3-2000

Development of a Real Time Guidance System for a Kinematic Test Vehicle

Orhan Cekic

Mar 2000

DEVELOPMENT OF A REAL TIME GUIDANCE SYSTEM FOR
A KINEMATIC TEST VEHICLE

THESIS

Orhan Çekiç
First Lieutenant, Turkish Air Force

AFIT/GE/ENG/00M-21

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

20000821 112

AFIT/GE/ENG/00M-21

DEVELOPMENT OF A REAL TIME GUIDANCE SYSTEM FOR
A KINEMATIC TEST VEHICLE

THESIS

Presented to the Faculty

Department of Electrical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Orhan Çekiç, B.S.C.E.

Lieutenant, Turkish Air Force

March 2000

Preface

Working on this thesis has been the highlight of my academic career. There are many people who have helped me along the way.

I would like to express my sincere appreciation to all AFIT faculty members, especially to my thesis advisor *Captain John Raquet*. I would like to thank him for his enthusiasm and eagerness to teach and for his insightful and experienced help. I would like thank to my readers, *LtCol Mikel Miller* and *LtCol Timothy Jacobs* for their guidance and gracious support throughout the course of this thesis. I would also like to thank *Mr. Don Smith* for the enormous level of support that he provided in the lab.

Most importantly, I would like to thank my wife *Mutlu*. She has been most supportive wife through the many long nights and weekends that it took to complete this research. She is also a wonderful mother to my precious daughter *Ecenaz*. I would also like to express my gratitude to my mother Murvet CEKIC.

Finally, I would like to dedicate to this thesis to my father *Tahir CEKIC* (in remembrance) from whom I got my inspiration and my encouragement, and to my wife *Mutlu CEKIC* with whom I love to share my life.

Thanks to the Turkish Air Force and the US Air Force for providing such a great experience.

DEVELOPMENT OF A REAL TIME GUIDANCE SYTEM FOR A KINEMATIC TEST VEHICLE

Orhan Çekiç, B.S.E.E
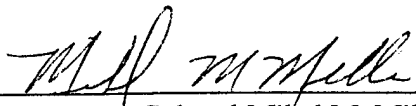Lieutenant, Turkish Air Force
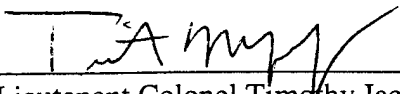
Approved:

_____

Captain John Raquet

Assistant Professor, Thesis Advisor

16 FEB 2000

Date

_____

Lieutenant Colonel Mikel M. Miller

Assistant Professor, Thesis Reader

16 Feb 2000

Date

_____

Lieutenant Colonel Timothy Jacobs

Assistant Professor, Thesis Reader

16 FEB 2000

Date

Table of Contents

List of Figures

List of Tables

Abstract

Differential carrier phase GPS measurements are capable of giving centimeter-level accuracies. These accuracies have many potential applications for safety and control of various types of vehicles.

For this research, a real-time guidance system is developed. The real-time guidance system can be divided into two components: hardware and software. The hardware component consists of two GPS receivers (one base, one mobile), two wireless 115 Kbaud transceivers, and two laptop computers. One computer is for the reference station and other is for the mobile receiver interface and graphical display. The guided vehicle is a golf car called the Remote Sensing Autonomous Vehicle for EN (RAVEN).

The research concentrated on developing real-time data processing algorithms and using these algorithms to show guidance information to the user via a graphical interface. The developed software reads the real-time GPS data using an RS-232 interface and converts it to a usable form for the data processing algorithms. The data processing algorithm compares the real-time data with the desired track and outputs the guidance information to the guidance display. Based on the information from the guidance display, the user is able to drive on the desired track.

Four tests were performed to evaluate the guidance system performance and human factors under different circumstances. These tests include update rate tests, a single instrument test, varying parameter tests, and an under hood test. Tests results and user feedback show that the system performs well under most conditions.

DEVELOPMENT OF REAL TIME GUIDANCE DISPLAY FOR A KINEMATIC TEST VEHICLE

1. Introduction

1.1 Background

The Global Positioning System (GPS) is a great technological success developed by the Department of Defense (DoD) to provide precise estimates of position, velocity, and time. Civil use was a secondary objective. On the basis of national security considerations, the civil users of GPS have been limited to a purposefully degraded signal. Nevertheless, the civil applications of GPS have grown at an astonishing rate. GPS has found applications in land transportation, civil aviation, maritime commerce, surveying and mapping, construction, mining, agriculture, earth sciences, electric power systems, telecommunications, and outdoor recreational activities. The system is being used to provide accuracy levels which would have been unthinkable 20 years ago. The commerce in GPS-related products and services has grown rapidly in the 1990's. GPS is on its way to becoming a part of our daily lives as an essential element of the commercial and public infrastructure. [1]

1.2 Problem Definition

Differential carrier phase GPS measurements are capable of providing centimeter-level accuracies. This level of accuracy is required for applications involving safety and the control of vehicles.

One of the intents of this research was to show the utility of GPS in a real-time guidance system. To be able to reach this goal, software was to be developed which incorporate real-time GPS data. This software provides real-time guidance information with respect to a *"desired path."*

The second intent of this research was to develop a guidance algorithm that could be applied to many different vehicles. Note that the desired travel path could be extended anywhere, and with minor software modifications, the test vehicle could be exchanged with other types of vehicles (including aircraft).

The third intent of this research was to investigate the performance of the guidance interface with different people to determine guidance system performance and their human factors that affect it could be observed.

1.3 Assumptions

The following assumptions were used in this thesis:

a) The true coordinates for all reference receivers are known.

b) All data is processed in real-time.

c) There is no GPS jamming that would adversely affect the base and mobile receiver while using the system.

d) All receivers are working to factory specifications.

e) The test area is within the data transceiver limit from the base receiver.

f) All calculations performed use the Local and Earth Centered Earth Fixed (ECEF) coordinate systems.

1.4 Thesis Overview

Five chapters and three appendices are included within this thesis. Chapter 1 provides high level background and research description information. Chapter 2 provides the necessary background to understand the terminology, basics of GPS, and hardware and software components that were used during the thesis research. Chapter 3 combines the hardware and software components that were described in Chapter 2, and the design and implementation of software that is written for this research is described with sufficient details to understand the system. Chapter 4 presents the result of different tests that were done with different users. The different tests were update rate test, one instrument test, different parameter set tests and the under hood test.

1.5 Summary

This chapter has provided the introduction for this thesis. The background, problem statement, assumptions, thesis overview has been discussed. In the next chapter, the detailed background and the overview of system instruments will be presented to provide the reader necessary insight to understand this research.

2.Background

2.1 Brief GPS Description

The Global Positioning System (GPS) is a worldwide, all-weather radio-navigation system consisting of satellites and their ground stations. GPS was developed and has been controlled by US Department of Defense (DoD).

There are two kinds of service --Standard Positioning Service (SPS), which is unrestricted for civil users, and Precision Positioning Service (PPS) which is available for the DoD authorized user only. The PPS service is specified to provide 0.1 m/s velocity accuracy and 100 nanoseconds time accuracy. The SPS has a position error of 100 meters (2 DRMS). The difference between PPS and SPS specifications is mostly due to Selective Availability (SA). Selective availability introduces an intentional error into satellite clock and ephemeris parameters thereby degrading user performance.

2.1.1 GPS Architecture

GPS consists of three segments, which are the space segment, the control segment and the user segment. The space segment consists of the satellites in orbit that provide the navigation signal and the data message to the user. Typically, there are 24 satellites in six orbital planes, which are at a 55 degrees inclination. The satellites have an orbital radius of 26,600 km and an orbital period of approximately 12 hours. The satellites are tracked from five monitoring sites spread around the globe (Ascension Island, Diego Garcia, Kwajalein, Hawaii, and Colorado Springs) for orbital prediction and health indicators. Three of these monitoring stations, Ascension Island, Diego Garcia, and Kwajalein also have the communication capability to upload (via S-band radio links) data to be broadcast

by the satellites. The Master Control Station located at Schriever (formerly named Falcon) Air Force Base manages the operations. Finally, the user segment consist of receivers located worldwide on the ground, in the air, and on the sea.

2.1.2 How GPS Works

GPS satellites continuously transmit their signal to the ground. Any GPS receiver that has line of sight visibility to the satellite passively receives the satellite signal. The basic GPS concept is based on very accurate time information, because the position is calculated as a matter of time and speed of light. The satellite transmits the signal, which travels at the speed of light and arrives at the GPS receiver. The receiver then uses a correlation process to determine the signal travel time, which can be converted to a distance by multiplying the speed of light. GPS receivers typically have five to twelve channels that can each track different satellite.

2.1.3 GPS Signals and Measurements

Each satellite transmits continuously at two frequencies in L band: 1575.42 Mhz (L1) and 1227.6 Mhz (L2). The portion of signal intended for unrestricted use is broadcast by each satellite at L1, and it is modulated by a pseudorandom noise (PRN) code called Course Acquisition (C/A) code. The C/A code has a chipping rate of 1.023 Mhz. Each satellite also broadcasts a second signal on L1 and L2 in phase quadrature with the C/A code signal. Access to these signals is controlled by an encryption code. These signals are called P code, or with encryption, P (Y) code.

There are three measurements: pseudorange, Doppler and carrier-phase. The receiver attempts to acquire the known C/A-code with an initial time uncertainty of 1023 code chips and frequency uncertainty of up to 5 KHz due to Doppler shift. Acquisition of P(Y) code, if the user is authorized, is based on the coherence of the C/A and P(Y) codes. The receiver first acquires the C/A code and then acquires the P(Y) code with the aid of the timing information in the data message. Direct acquisition of a P(Y) code is difficult by design due to the length of the code.

The PRN code transmitted by each satellite is known to the receiver, which generates a replica of it. The delay between this code replica and the signal received from the satellite is the apparent transit time of the signal. Basically, the receiver slides the code replica in time until it matches the code received from the satellite. This process of correlating the received signal with the receiver-generated replica gives the apparent transit time of the signal modulo 1 ms. Multiplying the apparent transit time by the speed of light gives pseudorange.

Doppler shift and carrier phase measurements are formed in the carrier tracking loop. The Doppler shift, caused by the relative motion of a satellite and the user, is the projection of the relative velocity on the line of sight, and it can be converted into pseudorange rate. The carrier phase measurements are formed by integrating the Doppler measurements.

2.2 Differential GPS Concept (DGPS)

The DoD purposely limits the accuracy of GPS position accuracies through the use of SA, but this is not the only thing that can degrade GPS accuracy. The receiver clock

6

error, ionospheric error, trophosheric error, and multipath also contribute. However, SA is the dominant error for civil users. DGPS can reduce many of these errors through the use of a reference receiver at a known position.

The reference receiver obtains code-based GPS pseudorange measurements, just as any standard GPS receiver, but because the monitoring station knows its precise position it can determine the "biases" in the measurement. For real-time applications, the reference station transmits these biases, which are called *"differential corrections,"* to all users in the coverage area. By using DGPS, many GPS errors (such as ionospheric and trophosheric delays, satellite ephemeris errors, and clock errors) can be significantly reduced or eliminated.

DGPS can provide meter-level and even centimeter-level position estimates depending upon the closeness of the user to a reference station, the latency of the corrections transmitted over the radio links, and the DGPS technique that is used.

2.2.1 Centimeter-Level DGPS (Ambiguity Resolution)

Code-based DGPS can attain meter level accuracy. The DGPS carrier phase concept is to use the phase measurement in addition to code measurement. The phase information is extremely precise position information, but this information is ambiguous, because the particular cycle that is being measured is not known. (The receiver precisely knows where it is in the cycle, but it doesn't know exactly which cycle it is in). By using advanced processing techniques, the integer portion of cycle can be found. This results in position accuracies at the cm level. This concept is called carrier phase ambiguity resolution [2].

2.3 Description of DGPS System that is Used for this Research

2.3.1 GPS Receiver (Z-Surveyer)

The Z-surveyor process signals from the GPS satellite constellation. The receiver provides real-time position, velocity, and time measurements using twelve dedicated separate and parallel channels for coarse acquisition (C/A) code-phase and carrier-phase measurements on both the L1 and L2 bands. The receiver obtains the satellite signal via an L-band antenna and Low Noise Amplifier (LNA). The Z-surveyor receiver is shown in Figure1.

The Z-surveyor provides centimeter level DGPS accuracy in real-time. Its dual frequency capability alleviates the effect of ionospheric refraction so that baseline measurements are consistently accurate. The Z-surveyor offers improved satellite tracking under adverse visibility conditions such as under a tree canopy, in an "urban canyon", or between buildings.

Using 7.5 watts of power, it operates up to 4.5 hours on a single internal battery. The Z-surveyor can be connected directly to a computer using one of the RS-232 serial ports. This is the port used by the software that written for this research [2].

Figure 1. Z-Surveyor Receiver

The performance that is reached with real-time Z-surveyor kinematic position is as follows: while moving --horizontal 3 cm, vertical 5 cm; while static --horizontal 1 cm, vertical 1.7 centimeter. The initialization time after acquisition of 8 or more satellites is 30 seconds with a reliability of 99.9% [2].

The receiver calculates the precise position, the position information is transferred to the computer through the RS-232 port, and the real-time position information is used in the guidance algorithms that will be described later.

2.3.2 Data Transceiver (DGR-115)

The FreeWave DGR-115 transceiver can operate in point to point or point to multi-point modes with data rates up to 115.2 Kbaud over a distance of 20 miles. The transceiver/receiver is used for sending data from base station, and the rover (moving vehicle) receives the data. It is shown in Figure 2 [3].
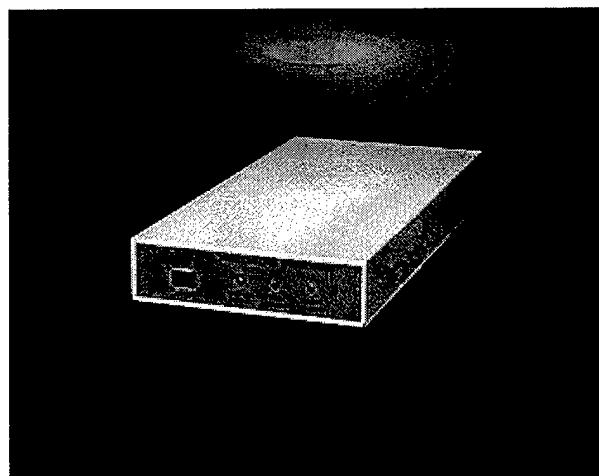
Figure 2. DGR-115 Data Transceiver

2.4 Video Graphics Library (OpenGL)

OpenGL is a software interface for graphics programmers to produce high-quality color images. OpenGL was chosen for this research because it is stable, reliable, flexible and portable. Portability is very important for this research. Any graphical application that is chosen must work in a laptop environment. The laptop environment is important because the software to be created must be portable and should be compatible to that environment. Flexibility is another key point, because, based on input from the users and the development of research, the visual interface needs to be adaptable. Most commercial software is designed for a specific type of application. For this research, it was important to have the flexibility to design the display exactly as desired, which is not possible with most commercial software.

2.5 The Vehicle (RAVEN)

The vehicle that is used for this research is typical golf car called Carryall II, built by Club Car Incorparation. At AFIT it is referred to as RAVEN (Remote Sensing Autonomous Vehicle for EN). The RAVEN has 14.6 cubic feet of pickup-bed capacity. In this research, the RAVEN electrical power supply is used to power the transceiver and laptop [4].

2.6 Summary

To be able to understand the guidance display and the system, the reader needs to know the basics of GPS and the DGPS concept, as well as understand the components of the guidance system. This chapter has presented an overview of these essential areas.

3. System Description

The system can be described in terms of two areas: hardware and software. The hardware components used for this research consist of two cm-level accuracy GPS receivers (one base, one mobile), two wireless 115 Kbaud transceivers, and two laptop computers. One computer is for the reference station and other is for mobile receiver interface and graphical display. The vehicle to be guided is a golf car called the RAVEN.

The software obtains the Z-surveyor-generated GPS position, compares it with the desired track, and generates a display that shows the user what action should be taken. Both the hardware and the software algorithms will be explained in detail in the section that follows.

3.1 Hardware Description

The receiver operates as either a base (reference) station or a remote (rover) station, providing real-time centimeter-level differential GPS accuracy using the carrier measurements. At the base station, corrections are calculated for each measurement. The correction is transmitted by a transceiver to the golf car that has another transceiver to get the correction. Figure 3 shows the general data flow of the real-time guidance system.
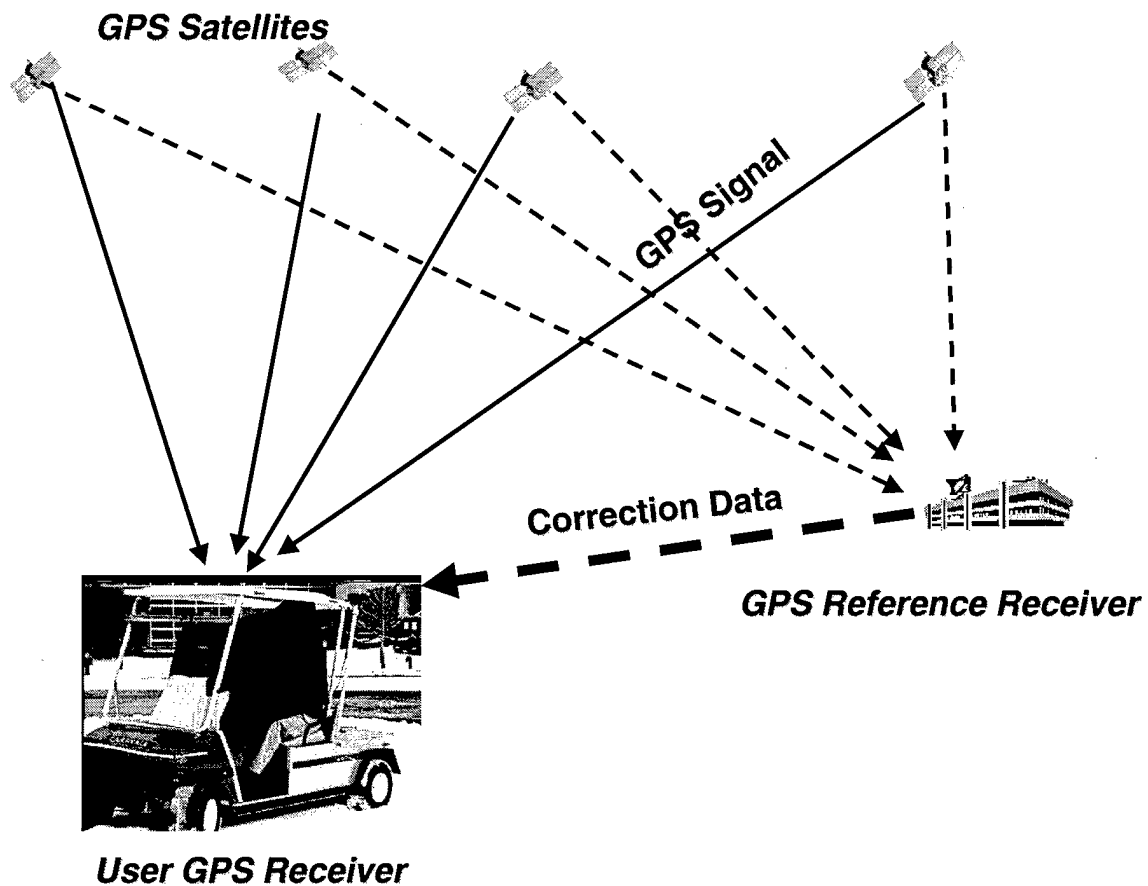
Figure 3. General Data Flow of Real Time Guidance System

The base station is set up on the roof of AFIT's building 640. The stationary antenna

is positioned at a known location. The reference receiver antenna is connected to the GPS

receiver using an antenna cable. The receiver output port is connected to the DGR-115

transceiver so that the differential corrections can be transmitted to the RAVEN. The

configuration of base station can be seen in Figure 4. The GPS antenna is connected to

the GPS receiver, and the wireless transceiver is placed on the roof of AFIT. The receiver

configuration is established using the Commander software will be described later.

