Programming Assignment 1: Atmosphere Drag

Zachary Tschirhart

Department of Aerospace Engineering and Engineering Mechanics

ASE 167M (Wed 3:00-4:00)

Lab Partners: Zachary May, Joshua Eboh, and Brian Huber

TA: Noble Hatten

February 27, 2013

Contents

Purpose of Program

The purpose of this program is to calculate the atmospheric properties at a given altitude based on the standard atmosphere model; the functions of altitude calculated are temperature, pressure, density, and the speed of sound. By using these properties, the flight variables can be can be estimated, namely, the drag on the aircraft, at any given altitude. The drag of an aircraft is calculated with values of density and speed of sound as well as three additional known values: the aircraft values of platform area, the drag coefficient, and the velocity relative to the atmosphere.

Mathematical Technique

Drag

The drag of an aircraft at altitude h is shown as,

$$D(h) = \frac{1}{2}C_D(C_L, M)\rho(h)SV^2 \tag{1}$$

where C_D is the coefficient of drag, S is the platform area, and V is the velocity relative to the atmosphere.

The initial and final conditions for altitude, temperature, and pressure that define each atmospheric layer can be found alongside the lapse rates can be found in a table in Figure 1.

Temperature

The temperature of the atmosphere is understood to fluctuate linearly as a function of altitude. Known as the lapse rate, the rate of variation contains different values based on the altitude. Lapse rate is calculated by,

$$\beta = \frac{d\tau}{dh} \tag{2}$$

where dτ is the differential temperature and dh is the differential height, or altitude. Due to the varying lapse rate, the atmosphere must be divided into multiple layers where the lapse rate remains constant or is zero. In layers where temperature is constant, the lapse rate will be zero. Using these lapse rates, the equation for temperature as a function of altitude is,

$$\tau_h = \tau_o + \beta(h - h_o) \tag{3}$$

where τ_o is the initial temperature, h_o is the initial altitude of the layer, and β is the lapse rate of the particular layer.

Perfect Gas Law

The perfect gas law is an equation of state, which demonstrates the relationship between the variables of pressure, density, and temperature of a perfect gas. This law assumes that the temperatures are within a given range, and therefore more error will be introduced at higher altitudes due to low temperatures. Although this equation is ideally used to define the behavior of a perfect gas, it is an effective approximation for air and is given by,

$$p = \rho R \tau \tag{4}$$

where p is the pressure, ρ is the density, R is the specific gas constant, and τ is the temperature.

Hydrostatic equation

The hydrostatic equation is the summation of vertical forces on a differential cube of air when at rest. The equation shows the fluctuation in pressure due to change in height, or altitude, and is expressed as,

$$dp = -\rho g dh \tag{5}$$

where dp is the differential pressure, g is the acceleration of gravity, and dh is the differential height, or altitude. This equation assumes the acceleration of gravity is constant for all altitudes, which is not true, therefore a small error may be introduced.

Pressure equations

In atmosphere layers where temperature remains constant, pressure as a function of height, or altitude, is defined as the quotient of equation 5 divided by equation 6, followed by integration of the result over the boundary conditions, which results in,

$$\int_{p_1}^{p} \frac{dp}{p} = \int_{h_1}^{h} \frac{-\rho g dh}{\rho R \tau(h)} \tag{6}$$

Then solving for pressure,

$$p(h) = \exp \frac{-g}{R\tau}(h - h_1) \tag{7}$$

In atmosphere layers where the lapse rate varies as a function of height, the pressure can be found by inserting equation 2 into equation 6 and integrating results in equation 7.

$$p(h) = p_1 \frac{\tau(h)}{\tau_1} \tag{8}$$

In order to calculate the density as a function of altitude, equation 7 or equation 8 is used in conjunction with equation 4 depending on which lapse rate equation is applicable.

