# SYTEM DESIGN AND SOFTWARE ARCHITECTURE OF A DIFFERENTIAL GLOBAL POSITIONING SYSTEM FOR AN AUTONOMOUS ALL TERRAIN VEHICLE

by

Siddharth Haresh Ahuja

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2009

Approved by:

______________________________
Dr. James M. Conrad

______________________________
Dr. Bharatkumar Joshi

______________________________
Dr. Yogendra Kakad

# ABSTRACT

SIDDHARTH HARESH AHUJA. System Design and Software Architecture of a Differential Global Positioning System for an Autonomous All-Terrain-Vehicle. (Under the direction of Dr. James M. Conrad)

Over the years, the Global Positioning System (GPS) has evolved to become the de facto navigational and positional system and is used widely across the world. The system promises high accuracies if the navigational signals transmitted by the GPS satellites are observed accurately. Most GPS receivers produced by several manufacturers and used by civilians for navigational purposes have accuracies in the range of 3 to 10 meters. However several scientific and geological surveying applications need the system to provide an accurate positional result in the range of a few millimeters. The Differential GPS (DGPS) technique provides a user with the capability of calculating their position accurate within a few millimeters by using a fixed GPS receiver in addition to the roving one. This thesis describes the DGPS technique and provides a system design to implement this technique for an autonomous All Terrain Vehicle (ATV). The application at hand requires accurate real-time positional data and the architecture of the software used to calculate the same has been described in this report. Both the system design and the architecture are portable and can be used for any real-time positioning application.

# ACKNOWLEDGMENTS

The work contained in this thesis report and my tenure as a graduate student at UNC Charlotte would not have been possible to complete without the help, assistance and encouragement given to me by my advisor Dr James Conrad. The two years I have spent as a student in the department of Electrical and Computer Engineering have been the most enlightening and fruitful years of my life till this day. I take this opportunity to thank my advisor Dr James Conrad, my committee members Dr Bharat Joshi and Dr Yogendra Kakad, and all the faculty and staff of the department of Electrical and Computer Engineering for their advice and support.

To all new and existing members of the Embedded Systems Research lab, especially Richard McKinney, I thank you all for your timely help and support. The lab had been my second home for the better part of my time here at UNC Charlotte and the best ideas came from the debates and the discussions we had from time to time.

I would also like to thank Zapata Engineering for giving the Embedded Systems Research lab, the opportunity to design and implement the autonomous All Terrain Vehicle – the platform for my thesis.

Lastly, I thank my sister Jaya, my parents Haresh and Rashmi, and my friends at UNCC especially Siddharth Taduri and Sriram Rajagopalan for giving me the much needed push every once in a while to complete my thesis.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF EQUATIONS

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

The Global Positioning System (GPS), officially known as the NAVSTAR-GPS[1], is a worldwide, three dimensional, radio navigation and positioning system developed by the United States Department of Defense that reached its full operational capability in1995[2]. A constellation of 24 satellites carefully monitored and controlled by five major base stations in the US make up the space and the control segment of the GPS system. A GPS receiver that calculates its position by measuring its distance from all the satellites in its range makes up the user segment of the GPS system.

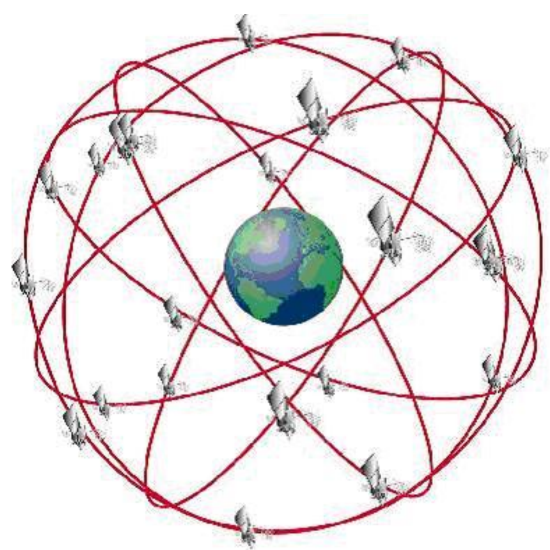

Figure 1-1: An illustration of the GPS constellation[3].

Originally intended and designed for military applications serving a total of 40,000 military users[2], the Department of Defense (DoD) approved the first basic architecture of the GPS system in 1973[2]. The first experimental Block-1 GPS satellite was launched in 1978[2] and the system reached operational capability in 1995.

## 1.1 Motivation

In the summer of 2008, the Embedded Systems Research lab at UNC Charlotte was tasked with designing and implementing an autonomous All Terrain Vehicle (ATV) that would tow a trailer carrying sensitive ground scanning equipment. The main objective of this vehicle was to autonomously scan the ground for unspent ammunition within a specified perimeter. To achieve autonomy the vehicle had to rely on an accurate real time positioning system that could provide generic position coordinates. The GPS came as a natural choice since GPS receivers are readily available and the system has the inherent ability to provide reasonably accurate positional data in a standard geodetic coordinate system (latitude, longitude and altitude).

An initial study on the accuracies of GPS receivers showed that accuracy came at a high cost. A standard GPS receiver similar to the one used by many of us in our cars for navigation had accuracy between 3 and 6 meters (approximately 10 – 20 feet). The autonomous ATV was to tow a trailer 3 feet wide, that had a ground-scanning-radar with a sensing coil of a few centimeters hence the accuracy requirement for the positioning system was set to ±0.5 meters.

Few companies manufacture GPS receivers with accuracies as high as 20 to 30cm. Those that do charge exorbitantly for these receivers and require a subscription to a GPS correction service for accuracy. The correction service provided by these manufacturers is available only in selected areas. The autonomous ATV finds it's applications in largely remote areas and hence it was required to achieve the specified accuracy using commercially available, low cost GPS receivers.

## 1.2 Evolution of the GPS

The GPS was originally intended to be used for military applications, but after a Korean Airline flight was shot down in the USSR in 1983[4], president Ronald Reagan issued a directive making the system accessible for civilian applications as a common good. Concerns that a potential enemy may use the system to target weapons against the US led the government to introduce satellite timing errors to reduce the positioning accuracy. This deliberate introduction of errors was called Selective Availability (SA)[5] and it degraded the accuracy of a standalone GPS receiver by 100 meters in the horizontal components and 156 meters in the vertical components. Although this range was good to navigate large distances, it did not allow precise navigation required by aircrafts to land and by ships to safely dock in bad weather. Moreover, using the differential GPS techniques, the errors due to SA could be eliminated by the common man. On the 1st of May 2000, U.S. President Bill Clinton's administration announced the decision to turn off SA for the common good[6].

Over the years the GPS has evolved to become a global standard in navigation and positioning. It is widely used in a plethora of applications such as positioning, meteorological surveys, climates studies, military, etc. In 2000, the US congress authorized the modernization[6] of the current GPS system. The modernization plan calls for new ground stations and new satellites with additional civilian, military signals and a new "Safety of Life" (L5) signal[7], to further improve the accuracy and availability of the GPS signals.

Several companies manufacture civilian GPS receiver units that are extensively used by consumers, industries and scientists. At the very basic, a GPS receiver measures the “pseudo-range” or the distance between the satellite and the receiver, to calculate its position on earth. Precise distance can be measured by carefully measuring the differences in time, carrier code and carrier phase, from transmission to reception. The technique to measure the pseudo-range differs from the type and manufacturer of the receiver. While most receivers operate on a single frequency (L1), a small number of receivers have the capability to receive GPS signals in both (L1 / L2) frequencies. This is mainly because the coarse acquisition (C/A) signal was not available on the L2 frequency until recently, and dual frequency receivers tend to be more costly than the single frequency ones.


Figure 1-2: Types of GPS receivers.

## 1.3 Previous solution to the problem

Several techniques have been proposed to reduce positional errors using standard GPS receivers. Some of these techniques are as simple as logging and averaging positional data at a fixed point for a given length of time. Other techniques use filtering techniques such as the Kalman filter to estimate the position based on a series of readings for a given point. Most of these techniques use data from a GPS receiver placed at a fixed point. For applications where mobile accuracy is important but not immediately required, erroneous GPS data is collected and then post processed. The post processing of this collected data yields positions relative to known points such as survey markers in the area.

## 1.4 Proposed solution to the problem

To get accurate real time positional data from a roving receiver, we need to process the data the very instant we receive it. This requires us to know the errors in the pseudorange measured by the receiver in advance. To measure or estimate the error we need to have another fixed receiver at a point whose positional coordinates are known. This receiver will also measure the pseudorange, but since the exact position of this receiver is known, the correct pseudorange for each satellite it sees can be calculated. The difference between the calculated and the reported pseudorange gives us the error in pseudorange measured for the given satellite. Once we deduct this error from the pseudorange measured (for the given satellite) by the roving receiver we obtain a fairly accurate positional fix. This technique or correcting the readings of a roving receiver using a stationary one is called Differential GPS (DGPS).

## 1.5 Organization of the thesis

This thesis report is organized into six chapters. Chapter 1 provides the introduction to the report and provides an insight to the problem at hand and the proposed solution. Chapter 2 provides the reader with an introduction to the working of GPS system which is required to fully understand the subsequent chapters. Chapter 3 describes the errors that cause the inaccuracies in the GPS and provides mathematical models to predict these errors. Chapter 4 provides the theory to implement the differential GPS (DGPS) technique. Chapter 5 contains information about the Garmin GPS 16 HVS / LVS receiver and provides information to interface it with the external world and communicate with it. Chapter 6 describes the design of the system with all interfaces and the architecture of the software to implement DGPS.

## 1.6 Thesis Statement

System design and software architecture of a differential global positioning system (DGPS) for an autonomous all terrain vehicle (ATV).

# CHAPTER 2: UNDERSTANDING THE GPS

Originally intended to be used only by the military, the GPS is a fairly complex but reliable system. To understand the working of such a system, one needs to first understand the components that form the system. This chapter describes these components, describes the links between these components and also explains the basic principle of calculating position.

## 2.1 The GPS segmentation

The Global Positioning System can be categorized into three segments; the space segment, control segment and the user segment. A brief description of the three segments is provided below.

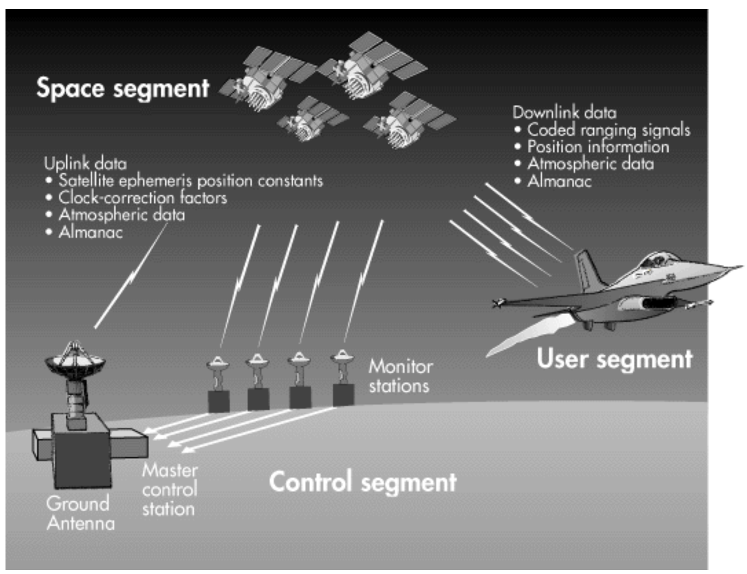


Figure 2-1: An Illustration of the 3 Segments of the GPS[8].

