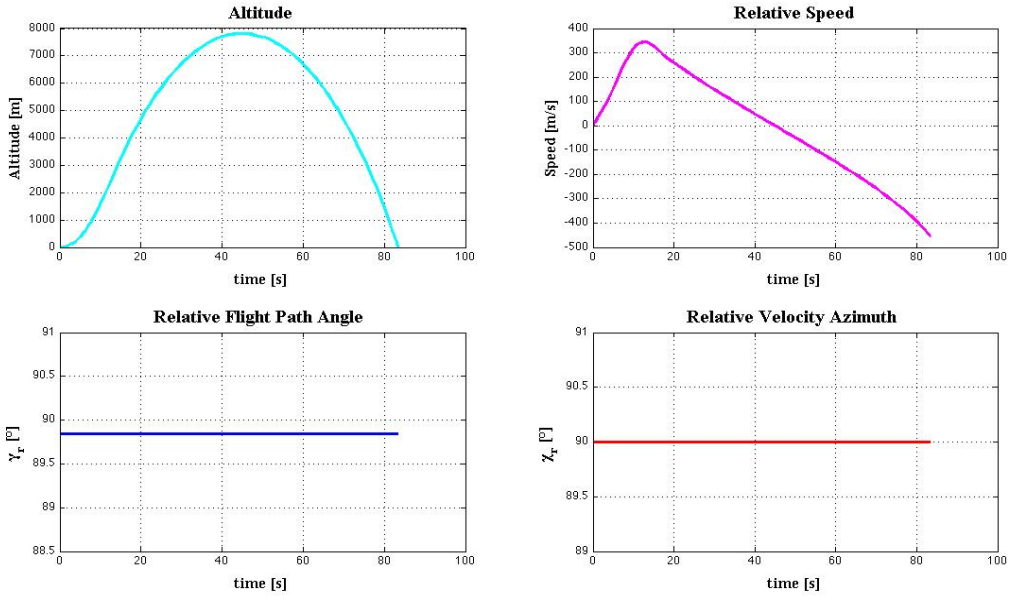

Figure 2: Ground Based Launch from Cape Canaveral

Since the goal of this design is to develop an upper stage rocket to reach LEO, the rocket will in reality not be launching from the ground at zero velocity. If this rocket were actually put into use, it would be launching from a high altitude with an initial velocity. Based on research on historical second stage launches, it was determined that second stage rockets are usually launched form around 40 km with an initial velocity of roughly 1,500 m/s. Using these as the initial conditions for the launch and a flight path angle of 23° the rocket is predicted to reach 213,150 m (which is within the range of LEO) with a maximum relative velocity of 1,787 m/s (see Fig. 3).

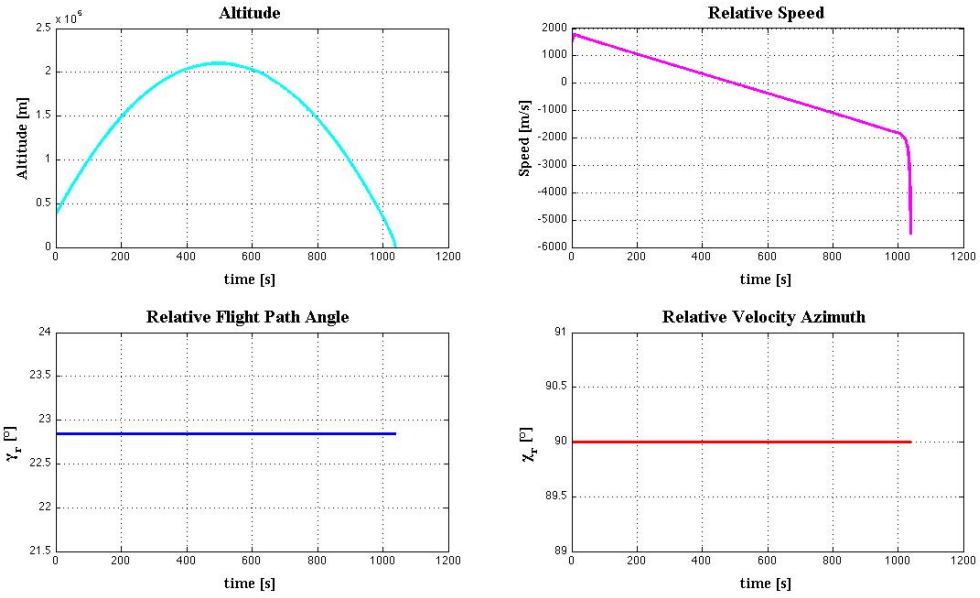

Figure 3: High Altitude Launch Simulation

As mentioned above, there will be an additional figure of the face of the Earth. Below is an example of that figure launching from the Cape. There is an extremely small dot marking the point of the launch with no pitch program enabled.

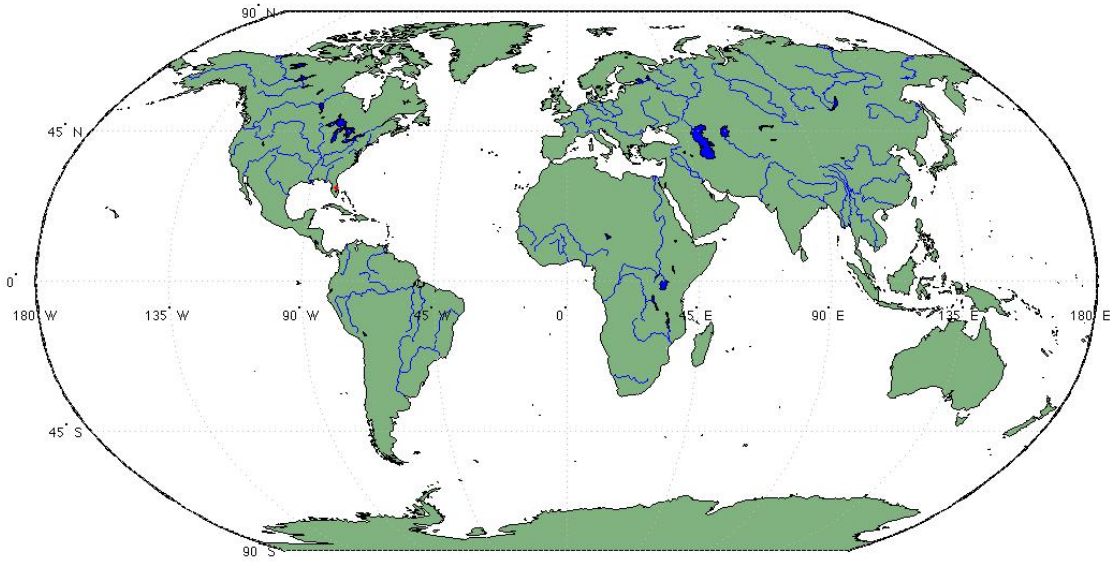

Figure 4: World Map Figure from the Code

D. Tank Design

There were several factors that affected the design of the oxidizer tank for the hybrid rocket. It is also important to note that the tank was initially designed while we still were going to use nitrous oxide as the oxidizer with aft end, vortex injection and a regeneratively cooled nozzle. A Matlab code was written in order mostly to determine the mass of a tank given the mass of oxidizer required, length of the tank, as well as other variables. This code may be found in the appendix. It was determined that a composite overwrapped pressure vessel was to be used because of its high strength and light weight. Initially this was achieved through a titanium shell wrapped in carbon fiber, with the carbon fiber being thick enough to support the tank pressure alone and the titanium shell simply to contain the gas. After investigation into the safety measures of working with nitrous oxide, it was found that nitrous oxide is able to erode reactive metals given enough time. This problem was fixed fairly easily since the titanium simply served as the shell and the carbon fiber provided the strength. This was accomplished by switching the titanium to aluminum. The tank also went through several iterations as the group varied the amount of oxidizer carried. Eventually the group decided upon 50 pounds of nitrous oxide. At the end of the semester, the group was curious if using gaseous oxygen as the oxidizer instead of nitrous oxide would lower the weight of the oxidizer tank. It was calculated through the code provided that for any given amount of oxidizer, nitrous oxide would result in a smaller, lighter tank. This is mostly due to nitrous oxide's liquid-vapor mix at room temperature which increases the density. Granted this does not account for the changing oxidizer to fuel ratio, but using gaseous oxygen would also result in the loss of the regenerative cooling effects of nitrous oxide. Ultimately the advantages and disadvantages of the two systems should be considered by any party that intends to build a hybrid rocket, but it is of the author's opinion that the benefits of nitrous oxide outweigh the benefits of gaseous oxygen. (Appendix A)

E. Nozzle Sizes Optimization

To ensure that optimum levels of thrust could achieve, it was necessary to determine the ideal expansion ratio for the nozzle. The maximum size that could be printed was 11”x9”x9” or less, the nozzle was also designed to accommodate a 3 inch buffer to allow for the injection manifold. Due to this limiting factor, the exit area of the nozzle was determined first and following that that largest area ratio in comparison with the throat was determined through trial and error. Initially, the total length of the nozzle including the converging section was 6.65 inches and the area ratio with the most desirable outcome was 5. The initial weight for the first design of the nozzle was

estimated as 13.2275 pounds when the channels were included. With pressure set at 500, mass flow rates of 3, 4, and 5 pound mass per second were analyzed and the areas and area ratios were determined.

Table 2: Nozzle Areas

$\dot{m}$ (lbm/s)	$A^* in^2$	Ae/A^*
3	0.9298	38
4	1.2	26
5	1.4765	24

For a length of 8.9 inches, the maximum diameter in inches was determined as 6.86 inches and the throat radius was 0.686 inches.

The hottest section of the throat was experimentally found to exist at a segment just upstream of the throat for m = 5 pounds and Pc=600. The hottest conditions for the nozzle were also located in the center. For the previously stated conditions, a thickness of 0.02 inches will not small enough and the inner surface will regress until you reach a thickness near 0.011 inches at which point the cooling from the nitrous will be enough to maintain low enough temperatures ad stop wall regression.

F. Nozzle Design Timeline

The design of the nozzle, head end plate, and aft end plate all evolved as the objectives of the project progressed. Initially, the group expected for NASA Marshall to use their Inconel printer to print both plates as well as the nozzle. As time progressed, it became apparent that printing the plates was not economically feasible. Mechanically, the project went from aft end vortex injected, regeneratively cooled nozzle to head end vortex injected, water cooled nozzle to head end injected water cooled nozzle. The choice of oxidizer went from nitrous oxide to gaseous oxygen to compressed air. The number and size of the cooling channels changed with more in depth heating analyses of the nozzle as well as testing of how small channels could the printer successfully create. How the water got to the channels, once we agreed upon a water cooled nozzle, also changed. Various manifolds and whether they were to be printed or attached later also appeared in the copious variations. Several limitations were discovered of the printer as time elapsed. First, the edges had to be rounded to avoid thermal stress fracture in the printed object. Next, interior surfaces of the object had to be within forty five degrees of the vertical printing axis in order to eliminate sagging of the molten metal. The various versions of the nozzle should tell the story of the evolution of the project as a whole through the various features.