$$\rho(h) = \frac{p(h)}{R\tau_1} \tag{9}$$

Speed of sound and Mach number

The speed of sound, a, through a gas is given by,

$$a(h) = \sqrt{\gamma R \tau(h)} \tag{10}$$

The Mach number M is the ratio between the aircraft's velocity relative to the atmosphere and the speed of sound at the aircraft's altitude given by,

$$M = \frac{V}{a(h)} \tag{11}$$

Program Listing

atmos.m

```
%The function atmos(h) outputs the values of temperature(t),
%pressure(p),density(rho),and speed of sound(sos) at given
%the altitude input(h). This function requires that
%altitude be given in [ft] and the outputs of temperature,
%pressure, density, and speed of sound are given as [R],
%[lb/ft^2], [slug/ft^3], and [ft/sec] respectively.
%
% Atmosphere Generator Function (English Units)
%
% Assumptions:
%   -Gamma is constant and equal to 1.4
%   -Gravity is constant throughout all layers
%
% Input:
%
%   h:   Altitude from mean sea level       (ft)
%
```

```
% Output:
%
%   t:   Tempreture                          (Rankine)
%   p:   Pressure                            (lb/ft^2)
%   rho: Density                             (slug/ft^3)
%   sos: Speed of Sound                      (ft/sec)
%
% Example Command Issue:
%   [t,p,rho,sos] = atmos(30000);

function [t,p,rho,sos]=atmos(h)
%Determine Temperature, Pressure,
while(1)
    %Outside of valid range Error
    assert(h<2000000 | h>0,'Altitude is not valid');

    %Layer 1
    t   = 518.69 - 0.00356616*h;
    p   = (1.137193514E-11)*(t^5.2560613);
    rho = (5.8261E-4)*(p/t);
    sos = 49.0214*t^(1/2);
    if h>=0 & h<=36089.2;break;end

    %Layer 2
    PreLayer = rho*t/p;
    t   = 389.9901385;
    p   = 2678.38681*exp((-4.80624E-5)*h);
```

```
rho = PreLayer*p/t;
sos = 49.0214*t^(1/2);
if h>36089.2 & h<=65616.8;break;end

%Layer 3
PreLayer = rho*t/p;
t   = 353.9901374 + 0.00054864*h;
p   = (3.802075778E90)*(t^-34.1643987);
rho = PreLayer*p/t;
sos = 49.0214*t^(1/2);
if h>65616.8 & h<=104986.9;break;end

%Layer 4
PreLayer = rho*t/p;
t   = 250.3103243 + 0.00153619*h;
p   = (1.44217649E33)*t^-12.20158686;
rho = PreLayer*p/t;
sos = 49.0214*t^(1/2);
if h>104986.9 & h<=154199.5;break;end

%Layer 5
PreLayer = rho*t/p;
t   = 487.1900543;
p   = 873.6400826*exp((-3.84736009E-5)*h);
rho = PreLayer*p/t;
sos = 49.0214*t^(1/2);
if h>154199.5 & h<=170603.7;break;end
```

```
    %Layer 6
    PreLayer = rho*t/p;
    t    = 674.3900821 - 0.00109728*h;
    p    = (1.509860827E-46)*t^(17.08219937);
    rho = PreLayer*p/t;
    sos = 49.0214*t^(1/2);
    if h>170603.7 & h<=200131.2;break;end

    %Layer 7
    PreLayer = rho*t/p;
    t    = 893.9900454 - (2.19456E-3)*h;
    p    = (7.577930948E-24)*t^8.541099698;
    rho = PreLayer*p/t;
    sos = 49.0214*t^(1/2);
    if h>200131.2 & h<=259186.0;break;end

    %Layer 8
    PreLayer = rho*t/p;
    t    = 325.1908172;
    p    = (6.669501806E4)*exp(-5.763986782E-5*h);
    rho = PreLayer*p/t;
    sos = 49.0214*t^(1/2);
    if h>259186.0 & h<=2000000;break;end
end


end
```