### 2.1.1 The Space Segment

The space segment of the GPS currently consists of 31 active satellites that constantly emit navigational signals in six orbital planes. One orbital plane has six active satellites and the other five orbital planes have five active satellites each[9]. These six orbital planes have an inclination of approximately 55° and are separated by 60° right ascension of the ascending node[10]. These satellites orbit around the earth at an altitude of approximately 20,200 Km and complete about two orbits in a day, thus a given satellite will be visible twice a day for varied lengths of time that depend on the users position and the satellite's orbit. The high altitude of the GPS satellites has its advantages. The satellites are well out of the earth's atmosphere; hence it is easier to maintain their orbits. A high orbiting altitude also makes several satellites (more than 9) visible from most points on earth.

### 2.1.2 The Control Segment

The Control Segment is made up of 12 GPS monitor stations that monitor the flight path, clocks and the navigational message transmitted by each GPS. These stations use precise radar to track the satellite's exact position, altitude and velocity[9]. The Master Control Station (MCS) is operated by the US Air Force at its Schriever Air Force Base in Colorado Springs, Colorado[9]. The MCS collects real time data from the monitoring stations and then send updates to the GPS satellites. The satellites use these updates to synchronize their clocks and also update the transmitted navigational signal. In addition to the MCS, the US Air Force operates five other monitoring stations in Hawaii, Ascension Island, Diego Garcia and Florida. Real time monitoring data is also collected from monitoring stations operated by the National Geospatial-Intelligence Agency (NGA)[11] in Washington, DC, England, Argentina, Ecuador, Bahrain and Australia. This enables the MCS in Colorado Springs to see each satellite from at least two monitoring stations.

### 2.1.3 The User Segment

The user segment of the GPS consists of the GPS receivers we use directly or indirectly in our daily lives. Several manufacturers produce GPS devices that are used in cars, boats, airplanes and watches. GPS receivers have found applications in cell phone devices as well. The FCC had made it mandatory for wireless service provides to implement the E-911 system by October 1st 2001[12][12], wherein wireless service providers have to provide emergency services a location of any person trying to call them. Most wireless phone manufacturers and wireless service providers use GPS receivers in the cell phones to obtain the user's position. These phones also provide regular positioning services (non-emergency) as an added feature to the cell phone user.

## 2.2 Basic principle for calculating position

The statement "A GPS receiver calculates its position by measuring its distance from the satellites it sees" is the simplest way of explaining the working of a GPS system. However calculating the position from the measured distances and measuring the distances are far more complex procedures that need very accurate timing. This section explains the principles used by a standard GPS receiver to calculate its position assuming the distances it measures are correct and error free. We also assume we have measured distances from at least three satellites.

### 2.2.1 Trilateration

Once the distances from all visible satellites have been measured, the GPS receiver uses a principle called Trilateration to calculate its position. Trilateration is a method of determining the relative position of objects using the geometry of triangles. In the case of a GPS system, the receivers calculate their position relative to the position of the visible satellites. Since the satellites traverse fixed orbits, a position relative to the satellites translates to a position that is fixed on earth.

To understand trilateration let us assume we have measured our distance P1 from a single satellite S1. Thus our position can be anywhere on the surface of a sphere that has the position of S1 as its center and a radius of P1.

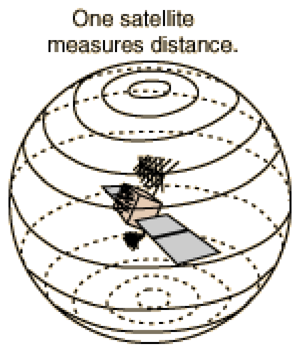


Figure 2-2: Distance measured from one satellite[14].

Now we know the distance P2 from another Satellite S2, we can draw a sphere of radius P2 around the position of S2. These two spheres will intersect forming a circle at the point of intersection. Now our position is narrowed down to a point that may lie anywhere on the surface of this circle.

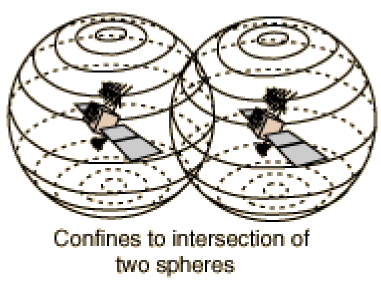


Figure 2-3: Distance measured from two satellites[14].
The Intersection is a Circular plane normal to the surface of the figure.

We now draw a sphere of radius P3 around the third visible satellite S3. This sphere not only intersects with the two spheres we had drawn before, but also intersects with the circle formed by the intersection of the first two spheres. This narrows down our position to two points on the surface of the circle, one is our true position and the other one is not.

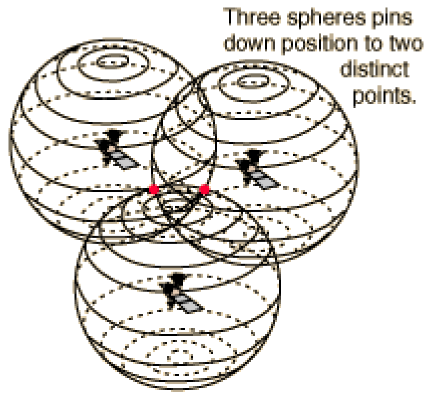


Figure 2-4: Distance measured from three satellites[14].
The two dots represent the two intersection points

To figure out which of the two points is our position we can use the distance measured from a fourth satellite, but it's likely that a sphere drawn around the fourth satellite will NOT intersect with any of these two points. This is because there are several factors that cause errors in the measured distances, and unless these distances are not compensated for these errors, all four spheres will not intersect at the same point. Thus the receiver looks for a single correction factor and subtracts this from its distance measurement yielding a single position.

In the steps above we drew spheres around the satellites to calculate our position, but to achieve calculating the position we need to know the position of the satellites as well. This information can be obtained from the Almanac or more precisely from the Ephemeris.

### 2.2.2 The GPS Almanac

The GPS Almanac is like a route or a flight plan that specifies the orbit for all satellites and helps us predict where a given satellite is going to be at a given time. The Almanac is updated daily and also contains general orbital and health information about all the satellites.

### 2.2.3 The GPS Ephemeris

The Ephemeris is like the Almanac, but it is real time positional information for a single satellite that is continuously transmitted by each satellite. Every satellite transmits its own ephemeris in a navigational message that is transmitted every 30 seconds. A receiver decodes this navigational message to extract the ephemeris data and from this data it can calculate the precise position of the satellite.

## 2.3 The navigational message

The navigational message contains data needed by the receivers to calculate its position. It serves as the link between the GPS space segment and the GPS user segment. Without the navigational message, the receiver will not know the position of the satellite and hence will not be able to carry out trilateration. This message is modulated on two other signals, the C/A code and the P(Y) code. The C/A code is further modulated onto the L1 frequency and the P(Y) code is modulated on both L1 and L2 frequencies. This section provides brief descriptions of the GPS carrier frequencies, the C/A code and the P(Y) code.

### 2.3.1 The C/A code

The Coarse Acquisition (C/A) Code, also known as the “Gold Code”, is a unique sequence with a length of 1023 bits and a frequency of 1MHz, transmitted by each satellite. Since all satellites transmit at the same frequencies, the C/A code helps differentiate the signals from different frequencies. Each satellite has its own unique sequence that looks like noise, but is not. This unique sequence is called the Pseudo-Random Noise (PRN). The navigational data is modulated on the C/A code which is in turn modulated onto the carrier signal L1.

2.3.2 The P(Y) code

The P code is similar to the C/A but this code has a length of seven days. This code is encrypted and is intended to be used by the military. When an Anti-Spoofing mode of operation activated, this code is known as the Y code. Thus this code is called the P (or Y), hence P(Y) code. The navigational signal is modulated on the P(Y) code as well.

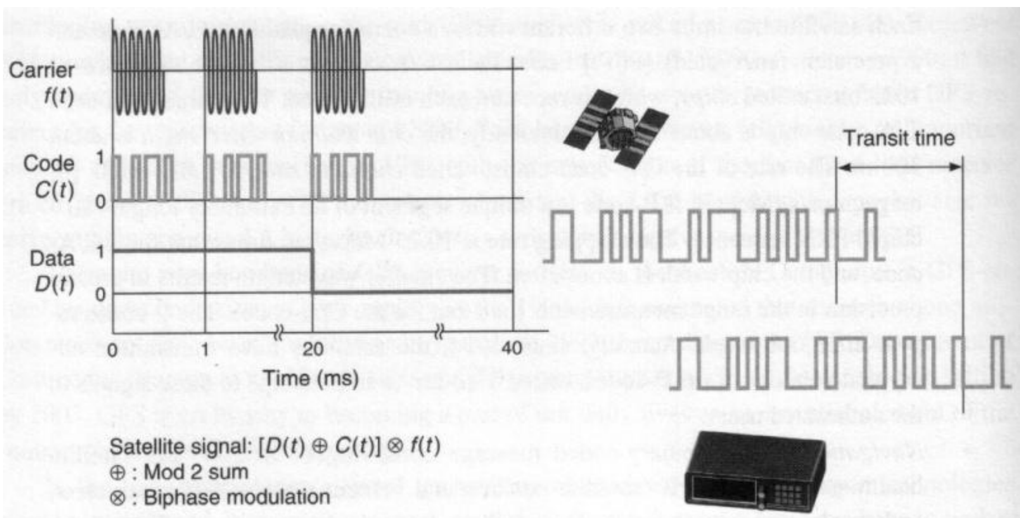


Figure 2-5 : Modulation on the carrier (L1 or L2)[15].

2.3.3 The navigational data

The navigational message *d(t)* is transmitted by each GPS satellite includes navigational data such as the time of week, GPS week number, satellite status and health, satellite clock bias, ephemeris data and almanac data[16]. This message is modulo-2 added to the C/A and the P(Y) codes resulting in two separate bit streams- C/A + *d(t)* and P(Y) + *d(t)*. While both the C/A + *d(t)* and the P(Y) + *d(t)* bit streams are transmitted on the L1 frequency, the MCS controls which bit stream is transmitted on the L2 frequency (generally P(Y)). The structure of the navigational data is as shown below:

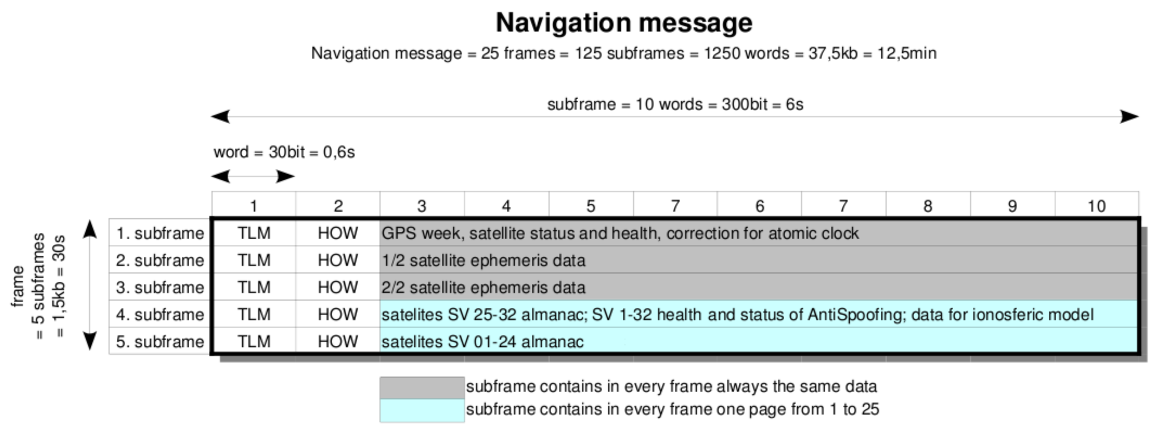


Figure 2-6 Structure of the navigation message[17].