The first nozzle was less of a functional design and more of a rediscovering of 3D modeling CAD software. It has an area ratio of twenty seven, was designed for a mass flow rate of five pound mass per second, demonstrated the general shape of a nozzle, and gave a rough idea of what the cooling channels would look like for such a nozzle. This regeneratively cooled nozzle was drawn while using aft end, vortex injected nitrous oxide as the oxidizer. The pictures below demonstrate these characteristics:

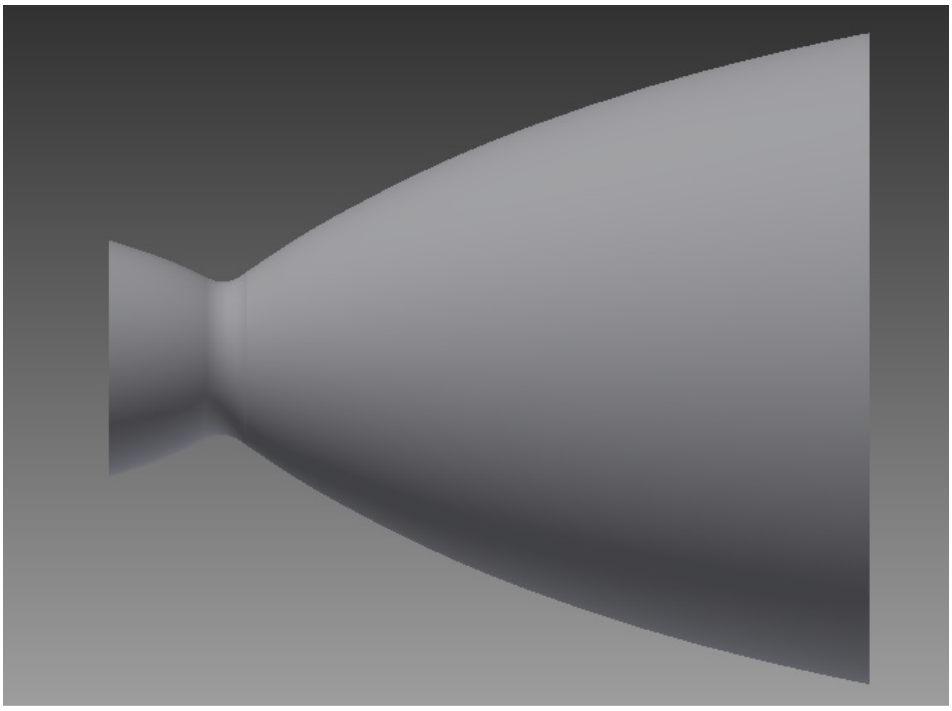

Figure 5: Initial nozzle design

The following picture portrays the cooling channel exit at the head end of the nozzle:

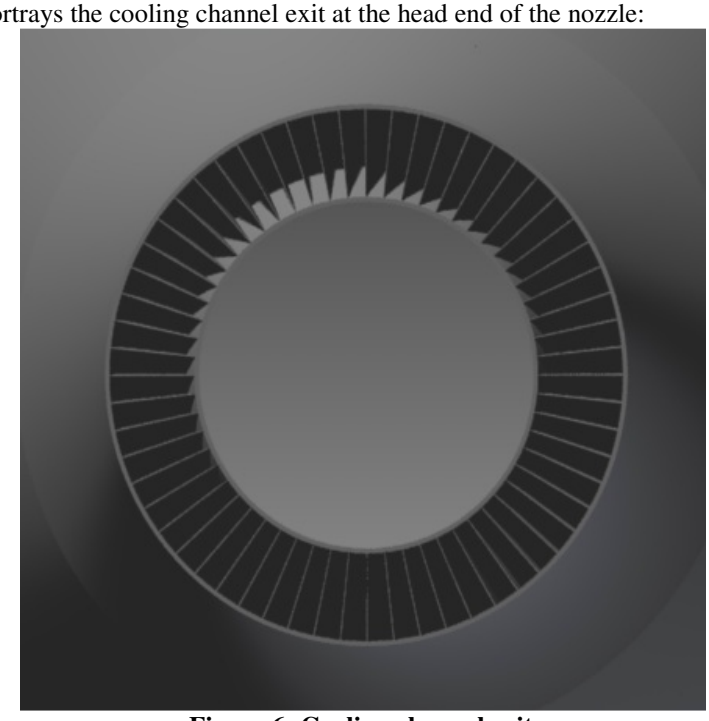

Figure 6: Cooling channel exit

The second iteration of the nozzle shows the change to a water cooled nozzle, but not to gaseous oxygen as the oxidizer. It still, however, is based on the aft end, vortex injected oxidizer. Close inspection might also reveal the rounding of sharp edges on the printed pieces. While this may seem a trivial addition, when working with wall thicknesses on the order of tens of thousandths of an inch this can become difficult to achieve. This iteration also marks the beginning of the printed plates. Printed plates allowed for the aft end, vortex injection to be achieved through an oxidizer manifold built into the plate. It was to be injected into the fuel grain at thirty degrees from four

different injectors. This also had to be managed around a pressure tap in order to take combustion chamber pressure readings. Finally, the group intended to place a manifold on the head end of nozzle to prevent having to weld a manifold on the nozzle near the plate. Hopefully the pictures below demonstrate these characteristics:

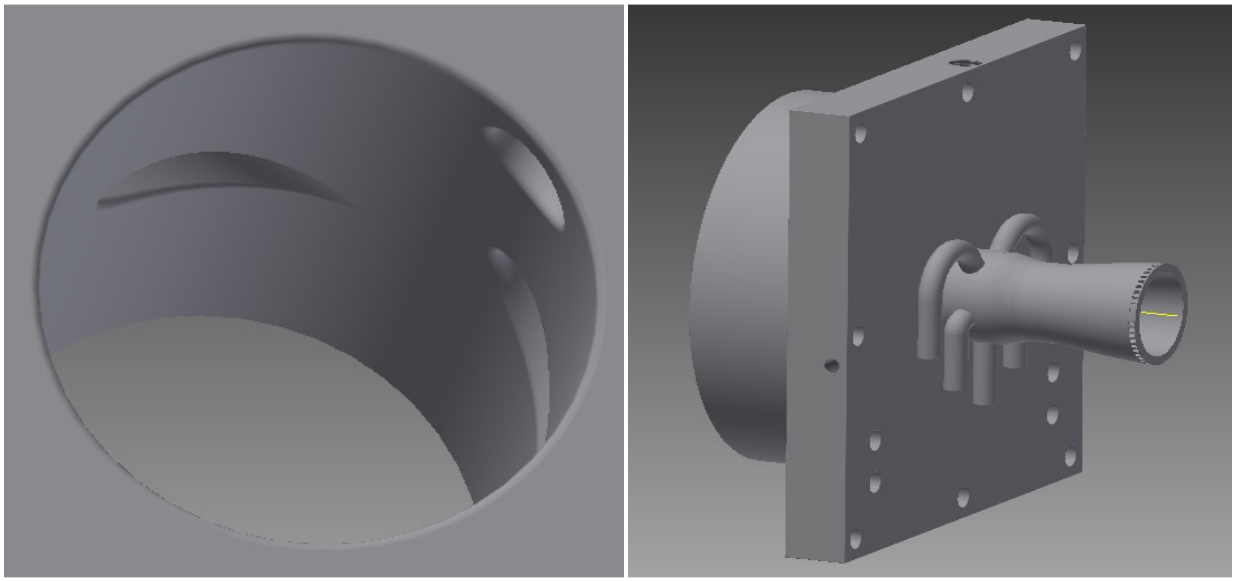

Figure 7: Injector and Nozzle Manifold End Plate

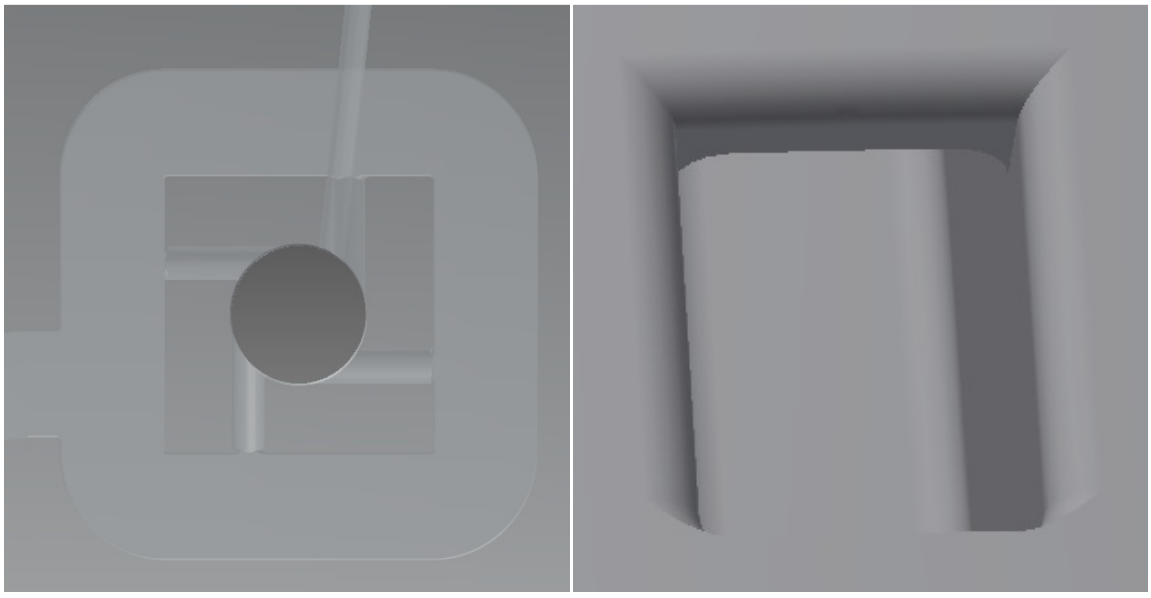

Figure 8: Injector and Cooling Channel Entrance

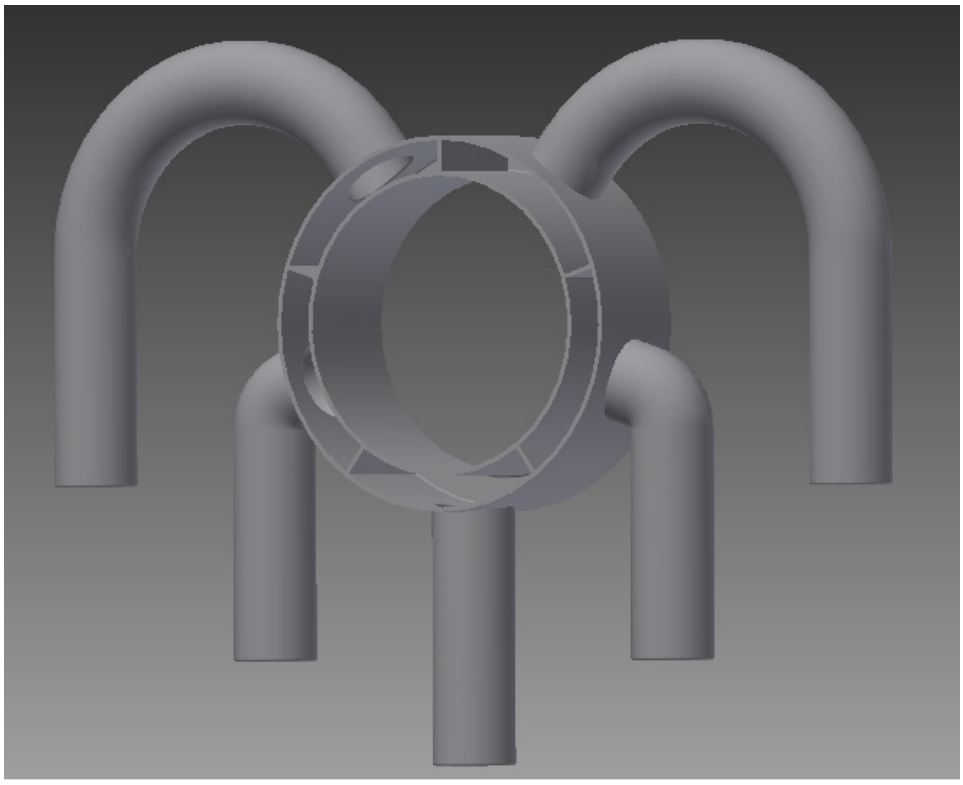

Figure 9: Nozzle Manifold

The third variant of the nozzle changed the channel inlets to a circular inlet so as to eliminate the trouble with rounding off of small edges as seen in the close up of the channel inlet in the last model. It also changed to a head end, vortex injection and made the nozzle a separate piece from the aft end plate. Lastly, it got rid of the manifold on the head end of the nozzle.