atmosplot.m

```
function [property] = atmosplot()
%The atomsplot function calculates atmosphereic
%properties and plots them
clc;clear all;

i = 1;

% Load the Standard Atmosphere reference values
stdAtm = load('StandardAtmos.mat');
stdAtm = stdAtm.stdAtmFH;

for height = stdAtm(:,1)'
 [t,p,rho,sos]=atmos(height);
 property(i,:)=[height,t,p,rho,sos];
 i=i+1;
end

% calculates percent error
ERROR=abs((property-stdAtm)./stdAtm.*100);

%Plot t, p, rho, and sos
figure(1);
plot(property(:,2),property(:,1));
title ('Altitude vs. Temperature ');
xlabel('Temperature [R]');
```

```
ylabel('Altitude [ft]');

figure(2);
plot(property(:,3),property(:,1));
title ('Altitude vs. Pressure');
xlabel('Pressure [lb/ft^2]');
ylabel('Altitude [ft]');

figure(3);
plot(property(:,4),property(:,1));
title( 'Altitude vs. Density');
xlabel('Density [slug/ft^3]');
ylabel('Altitude [ft]');

figure(4);
plot(property(:,5),property(:,1));
title( 'Altitude vs. Speed of Sound');
xlabel('Speed of Sound [ft/s]');
ylabel('Altitude [ft]');

%Plot Error of t, p, rho, and sos
figure(5);
plot(ERROR(:,2),property(:,1));
title ('Altitude vs. Temperature Error');
xlabel('Percent error [%]');
ylabel('Altitude [ft]');
```

```
figure(6);
plot(ERROR(:,3),property(:,1));
title ('Altitude vs. Pressure Error');
xlabel('Percent error [%]');
ylabel('Altitude [ft]');

figure(7);
plot(ERROR(:,4),property(:,1));
title ('Altitude vs. Density Error');
xlabel('Percent error [%]');
ylabel('Altitude [ft]');

figure(8);
plot(ERROR(:,5),property(:,1));
title ('Altitude vs. Speed of Sound Error');
xlabel('Percent error [%]');
ylabel('Altitude [ft]');

end
```

Data Runs and Test Cases

I/O Results

Executing the atmos.m function with an input altitude calculates and returns the values of temperature, pressure, density, and speed of sound required to perform further calculations at that specific altitude. In order to test the atmosphere model, two examples were calculated in the following sub sections.

Space Shuttle

The space shuttle orbiter is flying bottom forward at an altitude of 38 nautical miles. The cross-sectional area of the orbiter is assumed to be S = 5200 ft^2. Under these conditions, the flow near the shuttle can be considered Newtonian flow. What is the drag force acting on the shuttle if it is in a circular orbit? What would the drag force be if the orbiter were flying airplane style and the cross-sectional area was S = 400 ft^2?

First, the perpendicular velocity of a circular orbit at an altitude of 38 nautical miles is found by:

$$V = \sqrt{\frac{\mu}{r_e + h}} = \sqrt{\frac{8.132 * 10^{11} \frac{nauticalmiles^3}{hour^2}}{3443.92 + 38 nauticalmiles}} = 1.528 * 10^4 \frac{nauticalmiles}{hour} \tag{12}$$

Second, a coefficient of drag needs to be defined for the space shuttle. Approximate values were found using a paper on the aerodynamics of reentry of the space shuttle (Flight-Determined Subsonic Lift and Drag Characteristics of Seven Lifting-Body and Wing-Body Reentry Vehicle Configurations With Truncated Bases, *Edwin J. Saltzman*). According to their measurements, a C_D was found to be 0.078. Then using the atmosphere model at a height of

230892 feet, a ρ_∞ was found to be $1.451 * 10^{-07} \frac{slugs}{ft^3}$

Solving for Drag using Equation 1 and using $S = 5200 ft^2$ results in:

$$\begin{aligned}
D(h = 230890) = \frac{1}{2} C_D \rho(h) S V^2 = \\
\frac{1}{2}(0.078)(1.451 * 10^{-7} \frac{slugs}{ft^3})(1.528 * 10^4 \frac{nauticalmiles}{hour})^2 \\
(6076.12 \frac{ft}{nauticalmile})^2 (\frac{1}{3600} \frac{hour}{seconds})^2 (5200 ft^2) = \\
19571.754 lbs.
\end{aligned} \tag{13}$$

Then, solving for Drag using Equation 1 and using $S = 400 ft^2$ results in:

$$\begin{aligned}
D(h = 230890) = \frac{1}{2} C_D \rho(h) S V^2 = \\
\frac{1}{2}(0.078)(1.451 * 10^{-7} \frac{slugs}{ft^3})(1.528 * 10^4 \frac{nauticalmiles}{hour})^2 \\
(6076.12 \frac{ft}{nauticalmile})^2 (\frac{1}{3600} \frac{hour}{seconds})^2 (400 ft^2) = \\
1505.519 lbs.
\end{aligned} \tag{14}$$

Standard Drag

What would be the drag force acting on an aircraft flying at h = 30000ft altitude at an airspeed of V = 400knots if its drag coefficient is 0.05? (Use S = $600 ft^2$)