### 2.3.4 The GPS Frequencies

All satellites broadcast navigational data on at least two frequencies (L1 and L2) that serve as the interface between the Space Segment and the User Segment. The satellites also transmit two other signals that are used by governmental agencies for varying applications. The newer satellites will also transmit a signal on the new L5 frequency that will be used to research and develop new Safety of Life applications. On April 10th 2009, the US Air Force successfully tested the transmission of the new L5 signal from a GPS IIR-20(M) satellite that was launched on March 24th 2009[18]. The frequencies and the signals transmitted on them are shown below:

Table 2-1: List of signals transmitted by GPS satellites.

| Signal | Frequency (MHz) | Contents |
|---|---|---|
| L1 | 1575.42 | C/A and P(Y). Future L1C civilian code planned. |
| L2 | 1227.60 | P(Y). Future L2C civilian code planned. |
| L3 | 1381.05 | Used by the Nuclear Detonation (NUDET) Detection System Payload (NDS) to enforce Nuclear Test Ban treaties. |
| L4 | 1379.913 | Used to study effects due to the ionosphere. |
| L5 | 1176.45 | Used to research and Develop new Safety of Life (SoL) applications[6]. |

# CHAPTER 3: SOURCES OF ERRORS

Since the operation of a GPS receiver depends on precisely calculating the pseudorange (distance from the satellite to itself), an error in calculation of the pseudorange translates to an error in calculating position. Most positional errors are caused due to errors in the pseudorange, hence it is important that we understand these errors so we can try to eliminate them. This chapter describes the more significant errors, their models and techniques to improve accuracy by eliminating these errors.

## 3.1 Sources of errors and error models

The transmitted signals travel through six layers of the earth's atmosphere before reaching the receiver. Several factors influence the speed of the transmitted signal during its journey from the satellite to the receiver. These factors cause errors in GPS measurements and hence reduce the accuracy of the system. To illustrate, consider a situation where the receiver is off by a thousandth of a second – at the speed of light this translates to a positional error of 200 miles. In this section we study the sources of errors and discuss methods to compensate for such errors.

### 3.1.1 Clock Errors

GPS satellites have precise atomic clocks whereas a standard GPS receiver's clock is not nearly as accurate. Since the pseudorange is calculated by measuring the time of flight of the GPS signal and multiplying this time with $c$, the speed of light, a small drift in the receiver's clock causes significant errors in the measurement of pseudorange, and hence the position. Clock bias errors are not unique to the receiver alone. The atomic clocks on the GPS satellites also drift slightly over time; however the base stations detect these drifts and transmit a clock bias to the user. The equation below shows a simple model for the clock error $\delta t$:

$$\delta t = b + dt + at^2$$

Equation 3-1: Model for clock error

Where $b$ is the Satellite clock bias, $d$ is the receiver's clock drift and $a$ is the acceleration of the related clock. The calculated clock error $\delta t$ is subtracted from the time of flight of the satellite's signal thus eliminating the positional error due to clock drifts.

### 3.1.2 Ionospheric Errors

The navigational signal transmitted by the satellites passes through six layers of the earth's atmosphere. If errors due to the atmosphere were negligible it could have been assumed that the speed of the transmitted signal is the same as the speed of light in vacuum. Although the ionosphere and the troposphere are the two layers of the atmosphere that can significantly impact the speed of the navigational signal as it passes through it. Thus we can get an erroneous positional reading if these delays or advances in the signal are not accounted for.

In general it is difficult to model the effect the ionosphere has on the signal since this error depends on complicated interactions between the geomagnetic field and solar activities[19].

The effects of the ionosphere on the transmitted signal depend on the frequency of the signal passing through it due to dispersion. The ionospheric delay is inversely proportional to the square of the frequency[20]. Thus this error can be eliminated by using a dual frequency receiver. The receiver will measure the pseudorange using both L1 and L2 frequency. An approximate estimation of the ionospheric error can be shown as[19]:

$$I = \left(\frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2}\right) * (P_{L2} - P_{L1})$$

Equation 3-2: Approximate model for Ionospheric error

Where $f_{L1}$ and $f_{L2}$ is frequency of carrier L1 (1575.76Mhz) and L2 (1227.60Mhz) and $P_{L1}$ and $P_{L2}$ are pseudoranges measured by observing L1 and L2 frequencies respectively.

### 3.1.3 Tropospheric Errors

The troposphere is the other layer of the atmosphere that causes significant delays or advances in the transmitted signal. Unlike the ionosphere, the delay or advance in the signal caused by the troposphere does not depend on the frequency of the signal passing through it[19], but it depends on the temperature, pressure, humidity and the location of the GPS receiver[19]. These factors can change rapidly and thus it is harder to accurately estimate the error due to the troposphere. An approximate model of the tropospheric errors can be shown as[19]:

$$\delta = \int (n-1)ds$$

Equation 3-3: Approximate model of the Troposheric error

Where $n$ is the refractive index of the troposphere, and the integration is taken along the path of signal transmission.

Other more accurate models such as the modified Saastamoinen Model[19] and the Modified Hopfield Model[19] can be used to accurately determine the errors due to the troposphere.

### 3.1.4 Multipath Errors

It is assumed that the transmitted signal will travel the shortest path from the satellite to the receiver. This path is the path traced by the line of sight between the receiver and the satellite, but sometimes the receiver may receive a scattered or a reflected version of the same signal. In urban environments it is common to receive a reflected signal without ever receiving the original one. The reflected signal has traveled a greater distance and thus taken longer to reach the receiver causing errors in the calculated position. Since the satellites are constantly moving, the multipath error can be generalized as a function of time.

In cases where the receiver receives both the original and the reflected signal, the second signal is generally rejected by the nature of the receiver's antenna. The GPS navigational signals are Right-Hand Circularly Polarized (RHCP); therefore conventional receivers use RHCP antennas and since the reflected signal has changed its polarity and has also lost its signal to noise ratio (SNR).

### 3.1.5 Other Errors

Many other factors and assumptions in calculations affect the position calculated. Some of these errors such as the effect of relativity on the atomic clocks, Sagnac distortion, and other sources of interference from unknown natural and artificial sources, are very small (to the order of $10^{-10}$) and hence can be neglected although there are models available to correct for these errors.

## 3.2 Methods to improve accuracy

As seen from the section above, several factors affect the accuracy of a GPS receiver. To improve accuracy one must eliminate all errors, but since we cannot predict most errors before making our calculation, a certain degree of inaccuracy is inevitable. This section describes methods used to reduce the error and thereby improve accuracy.

### 3.2.1 Differential GPS – Post Processing

A sum of all errors in the measured pseudorange from a receiver to a satellite can be accurately determined if the receiver is placed at a fixed position whose coordinates are known. This error can then be transmitted to other roving GPS receivers in the vicinity. This technique is known as Differential GPS (DGPS). DGPS can be implemented in real time or by post-processing. In the case of post processing data is collected from a single roving receiver and stored for further processing. Once the application using the roving receiver is complete, the data is processed to obtain precise positions that are relative to a known point in the area surveyed. The only drawback of post processing is the essential requirement of having a known point in the surveyed region. However, post processing requires only a single GPS receiver.

### 3.2.2 Differential GPS – Real Time Processing

The real time DGPS technique is the same in principle as the post processing technique but requires an additional receiver. One receiver is placed at a known point and the pseudorange to each satellite is calculated mathematically. This calculated pseudorange is then compared to the measured one and the difference in the two is the sum of all errors present. This error is transmitted to the roving receiver that subtracts it from its measured pseudorange, thus resulting in an accurate positional fix.

### 3.2.3 Augmentation

Augmentation is a method of implementing real time DGPS where a governmental or commercial agency carries out the tasks of the fixed receiver.

Augmentation can be implemented on the ground (Local Area Augmentation System – LAAS) wherein ground based augmentation stations calculate the errors and broadcast correctional signals in the VHF or the UHF bands. This augmentation data is good for all receivers within 20 – 30 miles of the LAAS ground station.

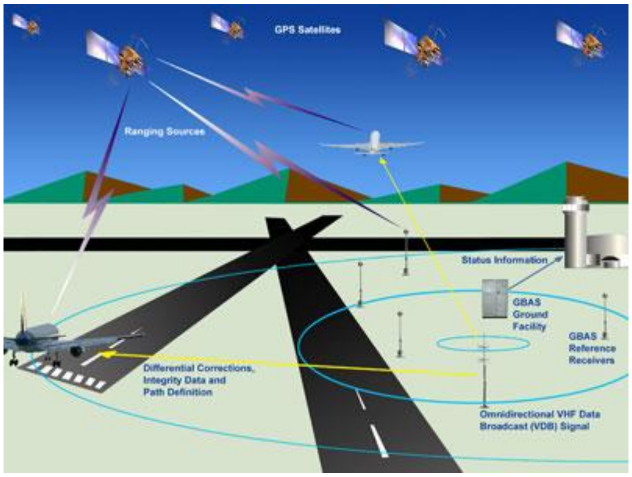


Figure 3-1: An illustration of the local area augmentation system (LAAS)[22].

Augmentation can also be implemented by using satellites (Satellite Based Augmentation System – SBAS). This uses the same principle as the LAAS but correctional data is transmitted by geostationary satellites operated by the Federal Aviation Administration (FAA)[21].


Figure 3-2: An illustration of the wide area augmentation system (WAAS)[23].

The original objective of implementing the WAAS was to enable aircrafts to rely on WAAS for all phases of the flight including precision approach. The WAAS collects observational data from several ground stations spread across the country and then sends this data to the geostationary satellites.

### 3.2.4 Use of Dual Frequency

Since Selective Availability (SA) was turned off, the largest contributor of inaccuracy in the GPS is the unpredictable error due to ionospheric delays. The ionospheric errors can be determined in real time by observing both L1 and L2 frequencies, but the only code that can be observed on both the L1 and the L2 band is the encrypted P(Y) code. Only military or “authorized” users have access to keys that are used to decrypt this code. However the GPS Joint Program Office has announced the availability of new signals for civilian use on both the L1 and L2 frequencies. Once these signals are made available, commercial GPS receivers will be able to accurately determine the error due to ionospheric effects. Refer to section 3.1.2 for more details.