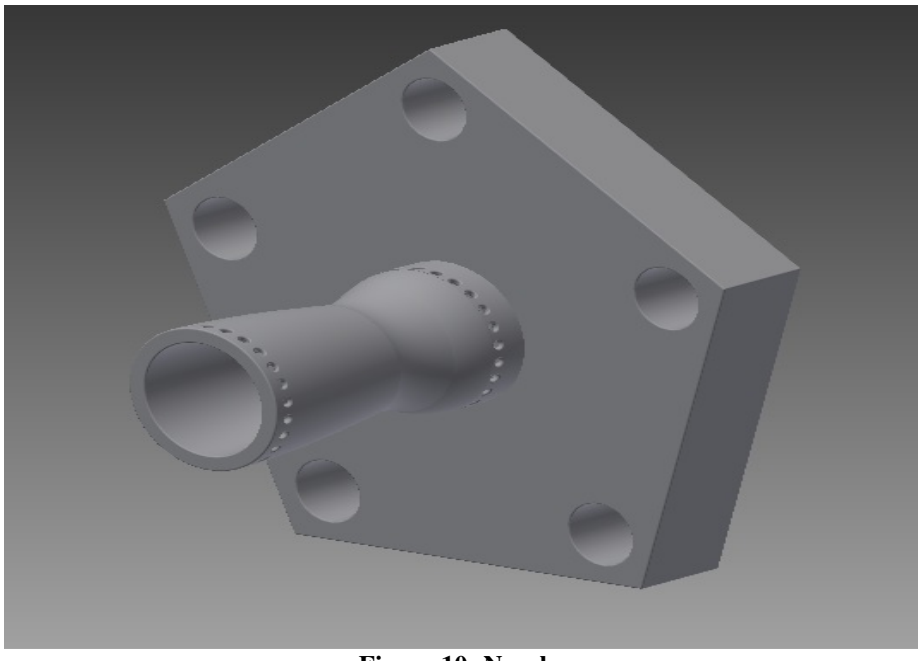

Figure 10: Nozzle

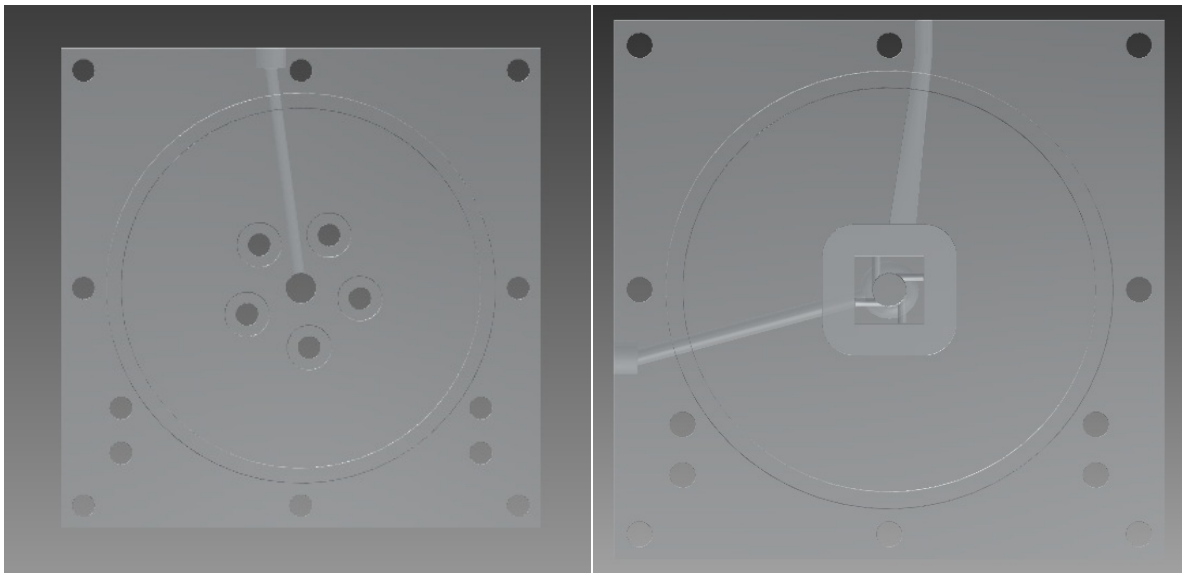

Figure 11: Aft End Plate and Head End Plate

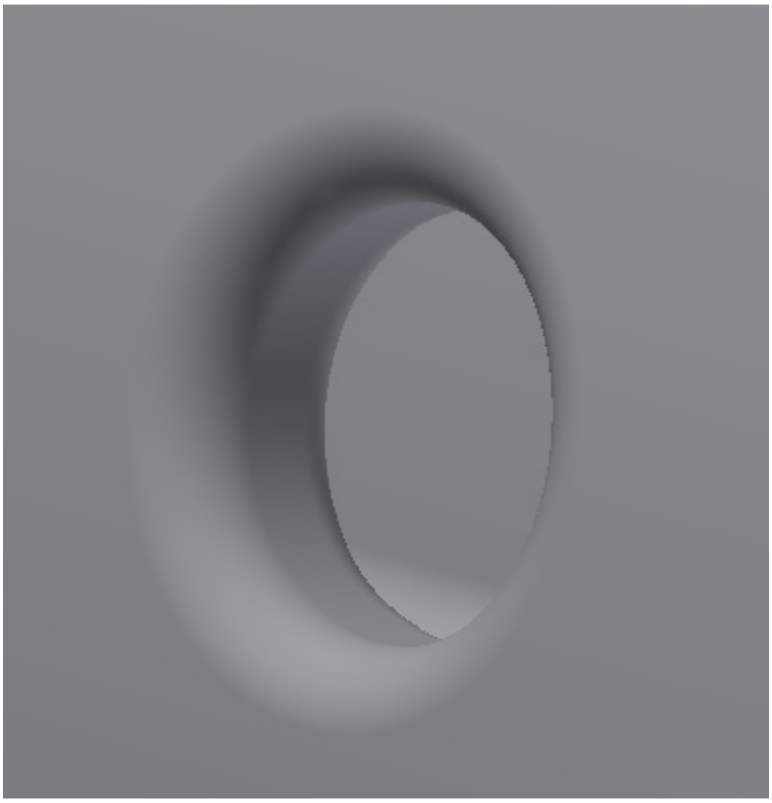

Figure 12: Cooling Channel Entrance

The forth nozzle design was based on the confidence of NASA for the excess material in the cooling channels of the nozzle to clear out. Based on that, the group could print manifolds onto both ends of the nozzle that converged to one inlet and one outlet as there was no need to test that each individual channel was cleared. Other than this, the design remained largely unchanged.

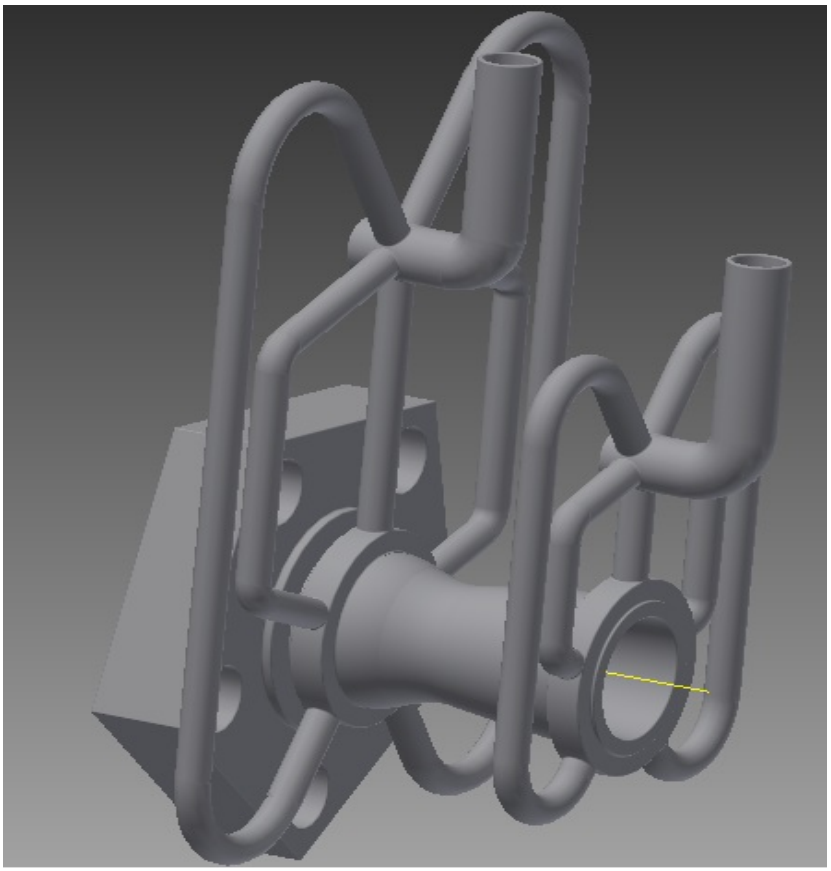

Figure 13: Nozzle Apparatus

The fifth iteration of the nozzle was inspired by the discovered limitation of interior horizontal surfaces. Based on how the 3D printer works, if there is an interior horizontal surface it will sag as there is no way to support it from the outside. This limitation led to the scrapping of the original manifolds in favor of more compact, more streamlined manifold presented below. Upon presentation of the nozzle, NASA was concerned about the structural integrity. The stress analysis showed that for the given thrust, the nozzle had a safety factor of 50.

