Global Positioning System Navigation and Digital Communications for Cooperating Unmanned Air Vehicles

by

Richard Poutrel

Submitted to the Department of Aeronautics and Astronautics in
partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author ....
Department of Aeronautics and Astronautics
May 24, 2002

Certified by ....
John J. Deyst
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
Wallace E. Vander Velde
Chairman, Committee of Graduate Studies

Global Positioning System Navigation and Digital Communications for Cooperating Unmanned Air Vehicles

by

Richard Poutrel

Submitted to the Department of Aeronautics and Astronautics on
May 24, 2002, in partial fulfillment of the requirements for the
degree of Master of Science in Aeronautics and Astronautics

Abstract

The Parent and Child Unmanned Aerial Vehicle (PCUAV) is the second project to originate from the MIT / Draper Technology Development Partnership, which aims to develop first-of-a-kind systems. The PCUAV proposes a low-cost solution to long distance close up observation, using the cooperative action of a fleet of small UAVs. The project team will prove the potential of the system by demonstrating key enablers such as autonomous air Rendezvous.

This thesis presents the key technologies used to perform the Rendezvous of two autonomous UAVs. The Global Positioning System (GPS) and the way its information is used to provide the navigation, guidance and control system with accurate information is first introduced. Communication induced by the different ways Rendezvous is performed is then emphasized. The overall system used for navigation is eventually discussed.

Thesis Supervisor: John J. Deyst
Title: Professor of Aeronautics and Astronautics

Table of Content

List of Figures

List of Tables

Acknowledgments

First I would like to thank the C.S. Draper Laboratory, sponsor of the project, for giving me and other students in the project the opportunity to participate in such a stimulating learning experience. Special thanks to Dr. Brent Appleby and Mr. Sean George for their advice and contributions to the project.

I would like to thank Professor John Deyst for giving me the opportunity to study at MIT and take part in this great project. During these years at MIT, I have met people from all around the world, people with different personalities and perspectives, people with great talent and knowledge. I also would like to thank Jonathan How and his students Nicholas Pohlman and Chanwoo Park for their inestimable help.

I would like to thank all the team members of the PCUAV project. More precisely Francois Urbain, Sanghuyk Park, Thomas Jones, Jason Kepler, Sarah Saleh, Damien Jourdan, Carmen Carreras and Alexander Omelchenko. My life in Boston would certainly not have been the same without my roommates Thien-loc, Flo and Peter for all the fun we had. I also would like to thank my friends for all the crazy days and nights we had together: Joan, Ludo, Craig Cloutier, Rahul, Craig Runyan, Jeni, Maya, Pippa, Tera, Athicha, Markus, Nico, Sandiph, Adrien, Mirna, Brice, Simon ... I also would like to thank my friends from France who visited me, and who have never forgotten me.

This adventure would not have been possible without the help of my family. Cocole, for letting people from all the world share our life, and for giving me the opportunity to travel. Robert, Lola and Francois for not only visiting me, but also giving me support.

Anne, life in Boston would not have been the same without your visits and your everyday support. Everywhere you go, you bring joy, happiness, and love.

List of Acronyms

0x##, ##h	Refers to hexadecimal number (0x01, 01h, 0x02, 02h, ...)
$\mathbf{1}_i$	Line if Sight from a user's GPS receiver's Antenna to a Satellite i
a	a=6378.137 km: Equatorial Cross-Section of the Earth
ATA	Avionics Test-bed Aircraft
b	b=6356.7523142 km: Polar Cross-Section of the Earth
b_u	Unknown Time GPS Receiver Offset Relatively to the GPS System Time
c	Speed of Light in Vacuum
C/A-code	Coarse/Acquisition (or Clear/Access) Code
CDGPS	Carrier Phase Differential Global Positioning System
CG	Center of Gravity
CIC	Computer In Control
CMC	Canadian Marconi Company
CmplID	Complementary Identification Number (ID)
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DB9	9-pin-D-Type Connector
Δf_{Di}	Doppler Shift due to the Relative Motion of a Satellite i and a GPS Receiver
Δf_i	Observed Carrier Frequency Shift between a GPS Receiver and a Satellite i
$\Delta\Phi$	Differential Carrier Phase
DGPS	Differential Global Positioning System
DLAB	Divisor Latch Access Bit
DOP	Dilution of Precision
$\mathbf{dX}_{MP}$	Difference in Position between the Mini's Raw GPS position and the Parent's Raw GPS Position in the Local Frame in Local Frame Coordinates
δ_i	Pseudo-Range Rate or Delta-Range
δ_u	Receiver Clock Frequency Bias
e	Eccentricity
E	East
ECEF	Earth Centered Earth Fixed Coordinates
ε_i	Error in the Observation of the Pseudo-Range
EIA	Electronic Industries Association
FCC	Federal Communications Commission
FIFO	First In / First Out
FHSS	Frequency Hopping Spread Spectrum
f_o	GPS Fundamental Frequency: f_o=10.23 MHz
f_u	Unknown GPS Receiver Frequency Bias
ϕ	Geodetic Latitude
Φ	Carrier Phase
GPS	Global Positioning System
h	Geodetic Height
HDOP	Horizontal Dilution of Precision
HOW	Hand-Over Word
ID	Identification Number

λ	Geodetic Longitude
η_i	Errors in the Frequency Shift Observation
I/O	Input/Output
IRQ	Interrupt Request
L1	The 1575.42 MHz GPS Carrier Phase Frequency
L2	The 1227.60 MHz GPS Carrier Phase Frequency
LED	Light-Emitting Diode
LNA	Low Noise Amplifier
LSB	Least Significant Bit
LZW	Lempel-Zif-Welsh
μ	WGS-84 value of the Earth's Universal Gravitational Parameter: $3.986005.10^{14}$
MAV	Micro Autonomous Vehicle
MPIM	Mini-Parent Integration Mechanism
MSB	Most Significant Bit
N	North
$(\mathbf{N},\mathbf{E},\mathbf{H})$	Axis Vectors of the Local Frame in ECEF Coordinates
(N_l,E_l,H_l)	Position of the User in Local Frame Coordinates
NMEA	National Marine Electronics Association
OHS	Outboard Horizontal Stabilizer
Π,Π_i	Code Phase
P-code	Precision (or Protected) Code
PCUAV	Parent and Child Unmanned Air Vehicle
PDV	Payload Delivery Vehicle
P/N	Part Number
PPS	Pulse Per Second
PRN	Pseudorandom Noise
ρ	Pseudo-Range
ρ_i	Pseudo-Range between a User and a Satellite i
RF	Radio Frequency
R/C RC	Remote Control
$\mathbf{r}_u$	User's GPS Position at Time of Reception of a GPS Signal
$\mathbf{r}_i$	Satellites' Position at the Signal Transmit Time
RS232	Recommended Standard 232 C: refers to EIA/TIA-232-E
Rx, RX	Reception
S	South
S/A SA	Selective Availability
SNR	Signal to Noise Ratio
SOH	Start of Header
SV	Satellite in View
t	Time
τi, $\tau 2$, $\tau 1$	Time Constant (where i is a number: 0 1 2 ...)
TDOP	Time Dilution of Precision
TIA	Telecommunications Industry Association
TLM	Telemetry Word
t_{pd}	Observed Propagation Delay