# CHAPTER 4: DIFFERENTIAL GPS

A standard GPS receiver calculates its position by measuring the pseudorange from itself to the visible satellites. This calculation of the pseudorange is imperfect and contains several errors, the effect of which can be reduced and sometimes eliminated. Even after minimizing the errors, the positional accuracy of the receiver is in the order of a few meters and the position calculated varies with time (even if the receiver is fixed at a single point). Differential GPS is a technique to remove the errors in the measured pseudorange of a roving receiver by analyzing the pseudorange measured by a receiver that is placed at a known position. This chapter describes the theory of the Differential GPS technique and the different ways in which it can be implemented.

## 4.1 DGPS on position

This is the simplest DGPS technique where a GPS receiver is placed at a known position ($X_{known}$, $Y_{known}$, $Z_{known}$) and it continuously calculates its position ($X_{calc}$, $Y_{calc}$, $Z_{calc}$) by measuring the pseudoranges from all visible satellites. The receiver then deducts each coordinate of the calculated position from the coordinates of the actual position yielding the difference or the errors in the coordinates, $\Delta X$, $\Delta Y$ and $\Delta Z$.

The roving GPS receiver that calculates the real time position for an application is assumed to be in the vicinity of the fixed receiver at the known position. Since both receivers track the same satellites and the navigational signals observed by them travel the same path through the atmosphere, it can be assumed that the same errors affect both receivers with the same magnitude. Hence the fixed receiver continuously transmits the observed difference in position $\Delta X$, $\Delta Y$ and $\Delta Z$ and the roving receiver deducts these corrections from its own calculated positions.

This technique eliminates all atmospheric errors since both receivers are in the same geographical region and the atmospheric effects act on the measurements made by both receivers in the same manner. However, the clocks on both the receivers are not perfect and drift causing a clock bias. This clock bias is different for both the receivers, hence the position calculated by the receivers have different receiver clock bias errors and simply deducting the difference in the positional coordinates does not eliminate the clock errors. Therefore this technique will only be able to provide accuracy in the range of meters.

## 4.2 Code-Phase DGPS

In this technique, the pseudorange from all visible satellites is measured by a receiver that is placed at a known position and the position of the satellites is calculated from the ephemeris data that is transmitted by the GPS satellites every 30 seconds. Since the fixed receiver's position and the positions of all the visible satellites are known, the individual pseudorange from each satellite can be precisely calculated. Now for each satellite, the calculated pseudorange can be deducted from the measured pseudorange to determine the net error due to the atmosphere in the measured pseudorange for that particular satellite. This error will be the same for any receiver (roving or fixed) in the vicinity that is measuring the pseudorange from the given satellite.

The fixed receiver at the known position constantly calculates the error in the pseudorange for a given satellite and broadcasts this error wirelessly using a radio transmitter. This combination of the receiver, radio transmitter and / or a computing device is known as the base station. The receiver that receives these corrections and uses them to improve the accuracy of its position is known as the roving receiver.

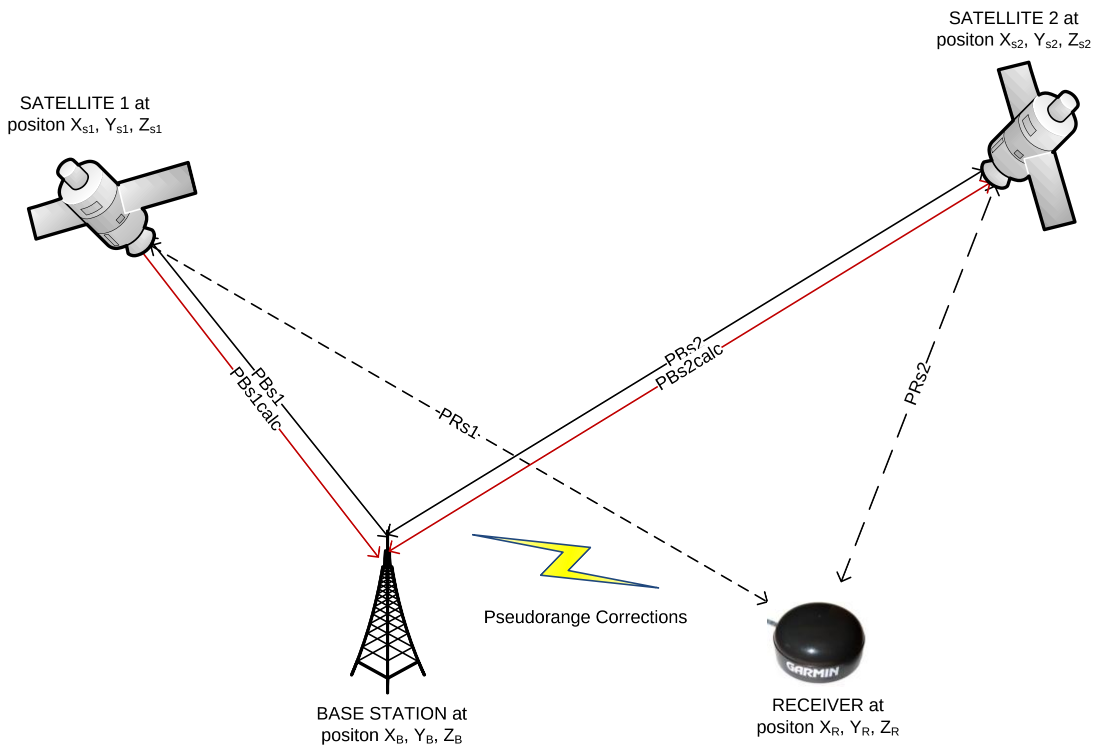


Figure 4-1: Concept of Code-Phase DGPS

The diagram above shows the base station measuring the pseudorange from two satellites S1 and S2. The pseudoranges measured for satellites S1 and S2 are $PB_{s1}$ and $PB_{s2}$ respectively.

The receiver is also measuring the pseudoranges from the same satellites S1 and S2. The pseudoranges measured by the receiver for the satellites S1 and S2 are $PR_{s1}$ and $PR_{s2}$ respectively.

Since the position of the base station ($X_b$, $Y_b$, $Z_b$) and the position of the satellite S1 ($X_{S1}$, $Y_{S1}$, $Z_{S1}$) is known, the calculated pseudorange from the base station to satellite S1 can be calculated as:

$$\boldsymbol{PB_{S1calc} = \sqrt{(X_b - X_{S1})^2 + (Y_b - Y_{S1})^2 + (Z_b - Z_{S1})^2}}$$

Equation 4-1: Calculation of range from receiver to satellite S1.

Similarly, the pseudorange from the base station to the satellite S2 can be calculated as:

$$\boldsymbol{PB_{S2calc} = \sqrt{(X_b - X_{S2})^2 + (Y_b - Y_{S2})^2 + (Z_b - Z_{S2})^2}}$$

Equation 4-2: Calculation of range from receiver to satellite S2.

Deducting the calculated pseudorange from the measured pseudorange for a given satellite gives us the error in the pseudorange for that particular satellite, and can be shown as:

$$\boldsymbol{dPB_{S1} = PB_{S1} - PB_{S1calc}}$$

Equation 4-3: Calculated error in observed and measured pseudorange.

The receiver in the vicinity of the base station uses this correction of pseudorange for the given satellite and calculates the corrected pseudorange as follows:

$$\boldsymbol{PR_{S1corrected} = PR_{S1} - dPB_{S1}}$$

Equation 4-4: Correction of pseudorange.

Similarly the base station transmits corrections in the pseudoranges for all visible satellites and the roving receivers use these corrections to correct their measured pseudoranges. The receivers then use the corrected pseudoranges to calculate their position using trilateration.

Like the positional DGPS method, this method also does not account for the receiver’s clock bias, but since pseudoranges are corrected directly errors due to the atmosphere and satellite clock bias are eliminated or mitigated. Thus this technique too, yields positional accuracy in the order of meters.

## 4.3 Carrier-Phase DGPS

The pseudorange is measured by observing the navigation message or the code (C/A code or P(Y) code) transmitted by the satellite. Thus the DGPS technique that uses corrections in pseudorange measurements to improve positional accuracy is called the Code-Phase DGPS. The Carrier-Phase DGPS method improves positional accuracy by observing the carrier signal and counting the number of wavelengths between the satellite and the receiver. The frequency of the L1 carrier is 1575.42MHz and this translates to a wavelength of 0.19 meters. Thus, if the number wavelengths are measured correctly, and all errors in the measurements are eliminated, it is possible to achieve an accuracy of 0.19 meters. The Garmin GPS 16 also provides the phase information, which tells us the number of wavelengths in fractions of the wavelength.

Moderate, good and high level of accuracy can be achieved while implementing carrier-phase DGPS techniques using single-difference, double-difference and triple-difference techniques respectively. These techniques are discussed below.

### 4.3.1 Single difference DGPS

This technique is similar to the positional DGPS technique or the code-phase DGPS technique. The number of wavelengths '*n*' of the L1 carrier between the receiver and the satellite are measured at both, the base station and the roving receiver.

Since it is not possible to determine the exact number of wavelengths $\lambda$ between the carrier and the receiver, the DGPS technique starts off by an approximation. This approximation is called the Ambiguity and is commonly depicted as '*n*'. The number of wavelengths measured or the ambiguity is not a whole number and the fraction of a wavelength is given by the phase φ. Thus the distance or range Φ also known as the observation for the given satellite S between the satellite and a receiver can be given by:

$$\Phi^{S} = n\lambda + \varphi$$

Equation 4-5: Calculation of ambiguity.

The initial estimate of '*n*' can be made if the position of the receiver and the observed satellite is known. The approximate number of wavelengths *n* can then be calculated by calculating the range between the receiver and the satellite and dividing it by the wavelength.

These observations can then be combined into single differences which are the differences of the observations measured by two receivers (base station *B* and roving receiver *R*) for a given satellite and can be defined as[24]:

$$S_1^{BR} = \Phi_1^{B} - \Phi_1^{R}$$

Equation 4-6: Calculation of single difference error for satellite 1.

Where the superscript indicates the receiver and the subscript indicates the number of the satellite being observed. Similarly the single difference between the base station receiver *B* and roving receiver *R* for observations measured for satellite 2 can be defined as[24]:

$$S_2^{BR} = \Phi_2^{B} - \Phi_2^{R}$$

Equation 4-7: Calculation of single difference error for satellite 2.

Using the single differences above, a fairly accurate position can be calculated. Using single differences eliminates the satellite clock errors and atmospheric errors as both the receivers observe the same satellite. However the receivers have their own clock biases and these are not corrected by the single difference method. Thus the accuracy achieved using carrier-phase single differences will be in the range of 10s of centimeters.

### 4.3.2 Double Difference DGPS

A double difference can be expressed as a difference between two single differences which is the difference in observation measured for two satellites by two receivers. A double difference for two receivers (base station *B* and roving receiver *R*) for two satellites (1 and 2) can be defined as[24]:

$$dd_{12}^{BR} = \left(\Phi_1^B - \Phi_1^R\right) - \left(\Phi_2^B - \Phi_2^R\right)$$

Equation 4-8: Calculation of double difference error for satellites 1 and 2.

