REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	2 May 2003	

4. TITLE AND SUBTITLE
Determination of atmospheric density in low-Earth orbit using GPS data from the USNA satellite

5. FUNDING NUMBERS

6. AUTHOR(S)
Groenenboom, Katherine E. (Katherine Elise), 1981-

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PE[illegible]

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
**US Naval Academy
Annapolis, MD 21402**

10. [illegible]
Trident Scholar project report no. 308 (2003)

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
This document has been approved for public release; its distribution is UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT: The goal of this project was to use real time data from the US Naval Academy's Prototype Communications Satellite (PC-Sat) to calculate atmospheric density in the satellite's orbit over various time intervals. This involved using the latitude, longitude, altitude, and time data from a GPS receiver on board PC-Sat and transforming them into the orbiter's classical orbital elements (COEs). From these, the change in the size of the orbit can be determined via the change in the semi-major axis. Changes to the orbit are due primarily to the non-spherical Earth and the atmosphere. Therefore, by accounting for the change in semi- major axis due to the non-spherical Earth, the researcher can conclude that the remaining change is due solely to atmospheric density. The ability to determine atmospheric density in a specific orbit by knowing only the position of the satellite and few characteristics of the satellite itself will allow many small satellites with GPS receivers to contribute to the collection of data about the upper atmosphere. Being able to measure the Earth's atmospheric density with increased accuracy will then allow satellite orbit and fuel usage predictions to be much more accurate and has the potential to lower the cost of missions.

14. SUBJECT TERMS:
Atmospheric density; Atmospheric drag; GPS

15. NUMBER OF PAGES
73

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

U.S.N.A. --- Trident Scholar project report; no. 308 (2003)

DETERMINATION OF ATMOSPHERIC DENSITY IN LOW-EARTH ORBIT USING GPS DATA FROM THE USNA SATELLITE

Midshipman Katherine E. Groenenboom
United States Naval Academy
Annapolis, Maryland

_____
(signature)

Certification of Advisers' Approval

Dr. Richard P. Fahey
Department of Aerospace Engineering

_____
(signature)

_____
(date)

Professor Daryl G. Boden
Department of Aerospace Engineering

_____
(signature)

_____
(date)

Acceptance for the Trident Scholar Committee

Professor Joyce E. Shade
Deputy Director of Research & Scholarship

_____
(signature)

_____
(date)

USNA-1531-2

Abstract

The goal of this project was to use real time data from the US Naval Academy's Prototype Communications Satellite (PC-Sat) to calculate atmospheric density in the satellite's orbit over various time intervals. This involved using the latitude, longitude, altitude, and time data from a GPS receiver on board PC-Sat and transforming them into the orbiter's classical orbital elements (COEs). From these, the change in the size of the orbit can be determined via the change in the semi-major axis. Changes to the orbit are due primarily to the non-spherical Earth and the atmosphere. Therefore, by accounting for the change in semi-major axis due to the non-spherical Earth, the researcher can conclude that the remaining change is due solely to atmospheric density.

The ability to determine atmospheric density in a specific orbit by knowing only the position of the satellite and few characteristics of the satellite itself will allow many small satellites with GPS receivers to contribute to the collection of data about the upper atmosphere. Being able to measure the Earth's atmospheric density with increased accuracy will then allow satellite orbit and fuel usage predictions to be much more accurate and has the potential to lower the cost of missions.

Keywords:

Atmospheric density
Atmospheric drag
GPS

Acknowledgements

I would like to thank Dr. Richard Fahey for his encouragement and patience with me throughout this project. I would also like to thank Dr. Daryl Boden for his input and enthusiasm for my project. Thank you to CDR Bruninga for helping me with the PC-Sat aspect of the project. Dr. Pisacane, CDR Hailey, and Dr. Parise all had valuable suggestions along the way to get me on track or to help me see a different side of it. Mr. Leung from DLR was very helpful in understanding the capabilities of the GPS receiver and his analysis of the receiver's performance. Without the help of all of these wonderful people, I would not have been able to complete my research, nor had as much fun as I did.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Introduction

The Naval Academy launched its first midshipman-designed and built spacecraft on September 30, 2001 from Kodiak, Alaska. The satellite is Prototype Communications Satellite, (PC-Sat), and its main mission is to "provide mobile and handheld satellite digital communications for amateur satellite operators worldwide using the Automatic-Position-Reporting-System (APRS)."[1] The satellite also has a Global Positioning System (GPS) receiver on board that transmits the satellite's position and time.

MIDN John Young completed a Trident project in 2001 in which he developed an algorithm to calculate atmospheric density using real-time GPS data. His goal was to determine a method for calculating the density of the atmosphere at a specific point at a specific time in a single satellite's orbit. As more satellites with GPS receivers are used, a more accurate model of atmospheric density can thus be determined.

The atmospheric density over several periods of time in the orbit of PC-Sat has been determined using the GPS data received and the algorithm developed by MIDN Young. The GPS data are received as latitude, longitude, altitude, and time in the Earth-Centered, Earth-Fixed (ECEF) rotating reference frame of the Earth. The satellite is most easily described in a reference frame which is inertial with respect to the center of the Earth and the stars. Therefore the position in the rotating reference frame must be transformed into the inertial reference frame using rotation matrices. Once the position is known in the inertial reference frame, the velocity vector of the satellite must be calculated for each position vector. The position and velocity vectors at each time can be used to calculate the classical orbital elements of the satellite at each time. The challenge to the researcher lies in finding the velocity vectors of the satellite from only the position vectors.

Until recently, GPS receivers have not been very common on spacecraft. Most users have not had a need for the accuracy available with GPS for use with space operations. The idea of using GPS onboard spacecraft is somewhat novel, and this is among the first research to use GPS for a purpose other than positioning.

Background

PC-Sat

PC-Sat is a 10-inch cube with solar panels on all six faces. Five of the six faces use off-the-shelf solar panels. The sixth face has a solar panel designed for space use, which failed shortly after launch. This severely limits the amount of power PC-Sat collects while it is in the sunlight, especially when the failed solar panel is facing towards the sun. Batteries must be used while the satellite is not in view of the sun and when the satellite is using more energy than it is producing from the solar arrays. The coordinate system corresponding to PC-Sat is shown in Figure 1.

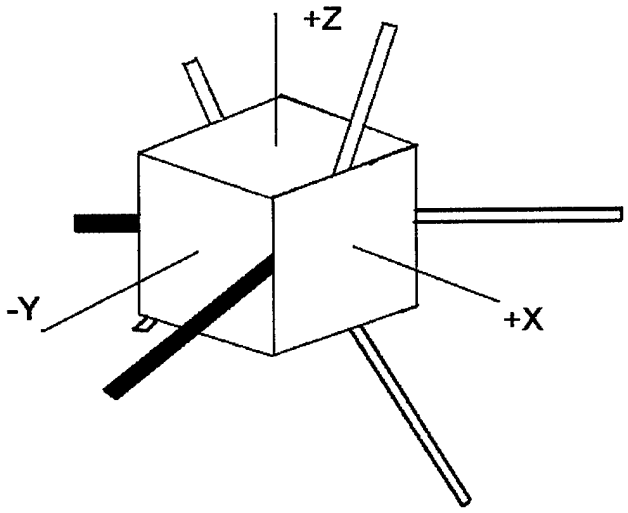

Figure 1: PC-Sat coordinate system

The orbital period of PC-Sat is approximately 100 minutes. During its orbit it spends a maximum of 35 minutes in eclipse and a minimum of 65 minutes in the sun. Figure 2 shows the eclipse times in minutes from October 2001 to May 2002.

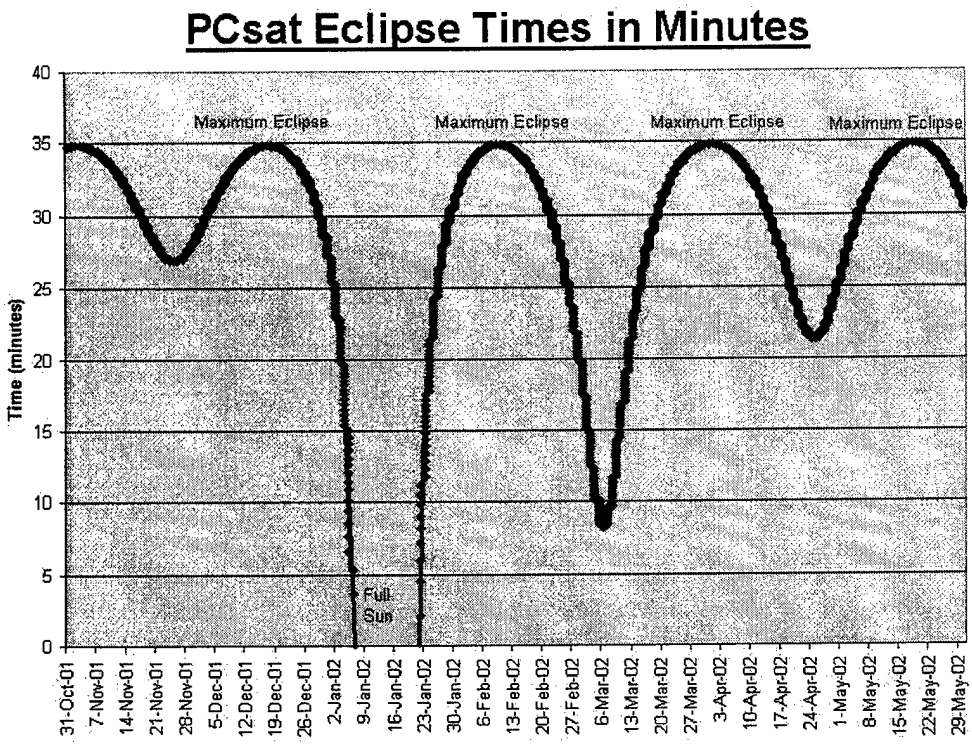

Figure 2: PC-Sat eclipse times for November 2001-May 2002

The high inclination of PC-Sat's orbit (67 degrees) and the angle of Earth's spin axis sometimes allow PC-Sat to be in the Sun for an entire orbit for several days, the first occurrence of which was January 10-21, 2002. During these twelve days of continuous sun, PC-Sat had enough power to allow the GPS receiver to remain on continuously. PC-Sat does not have memory on board and cannot store data to transmit once it passes over a ground station. Therefore it transmits telemetry continuously, but only that which is received by a ground station is recorded. With the help of ground stations in South Africa and Great Britain, PC-Sat's position according to the GPS receiver was recorded during that period.

The data received from the GPS receiver on PC-Sat comes in the "GGA" format shown in Figure 3. It gives the time the position was taken, the latitude, longitude, and altitude of the satellite, and the number of satellites tracked.

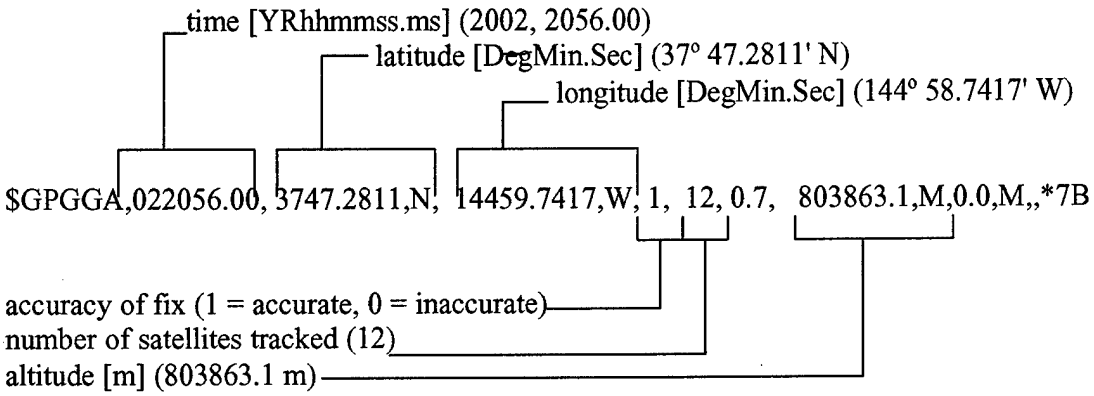

Figure 3: GPS format

Orbits

An orbit is an ellipse, as was discovered by Kepler in the 1600s. An ellipse is one of the curves in the family of conic sections. Figure 4 shows the various curves created by the intersection of a plane and a right circular cone, all of which are possible orbits. The elliptical orbit is most common, the circular orbit being a special case of the elliptical orbit. Parabolic and hyperbolic trajectories also exist, although bodies in these orbits only pass by once.

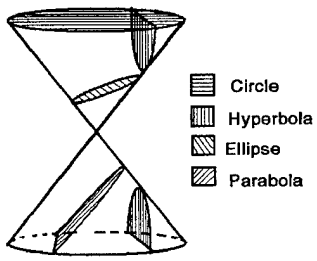

Figure 4: Family of conic sections; orbital paths

Ellipses have foci as shown in Figure 5. For a satellite in orbit about Earth, Earth is at one focus and the other focus is vacant. Perigee is the point of the ellipse that comes closest to the Earth. The apogee is the point farthest from the Earth.

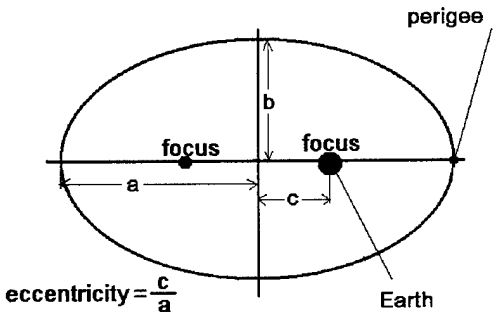

Figure 5: Parts of an elliptical orbit

A satellite's orbit is defined by six classical orbital elements (COEs), much as a position on the earth is described by latitude and longitude. The COEs are in the inertial reference frame with respect to the stars, meaning the coordinate system does not appear to move relative to the stars. In defining this coordinate system, the first step is to determine the first point of aries, ?, which is the line from the Earth to the Sun on the first day of spring. The COEs are based on the first point of aries. Figure 6 illustrates these elements.[2]

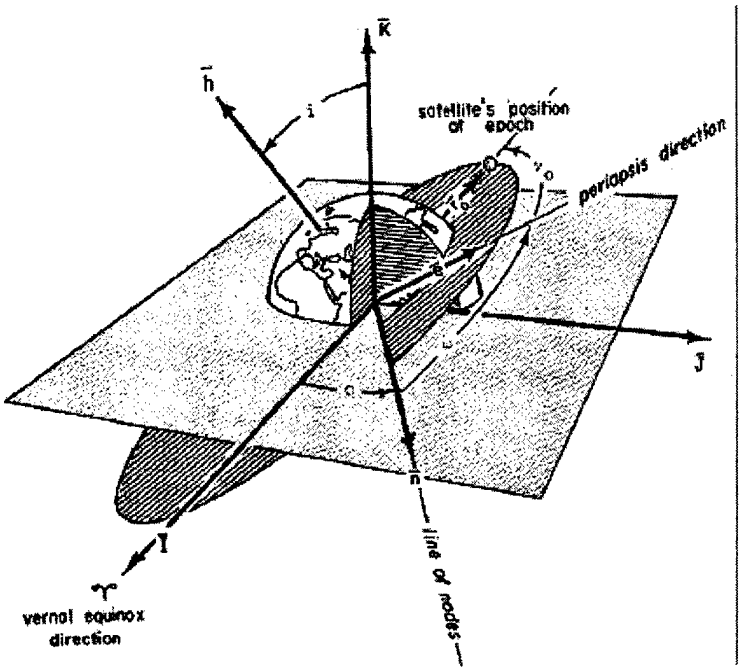

Figure 6: Classical Orbital Elements

Semi-major axis, a, is the length of the long axis of the ellipse, and also the mean radius of the path. The eccentricity, e, is the variance of an orbit from a perfect circle, and a measure of the difference between the long and short axes of the ellipse. If the semi-major axis and semi-minor axis, b, are the same length, the orbit is circular and the eccentricity is zero. As the semi-major axis increases relative to the semi-minor axis, the orbit becomes more elliptical and the eccentricity increases. The eccentricity of an ellipse is between zero and one. When the eccentricity equals 1, the orbit is the special case of a parabola. A hyperbola has an eccentricity greater than 1.