First, the density must be found at a height of 30000 feet from the standard atmosphere model, which comes out to be $411.705 \frac{slugs}{ft^3}$. Then convert

400 knots to $675.124\frac{ft}{second}$. Lastly, substitute the numbers into Equation 1.

$$D(h = 30000) = \frac{1}{2}C_D\rho(h)SV^2 = \frac{1}{2}(0.05)(8.893 * 10^{-4}\frac{slugs}{ft^3})(675.124\frac{ft}{second})^2(600ft^2) = 6080.042lbs. \tag{15}$$

Plots and Discussion

When graphically shown, the relation between temperature and speed of sound is quite obvious, as speed of sound is only a function of temperature in a uniform gas. Pressure and density also denote the traits of an exponential function and follow the same trend as density is calculated as a function of pressure. In the function atmosplot.m, the function atmos.m is called iteratively over each range specified in the standard model .mat file provided. In order to test a larger range of data, the standard .mat file was expanded to include altitudes from 0 to up to 2 million feet using the U.S. Standard Atmosphere as a reference. A plot of each property relative to the accepted values was plotted using the relative error for each of the four properties. The four property plots are found in figures 2 through 5 and the error plots in figures 6 through 9.

Bibliography

[1] Eduardo Gilden, Greg Holt, Kyle DeMars, George Jacobellis *ASE 167M Flight Dynamics Laboratory Flight Simulator Experiments and Computer Projects. s.l. : The University of Texas at Austin Department of Aerospace Engineering* 2012.

[2] Edwin J. Saltzman, K. Charles Wang, Kenneth W. Iliff *Flight-Determined Subsonic Lift and Drag Characteristics of Seven Lifting-Body and Wing-Body Reentry Vehicle Configurations With Truncated Bases* 1999.

[3] National Oceanic And Atmospheric Administration National Aeronautics And Space Administration United States Air Force *U.S. Standard Atmosphere* 1976.

Appendix

Layer	Height above Sea Level		Static pressure		Temperature		Temperature Lapse Rate	
	(m)	(ft)	(pascals	(lbf/ft^2)	(°K)	(°R)	(°K/m)	(°R/ft)
1	0	0	101325	2116.2132	288.15	518.67	-0.0065	-0.003566
2	11,000	36,089	22632.1	472.68046	216.65	389.97	0	0
3	20,000	65,617	5474.89	114.34527	216.65	389.97	0.001	0.0005486
4	32,000	104,987	868.019	18.128924	228.65	411.57	0.0028	0.0015362
5	47,000	154,199	110.906	2.3163162	270.65	487.17	0	0
6	51,000	167,323	66.9389	1.3980457	270.65	487.17	-0.0028	-0.001536
7	71,000	232,940	3.95642	0.0826314	214.65	386.37	-0.002	-0.001097
	84,852	278,386	0.3734	0.0077986	186.95	336.51	-	-