Similarly the double difference for the same two receivers observing satellites 2 and 3 can be defined as[24]:

$$dd_{23}^{BR} = \left(\Phi_2^B - \Phi_2^R\right) - \left(\Phi_3^B - \Phi_3^R\right)$$

Equation 4-9: Calculation of double difference error for satellites 1 and 2.

A set of all double differences measured for all visible satellites is generated and a fairly accurate position can be calculated from this set of double differences.

Using double differences eliminates the receiver clock bias errors[19] a property lacked by the single difference technique. Thus the accuracy achieved by a double difference technique will be in the order of centimeters or sometimes even millimeters.

Both single and double difference techniques depend on accurately measuring the ambiguity ‘$n$’. Sometimes when the ambiguity is close to whole number, i.e. phase is close to 360° the receiver may lose or count a few extra cycles or wavelengths. This phenomenon of not resolving the ambiguity correctly is known as cycle slip. In the double differencing technique the cycle slip is the major cause for errors in positional calculations as a cycle slip cannot be detected.

### 4.3.3 Triple Difference DGPS

A triple difference is the difference between two double differences from adjacent epochs i.e. the difference between the double difference measured at time *t1* and the double difference measured at time *t2* and can be defined as[19]:

$$td_{12}^{BR}(t1, t2) = dd_{12}^{BR}(t1) - dd_{12}^{BR}(t2)$$

Equation 4-10: Calculation of triple differences

Triple differences are calculated for the same set of satellite pairs used to calculate the double difference. Since the triple difference technique is performed over two epochs, it gives us the ability to detect and hence eliminate a cycle slip thereby improving the achieved accuracy to the range of millimeters.

# CHAPTER 5: THE GARMIN GPS 16 XVS GPS RECEIVER

The embedded systems research group at UNC Charlotte chose to use the Garmin GPS 16 Receiver for the autonomous ATV project. The GPS 16 HVS and the low voltage GPS 16 LVS receivers are part of Garmin's OEM receivers range and cost under $200 each. The main selection criteria for selecting the receivers was the positional accuracy it provided without using correctional services, its cost and the navigational data it provides. The Garmin GPS 16 is comparatively cheap, provides pseudorange and carrier phase data once every second. The receiver also has the capability to implement DGPS as it can receive real-time WAAS or RTCM corrections. This chapter lists the features of the Garmin GPS 16, methods to communicate with the receiver and the Garmin protocol.


Figure 5-1: The Garmin GPS 16 HVS GPS receiver

## 5.1 Features of the receiver

The following features of the Garmin GPS 16 make it the ideal candidate for implementing Differential GPS on the autonomous ATV:

- The receiver has 12 channels and thus can track up to 12 satellites.
- Provides navigational output in NMEA mode or the proprietary Garmin mode.

- Provides real time receiver measurement data (pseudorange and carrier phase) once a second.
- Pseudorange data has a resolution of 1m and Carrier Phase has a resolution of 0.001 degrees.
- Provides real time ephemeris and almanac data.
- Flexible input voltage levels and low power consumption.
- All navigational and receiver measurement data is provided on a serial port with a configurable baud rate.

## 5.2 Communication

The Garmin GPS 16 has two serial ports (COM1 and COM2) and provides all navigational and receiver measurement data on COM1. This port is also used to receive commands from the external world. COM2 is used to receive DGPS correctional data and also provides an accurate pulse-per-second (PPS) signal. Navigational data is provided in two formats; the ASCII bases NMEA 0183 and the proprietary binary Garmin format.

In the Garmin mode, data to and from the receiver is transmitted in byte oriented packets specified by Garmin [20]. The format of the packet consists of a three byte header followed by the data which is then followed by a three byte trailer. The format of the packet can be shown as below:

Table 5-1: Generic Garmin Data Packet Format

| Byte Number | Byte Name | Description |
|---|---|---|
| 0 | Data Link Escape | Header indicating beginning of packet (0x10). |
| 1 | Packet ID | Hexadecimal value identifying packet type. |
| 2 | Size | Size of data / payload (n). |
| 3 to n-4 | Data / Payload | Data of length = n. |
| n-3 | Checksum | Checksum of bytes from byte 0 to byte n-4 |
| n-2 | Data Link Escape | Trailer indicating end of data and checksum. |
| n-1 | End of Transmission | Trailer indicating end of packet. |

The default baud rate settings to communicate with the receiver in the Garmin mode is 9600 baud, 8 data bits and no parity.

## 5.3 The Garmin phase data packet formats

The GPS 16 can be configured to output positional and receiver measurement data (satellite data, phase and pseudorange) once every second. This data is available only in the Garmin mode with phase data output enabled, and is transmitted in three separate packets with packet IDs 0x33, 0x35 and 0x72. The description of these packets and their format specification is shown below.

### 5.3.1 Position Record (Packet 0x33)

The position record has a packet ID of 0x33 and a size of 64 bytes. This packet provides position-velocity-time (PVT) information. Using information contained in this packet, one can calculate the UTC time and their position in three dimension coordinates. The data structure shown below describes the packet:

```
typedef struct                  //Structure for Packet 0x33
{
        float           alt;            //Altitude (meters)
        float           epe;            //Estimated positional error (meters)
        float           eph;            //Horizontal positional error (meters)
        float           epv;            //Vertical positional error (meters)
        int             fix;            //Type of fix
        double gps_tow;         //GPS time of week (seconds)
        double lat;             //Latitude (radians)
        double lon;             //Longitude (radians)
        float           lon_vel;        //Longitude velocity (meters/second)
        float           lat_vel;//Latitude velocity (meters/second)
        float           alt_vel;//altitude velocity (meters/second)
        float           msl_hght;       //Mean sea level height (meters)
        int             leap_sec;       //UTC leap seconds
        long            grmn_days;      //Garmin days (since December 31, 1989)
} cpo_pvt_data;
```

### 5.3.2 Receiver Measurement Record (Packet 0x34)

The receiver measurement record packet has a packet ID of 0x34 and a size of 226 bytes. This packet is contains the most important data essential for implementing DGPS. This packet contains pseudorange, phase and ambiguity data for each satellite. Since the GPS 16 is a 12 channel receiver, packet 0x34 contains 12 packets of the following kind:

```
typedef struct                          //One satellite record in 0x34
{
        unsigned long cycles; //Number of cycles - Ambiguity - n
        double          pr;             //Pseudorange (meters)
        unsigned int    phase;  //Phase from 0 - 359.999
        char                    slp_dtct;       //Flag indicating slip detect
        unsigned char snr_dbhz;         //Signal strength
        unsigned char svid;             //Space Vehicle Identification
        char                    valid;  //Flag indicating info validity
} cpo_rcv_sv_data;
```

In addition to the receiver measurements from 12 satellites, packet 0x34 also contains time data. This is provided as the number of weeks since January 6$^{th}$, 1980 and the number of seconds since 00:00:00 UTC of the previous Sunday. Packet 0x34 as a whole can be shown as:

```
typedef struct{
        double          rcvr_tow;     //Receiver time of week (seconds)
        int                      rcvr_wn;     //Receiver week number
        cpo_rcv_sv_data          sv[12]; //Array of 12 receiver measurements
} cpo_rcv_data;
```

5.3.3 Satellite Data Record (Packet 0x72)

The satellite data record packet has a packet ID of 0x72 and a size of 84 bytes. This packet contains elevation, azimuth, SNR and health information for a satellite. Since the GPS 16 is a 12 channel receiver, packet 0x72 contains for 12 channels. The data structure below represents the data for a single satellite:

```
typedef struct
{
        unsigned char svid;          //Space Vehicle Identification
        unsigned int  snr;           //Signal-to-noise ratio
        unsigned char elev;          //Satellite Elevation in degrees
        unsigned int  azmth; //Satellite Azimuth in degrees
        unsigned char status;  //Status bit field
} cpo_sat_sv_data;
```

The data structure that is used to hold all data from packet 0x72 contains an array of 12 cpo_sat_sv_data and can be shown as below:

```
typedef struct
{
        cpo_sat_sv_data          sv[12]; //Array of 12 cpo_sat_sv_data
} cpo_sat_data;
```

5.4 The Garmin Ephemeris packet format

The ephemeris of a satellite provides orbital information about a satellite. The exact position of a satellite at the time of transmission can be calculated from the ephemeris data. The GPS 16 provides the ephemeris data only on request and after a series of handshaking transmissions between the receiver and the host machine. The sequence of events that occurs to request an ephemeris packet can be found in the GPS 16/17 technical specifications by Garmin International. The ephemeris data record packet has a packet ID of 0x35 and a size of 120 bytes. The structure of an ephemeris packet is similar to that of the receiver measurement packet (packet 0x34) and contains ephemeris data for 12 satellites. The following data structure represents the format of a single ephemeris record in packet 0x35:

```
typedef struct                          //Ephemeris Record from
{
        signed int      wn;             //Week Number
        float           toc;            //reference time of clock
        float           toe;            //reference time of ephemeris parameters
        float           af0;            //Clock correction coeff - group delay
        float           af1;            //Clock correction coefficient
        float           af2;            //Clock correction coefficient
        float           ura;            //user range accuracy
        double          e;              //eccentricity
        double          sqrta;          //square root of semi-major axis
        double          dn;             //mean motion correction
        double          m0;             //mean anomaly at reference time
        double          w;              //argument of perigee
        double          omg0;           //right ascension
        double          i0;             //inclination angle at reference time
        float           odot;           //rate of right ascension
        float           idot;           //rate of inclination angle
        float           cus;            //argument of latitude correction, sine
        float           cuc;            //argument of latitude correction, cosine
        float           cis;            //inclination correction, sine
        float           cic;            //inclination correction, cosine
        float           crs;            //radius correction, sine
        float           crc;            //radius correction, cosine
        char            iod;            //issue of data
        char            svid;           //ID number of the SV
        char            valid;  //Data validity flag
        } SDM_spc_sph_type;
```

# CHAPTER 6: SYSTEM DESIGN AND SOFTWARE ARCHITECTURE

The main goal of this thesis is to design a low cost, yet accurate DGPS system for an autonomous All Terrain Vehicle (ATV). The hardware design uses commercially available, low cost components and the software is designed to run on an open source platform, thus eliminating the high costs of licensed software. This chapter describes the high level system design, the hardware interfaces and the software architecture for the DGPS system.

## 6.1 System Design

The principle of a DGPS system requires pseudorange and phase measurements from the visible satellites at a known location. Therefore, for the system to function, a known position with positional coordinates accurate to ±5mm is an essential requirement. A GPS receiver is placed at this position and its measurements of the pseudorange and phase for all visible satellites is transmitted by a radio transmitter via a wireless medium to the roving ATV. The GPS receiver and the radio transmitter form the Base Station.

On the ATV side, the GPS receiver is mounted as high and as close to the front of the vehicle as possible. A Wi-Fi enabled computer on the ATV runs the DGPS software and is connected to the GPS receiver on the ATV via a single serial port. A radio receiver that receives the signal transmitted by the base station is also mounted on the ATV and connected to the computer via another serial port. The computer will only need to have the Wi-Fi enabled if the supervisory station is implemented.