12

GPS Antenna

Correction Data

Wireless Antenna

Correction Data

power

RS-232 two way communication
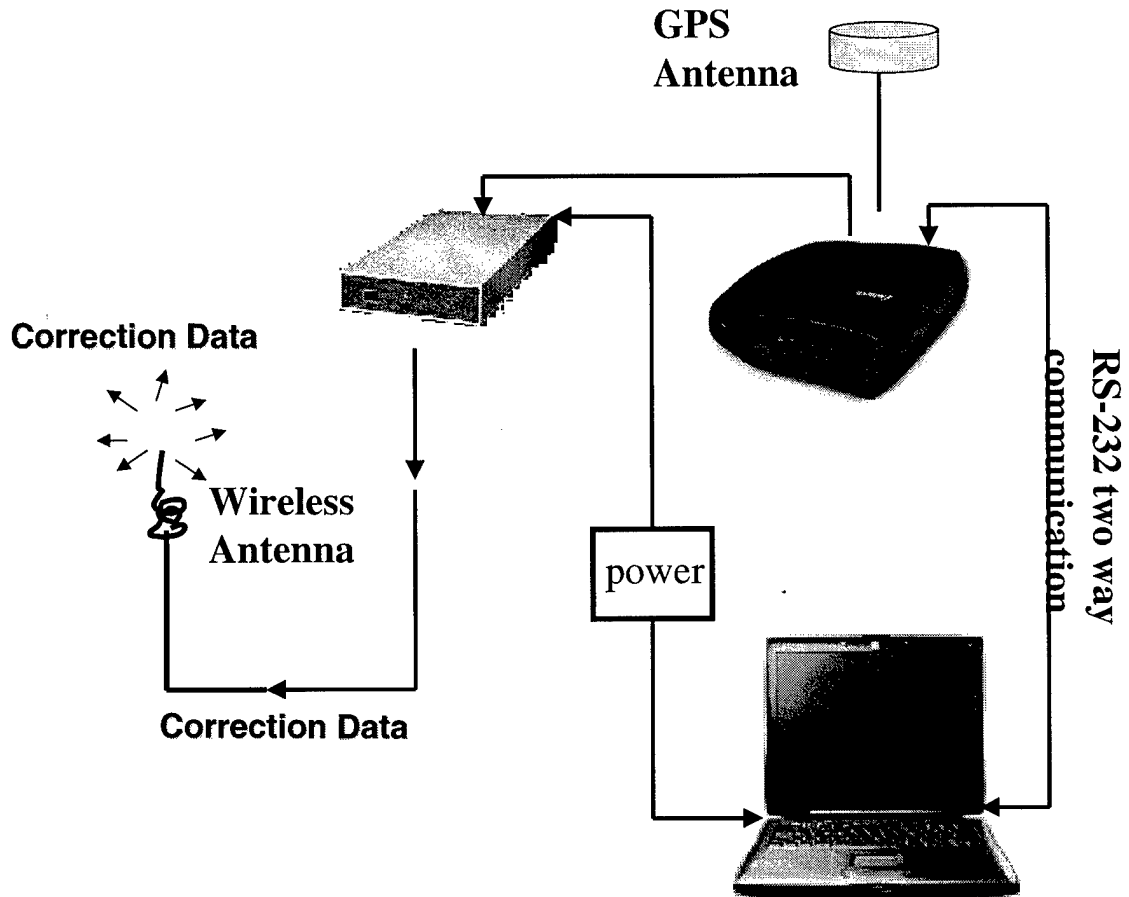
Figure 4. Differential GPS Base Station

The configuration of the RAVEN is shown in Figure 5. The GPS antenna is placed on the center of the roof of the vehicle. The wireless transceiver antenna is placed on the left top of vehicle for receiving the differential correction. The transceiver is connected to port A of the Z-surveyor. The PC and Z-surveyor is connected by a two-way RS-232 connection.
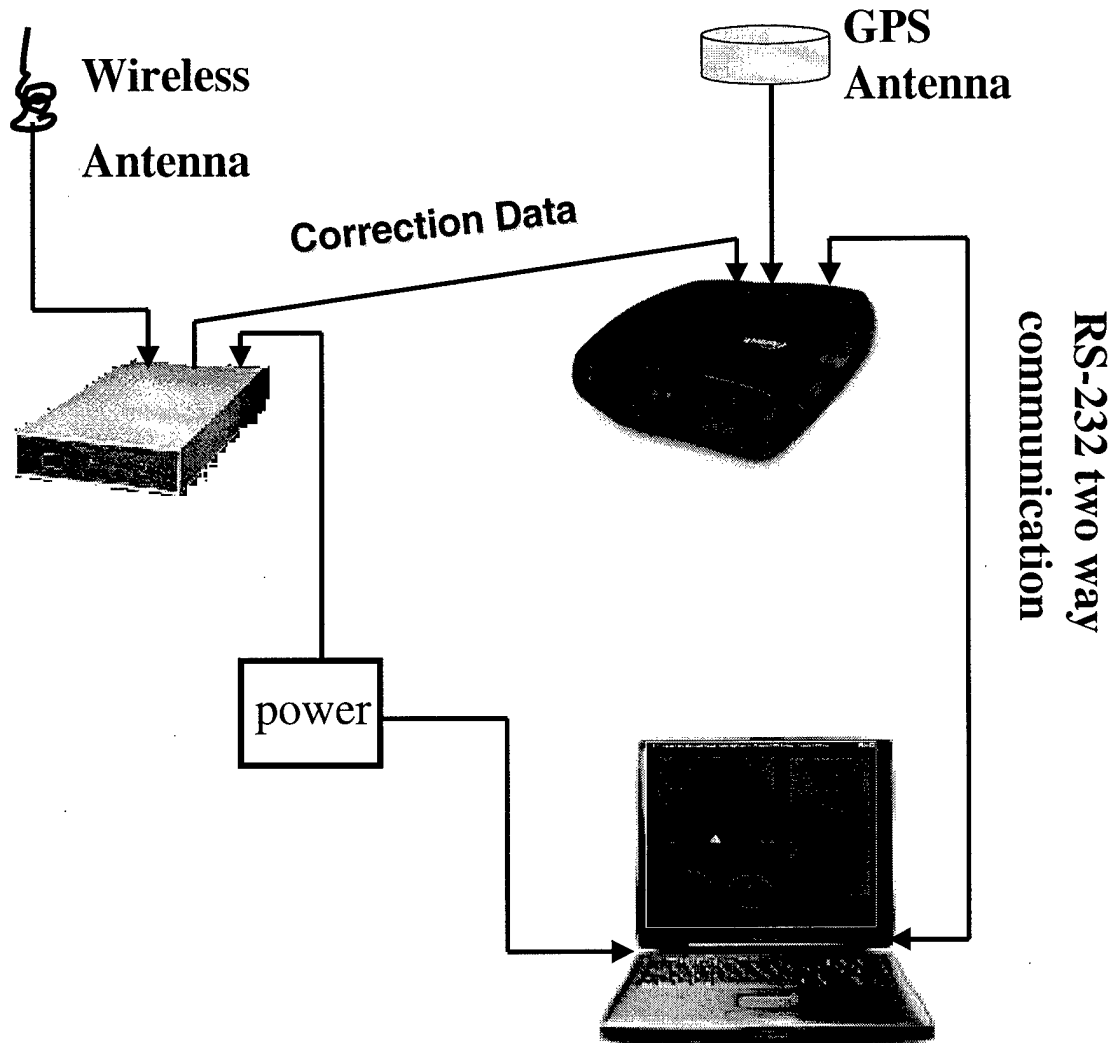
Figure 5. Hardware Configuration on the Mobile Vehicle (RAVEN)

It was a challenge to find an ergonomically sound location on the vehicle. Since the REVAN is a typical golf car, there was no convenient mount for a laptop computer. One of the considerations was that the laptop had to be high enough for easy user interaction. It was desirable to have the top of the monitor at the user's eye level. The other consideration was that the user must be at a comfortable distance from computer screen. Note that there were some limits on what could be done. The RAVEN's chair is

not adjustable (up/down, forward/backward), so the mount was designed for an average person (approximately 5 feet 10 inch).

An initial laptop mount was designed and implemented, but it was hard to anticipate problems prior to testing in the field. After a test it was evident that the screen was too far away, making human/computer interaction difficult. The other consideration was sunlight. When the sunlight came from the front, the user could easily see the display. When the sunlight came from the back, however, it was very difficult to see the display. For these reasons the mount was redesigned to move the screen closer to the user to solve the distance problem. Also, a shade was designed around the laptop computer to solve the sunlight problem. The final design is shown is Figure 6.

Figure 6. A Volunteer Driving the RAVEN Based on Guidance Display

3.1.1 System Operation Procedure

The receiver is activated when power is applied (using the RAVEN's battery). After self-test, the receiver initializes its 12 channels and begin searching for all space

vehicles (SV) within the field of view of antenna. The receiver can track all visible GPS satellites. As the receiver acquires (locks onto) each SV, it starts tracking the time and then collects the ephemeris data.

There are two ways to change the run-time parameters of the Z-surveyor system. The first is to use the Ashtech "Commander" software that allows the user to set up any of the receiver parameters. The second is through the front control panel of the receiver itself. The front control panel has a menu that the user can use to change parameters settings. Using the Commander software, once the parameters are set appropriately, they can be saved so that they remain unchanged next time the receiver is powered up.

Using the Commander software, the ambiguity fixing parameter can be set to a confidence level between 90.0 and 99.9. Higher confidence levels results in longer search times, but increase the reliability of ambiguity solution. For example, changing the confidence level from 99 to 95 decreases the time to solve the ambiguity and give the fixed solution, but also increases the chance that the ambiguity is fixed incorrectly. Thus, the default value is applied (99). When selecting the dynamics, the "automobile" mode was selected as being the most representative of the expected vehicle dynamics. The rule of thumb for selecting the dynamics is to select the dynamics for the fastest acceleration that is expected. If the dynamics are not set properly, the carrier phase differential solution will be less accurate. (Other choices are static, walking, ship and aircraft). After setting the parameter, the operator clicks on the *SEND* command, which sends the needed parameter to the receiver. Figure 7 shows RAVEN set up using the Commander software.

Figure 8 shows the configuration for base receiver. The *$PASHS CPD PRT* sets the

port output to the desired port. In this case, it is port that is connected to the transceiver.

In addition, the *$PASH RCI* command can be used to set the interval for data recording

and raw data output.
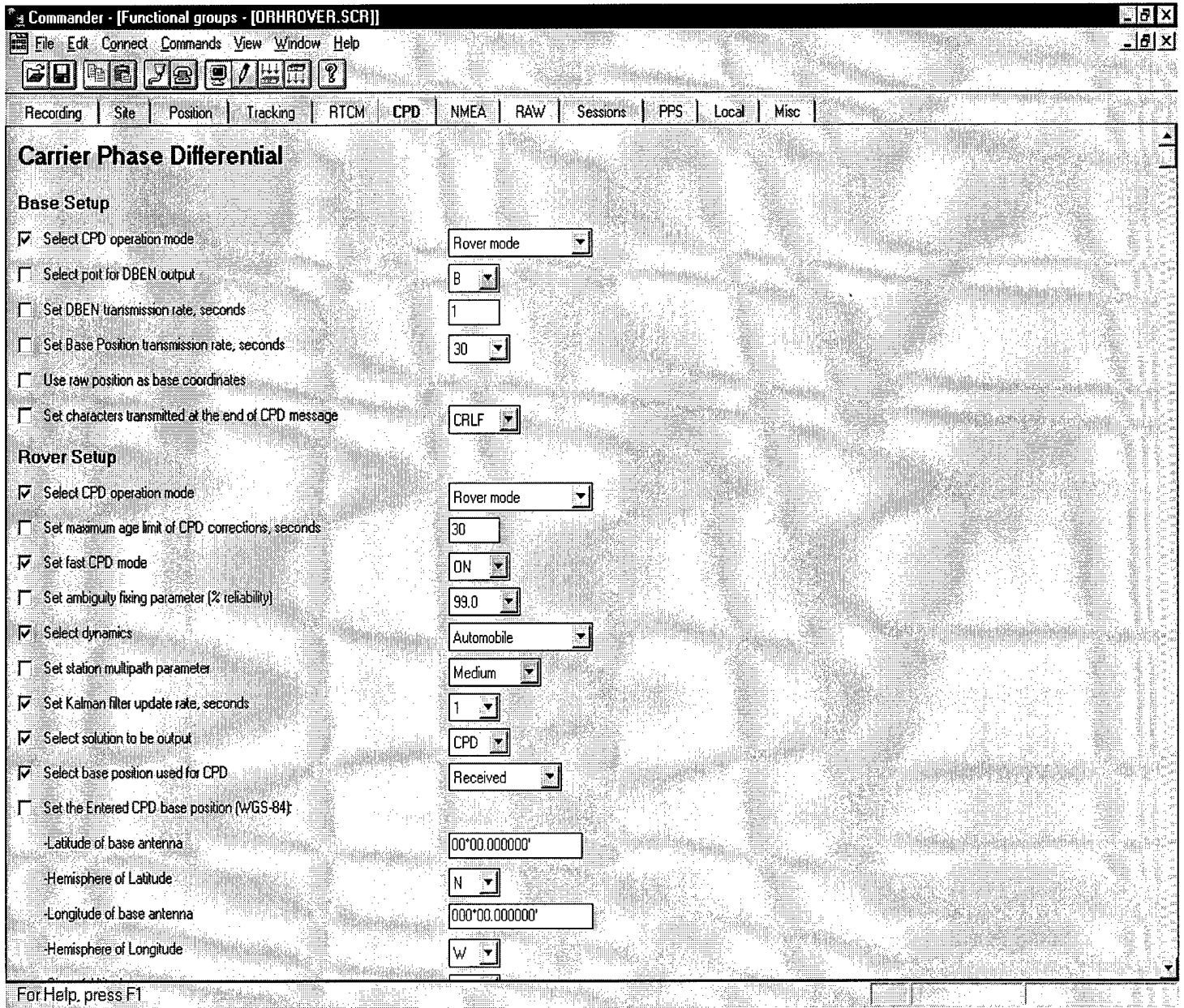
Figure 7. Mobile (Rover) Set up Using the Commander Software
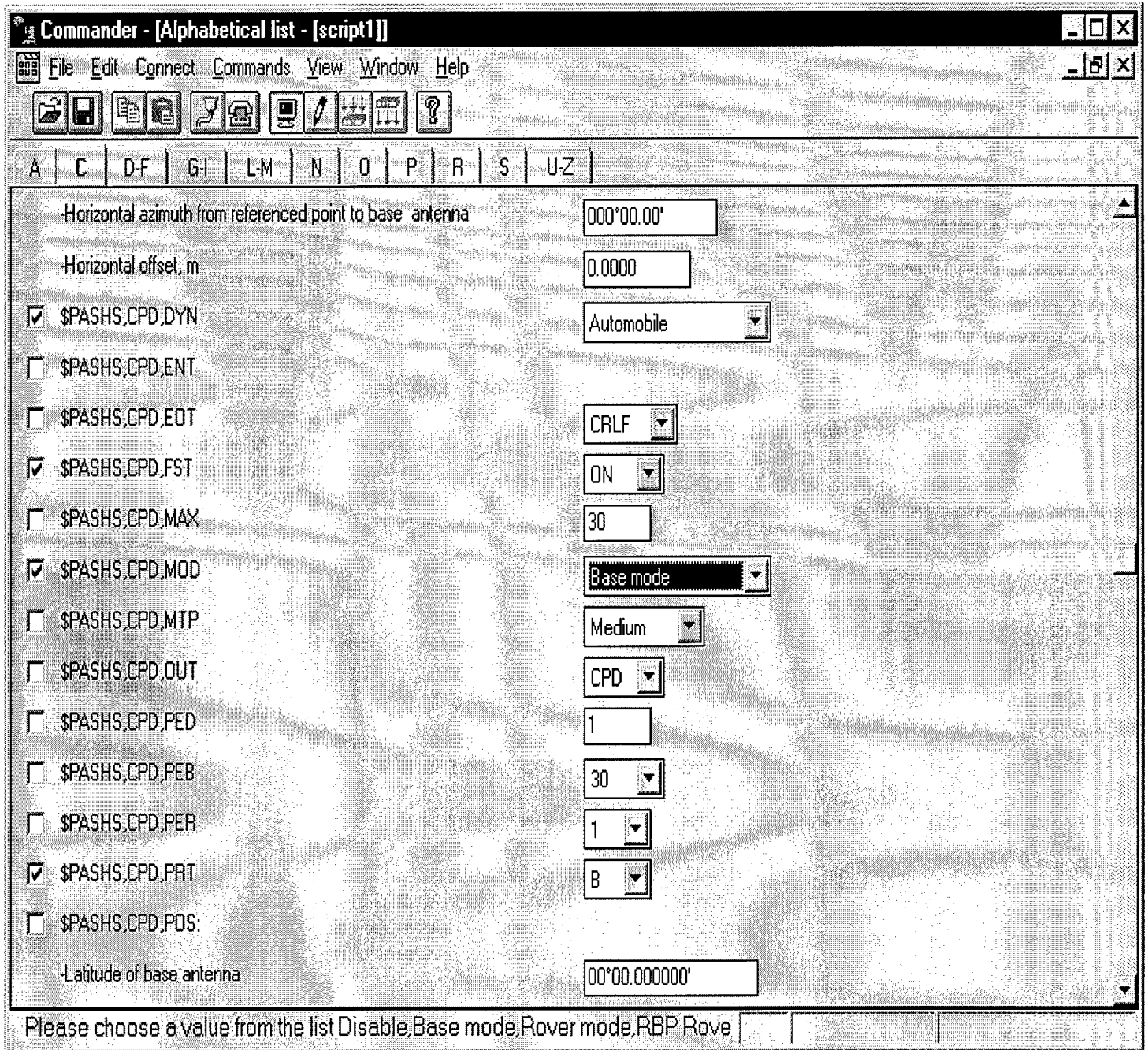
Figure 8. Reference Station (Base) Set up of the Commander Software