TOW	Time Of Week
TTL	Transistor-Transistor Logic
Tx, TX	Transmission
$\mathbf{u}=(x_u,y_u,z_u)$	Position Vector and Coordinates of a User in the ECEF Frame
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Air Vehicle
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision
Vr	Reintegration Velocity
VSWR	Voltage Standing Wave Ratio
$\mathbf{v}_u$	User's GPS Velocity at Time of Reception of a GPS Signal
$\mathbf{v}_i$	GPS Satellites' Velocity at the Signal Transmit Time
W	West
WAAS	Wide Area Augmentation System
Ωe	WGS-84 value of the Earth's rotation rate: $7.2921151467.10^{-5}$
WGS-84	World Geodetic System 1984
(X_u,Y_u,Z_u)	Position of a user in the ECEF Coordinates
(X_l,Y_l,Z_l)	Position of the center of the Local Frame in the ECEF Frame
$\mathbf{X}_{FM}$	Filtered Position of the Mini in the Local Frame in Local Frame Coordinates
$\mathbf{X}_{FP}$	Filtered Position of the Parent in the Local Frame in Local Frame Coordinates
$\mathbf{X}_{M}$	GPS Raw Position of the Mini in the Local Frame in Local Frame Coordinates
$\mathbf{X}_{P}$	GPS Raw Position of the Parent in the Local Frame in Local Frame Coordinates
ζi	Error in the Observation of Pseudo-Range Rate

Chapter 1

Introduction

1.1 Background and Motivations

The Parent and Child Unmanned Air Vehicle (PCUAV) is a research project at MIT funded by the Charles Stark Draper Laboratory. The PCUAV project has been spread over the past four years, from Fall 1998 to June 2002.

During the first year, the students evaluated market needs and assessed the national armed force interests and came up with the concept of "up-close aerial surveillance from long distance at low altitude at low cost". The team chose to use a fleet of UAVs of different sizes that operate cooperatively. A Parent vehicle would carry Mini and Micro vehicles over a distance of up to 300 km, would deploy them, and would serve as a relay for communication between the vehicles and a ground station. When the mission is over, the Mini Vehicles would reintegrate with the Parent that would then fly back home.

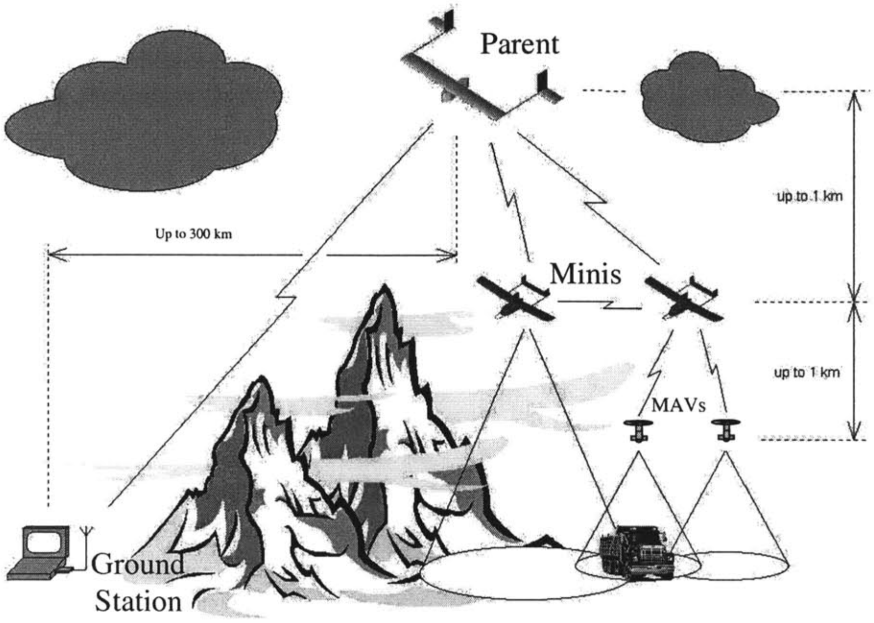


Figure 1.1 Distance Surveillance Using Multiple UAVs

The system has the following characteristics:

- The Parent leaves the base carrying Mini (Children) and Micro vehicles and flies autonomously to the mission site.
- The Parent serves as a relay for the fleet communication with the base station.
- The system supports itself to achieve sustained surveillance.
- Once the mission is completed, the Mini vehicles (Children) reintegrate with the Parent that carries them back home.

The second year of the project was spent designing and building the vehicles and the guidance, navigation and control system for the autonomous flight of a Mini vehicle. The author joined the project at the beginning of year three during which the team decided to focus on the demonstration of autonomous reintegration.

1.2 Thesis Overview

The objectives of this thesis are to describe the development of key technologies necessary to perform reintegration of two autonomous vehicles and to summarize the author's work within the project. Since PCUAV is a team project, this thesis will often refer to work done by other team members. The author's main contribution has been to design and develop a communication system between the vehicles and to provide the vehicles with accurate GPS information.

Chapter 2 describes the reintegration concept as part of the PCUAV project. It focuses on how Rendezvous is performed between two autonomous vehicles and introduces requirements on communication and position accuracy.

Chapter 3 describes the way GPS information is used and provides results of ground and flight tests.

Chapter 4 focuses on the different aspects of communication involved between different UAVs and a ground station.

Chapter 5 summarizes the overall architecture of the system and introduces some ideas for future work.

Chapter 2

PCUAV System

2.1 Chapter Overview

This chapter presents a top-level discussion of the PCUAV project. First, the PCUAV concept, requirements, and missions are presented. Then reintegration is emphasized as the key enabler of the system. Finally, the requirements for positioning the vehicles and for communication are discussed.

2.2 The Parent Child Unmanned Air Vehicle Project

The PCUAV concept was built in order to "perform real-time and continuous up-close surveillance, from a long distance, of a low altitude cluttered environment, using low-cost assets".

2.2.1 The PCUAV Concept

The proposed concept is a fleet of UAVs organized in a three-tiered structure with:

- Small vehicles (Micros) which can operate close to the point of interest and:
  - can perform up-close surveillance
  - can deploy ground robots
  - are expandable.