Figure 1: The atmospheric table listing U.S. Standard Atmosphere properties

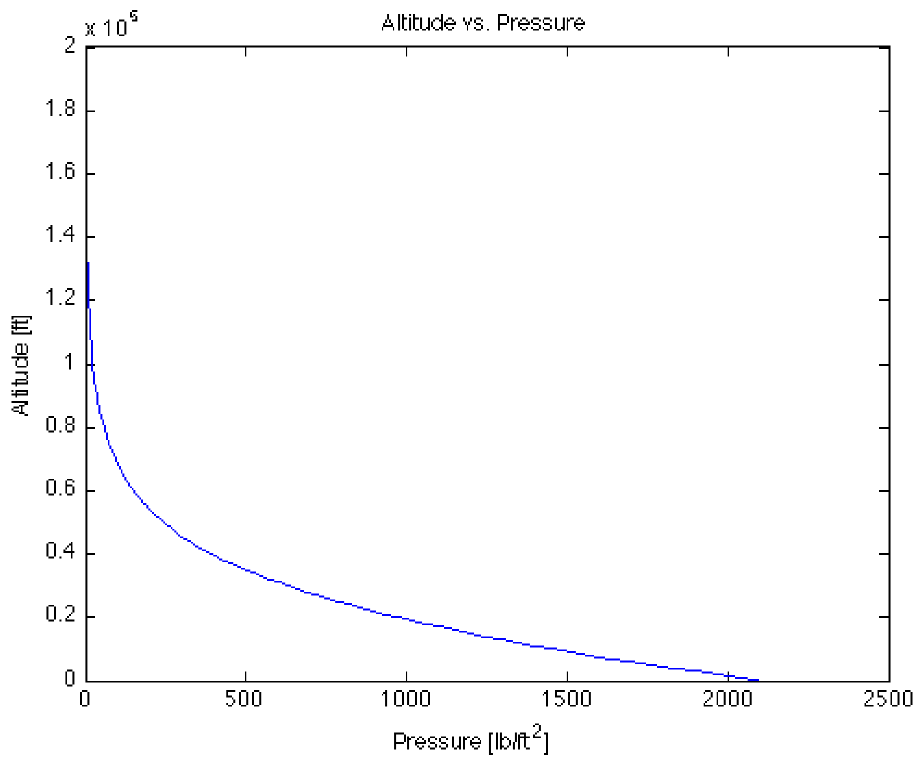

Figure 2: Altitude versus Pressure

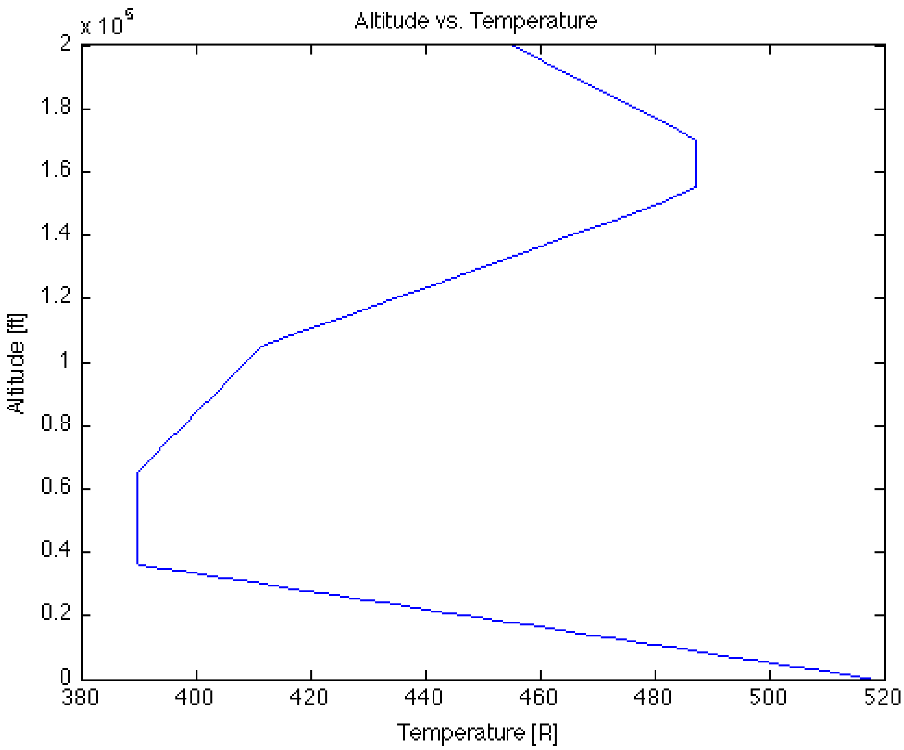