The orbit of a satellite does not necessarily lie in the plane of the Earth's equator, and inclination, i, is the angular measurement from the equatorial plane to the plane of the orbit. An orbit is also defined by the longitude at which the plane of the orbit and the equatorial plane cross. This actually occurs at two points, so the crossing is defined as the longitude of the

ascending node (O) and is the longitude at which the satellite ascends from the southern hemisphere to the northern hemisphere. The angle from this line to the point of perigee is called the argument of perigee (?). The position of the spacecraft along the orbit from the point of perigee is the true anomaly (?) and is measured as the angle between the vectors connecting the center of the earth to perigee and the spacecraft. These six elements, a, e, i, Ω, ω, and ν define a single orbit and the satellite's position in that orbit. Table 1 shows the approximate values of the COEs for PC-Sat as taken from NORAD.

Table 1: PC-Sat Classical Orbital Elements

COE	value
a	7178 km
e	0.0005
i	67 deg
Ω	varies (~210 deg in Jan 2002)
ω	varies (~275 deg in Jan 2002)
ν	varies with time

GPS

The Global Positioning System (GPS) is made up of 3 segments: the space segment, control segment, and user segment. The space segment is a constellation of 24 satellites in semi-synchronous orbit, meaning they each orbit the Earth twice a day. From any point on the surface of the Earth, at least 4 satellites are in view at any time. This means that a receiver on Earth, barring possible blockage by local obstructions, can determine its position anywhere. Each GPS satellite transmits 2 microwave carrier signals with a navigation message (time corrections, space vehicle orbital data sets, etc.) and code signals (the fingerprint of each space vehicle). The control segment is the ground network that tracks the GPS satellites and corrects the clocks on each satellite to ensure very precise location measurements. The user segment consists of the

GPS receivers and the user community. The GPS receiver uses the time difference between the times sent from the satellite and the receiver time to calculate what is known as the pseudo-range, the distance from each individual satellite to the GPS receiver. A receiver's position is computed using triangulation from four pseudo-ranges from four different GPS satellites. Figure 7 shows the geometry of the satellites relative to the receiver, the lines connecting each showing the pseudo-range.[3] As more satellite signals are processed and as the angles between the satellites increase, the position determined by the GPS receiver is more accurate.

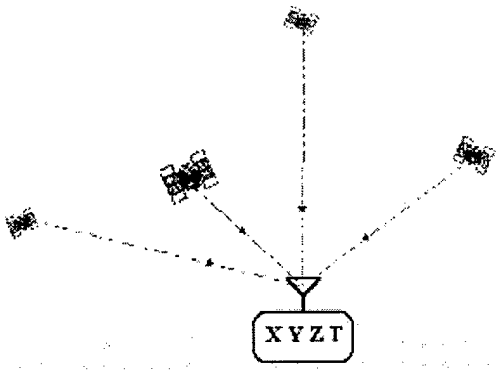

Figure 7: Optimum geometry of GPS satellites relative to receiver for position fix

For a receiver on the ground, only satellites that are above the horizon are in view and are available for determining position. The actual number in view is a minimum of 4, but could be 5 or 6 satellites. At 800 km, however, there is an average of thirteen satellites in view at any given time. The receiver on PC-Sat can track only twelve of the satellites in view, and on average tracked ten to eleven satellites at any given time.[4]

The positions from GPS receivers are provided in latitude, longitude, and altitude above the geoid Earth. The Earth is not a perfect sphere; it is wider at the equator than it is between the

poles. It also has a slight egg-shape to it. Therefore, the position on a geoid earth is more accurate than the position on a spherical earth. World Geodetic System 1984 (WGS-84) is the datum or mathematical model of the Earth's shape used by GPS.[5] This is also the datum used for maps used every day, and thus makes the conversion from GPS position to a map position very simple.

GPS Receiver

Most GPS receivers are designed for ground use, where the motion between the receiver and the GPS satellites is relatively small. The velocities of these terrestrial GPS receivers with reference to the Earth are also very small. For a spacecraft traveling at 7-8 km/s, however, the relative motion between the GPS satellites and the GPS receiver becomes an issue. "The DLR Orion GPS Receiver [onboard PC-Sat] was originally designed for terrestrial applications, but has received numerous modifications to provide accurate and reliable tracking under highly varying signal dynamics encountered in space applications."[6] This modification includes an orbit propagator in the receiver that predicts the satellite's position before the GPS signals are received. From this estimated position, the Doppler shift of the signals as observed by the receiver is calculated. Knowing the expected shift in frequencies allows the receiver to track the specific satellites in view rather than searching through the wide Doppler frequency band for all the GPS satellites.[7] For a GPS satellite directly above the receiver relative to the center of the Earth, there is no Doppler shift. It is well known within the GPS user community that using satellites directly overhead for a fix have poor altitude accuracies. The Orion receiver does not discard signals from satellites directly overhead, and could have some error in the calculation of altitude.

The Inter-agency GPS Executive Board (IGEB) published the US Government's standard precision of GPS in 2001, after the removal of Selective Availability. They publish a horizontal accuracy of 13 meters, and a vertical accuracy of 22 meters.[8] Since the satellite is in space, its receiver does not experience as much refraction of the atmosphere to cause errors, and theoretically should have equal or better accuracy. The receiver has a mask angle of –15 degrees, meaning it does not look for satellites that are below 15 degrees of the horizon of the antenna. The manufacturer of the Orion GPS receiver confirmed the accuracy of the receiver. In testing the receiver before it was installed on PC-Sat, the receiver had an accuracy of 3.4 meters in position and 0.8 m/s in velocity.[9]

The antenna used on PC-Sat to receive the GPS signals is a small monopole, quarter-wavelength antenna located on one corner of PC-Sat. The altitude of the orbit in which PC-Sat is located is far below the altitude of the GPS transmitting satellites, and is therefore still within the constellation of GPS satellites. NASA has successfully developed a GPS receiver, named PiVoT, for use in highly elliptical orbits and orbits that move beyond the orbits of the GPS satellite.[10] This receiver has received numerous improvements for detecting very weak and highly varying signals. The problems encountered when using GPS in this higher orbit are much greater than those experienced at the 800 km altitude of PC-Sat. Although a receiver for space use does require some modifications, the receiver that is on board PC-Sat does not need the same rigorous changes that PiVoT does.

Based on the testing of the Orion receiver, the GPS data returned should have been highly accurate and reliable. The tumbling of the satellite, however, does limit some reception to the single receiving antenna. The antenna was placed on the corner so that it would have the least amount of area blocking it from incoming signals. Shortly after launch the GPS was turned on,

and the receiver was unable to get a single position. Several months later, the receiver was turned on again and data were taken over 12 days. The difference between the two data-gathering attempts was the tumbling of the satellite. Between the on-board magnets aligning the z-axis with the Earth's magnetic field and the painted antennas regulating the speed of rotation, PC-Sat settled into a slower tumble. This allowed the GPS receiver to pick up GPS signals and predict forward to the next expected signal frequency. As the satellite tumbles, however, it still could lose one satellite, pick up a new satellite, and not realize what occurred. The receiver would report an inaccurate position, but label it as accurate. Figure 3 shows how the accuracy of the fix is reported. There are several points in the GPS data where it appears this has occurred. The positions of the satellite track along an apparent orbit, one to twelve consecutive positions appear drastically out of place, and then the positions return to the expected orbit. In the 12 days of data and over 1600 positions reported, this occurs only 7 apparent times. Figure 8 shows one example of the outliers. The points that are stars should be between the squares as expected for a nearly circular orbit

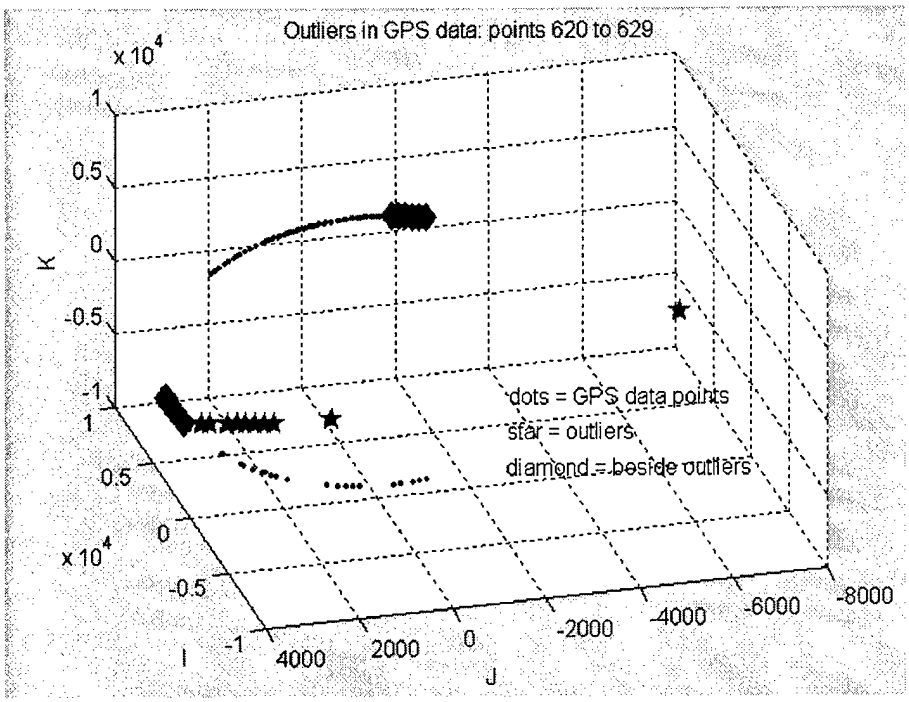

Figure 8: Outliers in GPS data

NORAD

"The North American Aerospace Defense Command (NORAD) is a bi-national United States and Canadian organization charged with the missions of aerospace warning and aerospace control for North America. Aerospace warning includes the monitoring of man-made objects in space."[11] "The Space Control Center (SCC) supports the US Space Command's, (USSPACECOM's), missions of surveillance and protection in space. The center's primary objective in performing the surveillance mission is to detect, track, identify, and catalog all man-made objects orbiting the earth. The SCC currently tracks over 8,300 objects including payloads, rocket bodies and debris. Knowing where these objects are contributes to several mission areas, including collision avoidance for the space shuttle crew."[12]

NORAD is able to track these Earth satellites using a system called Space Detection and Tracking System (SPADATS). This system is "capable of detecting and tracking space vehicles from earth and reporting the orbital characteristics of these vehicles to a central control facility."[13] SPADATS is made up of two parts: US Navy's Space Surveillance System (SPASUR) and the US Air Force's SPACETRACK system. SPASUR detects and determines the orbital elements of man-made objects orbiting Earth by using an electronic fence, a continuous wave of energy beamed vertically across the continental United States. This 'fence' stretches from Georgia to California, 1,600 km off each coast, and 24,100 km up into space. It uses bistatic radar made up of three transmitters (each up to two miles in length) and six receivers. The receivers are located separately from the transmitters and receive signals from the transmitters that have reflected off of an object in space. Observations from two or more receivers are used to calculate the position of the object.[14] "SPACETRACK consists of a worldwide network of radars, space-probing cameras, and communications. An operational control center with a central data-processing facility called the Space Defense Center, located at NORAD headquarters, serves to integrate the entire network of both Navy and Air Force information."[15] The result of all these transmitters, receivers, radars, cameras, and communications is a sophisticated satellite tracking system. NORAD then publishes the orbital information of each object on the World Wide Web in the 2-line element format. Figure 9 shows a sample 2-line element set and what the different numbers represent.

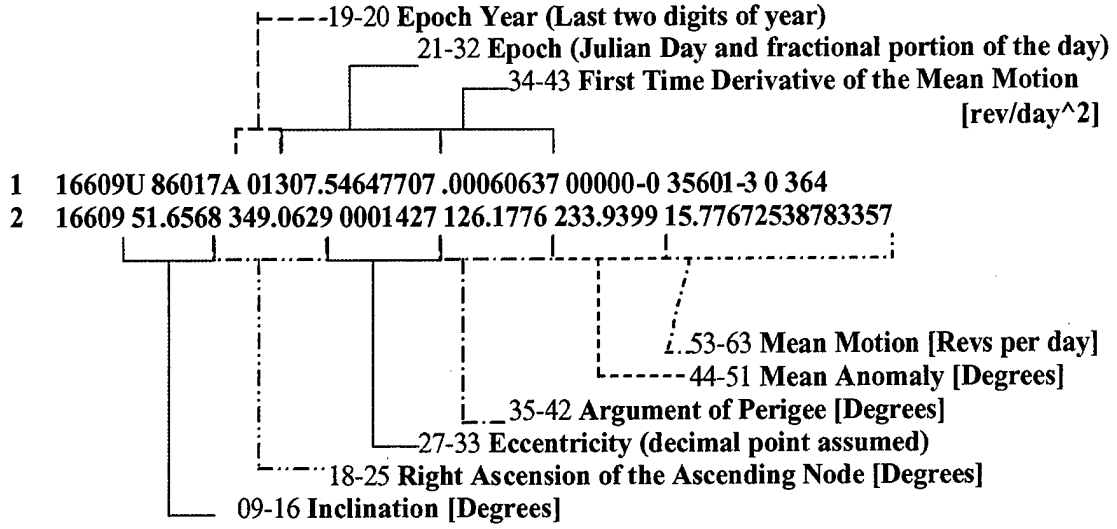

Figure 9: Sample NORAD 2-line elements

PC-Sat Details

PC-Sat has no propulsion system on board and is free to rotate. It uses metal tape measures of different lengths as antennas for the different downlink and uplink frequencies. Those that are in the x-y plane are painted white on one side and black on the other to allow solar pressure to create rotation about the z-axis. As photons from the sun hit the antennas that are painted black, they are absorbed by the antenna and momentum transfer occurs. When the photons hit the white sides of the antennas, they are reflected, and the antenna receives a momentum exchange of twice what the black antenna receives. The greater torque on the white antenna causes the spacecraft to spin. Figure 10 shows the rotation about the z-axis created by pressure in the x- and y-axes.

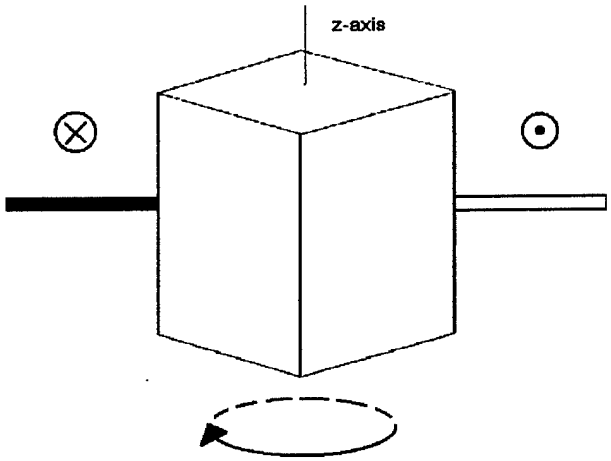

Figure 10: Rotation of PC-Sat about the z-axis

Theoretically the spacecraft would continue to speed up indefinitely. However, PC-Sat is operating in the Earth's magnetic field. As a conductor is moved through a magnetic field, a voltage is induced in the conductor. The current which is produced creates a magnetic field in the opposite direction. Therefore, as PC-Sat is spinning through the Earth's magnetic field, an equilibrium spin rate is reached when the magnitude of the force of the induced magnetic field equals the magnitude of the force of the sunlight's momentum on the antennas. The spin rate is also moderated by the time in eclipse. When the sun is blocked from the satellite's view by the Earth, it no longer causes the satellite to spin, and the latter slows down.

PC-Sat also has temperature sensors on each solar panel face. These temperature readings are sent back to Earth as a part of the telemetry data continuously transmitted by PC-Sat. The USNA ground station collects these telemetry data every time PC-Sat passes over Annapolis, Maryland. CDR Robert Bruninga, USNA satellite ground station director and PC-Sat main operator, has monitored these temperature telemetry data over the past year and has determined the spin rate to be about 0.6 rpm while orbit is in eclipse season, and 0.8 rpm during full sun periods.[16]

The orientation of PC-Sat with the magnetic field is accomplished by magnets placed in PC-Sat. The +z face is designed to point towards the south pole, and the –z face to point to the north pole. Therefore, the satellite is always aligned with the magnetic field of the Earth. If PC-Sat were in a polar orbit, meaning it had an inclination of 90 degrees and went directly over both poles, it would have a tumble rate of 2 complete tumbles per orbit. This is not easy to visualize, but think of the satellite above the North Pole with the -z face towards the earth. Then at the equator, the z-axis is aligned with the –z face now pointing to the north and the +z face pointing towards the south. When the satellite is over the south pole, the +z face is now facing the earth, but it is again in the same orientation as it was at the North Pole. Therefore it underwent an entire rotation about the x- or y-axis in half of an orbit. Figure 11 illustrates this motion.