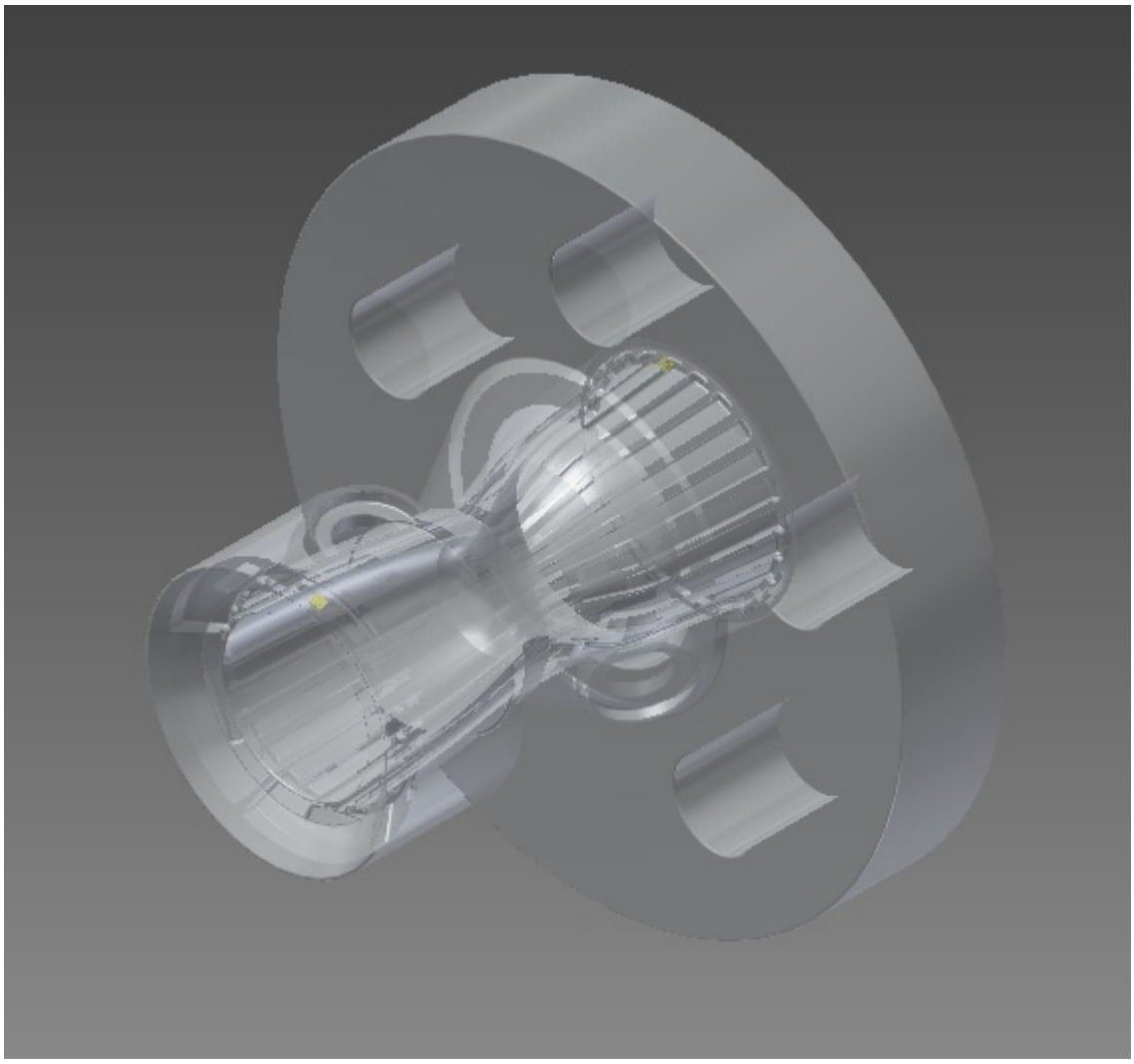

Figure 14: Final nozzle design

The last iteration of the nozzle was simply to make another inlet and outlet and increase the size of all four. This change was to ensure that the water flow for the nozzle was choked at the channel inlets as opposed to any of the manifold openings. To accommodate these openings, the connection plate was also changed to have four bolt holes so as to keep the openings away from bolts sticking out.

G. Initial Thrust Stand Analysis

1. Goals

The main purpose in using the thrust stand is to validate the data obtained from the analytical thrust code. Experimental data for thrust, combustion pressure, coolant temperatures, coolant mass flow rates, and oxidizer mass flow rates will be obtained from instrumentation and compared with theoretical values. In addition to these values, a future goal for the thrust stand is to determine the failure criteria for specific nozzles.

2. Nox/ABS

The thrust stand was initially designed for a mix of Nitrous Oxide and Acrylonitrile butadiene styrene (ABS) plastic. Most of the initial calculations were conducted for this mixture and an oxidizer to fuel ratio of 6.5:1. However, after meeting with representatives from NASA and Utah State University, Nitrous Oxide was determined to be too dangerous an oxidizer to work with. This was due to the regenerative cooling aspect of the nozzle design. Nitrous dissociates exothermically at temperatures around 1000° F. This can result in violently explosive reactions. In order to keep the nitrous oxide from dissociating, the oxidizer mixture must be kept below a quality of 60%. This area of the experiment had a large number of dangerous unknowns, so the decision was made to change the oxidizer

to oxygen. This would allow a water cooling setup to be implemented to determine the heat flux experienced by the coolant. Using this data, a projection of the potential temperature change experienced by nitrous oxide can be made. This will be invaluable in keeping the oxidizer temperature away from dissociation.

3. O2/ABS

The next experimental setup used a mix of oxygen and ABS plastic. This mixture would run with an oxidizer to fuel ratio of 2:1. While initially viewed as a better alternative than using nitrous oxide, oxygen also proved to be a dangerous choice of oxidizer. Analysis of combustion experienced by the nozzle while burning oxygen indicated temperatures would exceed the limit of our materials. The oxygen mixture would burn so hot that the structural integrity of the nozzle could not be guaranteed. To fix this, the thrust stand was then retooled to use air as the oxidizer.

4. Air/ABS

The final and current setup for the thrust stand uses air as the oxidizer. This would result in an oxidizer to fuel ratio of 10.5:1. Air does not dissociate exothermically like nitrous oxide does and has a much more manageable combustion temperature than oxygen. Air is the current oxidizer being used with the thrust stand.

III. Experimental Setup

A. Cooling System Design

After the meeting at NASA, Marshal; the decision was made to revise the rocket motor design from being regeneratively cooled by incoming nitrous oxide oxidizer to being cooled by liquid water. It was decided to retain a nozzle cooling system instead of switching to an ablative nozzle. Initially, the same nozzle design was used, but with the connections for water to enter and exit and without an outlet to pump coolant into the combustion chamber (which would have been used when the coolant was also the oxidizer). The initial cooling system design was based off of nozzle cooling channel geometry, expected heat flux, and the minimum temperature change resolvable by K-type thermocouples. Additionally, it was required that the total cross sectional area of the cooling system be kept relatively constant in order to keep flow velocities relatively constant throughout the cooling system and to minimize entry lengths at each component. Minimizing entry lengths was thought to be useful as it would produce fully developed flow throughout the majority of each component thus simplifying analyses.

In order to meet these requirements, the initial design was relatively high pressure and required either a high pressure pump or a supply of pressurized gas to force the cooling water through the system. At this point three possible cooling system pressurization systems were considered. Assuming that a pump could be found that could provide sufficient pressure and mass flow; the first and simplest design consists of an unpressurized water tank and constant pressure pump. The other two designs required pressurization of the water tank. The first of these designs consisted of a pressurized single port water tank with pre-run pressurization coming from a compressed air tank and cooling system pressure regulation controlled by a water pressure regulator. The last design consisted of a dual port water tank with a pressurizing air tank attached to the upper port and the lower port used as the water outlet with cooling system pressure regulation accomplished by a high flow gas pressure regulator between the air tank and the air/water tank. See Figures 15-17, below and on the next page.

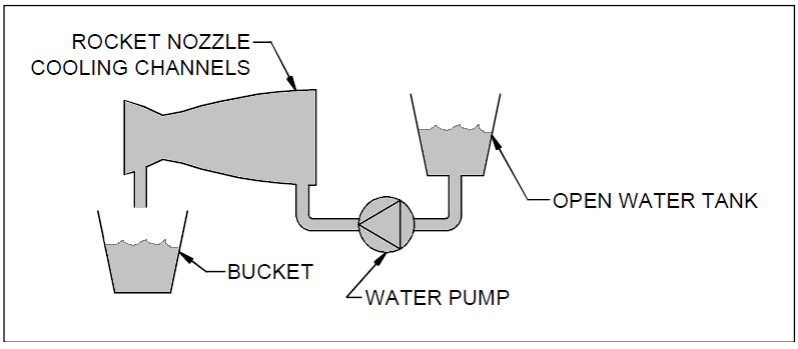

Figure 15: Pump Based Water Exhaust Nozzle Cooling System

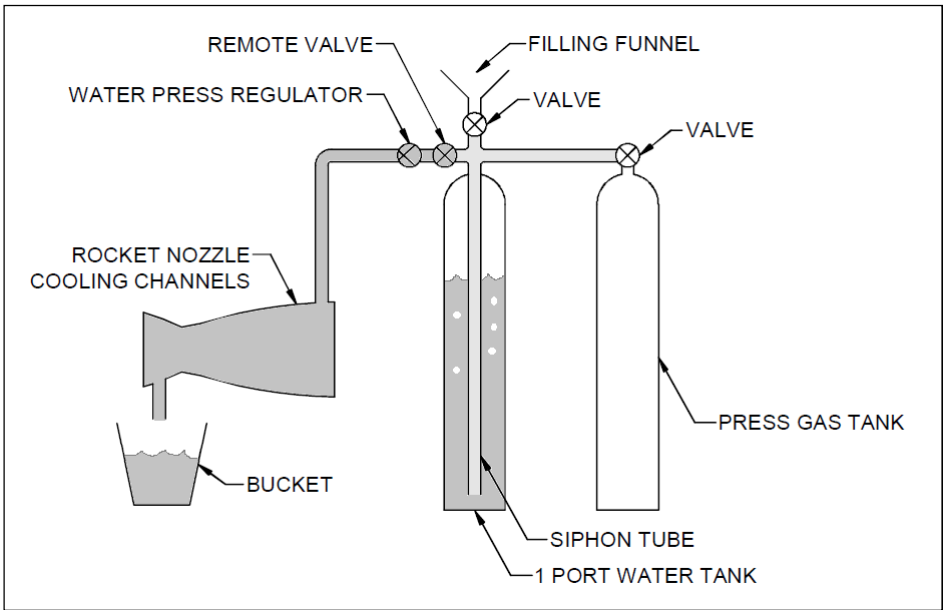

Figure 16: Single Port Water Tank Exhaust Nozzle Cooling System

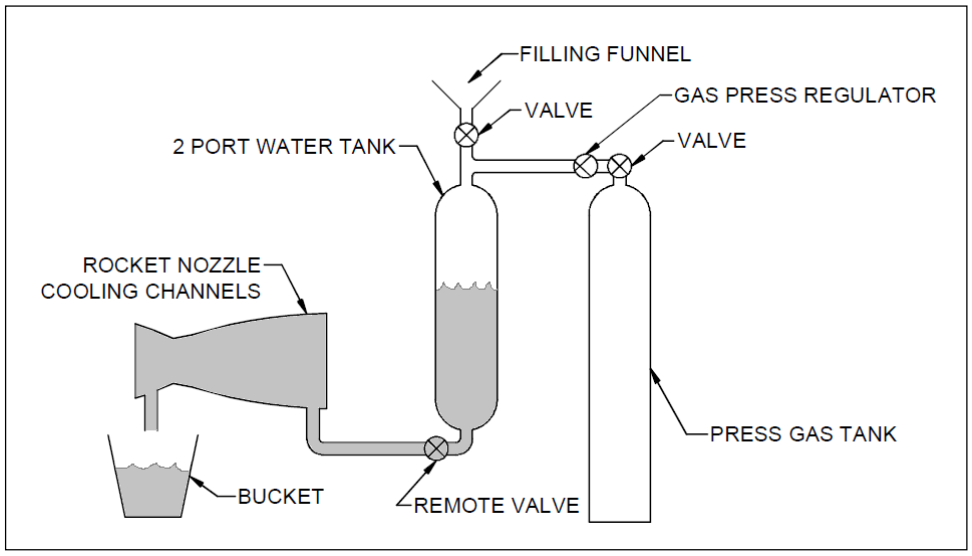

Figure 17: Dual Port Water Tank Exhaust Nozzle Cooling System

While the pump based cooling system is operated simply by ensuring that the water tank is filled and then turning on the pump when needed, operation of the other two designs is more complex. For the single port design, there are three valves. At any point in time only one of these should be open. The procedure for a single run consists of: First, filling the tank with a known quantity of water so that there is sufficient head room for the compressed air that will force the water out. Second, the water tank is pressurized with gas from the pressurized gas tank and then the valve between the two tanks is reclosed. Finally, a remote valve is opened to flow water through the rocket nozzle cooling channels during a firing. For the dual port design, the water tank is filled before a run and then the water tank is pressurized by opening the valve between the two tanks. This valve is then left open for the run. Then, when it is time for a firing, a remote valve is opened and pressurized water flows through the cooling system.