3.2 Software Overview

The guidance software can be divided into three main parts. These are the graphical interface, the data processing, and the port reading. Figure 9 shows the top level guidance software description.
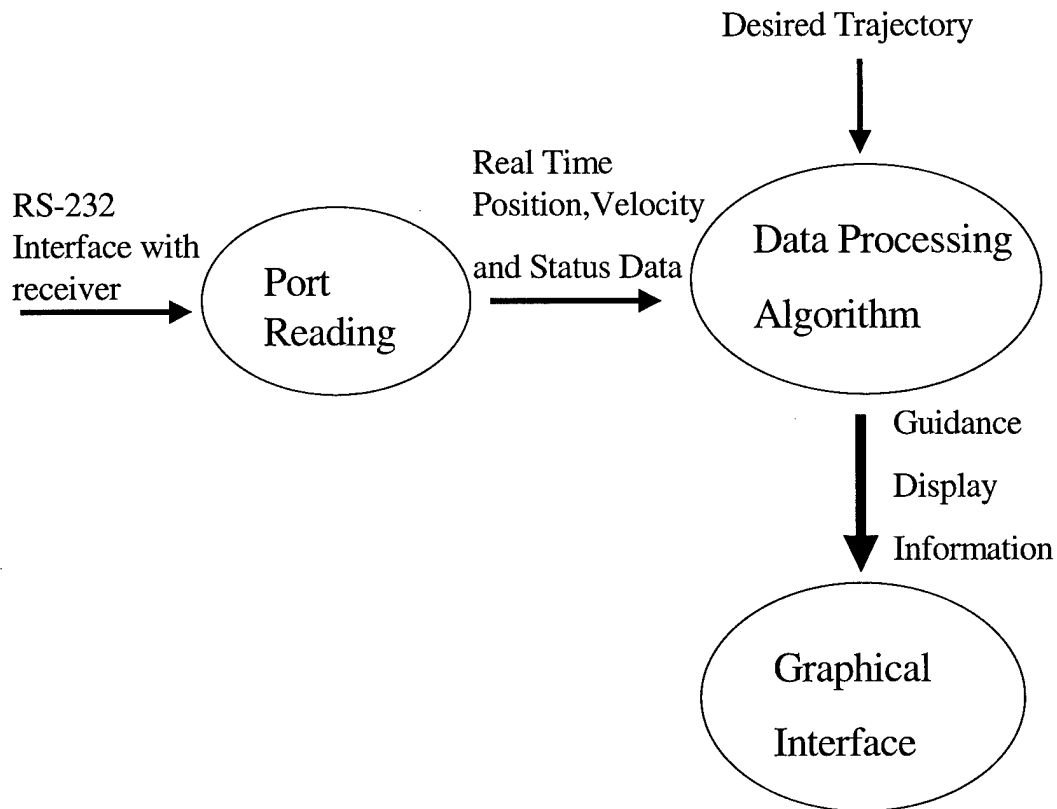
Figure 9. High Level Software Description

The desired trajectory is stored as three data files obtained from inside of, in the center of, and outside of the desired track. The stored data files (inside, center and outside) are processed upon algorithm startup. The required information is created (such as positions and headings) for later use by the data processing function. The port reading

function reads real-time data coming from the port and converts it to usable form for the data processing function. The outputs from the port reading function are position, velocity, heading, and a solution flag (showing the current system reliability information).

The information that is needed by the data processing function to generate the steering command is as follows:

- *Port pointer*: The port is opened at software start up, and the port information is tracked by that pointer.

- *Heading*: The heading information includes real-time heading and stored data points' headings (inside, center and outside data points headings).

- *Position:* The position information includes real-time position and stored data points' positions

The information that is output from data processing function to the graphical display function to display the steering command to the user is as follows:

- *The off center distance*: The distance from current user position to the desired track.

- *The correction angle:* This is the calculated heading that user need to follow.

- *Velocity:* The current velocity of user vehicle

- *Solution flag:* Shows the system reliability condition, and whether or not the output is a fixed integer 1 cm-level (accuracy) solution.

3.3 Port Reading

Initially, Microsoft port reading software was evaluated for its suitability for this project. Even though software existed to read data from the RS-232 port and write it to the disk, it proved to be too complex to be modified for the current project.

Finally Green Leaf Software called CommX 1.1 was chosen as a good solution for this research. The software stated that it did not support Win32 Applications in Visual C++ (which was how the prototypes were designed). However, the company provided sample code that showed how to use the port libraries with a Win32 application. The Green Leaf CommX software is a library that can be imported to a project. In the final configuration, the library is imported with an *"#import"* command, and then the rest of the port I/O commands can be used by the software.

The software reads the data in ASCII format on a character by character basis. The format of the "CBN" data message from the mobile receiver is consistent from line to line, with each individual parameter separated by commas. The CBN message provides complete information on the position, velocity, solution, and number of satellites. The exact CBN format is as follows:

PASHR, CBN, M1, S2, D3, F4, M5, C6, M7, C8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F9, F20, F21, F22 *c

The interpretation of each of the parameter is given in Table 1 [2].

Table 1 "CBN" Data Interpretation

Parameter	Description	Range
M1	Receiver time UTC (hhmmss.ss)	0-235959.99
S2	Four character site identification	
D3	Number of satellites used in position computation	0-12
F4	PDOP	0-999.9
M5	Latitude in degrees and decimal minutes Ddmm.mmmmmmmm	0-90
C6	Latitude direction	'N/S'
M7	Longitude in degrees and decimal minutes Ddmm.mmmmmm	0-180
C8	Longitude direction	'E/W'
F9	Ellipsoid Height (meters)	-1000 TO 180000
F10	Standard Deviation of latitude component (meters)	0-99.99 m
F11	Standard Deviation of longitude component (meters)	0-99.99m
F12	Standard Deviation of ellipsoid height (meters)	0-99.99m
F13	Cross correlation of XY	+/- 30.00m
F14	Cross correlation of XZ	+/- 30.00m
F15	Cross correlation of YZ	+/- 30.00m
S16	Solution type flag containing 6 Parameters (Indicates validity of the solution)	
F17	Velocity in East Direction	+/- 500.00m/s
F18	Velocity in North Direction	+/- 500.00m/s
F19	Velocity in Upper Direction	+/- 500.00m/s
F20	Standard Deviation of East Velocity	0-99.99m/s
F21	Standard Deviation of North Velocity	0-99.99m/s
F22	Standard Deviation of Upper Velocity	0-99.99m/s
*c	Checksum	

The software reads in an entire line (as indicated by the "end-of-line" character) before it processes that line. For initial software development, a file which had exactly the same GPS data as would be expected from the port was used to be sure that the concept was correct. This method was convenient, easy to work with, and easy to debug compared to real time work. Then the same logic was applied to the real time port reading. There were some differences working with the real-time port compared to the file. The most important one is that when the software reads from a file, there is always a new character available. For real-time port reading, however the program has to verify every time that there is something to read from the port, or the program will crash.

After a single line from the port is read, the data is stored in an array in character form. Next, it must be converted to numbers to make the calculations. The C++ command *"sscanf"* was used to parse the position, heading, velocity and solution flag.

3.4 Data Processing Algorithm

The primary requirement of this research is to direct the user to follow a "desired track," i.e., to display the user position with respect to a known path and provide appropriate commands that will be enable the user to follow that path. This must be done by the software in such a way to provide "user-friendly software." The user-friendly software can be described in hundreds of pages, but for this research it can be summarized as, easy to learn and easy to use, with minimum user loading to ensure the objectives are reached.

The desired trajectory was collected as three data sets, which are the center of the path, the inside of the path, and the outside of the path. One goal of the algorithm was to determine where the user is with respect to the desired trajectory data points.

The algorithm has to find where the user is with respect to the path and what the user needs to do to acquire and stay on the track. The navigation data output from the receiver is in local frame reference (Check the CBN message format on page 30). For convenience, the distance with respect the reference point is converted from radians (latitude or longitude differences) to meters. This conversion is performed using latitude and longitude multiplication factors. These factors were calculated for one of the points in the area. Since the changes in multiplication factors for small areas are negligible, the same longitude and latitude factors were used for all points in the real-time system.

The process to find the factor can be explained in two steps:

1. Find the position data in degree units and convert it to radians. Notice that the data format in degree and decimal unit format (DDmm.mmmmm). (The conversion for 39 degree latitude is shown in equation one.)

$$Lattitude = ((Latitude - 3900)/60 + 39) \times \frac{\Pi}{180} \qquad (1)$$

The equations used to calculate latitude and longitude factor are:

$$Lat\_factor = \frac{a \times (1-e)}{(1-e \times \sin^2(Latitude))^{1.5}} + h \qquad (2)$$

$$Lon\_factor = Cos(Latitude) \times \left(\left(\frac{R}{\sqrt{1 - e \times \sin^2(Latitude)}} \right) + h \right)$$

Lat_factor : Latitude factor

Lon_factor : Longitude factor

e　　　 : Eccentricity (0.00669437999013)

R　　　 : Radius of earth (6378137.0)

h　　　 : Ellipsoid height (for this application)

One point on the corner of the area was selected as a reference point for the stored data and the real-time data calculation. The X and Y coordinates in a local horizonal east-north frame are calculated as follows:

$$X\_dist = lat\_factor(latitude_R - latitude_F) \qquad (4)$$

$$Y - dist = lon\_factor(longitude_R - longitude_F) \qquad (5)$$

X_dist 　: X distance information for data processing algorithms

Y_dist 　: Y distance information for data processing algorithms

Latitude_F : Fixed point latitude information

Longitude_F : Fixed point longitude information

Latitude_R : Real time latitude information

Longitude_R : Real-time longitude information

Lat_factor : Latitude factor

Lon_factor : Longitude factor

Another item used by the data processing algorithm is the distance formula as shown below.

$$a = \left(b^2 + c^2 \right)^{1/2}$$

(6)

a : Distance to reference point

b : North (X) difference to reference point

c : East (Y) difference to the reference point

At this point, the algorithm has one real time data point and a set of desired trajectory data points to compare. The closest point (from stored center data points) to the current position is found based on the information described above. The distance from real time position and stored data points' positions are calculated and a stored data point that has the minimum distance to real time position is declared the closest point. The algorithm to find the closest point out of stored points is shown in Figure 10.

```
Min Distance = Very big number

For all center data points

    Difference = Real time position – Stored center data point

    If Difference<Min Distance

        Min Distance = Difference

        Closest point =  Related data point

    End(if)

End (for all center data points)
```

Figure 10 Procedure for Finding the Closest Point

Once the closest point from desired trajectory (stored center data points) is found, the next task is determine what the steering command should be. To do this, the algorithm must determine how far the user is from the desired trajectory, and whether the error is towards the inside or the outside of track.

Several methods for determining whether the error is on the inside or outside were considered. It couldn't be determined whether the current position was on the inside or outside with respect to a constant north or east value because the road traveled was curved. The second algorithm that was tried took the distance with respect to center of the area and then worked from there to determine if the current position was on the inside or on the outside. This algorithm worked for most cases but not every case. This accounted for the fact that the spacing of the data points was not identical between the inside, center, and outside trajectories, due to the varying velocities collected. The stored data points were not identical from an array-index point of view. It was possible to have more than one data point that corresponded to the same center point. The data was processed only one time during the implementation of guidance system in such a way as to have only one data point correspond to one user position for each data file. This process is called the pre-processing of data.

To determine if the user is inside of the track or outside of the track, after finding the closest center point, both the inside and outside data points are searched. Since inside, center, and outside stored data points are in an array format, any acceptable number could be chosen for the search start up, but in this research 100 data points is accepted. This means that the search started from −100 data points of closest-center-data point-index

(stored data files), and ended at +100 data points from the same index to find the minimum distances to the real-time position from the inside and outside stored data points. Then, the minimum distances from the search results (one from the inside data files, one from the outside data files) are compared. Whichever side has the minimum distance is the side that is closest to the current position.

The distance from the centerline is calculated using the closest data point, current position and the next closest data point. The goal is to calculate the line segment "**c**" in Figure 11.
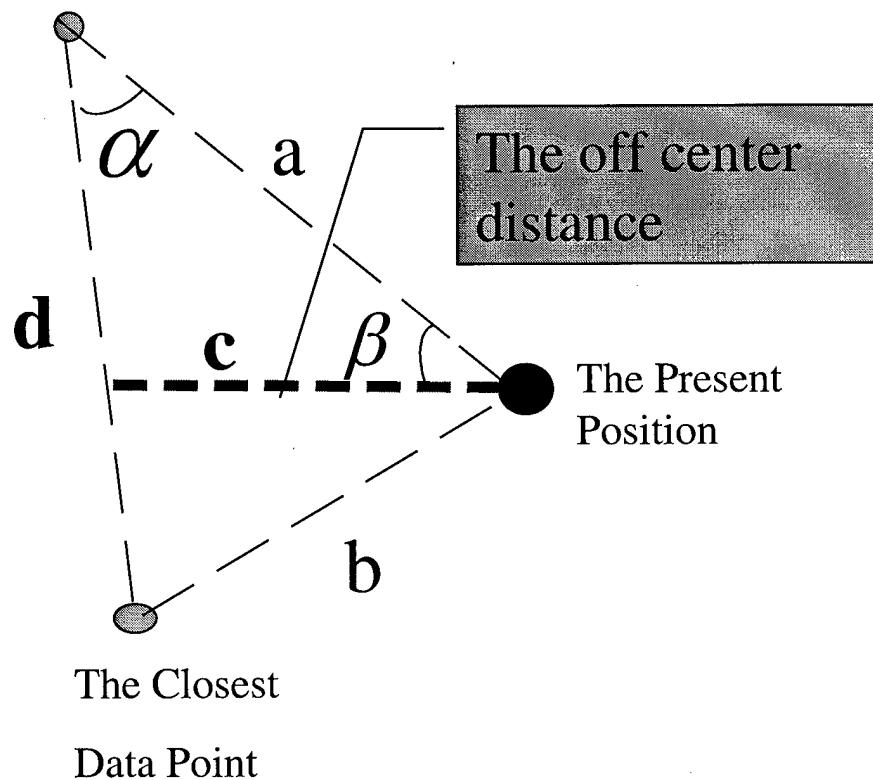
Figure 11. The Off-Center Distance Geometry

28

$$\alpha = \arccos\left(\frac{a \times d}{|a||d|}\right) \qquad (7)$$

$$c = |a| \sin\alpha \qquad (8)$$

a : Vector between the next closest point and the present point

b : Vector between the closest point and the present point

c : Off-center distance

d : Vector between the closest point and the next closest point

Note that, the off-center distance is generally on the order of one meter or less. To be able to display and adopt this information to screen, a coefficient is required. Number 9 is chosen as this coefficient number, and the off-center distance from the calculation above is multiplied by 9 for display on the screen. To make this number big is a conservative approach, because the display will show the off-center distance bigger than it is. If this coefficient is small, the display screen will be smaller than it actually is. A compromise is required, because to make the error smaller is definitely non-acceptable, but on the other hand to make the error very big is not a solution either. Values of 9 and 1 were tested to find optimum coefficient number (see parameter tests in Chapter 4).

The heading information is found based on velocity. The north velocity and east velocity information is available in the CBN message at the output of GPS receiver. The heading calculation can be seen in Equation (9).

$$\psi = \arctan(\frac{Ve}{Vn}) \qquad\qquad (9)$$

ψ : Heading

Ve : Velocity east

Vn : Velocity north

Notice the heading calculation is based only on the velocity. If the velocities (north, east) are very small, the heading may not be as accurate. (See the results in Chapter 4). This can especially be a problem if the velocity is very small and heading changes are sharp.

As a preprocessing step, headings are calculated for every point on each of the desired trajectories (inside, center and outside). The real-time heading information is compared with the stored heading information to find the heading error. In order to be more accurate when calculating the heading error, every center point heading is combined with its inside and outside data points' headings to avoid any small error that could be caused by driving. During field operation, the inside, center, outside data points heading may not precisely reflect, individually, the heading of that point. For example, stored center heading for a single point can be two degree off what it should be, but after combination of three headings, the off degree error should be minimized as shown in Equation (10).

$$H = (H_i + H_c + H_o)/3 \qquad\qquad (10)$$

H : Stored heading calculation for a point

H_i : Stored heading from inside of the track

H_c : Stored heading from center of the track

H_o : Stored heading from outside of the track

To find the heading to compare with the real-time heading, the following must be considered: The heading information provides some idea of the future position. If only the stored closest data point heading information is used in the display, the user will not be prepared for future events. When driving any vehicle, the most valuable information is that which prepares the drivers for the up coming action. The stored data points were collected on the same driving path, so all the information about the desired path is known. This information can be used to prepare the user for the up coming events. Notice that, the information is stored in an array format. Thus, the future point can be described as some number of data points later. If number 8 is chosen, then 8 stored data points later (it can be thought also as few seconds later) is called the future point heading. This number is not an absolute number, and it can be changed by dynamics of vehicle if necessary. The future reference point heading and closest reference point heading information are combined in a weighted way. The combination can be seen in Equation (11):

$$H_s = w_c \times H_c + (1 - w_c) \times H_f \qquad\qquad \textbf{(11)}$$

H_s : Calculated stored heading

w_c : Weight of closest point heading

H_c : Closest point heading

H_f : Future point heading

After finding the stored heading information, the only calculation that remains for the calculating heading error is to subtract the real-time heading shown in Equation (12).

$$H\_error = H_s - H_r \qquad (12)$$

H_error: Heading error

H_s : Calculated stored heading

H_r : Real-time heading

3.5 Graphical Interface

The research involves creating a real time user interface that is able to work in a kinematic environment (the RAVEN). Based on the "designing visual interface principle," the minimization of the number of components and simplifying the relationships between them are very important for designing a good visual user interface [6].

To find the "correct" instrument for the display was a challenge. The research started with a course indicator. The course indicator instrument, which is an instrument used in aircraft, can show the user the position with reference to the center of a path. Another idea was to show the road position from the user's point of view.

The software life cycle is a continuous process but to present it, some significant stages are chosen. Different prototypes will be presented in order to show the progression of thought during the software development.

3.5.1 First Prototype

The first prototype is shown in Figure 12. The right-upper rectangle was intended to be the place to put the desired track. The OpenGL command that is used for rectangle is *glRect*, which allows the programmer to define the limits of the rectangle. (See Appendix A for more information about the OpenGL commands used in this section [7]).

The circle on the left is the course indicator. It is created by using *glutSlolidCone*. This command is used to create a cone, but the height is specified to be zero so it is actually an ellipse. In the ellipse, the line is used to show the desired path position with respect to the current position. The commands *glBegin* and *glVertex* were used together to draw a line between two points.