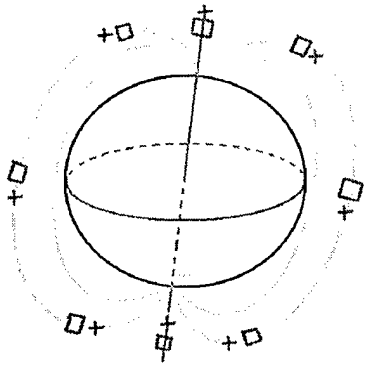

Figure 11: Rotation of magnetically aligned satellite in polar orbit

PC-Sat is not in a polar orbit, however, and therefore will not completely roll over the z-axis. Instead it will wobble along the z-axis. The motion that results from the spin and wobble is not a simple addition of two rotations as a spinning top is. Instead it is a complex tumble, the modeling of which could be a project of its own.

Atmospheric Density

The ideal gas law connects temperature T, pressure p, and density ?, of a gas of molecular weight M, as shown in equation (1) where R is the gas constant.

$$\frac{p}{\rho} = \frac{RT}{M} \tag{1}$$

The change in pressure with height y is given by the hydrostatic equation in equation (2) and is accurate up to several hundred kilometers.

$$\frac{dp}{dy} = -\rho g \tag{2}$$

These two equations are the basis for the exponential decrease of the density of the atmosphere as a function of height, y, the result of which is shown in equation (3) where h is the scale height, the altitude at which the atmospheric density is 1/e the density at the surface of the Earth.[17]

$$\rho = \rho_0 e^{\left(-\frac{y-y_0}{h}\right)} \tag{3}$$

In practice, it has been determined that equation (3) loses accuracy above approximately 500 km.[18] This is due mostly to the mean free path of the air molecules being larger than the scale height, and therefore the molecules have ballistic paths rather than colliding with one another. Above this 500 km threshold a much more complicated model includes the effects of day and night, the Sun's solar activity, the geomagnetic planetary index, and a still unknown variability of ±10 percent. Accurate density predictions become extremely difficult as the number of variables increases. Equation (3) also incurs some errors from the way it was calculated. The acceleration due to gravity, g, is also not a constant as the altitude increases, and the simple integration of equation (2) to equation (3) is not accurate over large differences in height.

There are many published models of atmospheric density, broken into various categories and taking into account a range of variables. Some cover specific conditions or geographic regions, such as the International Tropic Reference Atmosphere. Others are general models for the whole Earth, such as the Jacchia J70 model. The choice of density model is dependent upon the purpose of its use, and so the selection of a model can be difficult.[19]

The atmospheric density that is expected at 800 km using the exponential model (Equation 3) is on the order of 10^{-14} kg/m^3.

Young's Algorithm

In MIDN John Young's Trident project, he developed an algorithm for calculating atmospheric density from a satellite's GPS data. He used Satellite Tool Kit (STK) and ran two models of a satellite's orbit; one without atmospheric drag and one with atmospheric drag. He used the difference between the two models to calculate the change in the orbit due to drag. The position and velocity data taken from a GPS receiver were transformed into the COEs of the orbit. The semi-major axis, a, is a constant throughout an undisturbed orbit, and therefore the change in a is a good measure of the change in the orbit. The density can be related to the change in the semi-major axis over time, as well as to the eccentricity of the orbit and the true anomaly. Equation (4)[20] shows the equation for atmospheric density from COEs.

$$\rho = -\frac{da}{dt} * \frac{m}{C_D * A} * \frac{1}{v_{rel}^2} * \left(\frac{n * \sqrt{1-e^2}}{\sqrt{1+e^2+2*e*\cos(\nu)}} \right) \qquad (4)$$

The mass (m), coefficient of drag (C_D), and cross-sectional area (A) are all properties of the satellite. The velocity of the satellite relative to the atmosphere, rather than to the surface of the Earth, is v_{rel}. The classical orbital elements in this equation are n, e, ?, and a.

Young made several assumptions in his analysis. One was that the spacecraft had a constant cross-sectional area and coefficient of drag. Another was that the atmosphere rotates with the surface of the earth. He also assumed that the effects on the orbit due to the shape of the earth and the atmosphere were independent of one another. Young concluded that the largest potential source of error would be the rotation speed of the atmosphere relative to the satellite.

To solve for the expected change in semi-major axis over time for a given density, Equation 4 can be worked backwards. For the density of the exponential model of $1x10^{-14}$ kg/m^3, the expected change in semi-major axis over time is $-8.9x10^{-6}$ m/s. Table 2 shows the expected changes over various time intervals.

Table 2: Expected changes in semi-major axis for density of 10^-14 kg/m^3

Time interval	Change in semi-major axis
100 minute orbit	-0.053 meters
1 day	-0.768 meters
12 days	-9.212 meters
1 year	-280.2 meters

Spacecraft Mass

PC-Sat has no propulsion system on board and therefore has a constant mass. The mass of the spacecraft is 12 kg. Therefore in the equation for atmospheric density, this mass is also a constant for all orientations and positions.

Cross-Sectional Area

In order to calculate the atmospheric density from Young's algorithm, the cross-sectional area of PC-Sat is needed. Because it is tumbling, the area is continuously changing. Therefore an assumed average value of the cross-sectional area for the entire orbit is needed. This average value was calculated in a way similar to how the average power output of the solar panels was calculated. We know that the rotation of PC-Sat is random and is a tumbling motion as opposed to a spinning motion. Assuming each orientation has an equal chance to be facing the direction of motion, each orientation was weighted according to its frequency of occurrence on the spacecraft. A cube has 6 faces, 12 edges, and 8 corners for a total of 26 "basic" orientations. The three integral orientations (face, edge, and corner) are not the only possible orientations as the satellite can be oriented at any angle between. Figure 12 shows these three orientations, their dimensions, and their respective areas.

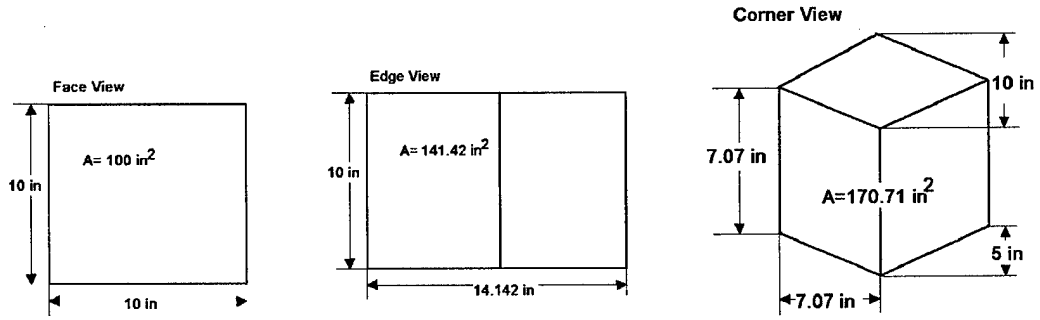

Figure 12: Cross-sectional areas for the 3 major orientations

The face view occurs six times out of 26 orientations, so the area of 100 in^2 is weighted 6/26. After weighting each of these areas according to their frequency of occurrence, the average cross-sectional area is 140.9 in^2 (0.091 m^2). A sphere touching the corners of the cube would have a cross-sectional area of 157.0 in^2 (0.101 m^2).

NORAD includes in its 2-line elements a term called Bstar (B*). This is not the ballistic coefficient, but is related to it. The ballistic coefficient is a part of the equation used to calculate density from the change in semi-major axis. Equation (5) shows how the two terms are related.

$$\frac{R_{\oplus}\rho_0}{2B^*} = BC \tag{5}$$

The atmospheric density here is the density at the orbit's perigee and is assumed to be constant for a given altitude. The radius of the Earth is 6378.135 km, giving a simplified equation for the ballistic coefficient.[21]

$$BC = \frac{1}{12.741621B^*}\frac{kg}{m^2} \tag{6}$$

Figure 13 shows the variance of the ballistic coefficient over a year. The points that are square are the days for which GPS data were collected in January of 2001.

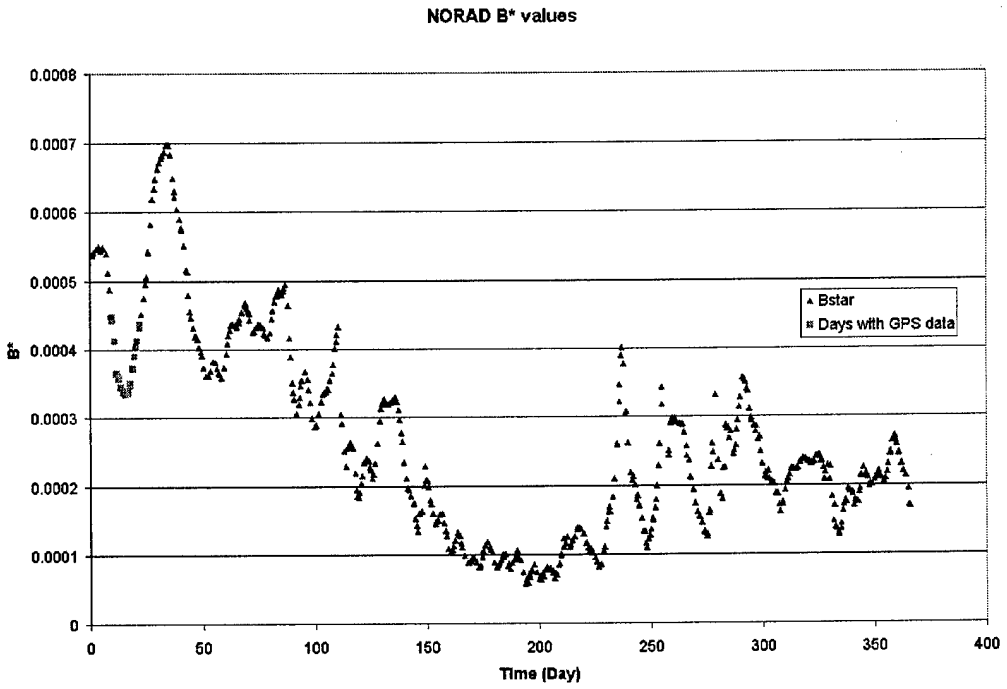

Figure 13: B* values from NORAD 2 line elements

The method by which NORAD calculates this B* value is not well-published. In order to determine the feasibility of using a constant ballistic coefficient, a comparison between the atmospheric densities calculated using the variable B* and the constant ballistic coefficient was done. The result is shown in Figure 14. NORAD apparently uses a constant density, based on the altitude of the satellite, to calculate a B* to fit a smooth atmospheric model. Figure 14 does not show this smooth density that is expected with the B* values, but it does show a smooth density for the constant ballistic coefficient. The scale of the graph is also important to note, as the density is of the same magnitude as predicted using equation (3). The values for atmospheric density that were calculated using equation (4) and NORAD data only are reasonable.

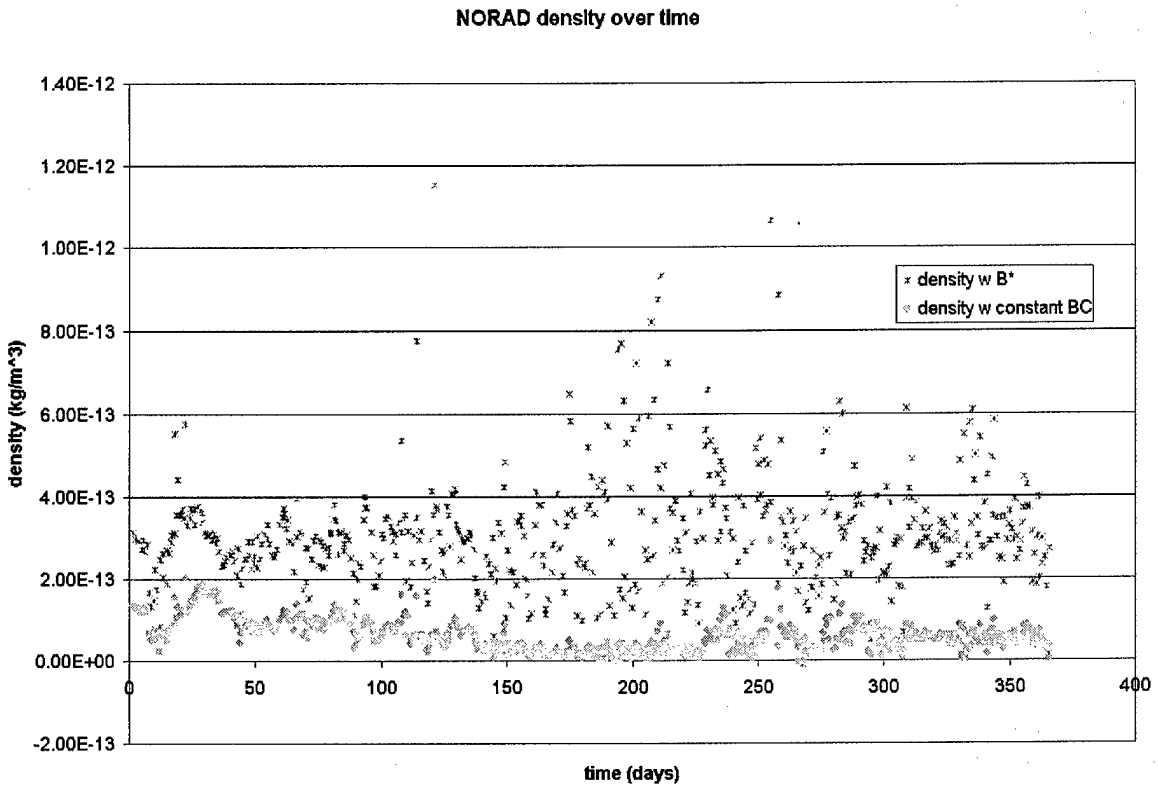

Figure 14: Comparison of variable B* and constant ballistic coefficient

Aerodynamic Drag: Coefficient of Drag

It is important to know the density of the atmosphere through which objects are traveling because of the aerodynamic drag force it creates on the object. Although spacecraft operate in space, there is still a significant amount of atmosphere at low-earth orbit (LEO) altitudes. At an altitude of 800 km, the atmospheric density is approximately twelve to thirteen orders of magnitude less than that found at sea level, around 10^{-14} kg/m^3.[22] This, however, is still significantly denser than the vacuum of space at higher altitudes and therefore this drag effect must be taken into account at lower altitudes. The amount of drag an object experiences is dependent on the amount of atmosphere present, or the atmospheric density. Therefore, the drag

force due to the atmosphere must be accounted for in predicting the orbit of a satellite in low-earth orbit.

There is a major difference between airflow on the surface of the earth and the flow of the atmosphere at orbital altitudes. The airflow experienced on Earth as wind and weather is the result of continuum-flow aerodynamics, meaning the particles in the atmosphere continuously collide with one another and interfere with the path of other particles. Satellites generally experience free-molecule flow in which the molecules are reflected or re-emitted from the spacecraft and do not interfere with approaching or incident molecules. This is a legitimate assumption for spacecraft above 200 km because of the large mean free path of the air molecules relative to the size of the satellite.[23]

Aerodynamic drag is created by particles in the atmosphere colliding with a spacecraft and transferring part of the particle's momentum to the spacecraft. The exact amount of momentum transfer is dependent on the angle of the surface compared to the fluid flow. Equation (7) gives the momentum transfer for any collision where p_i is the momentum of the incident particle and p_r is the momentum of the reflected particle. Equation (8) shows equation (7) in terms of p_i, which is usually a known quantity.[24] The angle ? is the angle of incidence measured with respect to the normal of the surface.

$$\Delta p = p_i + p_r \tag{7}$$

$$\Delta p = p_i\left(1+\frac{p_r}{p_i}\right) = p_i\left(1+f\left(\theta\right)\right) = mv(1+f(\theta)) \tag{8}$$

In a given amount of time, Δt, the total mass deposited on a surface is shown in equation (9).

$$m = \rho A v \Delta t \tag{9}$$

By substituting equation (9) into equation (8) and then using the definition of a force as the change in momentum over the change in time, the result is equation (10).

$$F_d = \frac{1}{2}\rho v_{rel}^2 A C_D . \tag{10}$$

The atmospheric density is ?, the velocity of the spacecraft relative to the atmosphere is v_{rel}, the cross-sectional area is A, and C_D is the coefficient of drag of the spacecraft.