Soon after, the size of the rocket motor was scaled down and the cooling channels became shorter as a result of this. This meant that the pipe losses through the cooling channels were reduced significantly. Which, reduced the pumping pressure requirements for the system significantly and a design using an electric pump to supply water to the nozzle cooling channels the best choice.

A further redesign was required when the cross sectional area specified for the cooling channels was greatly reduced in order to keep the water pressure in the system high enough to prevent steam from forming at the channel walls and impeding heat transfer. This raised the required pumping pressure for the system significantly. Thus, no pumps were economically available that could provide both the necessary pumping pressures and sufficient water mass flow rate. One of the two systems using pressurized gas for pumping would need to be used.

Sensors, tubing, adapters, and other parts were selected to meet these design requirements. This design used flared steel tubing to connect the various components together because tubing can be made to any desired length, can bent to meet geometric requirements, and is economical. An Omega flow meter was selected that met our flow and pressure requirements and Nanmak pipeline thermocouples were selected because they are the only thermocouples we found that are designed specifically for measuring temperature in pipes. Additionally, a custom high pressure water tank was ordered with a connection on top to accept pressurized gas and a high pressure water outlet on the bottom. Later, it was decided that Omega pipe plug thermocouples should be used instead of pipeline thermocouples in order to meet budgetary constraints.

A further redesign was initiated when it was decided that the area of tubing prior to the rocket nozzle connections should be increased in order to ensure that the nozzle cooling channels were the limiting point in the flow path. Again a parts list was created to meet this new requirement. Then, it was decided that hydraulic hoses should be used instead of flared tubing and a new parts list was drawn up.

Then, it was discovered that personnel at NASA had accidentally printed one of their own nozzle designs. Because the wait time for the nozzle design that we had been working on was prohibitive; it was decided that we should use this nozzle instead. A new parts list was made up and finalized. Then the parts were ordered. Parts have arrived. No further action has been taken.

Connections for water intake and exhaust also went through multiple iterations. Initially, most designs for the exhaust nozzle included several inlets and outlets. Multiple connections required that external manifolds be designed to break the flow from the water tank into an appropriate number of lines while keeping the pressure of the water inlet and outlet lines equal across each set. So that, cooling would be even on all sides of the nozzle. The diminutive size of the lines between each manifold and the nozzle led to the decision to print the manifolds, lines, and nozzle as one piece. See, Figure 18 below. However, this design was deemed too prone to damage. So, it was redesigned so that two large inlet and two large outlet connections were integrated into the main nozzle body. This simplified the external fitting requirements. So that only a single tee-fitting was needed to separate the water flow on the inlet side of the nozzle. See, Figure 19. This final design featured internal baffling to even out water pressure from the two inlets and a large volume on either side of the baffles to allow incoming flow to slow in order to further prevent uneven pressure. The design from NASA, which replaced our final design, features six inlet and six outlet ports. Thus requiring an external manifold similar to our initial designs. Please see, Figure 20, next page.

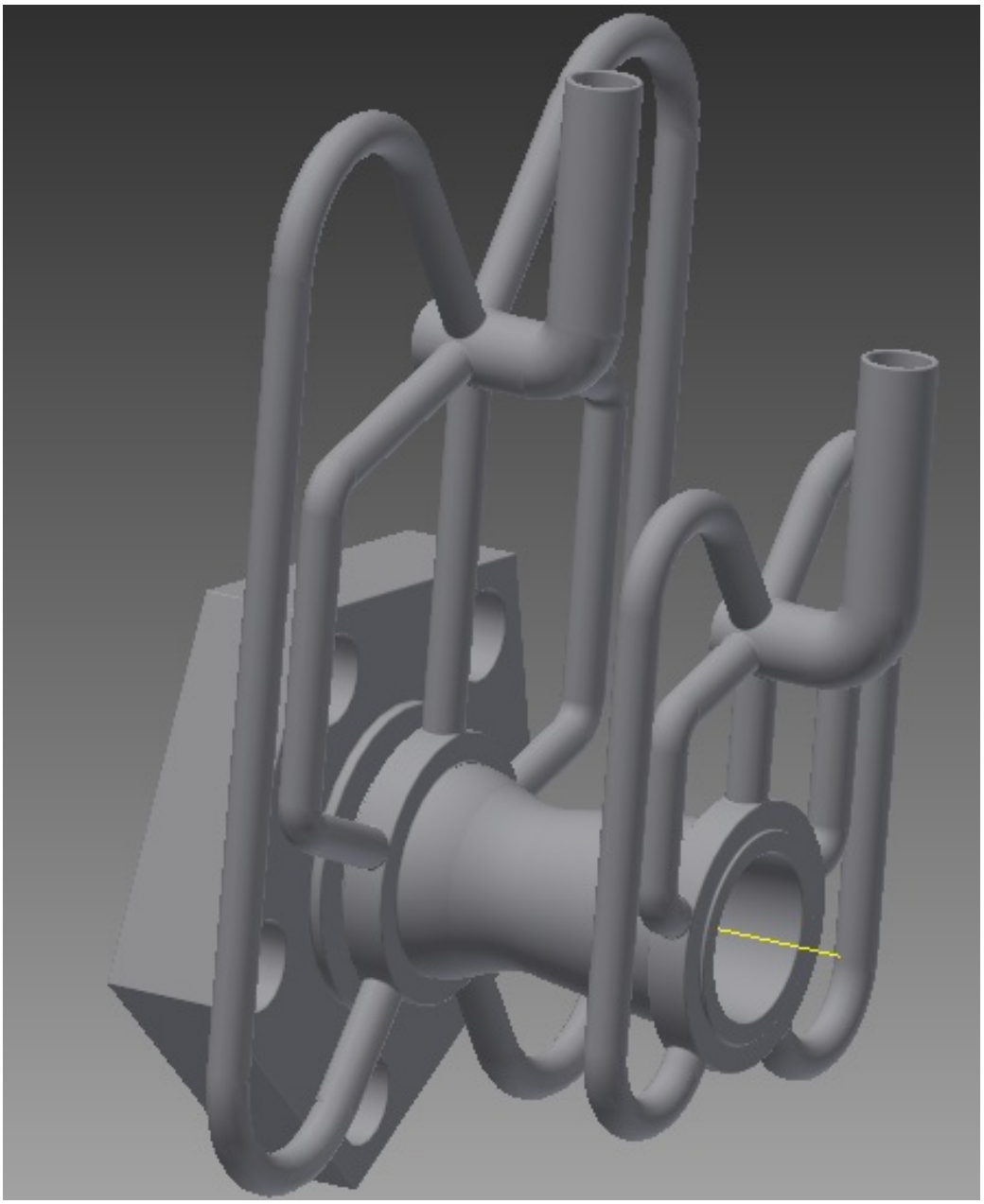

Figure 18: Initial Cooling System Manifold Design with integrated One-to-Five Manifolds

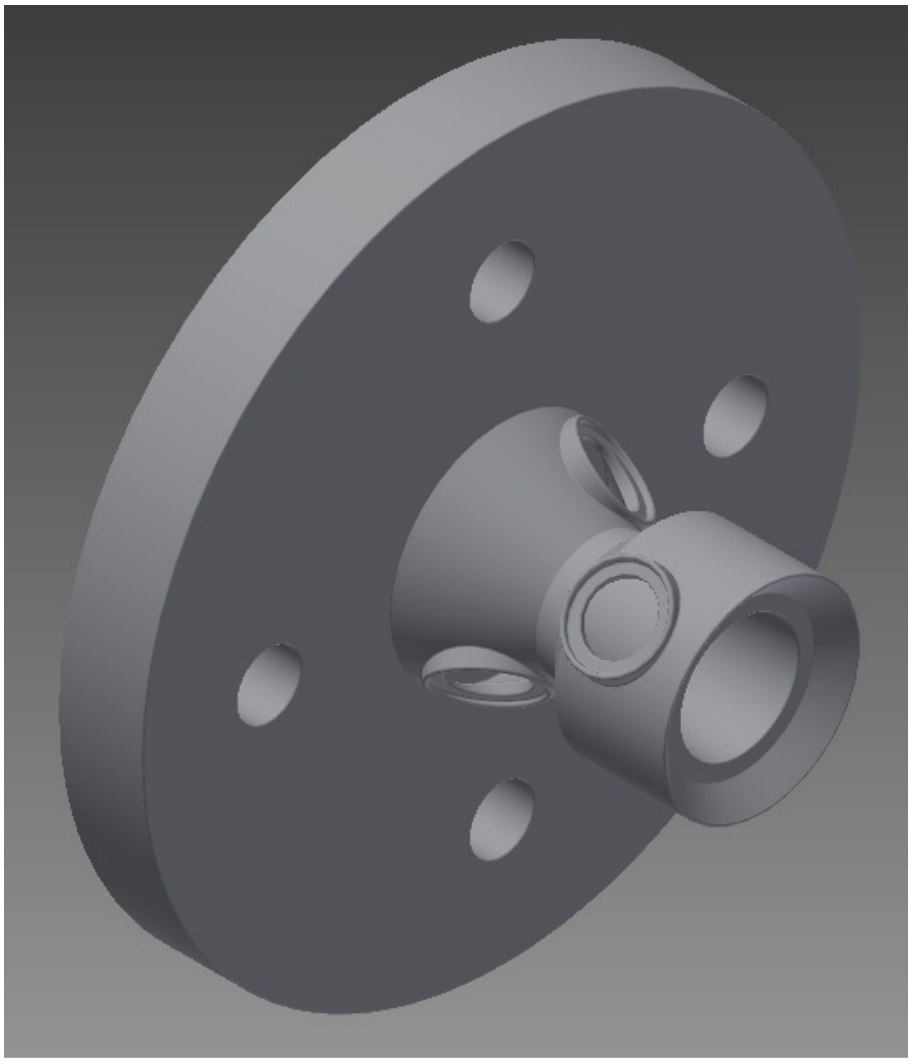

Figure 19: Final Cooling System Manifold Design with 2 Inlets, 2 Outlets, and Internal Baffling

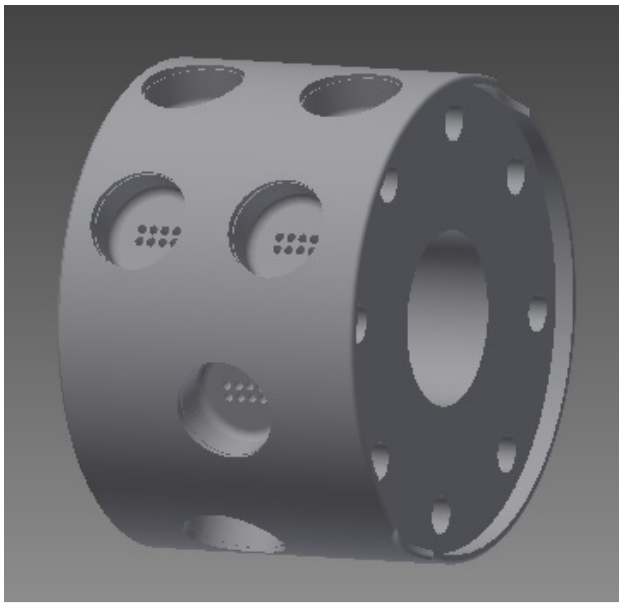

Figure 20: NASA Nozzle Cooling Manifold Design Featuring 6 Inlets and 6 Outlets (Channel Inlets Visible in Ports)

B. Data Acquisition

The data acquisition (DAQ) setup used 8 instruments to record data on temperatures, flow rates, pressures, and the thrust of the rocket. This data comes from the cooling system and the combustion chamber of the rocket. Using a 12 volt battery as an external power source, a common voltage was supplied to all of the instruments. A circuit was constructed with power being supplied to the instruments from the battery, instrumental output being directed to the DAQ board, and instrument grounds being directed to the power source and the DAQ board. Figure (21) illustrates the setup.