- A large vehicle (Parent) which:
    - can observe from a higher altitude
    - can transport and deploy Mini and Micro vehicles
    - can enable communication between the fleet and the ground station
- Mid sized vehicles (Minis) which:
    - provide the intermediate communication and deployment link between the Parent and Micros.
    - can observe from lower altitudes
    - can be used for sampling and retrieval

The baseline concept is a Parent vehicle that can carry two Mini vehicles and four Micro Autonomous Vehicles (MAVs) or a payload of sensors for delivery to the ground.

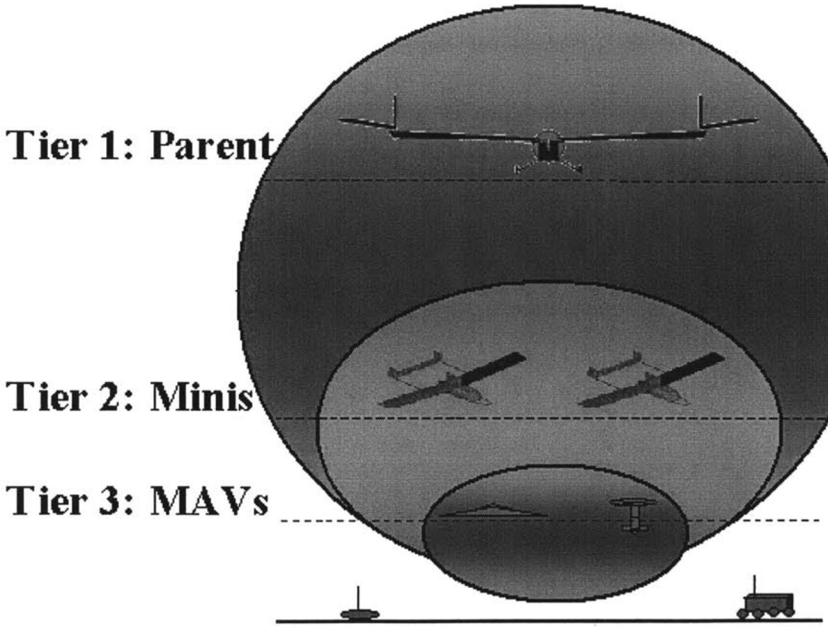


Figure 2.1 Three-tiered PCUAV Concept

2.2.2 The Importance of Reintegration

The requirements for this system have been discussed in the master's theses of Sanghyuk Park [15] and Francois Urbain [6]. They identified a key technology, which had never before been demonstrated, as the reintegration of the Mini vehicles with the Parent. Therefore, the team focused on that aspect of the project and named this phase Rendezvous. With Rendezvous, it is possible:

- to reintegrate the Minis with the Parent so that:
    - they can be reused for further missions
    - much data can be saved onboard without being transmitted and can then be analyzed at the base (information needed for real-time operations can be transmitted to the ground station via the Parent if needed).
- to refuel them:
    - to extend the Minis' endurance and the time of deployment of the fleet
    - to enable continuous surveillance (as long as there are at least two Minis and replacement of Parent when its resources are depleted)

2.3 The Parent-Child Rendezvous

Different approaches of the Rendezvous between two vehicles have been discussed ([15], [6]) involving either one or two Minis. In order to demonstrate reintegration, and due to the difficulty to accomplish it, the team decided to demonstrate it with the Parent and one single Mini vehicle. The chosen configuration is presented in Figure 2.2.

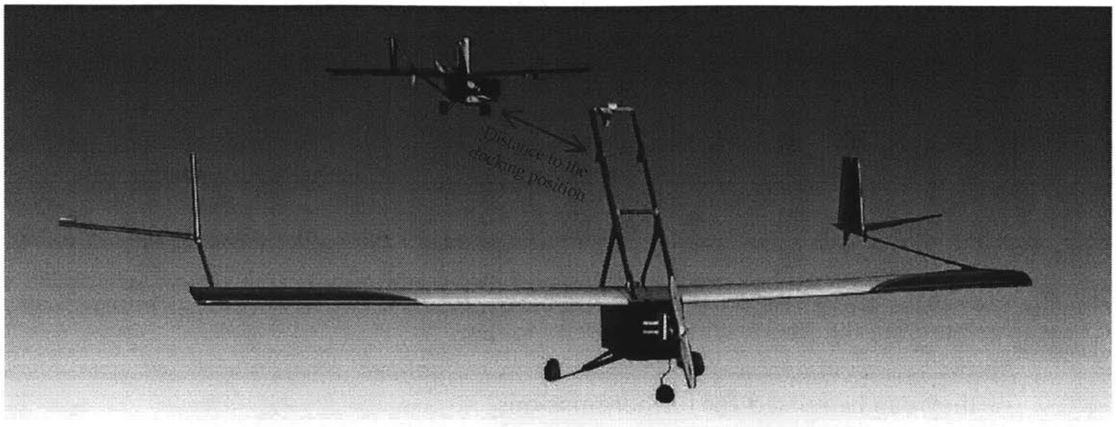


Figure 2.2 Demonstration Parent/Mini Reintegration Concept

2.3.1 Aerodynamics Concerns

The Mini vehicle (Figure 2.3) has been built with the following considerations in mind:

- a pusher configuration in order to enable reintegration with the Parent

- a direct side force and lift control surfaces to simplify the control problem by allowing the airplane to translate vertically and horizontally without performing pitch and roll rotations.

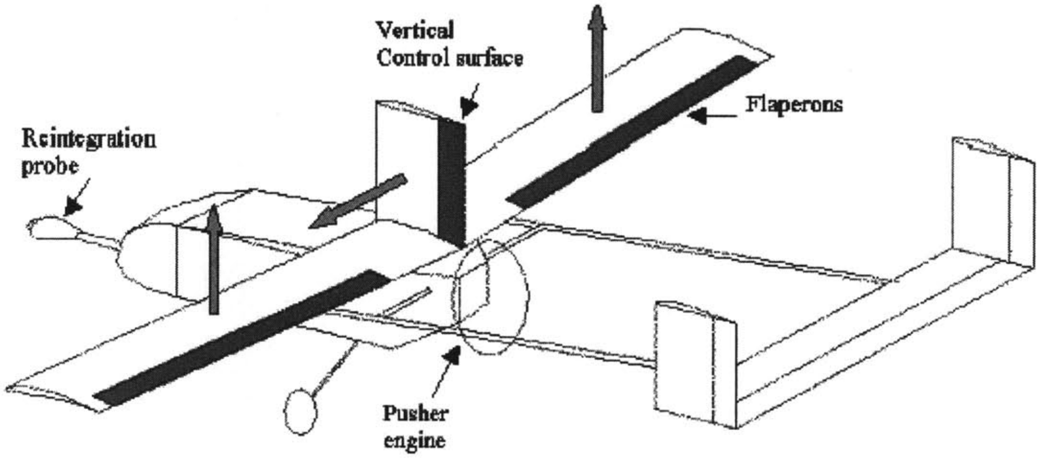


Figure 2.3 Mini Direct Lift and Side Force Controls

The Parent vehicle has an Outboard Horizontal Stabilizer (OHS) configuration (Figure 2.4). In this configuration, largely described by Francois Urbain [6] and Jason Kepler [8], the horizontal tails of the OHS aircraft are placed outboard of the main wing so that the trailing vortices from the wings have an upwash effect on the horizontal tail, which provides more efficiency and stability. In addition, the tail location provides a large clear region behind the fuselage for reintegration with the Mini vehicle.