The coefficient of drag is a dimensionless quantity used to describe the interaction between a fluid and an object traveling through the fluid. It is defined as shown in equation (11), taken from equations (8) and (10).

$$C_D = 2\left(1 + f(\theta)\right) \tag{11}$$

The exact value of f(?) is difficult to calculate theoretically because of the uncertainty of the nature of individual atomic collisions. Particles may scatter either elastically or they may adhere to the surface until thermal equilibrium is achieved and scatter randomly. The first is called specular reflection; the latter diffuse reflection.[25] Therefore the most accurate method of determining a spacecraft's coefficient of drag is to determine it experimentally. Drag coefficient values can range from 1.9 to 2.6, with the average value at 2.20.[26]

The coefficient of drag is constant for a spacecraft that has a constant orientation with respect to the atmosphere. This is true for satellites that are perfect spheres or have attitude control systems that keep them in a specific orientation. PC-Sat does not meet either of these qualifications, and therefore does not have a constant coefficient of drag. It is tumbling through space and the faces of the cube are constantly changing angles relative to the atmosphere. Figure 15 shows experimental coefficients of drag for a flat plate at various angles relative to the direction of motion.[27]

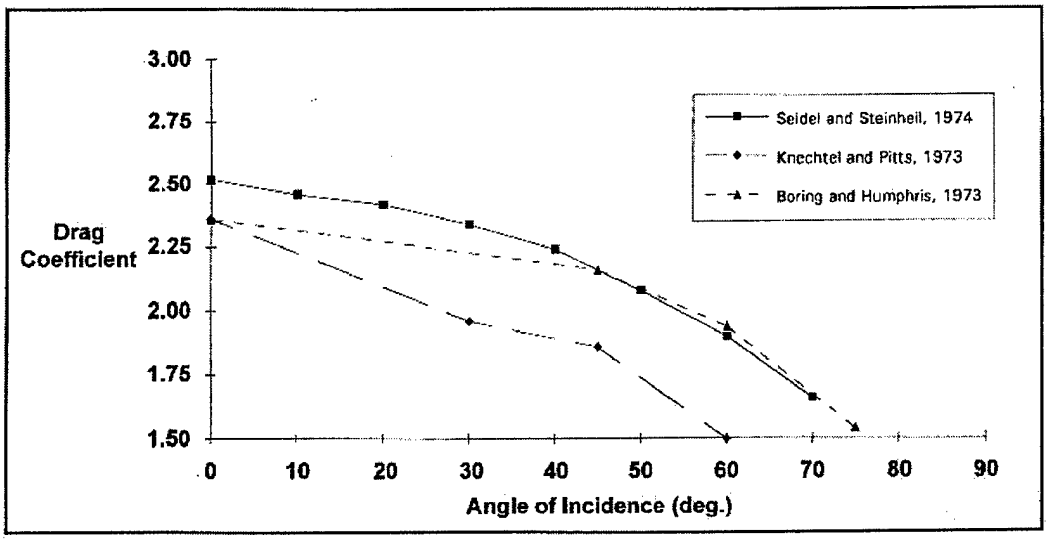

Figure 15: Experimental Coefficients of Drag for a flat plate at varying angles

If PC-Sat were flying face-on with the normal of the face in the direction of motion, it would have a coefficient of drag of about 2.5. If it was flying with an edge in the direction of motion, PC-Sat would effectively be two flat plates flying at a 45 degree angle and the coefficient of drag would be about 2.00. If PC-Sat had a corner in the direction of motion, it would effectively be three flat plates flying with an angle of incidence of 45 degrees. This would also give a coefficient of drag of about 2.00. Using the same method as for the cross-sectional area to weight the different occurrences of each orientation, the average coefficient of PC-Sat is 2.11.

The error that comes from using the average value of 2.11 for the drag coefficient can be estimated. The equation for density as given in equation (4) shows that density is inversely proportional to the coefficient of drag. If the average coefficient of drag is 2.11 and the instantaneous coefficient of drag is the maximum theoretical value of 2.5, the increase in drag is 18 percent of the average. This in turn causes a change in density of about 18 percent.

Considering the goal of using this method is to reduce the uncertainty in measurement to below 15 percent, using an average value for coefficient of drag will not allow this. The conclusion section will further address this issue.

Relative Velocity

The velocity of the satellite relative to the fluid through which it is traveling is the relative velocity referred to in equation (4). The velocity that is calculated for the satellite in the inertial reference frame is relative to the center of the earth. The velocity of the atmosphere relative to center of the center of the earth must be known in order to calculate the velocity of the satellite relative to the atmosphere.

King-Hele derives an equation for the velocity relative to the atmosphere dependent on the spacecraft's velocity relative to the center of the Earth, v, the spacecraft's distance from the center of the Earth, r, and the inclination of the spacecraft's orbit, i.[28] Equation (12) shows the result, with w as the angular rotation of the atmosphere in the east-west direction.

$$v_{rel} = v\left(1 - \frac{rw}{v}\cos i\right) \tag{12}$$

If the atmosphere is assumed to rotate with the surface of the Earth, at 800 km the atmosphere will be moving at approximately 0.4 km/s. The atmospheric rotation in the north-south direction is assumed to be zero because it is significantly smaller than the east-west winds of 0.4 km/s. PC-Sat has an orbital speed on the order of 7.5 km/s. The east-west winds are then seen to be about five percent of the orbital speed. In equation (4), the atmospheric density is inversely proportional to the square of this relative velocity. Altering the relative velocity by

five percent gives a change in the density of about ten percent. If this were the only uncertainty in the calculation of atmospheric density, this would give a more accurate density than the current model.

Velocity from Position

The GPS data received from PC-Sat are in the form of time, latitude, longitude, and altitude. The vital piece of information that is missing is the velocity vector. Knowing the position vectors of the satellite at two different times should make the velocity easy to find. Unfortunately this is not the case. The path over which the satellite has traveled during the time between GPS fixes is not a straight line, so v = ?d/? t is not an accurate calculation. Figure 16 shows the difference in distances between the straight and curved sections. This difference leads to large errors in the calculated velocity. The velocity of the satellite is very important because it is necessary for extracting the classical orbital elements from the GPS positions, the COEs that are in turn necessary to calculate the atmospheric density.

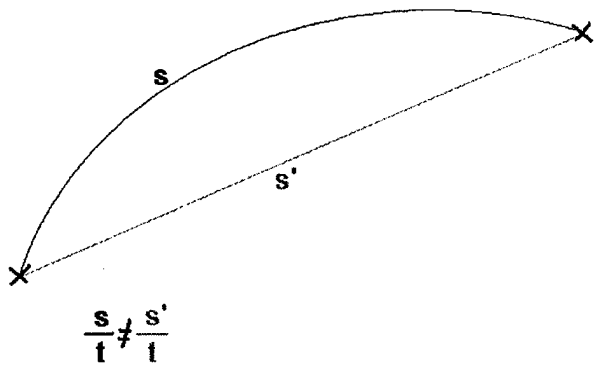

Figure 16: Difference in path length of curve and straight line

GPS and NORAD

One source of PC-Sat's velocity is the NORAD data. The NORAD classical orbital elements are taken only once or twice a day. However, the velocity can be easily determined from the classical orbital elements for those one or two readings each day. Figure 17 shows the velocities from the NORAD data through the year.

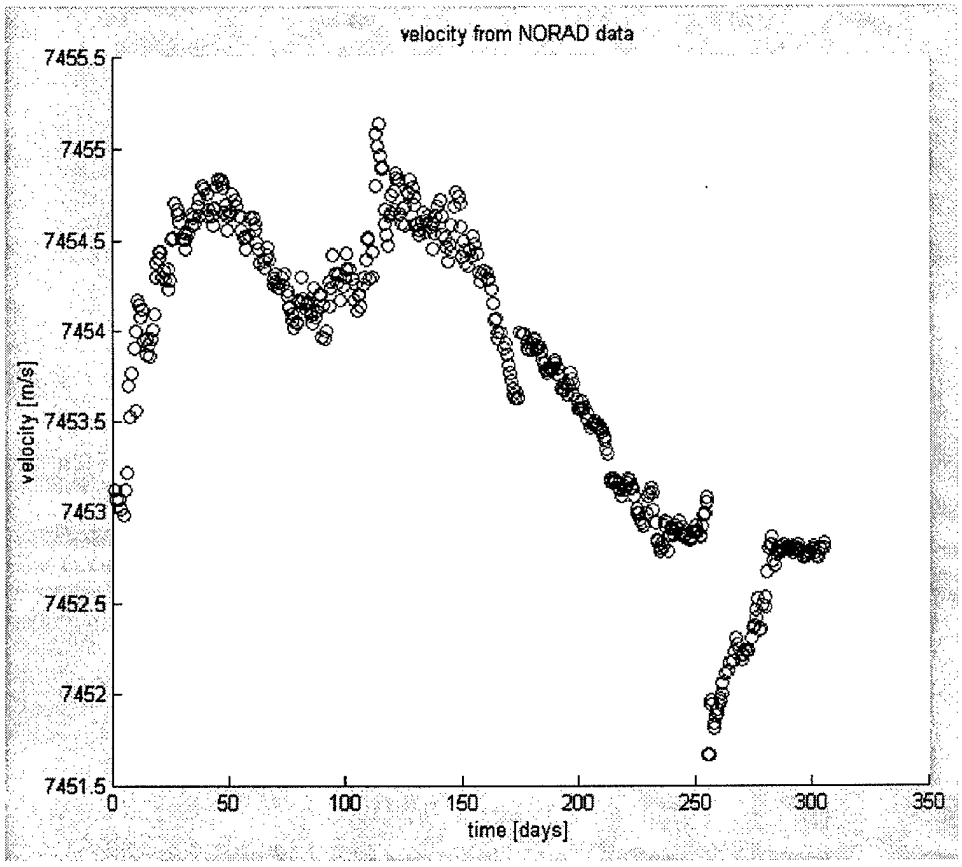

Figure 17: Velocity of PC-Sat from NORAD data

Although the velocities appear to be randomly scattered, they are in a very narrow window between 7451 m/s and 7456 m/s. The reason the velocity is changing over time is because the NORAD elements are averages over a section of the orbit. The orbit is slightly elliptical, so the velocity at perigee is slightly faster than the velocity at apogee. The NORAD elements are taken

at the same position from Earth each time, but the satellite is in a different part of its orbit, and so the velocity captured by NORAD changes slightly. This average velocity is not suitable for calculations of atmospheric density. The instantaneous values are necessary for any analysis of the atmospheric density from GPS data.

Herrick-Gibbs Method

One method of calculating instantaneous velocity is to use only position vectors to calculate the orbit and thus the velocity. This Gibbs method uses three time-sequential coplanar position vectors of a satellite in its orbit.[29] The result of the method is a velocity vector that corresponds to the middle position vector. When this method is used the velocity ranges wildly with an apparent maximum velocity about 7.6 km/s. This indicates that the method used to calculate velocity is incorrect rather than the data. The lower velocities could be due to larger time spans between position vectors and thus a similar result as shown in Figure 16 above.

The Herrick-Gibbs method is similar to the Gibbs method in that it uses 3 consecutive position vectors, but it also incorporates the time between the fixes into the solution for the velocity. The method uses a Taylor-series expansion to obtain the velocity vector for the middle position vector. The method is specifically designed for position vectors taken by a ground station, which are generally very close together. This is equivalent to the data received from the GPS receiver, as they are also position vectors spaced close together. Figure 18 shows the velocities calculated using Herrick-Gibbs method.

The Herrick-Gibbs program includes a test to ensure the three data points are actually co-planar before the velocity vector is calculated. Because PC-Sat does not have memory on board to store data between passes over the ground station, only the data retrieved real-time are

available. The result is gaps in the data between ground stations across which the velocity vectors cannot be accurately calculated. Therefore, if the spacing between two of the three position vectors is greater than 0.3 degrees, the velocity vector is set to zero.

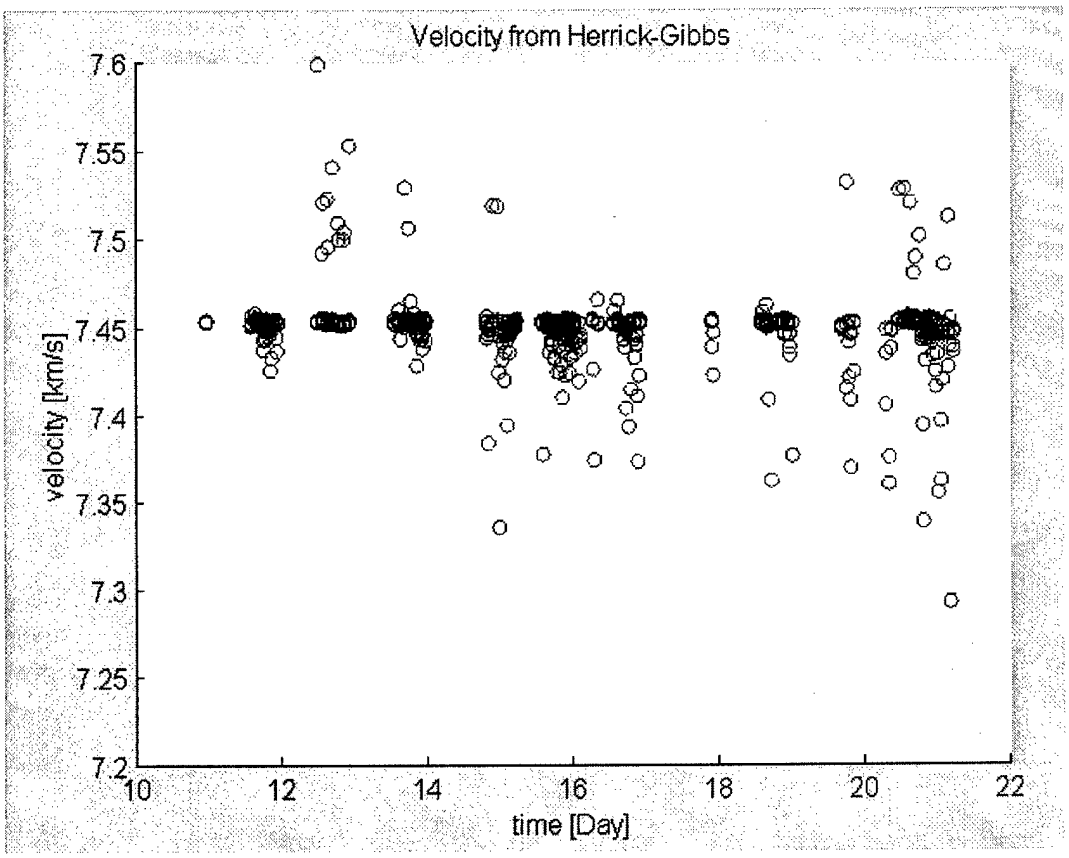

Figure 18: Velocity from Herrick Gibbs

The majority of the velocity magnitudes are around 7.45 km/s as is expected from the NORAD average velocities. The mean velocity calculated using the Herrick Gibbs method is 7.4484 km/s. From Figure 18, any velocity above 7.47 and below 7.44 appears to be an outlier. If these points are not included, the mean velocity is 7.4528 km/s. There appear to be a large number of outliers on this plot. When examined closer, these points are almost all the first or last

few GPS positions in a group. This phenomenon will be explained later in the GPS Outliers section.

Kalman Filter

An alternate way to calculate the instantaneous velocity vector is using a Kalman filter. This is "a technique for computing the best estimate of the state of a time-varying process."[30] It is a predictor-corrector technique, where the state is estimated at each successive observation time to compare to the actual observation. The Kalman filter also "carries all the information concerning past measurements in its current state and covariance estimates and therefore doesn't need to reprocess all of the past measurement information at each step."[31] Because an orbit is not linear, an extended Kalman filter must be used. The extended filter has a two-step method. The new state estimate is predicted using previous data, and then the prediction is updated with the new observations. It uses a set of matrices to bias the calculation of the propagated set of data based on accuracies of the current set of data.