The National Instruments LABVIEW programming software was initially used to conduct the experiment. LABVIEW was selected due to its ability to trigger different aspects of the experiment at different times. The original idea was for the launch operation program to be completely automated through LABVIEW. This program would control the opening and closing of valves at specific times as oxidizer was injected and the rocket was ignited. The ignition system would also be a part of this fully automated program. Furthermore the launch program would read and save data from the data acquisition system for further analysis. Unfortunately, due to time delays in instrument procurement, the implementation of an automated LABVIEW control program became unfeasible. Due to timing constraints, TracerDAQ was implemented to log experiment data.

TracerDAQ is a programming suite that ships with Measurement Computing hardware. Compatible with the Minilab-1008 DAQ board used in the experiment, TracerDAQ proved to be an effective short term substitute for LABVIEW. Using the strip chart function of TracerDAQ, data was read from the instruments in a continuous scanning mode (Appendix B**).** This output a plot of data versus time that was then saved in a text file for future analysis.

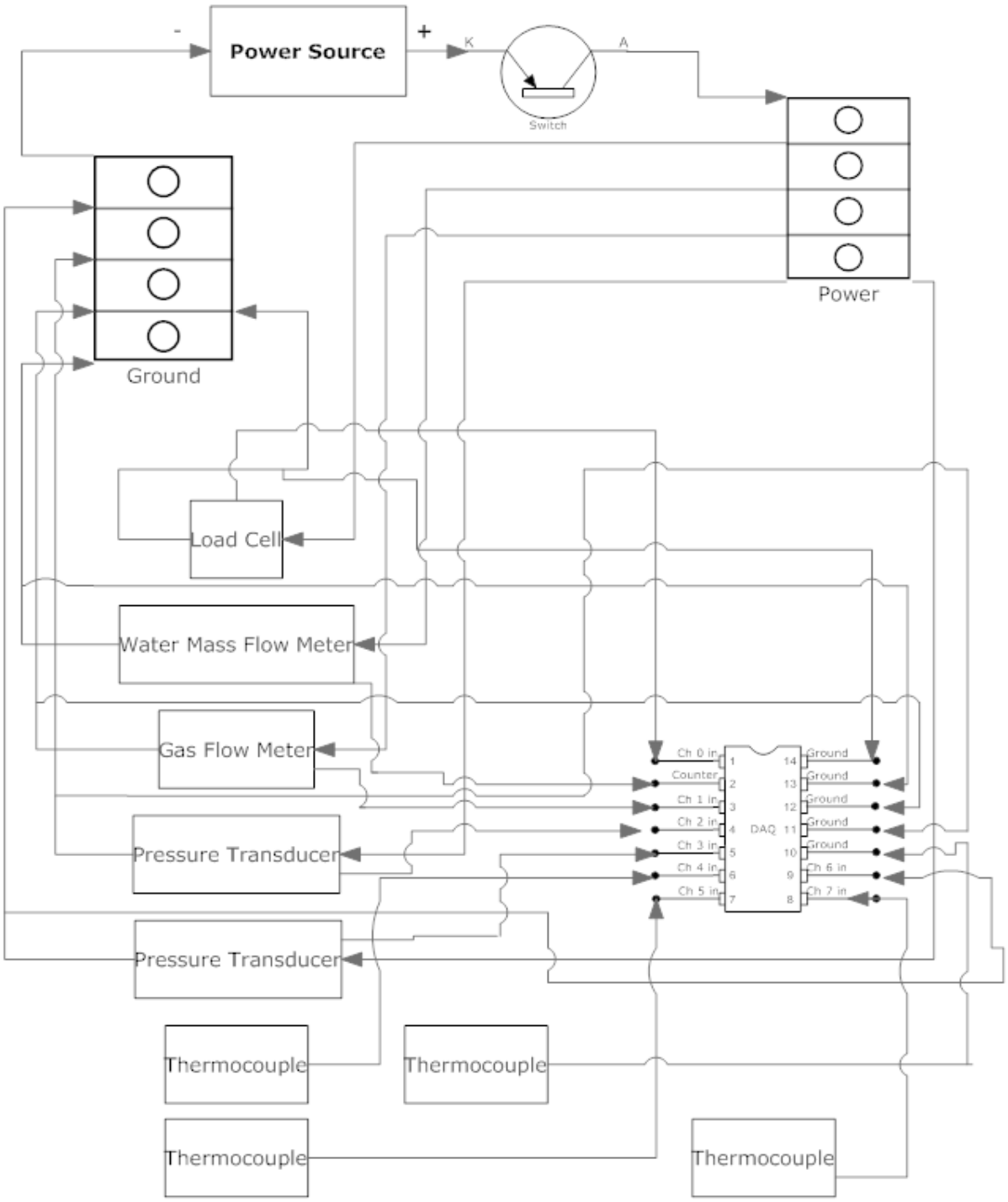

Figure 21: Electrical Diagram for instrument setup

C. Instrumentation

1. Measurement and Computing MINILAB 1008

The data acquisition apparatus used with the thrust stand utilized a Minilab 1008 analog to digital converter. Using the 8 single-ended inputs, an analog signal of 0-10 volts direct current was read from the instruments (Appendix G). Each instrument receives power from the 12 volt battery and directs its instrumental output to the data acquisition board. When coupled with the TracerDAQ software, data is gathered in a multi-channel continuous scan sequence with a maximum sampling rate of 1.2 kilo-samples per second. This sampling rate is aggregate, so when all eight channels of the DAQ are in use, the sampling rate is 150 samples per second. (Appendix G) The Minilab 1008 is then connected by USB to a laptop computer controller that records and saves the data.

2. Omega PX603 Pressure Transducer

When choosing the pressure transducer for the experiment, special consideration was given for the environment that the experiment would be performed in. The Omega PX603 pressure transducer was selected due to its stainless steel construction and high reliability. Effective at pressures ranging from 0-2000 psi, the PX603 was used to measure the pressure of the combustion chamber and the pressure of the coolant. This instrument output an analog signal of 1-5 volts direct current. (Appendix G)

3. Omega Type K Thermocouples

Thermocouples were chosen to accurately measure changes in water temperature around 70°F. The type K thermocouple was chosen due to the large range of temperatures it can accurately measure. It also has a low profile so it will not cause a large increase in drag when inserted into the water cooling system. Designed to be extremely rugged and resilient, the K class thermocouple also came with ¼ or 1/8 NPT fitting. This allowed it to be easily integrated into the coolant system.

4. Omega FPR203-PC-PS Water Flow Meter

A water flow meter was used to accurately measure the mass flow rate of water flowing through the coolant system at any given time. Flow meter selection was determined by the constraints that it had to be able to measure at least 2 lbm/s at a temperature of roughly 70°F. The Omega FPR203-PC-PS was chosen due to its economical scale of value. Using the pulse output, the flow rate was read as a frequency of counts into the CTR port of the Minilab 1008. (Appendix G)

5. Omega FLR9760D Gas Flow Meter

A gas flow meter was used to accurately measure the mass flow rate of oxidizer through the rocket fueling system. The main constraint on the gas flow meter selection was that it had to accommodate 0.35 lbm/s. The Omega FLR9760D was selected to due to its flow capacity and ability to withstand the oxidizer being used. This instrument output an analog signal of 0-5 volts direct current. (Appendix G)

6. Tovey Engineering "S" Type Load Cell

Able to withstand a load ranging from 0-250 pounds, the Tovey Engineering "S" type load cell was selected. Designed as a typical strain sensor, the load cell outputs a voltage of 3 milli volts per volt based on the force applied to it. A signal amplifier was used to increase the mill volt signal to a range of volts direct current detectable by the data acquisition board. (Appendix G)

D. Ignition Systems

There are a variety of different ignition systems the may be employed when designing a hybrid rocket. For the purposes of this project, two main types were the point of focus. The first was pyro-grain based in which a small solid rocket fuel grain from an Estes motor was to be ignited using an electric starter. The electric starter is meant to create a spark within the pyro-grain simultaneously with the introduction of the oxidizer. The second type of ignition was electrically based and entailed igniting the oxidizer from the heat produced between two wires. For this hybrid rocket design, the combustion type ignition was the first to be considered; it is the most common type of hybrid rocket ignition and has proved to be a reliable ignition system. The design for this ignition system consists of one to two model Estes rockets attached to a cap at the end of a fuel grain with black powder placed inside the fuel grain and the oxidizer injector inserted in the same area. The cap is sealed and the rockets are ignited causing a controlled explosion inside the sealed vessel.

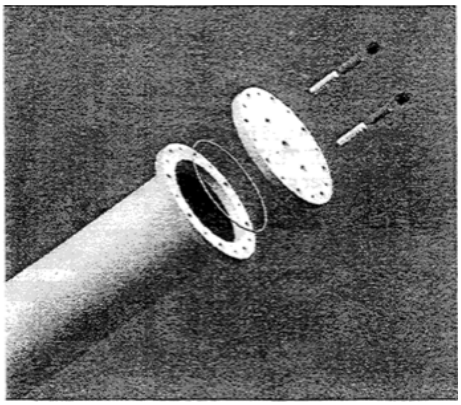

Figure 22: Rocket ignition system

One of the problems that would have been encountered in the design of this rocket, had the pyro-grain ignition system been utilized, would have been a way to mount the Estes rockets. The HTPB was used as both the fuel and the outside casing in these tests, and an extension would likely have to have been placed on the fore end of the rocket to allot sufficient room for the rocket motors. The second main type of ignition that was researched as a possibility was plasma ignition. In this type of ignition a mated wire is cut cleanly at one end and placed inside the nozzle and the other end is connected to a transformer and battery. One of the notable advantages to utilizing the plasma ignition system instead of the pyro-grain based system is that it is very reliable. While pyro-grains have been employed for a more extensive duration of time, plasma or electricity based ignition is easy to implement and in the event that a problem were to occur the power can be cut to the rocket and the test halted immediately. It was ultimately an electron ignition that was decided upon since it was possible to stop the experiment with this design as opposed to waiting for the fuel to burn down completely. For the three test fires completed for this rocket, a transformer was connected to a generator at 12 volts of power and the wires from it were connected to the fore end of the HTPB grain

E. Heat Transfer Analysis

The following heat transfer analysis was performed in order to obtain and compare the cooling capabilities of both water and nitrous oxide. The optimal design for the cooling system calls for flowing the oxidizer, nitrous oxide, through the cooling channels. Since this was designated as dangerous for us to experiment with, we will be using water as the coolant through the channels. In order to simulate the effectiveness of the nitrous oxide to cool the nozzle wall, it is critical to determine the conditions at which to flow the water to replicate the heat transfer performance of nitrous oxide.