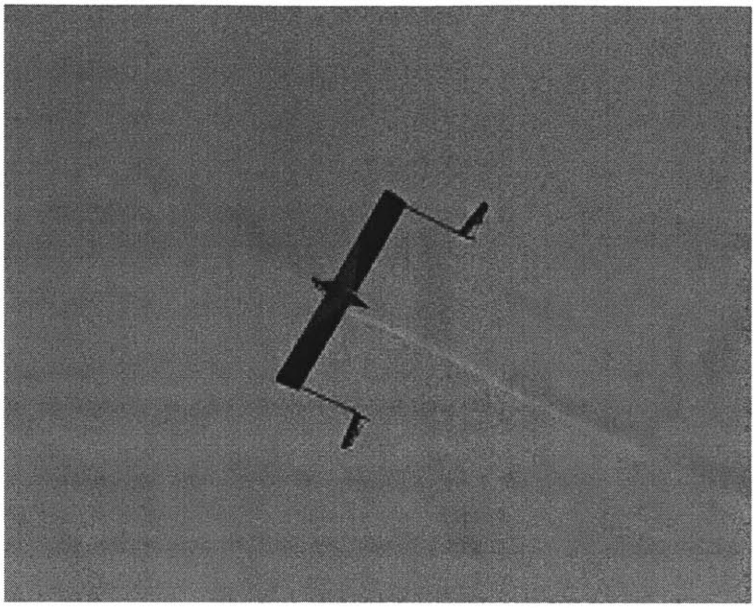


Figure 2.4 Parent OHS during a Flight Test

One last important aerodynamic aspect of this configuration is that it gives the Parent the availability to fly with or without the Mini integrated as well as with or without the Mini-Parent Integration Mechanism (MPIM) [6] developed by Carmen Carreras. Although the configuration shown in Figure 2.5 is expressly designed as a prototype demonstration, it is envisioned that an operational vehicle would position the vehicle directly behind the fuselage of the Parent to minimize the total system drag.

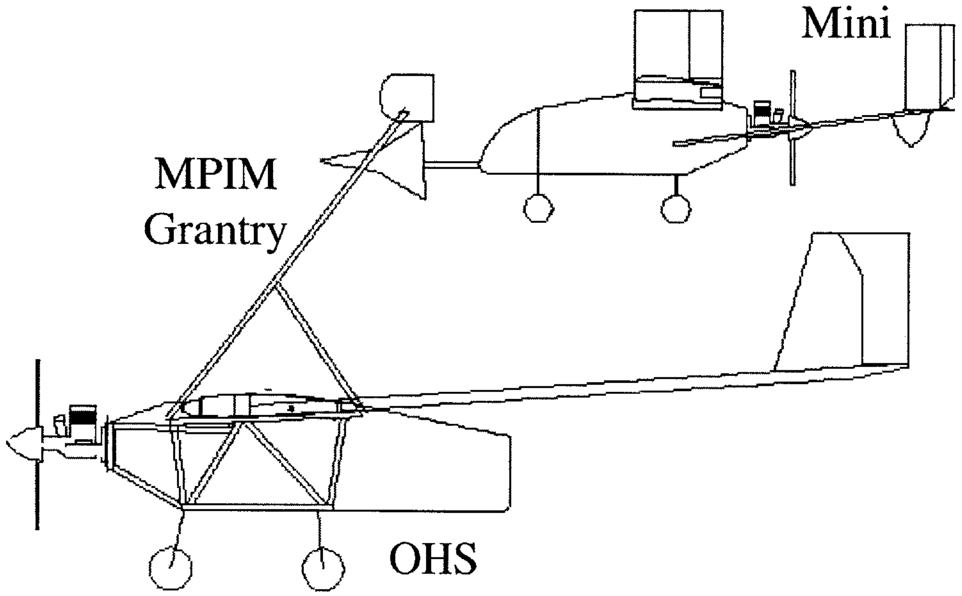


Figure 2.5 Parent-Mini Reintegration Configuration, side view

2.3.2 Velocity Concerns

In order to accomplish reintegration, both vehicles involved must have a similar speed with respect to both the ground and the air. The MPIM integration mechanism over the Parent vehicle is a source of drag, as analysed by Francois Urbain [6]. Figure 2.6 presents the OHS Parent Power Curve. In order to have enough speed for the Mini's control system to perform efficiently [15], the reintegration speed had to be chosen as high as possible.

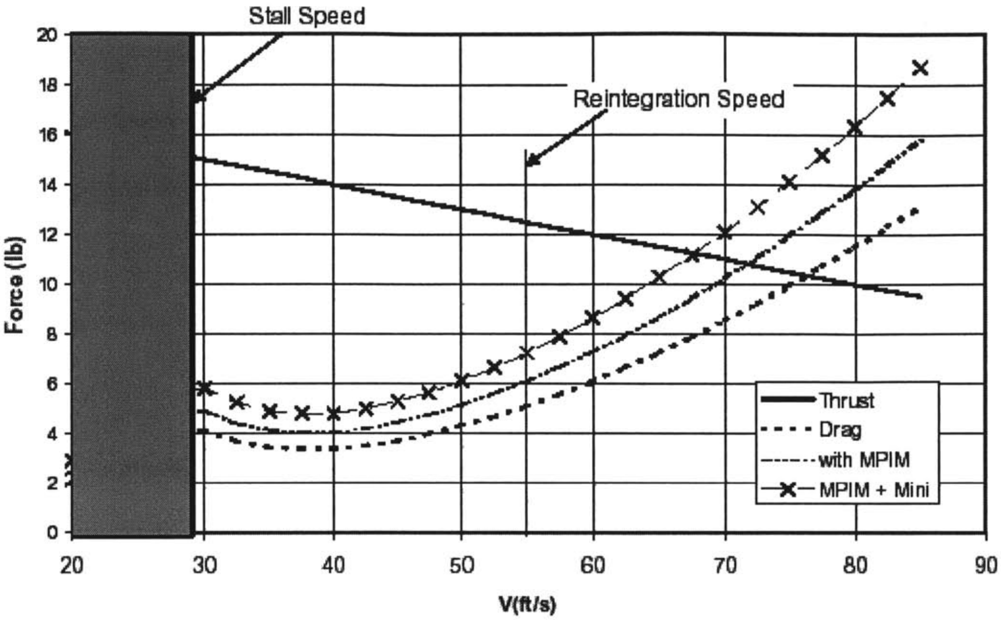


Figure 2.6 OHS Parent Power Curve

The first autonomous flight of the Parent has demonstrated it could achieve air speed in the range 18 to 28 m/s. Therefore the overlap with the Mini's safe air velocity (20 to 30 m/s) is large enough to perform reintegration.