An optional supervisory station consists of a computer and an IEEE 802.11 compliant Wi-Fi router. The computer onboard the ATV will send the supervisory station all telemetry data including the ATV's position and GPS receiver measurements from the base station and the roving receiver. This positional data can then be used to plot the ATV's position on a map. An illustration of the system design is shown below.

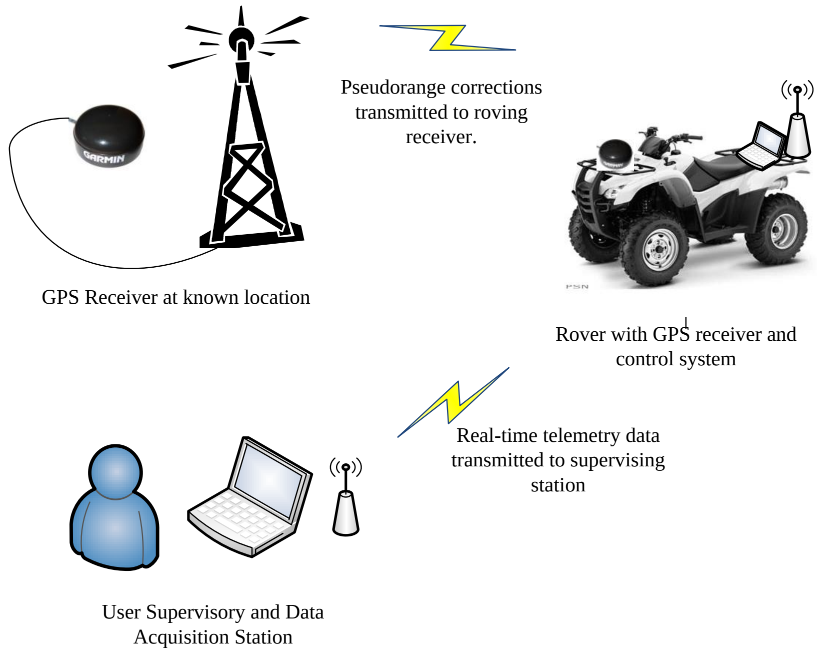


Figure 6-1: An illustration of the DGPS system design for an autonomous ATV

## 6.2 Hardware Design and Interfacing

As stated earlier, the hardware design uses commercially available low cost components to achieve the design goal. The two GPS receivers, one way radio and a computer are the essential components for the system. The IEEE 802.11 compliant Wi-Fi router and another computer for the supervisory station are optional and out of the scope of this design. This section describes the hardware used and methods to interface all the hardware components with the system.

### 6.2.1 GPS Receiver Interface

The GPS receivers transmit and receive data from the external system on a single serial port. When operating in Garmin mode, the default settings for the serial port is 9600 baud, 8 data bits, one stop bit and no parity. The data is transmitted in the RS 232 format and therefore a voltage pump such as the MAX232 is not required. The following illustration shows the wiring diagram to interface the GPS 16 with standard D B-9 connector on the computer’s or the radio’s serial port.

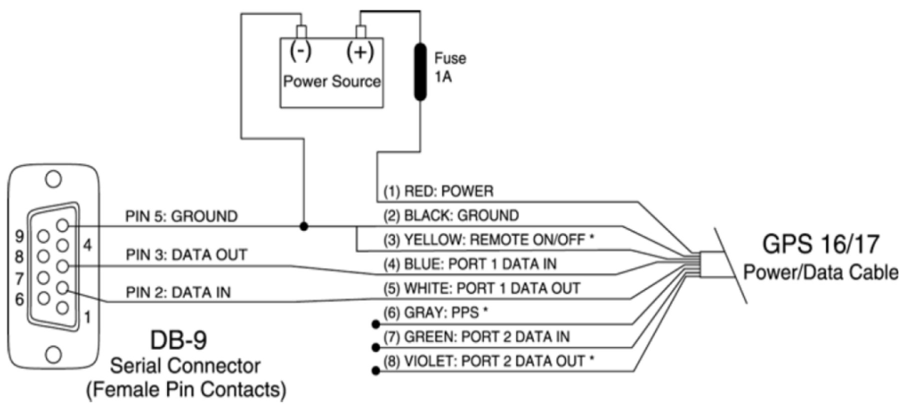


Figure 6-2: Wiring Diagram to interface the GPS 16 receiver with a computer

## 6.3 Software architecture

While the hardware is responsible for providing data to the system, the software is responsible for using the data sent from the receiver to implement DGPS and calculate the accurate position of the roving receiver. To get the best results the software must have a robust and efficient architecture that can complete calculating the position before new data is available (<1s). This design suffices with implementing sequential, single threaded execution. However the calculation can be sped up by using multi threaded implementation which can be implemented in the future.

### 6.3.1 Software Requirements

The following requirements have been identified for the DGPS software:

REQ 10. In order for this design to be cost effective, the software should be implemented on an open source operating system like LINUX.

REQ 20. Once invoked, the software shall run continuously with minimal intervention from the user.

REQ 30. The software shall provide positional coordinates as output on the standard output and in inter process messages via the operating system to any process that requires it.

REQ 40. The software shall take as input, pseudorange and phase measurement from each visible satellite every second from both receivers, and ephemeris data every 30 seconds from the base station receiver.

REQ 50. The software shall be 100% compliant with the Garmin Protocol when communicating with the GPS receivers.

REQ 60. The software should request and receive all data to and from the Garmin receivers on separate serial ports.

REQ 70. The software should complete all necessary calculations and provide an accurate position fix in less than one second so that it is ready to calculate the position when the new data comes in from the receiver.

REQ 80. The start-up time (time required for first fix) should not exceed 60 seconds.

REQ 90. The software should detect bad positional calculations and inform the user.

REQ 100. If three or more consecutive positional fixes are bad fixes, the software shall send an error message to the standard output and in an inter process messages to the processes using the output data.

REQ 110. The software should log all data (raw data, pre-processed and post-processed positional fixes) throughout the entire execution.

### 6.3.2 Software Class Diagram

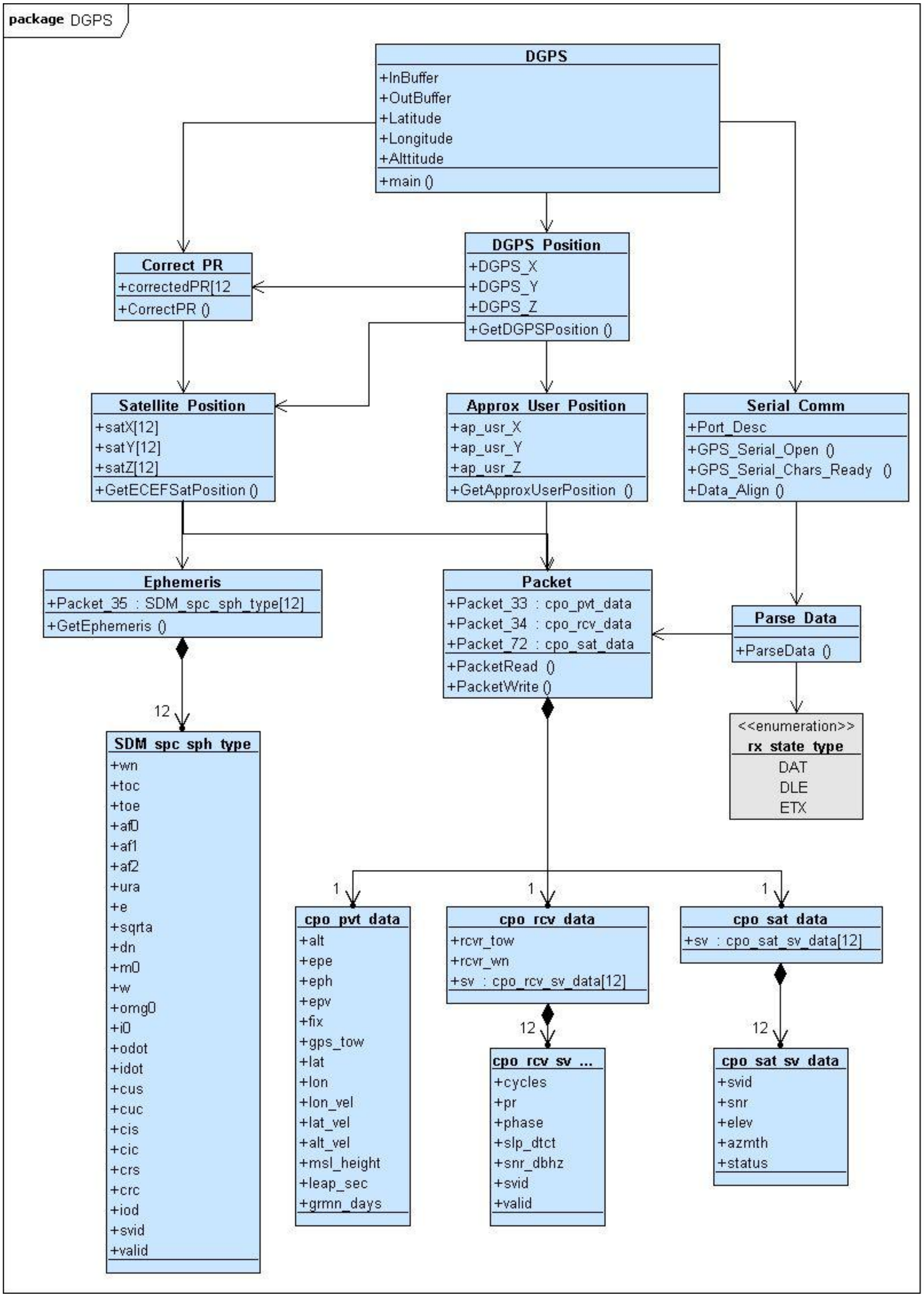


Figure 6-3: UML class diagram illustrating the over-all software architecture.

Although UML class diagrams such as the one above, are intended for object oriented programming (OOP) languages like C++ or JAVA, this diagram can also be used to describe the architecture of a non – OOP language such as C. Every class can be looked at as a combination of a header file and a source file. The classes that do not have any member functions (shown in the lower half of the diagram) can be looked at as data structures that can be defined in the header files of the classes that own them. Classes with member functions will contain method and variable definitions in the source file and method declarations in the header files.

The choice of the programming language is left to the implementer. This design provides the overall picture of the software that can be implemented in any language on any platform.

## 6.4 Software design

The software is designed to be modular so that the tasks are executed by independent functions that may get their parameters from the calling functions, but do not depend on any global parameters for their internal operation. The modular design allows the design to be implemented in single threaded or multi-threaded environments. Once invoked the routine will start with the main() function that calls other tasks sequentially. The serial ports are initialized first, followed by file descriptors that enable logging data to files. If any of these tasks fail, the software quits and informs the users about the errors it encountered. Once the serial ports and the files have been initialized, the software starts by requesting both units for positional data. If valid positional data is received from both units, the software begins calling the tasks to calculate the position of the satellites, correct the pseudorange and implement DGPS. If the receivers do not respond with valid positional data, the software resends the request and carries on resending the request for 60 seconds at intervals of 5 seconds or until valid positional data is received. This procedure is elaborated in detailed design of the DGPS class.

### 6.4.1 The Serial Communication Class

This class contains separate buffers for holding incoming and outgoing serial data. Member functions of this class write data to or read data from the serial port to / from the respective buffer.