The extended Kalman filter requires a starting position and velocity vector. The NORAD data are very useful for this. A Matlab program titled "NORADtoRV10.m" transforms the COEs from the 2-line elements of January 10, 2002 into position and velocity vectors measured at the time of the NORAD reading. This is the first NORAD data point after the start of the GPS data. The Matlab program "Predict.m" turns the position and velocity vectors back into the COEs and propagates them forward to the next observation time. The predict program uses only the effect due to the oblateness of the Earth and atmospheric drag to predict the future orbital elements. The predicted position is compared to the GPS measured position and the difference is recorded

in a matrix. The bias matrices then are used to calculate a final and "filtered" position and velocity vector using both the predicted and measured data. Figure 19 and Figure 20 show the velocities calculated using the Kalman filter.

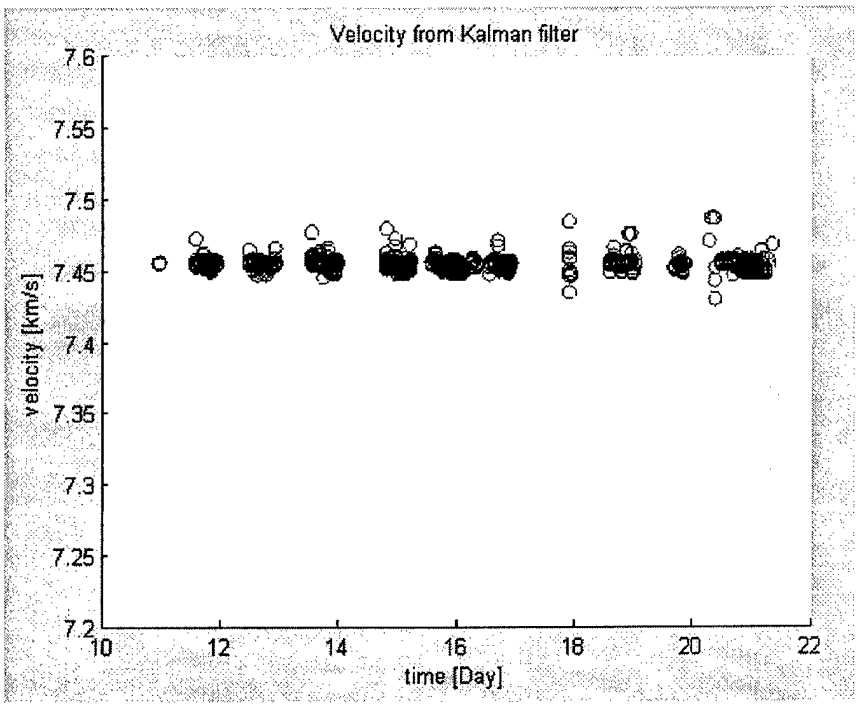

Figure 19: Velocity from Kalman filter: same scale as HGibbs velocity

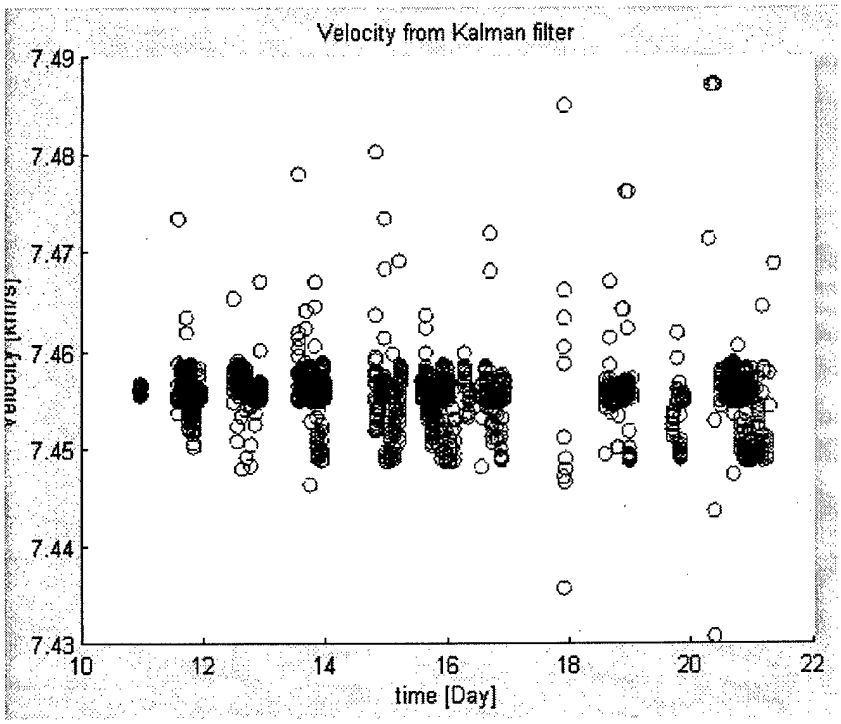

Figure 20: Velocity from Kalman filter: different scale from HGibbs velocity

Orbit Prediction: Non-Spherical Earth

The prediction program used for the Kalman filter only takes two disturbances into account as it propagates the satellite's position forward. The orbit of a satellite would be a perfect ellipse if it occurred about a point mass in a perfect vacuum. The classical orbital elements would be constants and the satellite would return to the same position in space every orbit. The earth is not a point mass, and space is not a perfect vacuum, especially at 800 km where PC-Sat is orbiting. The different perturbations affecting orbits can be accounted for mathematically.

"The Earth is appreciably oblate, the equatorial diameter being 42.77 km greater than the polar diameter."[32] The result of this oblateness is a change in the right ascension of the ascending node, Ω, and a change in the argument of perigee, ω. The orbit rotates about the Earth

in the inertial reference frame, and the orbit rotates in the orbital plate. The amount of each rotation is each dependent on the semi-major axis, the eccentricity, and the inclination of the orbit as shown in equations (13) and (14).[33]

$$\dot{\Omega} = -\frac{3}{2} n J_2 \left(\frac{R}{a(1-e^2)} \right)^2 \cos i \tag{13}$$

$$\dot{\omega} = \frac{3}{4} n J_2 \left(\frac{R}{a(1-e^2)} \right)^{3.5} (5\cos^2 i - 1) \tag{14}$$

Each of these equations is a simplified version using only the J_2 term, the expanded version of each uses J_n values 2 and above. The numerical values for the even J_n terms were developed using accurate measurements of $\dot{\Omega}$ for many satellites in different inclinations and solving for the smaller J_n values. The odd J_n values were determined by the same method but using the equation for the periodic variation in perigee distance. Table 3 shows the numerical values of J_n.[34]

Table 3: Numerical values of J_n

$10^9 J_2 = 1082626 \pm 1$	$10^9 J_5 = -245 \pm 5$	$10^9 J_8 = -210 \pm 8$
$10^9 J_3 = -2530 \pm 4$	$10^9 J_6 = 543 \pm 7$	$10^9 J_9 = -90 \pm 7$
$10^9 J_4 = -1624 \pm 2$	$10^9 J_7 = -336 \pm 6$	$10^9 J_{10} = -242 \pm 9$

Equations (13) and (14) are shown in the simplified form using only the J_2 term and assuming the other J_n terms are sufficiently small to be disregarded. The critical inclination for the change in longitude of the ascending node (equation (13)) is 90 degrees. At this inclination the longitude of the ascending node does not change as the cos(90 deg) is zero. Therefore for higher inclination orbits the change is less than for equatorial orbits. The critical inclination in equation (14) is 63.4 degrees, the inclination at which there is no rotation of the argument of

perigee due to the gravitational field of the Earth. The closer a satellite's orbit is to this inclination, the less the Earth's oblateness affects the position of the argument of perigee.

Orbit Prediction: Effects of the Sun and Moon

The Sun and the Moon also affect orbits about the Earth by their gravitational attractions and also by solar radiation pressure from the Sun. "The effects of the luni-solar gravitational attractions are generally small and periodic: the change in perigee distance rarely exceeds 2 km for a close satellite (less than 1500 km) with $e < 0.2$, although much larger changes occur for highly eccentric orbits. The four orbital elements e, i, O, and ? are all affected, but a remains constant."[35] Since the focus of this project is on the semi-major axis, a, the gravitational effects of the Sun and Moon can be ignored.

The solar radiation pressure is a periodic perturbation, unlike the atmospheric drag. As referenced earlier, this solar radiation pressure is the force which causes spin about the z-axis on PC-Sat, and is not accounted for beyond this.

Classical Orbital Elements

The equation for atmospheric density (equation (4)) requires several Classical Orbital Elements (COEs), including the eccentricity, mean motion, true anomaly, and semi-major axis. The average values of these COEs for PC-Sat can be directly taken from the NORAD two-line element sets. They can also be calculated from the position and velocity vectors at each instant for which we have data.

NORAD Data

The data from NORAD are initially in the form of the 2-line elements. Rather than the specific position of the satellite, the NORAD data give information about the overall orbit in which the satellite travels. The semi-major axis is not specifically given in the 2-line element set, but the mean motion is. The relationship between mean motion (n) and semi-major axis is

$$n \equiv \sqrt{\frac{\mu}{a^3}}. \tag{15}$$

Figure 21 shows the semi-major axis for PC-Sat calculated from NORAD over 10 months, with January 1, 2002 as day zero. The data overlaid are for the days that have corresponding GPS data. The decline in semi-major axis determined from the NORAD data across the 12 days of January containing GPS data is 60 m. Compared to Table 2, this change in semi-major axis is almost 10 times greater, which means a factor of ten difference between the NORAD density and the exponential model density should be expected.

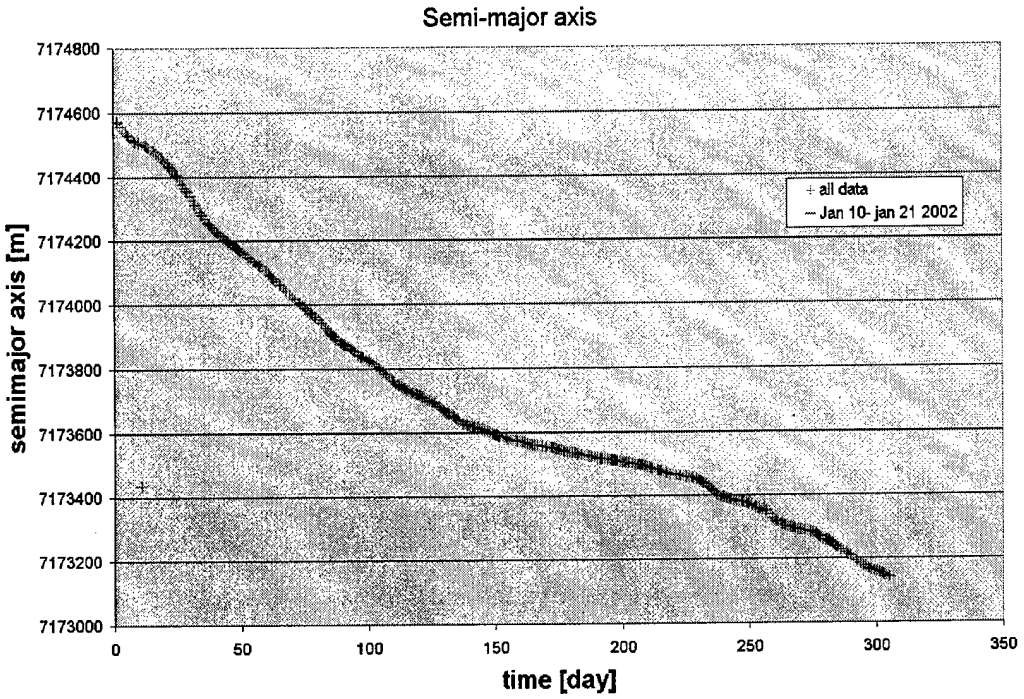

Figure 21: Semi-major axis from NORAD data

GPS Data

Eccentricity

The eccentricity vector can be determined directly from the position and velocity vectors at a point in time. The magnitude of the vector is the eccentricity of the orbit. The direction of the vector points toward perigee. PC-Sat's orbit is almost circular, so the actual position of perigee is somewhat difficult to define. Only the magnitude is necessary for final calculations, which will not count on the direction, but the vector is needed to calculate the velocity. Equation (16) shows the equation for the eccentricity vector, where μ is the gravitational constant.

$$\vec{e} = \frac{\left(v^2 - \frac{\mu}{r}\right)\vec{r} - (\vec{r}\cdot\vec{v})\vec{v}}{\mu} \qquad (16)$$

Mean Motion

The mean motion is a function of the semi-major axis. Equation (17) shows the relationship.

$$n = \sqrt{\frac{\mu}{a^3}} \tag{17}$$

True Anomaly

The true anomaly is the position of the spacecraft in its orbit, referenced to the point of perigee. It is calculated from the eccentricity and position vector. Equation (18) gives the equation.

$$\cos(\nu) = \frac{\vec{e} \cdot \vec{r}}{|\vec{e}||\vec{r}|} \tag{18}$$
$$\text{if}(\vec{r} \cdot \vec{v} < 0), \text{then} \nu = 360° - \nu$$

Semi-Major Axis

The semi-major axis is also a function of the position and velocity at any particular time. Equation (19) shows the relationship.

$$a = \left(\frac{2}{r} - \frac{v^2}{\mu} \right)^{-1} \tag{19}$$

Another method of finding the semi-major axis is to use the period. The period of the orbit is directly related to the semi-major axis, as shown in equation (20). The symbol μ is the Earth's gravitational parameter, the product of Earth's gravitational constant G and Earth's mass m. The numerical value of μ is 3.986×10^5 km^3/s^2.

$$P = \frac{2\pi}{\sqrt{\mu}} a^{3/2} \tag{20}$$

The difficulty of this method lies in the fact that the orbit is not a closed loop. The satellite does not return to the exact same position every orbit, one of the results of atmospheric drag. Therefore a definition of orbital period for PC-Sat must be agreed upon before determining the value of the orbital period. The right ascension and declination would be the best coordinates system for this because they are inertial with respect to space and the orbital plane, similar to longitude and latitude relative to the surface of the Earth. Therefore a certain right ascension could be defined as the starting point of the orbit and a period is complete when the satellite returns to that right ascension. The errors involved with this calculation come from the time between data points. In thirty seconds PC-Sat travels approximately 225 km. The time of crossing can be approximated by using linear interpolation between the two points on either side of the reference right ascension. This method assumes that the satellite is traveling at a constant speed between the two data points used for interpolation. This is true for a perfectly circular orbit, but PC-Sat's is slightly elliptical.

Experimentally, this method did not give the desired accuracies. The reliability of the semi-major axis calculated greatly depended upon the right ascension or declination chosen. The right ascension I chose was the longitude of the Naval Academy because most of the GPS data recorded were taken here, so the position coordinates of PC-Sat are much more frequent across this right declination. Table 4 shows the varying average semi-major axis as a function of the reference latitude. The difference between the semi-major axis calculated at the equator and at the top of the orbit is about 60 km. The chosen reference point for the beginning and end of the orbit has too much influence on the accuracy to be used with this data set to calculate the atmospheric density to within 15 percent. With smaller intervals between GPS positions, this

method could be possible. The thirty second intervals, however, show in Table 4 to be too large for accurate calculation of semi-major axis.

Table 4: Semi-major axis calculated using period

Reference latitude [deg]	Mean semi-major axis [km]
0	7134.2
10	7079.1
20	7001.9
30	7002.1
40	6982.8
50	6974.0
60	6973.6

Outliers

NORAD Outlier

The outlier in the time-series data of the NORAD semi-major axis (Figure 21) is somewhat confusing. We know from orbital dynamics that it is physically impossible for the satellite to have a semi-major axis decrease or increase that amount without a propulsion system or a significant collision with another object in space. For it to have decreased and then increased back to the original orbit by two collisions is impossible. Even if it was knocked into a lower orbit somehow, it would then be traveling at a higher velocity and it would not return to the same orbit it left. Therefore the single outlier in the NORAD data was discarded as being a corrupted data point.

GPS Outliers

The outliers in the GPS data tend to occur infrequently, several at a time, and afterwards the orbit returns to the expected orbit. As with the NORAD outlier, the satellite cannot move from one orbit to another and back again without some propulsion system or collision. PC-Sat

does not have a propulsion system, which leaves only collisions as a possibility. The return of the satellite to the predicted position after several outliers indicates that the orbit did not actually change as the data suggest.