2.3.3 The Different Phases of Reintegration

The process of reintegration has been broken down into three main phases developed by Damien Jourdan. Figure 2.7 illustrates these phases. Before reintegration, two vehicles (the Mini and the Parent) may be flying in different directions and at different altitudes.

- As soon as reintegration is initiated, which is the beginning of Phase I, the Parent sends its position to the Mini. The purpose of Phase I is to position the Mini in trail 10 to 20 meters behind the Parent. Phase I can be performed based on GPS and/or DGPS information.

- Once the target point of 10 to 20 meters behind the Parent is reached, Phase II begins. During this phase, the Mini uses a Vision based system developed by Sanghyuk Park [15] and revisited with the help of Thomas Jones.
- Then, the Mini reaches the point of reintegration, and the physical contact is called Phase III.

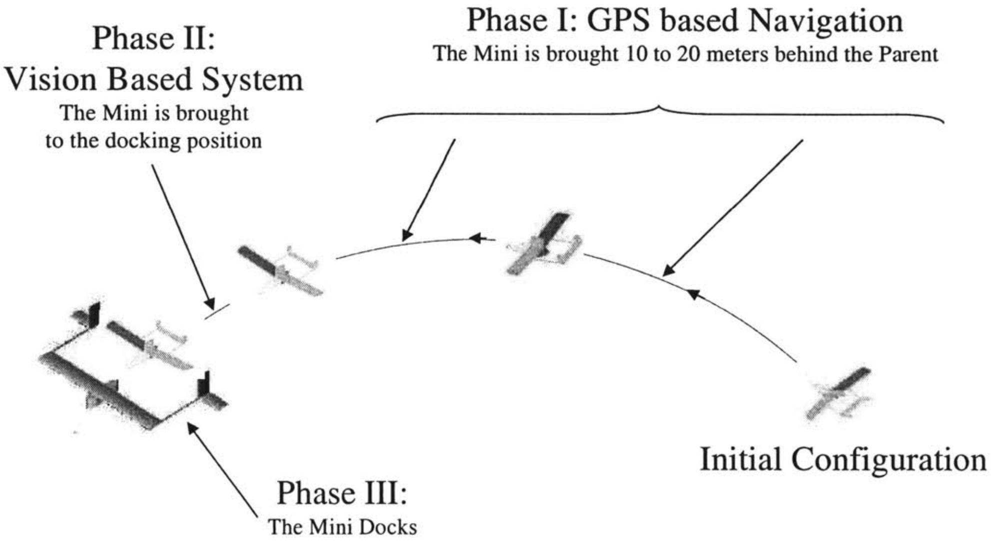


Figure 2.7 Reintegration Phases

In order to test Phase II and Phase III independently from Phase I many ingenious concepts have been studied and tested [6]. In particular, the team wanted to accomplish Phase II behind a van as shown on Figure 2.8. As mentioned by Francois Urbain [6], it has been decided abandon this path due to:

- the difficulty for a pilot to follow the required path with the good timing
- the proximity of the ground

- the difficulty to find an airport where to accomplish this task.

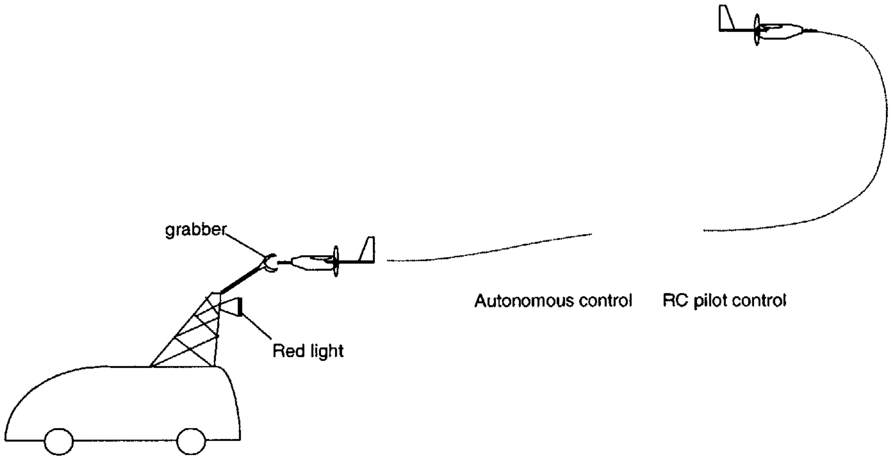


Figure 2.8 Phase II Test: Behind the Van Test Option

Hence, the team effort concentrated on Phase I. Phase II would then be tested in the air once Phase I is safe and secure.

2.3.4 Phase I of Reintegration

As it has been shown, Phase I plays a major role in demonstrating reintegration:

- it is a necessity in order to perform docking
- it has to be done autonomously as no remote pilot could safely get two airplanes close enough to each other (recently, the use of specific onboard cameras could have a pilot fly two RC planes close to each other after adequate practice).
- it is a necessity in order to demonstrate the other phases of the Rendezvous.

2.3.4.1 Phase I Requirements

Just before Phase I, the Parent and the Mini are in the air at different positions and velocities, both horizontally and vertically. Initially, the Mini needs to know the position of the Parent relative to itself in order to approach it. Once this position is known, the Mini needs to know where the Parent will be in the future. At the end of Phase I, the Mini should be in trail, 10 to 20 meters behind the Parent, with a 5 meters horizontal and vertical accuracy in order:

- not to collide with the Parent
- for the Vision system to be able to see the red LEDs placed on top of the MPIM.

Position Concerns

In order for the Mini to know the Parent's Position, different possibilities can be considered:

- a camera based system
- a radio(beacon) based system
- a pre-planned trajectory and pre-knowledge of the relative movement of each airplane based on time
- a GPS based system

The first three of these concepts are either difficult to deploy at long distance or judged to be impractical in the actual reintegration environment. Therefore, the team decided to use a GPS based system in order to know the position of each vehicle. GPS will be discussed in details in Chapter 3. Since each airplane uses GPS, it has been decided to transmit the Parent's position to the Mini via a transceiver. Details on communication will be developed in Chapter 4.

Carrier Phase Differential GPS (CDGPS) could lead to accuracy on the order of a centimeter, which might allow elimination of the Vision system. This option will be discussed in Chapter 5. However, in order to avoid the possibilities of multipath errors and the possible complexities of CDGPS, the team has decided to maintain a Vision based approach for Phase II.