### 6.4.2 The Parse Data Class

This class contains a single member function that parses the raw data in the input buffer and extracts the useful data by stripping the headers and the trailers of each packet. All data sent to or received from the GPS 16 receiver follows the Garmin protocol. The first byte in the packet is a header byte known as the Data Link Escape (DLE) byte and has a value of 0x10. This is followed by the packet ID and the size of the packet. The data of the given size is then transmitted followed by a checksum. The end of the packet is marked by a two byte trailer consisting of a DLE byte followed by an "End of Transmission" (ETX) byte with a value of 0x03.

It is possible that regular data in the packet may take up a value of 0x10 and the receiver may incorrectly interpret that as the end of the transmission. That's why an extra or "Stuffed" DLE byte is added to escape the value of 0x10, should it occur anywhere in the packet frame. The flowchart below explains how the header and the trailer are stripped off a received packet.

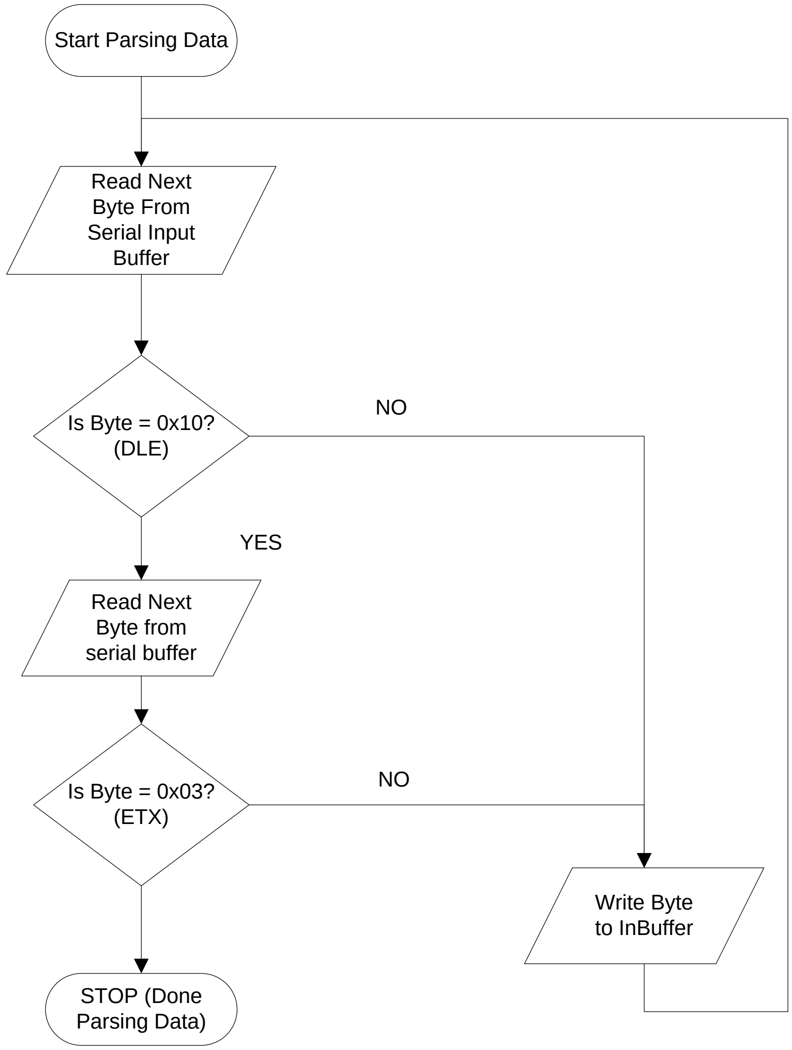


Figure 6-4: Flowchart for the parse data function.

### 6.4.3 The Packet Class

The packet class holds positional and receiver measurement data packets received from the receivers. These classes are cpo_pvt_data (packet 0x33), cpo_rcv_data (packet 0x34), cpo_sat_data (packet 0x72) and SDM_spc_sph_type (one out of 12 packets in Ephemeris). As stated earlier, in a non-OOP implementation, these classes are merely data structure types. The contents and order of contents of these data structures have been described by Garmin in the GPS 16/17 Technical Specifications []. The structure cpo_rcv_data contains pseudorange and phase measurement information for the visible satellites. Since the receiver is capable of looking at 12 satellites, i.e. it has 12 channels, the structure cpo_rcv_data actually contains an array of 12 cpo_rcv_sat_data, each of which contains pseudorange and phase measurement data for each satellite / channel. Similarly the structure cpo_sat_data contains an array of 12 cpo_sat_sv_data structures.

The member functions of this class read the specified packets from the raw data stored in a temporary buffer, and also construct packets in the Garmin format that need to be sent to the Receiver.

### 6.4.4 The Ephemeris Class

The Ephemeris class holds all ephemeris data received from the GPS receiver. Since the receiver is a 12 channel receiver, the Ephemeris class or structure holds data for 12 satellites / channels. This data is stored as an array of 12 SDM_spc_sph_type structures.

The sole member function of this class is used to get the Ephemeris data from the receiver. The procedure to request ephemeris data from the receiver consists of a sending and receiving a series of command and acknowledge messages before the ephemeris data is finally sent by the receiver. The host requests the receiver for the ephemeris data and the receiver acknowledges this request. The receiver then sends a packet containing the number of packets to expect. The host must then acknowledge the receipt of this packet before the receiver starts sending the ephemeris data over to the host.

This function must be invoked once every thirty seconds to ensure the ephemeris data is up-to-date. The flowchart below depicts this series of events and shows the behavior of the GetEphemeris function.

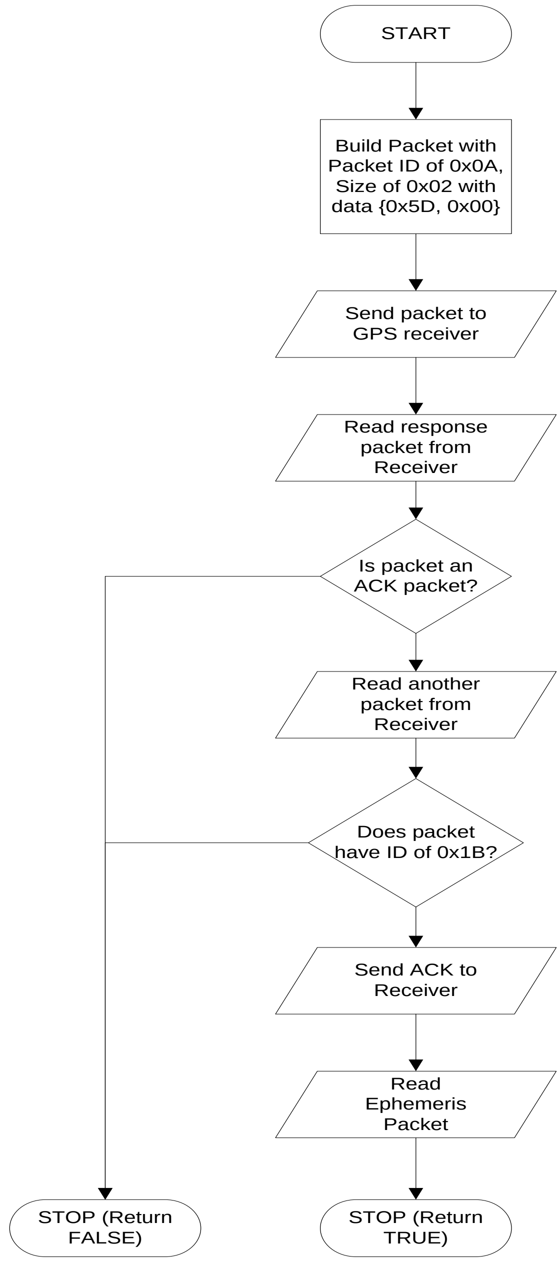


Figure 6-5: Flowchart for getting ephemeris data from the GPS receiver.

### 6.4.5 The Satellite Position Class

This class holds the most recent position of all visible satellites. Member functions of this class take the ephemeris data as input and then calculate the position of the visible satellites from this data. The function provides the position coordinates of the visible satellites in the Earth-Centered-Earth-Fixed ECEF coordinate system.

### 6.4.6 The Approximate User Position Class

This class holds the approximate or unprocessed position of the roving receiver. All calculations used to implement DGPS start off by assuming an approximate position and then correct the pseudoranges to improve the accuracy of the position. The member function of this class merely extracts and stores the positional data from Packet 0x33 once every second.

### 6.4.7 The Correct Pseudorange Class

This class holds two sets of pseudoranges for each visible satellite. One set is measured directly by the receiver and the other set is the pseudoranges corrected for errors such as clock errors, ionospheric and tropospheric errors. The sole member function of this class takes the unprocessed set of pseudoranges as the input and then deducts the calculated errors. Like other functions, this function too must correct the pseudoranges once every second.

### 6.4.8 The DGPS class

This class performs the core function of the software – the actual DGPS implementation. The member function to calculate the DGPS correction gets the measured pseudoranges from the base station and compares them with the calculated pseudorange for the given position of the base station. In case of a non-OOP implementation this class describes the application as a whole.

The member function to calculate the DGPS position takes the calculated DGPS correction and deducts this correction from the pseudorange of every visible satellite. The function then calculates the position of the roving receiver using the corrected pseudoranges. The flowchart below shows the working of the function.

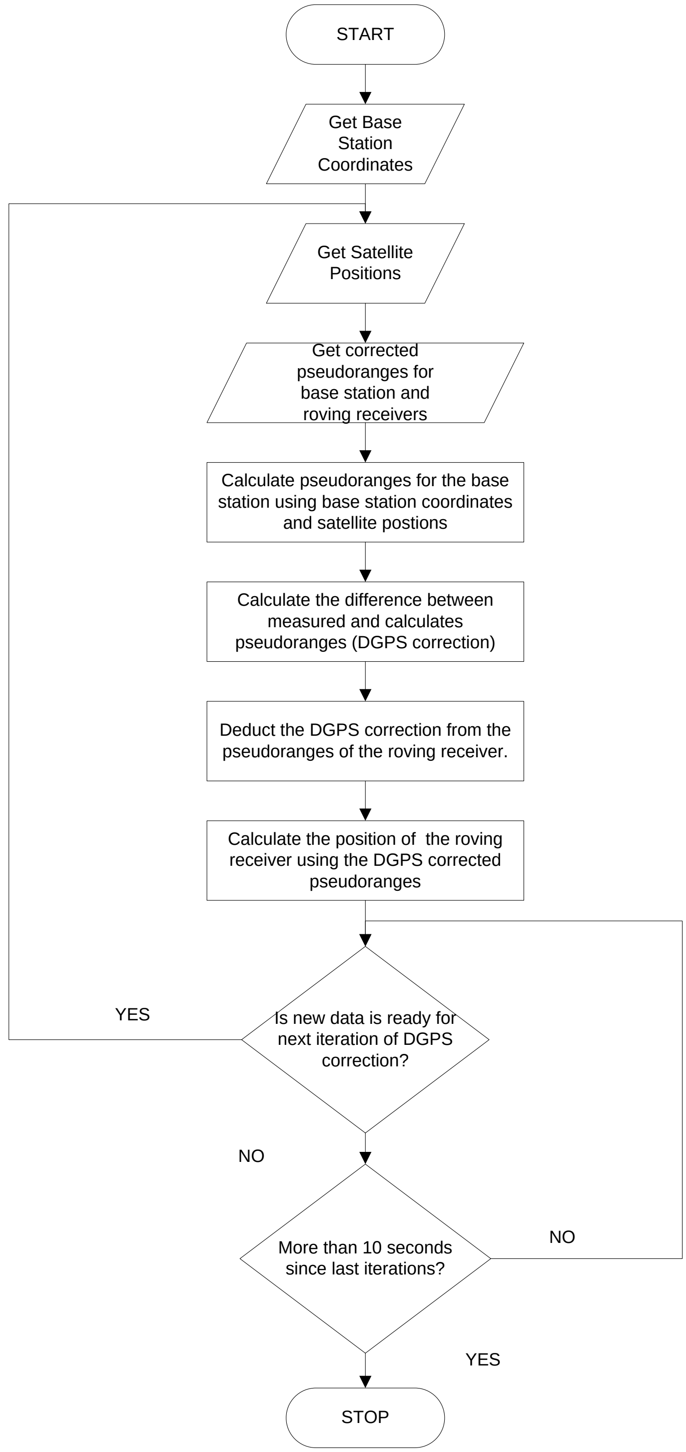


Figure 6-6: Flowchart for the main DGPS application.