The command *glTranslate* is used to move the object off-center by a given value. The problem is that when it moves off-center too much, it can be seen outside of the circle. In order to prevent this, *GlStencil* applies a test that compares a reference value with the value stored in stencil buffer. Depending on the test result, the pixel will be displayed on the screen or not, so that the unwanted part will not displayed (This has the effect of "masking" out a particular region of the screen).
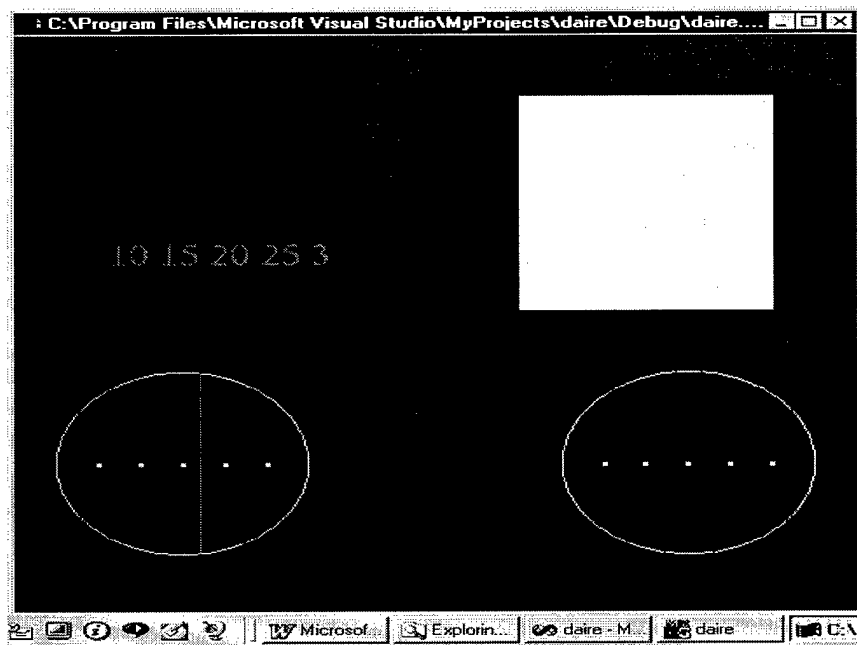
Figure 12. First Prototype of Guidance Display

The displaying of numbers and letters is somewhat difficult because they have the same properties as any graphical object. *GlutBitmapCharacter* is used to display a character and a loop is used to show more than one character.

The first prototype simulated motion using the mouse and keyboard. The *glutMouseFunc* was used to give the ability to interact with the user. When the right mouse button was clicked, the course indicator line would go right, and user position would be simulated on the left. The opposite was true for a left mouse click. The simulation of the road was a simple drawing consisting of two lines that looked like a road.

The first prototype was obviously not ready to implement in a real-life situation. However, the first prototype did obtain the goal of demonstrating, the basic graphics display capabilities. The lessons learned from the first prototype were as follows (positive shows what is gained from the first prototype, negative shows that either it could be implemented or it couldn't be satisfactorily implemented):

Positive

-   General idea of which instruments can be implemented.

-   How to implement a simple instrument.

-   How to move an element of an instrument.

-   How to mask an unwanted part of the display.

Negative

- The instruments were very simple.

- The zooming didn't work.

- The relative road display (left lower) was not able to display relative position.

3.5.2 The Second Prototype

The simulated velocity, heading, and current position displays were combined in one display. The thought was to create a display that was similar to an aircraft's HUD (Heads-Up Display). The same sort of information was to be displayed by the software package. The second prototype is shown in Figure (13).
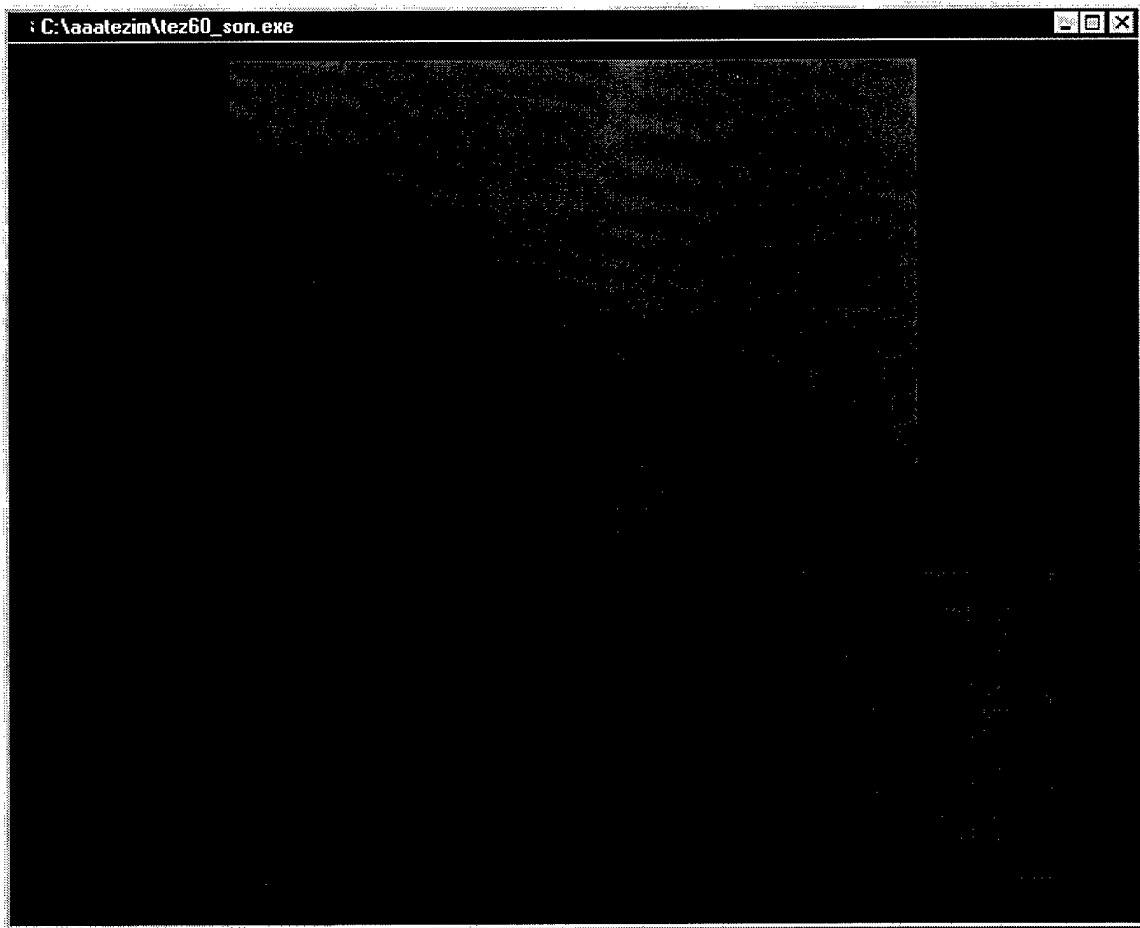
Figure 13. Screen Capture of Second Prototype

The map (right lower display) is generated using three rectangles with the same style that was described before, but a different color is used for the middle rectangle. One important feature for the map is zoom. The function *glScale* is used to stretch or shrink the object. One problem occurred after scaling, the current user position could be out of the window, because the stencil buffer was used for the path. For that reason, the appropriate translate functions were applied to ensure that the current position was always located in the region of the screen path. The other concern was that after zooming, the desired track could be seen in the part of the display that was not reserved

for the desired track. The use of *glStencil* and a different depth buffer solved this problem.

The map was divided into four different roads called the west, south, east and north roads. The random number generator was used in such a way that, in a different road the generator produced different random number. This was necessary, because in the south road (the bottom road), there is horizontal movement, but in the east road there is vertical movement. Thus, a random number was used to vary all of the instruments in order to be ready for the real time application.

The current position display with respect to road is called the "desired path display." To be able to display the current position with respect to road, some calculation had to be done. The desired path calculation requires off-center distance and heading information.

From Figure 14, the top and bottom center point of the desired path display is known and is called the bottom middle point and top middle point. The top distance and bottom distance are chosen experimentally to reflect the real road view. To be able to show the desired path display, the dynamics are to be calculated given the off-center distance and the heading error information. The following calculations are applied to adopt the related display to real-time.
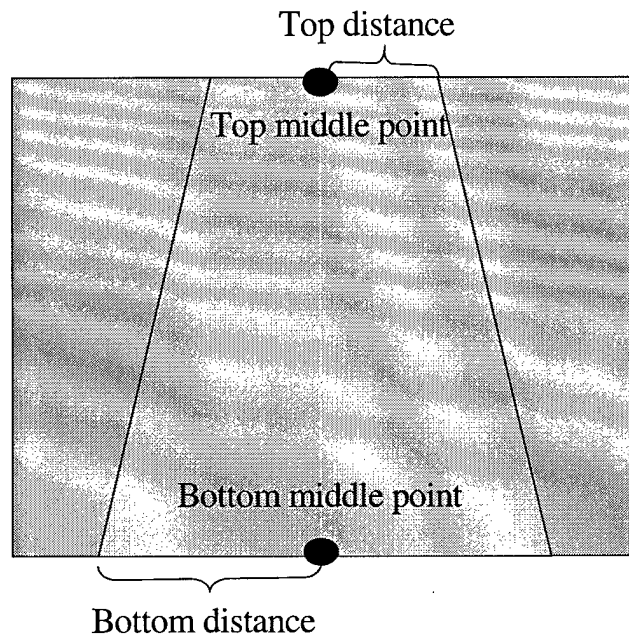
Figure 14. Desired Path Display

$$dyn\_b = b + dK \tag{13}$$

$$dyn\_t = dyn\_b + hk \tag{14}$$

dyn_b: Dynamic bottom middle point

dyn_t : Dynamic top middle point

b : Bottom middle point

d : Off-center distance

K : Coefficient for bottom

h : Heading error

k : Coefficient for top

After finding the calculated dynamics (bottom and top middle point) of the desired path display, the bottom and top distance value are added and subtracted to form the left side and the right side of the desired path display.

The second prototype was much more sophisticated, and the guidance display was ready to be connected to the real-time system. Notice that second prototype was not working with the GPS data, but with a computer generated random number. The summary of the gain of the second prototype can be seen below:

Positive

- The implementation of instrument idea was clear.(HUD idea)

- The relative road implementation (desired path display) was not a simple two lines movement, but was quite sophisticated compared to first prototype.

- Zooming was implemented

- All the moveable instruments were changed based on a random number. This was significant step in the transition to a real-time application.

Negative

- The map display was still a rectangle.

- The display instruments were not ready to reflect a real-time situation.

3.5.3 The Third Prototype

Once the RS-232 data port reading algorithm was working, the graphical interface was connected to the data processing algorithm. At this point, the software was capable of processing real-time data. The connection was made by using *glutIdleFunc* (Data Process Function), which specifies the function to be executed, if no other events are occurring. The Data Process Function's first job calls the port reading function. The third prototype is shown in Figure (15).

Figure 15. Screen Capture of Third Prototype

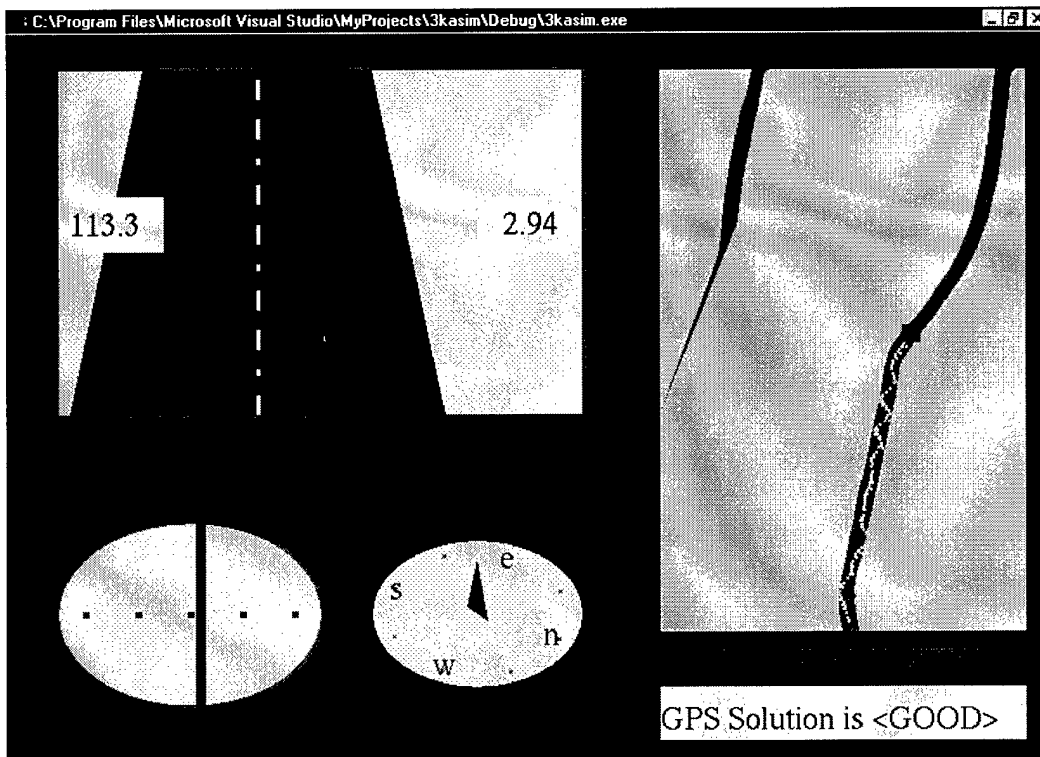The relative positioning display (top left display) concept remained the same as prototype two, and it was recognized as the main display instrument. *GL_QUADS*, which draws four-sided polygons, was used to show the current position with respect to the desired track in as realistic a picture as possible. The bottom left, bottom right, top left and top right coordinates of the desired path display (on the top left display), were calculated in the data processing. The dashed-line which is in the middle of the road is drawn by using *GL_LINE_STIPPLE*.

Note that the additional altitude window that was in the second prototype has been removed for simplicity. Within the relative positioning display, the left small window shows heading and the right small window shows velocity. One problem that was solved was that the velocity was in floating point format, but *glutBitMapChracter* displays only characters. The velocity was converted from the floating point number to characters using the *SPRINTF* command. The triangle in the bottom of the relative positioning display is similar to the aircraft radar, where the triangle represents the vehicle.

The map (upper right display) was made larger than in the second prototype, because in the second prototype it was hard to see, so it was not effective. The purpose of the map display is to give the big picture to the user so the user can see what might be expected. Using the zooming feature, the user can see a more detailed display near the current vehicle location. Recall that this system is designed to be able to use the vehicle without seeing the path. Because of this, the map display is a great benefit for establishing situational awareness. The considerations were such that the map should be scalable properly and fit the display part reserved for it, and for that reason an appropriate scale

factor was needed. All the real time and stored data was displayed using the same scale factors. The scale factor for the X and Y axes were found experimentally. The multiplication factors were different because of the compressing the map in the Y direction so that it fit better on the screen. The scale factors were calculated as follows:

$$Sx = 0.5Z \qquad\qquad (15)$$

$$Sy = 0.35Z \qquad\qquad (16)$$

S_x : Scale Factor for X

S_y : Scale Factor for Y

Z : Zoom Factor

When zooming is used, a portion of the desired track could exceed the display region. One issue was to determine when the current position should be placed on the center of the map. It was empirically determined that when the zooming factor is greater or equal to four, the current user position should be on the center of the map. For every increase of the zooming factor (prior to reaching four), the current display indicator will get closer to the center of map display. When the zooming factor is greater or equal to four it will be on the center, but the zooming will still cause the map to enlarge.

In addition to the other displays, the validity of the GPS solution is checked. If carrier phase differential does not work, the system does not have centimeter-level accuracy. If the position solution is not good enough anymore, the user should not continue to use the system. To understand if the solution is good or bad, the solution flag is passed to the display function from the real-time data port reading function. The desired solution flag is 221001, which means 3D solution, Carrier Phase Differential

(CPD) solution, float solution, updated solution with measurement update, normal CPD

solution and fast CPD solution [2].

The heading indicator (located in the bottom left window, right-side) is designed

for a user who is not familiar with reading and reacting to the numbers. It provides

enough information at a glance. The techniques used to draw the heading indicator are

the same as described above, but to get rotation, the function *glLookAt* is used.

The course indicator shows how the user should steer the vehicle. It is restricted

to limit how much the needle can go off-center, because if the off-center distance is too

big, the indicator line can move to a point that can't be seen. This is undesirable, because

even if it doesn't show off-center distance precisely, it still should show some

information about the correction that should take place. The course indicator is designed

to be more precise for small errors, yet at the same time be able to show big errors. The

following transformation is performed on the off-center distance information coming

from the data process algorithm.

$$N = 10 \times \frac{(X)^{0.8}}{10^{0.8}} \tag{17}$$

N : New off-center distance to display in course indicator

X : Off-center distance

Remember the off-center distance is multiplied by a coefficient (around 10) in the

data process algorithm to adopt to the graphical display. After applying Equation (17),

the off-center distances smaller than 1 meter can still be displayed precisely, and the off-

center distances bigger than 1 meter can still be displayed also. It is important to note that

these exact numbers are accurate for this application only, but they can be easily adopted to any application.

3.6 Summary

This chapter presented the design and implementation of guidance display system. It included all of the underlying hardware and software architecture. The port reading, data processing, and graphical algorithm from the software architecture was described in detail. The next chapter will present the test results of the guidance system.

4. Results

There are many tests that could be done with many different people, but unfortunately due to the weather conditions and time considerations, only a limited number of tests were performed.

Four tests were used to evaluate the guidance system performance and human factors under different circumstances. The users were told to drive at a comfortable speed and try to stay as close as possible to the center of track. After every test, the standard test evaluation, was given to the user to fill out (full version can be seen in Appendix B, the brief answers from the users can be seen on Table 2 and Table 3 on page 65 and 80 respectively).The tests conducted were update rate tests, single instrument test, varying parameter set tests, and under hood test. The guidance system was tested with four different update rates: 1, 2, 3.3 and 5 Hertz. A single volunteer was used to compare these different updates rates. For all of the other tests, two volunteers were used to test the guidance system. In the parameter tests, the objective was to monitor the guidance system performance with different parameter sets which will be explained in detail later. One instrument test's purpose was to see how much the instrument could be simplified, and the human reactions to this simplification.

4.1.The Update Rate Test

When comparing between different update rates, it will be seen that the number of data points that are between specific ranges will change in the results. This is because the number of data points increases as the update rate increases. Therefore, the reader should not compare the results based only on the absolute number of data points, when comparing different updates. The important observation are: 1) The center (mean), 2) the

total variation, and 3) the overall shape of the distribution. These three important observations are valid not only for the update rate tests, but all the tests that will be presented.

4.1.1 1 Hertz test

The user had average speed of 2.80 m/s, average distance from the center of 0.19 m, and standard deviation for the off-center distance of 0.25 meter. As can be seen from Figure 16, the user has more data points on the right of the track. There were 118 data points on the right verses 85 on the left. The reflection of this on the *mean error* (based on the assumption that the error on the left is negative and the error on the right is positive) is 0.07 m. The maximum off center distance was 0.85 m, meaning that even with a one Hertz update rate, he remained on the track at all times.
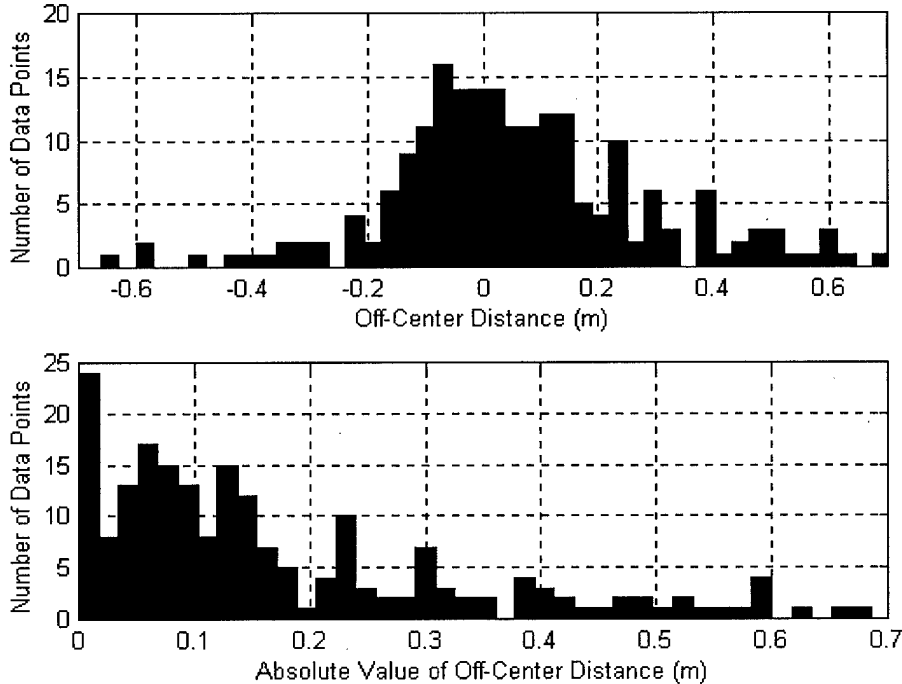
Figure 16. Position Error Histogram for 1 Hz Test.