The Kalman filter diverged several times rather than converging as it was designed. The divergence occurred when the predicted position and the observed position were drastically different than one another and the program could not resolve which position was correct. The source of the divergence was GPS data points that were not a part of the expected orbit. Figure 22 shows just one of these outlier groups in the data, along with the points around them to show where the orbit is expected to be.

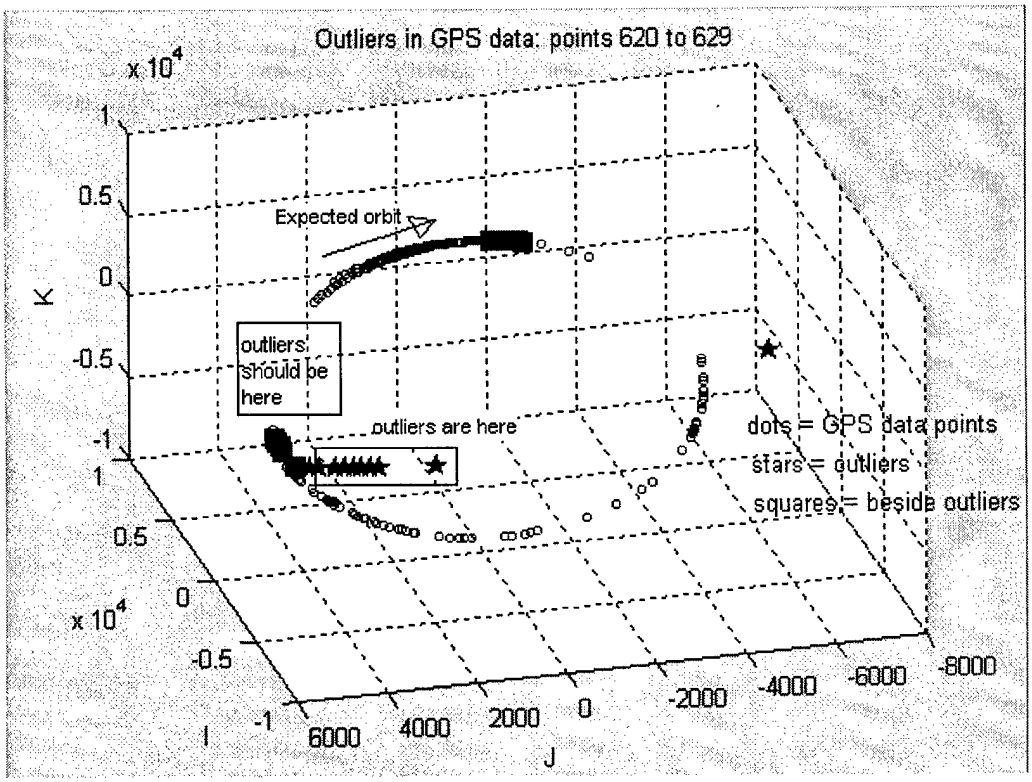

Figure 22: Outlier in GPS position vectors

In Figure 22, the small dots are the satellite's orbit. The larger squares clumped together are the points to either side of the outliers and are along the path expected for the satellite's orbit. The stars are the outliers, and should lie between the two groups of diamonds. They are in noticeably different positions from the rest of the expected orbit. The prediction program in the Kalman filter predicted a position vector to be along the expected orbit, and then compared the prediction to the outlier. The result was a large difference between the expected and reported position and filter was unable to predict the next position, causing a divergence.

These few outliers are most likely caused by the tumbling of the satellite. The GPS receiver that is on PC-Sat has in it a program to predict where each of the GPS satellites will be from one observation to the next. Of the 12 possible satellites from which the receiver can receive, 10 to 11 are constantly in view. The receiver tracks these satellites that are in view and predicts where they will be for the next pass. The tumbling, however, is not accounted for in the program. The receiver rapidly detects new GPS satellites in view, but in these three cases, probably is not as quick. New satellites were most likely added to the solution during these observations, and the receiver was either unable to track them or unaware that they were not the ones being tracked. The receiver notes these observations as being accurate, but the figures above show that they are not. The Kalman filter also shows that when the outliers are removed from the data set, the filter converges and produces positions and velocities at each observation time. With the outliers included, the Kalman filter diverges at the first one and the program stops.

Sunny Leung, German Space Operations Center, German Aerospace Center (DLR), has also noticed these few outliers and attributes them to the tumbling motion of PC-Sat. "Also bear in mind, there is a small spin on PC-Sat which caused frequent change in GPS visibility, and this

may lead to erroneous measurements."[36] The first and initial attempt to use the GPS receiver was immediately after launch of the spacecraft. "Rapidly varying antenna attitude with respect to the GPS constellation introduced great difficulty to the receiver in achieving bit and frame synchronization with a particular PRN. Although the Doppler aiding algorithm was providing correct Doppler shift information to the frequency search routing inside the receiver, the unfavorable tracking condition caused difficulty. ... In the second attempt, on the 31 October 2001, the GPS receiver was able to track 11-12 satellites. ... This observation indicated the improved GPS antenna attitude over the 30 day period, which would suggest a reduced spin rate and tumbling motion caused by the magnetic stabilization onboard."[37] The tumbling initially caused massive problems for the GPS receiver, but the magnets in PC-Sat have slowed the tumbling and allowed the satellite to receive signals and calculate its position. However, the few outliers that did occur are most likely due to the tumbling motion and can be eliminated from the data set.

Results

Table 5 summarizes the average atmospheric density over the twelve day period during which the GPS receiver was activated.

Table 5: Summary of average atmospheric densities using various methods

Method	Density [kg/m^3]
Theoretical	$\sim 1 \times 10^{-14}$
NORAD	0.91×10^{-13}
Herrick Gibbs	1.46×10^{-12}
Kalman Filter	1.29×10^{-12}

NORAD Results

From the NORAD data, there is a decrease in semi-major axis between each data point taken, each a half day to a day apart. This visible decrease in the orbit over the short period of time shows that the atmosphere at 800 km is affecting the satellite. Using equation (4) to calculate the density gives an average density of 0.91×10^{-13} kg/m^2 over the 12 day period. The average density of the atmosphere through which the satellite traveled between NORAD data points throughout the year 2002 was 6.29×10^{-14}.

The predicted atmospheric density between each pair of NORAD data points can also be calculated. Figure 23 shows the result of this calculation. The density does vary over each time period, which is expected because the slope of the semi-major axis in Figure 21 is not constant. The greater the slope, the higher the atmospheric density is expected to be. The days over which GPS data were taken have some of the steeper slopes in the data, and are therefore expected to have higher densities. There is no known relationship between the higher density recorded during the time the GPS receiver was turned on and the fact that the receiver was on.

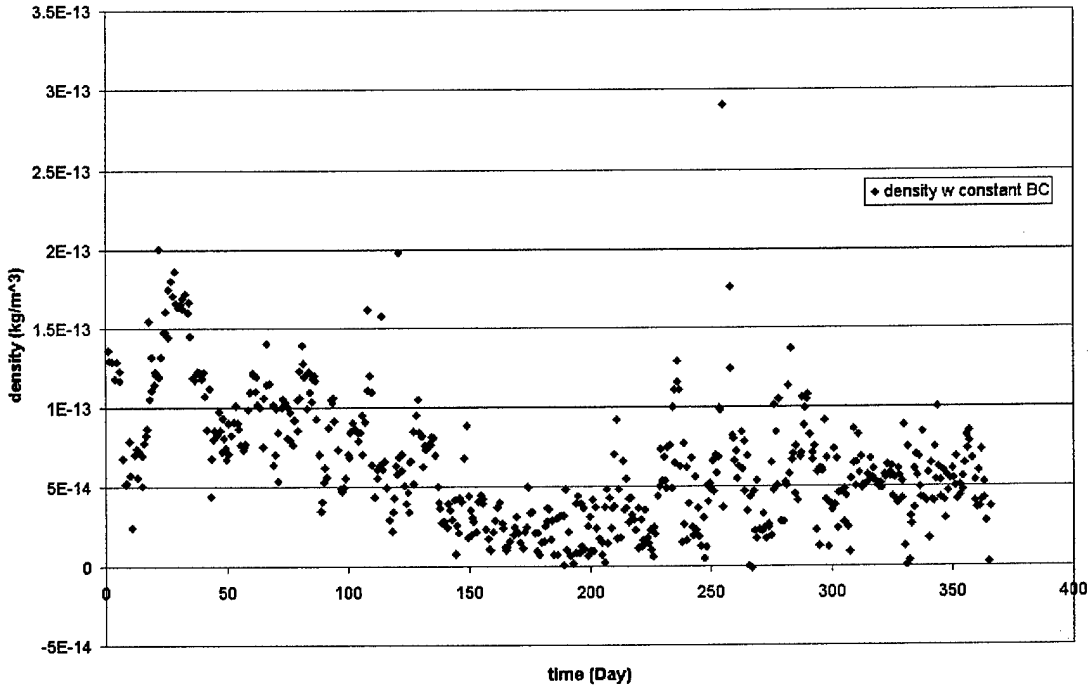

Figure 23: NORAD atmospheric density using constant BC

GPS Results

The goal of this project was to be able to calculate the atmospheric density from the GPS data over near-real time intervals. The first calculation was the overall change over the twelve days. The change here is expected to be comparable with the change expected from the NORAD two-line elements, giving similar density calculations.

First used was the Herrick-Gibbs approach to calculate the velocity and atmospheric density. Once the outliers were eliminated (discussed with Figure 18), mean density for the twelve days was 1.4554×10^{-12} kg/m^3. This differs by an order of magnitude from the simple theoretical density, as well as the density calculated using NORAD data. The density was also calculated by using the change in semi-major axis between each GPS data point, but that method

was not successful. The data were not smoothed in any way, so the calculated densities took into account all the noise in the data. The result was a mixture of positive and negative densities, physically impossible because the density cannot be negative. The Herrick-Gibbs was not the best choice for this research. Had the data points received from the GPS receiver on PC-Sat been completely continuous without any large gaps in time, the Herrick-Gibbs method could have proved very useful. However, the number of discontinuities severely limited the applicability of this method.

A better method turned out to be the Kalman filter. Because it smoothes the data as it runs, the Kalman filter gives results more free of high frequency noise. It is also able to use all the data points in the set without relying on the points to either side being close in time. Using the Kalman filter, the average atmospheric density for the twelve days was calculated to be 1.2919×10^{-12} kg/m^3. This is also about two orders of magnitude different than the theoretical and one order of magnitude from the NORAD calculated densities. The next step was to calculate the atmospheric density by binning the data over varying numbers of points. The results were also heavily dependent on the number of points averaged into a single calculated density. Table 6 shows the mean of the densities calculated using various numbers of data points per bin.

Table 6: Densities using Kalman filter, averaged across various data ranges

Number of data points per bin	Density (kg/m^3)
19	1.63×10^{-11}
219	1.58×10^{-12}
519	1.18×10^{-12}
819	7.26×10^{-13}
1119	1.19×10^{-12}
1519	1.28×10^{-12}

The densities calculated using the GPS data do not match what they were expected to be using simplistic theory, which is not of much concern. The theory uses an exponential atmospheric model, but it well known that the atmosphere does not decrease exponentially at 800 km altitude. TheGPS densities are only about an order of magnitude different from what was calculated using the NORAD data. To identify the source of this difference, the data themselves were examined. For an 800 km orbit with an approximate eccentricity of 0.0005, a difference of 7.2 km between perigee and apogee is expected. The plot of the magnitudes of the position vectors is shown in Figure 24. The difference between the largest and smallest of these numbers is 6.9 km.

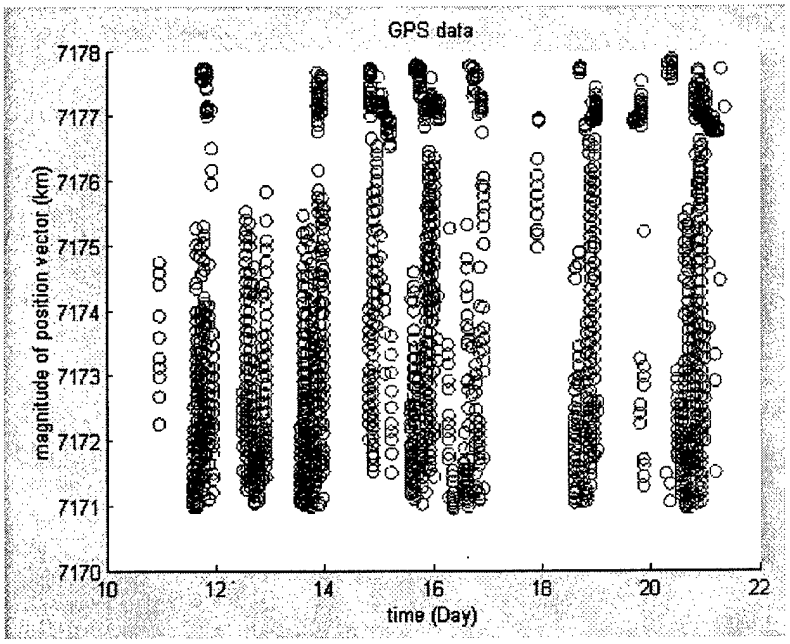

Figure 24: Magnitude of position vectors from GPS data

The calculation of these position vectors includes the eccentricity of the Earth itself and the altitude of the satellite. Without including the eccentricity of the Earth, the range would be 19.5 km rather than just 6.9 km. The semi-major axis calculated from these was expected to be

slightly decreasing over time. Instead what they showed was that the semi-major axis moved similarly to the orbits. Figure 25 shows that as the satellite moved closer to the Earth, the semi-major axis seemed to decrease, and as the satellite moved away from the Earth, the semi-major axis seemed to increase.

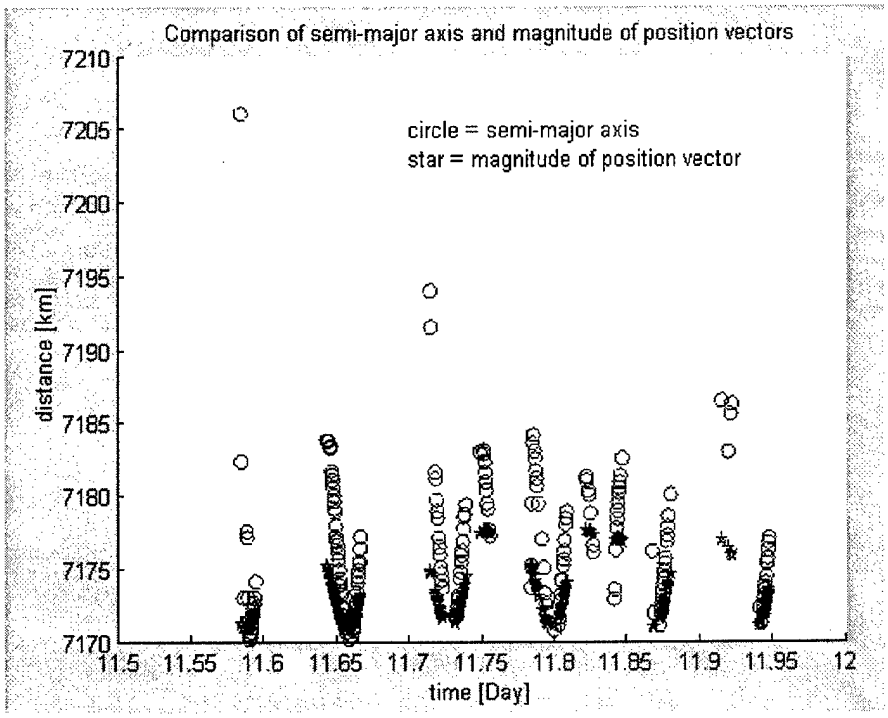

Figure 25: Comparison of semi-major axis and position vector

The semi-major axis should not depend on the satellite's position in the orbit and should be independent of the actual distance of the satellite from the center of the Earth. As the satellite gets closer, the speed should increase and the semi-major axis, a function of position and velocity shown in equation (19), should remain constant. This is not the case shown in Figure 25 and it appears that the calculated velocity is the probable source of the error.

I was able to calculate a density using the GPS data. The result was different than predicted by a simple model, but this acceptable because the simple exponential model is not recognized to be correct at 800 km. My results are also only an order of magnitude different than the average values calculated using NORAD.