The bulk temperature of the fluid flowing through the channel is predicted to be approximately 70°F. The following table lists the properties (density, dynamic viscosity, specific heat, and thermal conductivity) of both water and nitrous oxide at this temperature.

Table 3: Fluid Properties @ 70°F

Property	Water	Nitrous	Units
ρ	62.28	49.11	lb/ft^3
μ	6.42E-04	3.83E-05	lb/ft*s
c_p	0.9732	0.7614	BTU/lb*F
k	9.74E-05	1.59E-05	BTU/ft*s*F

These properties were used to determine the Reynolds and Prandtl numbers of each fluid. These numbers both represent dimensionless parameters that are specific to each fluid and are calculated using the following equations.

$$\mathrm{Re} = \frac{\rho V D_H}{\mu} \tag{9}$$

$$\mathrm{Pr} = \frac{c_p \mu}{k} \tag{10}$$

The Reynolds number is calculated using the average hydraulic diameter as the characteristic length of the cooling channels due to the fact that they are rectangular not circular. Using Eq. (11), the hydraulic diameter of non-circular channels can be calculated.

$$D_H = \frac{4A}{P} \tag{11}$$

The channels maintain a constant area along the length of the nozzle, but they have varying lengths and widths. Due to the perimeter changing along the nozzle, the average perimeter was used to obtain an average hydraulic diameter of 0.003634 feet.

In order to determine the heat transfer capability of a fluid flowing through the channels, the Nusselt number was calculated using the Dittus-Boelter equation.[5] The Nusselt number is a useful dimensionless parameter for the comparison of two fluids, because it is a function of the heat transfer coefficient of the flow. The Dittus-Boelter correlation applies to turbulent flow through smooth tubes with the following characteristics:

Table 4: Dittus-Boelter Flow Requirements
Re > 10,000
0.7 < **Pr** < 100

At 70°F, the Prandtl numbers for water and nitrous oxide were calculated to be 6.42 and 1.84 respectively. The Reynolds number of the flow varies based on the velocity of the flow. It was then determined that, at this temperature, water must flow at a minimum velocity of 30 ft/s while nitrous oxide must flow at a much lower speed of 3 ft/s in order to reach turbulent flow (Re > 10,000). Since the flow through the pipes meets the criteria detailed in Table 2, the Dittus-Boelter equation was used to determine the Nusselt number of the flow (Eq. 12).

$$Nu = 0.023 \mathrm{Re}^{0.8} \mathrm{Pr}^{n} \tag{12}$$

In this equation, $n = 0.4$ if the fluid is being heated and $n = 0.3$ if it is being cooled. Due to the fact that the fluid in this experiment is being used to absorb heat from the nozzle wall, $n = 0.4$ was used in our calculations. The following figure displays the Nusselt number calculated for both water and nitrous oxide when subjected to increasing flow velocities. To achieve the same Nusselt number, water must flow approximately seven times faster than nitrous oxide.

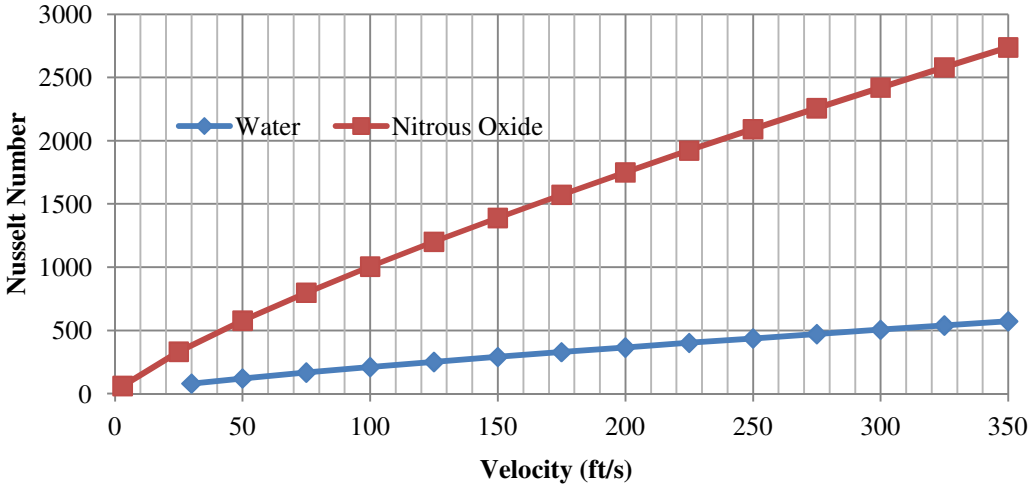

Figure 23: Nusselt Number vs. Flow Velocity of Water and Nitrous Oxide

The Nusselt number represents the ratio of convective to conductive heat transfer through any boundary that a flow is subjected to, in this case that boundary is the nozzle wall. As mentioned above, the Nusselt number can be directly correlated to the heat transfer coefficient. This can be done using Eq. 13.

$$Nu = \frac{hD_H}{k} \tag{13}$$

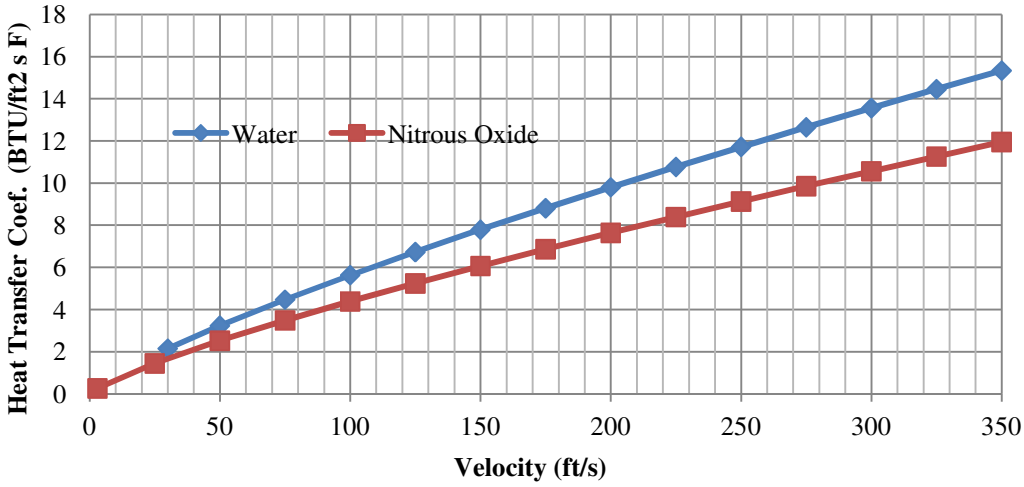

Figure 24: Heat Transfer Coefficient vs. Velocity of both Water and Nitrous Oxide

From the figure above, water and nitrous oxide flows result in the same heat transfer coefficient when nitrous oxide is flowing 1.33 times faster than water. This was interesting to discover since water must be flowing much faster to obtain the same Nusselt number as nitrous oxide. This discrepancy is due to the fact that the Nusselt number is inversely proportional to the thermal conductivity of the fluid, k.[5]

IV. Experimental Results

Using the University of Tennessee's rocket testing facility out on Alcoa highway, a series of tests were conducted using air as an oxidizer and ABS plastic as the fuel grain. These tests consisted of three successful test firings of the rocket. We attempted to gather data using a pressure transducer and a force transducer to measure combustion pressure and the thrust of the rocket. The figures below show the experimental setup.

Figure 25: Overview picture of the thrust stand

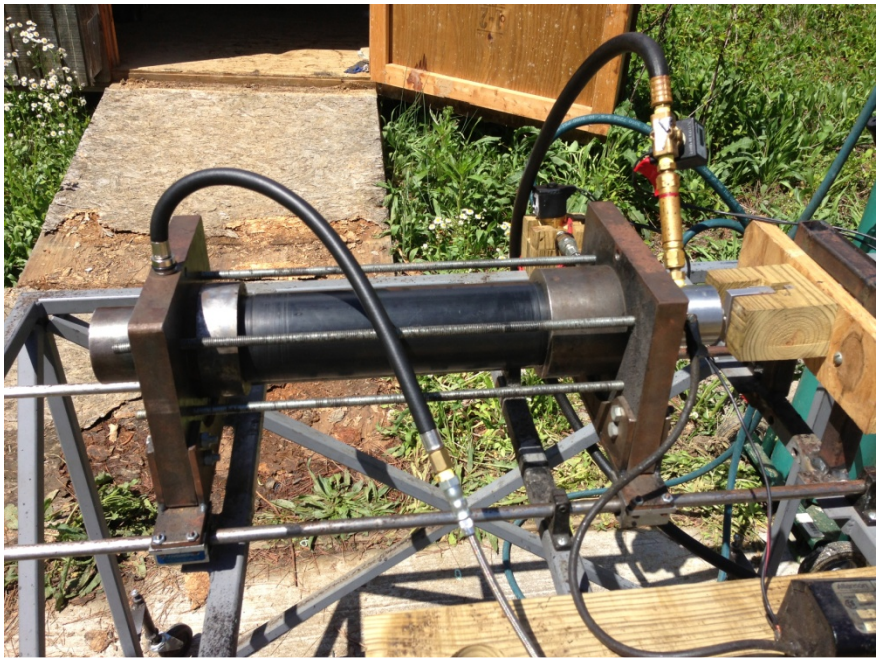

Figure 26: Fuel grain with pressure and force transducers

The test firings were overall largely successful using this setup. Data for thrust and pressure was collected for later analysis that had varying levels of success. The pressure transducer had originally showed a constant output when tested apart from the test apparatus. This constant output was consistent with the constant level of atmospheric pressure being measured so it was assumed to be functioning correctly. When incorporated with the experiment, the pressure transducer continued to give a constant output, even when the rocket was fired. Since the pressure data should have showed a large increase during the running of the rocket, that data was discarded as faulty. Future

design groups should focus on the faulty pressure transducer data as a potential source of error in the experiment and should work to remedy this. Solutions may lie in replacing the pressure transducers entirely or in troubleshooting the wire connections. Replacing the pressure transducers may be the most viable option since they were stored in a shed for a number of years in less than ideal conditions.

The force transducer, on the other hand, yielded more recognizable data than the pressure transducer. To start, a short calibration was done with a 5 kg weight applying its load to the force transducer. The figure below shows the output of the calibration.

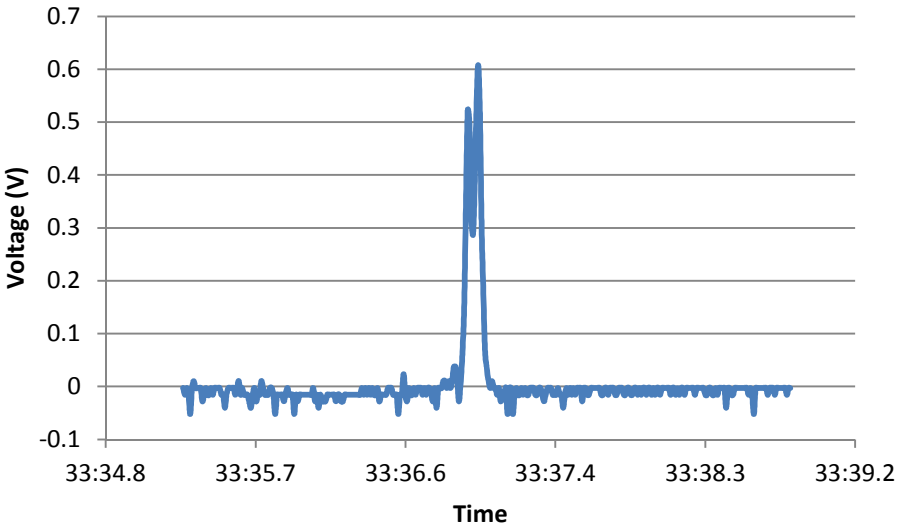

Figure 27: Force Transducer Calibration

The output of the force transducer is in milli volts. That signal is then run through a signal amplifier to give an output of volts. Using the above calibration plot, the relation of 5kg to 0.6 volts was used to convert the force transducer output to kilograms and then to pounds. Once this conversion had been done, a recognizable thrust curve was output.