Rendezvous Point Position Concerns

The Rendezvous could be accomplished at a Position decided in advance and the airplanes would only communicate the Rendezvous time. Such an option does not provide much flexibility to adapt to contingencies. Thus, during the third year of the project, the team decided to build a path planning guidance method, which has been developed by Damien Jourdan, and is introduced in Section 2.3.4.2 below.

2.3.4.2 Path Planning

In order to perform Phase I, the team looked at different options including either GPS or DGPS positioning. Phase I starts with arbitrary positions for the Parent and the Mini, and ends with the Mini 10 to 20 meters behind the Parent. The final distance depends on the performance of the Vision system used for Phase II and on the GPS system used (GPS or DGPS information).

Description of Phase I

Phase I is based on the following procedure designed by Damien Jourdan:

- Both airplanes start with arbitrary positions and velocities.
- The Parent Flies a 250 meter radius circle autonomously at the Reintegration Velocity (Vr).
- The Parent sends its position, its velocity and the characteristics of its trajectory to the Mini vehicle.
- The Mini starts a standard procedure (Figure 2.9) involving a climb (a), a straight line of length L (b), a turn (c) and another straight line (d). The length L is chosen

for synchronization purpose as the position of both airplanes are arbitrary at the beginning of the process.

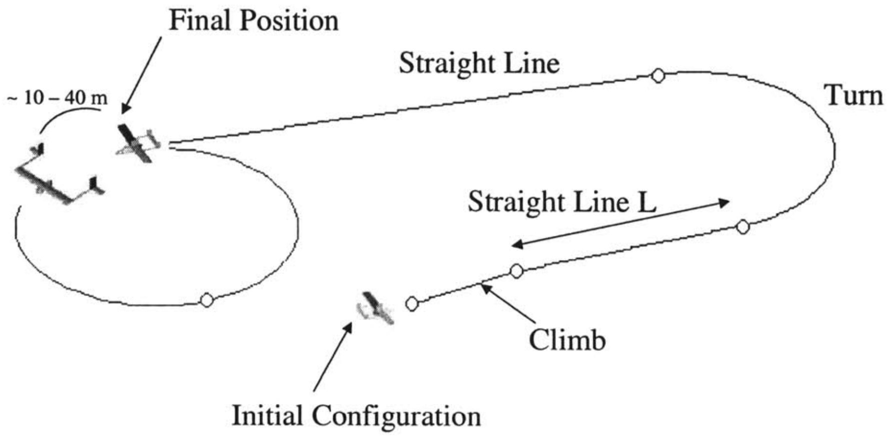


Figure 2.9 Phase I Nominal Trajectory

During this phase, the Parent and the Mini fly autonomously using GPS or DGPS information and proportional navigation: at each step of Phase 1, the vehicles aim at a point 100 meters away, on their respective trajectories. The accuracy of the difference of their position is explained in details in Chapter 3.

The Mini vehicle needs to know the Parent's position at all time (5Hz) for trajectory update. However, updates can be skipped for up to 1 second. Therefore, the choice between using GPS or DGPS for the Mini's Phase I is based on the following criteria:

- Accuracy of the horizontal precision expected:
  - the first Vision system developed required a final distance of 10 meters between both airplanes at the end of Phase I. Therefore, an accuracy of 3 meters horizontally is needed and DGPS is mandatory.
  - A Vision system recently developed for the Mini can localize the Parent 20 meters away from it. Hence, horizontal position accuracy of 7 meters is enough, and autonomous navigation based on GPS can be used.

- The flow of information that is achieved:
    - DPGS navigation requires the Parent's information to be sent to the Mini at a minimum of 2.5 Hz and is therefore highly dependant on communication.
    - GPS navigation requires update of the Parent's position every second and communication is therefore less of a constraint.

2.3.5 The Overall Control System

The major part of the vehicle onboard guidance, navigation and control systems have been developed by Sanghyuk Park [15] and Francois Urbain [6]. Apart from differences in architecture between the Parent's and the Mini's onboard software due to GPS and Communication constraints, both airplanes are very similar in terms of hardware architecture. Additional small differences are attributable to the presence of the Vision system on the Mini (Figure 2.10) and the presence of a left and right surface control system for the OHS Parent (Figure 2.11). Such differences have been covered in detail by Francois Urbain [6]. The most recent improvement is the use of PC104+ stacks on both the Parent and the

Mini.