The calculations show that 96% of the collected data for this test, has an off-center distance under 0.6 meters, 44 % of data points have an off-center distance smaller than 0.1 meter. The other data distributions were as follows: 23 % of data points were between 0.1 and 0.2 m, 11 % of data points were between 0.2-0.3 m, 9 % of data points were between 0.3m and 0.4 m, and the other 8 % of the data points were between 04 m-0.6 m.
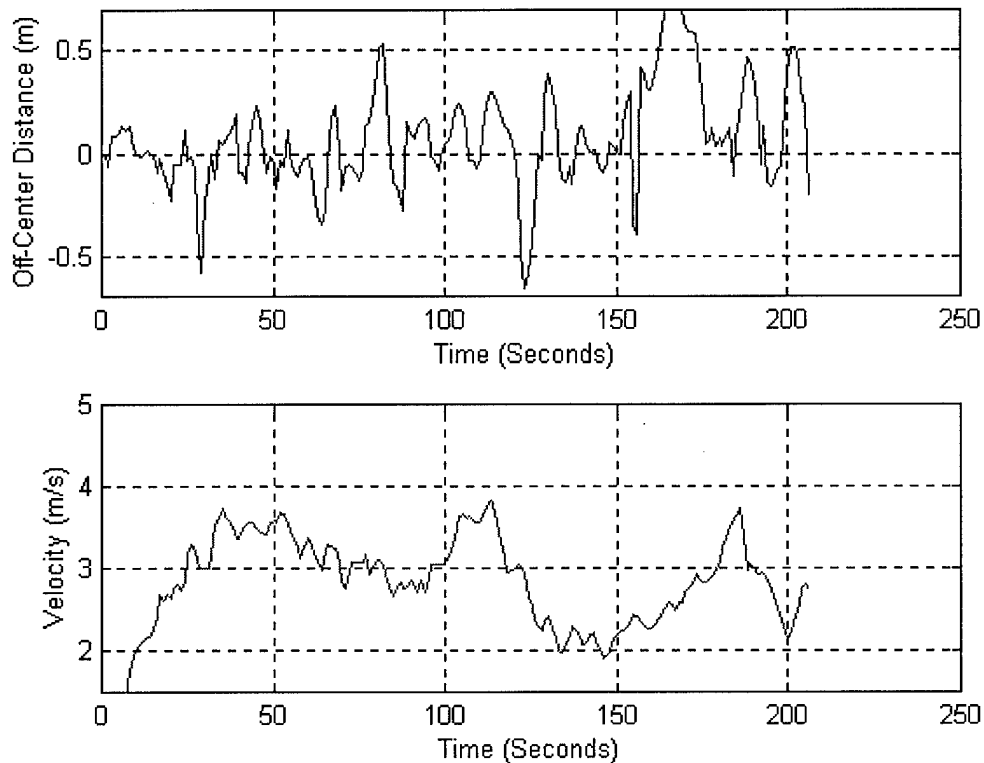
Figure 17. Position Error and Velocity Plot for 1 Hz Test.

Figure 17 shows how the user went off the track and recovered back, periodically. The speed varied between 2 and 3.7 m/s. Note that on the curved road (approximately 125- 175 seconds) the speed was significantly slower compared to the straight road (This is true for all test cases).

47

The user pointed out that the guidance system was working, but it did not really give enough confidence. The standard test evaluation paper was filled out by person A using a scale of one to ten. The answer to several of the key question are shown in Table 2 – page 65 - (a complete listing of the questions can be found in Appendix B).

In the final analysis, the 1 Hertz update rate is not fast enough to satisfy real-time requirements, because it does not give the required confidence to the user. The problem with a low (1 Hz) update rate is demonstrated in Figure 18. If the user is off-center, the software will indicate this, and the user will start to go to the center based on what is shown on the screen. Since the update is not available until next second, the user will believe that he or she is still on the left (for this example). The new update is not available until after the user crosses the centerline. Based on the new update, the user will start to give a new correction, but this causes an 'S' maneuver.
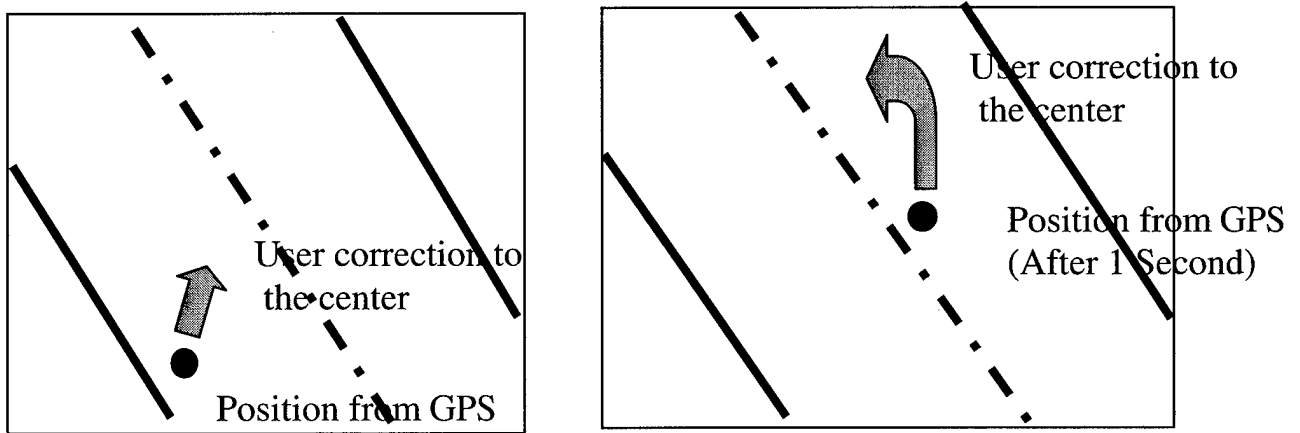
Figure 18. Navigation with Slower Update Rate
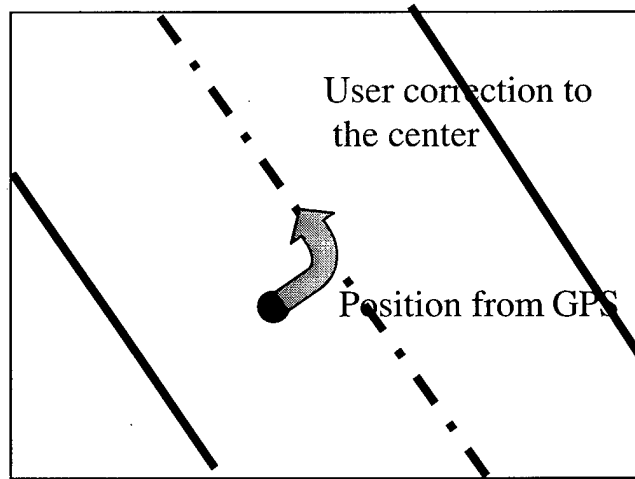
Figure 19. Navigation with Adequate Update Rate

When the update rate is increased the system will give more immediate feedback to the user, which will help to avoid the overshoot that is common for lower update rates. This concept is shown in Figure 19. If the update rate is increased, the guidance display will inform the user during the approach from left to right. Based on the information provided on the display, the user will be less likely to cross over the centerline.

4.1.2 2 Hertz test

The update rate was changed to 2 Hz, using the front control panel of the receiver. All other setting were the same as the 1 Hz update rate. The average speed was 3.25 m/s, which was almost 0.5 m/s or 17% higher than 1 Hertz test speed. Similarly, the average off center distance magnitude was 0.212 m which was only 0.02 m higher than 1 Hz test. The standard deviation of the off-center distance was 0.34 meter. The maximum off-center distance was 1.48 meter.

Figure 20 shows that the error quantities are considerably small at the beginning of the test. Another observation is that the errors (off-center distance) are bigger on the curved road compared to the straight road. The collected data points approximately

49

between 120 and 140 seconds are correspond to curved road. There can be two reasons for this increase. First, it is in the nature of driving to make more mistakes on the curve. The second and most important one is, when driving very slowly , the heading approximation may not be as accurate, due to the way heading is derived from the GPS measurements (See Chapter 3.4 for detailed information).
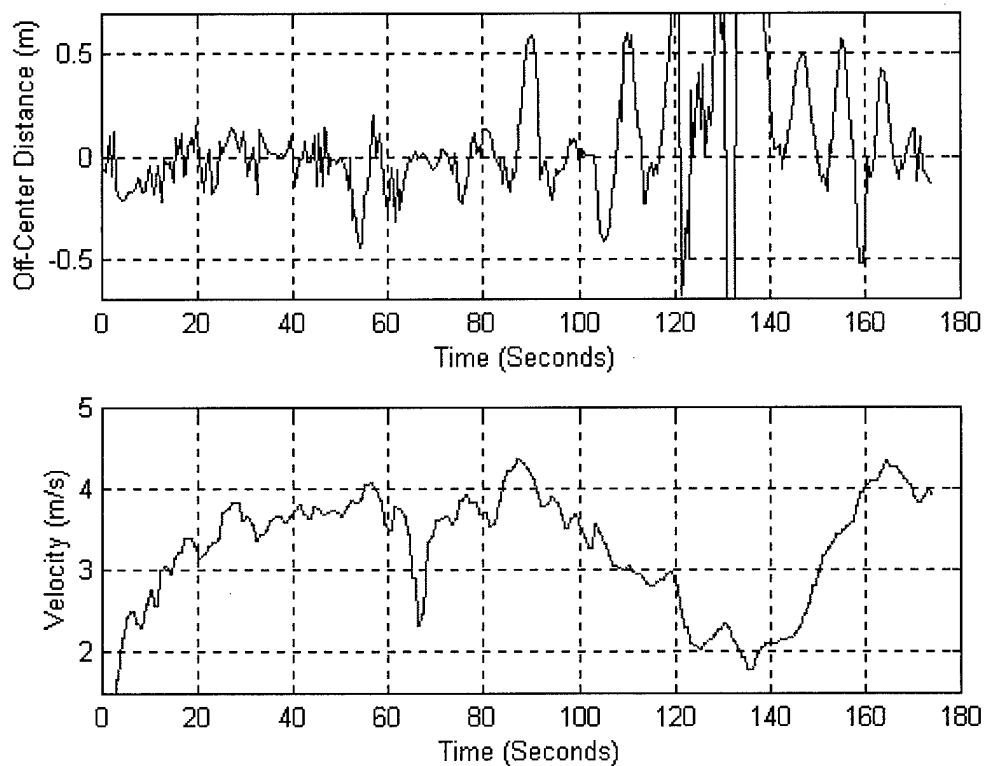
Figure 20. Position Error and Velocity Plot for 2 Hz Test.

As shown the Figure 20, the speed is generally between 3 and 3.5 m/s and the errors are more tightly bounded (except the curved road) than the 1 Hertz case. Now,

92% of collected data points were below an 0.6 m error and the 46% of collected data

points were below 0.1 meters. Also 24% of data points were between 0.1 m and 0.2 m,

7% of them were between 0.2 m and 0.3 m, and 14 % of data points were between 0.3

and 0.6 meters. Based on Figure 21, the general distribution of data is much more

normally distributed than the 1 Hertz test.
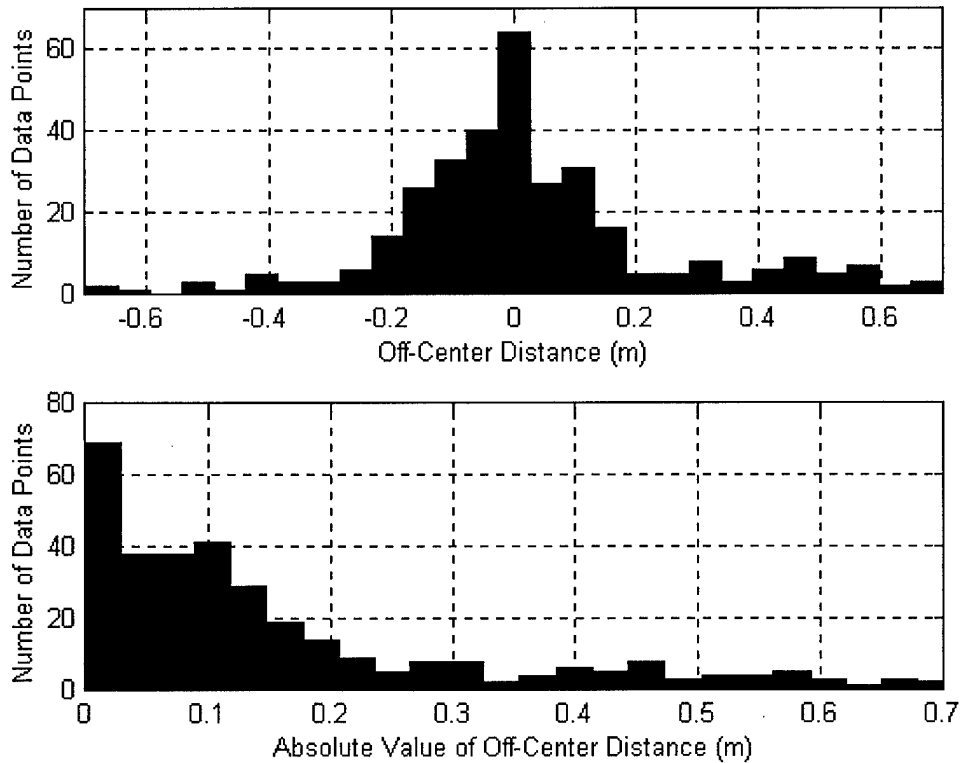
Figure 21. Position Error Histogram for 2 Hz Test.

From the user point of view, the 2 Hertz update rate was much better than a

1Hertz update rate. The evaluation of the user also shows that the user was fully satisfied,

because they evaluated every question with ten (based on a one to ten scale -- see Table

2. Page 65 section 4.1.5-- for evaluations). Clearly, the 2 Hz update rate give more

confidence compared to 1 Hz update rate based on user feedback.

4.1.3 3.3 Hertz Update Rate Test

The update rate changed to 3.3 Hz (every 0.3 second) from the control panel and all other settings were the same as the 1 Hz and 2 Hz update rate tests.

Average speed was 4.04 m/s, which was 44% higher than 1 Hertz test and 24% higher than 2 Hertz test. Average off-center distance magnitude was 0.25 m, which was 19% higher than 2 Hz test, 35% higher than 1 Hz test. The standard deviation of the off-center distance was 0.33 meter. Figure 23 shows position and velocity plot for 3.3 Hz update rate.
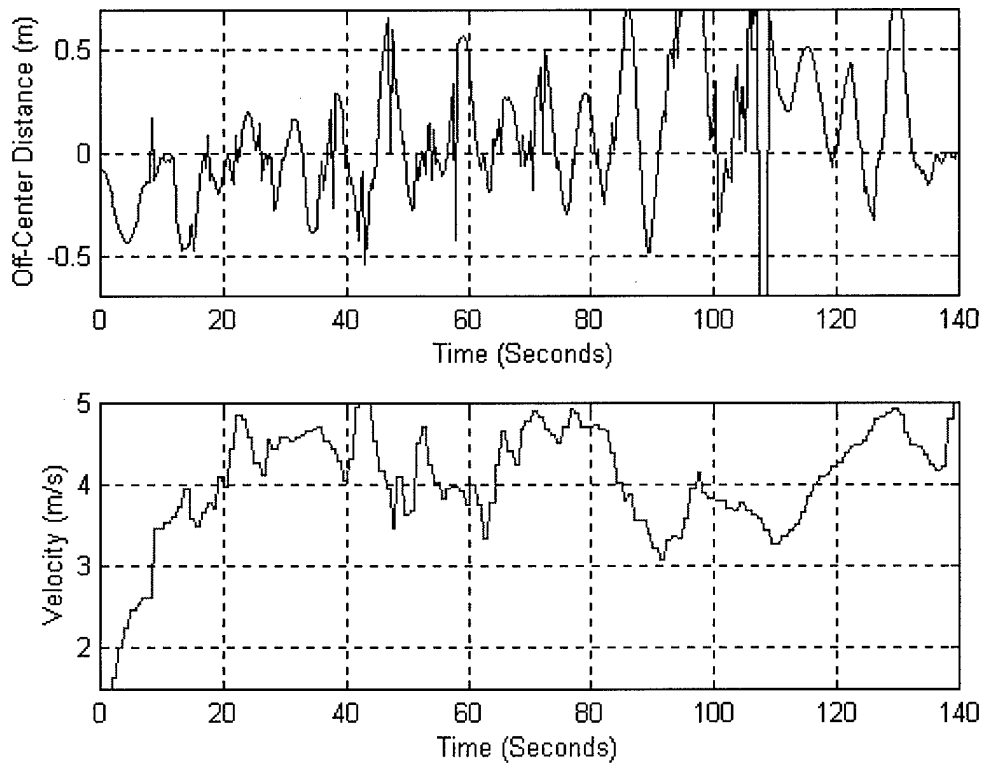
Figure 22. Position Error and Velocity Plot for 3.3 Hz Test.

The maximum speed was up to 5.39 m/s and the maximum off-center distance was 1.09 meters. There were 232 and 227 data points on the right and left sides respectively, and the mean error was 0.07 meters.