Figure 3: Altitude versus Temperature

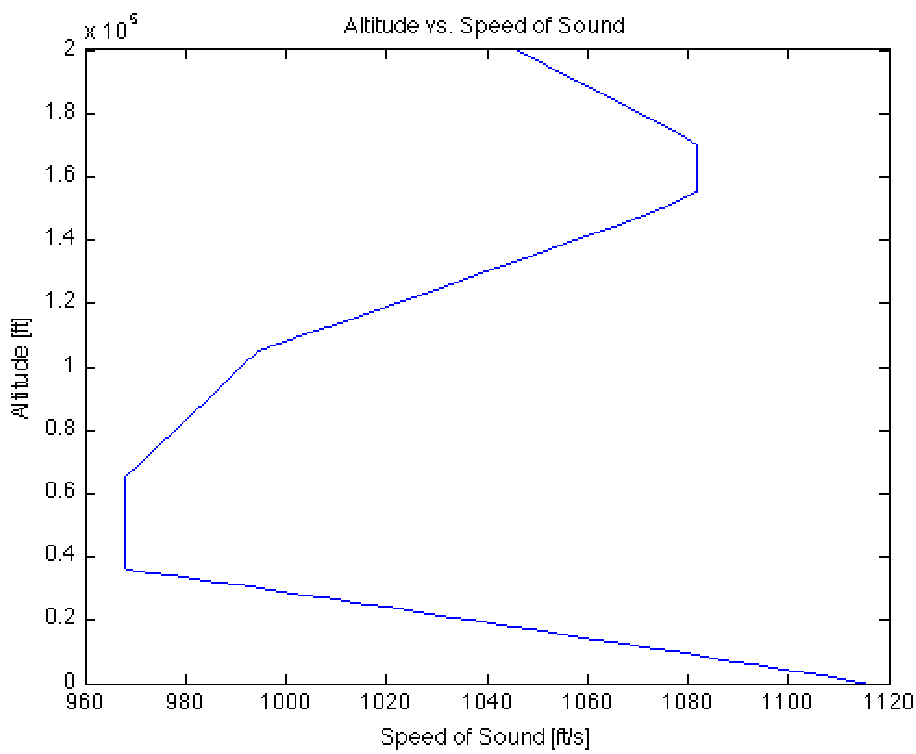

Figure 4: Altitude versus Speed Of Sound

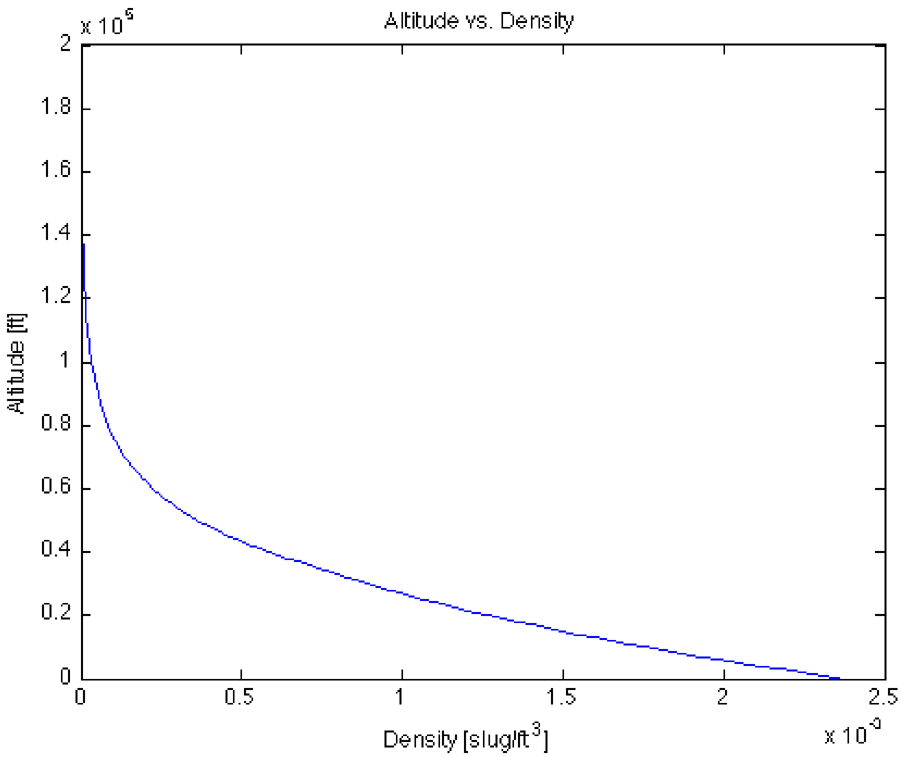