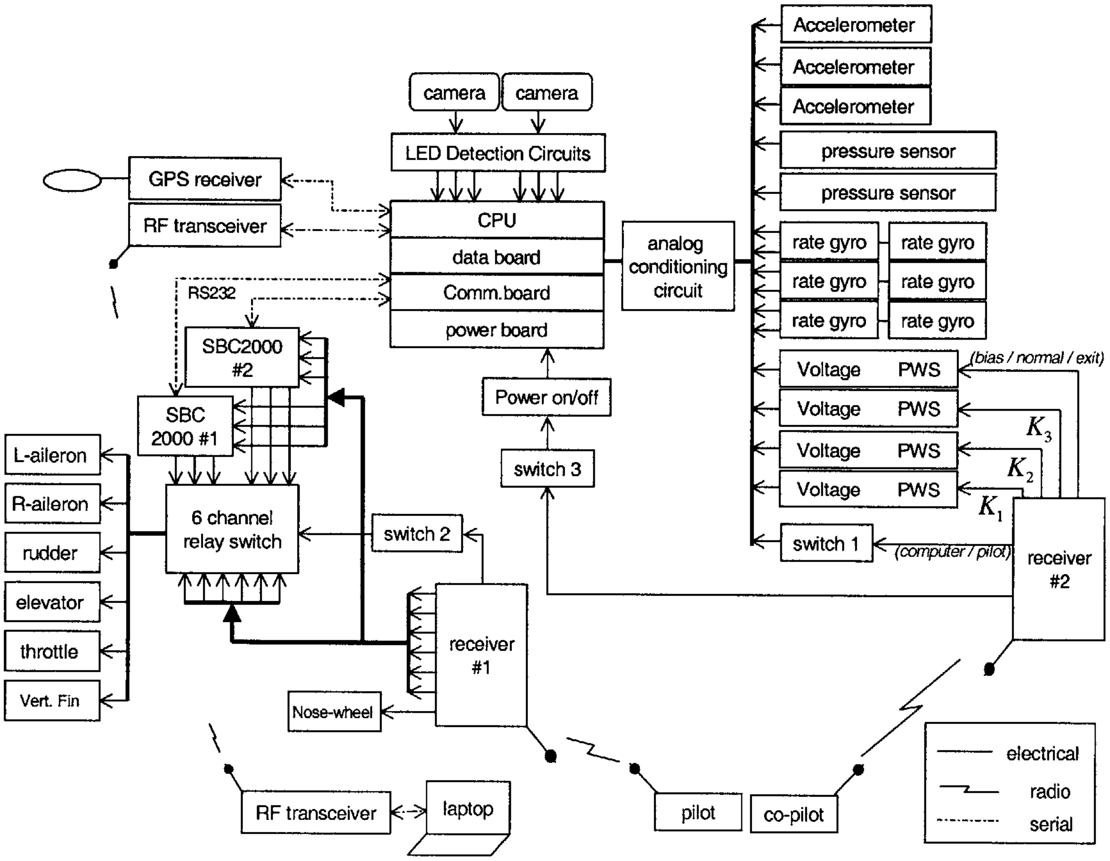


Figure 2.10 Mini Demonstration Vehicle Flight Control Avionics

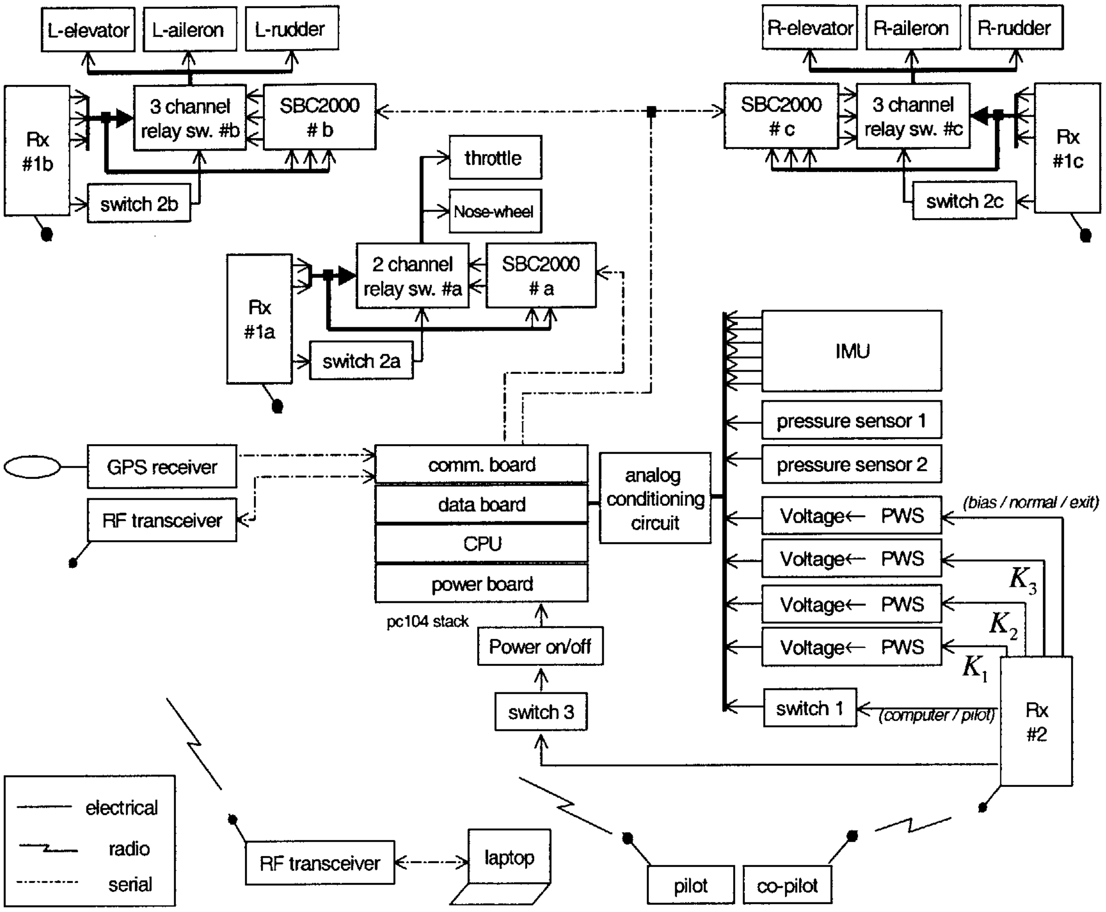


Figure 2.11 Parent Demonstration Vehicle Flight Control Avionics

2.4 Global Positioning System and Communication Requirements

One of the most critical aspects of the Rendezvous problem is the accuracy of the relative positions of the two autonomous vehicles. By dividing the Rendezvous problem into different phases, the team also believed that it could make the accuracy problem more manageable, and that GPS navigation provides the required accuracy during Phase I.

2.4.1 GPS Accuracy Requirements

For the controllers to perform correctly, the GPS signal has to be provided at a rate of at least 2.5Hz (the team chose 5Hz), and to be smooth enough to be handled by the controllers. Unfortunately, Raw GPS data (as defined in Chapter 3) are not smooth enough. Therefore they are filtered within a Kalman filter (as described in Chapter 3) by using velocity inputs. The filtered GPS data results serve as inputs for the controllers.

There is no specific requirement on the accuracy of the filtered GPS data in terms of absolute position with respect to the center of the earth. Rather, the filtered GPS position has to be precise according to the following criteria:

- for the Parent:
  - the position at time t+5s must be within at most 7 meters horizontally and 10 meters vertically, with 95% probability, of the expected position (at time t+5s) as predicted from the estimated position and velocity (at time t).
- for the Mini:
  - the accuracy of the estimate of the relative position with respect to the Parent must be within 3 meters horizontally and 5 meters vertically, with 95% probability, if the Mini is less than 20 meters behind the Parent (use of DGPS or Vision System mandatory).
  - the actual positioning of the Mini with respect to the Parent must be within 7 meters horizontally and 10 meters vertically, with 95% probability, if the Mini is not within 20 meters of the Parent.

2.4.2 Communication Requirements

The primary communications requirement is the necessity for the Parent vehicle to transmit its position to the Mini vehicle at 5Hz. Furthermore, for purposes of monitoring events, communication is necessary to display the position of each vehicle on a ground station computer. The main reason for this choice is flight safety. If it is noticed on the ground station that both vehicles get too close to each other, the mission can be readily aborted. During the different tests with one or several vehicles, it has been shown that the ground station is an excellent tool to understand how the UAVs perform. Its presence has certainly avoided a number of situations that could have led to crashes by enabling the pilot to take control of the UAV in time to avoid disaster. Moreover, the team decided that it could be useful to have some control over the airplane (such as the decision of the setup of the position of reference) on the ground station laptop. All these added features create constraints on communication, and the solution proposed will be discussed in Chapter 4.

2.5 Chapter Summary

This Chapter has presented an overview of the PCUAV project focusing on the major technological challenges presented by the Rendezvous concept. It then discussed the way the Rendezvous problem is solved. Finally, it introduced the different requirements that must be met by the GPS and Communication components of the project.