91% of the collected data points had an off-center distance magnitude below 0.6 m and 30% of them were equal to or below 0.1 meters. The other data distributions were as follows: 20% of them are between 0.1 m and 0.2 m error, 15% of them were between 0.2 m and 0.3 m error and other 24% of them were between 0.3 and 0.6 meters. Compared to the previous tests, the 3.3 Hz update rate had the highest average speed, indicating that the user felt most comfortable with this test. Figure 22 shows data distribution and the off-center distance histogram.
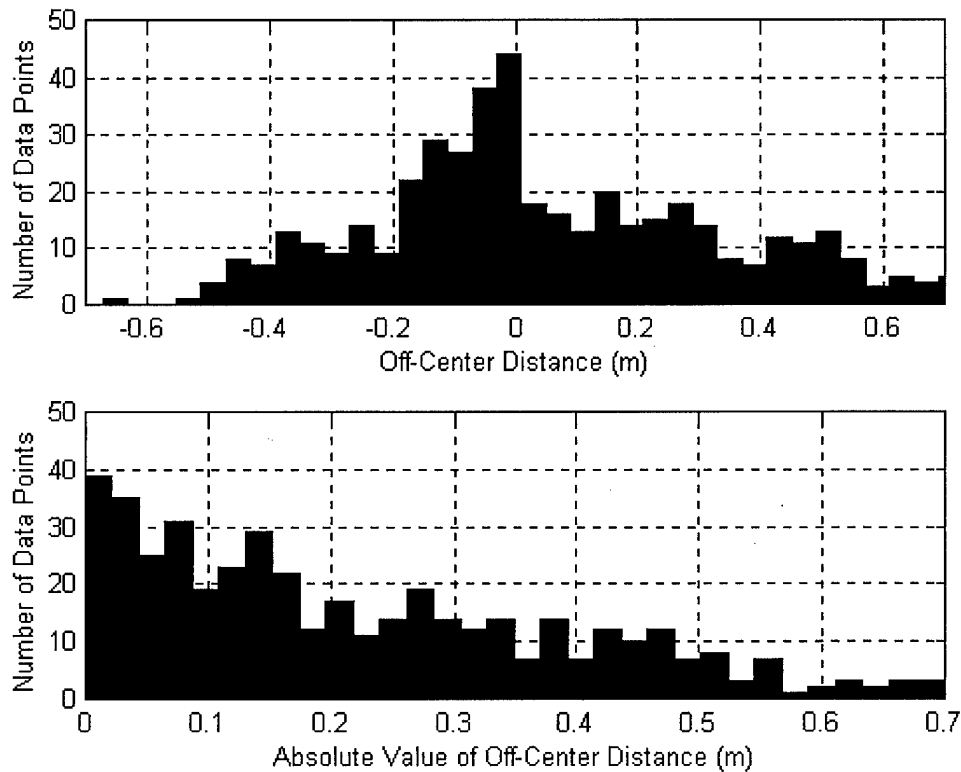
Figure 23. Position Error Histogram for 3.3 Hz Test.

4.1.4 5 Hertz Update Rate Test

The update rate was to changed to 5 Hz from the front control panel, and all other settings were the same as the other update rate tests.

The average speed was 3.70 m/s, the average off-center distance magnitude was 0.23 m and the standard deviation was 0.31 meters. Compared to 3.3 Hz update rate the velocity is almost 10% lower and the off center distance is 10% lower. This implies that the user felt less comfortable with 5 Hz update rate, but he did do better with respect to the off-center distance. Note that decreased velocity may be part of reason for the improvement in off-center distance.

The maximum speed was 5.02 m/s and the max off-center was 1.10 meter. 91% of the data points were below 0.6 m and 36% of the collected data points were equal or below 0.1 meters. 23% of the collected data points were between 0.1 and 0.2 m error, 13% of them were between 0.2 and 0.3 m error and 19% of the data was between 0.3 and 0.6m. There were 270 data points on the left versus 283 on the right and the mean error was 0.04 m. The off-center distance distribution can be seen on Figure 24.
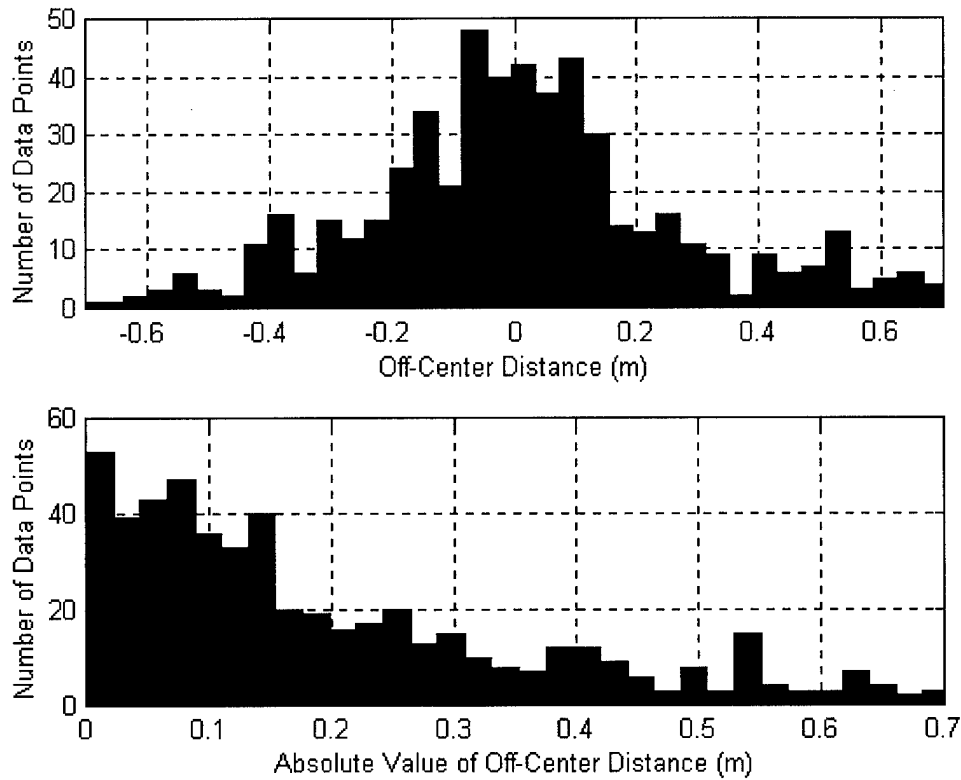
Figure 24. Position Error Histogram for 5 Hz Test.

Notice that his dynamics were also comparatively high, but not higher than the 3.3 Hz update rate test. The user thought that this update rate was too high to feel comfortable and the 3.3 Hz update rate gave a better confidence and comfort for the guidance system. The numerical results show that this update rate results in an accuracy nearly as good as the 3.3 Hz update rate. The off-center distance and velocity plot can be seen on Figure 25.
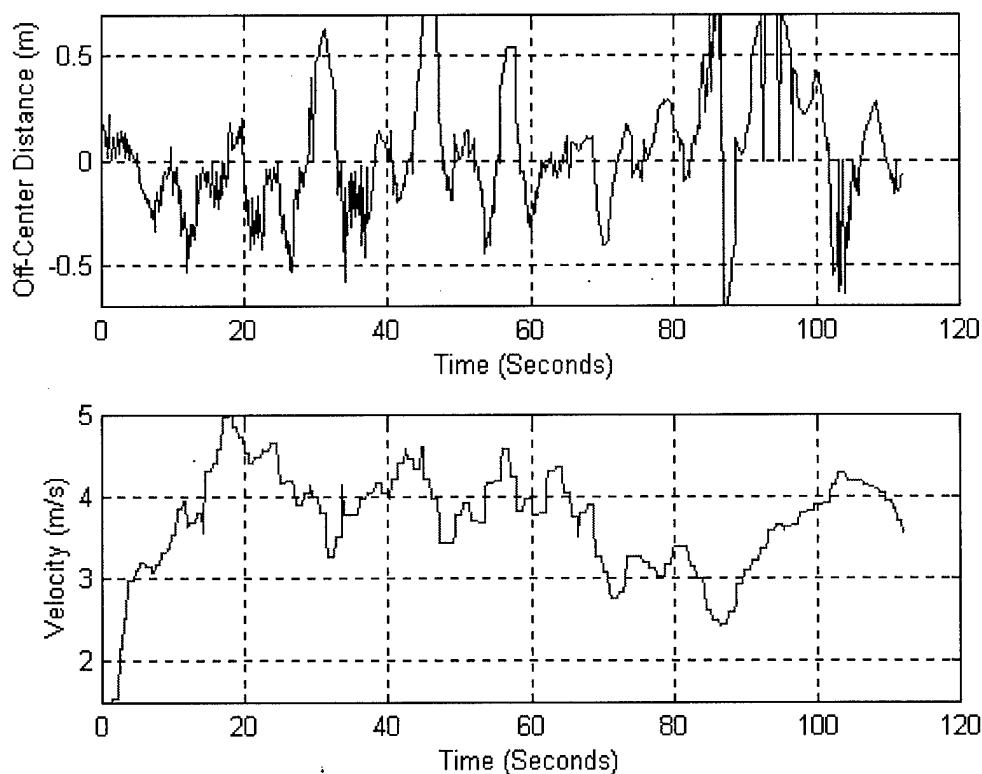
Figure 25. Position Error and Velocity Plot for 5 Hz Test.

4.1.5 Summary of Update Tests

Different update rate tests and the effect of them was explored during the update rate

tests. During these tests, especially the velocity difference between different update rates

was especially significant. Note that dynamics were directly related with confidence

level during tests and the error quantities. If it takes 1 second to reach the center of the

desired path with velocity of 2 m/s, it will take 0.5 seconds for a user who has a velocity

of 4 m/s. This will force the users to be smooth and cautious or make more errors. Note

that the algorithm that was used in this research assumed a constant velocity to find the

guidance information (for example, by always choosing a constant number as a future

reference point). The algorithm could be modified to take into account the velocity when determining the corrections and future reference point.

The feedback from the user shows that the 3.3 Hz update rate is the most convenient update rate for the comparatively high dynamics for the golf car. The test driver said that "the best update rate and best confidence" was during the 3.3 Hz update rate test. Based on these observations, the 3.3 Hz update rate appears to be best, compared to other tests (including 5 Hz). Still, some users may prefer a different update rate. The summary answers of the test evaluation questions can be seen in Table 2.

Table 2. Summary of Key Answers of the Test Evaluation Questions (Update Rate Tests)

	1Hertz	2 Hertz	3.3Hertz	5Hertz
The software meets the objective	9	10	10	10
The software is easy to use	10	10	10	10
The instrument is well organized	9	9	10	10
The instruments are adequate to use the car	9	10	10	10
The interface matches the real-time Situation	10	10	10	10
The overall experience rate	10	10	10	10

4.2 Single instrument test

In general, it is best to keep display as simple as possible. The single instrument test was developed to see if a single instrument (display) would be sufficient to enable the user to stay on track. Since the desired track display is able show to the position with respect to a known path and provide appropriate information that will be enable the user to follow that path, it was chosen as the single instrument display. The user interface shown in Figure 26, was developed. The status display (right lower) shows the current status of system reliability information and the "caution display" (left lower) warns the user to be more cautious in the high turn area.
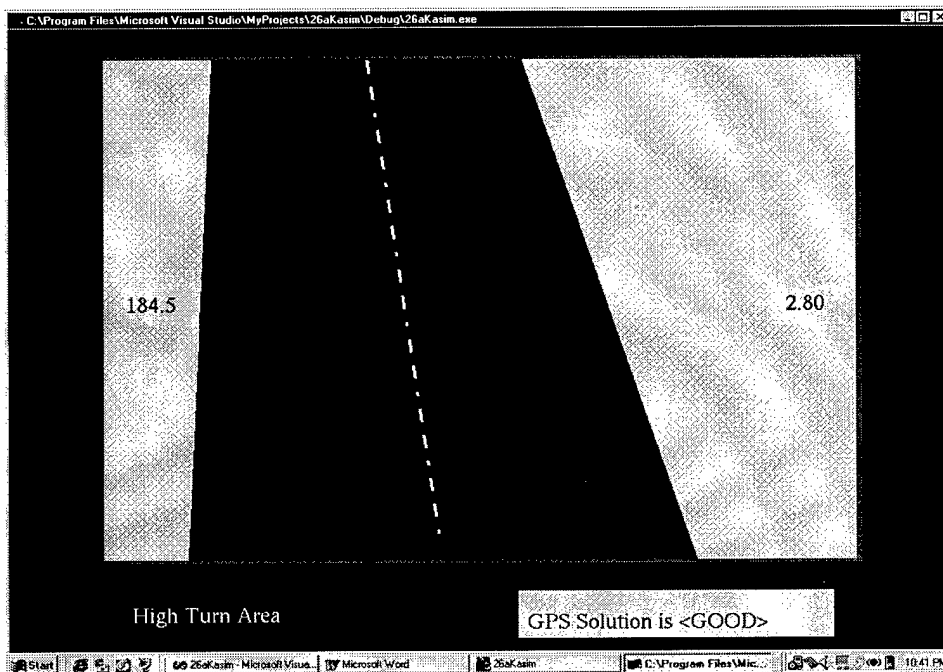
Figure 26. Single instrument Test Display.

Two people tested the user interface; person A had average speed 2.9 m/s and average off-center distance is 0.11 m. and the standard deviation was 0.20 meters. Person B had an average speed of 2.80 m/s, and the average off-center distance 0.15 m., and a standard deviation of 0.13 meters. A position error histogram for both users is shown in Figure 27.
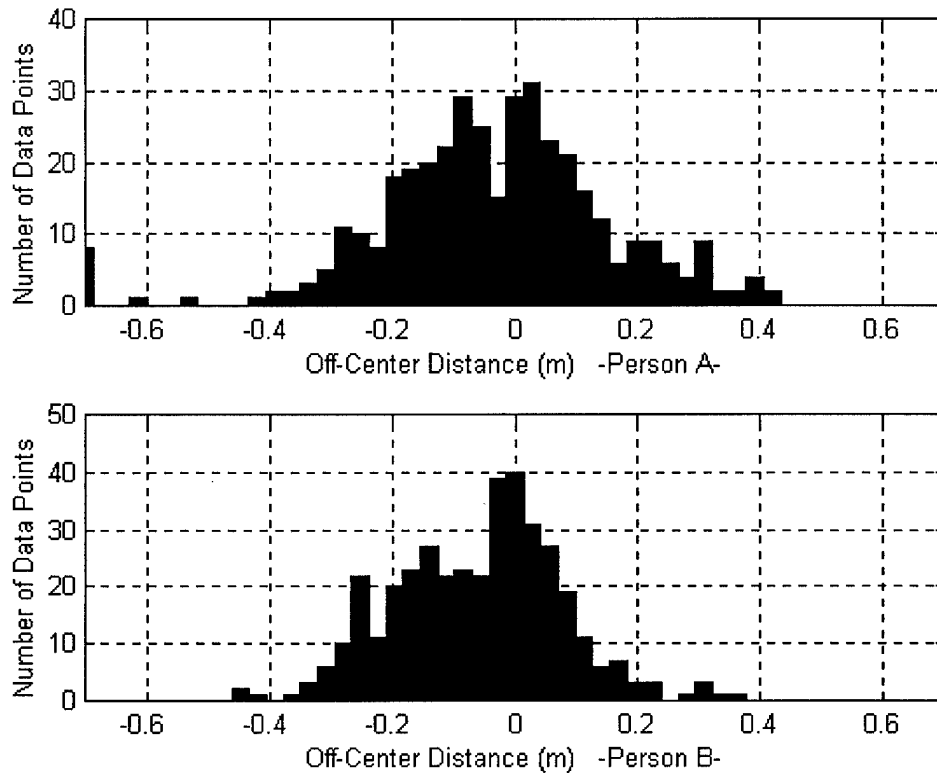
Figure 27. Position Error Histogram of Both Users for Single Instrument Test

For person A; 97% of the collected data points were below 0.6 m error and 45% of the collected data points were below 0.1 meter. 27% of them were between an 0.1 and 0.2 m error, 16% of them were between 0.2 and 0.3 m error, and 8% were between 0.3 and 0.6 m. There were 209 and 169 data points on the left and right sides respectively, and the mean error was -0.03 m.

For person B; 99.9% of the collected data points were below 0.6 m error and 53 of collected data points were below 0.1 meter. 29% of the collected data points were between 0.1 and 0.2 m error, 13% of them were between 0.2 and 0.3 m error and 4.4% were between 0.3 and 0.6m. The number of data points collected from the left of track and right of the track were 247 and 130 respectively, and the mean error was -0.05 m.

If the numbers are closely monitored, the personal differences between users can be explored. For example, person A had 46% of his collected data points under 0.1 m off-center distance. At the same time, person B had 53% of his data points under 0.1 meter off-center distance.
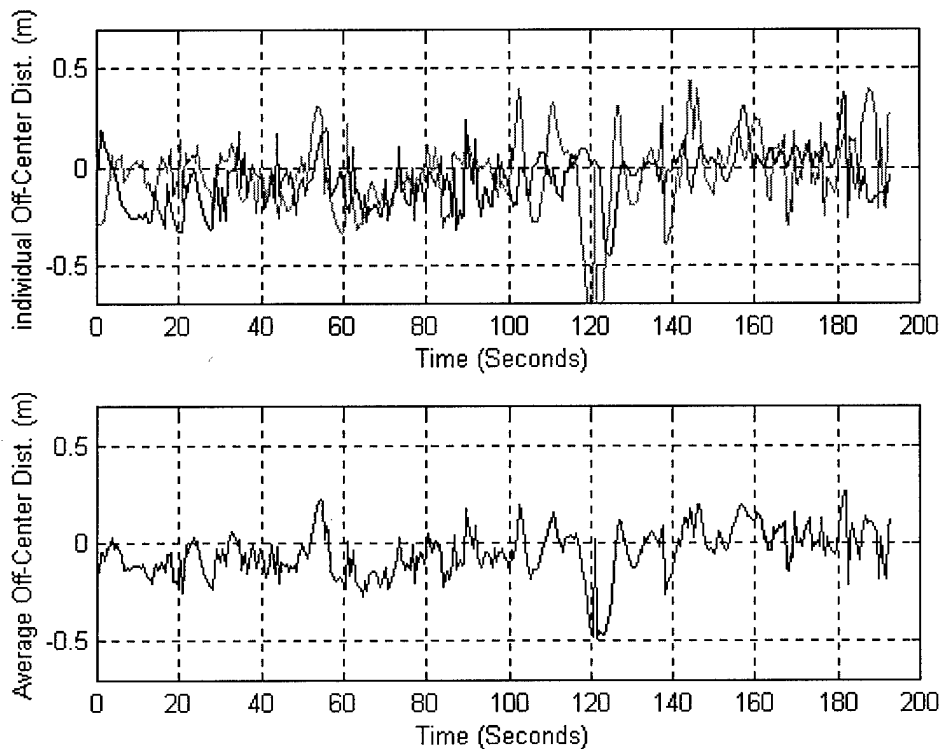
Figure 28. Position Errors Plots, for Single Instrument Test

Figure 28, shows the average off-center distance of both users. There is a similar pattern in terms of off-center distance between user A and B until 100 seconds. These errors are fairly small prior to 110 seconds, but between 110 and 130 seconds the error are bigger. This is because it is more difficult to stay on the center in high turn area than on the straight road, especially without the map display as a second instrument. The lower subplot shows the average of both users. The average speed was 2.87 m/s and average off-center distance was 0.1 m, and the standard deviation was 0.12 meters.

The maximum speed for person A was 4.70 m/s and maximum off-center distance was 0.45 m, at the same time, the maximum speed was 4.34 m/s and maximum off-center distance was 0.99 meters for person B (these numbers include both the curved and the straight part of track).

The consensus from the users for this test was that a single instrument was not sufficient, especially in the turn area, where the map especially becomes a necessary instrument. User A answered "the instruments are adequate to use the car" question (from the evaluation sheet) with 8 (on a one to ten scale), and he rated the overall experience with 9 (see Table 3 for other answers page 80 section 4.5). The bottom line is that it is easy to check only a single big instrument, but there should be more than a single instrument to give the user situational awareness.