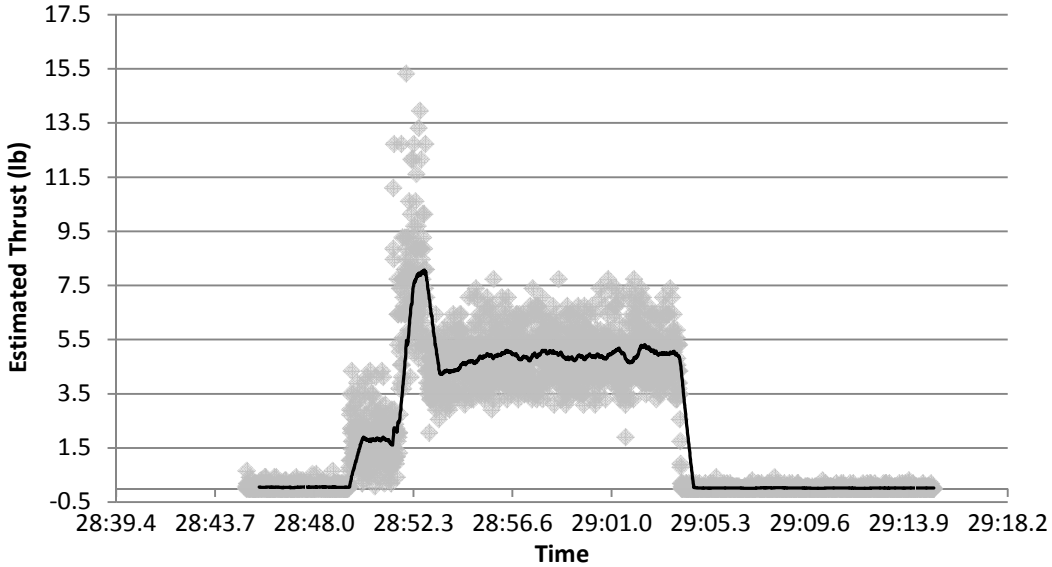

Figure 28: Rocket Thrust Curve

The data points in the above figure are displayed as grey data points. Over 10,000 data points were recorded for this run of the rocket. Due to the large amount of data, a moving average trend line was applied to the plot. This

trend line is extremely recognizable as a thrust curve. The initial step represents the ignition of the rocket using propane as an igniter. Then the force spikes as the rocket closes the short distance to the force transducer. The remainder of the 15 second test shows a relatively constant thrust of 5 pounds. The figure below shows the results of the theoretical analysis for a grain of similar size and under similar conditions.

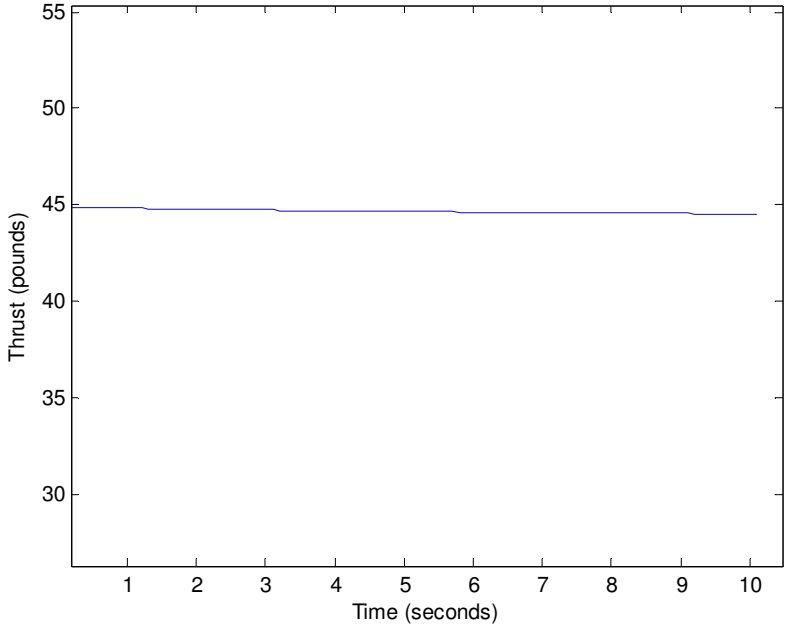

Figure 29: Theoretical Thrust curve

The theoretical output indicates that decreasing thrust levels starting at 45 pounds can be expected from a rocket of this size. This output is significantly different from the data recorded during the experiment. There are many possible reasons for this discrepancy in the data. A significant reason lies in the fact that the theoretical thrust analysis was designed for the nozzles displayed in the Nozzle Design Timeline section of the report. These nozzles have a smooth throat and a fairly shallow turn radius as the flow reaches the nozzle exit. This allows the flow to follow the contour of the nozzle while accelerating. The experimental nozzle, on the other hand, has an extremely sharp angle at the throat and does not have a large area ratio. This would drastically lower the thrust recorded by the experimental rocket. The Matlab analysis also makes a variety of simplifying assumptions that may reduce the accuracy of the data. These assumptions such as a uniform regression rate along the length of the grain can act to reduce the precision of the analysis. Lastly, the Mat lab analysis is programmed to use nitrous oxide as the oxidizer. This is used in the portion that integrates CEA since the oxidizer is an important component of the combustion process. The experimental rocket uses air instead which would result in the lower thrust levels seen in the data. These reasons and others can contribute to the error in thrust levels expected from the rocket. Despite these errors, the thrust trends seen in the experimental data are consistent with past thrust codes which lead us to conclude that qualitatively, there is some validity to the data. Future groups can delve further into this problem to hopefully solve it.

V. Conclusion

This senior design has been a very enjoyable and enlightening experience. After working on this rocket design project for the last six months we have come a long way in our understanding of rocketry, engineering design, and teamwork. We have made mistakes, and learned from them in continuing our design effort and we succeeded in firing the test stand with valid thrust data gathered. Through different iterations of the nozzle design and cooling system design, we embodied the engineering design process. Even though it was sometimes difficult to meet NASA's expectations, we persevered and succeeded in delivering this final project. In working on this project we learned some valuable lessons to pass on to future design groups. Firstly, relevant note taking is extremely important, with dates of meetings and relevant emails included. Making sure that all design requirements are known at the end of the conceptual phase is also important. After the conceptual phase, make sure to constantly review the

design so that everyone is up to speed on what the current iteration is. In working with NASA with changed our design almost every meeting. Being able to iterate small changes without a complete system redesign is a valuable asset. When firing the rocket make sure to be hygienically safe as rodent feces is a constant threat. In the end we designed a system to test a water-cooled rocket with data coming from a variety of sensors. We don't have all of the coolant systems and data acquisition systems assembled yet, but we have established a solid foundation for future graduate and undergraduate students to conduct solid experimentation and analysis.

Appendix A: Tank Design Matlab Code

```
function [L,Tc,Ta,Tp,m1,m2,m3,pr,pT]=tank(r,P,md,t,n,SF,j)

% Calculates the length, thickness, and pipe radius for a carbon fiber
% wrapped titanium tank given tank radius, tank pressure, mass flow rate
% time of flow, and number of pipes. Use j=1 for nitrous and j=2 for oxygen
close all; clc;
sigmac=4137000000; %Pa ultimate tensile strength of carbon fiber
sigmaa=414000000; %Pa ultimate tensile strength of aluminum alloy 2219
Tc=SF*P*r/sigmac; %m thickness of carbon fiber
Ta=SF*P*r/sigmaa/2; %m thickness of aluminum end caps
rhoc=1780; %kg/m3 density of carbon fiber
rhoa=2840; %kg/m3 density of aluminum alloy 2219
rho=[810.25,821.06,830.56,839.10,846.87,854.03
   64.079,73.488,82.947,92.452,102.00,111.57];...
%kg/m^3 density of liquid nitrous and oxygen
if P<5000000
   i=1;
elseif 6000000>P
   i=2;
elseif 6500000>P
   i=3;
elseif 7000000>P
   i=4;
elseif 8000000>P
   i=5;
else
   i=6;
end
V=md*t/rho(j,i); %m^3 volume of tank
L=(V-4/3*pi*r^3)/(pi*r^2); %m length of the tank
A=md/rho(j,i)/sqrt(287*1.4*290)/n; %m^2 area of pipe
pr=sqrt(A/pi); %m radius of pipe
Tp=SF*P*.5283*r/sigmaa; %m thickness of pipe
m1=md*t; %kg mass of oxidizer
m2=1.1*(pi*(r^2-(r-Tc)^2)*L*rhoc+pi*((r-Tc)^2-(r-Tc-.01)^2)*rhoa+...
   4/3*pi*(r^3-(r-Ta)^3)*rhoa); %kg mass of tank
m3=pi*((pr+Tp)^2-pr^2)*.0508*rhoa*n; %kg mass of pipe
%ep=(m-md*t)/m0; % tankage fraction
pT=m3/m2;
end
```

Appendix B: Matlab Trajectory Program

(Attached)

Appendix C: Matlab CEA Setup Program (CEAin)

(Attached)

Appendix D: Matlab CEA Retrieval Program (CEAout)

(Attached)

Appendix E: Matlab CEA Study Program (TableMaker)

(Attached)

Appendix F: Matlab Cooling System Analysis Program (Batch_Pipeloss_Calc)

(Attached)

Appendix G: Final Cooling System Parts Order (NASA)

Appendix H: Instrument Manuals

(attached)

Appendix I: Trajectory Equations

(attached)

References

[1] 2012 Space Propulsion Group, "Hybrid Rockets History", [website] URL: http://www.spg-corp.com/News_12.php

[2] Wikipedia.com, "Hybrid Rocket", [website] URL: http://en.wikipedia.org/wiki/Hybrid_rocket

[3] Gomes, Rocco, Rocco, and Iha, "Evaluation of polyurethane binder additivated with paraffin and tested with a swirl injector", AIAA 2012-4309

[4] Hill, P. and Peterson, C., *Mechanics and Thermodynamics of Propulsion*, 2nd ed, Addison-Wesley, New York, 1992

[5]Subramanian, R., "Heat transfer in Flow Through Conduits", [website] URL: http://web2.clarkson.edu/projects/subramanian/ch330/notes/Heat%20Transfer%20in%20Flow%20Through%20Conduits.pdf

[6] White, Frank M., *Fluid Mechanics,* 7th ed., McGraw-Hill, New York, 2011, Chaps 5,6.

[7] McBride, Bonnie J., and Gordon, Sanford, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications," NASA Reference Publication 1311, 1996.

[8]Omega Engineering, "Omega", [website] URL: http://www.omega.com/

[9] Tovey Engineering, "Tovey Engineering", [website] URL: http://www.testmart.com/webdata/mfr_pdfs/TOVEY/tovey_catalog_07.pdf