# CHAPTER 7: CONCLUSION AND FURTHER WORK

The correct implementation of the hardware and software design using the Differential GPS technique will yield a real-time positional fix accurate within 10cm with a Circular Error Probability (CEP) of 90%, refreshed at a rate of 1 Hz. As a part of this thesis, the software has been implemented up to a point and the implementation is described below.

## 7.1 Implementation

The software design has been implemented up to a point in C on a LINUX platform. All functions to interface with the GPS receiver use the serial port and hence use the “Portable Operating System Interface (for Unix)” better known as the POSIX interface. The detailed implementation of the major functions is described below. The source code for the implemented software is attached along with this report in a CD and can be easily compiled and linked by executing the makefile on a POSIX compatible LINUX box that has the GCC compiler installed and at least three serial ports.

### 7.1.1 Serial Communication

Three functions handle all operations that require initializing, reading or writing to a serial port. The GPS_Serial_Open function takes the name of a serial port as a input and provides two outputs – a handle or a descriptor to the port and a Boolean value indicating success or failure in opening the port. If the function successfully opens the port it returns a value of 1 along with the port handle.

The GPS_Serial_Chars_Ready function takes the port handle as an input and polls the serial port to see if any characters are ready for reading. If a character is ready to be read out of the buffer, the function returns a value of 1 else it returns a value of 0.

Since the Garmin GPS 16 is in phase mode and is continuously transmitting data, it is important for the software to begin reading and using the data from a known point so that no incomplete packet is read in. The DataAlign function calls the GPS_Serial_Chars_Ready function and waits for a character to appear on the serial port buffer and reads in the byte when ready. The function first waits for a byte with a value of 0x10 to appear on the port. Once found, the function reads the next byte and checks if this byte has a value of 0x03. If the byte has the value of 0x03 it means we have read a 0x10 followed by a 0x03 – the terminating sequence of a Garmin packet. The 0x03 byte is returned to the caller that can now align itself with the first byte of the next packet.

### 7.1.2 Parsing Data

The data transmitted by the Garmin receiver is packed into a packet with heading and trailing bytes that indicate the start, size, checksum and the end of data (refer Table 5-1). This data needs to be processed to first strip off the heading and the trailing bytes. The PacketRead function first calls the DataAlign function to wait for the header of the next packet. Once aligned, the function reads the incoming bytes from the serial port and strips of the heading and the trailing bytes as per the design (refer Figure 6-4). The data along with the packet ID, size and checksum (a significant packet) is then stored in a global input buffer with a maximum size of 256 bytes called the InBuffer. This function takes a file pointer as an input to log data and returns a boolean value of success or failure.

The ParseData function is called right after the PacketRead function and parses the packet in the InBuffer to populate the relevant data structure. The function checks the packet ID field of the packet in the InBuffer and then calls the relevant function to read bytes of data from the buffer and assign the read bytes to the relevant variables.

The process_x33, process_x34 and process_72 functions are called by ParseData to populate the data structures of Packet_33, Packet_34 and Packet_72 respectively. The process_x34 function also stores the pseudoranges in a global array PRraw indexed by the channel number. The structure definitions of the three packets are defined in the header file corresponding to the source file containing these functions.

#### 7.1.3 Getting Ephemeris

Due to the complexity of the process, several functions are required to get the ephemeris data from the GPS receiver. The GetEphemeris function is the main function that calls the relevant functions to achieve the transfer of ephemeris data. This function takes the port handle of the port that has the GPS receiver used to get the ephemeris data and also takes in a file pointer that points to a file for logging data. Just like the PacketRead and ParseData functions, the GetEphemeris function populates the Packet_35 data structure that holds the ephemeris data for up to 12 satellites.

#### 7.1.4 Calculating Satellite Position

The GetECEFSatPosition function calculates the position of the satellites from the ephemeris data stored in Packet_0x35. The resultant satellite positions calculated for the 12 channels are stored in three global arrays – one for each coordinate X, Y and Z indexed by the channel number which is also the order of appearance of data transmitted by the GPS receiver.

### 7.1.5 Correcting Pseudorange

The raw pseudoranges measured from the 12 channels of the receiver are stored in a global array PRraw by the process_x34 function described above. The GetCorrectedPr function takes the approximate position of the user in ECEF coordinates as inputs. The function also takes in an integer value that specifies the unit (base or roving). The function uses the satellite position array along with the approximate user position to get a rough pseudorange measurement. The pseudoranges measured by the receiver drift rapidly with time and the code that obtains the pseudorange data may have to be revisited. The function then calls other functions to calculate the clock, Ionospheric and Tropospheric corrections. The resultant errors calculated by these functions are then added or subtracted as necessary from the array of pseudoranges.

### 7.1.6 Calculating standalone position

The GetStandAlonePos function takes the approximate user position in ECEF coordinates as the input and then uses the satellite position and the corrected pseudoranges to calculate the single receiver, non-DGPS position of the roving receiver. The results yielded by this function are erroneous and the algorithm to calculate the single position needs to be revisited.

## 7.2 Future work

The calculation of the standalone position has not yielded the desired result, hence it will be futile to proceed implementing the DGPS algorithm without revisiting the algorithm to calculate the standalone position. Other methods such as the sum of least squares or the Kalman filter technique need to be studied for feasibility and the correct algorithm needs to be implemented.

## 7.3 Conclusion

Once the correct standalone position is calculated, the pseudoranges can be corrected by measuring double difference DGPS corrections and the same method used for calculating the standalone position can be re-implemented with the corrected pseudoranges. The standalone position calculated after DGPS corrections will be accurate within 10cm with a 90% CEP. The only error that the double difference DGPS technique cannot mitigate is the cycle-slip in the ambiguity calculation.

# BIBLIOGRAPHY

1. Professor P. Daly, December 1993. NAVSTAR GPS and GLONASS: global satellite navigation systems. Electronics and Communication Engineering Journal.

2. Professor Per Enge, August 2003. GPS Modernization: Capabilities of the New Civil Signals. Australian International Aerospace Congress.

3. TeachEngineering, September 2008. State Your Position. http://www.teachengineering.org

4. Major Jeffrey K. McGee, United States Army, November 1999. Global Positioning System Selective Availability: Legal, Economic, and Moral Considerations. School of Advanced Military Studies, United States Army Command and General Stall College, Fort Leavenworth, Kansas.

5. ARINC Research Corporation, March 1991. GPS NAVSTAR Global Positioning System, User's Overview.

6. R. M. Alkan, H. Karaman & M. Sahin, 2005. GPS, GALILEO and GLONASS Satellite Navigation Systems & GPS Modernization. IEEE Recent Advances in Space Technologies (RAST) international conference.

7. Los Angeles Air Force Base, April 10th, 2009. Air Force Successfully Transmits an L5 Signal From GPS IIR-20(M) Satellite. http://www.losangeles.af.mil/news/story.asp?id=123144001

8. Department of Physics, Syracuse University, 1998. PHY 312 Global Positioning System.

9. United States Cost Guard Navigation Center. GPS General Information. http://www.navcen.uscg.org

10. The NAVSTAR Joint Program Office, 2006. GPS Overview.

11. United States Cost Guard and Navigation Center, September 9th 2005. General GPS News.
http://www.navcen.org/gps/gpd_news_090905.htm

12. James M. Conrad and Santosh Kolenchery, July 2000. Wireless Phones, GPS and Data Applications. The 6th International Conference On Information Systems Analysis And Synthesis (ISAS 2000), Orlando, FL.

13. S. S. Wang, Marilynn Green and Maged Malkawai, Nokia Research Center, 2002. Mobile Positioning Technologies and Location Services. IEEE Radio and Wireless Conference 2002.

14. Carl R. Nave, 2006. Global Positioning System. Department of Physics and Astronomy, Georgia State University, Atlanta, GA.
http://hyperphysics.phy-astr.gsu.edu/hbase/gps.html

15. Manuel Hernandez Pajares, 1998. Global Positioning System (GPS). Research Group of Astronomy and Space Geodesy, Universitat Politècnica de Catalunya, Barcelona, Spain.
http://maite152.upc.es/~manuel/tdgps/node10.html

16. GPS Joint Program Office, December 7th 2004. NAVSTAR Global Positioning System - Interface Specification.

17. Wikimedia Commons, April 2009. The GPS navigation message.
http://commons.wikimedia.org/wiki/File:Navigation_message_GPS.png

18. The Los Angeles Air Force Base, April 2009. Air Force Successfully Transmits an L5 Signal From GPS IIR-20(M) Satellite.
http://www.losangeles.af.mil/news/story.asp?storyID=123144001

19. Guochang Xu, 2007. GPS Theory, Algorithms and Applications, Second Edition. Springer-Veriag Berlin Heidelberg.

20. Nao Akiyama, Toshiyuki Tanaka, 2007. Improvement of GPS positioning accuracy by ionosphere delay correction. Masashi Yonekawa School of Science and Technology, Keio University, Japan - SICE Annual Conference 2007

21. Federal Aviation Administration (FAA) October 9th 2007. The GNSS Library.

22. Federal Aviation Administration (FAA), September 2008. Navigation services. http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/laas/

23. Chris Rizos, 2005. How can my position on the paddock help my future direction? Geospatial Information and Agriculture Conference, Denver, Colorado. www.regional.org.au/au/gia/17/557rizos.htm

24. Chris Alber, Randolph Ware, Christian Rocken, John Braun, 2000. Obtaining single path phase delays from GPS double differences. Geophysics Research Letters.

25. Ahmed El-Rabbany, 2006. An autonomous GPS carrier-phased-based system for precision navigation. IEEE Intelligent Transportation Systems Conference, Toronto, Canada.

26. Jonathan Michael Frentzel, 2003. Relative Accuracy and Precision of Differentially Corrected GPS on a Moving Vehicle. Virginia Polytechnic Institute and State University.