Conclusion and Future Work

This project has turned out to be true research. The method that was originally designed to determine semi-major axis did not work, and neither did several others. The final result was that neither the Herrick-Gibbs method nor the Kalman filter were able to calculate the accurate velocities needed for determination of the semi-major axis. This in turn did not allow accurate calculation of the atmospheric density.

A part of the difficulty was that PC-Sat was not the optimal satellite to use for this research. An ideal satellite would be spherical and, since the effects of atmospheric drag are also much greater at lower altitudes, a lower altitude orbit would also be better for this analysis. The GPS receiver on PC-Sat took position fixes about three to four times per rotation, during which the coefficient of drag changed. This causes a significant error of up to 18 percent. Another difficulty was the lack of velocity data from the GPS receiver. The receiver can calculate these data, but velocity was not a part of the data received from PC-Sat. A higher rate of positioning would also give better results in calculating density. The receiver on PC-Sat was set to fix at 30 second intervals, during which the satellite travels almost 225 km. Shorter fix intervals would give smaller errors between predictions and GPS positions and perhaps improve the accuracy of the Kalman filter in calculating the velocity.

Future work with this particular data set could continue. There are other ways of calculating the velocity of a spacecraft given its position than were examined during this research. It would also be interesting to study a different satellite with different orbital characteristics. A better GPS receiver, previously space tested and designed to measure height above the surface of the Earth, would also improve the results of this method for calculating atmospheric density. Being able to use real-time or near-real time data to calculate the atmospheric density for a small interval of an orbit would be a great leap in knowledge of the Earth's upper atmosphere. With current accuracies at or greater than 15 percent, there is certainly room for improvement. Several satellites of known ballistic coefficients and circular orbits would make this research possible.

Endnotes

[1] http://web.usna.navy.mil/~bruninga/pcsat.html

[2] Bate, Roger R., Mueller, Donald D., White, Jerry E. Fundamentals of Astrodynamics. New York: Dover Publications Inc., 1971. pg 59.

[3] http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

[4] Leung, Suuny; Montenbruck, Oliver; Bruninga, Bob. "GPS Tracking of Microsatellites – PCSAT Flight Experience." 22 October 2002.

[5] http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

[6] http://129.247.179.151/dlr/gps/orion/index.html

[7] Leung, Sunny. Email dated 27 Mar 2003.

[8] http://www.igeb.gov/SPS-2001-final.pdf

[9] Leung, Suuny; Montenbruck, Oliver; Bruninga, Bob. "GPS Tracking of Microsatellites – PCSAT Flight Experience." 22 October 2002.

[10] Frank H. Bauer, "High Earth Orbit GPS Flight Experiment AMSAT-OSCAR 40" briefing at NASA GSFC, 01 November 2001.

[11] http://www.norad.mil/

[12] http://www.norad.mil/CMOC_FACT_SHEET.htm

[13] http://www.sew-lexicon.com/gloss_s.htm#SPACE_DETECTION_AND_TRACKING_SYSTEM

[14] http://www.sew-lexicon.com/gloss_s.htm#SPACE_DETECTION_AND_TRACKING_SYSTEM

[15] http://www.rc3.org/archive/inform/2/4.html

[16] Bruninga, CDR Bob, Personal interview, November, 2002.

[17] David A Vallado. Fundamentals of Astrodynamics and Applications, Second Edition. New York: The McGraw Hill Companies, 2001. pg. 532.

[18] Desmond King-Hele. Satellite Orbits in an Atmosphere: Theory and Applications. Glasgow: Blackie and Son Ltd., 1987. pg 12-13.

[19] Guide to Reference and Standard Atmospheric Models, 1997, American Institute of Aeronautics and Astronautics.

[20] David A Vallado. Fundamentals of Astrodynamics and Applications. New York: The McGraw Hill Companies, 1997. pg 605.

[21] Vallado, 114.

[22] Tribble. pg 72.

[23] King-Hele, pg 21.

[24] Alan C. Tribble. The Space Environment: Implications for Spacecraft Design. Princeton, NJ: Princeton University Press, 1995. pg 74.

[25] Tribble, pg 75.

[26] Tribble, pg 76.

[27] Tribble, pg 75.

[28] King-Hele, pg .

[29] Vallado, pg. 408.

[30] Vallado, pg 718.

[31] Vallado, pg 718.

[32] King-Hele. pg 5.

[33] King-Hele. pg 276.

[34] King-Hele. pg 276.

[35] King-Hele, pg 9.

[36] Leung, Sunny. Email to m032712@usna.edu. Received 27 Mar 2003.

[37] Leung, Suuny; Montenbruck, Oliver; Bruninga, Bob. "GPS Tracking of Microsatellites – PCSAT Flight Experience." 22 October 2002.

Bibliography

Bate, Roger R., Mueller, Donald D., White, Jerry E. Fundamentals of Astrodynamics. New York: Dover Publications Inc., 1971.

Bauer, Frank H. "High Earth Orbit GPS Flight Experiment AMSAT-OSCAR 40" briefing at NASA GSFC, 01 November 2001.

Bruninga, CDR Bob, Personal interview, November, 2002.

http://web.usna.navy.mil/~bruninga/pcsat.html

http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

http://129.247.179.151/dlr/gps/orion/index.html

http://www.igeb.gov/SPS-2001-final.pdf

http://www.norad.mil/

http://www.sew-lexicon.com/gloss_s.htm#SPACE_DETECTION_AND_TRACKING_SYSTEM

http://www.rc3.org/archive/inform/2/4.html

King-Hele, Desmond. Satellite Orbits in an Atmosphere: Theory and Applications. Glasgow: Blackie and Son Ltd., 1987.

Leung, Sunny. Email dated 27 Mar 2003.

Leung, Suuny; Montenbruck, Oliver; Bruninga, Bob. "GPS Tracking of Microsatellites – PCSAT Flight Experience." 22 October 2002.

Leung, Sunny. Email to m032712@usna.edu. Received 27 Mar 2003.

Tribble, Alan C. The Space Environment: Implications for Spacecraft Design. Princeton, NJ: Princeton University Press, 1995.

Vallado, David A. Fundamentals of Astrodynamics and Applications. New York: The McGraw Hill Companies, 1997.

Vallado, David A. Fundamentals of Astrodynamics and Applications, Second Edition. New York: The McGraw Hill Companies, 2001.

Appendix A: Matlab Programs

ReadGPS.m

```
%z is the length of the arrays
%data is the [z,4,12] array of all the data taken
%latS is the array of satellite latitudes
%longS is the array of satellite longitudes
%altS is the array of satellite altitudes
%timeS is the array of time in [hhmmss]
%seconds is the number of seconds at each time data point
%minutes is the number of minutes at each time data point
%hours is the hour of the time data poin t
%time is the array of time in [hh mm ss]

file='diskD';
if file=='diskD'
   data10=csvread('D:\Text files\PCSAT_020110_GPS.txt');
   data11=csvread('D:\Text files\PCSAT_020111_GPS.txt');
   data12=csvread('D:\Text files\PCSAT_020112_GPS.txt');
   data13=csvread('D:\Text files\PCSAT_020113_GPS.txt');
   data14=csvread('D:\Text files\PCSAT_020114_GPS.txt');
   data15=csvread('D:\Text files\PCSAT_020115_GPS.txt');
   data16=csvread('D:\Text files\PCSAT_020116_GPS.txt');
   data17=csvread('D:\Text files\PCSAT_020117_GPS.txt');
   data18=csvread('D:\Text files\PCSAT_020118_GPS.txt');
   data19=csvread('D:\Text files\PCSAT_020119_GPS.txt');
   data20=csvread('D:\Text files\PCSAT_020120_GPS.txt');
   data21=csvread('D:\Text files\PCSAT_020121_GPS.txt');
end

z=274;
dataGPS=zeros(z,4,12);
[m,n]=size(data10);
dataGPS(1:m,:,1)=data10;
[m,n]=size(data11);
dataGPS(1:m,:,2)=data11;
[m,n]=size(data12);
dataGPS(1:m,:,3)=data12;
[m,n]=size(data13);
dataGPS(1:m,:,4)=data13;
[m,n]=size(data14);
dataGPS(1:m,:,5)=data14;
[m,n]=size(data15);
dataGPS(1:m,:,6)=data15;
[m,n]=size(data16);
dataGPS(1:m,:,7)=data16;
[m,n]=size(data17);
dataGPS(1:m,:,8)=data17;
[m,n]=size(data18);
dataGPS(1:m,:,9)=data18;
[m,n]=size(data19);
dataGPS(1:m,:,10)=data19;
```

```
[m,n]=size(data20);
dataGPS(1:m,:,11)=data20;
[m,n]=size(data21);
dataGPS(1:m,:,12)=data21;



for i=1:12
  latStemp=fix(dataGPS(:,2,i)./100);
  latS(:,:,i)=(latStemp+(dataGPS(:,2,i)-latStemp.*100)/60).*pi/180;              %[rad] latitude of satellite
  longStemp=fix(dataGPS(:,3,i)./100);
  longS(:,:,i)=(longStemp+(dataGPS(:,3,i)-longStemp.*100)/60).*pi./180;      %[rad] longitude of satellite
  altS(:,:,i)=dataGPS(:,4,i)/1000;                    %[km] altitude of satellite
  timeSGPS(:,:,i)=dataGPS(:,1,i);                     %[hhmmss] time data was taken
  hoursGPS(:,:,i)=fix(timeSGPS(:,:,i)./10000);
  minutesGPS(:,:,i)=fix((timeSGPS(:,:,i)-hoursGPS(:,:,i).*10000)./100);
  secondsGPS(:,:,i)=fix((timeSGPS(:,:,i)-hoursGPS(:,:,i).*10000-minutesGPS(:,:,i).*100));
  timeSGPS(:,1,i)=hoursGPS(:,1,i).*60.*60+minutesGPS(:,1,i).*60+secondsGPS(:,1,i);  %time in seconds
  timeGPS(:,1,i)=hoursGPS(:,:,i);
  timeGPS(:,2,i)=minutesGPS(:,:,i);
  timeGPS(:,3,i)=secondsGPS(:,:,i);
  DateGPS(1,1,i)=i+9;                          %days; 10Jan2002=10 etc
  for j=1:z
    if hoursGPS(j,1,i)==0 & minutesGPS(j,1,i)==0 & secondsGPS(j,1,i)==0
      DaysGPS(j,1,i)=0;
    else
      DaysGPS(j,1,i)=DateGPS(1,1,i)+hoursGPS(j,1,i)/24+minutesGPS(j,1,i)/1440+secondsGPS(j,1,i)/86400;
%days
    end
  end
end

f=1;
for i=1:12
  for j=1:z
    if DaysGPS(j,:,i)~=0
      Day(f,1)=DaysGPS(j,1,i);
      lat(f,1)=latS(j,1,i);
      long(f,1)=longS(j,1,i);
      alt(f,1)=altS(j,1,i);
      hour(f,1)=hoursGPS(j,1,i);
      minute(f,1)=minutesGPS(j,1,i);
      second(f,1)=secondsGPS(j,1,i);
      seconds(f,1)=timeSGPS(j,1,i);
      time(f,:)=timeGPS(j,:,i);
      f=f+1;
    end
  end
end
g=f-1;
for j=2:g
  dtGPS(j,1)=(Day(j,1)-Day(j-1,1))*86400;
end
```

LST.m

```
%LST.m calculates the local sidereal time from the julian day

thetaG0=1.753637361;    %rad for 1Jan2002

thetaGGPS=thetaG0+1.002737909350795*2*pi.*(Day-1);   %rad
  thetaGPS=thetaGGPS+long;   %rad
  thetaGPS=mod(thetaGPS,2*pi);
```

GPSPosition.m

```
%Read data files into array data(z,4,12)
%turn data from time, lat, long, and alt arrays into position in IJK frame

ReadGPS;

mu=3.986004415e5;                                    %km^3/s^2

%Calculate site time data
LST;

%ecc is the eccentricity of the earth
%radius is the radius of the earth
%altA is the altitude of the ground point above spherical earth
%xR and zR are the distances from the center of the earth along
%  the x-y plane and in the z-direction, respectively
%RGroundPoint is  the position of the satellite ground point in XYZ
%RGPmag is the magnitude of the RGroundPoint vector

%ecc=0;
ecc=0.081819221456;
radius=6378.1363;   %km

xR=(radius./sqrt(1-ecc^2.*(sin(lat)).^2)+alt).*cos(lat);   %km
zR=(radius.*(1-ecc^2)./sqrt(1-ecc^2.*(sin(lat)).^2)+alt).*sin(lat);    %km
radiusE=sqrt(xR.^2+zR.^2);
rIJKGPS(:,1)=xR.*cos(thetaGPS);
rIJKGPS(:,2)=xR.*sin(thetaGPS);
rIJKGPS(:,3)=zR;
rIJKGPSmag=sqrt(rIJKGPS(:,1).^2+rIJKGPS(:,2).^2+rIJKGPS(:,3).^2);
```

NORADtoRV10.m

```
mu=3.986004415e5;                                   %km^3/s^2
Rearth=6378.1363;                                   %km

%read in classical orbital elements at starting point from line elements
i1=67.0482;                                         %inclination, degrees
bigomega1=214.3693;                 %right ascension of the ascending node, degrees
e1=0.0004963;                               %eccentricity, no units
littleomega1=276.9578;              %argument of perigee, degrees
M1=83.0837;                                 %mean anomaly, degrees
n1=14.2861444;              %mean motion, revs/day
year=2002;                                                  %epoch year, used for thetaG0
day1=010.94044338;                  %day and fraction of day of tracking data
ndot1=0.00000955*2;                 %first time derivative of mean motion, rev/day^2
ndot1=ndot1*2*pi/86400^2;  %rad/s^2


%start calculations
n1rad=n1*2*pi;                                      %rad/day
a1=(mu/(n1rad/86400)^2)^(1/3);      %km

%change degrees to radians
i1rad=i1*pi/180;
bigomega1rad=bigomega1*pi/180;
littleomega1rad=littleomega1*pi/180;
M1rad=M1*pi/180;

%calculate eccentric anomaly from mean anomaly
ETrad=-10000;
E1rad=M1rad;
while abs(E1rad-ETrad)>0.0001
  ETrad=E1rad;
  E1rad=M1rad-e1*sin(ETrad);
end
E1rad;
E1=E1rad*180/pi;
%calculate nu from eccentric anomaly
nu1rad=acos((cos(E1rad)-e1)/(1-e1*cos(E1rad)));
if E1rad > pi
   nu1rad=2*pi-nu1rad;
end
nu1rad;

%calculate r and v in PQW coordinates
p1=a1*(1-e1^2);
r1PQabs=p1/(1+e1*cos(nu1rad));                      %km
r1PQ=[r1PQabs*cos(nu1rad); r1PQabs*sin(nu1rad);0];          %km, [P;Q;W]
v1PQ=sqrt(mu/(p1)).*[-sin(nu1rad); (e1+cos(nu1rad));0];     %km/s, [P;Q;W]
v1PQabs=sqrt(v1PQ(1)^2+v1PQ(2)^2);                                          %km/s

%calculate in IJK coordinates
R=zeros(3,3);
R(1,1)=cos(bigomega1rad)*cos(littleomega1rad)-sin(bigomega1rad)*sin(littleomega1rad)*cos(i1rad);
R(1,2)=-cos(bigomega1rad)*sin(littleomega1rad)-sin(bigomega1rad)*cos(littleomega1rad)*cos(i1rad);
```

```
R(1,3)=sin(bigomega1rad)*sin(i1rad);
R(2,1)=sin(bigomega1rad)*cos(littleomega1rad)+cos(bigomega1rad)*sin(littleomega1rad)*cos(i1rad);
R(2,2)=-sin(bigomega1rad)*sin(littleomega1rad)+cos(bigomega1rad)*cos(littleomega1rad)*cos(i1rad);
R(2,3)=-cos(bigomega1rad)*sin(i1rad);
R(3,1)=sin(littleomega1rad)*sin(i1rad);
R(3,2)=cos(littleomega1rad)*sin(i1rad);
R(3,3)=cos(i1rad);
R;

rIJK1=R*r1PQ;                                   %km
vIJK1=R*v1PQ;                                   %km
rmag1=sqrt(rIJK1(1)^2+rIJK1(2)^2+rIJK1(3)^2);           %km/s
vmag1=sqrt(vIJK1(1)^2+vIJK1(2)^2+vIJK1(3)^2);           %km/s
X1=[rIJK1(1);rIJK1(2);rIJK1(3);vIJK1(1);vIJK1(2);vIJK1(3)];   %initial state vector for kalman filter
```