Figure 5: Altitude versus Density

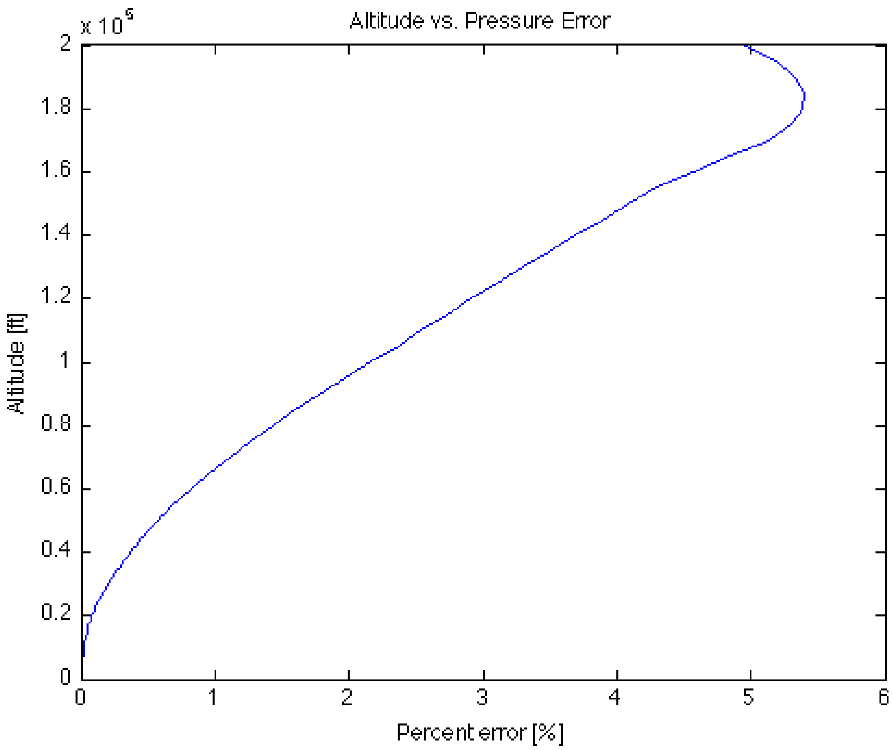

Figure 6: Altitude versus Pressure Error

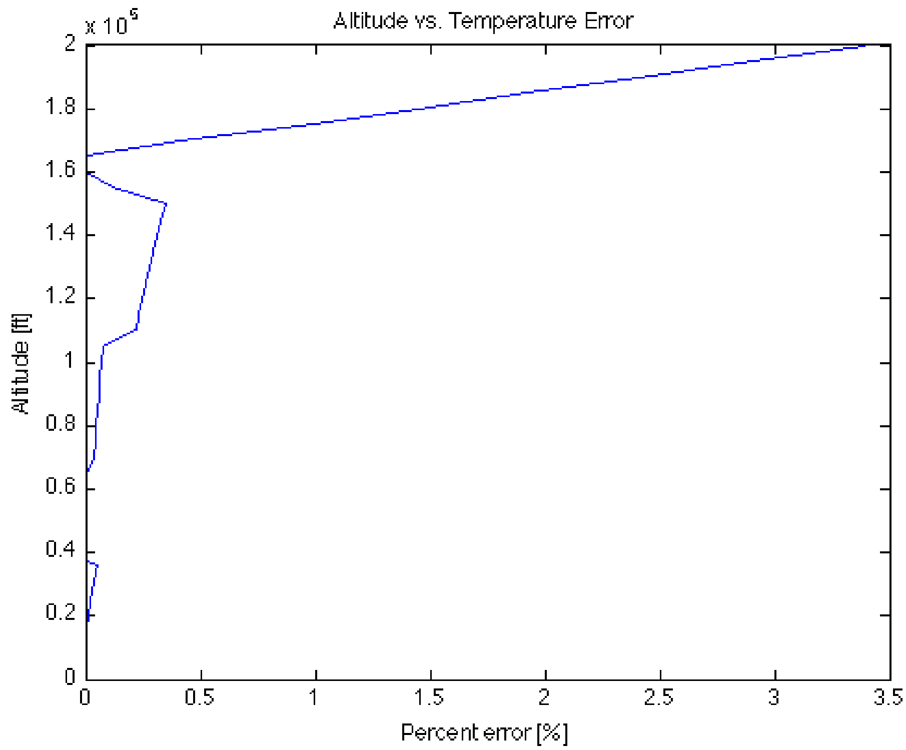

Figure 7: Altitude versus Temperature Error