4.3 The Parameter Test

Three parameters were simultaneously varied in the user interface, and the two parameter sets were tested in the field. The parameters are described in the equations in Section 3.4. The calculated off-center distance coefficient was accepted as 9 to get a "good fit" for the graphical interface. In other words, this is the scaler used for the off-

center distance for the graphical display. Off-center distance coefficients of 9 and 11 were used for parameter sets A and B, respectively.

The second parameter involves heading. When the current heading is compared to the stored heading to find the heading difference to display, the stored heading is found by a combination of two heading points. These two points are the closest point and a future point. The closest point is the nearest stored point to the current position and the future point is a point in the future (See Section 3.4 for a more detailed description). Parameter set A uses the heading information, 8 data points in the future, while the parameter set B uses the heading information, 4 points in the future.

The third parameter is relative weighting between the future and closest point heading (w_c in Equation 11 in Section 3.4). This weighted heading is constructed from 10% current headings and 90% future headings in parameter set A (w_c=0.1), and from 30% current headings 70% future headings in the parameter set B (w_c=0.3).

4.3.1 Result Using Parameters Set A

For person A, the average speed was 2.44 m/s, the average off-center distance was 0.13 m and the standard deviation was 0.18 meters. At the same time, person B had an average speed of 2.93 m/s, his average distance from the center of track was 0.1353 m, and the standard deviation was 0.16 meters.

The maximum speed for person A was 4.09 m/s and the maximum off-center distance was 0.52 meters. For person B, the maximum speed was 3.95 m/s and the maximum off-center distance was 0.55 meters.

As shown in Figure 29, for person A, 172 of the data points were on the left of the track, 210 of them were on the right of the track, and the mean error was 0.02 m. Person

B had 271 data points on the left and 114 data points on the right, and the mean error was -0.06 meter. This result indicates that the user B may have a "left tendency. "
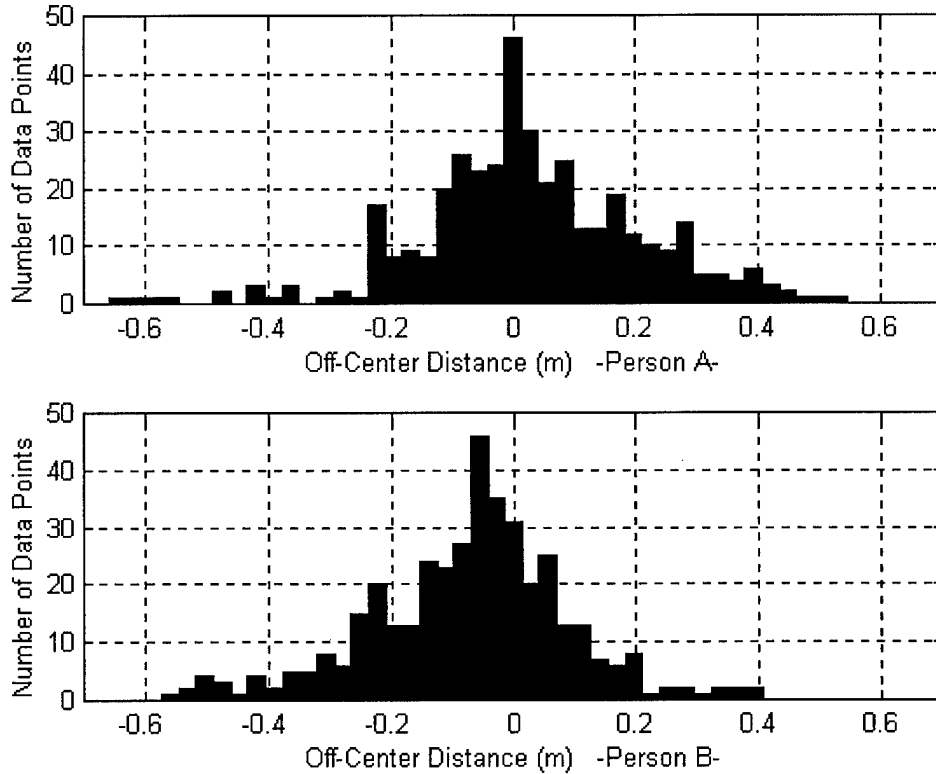
Figure 29. Position Error Histogram of Both Users for Parameter Set A

For person A; 99% of collected data points had off-center distance magnitude smaller than 0.6 m error and 50% of them had an off-center distance magnitude smaller than 0.1 meters. 23% of them were between an 0.1 and 0.2 m error, 16% of them were between an 0.2 and 0.3 m error and the other 9% were between 0.3 and 0.6m error. For person B; 100 % of collected data points had an off-center distance magnitude smaller than an 0.6 m error and 51% of collected data points were below an 0.1 meter. 25% of them were between an 0.1 and 0.2 m error, 14% of them were between 0.2 and 0.3 m error, and the other 10% were between an 0.3 and 0.6 meter error.
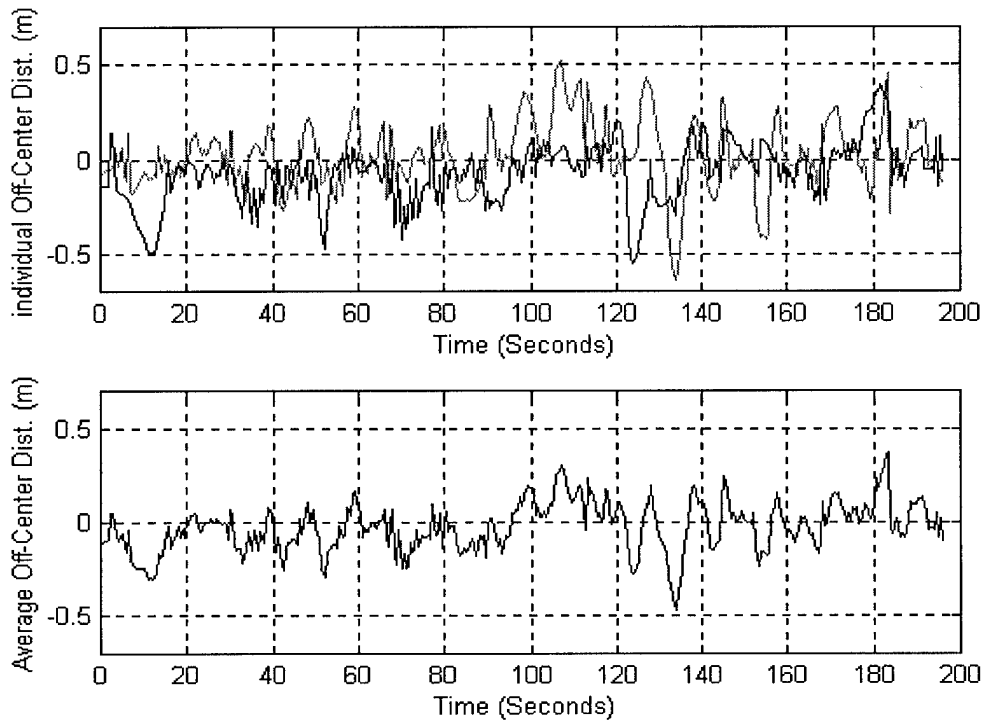
Figure 30. Position Error Plots, for Parameter Set A

Figure 30 shows the position error plots for both users. The lower subplot on Figure 30 shows the average of both users. The average speed was 2.68 m/s, the average off-center distance was 0.11 m, the standard deviation was 0.13 m, and the maximum off-center distance was 0.46 meters.

4.3.2 Results Parameter From Set B

Person A had an average speed of 2.78 m/s and average distance from the center of track was 0.17 m, and the standard deviation was 0.25 meters. Person B had average speed of 2.93 m/s and his average distance from the center of track was 0.12 m and the standard deviation was 0.19 meters. As can be seen from the numbers, person B adapted

to these parameters better than the person A. This was apparent since his velocity was higher and his average off-center distance was lower. This result shows how human factors should be a consideration for every test that has an user interface. For person A, maximum speed was 4.07 m/s and maximum off-center was 1.10 meter. For person B, maximum speed was 4.07 m/s and maximum off-center was 0.58 meter.

For person A, there were 181 data points on the left and 203 on the right, and the mean error was zero. For person B, there were 235 data points on the left and 150 on the right, and the mean error was –0.03 meters. This result indicates that is the user B may have a "left tendency" problem while driving. The position error histogram can be seen on Figure 31.
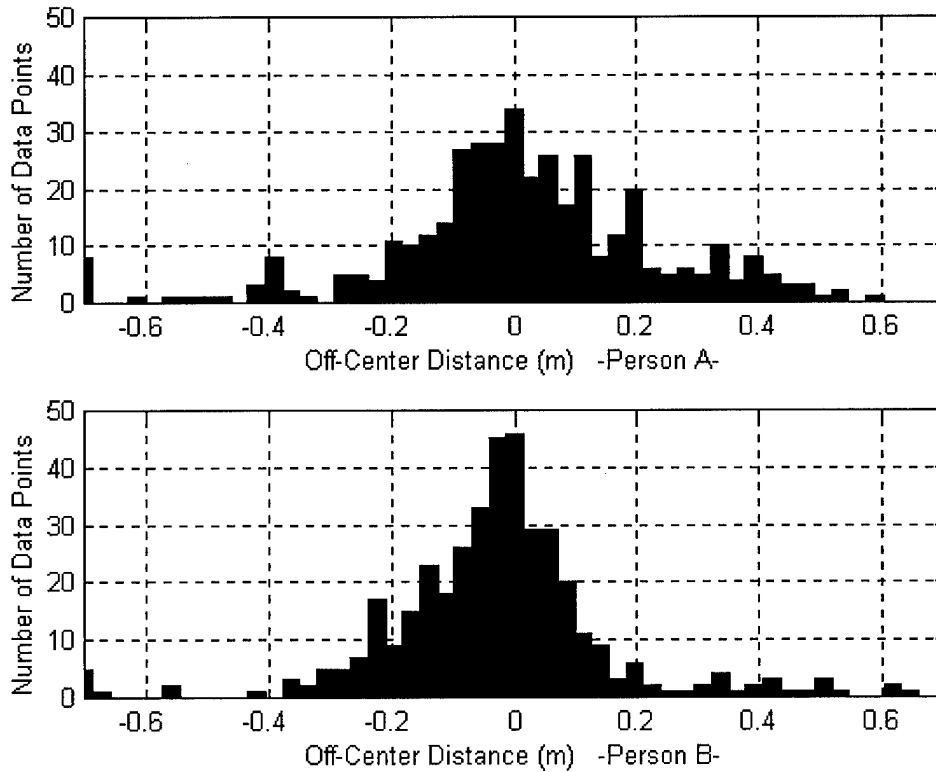
Figure 31. Position Error Histogram of Both Users for Parameter Set B

For person A; 98% of the collected data points were below a 0.6 m error and 46% of the collected data points were below 0.1 meter. 25% of them were between an 0.1 and 0.2 m error, 9% of them were between an 0.2 and 0.3 m error and the other 6 % were between an 0.3 and 0.6m. For person B; 97% of the collected data points were below an 0.6 m error and 59% of collected data points were below an 0.1 meter. 22 % of them were between 0.1 and 0.2 m error, 10 % of them were between 0.2 and 0.3 m error and the other 3 % were between 0.3 and 0.6m.

Figure 32 shows that first 100 seconds for both users have errors smaller than approximately 0.2 meters. The users had comparatively big errors between 120 and 165 seconds which is in the curved area of the trajectory.
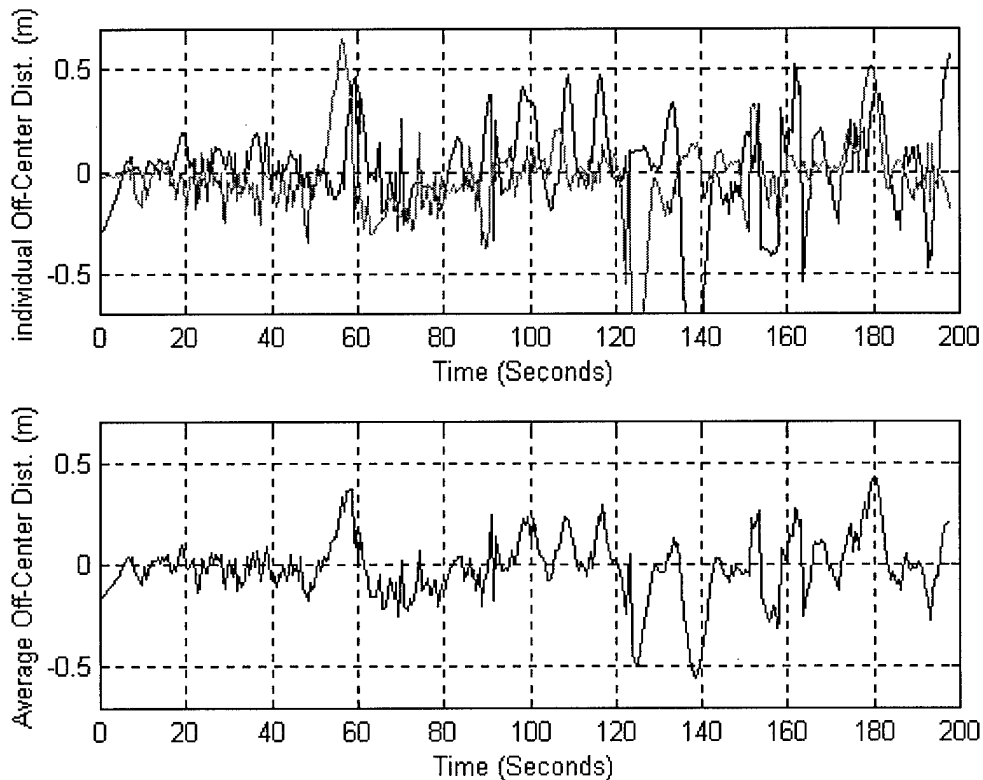
Figure 32. Position Errors Plots, for Parameter Set B

The lower subplot on Figure 32 shows the average of both users. The average speed of both users was 2.68 m/s, the average off-center distance was 0.11 m, and the standard deviation was 0.15 meters.

In the parameter Set A, user A was 22% closer to the center of the path, on the average, but his speed was also 13% lower, generally it can be assumed that he did the best on this test. For user B, he had almost the same velocity, but his the off-center value was slightly worse compared to the parameter test. The goal of the parameter tests was to find best parameter set, and the results (especially the user feedbacks) show that parameter set A is better than parameter set B. However, these results show that human factors are also important consideration.

4.4 The under hood test

All tests described up to this point were performed by users just looking at the display. Due to the computer screen and the large shade that it was placed in, the users didn't have the front view for the road, but the users still had a peripheral view. The built-in shade blocked out 140-160 degrees of the front view, and the users tried to drive based on the information on the display only. Nevertheless, it was still possible for the user to take advantage of some unintentional information outside of cover (using peripheral vision). The under hood test was created by using a black hood over the user's head that fully covering all sides of the user (except for the display). In other tests the user could not see the road but in the under hood test there was no peripheral view either. Figure 33 shows the test set up for the under hood test.

Figure 33. User Testing the Guidance System Under a Hood

The average speed of user A was 2.39 m/s, and the average off-center distance magnitude was 0.49 meter, and the standard deviation 0.66 meter. For user B, the average speed was 1.75 m/s, the average off-center distance magnitude 0.4 m, and the standard deviation was 0.75 meter. Compared to the tests described previously the off-center result is approximately three times worse, and the velocity is much lower. To see personal differences, notice that the two users were comfortable at very different speeds. As shown in Figure 34, every off-center distance had some data until 1 m and the distribution was more of a uniform distribution. Also, the percentage of data points that are greater than 0.6 m error was approximately 30%.
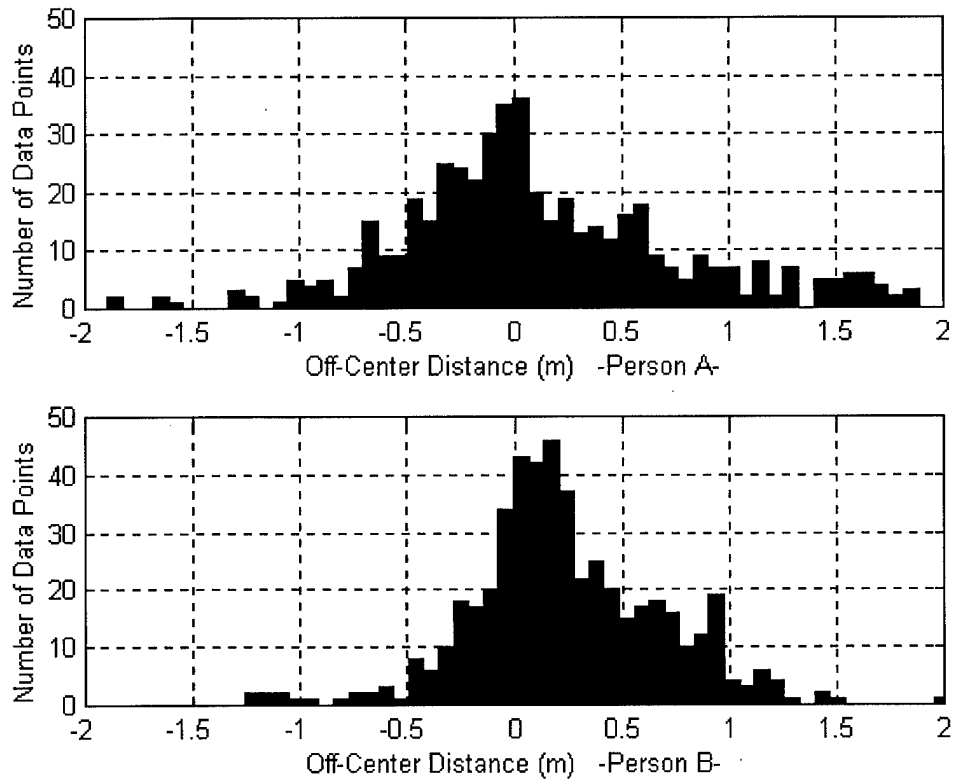
Figure 34. Position Error Histogram of Both Users for Under Hood Test

For person A; 68% of the collected data points were below a 0.6 m error and 19 % of the collected data points were below a 0.1 meter error. 12 % of the collected data points were between a 0.1 and 0.2 m error, 12% of them were between a 0.2 and 0.3 m error and 10% were between a 0.3 and 0.4m. 25% of the collected data points were between 0.3 and 0.6 meter. For person B; The 77 % of the collected data points were below a 0.6 m error, and the 22 % of collected data points were below a 0.1 meter. 18 % of the collected data points were between a 0.1 and 0.2 m error, 14 % of them were between a 0.2 and 0.3 m error, and 22% of the collected data points are between a 0.3 and 0.6 meter These numbers show a distribution that was very different from the other tests. Note that for the

other tests, the percentage of data points under a 0.6 m error was normally between 90%

and 100% of collected data points.

In Figure 35, when person B came to the center of track, he would tend to stay on the

center for some period of time, but person A almost never was able to stay in the center

of track.