Predict.m

```
mu=3.986004415e5;                                   %km^3/s^2
Rearth=6378.1363;                                   %km
J2=0.0010826269;                                    %assume units as [ ]
%J2=0;
%Turn position and velocity vectors into classical orbital elements

h=cross(rIJK1,vIJK1);              %km^2/s
hmag=sqrt(h(1)^2+h(2)^2+h(3)^2);   %km^2/s
K=[0; 0; 1];
LoN=cross(K,h);                    %km^2/s
LoNmag=sqrt(LoN(1)^2+LoN(2)^2+LoN(3)^2);   %km^2/s

e1=((vmag1^2-mu/rmag1)*rIJK1-dot(rIJK1,vIJK1)*vIJK1)/mu;
e1mag=sqrt(e1(1)^2+e1(2)^2+e1(3)^2);
Energy=vmag1^2/2-mu/rmag1;         %km^2/s^2

a1=-mu/(2*Energy);                 %km
p1=a1*(1-e1mag^2);                 %km
mm1=sqrt(mu/a1^3);                 %rad/s

i1=acos(h(3)/hmag);                %rad
bigO1=acos(LoN(1)/LoNmag);         %rad
if LoN(2)<0
  bigO1=2*pi-bigO1;
end
littleO1=acos(dot(LoN,e1)/(LoNmag*e1mag));  %rad
if e1(3)<0
  littleO1=2*pi-littleO1;
end
nu1=acos(dot(e1,rIJK1)/(e1mag*rmag1));   %rad
if dot(rIJK1,vIJK1)<0
  nu1=2*pi-nu1;
end

E1=acos((e1mag+cos(nu1))/(1+e1mag*cos(nu1)));  %rad
if nu1 > pi
  E1=2*pi-E1;
end
M1=E1-e1mag*sin(E1);               %rad

%Update for perturbations, no drag when ndot1=0
a2=a1-2*a1/(3*mm1)*ndot1*dt;       %km
e2=e1mag-2*(1-e1mag)/(3*mm1)*ndot1*dt;    %[]
bigO2=bigO1-(3*mm1*Rearth^2*J2)/(2*p1^2)*cos(i1)*dt;   %rad
littleO2=littleO1+(3*mm1*Rearth^2*J2)/(4*p1^2)*(4-5*(sin(i1)^2))*dt;   %rad
M2=M1+mm1*dt+ndot1/2*dt^2;         %rad
p2=a2*(1-e2^2);                    %km
i2=i1;                             %rad

%calculate eccentric anomaly from mean anomaly
M2=mod(M2, 2*pi);
ET=-10000;
E2=M2;
```

```
while abs(E2-ET)>0.00000001
  ET=E2;
  E2=M2+e2*sin(ET);
end
%calculate nu from eccentric anomaly
nu2=acos((cos(E2)-e2)/(1-e2*cos(E2)));  %rad
if E2 > pi
   nu2=2*pi-nu2;
end
mm2=sqrt(mu/a2^3);

%Turn COEs into new position and velocity vectors
rPQW2=[p2*cos(nu2)/(1+e2*cos(nu2)); p2*sin(nu2)/(1+e2*cos(nu2)); 0];   %km
vPQW2=[-sqrt(mu/p2)*sin(nu2); sqrt(mu/p2)*(e2+cos(nu2)); 0];          %km/s

R=zeros(3,3);
R(1,1)=cos(bigO2)*cos(littleO2)-sin(bigO2)*sin(littleO2)*cos(i2);
R(1,2)=-cos(bigO2)*sin(littleO2)-sin(bigO2)*cos(littleO2)*cos(i2);
R(1,3)=sin(bigO2)*sin(i2);
R(2,1)=sin(bigO2)*cos(littleO2)+cos(bigO2)*sin(littleO2)*cos(i2);
R(2,2)=-sin(bigO2)*sin(littleO2)+cos(bigO2)*cos(littleO2)*cos(i2);
R(2,3)=-cos(bigO2)*sin(i2);
R(3,1)=sin(littleO2)*sin(i2);
R(3,2)=cos(littleO2)*sin(i2);
R(3,3)=cos(i2);

rIJK2=R*rPQW2;     %km
vIJK2=R*vPQW2;     %km/s
rmag2=sqrt(rIJK2(1)^2+rIJK2(2)^2+rIJK2(3)^2);      %km
vmag2=sqrt(vIJK2(1)^2+vIJK2(2)^2+vIJK2(3)^2);      %km/s

X2=[rIJK2(1);rIJK2(2);rIJK2(3);vIJK2(1);vIJK2(2);vIJK2(3)];
```

Kalman1.m

```
%Kalman1.m uses a kalman filter for the GPS data converted to position vectors.
%The initial position and velocity vectors are taken from the NORAD TLEs.

mu=3.986004415e5;                                   %km^3/s^2
Rearth=6378.1363;                                   %km

%First read in NORAD TLEs and transform the COEs into position and velocity
%Returns the initial state vector X1 ([rx;ry;rz;vx;vy;vz])
NORADtoRV10;
rIJK1;
vIJK1;
rmag1;
vmag1;

%Calculate the r vectors from the GPS data
%Returns the future state vector z ([rx;ry;rz])
GPSPosition;

%Initialize H vector
H=[1 0 0 0 0 0
  0 1 0 0 0 0
  0 0 1 0 0 0];

%Initialize P matrix
P=[1 0 0  0  0  0
  0 1 0  0  0  0
  0 0 1  0  0  0
  0 0 0 .0001 0  0
  0 0 0  0 .0001 0
  0 0 0  0  0 .0001];
%counter for filteredX and filteredb
c=1;

%for i=1:12
   for j=1:g
      if j>=619 & j<=633
         j=634;
      end
      if j==723
         j=724;
      end
      if j>=1190 & j<=1192
         j=1193;
      end
      if j>=1213 & j<=1215
         j=1216;
      end
      if j>=1257 & j<=1258
         j=1259;
      end
      if j==1085
         j=1086;
      end
      if j>=1250 & j<=1254
```

```
    j=1255;
  end
  if j>=1347 & j<=1354
    j=1355;
  end

%Determine the time difference between the last data point and the next
day2=Day(j,1);
dt=(day2-day1)*86400;     %seconds

%Predict the next position vector
%Find closest ndot from NORAD data
%Input rIJK1, vIJK1, rmag1, vmag1
%Return X2 ([rx;ry;rz]) (rIJK2, vIJK2, rmag2, vmag2)
Predict;

%Calculate phi matrix
phi(1,1)=1+(3*mu*dt^2*rIJK1(1)^2)/(2*rmag1^5)-(mu*dt^2)/(2*rmag1^3);
phi(1,2)=(3*mu*dt^2*rIJK1(1)*rIJK1(2))/(2*rmag1^5);
phi(1,3)=(3*mu*dt^2*rIJK1(1)*rIJK1(3))/(2*rmag1^5);
phi(2,1)=phi(1,2);
phi(2,2)=1+(3*mu*dt^2*rIJK1(2)^2)/(2*rmag1^5)-(mu*dt^2)/(2*rmag1^3);
phi(2,3)=(3*mu*dt^2*rIJK1(2)*rIJK1(3))/(2*rmag1^5);
phi(3,1)=phi(1,3);
phi(3,2)=phi(2,3);
phi(3,3)=1+(3*mu*dt^2*rIJK1(3)^2)/(2*rmag1^5)-(mu*dt^2)/(2*rmag1^3);

phi(4,1)=(3*mu*dt*rIJK1(1)^2)/(rmag1^5)-(mu*dt)/(rmag1^3);
phi(4,2)=(3*mu*dt*rIJK1(1)*rIJK1(2))/(rmag1^5);
phi(4,3)=(3*mu*dt*rIJK1(1)*rIJK1(3))/(rmag1^5);
phi(5,1)=phi(4,2);
phi(5,2)=(3*mu*dt*rIJK1(2)^2)/(rmag1^5)-(mu*dt)/(rmag1^3);
phi(5,3)=(3*mu*dt*rIJK1(2)*rIJK1(3))/(rmag1^5);
phi(6,1)=phi(4,3);
phi(6,2)=phi(5,3);
phi(6,3)=(3*mu*dt*rIJK1(3)^2)/(rmag1^5)-(mu*dt)/(rmag1^3);

phi(1:3,4:6)=[dt  0  0
              0 dt  0
              0  0 dt];

phi(4:6,4:6)=[1 0 0
              0 1 0
              0 0 1];
phi;
%Calculate predicted error covariance
  %Initialize Q matrix
Q=[.01 0  0  0   0   0
   0 .01 0  0   0   0
   0  0 .01 0   0   0
   0  0  0 .0001 0   0
   0  0  0  0  .0001 0
   0  0  0  0   0  .0001];
```

```
  %Calculate next Ptemp
Ptemp=phi*P*phi'+Q;

%Input next measured position: z=[rx,ry,rz]
zA=[rIJKGPS(j,1);rIJKGPS(j,2);rIJKGPS(j,3)];

%Calculate b vector: difference between predicted and measured position
b=zA-H*X2;

%Calculate Kalman gain, K
R=[.25  0   0
   0   .25  0
   0   0   .25];

tempA=H*Ptemp*H'+R;
tempB=inv(tempA);
Kgain=Ptemp*H'*tempB;

%Update state change estimate
dx=Kgain*b;

%Update Error covariance estimate
P=Ptemp-Kgain*H*Ptemp;

%Update state estimate
Xfinal=X2+dx;

%print filtered position and velocity with time
day2;
Xfinal;
r=sqrt(Xfinal(1)^2+Xfinal(2)^2+Xfinal(3)^2);
v=sqrt(Xfinal(4)^2+Xfinal(5)^2+Xfinal(6)^2);
r;
v;


%save data
filteredX(:,c)=Xfinal;
filteredb(:,c)=b;
filteredt(:,c)=day2;
c=c+1;

%reassign initial values for next iteration
rIJK1=Xfinal(1:3);
vIJK1=Xfinal(4:6);
rmag1=r;
vmag1=v;
day1=day2;

end
%end
```

Kalman1toCOE.m

```
count=size(filteredX);
count=count(2);

Position=sqrt(filteredX(1,:).^2+filteredX(2,:).^2+filteredX(3,:).^2);
Velocity=sqrt(filteredX(4,:).^2+filteredX(5,:).^2+filteredX(6,:).^2);

%calculate semi-major axis
        aK(1,:)=1./((2./Position(1,:))-(Velocity(1,:).^2/mu));
        aK=aK';
        for j=2:count
           dadtK(1,j)=(aK(j,1)-aK(j-1,1))./dtGPS(j,1);
        end
        mmK=sqrt(mu./aK.^3);
        hK=cross(filteredX(1:3,:),filteredX(4:6,:));
        hKabs=sqrt(hK(1,:).^2+hK(2,:).^2+hK(3,:).^2);
        vcrosshK=cross(filteredX(4:6,:),hK);
        for k=1:3
           eK(k,:)=(vcrosshK(k,:)./mu)-(filteredX(k,:)./Position(1,:));
        end
        eKmag=sqrt(eK(1,:).^2+eK(2,:).^2+eK(3,:).^2);
     for j=1:count
        if hKabs(1,j)~=0
           incK(1,j)=acos(hK(3,j)./(hKabs(1,j)));
        else
           incK(1,j)=0;
        end
        nodevectorK(:,j)=cross(K,hK(:,j));
     end
        nodeK(1,:)=sqrt(nodevectorK(1,:).^2+nodevectorK(2,:).^2+nodevectorK(3,:).^2);
     for j=1:count
        if nodeK(1,j)~=0
           bigomegaK(1,j)=acos(nodevectorK(1,j)/nodeK(1,j));
           littleomegaK(1,j)=acos(dot(nodevectorK(:,j),eK(:,j))/(nodeK(1,j).*eKmag(1,j)));
           nuK(1,j)=acos(dot(eK(:,j),filteredX(1:3,j))./(eKmag(1,j).*Position(1,j)));
        else
           bigomegaK(1,j)=0;
           littleomegaK(1,j)=0;
           nuK(1,j)=0;
        end
        if nodevectorK(2,j)<0
           bigomegaK(1,j)=2*pi-bigomegaK(1,j);
        end
        if eK(3,j)<0
           littleomegaK(1,j)=2*pi-littleomegaK(1,j);
        end
        if dot(filteredX(1:3,j),filteredX(4:6,j))<0
           nuK(1,j)=2*pi-nuK(1,j);
        end
     end

bigomegaK=bigomegaK.*180./pi;
littleomegaK=littleomegaK.*180./pi;
```

```
%filter out the erroneous data points: SMA below 7170 km and above 7186 km
aboveA=find(aK>7186);
belowA=find(aK<7170);
goodA=find(aK>=7170 & aK<=7186);
count2=size(goodA);
count2=count2(1);
aKfinal=aK(goodA);
Dayfinal=Day(goodA);
Positionfinal=Position(goodA);
Velocityfinal=Velocity(goodA);
incKfinal=incK(goodA);
mmKfinal=mmK(goodA);
eKmagfinal=eKmag(goodA);
nuKfinal=nuK(goodA);

figure;
scatter(Day,aK);
hold;
scatter(Dayfinal,aKfinal,'g');
xlabel('time [Day]');
ylabel('semi-major axis [km]');
```

COEtoDensity.m

```
%COEtoDensity.m takes the COEs from Kalman1toCOE.m and calculates the density using a running average
%over a certain time period.

%run Kalman1.m and Kalman1toCOE.m first

%calculate the relative velocity of the satellite to the atmosphere, with initial assumption
%that the atmosphere rotates with the earth
omega=7.292115856e-5;     %rad/s
omegaE=[0,0,omega];

Vrel=Velocityfinal.*(1-omega.*Positionfinal./Velocityfinal.*cos(incKfinal));
Vrel=Vrel.*10^3;      %m/s

%calculate the ballistic coeeficient
mass=12; %kg
cD=2.11;
CSArea=140.9;  %in^2
CSArea=CSArea*0.0254^2; %m^2
BC=mass/(cD*CSArea);

%calculate the average density
slopeAll=polyfit(Dayfinal,aKfinal,1);
densityAll=-slopeAll(1).*1000./86400.*(BC).*(1./((mean(Vrel)).^2)).*(mean(mmKfinal).*sqrt(1-
mean(eKmagfinal).^2)./sqrt(1+mean(eKmagfinal).^2+2.*mean(eKmagfinal).*cos(mean(nuK))));

%calculate the change in semi-major axis
p=1;
datapoints=1519;  %must be an odd number; is the number of points to be averaged across
middle=(datapoints+1)/2;      %this is the point for which the average decrease in semi-major axis will be
calculated
delta=middle-1;
order=1;
for j=middle:(count2-middle)
  slope(j,:)=polyfit(Dayfinal((j-delta):(j+delta),1),aKfinal((j-delta):(j+delta),1),order);
  da(j,1)=0;
  for loop = 1:order
    da(j,1) = slope(j,loop)*(order+1-loop)*Dayfinal(j,1)^(order-loop)+da(j,1);
  end
  Vmean(1,j)=mean(Vrel(1,(j-delta):(j+delta)));
  mmKmean(j,1)=mean(mmKfinal((j-delta):(j+delta),1));
  eKmagmean(1,j)=mean(eKmagfinal(1,(j-delta):(j+delta)));
  nuKmean(1,j)=mean(nuKfinal(1,(j-delta):(j+delta)));

%calculate the atmospheric density
  if eKmagmean(1,j)<0.1
    density(j,p)=-da(j,1).*1000./86400.*(BC).*(1./Vmean(1,j).^2).*(mmKmean(j,1).*sqrt(1-
eKmagmean(1,j).^2)./sqrt(1+eKmagmean(1,j).^2+2.*eKmagmean(1,j).*cos(nuKmean(1,j))));
  else
    density(j,p)=0;
  end
end
```

```
%plot
figure;
hold;
scatter(Day(middle:count2-middle,1),density(middle:count2-middle,1));
xlabel('time [Day]');
ylabel('density [kg/m^3]');

meandensity=mean(density(middle:count2-middle));
fprintf('\n The mean density using %i data points per density calculation is %i.', datapoints, meandensity);
```