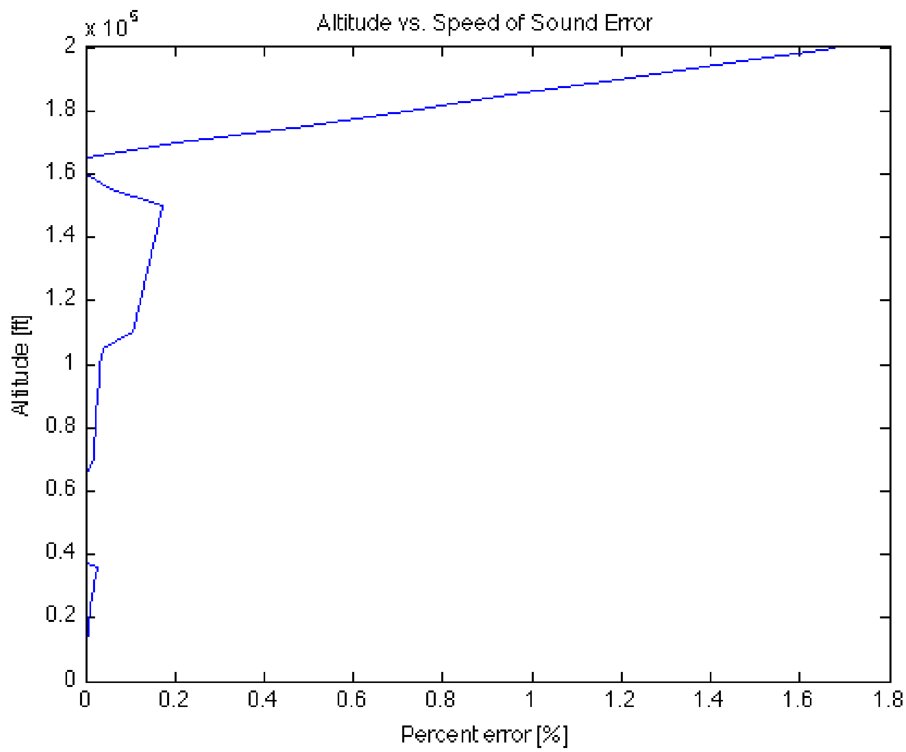

Figure 8: Altitude versus Speed Of Sound Error

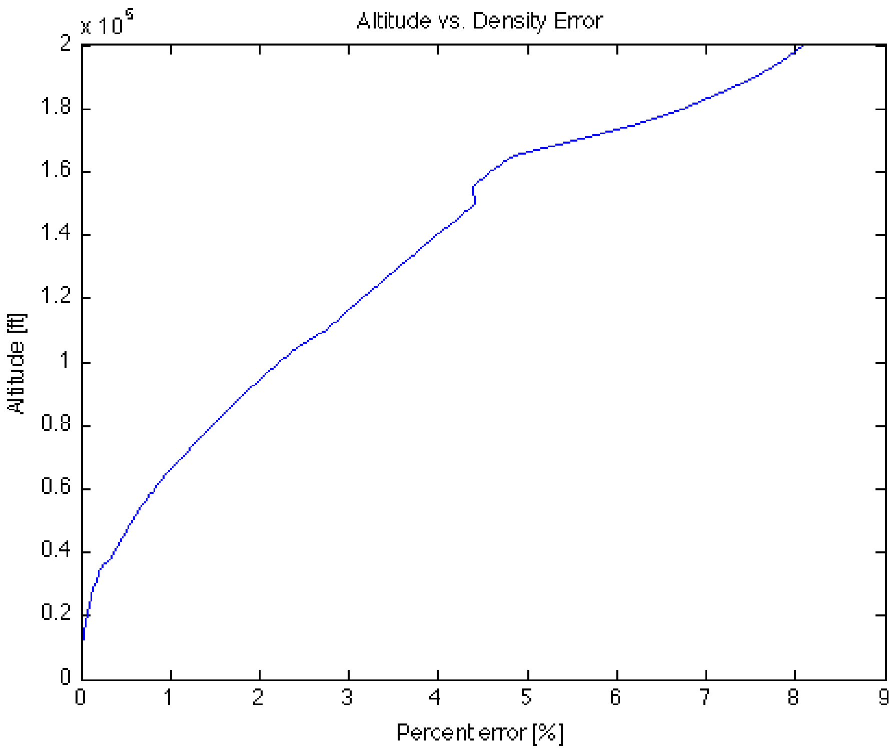

Figure 9: Altitude versus Density Error