Person A had total 494 data points that were collected from this test. Of these, 239

were collected on the left of the track and 248 were collected on the right and the mean

error was zero. Where as person B had 354 data points collected on the right track, and

131 data points collected on the left, and the mean error was 0.26 meters.
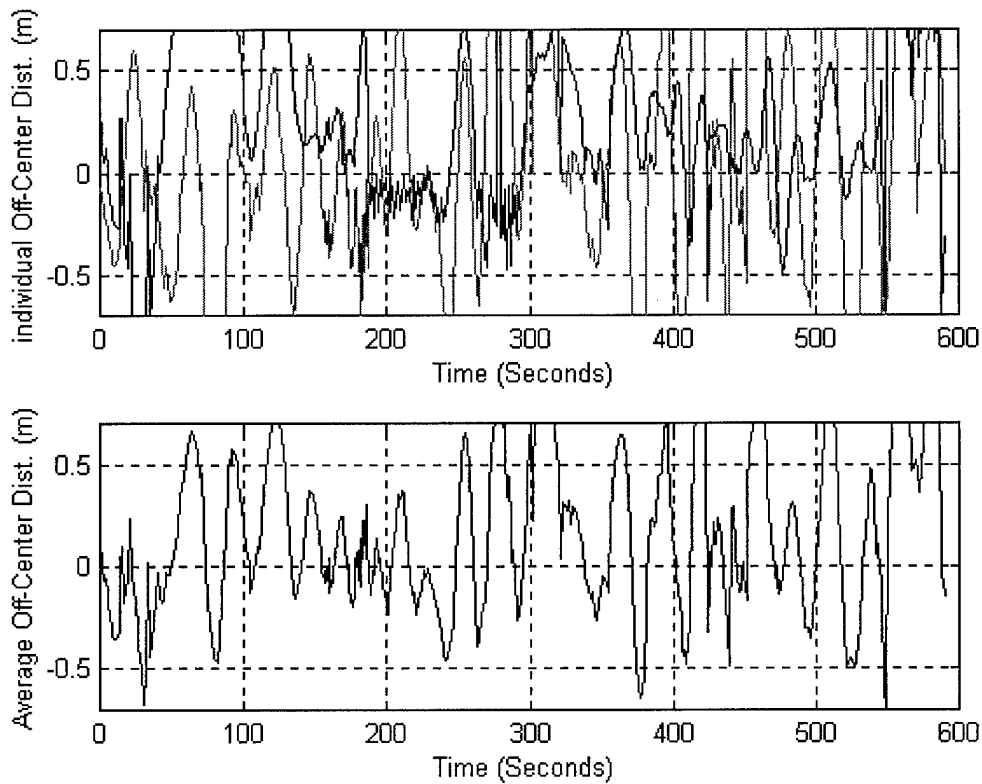
Figure 35. Position Errors Plots, Under Hood Test

The lower subplot on Figure 35 shows the average of both users. The average speed was 2.08 m/s, the average off-center distance magnitude was 0.35 m, and the standard deviation was 0.53 meters.

Notice that for under hood test, everything should be smooth (including the command force to apply to the wheel), but under the test environment this was not true. The second consideration is that an under hood test requires more experience than other tests. The human being is used to driving a car based on real views, and in the other tests a peripheral view may have provided a more familiar "feel" for this requirement. This is very natural, and with practice, it would be expected that the results similar to the other tests could be obtained in the under hood test. User B stated on the standard test evaluation form that once he was off course, it was difficult to get back on course.

4.5 Summary of Test Evaluations (other than Update Rate Tests)

The results show that there are some errors that almost all of the users made at the same place of the track, and there are some errors that the users individually made. Because of the nature of driving on the curve part of the track, the off-center distance was bigger there for every user. As previously described, the error nature of driving on a curved road and the heading approximation inaccuracy contribute to increased errors on the turn for all of the tests.

The single instrument test generally shows good positioning results, but the users felt that it did not give enough confidence for the curved road. They believe there should be additional information provided to make the user more aware of the whole situation.

As mentioned before, many parameters can be tested, but between the two sets tested, parameter set A is better. The parameter Set A used 9 as a off-center distance

multiplication number for the graphical display (set B used 11), 8 for the future reference point (set B used 4), and 90% weight on the future heading (set B used 70 %). The bottom line is that it is important to give information about the future to the user for good, confident performance.

In the under hood test, the off-center distance values of both users were 0.49 m and 0.40 m , whereas similar values on other tests (without hoods) were under 0.2 meters. Notice that in every test the user's goal was to use only guidance display to drive. This implied that the peripheral view is important and should be considered during driving tests. The distribution of the under hood test, had a much wider distribution then the previous tests, but it maintained a normal distribution shape. The Table 3 shows the summary of users answers for the update rate tests.

Table 3.Summary of Answers for Key Tests Evaluation Questions

User	Single Instrument Test		Parameter Set A		Parameter Set B		Under hood Test	
	A	B	A	B	A	B	A	B
The software meets the objective	9	10	10	10	8	9	8	8
The software is easy to use	10	10	10	10	6	9	7	7
The instrument is well organized	9	10	10	10	8	7	9	8
The instruments are adequate to use the car	9	8	10	10	8	7	8	7
The interface matches the real-time Situation	10	10	10	10	8	8	9	7
The overall experience rate	9	9	10	10	8	9	8	8

4.6 Summary

The update rate test, single instrument test, under hood test, and different parameters tests and thesis results were described in this chapter. The single instrument test didn't give the required guidance and confidence especially during turns. The slow and high update rate problems and results were investigated. The different parameter sets were tested to find better parameters. The under hood results were significant, because they showed the importance of peripheral view. Finally, perhaps most importantly, the impact to results indicated that human factor consideration must be accounted for in any user interface graphical.

5. Conclusions & Recommendations

5.1 Conclusions

This thesis presented a real-time guidance display including the algorithms that were designed for this research. The software written for this research was able to take the data from the GPS receiver and output guidance information to a user via a graphical interface.

Four tests were used to evaluate the guidance system performance and human factors under different circumstances. The tests were an update rate test, single instrument test, parameter set test, and the under hood test.

The guidance system was explored under update rates of 1, 2, 3.3 and 5 Hertz. The feedback from the user indicated that the 3.3 Hertz update rate was the most comfortable update rate for the high dynamics for the golf car. The consensus of users for the single instrument test was that a single instrument was not sufficient, especially in the turn area, where the map becomes an especially necessary instrument.

In the under hood test the off-center distance values were much higher than other tests. Notice that, in every test, the user's goal was to use only guidance display. This implies that the peripheral view is important and should be taken into consideration. The distribution of the under hood test data was much wider than the other tests, with position errors approximately two times worse than without the hood.

This system was able to consistently output the guidance information for the running path (that on average has a width under 2 meters). Airborne application flight-path widths

can be bigger by at least an the order of 100, depending upon the application. Thus, there is no obvious reason why the algorithm developed in this thesis shouldn't be applicable to more dynamic applications, with proper modifications to account for the different dynamics experienced by aircraft.

5.2 Recommendations

The tests should be continued with different parameter sets, and with larger number of people so more accurate system specifications could be created.

The guidance system's display should be modified to make a more effective and informative display. This new display should be tested by various users to investigate the performance of new display. This process should continue until the users are fully satisfied.

The laptop computer can be exchanged with a faster computer to decrease the time to process the GPS data coming from the receiver. This will allow it to work with higher update rates. There is a delay due to the time difference between getting the GPS data from the receiver -the time to display, the results to the user, and this delay is approximately 0.10-0.15 seconds. A faster computer will help to reduce the delay time, but it will not be zero. A new algorithm (similar to a Kalman filter) could be developed and this algorithm could estimate the real-time position for a short delay time. This would ensure that the user would not see the delay.

The guidance system worked based on constant assumption and did not take into account different velocities. The algorithm should be modified to take into account the velocity when determining the guidance information.

The presented algorithms that were created for this research should be a starting point to develop an automated vehicle. The reliability of the guidance system should be satisfactory unless the vehicle does not require a very high reliability. For the high reliability case, a single degree freedom gyroscope integration should be considered especially to make heading information more accurate.

Appendix A. The OpenGL commands

glRect (x1,y1,x2,y2): Draw the rectangle defined by corner points (x1,y1) and (x2,y2).

glTranslate (x, y, z): Multiplies the current matrix by a matrix that moves (translates) an object by given x, y, z values.

glScale (x, y, z): Multiplies the current matrix by a matrix that stretch, shrink, or reflect a object along the axes. Each x, y and z coordinate of every point in the object is multiplied by corresponding argument x, y, or z

glLookAt (eye x, eye y, eye z, center x, center y, center z, up x, up y, up z): The desired viewpoint is specified by eye x, eye y and eye z. The line of sight, typically there are some point in the center of the scene being look at.

glStencil: Stenciling applies a test that compares a reference value with the value stored at a pixel in stencil buffer. Depending of the result of the test, the value in stencil buffer is displayed or not.

glutMouseFunc(func): Specify the function that's called when a mouse button is pressed or released.

glutSolidCone (radius, height, slices, stacks): Based on given information it will draw a cone.

glutIdleFunc (function): Specifies the function to be executed if no other events are pending.

Sprintf: Returns the number of bytes stored in *buffer,* not counting the terminating null character. The function formats and stores a series of characters and values in *buffer.* Each *argument* (if any) is converted and output according to the corresponding format

specification in *format*. The format consists of ordinary characters and has the same form

and function as the *format* argument for *printf*.

Sscanf: It is a general purpose function that can read all the built-in data types and

automatically convert them into proper internal format.

Appendix B. The Standard Evaluation of Any Test

- Note this is the exact handout given to the volunteers who tested the system.

THE REAL-TIME DGPS KINEMATIC GUIDANCE SYSTEM

The system that you will use is Real Time DGPS implementation and guidance system which has *centimeter level* accuracy. This is the first implementation of the real-time guidance system based on GPS in AFIT.
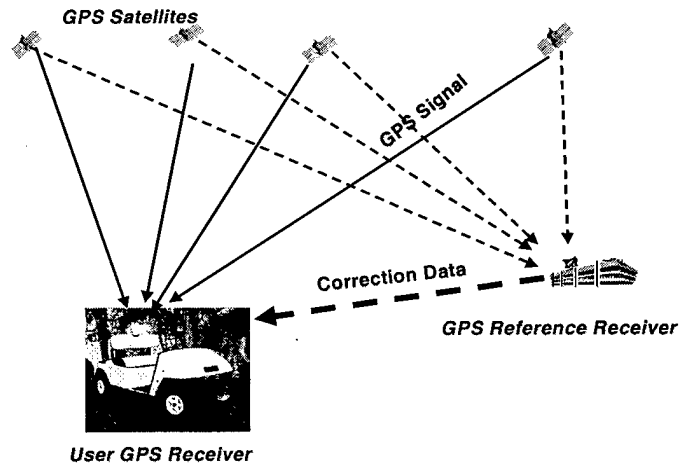
Figure 36. DGPS Concept for Pre-test Information

The DoD purposely limits the accuracy of GPS position accuracies through the use of Selective Availability (SA), which introduce an intentional error into satellite clock ephemeris parameters. SA is not the only error that can degrade the accuracy, the receiver clock error, trophosheric error and others. However, SA is the dominant error for civil users.

The DGPS concept is to use one or more base station, that have precisely known position. The base station calculate its position the same as stand-alone receiver and since

it know where position is precisely, the receiver can derive the correction to all user in the coverage area and send them to the receivers that can easily apply corrections to the calculated position. By using DGPS , many of GPS errors (such as ionospheric and trophosperic delays, satellite ephemeris errors, and clock errors) significantly reduced or eliminated.

VISUAL INTERFACE

The Map : It is intended to show where the user is on the area, and then if zoomed it shows the information in more detail. It is believed that it increases SA (situational awareness)

Figure 37 Guidance Display for Pre-test Information.

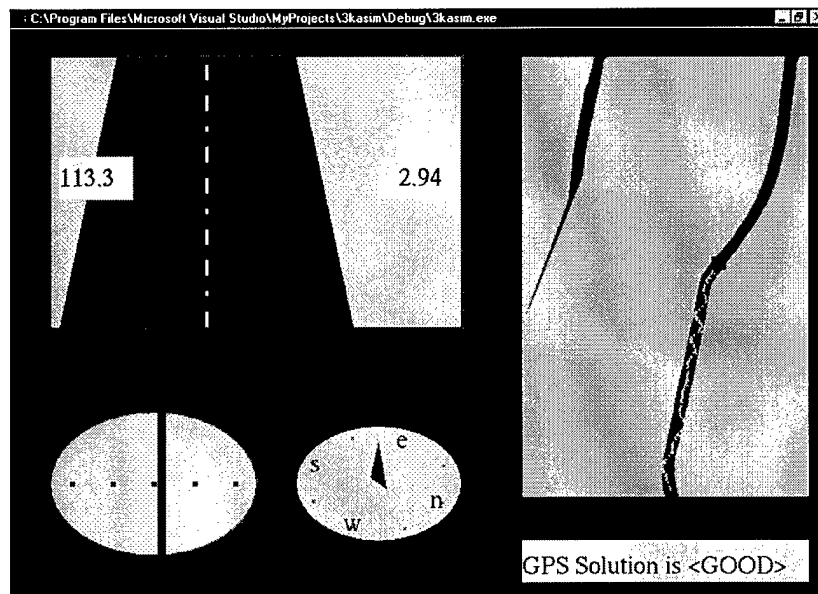The Solution Indicator: It can be "GPS solution is good" in blue or "GPS Solution Is Not Good" in red.

The Road: The user is assumed to be on the red triangle at the bottom of the road. The upper part shows how the road looks like from the user's point of view at the moment. The left little window of the road display shows heading and the right little window of road display shows velocity. Based on experience, the average speed for golf car is 2 m/s.

Other Indicators : The needle is showing you what the reaction should be. The concept is very easy: " put the needle on the center". The heading indicator is intended to increase situational awareness especially for users who find it difficult to read from numbers.

THE REAL-TIME DGPS KINEMATIC SYSTEM EVALUTION

Thanks for trying the guidance system. Please evaluate the system in one to ten scale and don't hesitate to add any commands that you have.

- The software meets the objective of allowing the user to stay on the course

  a) 1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

- The software is easy to use.

  a) 1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

- The instrument is well organized

  a) 1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

- The instruments are adequate to use the car.

  a) 1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9)10

- Are there any other instruments that can help and should be on the screen? If so, please describe it. (At the back of this paper)

- Background colors for map are appropriate

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ (if you use zooming) The zooming is useful.

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ To show "GPS solution is good " is something necessary (because I want to

know if DGPS is working or not

a) 1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ For the road display, the figure match the real time situation

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ I like the way the road moves.

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ The road color (on the map and on the left upper display) couldn't be better it

resembles a real road.!!! ☺

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ The circle with line moving left is called "course indicator". The course indicator

is designed to help to solve especially left and right problems and to complete the big

picture. Does it reach the goal?

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ How do you rate the overall experience?

a)1 b)2 c)3 d)4 e)5 f)6 g)7 h)8 i)9 j)10

◆ if you have any comments, please describe on the back of the paper or be quit to

the end of your life.!!! ☺

Appendix C. Acronym List

AFIT	Air Force Institute of Technology
DGPS	Differential Global Positioning System
ECEF	Earth-Centered, Earth-Fixed
GPS	Global Positioning System
INS	Inertial Navigation System
NRS	Navigation Reference System
PPS	Precise Positioning Service
PR	Pseudorange
RMS	Root-Mean-Square
SA	Selective Availability
SPS	Standard Positioning Service
SV	Satellite Vehicle

Bibliography

[1] Per Enge and Pratap Misra. *Scanning the Issue/Technology, Special Issue on Global Positioning System*, Proceedings of The IEEE,January, 1999 Volume 87 .

[2] Z-Surveyor Operation and Reference Manual, and Technical Reference Manual. Magellan Corporation-1998

[3] *Data transceiver (DGR-115)*, FreeWave Technology Inc.,www.freewave.com.

[4] *The golf car (RAVEN)*. Club Car Inc, www.clubcar.com.

[5] Bradford W. Pakinson and James J. Spilker Jr. , *Global Positioning System: Theory and Applications*, Volume 1 and Volume 2.,1996

[6] Donalt R. Gentner and Jonattan Grudin. *Designing Models for Computer-Human Interfaces*, Computer Vol 29, June 1996.

[7] Mason Woo, Jackie Neider and Tom Davis. *OpenGL Programming Guide Second Edition*, Silicon Graphics-1997.

[8] J. F. Raquet, *EENG 533 Navigation Using the Global Positioning System Class Notes*-AFIT, Spring 1999.

[9] Elliot D. Kaplan. *Understanding GPS: Principles and Applications*, Artech House Inc-1996.

[10] Herbert Schildt. *C++ The Complete Reference Third Edition*, McGraw-Hill-1998

[11] Dvane Hanselman and Bruce Littlefield. *Matlab User Guide Version 5*, The MathWorks-1997.

[12] J. F. Raquet, *EENG 633 Advanced GPS Theory and Application Class Notes*, AFIT, Summer, 1999.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2000	Master's Thesis

4. TITLE AND SUBTITLE
DEVELOPMENT OF A REAL TIME GUIDANCE SYSTEM FOR A KINEMATIC TEST VEHICLE

5. FUNDING NUMBERS

6. AUTHOR(S)
Orhan Cekic, 1Lt, Turkish Air Force

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P Street
Wright-Patterson AFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/00M-21

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Juan Vasquez,Major,USAF
AFRL/SNAR Building 620
2241 Avionics Circle
WPAFB, OH, 45433-7321

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Capt John F. Raquet, Thesis Advisor, (937) 255-3636 x4580, john.raquet@afit.af.mil
1Lt Orhan Cekic,Hava Kuvvetleri Komutanligi, Personel Bsk., Ankara/Turkey

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

    Differential carrier phase GPS measurements are capable of giving centimeter-level accuracies. These accuracies have many potential applications for safety and control of various types of vehicles.

    For this research, a real-time guidance system is developed. The real-time guidance system can be divided into two components: hardware and software. The hardware component consists of two GPS receivers (one base, one mobile), two wireless 115 Kbaud transceivers, and two laptop computers. One computer is for the reference station and other is for the mobile receiver interface and graphical display. The guided vehicle is a golf car called the Remote Sensing Autonomous Vehicle for EN (RAVEN).

    The research concentrated on developing real-time data processing algorithms and using these algorithms to show guidance information to the user via a graphical interface. The developed software reads the real-time GPS data using an RS-232 interface and converts it to a usable form for the data processing algorithms. The data processing algorithm compares the real-time data with the desired track and outputs the guidance information to the guidance display. Based on the information from the guidance display, the user is able to drive on the desired track.

    Four tests were performed to evaluate the guidance system performance and human factors under different circumstances. These tests include update rate tests, a single instrument test, varying parameter tests, and an under hood test. Tests results and user feedback show that the system performs well under most conditions.

14. SUBJECT TERMS	15. NUMBER OF PAGES
GPS, Global Positioning System, Differential GPS, Carrier-Phase, Real Time Guidance, OpenGL, Graphical Interface, RAVEN, Kinematic, GPS Receiver	94

16. PRICE CODE

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

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements.*

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with....; Trans. of....; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	-	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	-	See authorities.
NASA	-	See Handbook NHB 2200.2.
NTIS	-	Leave blank.

Block 12b. Distribution Code.

DOD	-	Leave blank.
DOE	-	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	-	Leave blank.
NTIS	-	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.