Chapter

3

GPS and DGPS

3.1 Chapter Overview

This chapter presents an in-depth discussion of the application of GPS to the PCUAV project. The receiver the team decided to use, based on the requirements presented in Chapter 2, is first introduced. Then the approach to perform autonomous navigation with one single GPS receiver is presented, followed by a presentation of the Differential GPS.

3.2 GPS Signal Characteristics

GPS is a satellite based navigation system designed to provide properly equipped user with position and velocity information anywhere on earth. The GPS space segment consists of 24 satellites (21 primary and 3 spare satellites) arranged in six orbital planes. Each orbital plane contains 4 satellites in 12-hour orbits. This constellation provides a user located anywhere on the world with the visibility of four or more satellites at any time. This is necessary for continuous positioning capability [2]. Since the satellites are not in geosynchronous orbits, the satellite geometry, as seen by an observer at a fixed location on earth, is continuously changing.

The oscillators on board the GPS satellites generate a fundamental frequency f_0 (f_0=10.23 MHz) with a stability in the range 10^{-13} over one day. Two carrier signals are then generated by integer multiplications of f_0:

- L1 at 1575.42 Mhz (=154f_0)
- L2 at 1227.60 Mhz (=120f_0)

The carrier frequencies are modulated by various spread spectrum signals that contain information necessary to determine position, velocity and time. These codes will be discussed in Section 3.2.3 and Appendix D.3. The basic principles of position and velocity determination are explained below.

3.2.1 Principle of Position Determination

The basic function of GPS positioning is the determination of the user's position $\mathbf{r}_u$ from the GPS signal in a given frame of reference. The most commonly used coordinate system currently used by GPS is the 1984 World Geodetic System (WGS-84), but the user's position can be expressed in any desired coordinate system. In the WGS-84, coordinates are commonly expressed in terms of latitude, longitude, altitude (can be positive or negative relatively to the WGS-84 surface at the given latitude and longitude).

The GPS receiver determines the position of its antenna by using the propagation delays of the GPS signals from an array of satellites to the receiver's GPS antenna (i.e. the time interval for the signals to travel from the satellites to the receiver). The time reference of the GPS satellites is precisely known (GPS system's time). If differences between satellite clocks occur, they can be quantified exactly and modulated on the carrier as navigation messages. The GPS receiver's time, on the other hand, relies on its internal clock, which, due to inherent clock errors, will be offset from the GPS system time by an unknown bias

b_u. The observed propagation delay t_{pd} of the GPS signal from each satellite is, thus, offset from the actual signal travel time by the same bias b_u.

The observed propagation delay t_{pd} from each satellite, scaled by the speed of light in vacuum c, corresponds to a measurement of range from the satellite to the receiver, and is commonly referred to as pseudo-range ρ, i.e.:

$$\rho = c.t_{pd} \qquad \textit{(eq. 3.1)}$$

The term pseudo-range is used because ρ is the sum of the geometric range from the satellite and the range error due to the receiver's clock offset. The pseudo-range observation ρi between a user and satellite i can be related to the user's position and clock offset by:

$$\rho i = |\mathbf{r}_i-\mathbf{r}_u|+c.b_u+\varepsilon_i \qquad \textit{(eq. 3.2)}$$

where $\mathbf{r}_i$ is the three dimensional satellite's position at the signal transmit time, $\mathbf{r}_u$ is the receiver's position at the reception time, and ε_i is the composite of various error sources including atmospheric delays, Selective Availability (SA) when it is switched on, and receiver's noise.

The system (eq. 3.2), consisting of four unknown variables ($\mathbf{r}_u$ and b_u), can be solved as long as at least four satellites are tracked by the receiver. The process of tracking the satellites will not be discussed, and the reader may refer to [2] for further details.

3.2.2 Principle of Velocity Determination

Three dimensional user velocity $\mathbf{v}_u$ can be determined from the observed frequency shift of the received GPS carrier signal. The observed carrier frequency differs from the nominal L1 or L2 carrier frequency due to Doppler shifts caused by the relative motion of the user with respect to the satellites, as well as a receiver clock frequency bias f_u.

The Doppler shift caused by the relative motion of satellite i and the user (Δf_{Di}) is given by the projection of the relative velocity onto the line of sight, scaled by the ratio of the transmitted carrier frequency L1 (in most receiver cases) to the speed of light c, i.e.:

$$\Delta f_{D_i} = \left(-\frac{\mathbf{v}_i - \mathbf{v}_u}{c} \bullet \mathbf{1}_i\right) \bullet L_1 \qquad \textit{(eq. 3.3)}$$

where $\mathbf{v}_i$ is the velocity of satellite i, $\mathbf{1}_i$ is the line of sight from the user's receiver antenna to the satellite i. The satellite's velocity vector $\mathbf{v}_i$ can be computed in the receiver using ephemeris information modulated as a navigation message onto the carrier signal sent by the satellite (see Appendix D.3). The observed carrier frequency shift can be related to the user's velocity and unknown receiver's clock frequency bias f_u as:

$$\Delta f_i = \Delta f_{Di} + f_u + \eta_i \qquad \textit{(eq. 3.4)}$$

where η_i is the composite of various errors in the frequency shift observation including effects of SA and receiver's noise.

By combining these equations, the following system remains to be solved:

$$\frac{c}{L_1}\Delta f_i = \delta_i = (v_i - v_u) \bullet 1_u + \delta_u + \zeta_i \qquad \textit{(eq. 3.5)}$$

where δ_i is referred as pseudo-range rate (or delta range) from the user to satellite i, δ_u is the receiver clock frequency bias and ζi is the error in the observation, all in m/s. Once again, the system can be solved as long as there are at least four satellites. As it is necessary to know the line of sight in order to compute the velocity, the user's position has to be computed first.

3.2.3 The GPS Modulated Signals

The GPS carriers are modulated by codes to provide satellite clock readings to the receiver and to transmit information such as the orbital parameters. The codes consist of sequences of binary values 0 and 1. Parts of the components of the signal are the following:

- P-code at f_0=10.23 Mhz
- C/A-code at $f_0/10$=1.023 Mhz
- Navigation Message at $f_0/204600 = 50.10^{-6}$ Mhz

The Coarse/Acquisition (C/A) code and the Precision (P) code are used for satellite clock readings and are characterized by a pseudorandom noise (PRN) sequence [2]. The C/A-code is repeated every millisecond and the P-code is repeated approximately once every 266.4 days. Both are in quadrature (i.e. 90° offset). The coding of the Navigation Message requires 1500 bits and, at the frequency of 50Hz, is transmitted in 30 seconds. If the unmodulated carrier is denoted $L_i(t)=a_i\cos(f_i t)$ and the P-code, C/A code and Navigation Message respectively P(t), CA(t) and D(t), the modulated carriers are represented by (eq. 3.6) and (eq. 3.7):

$$L_1(t) = a_1P(t)D(t)\cos(f_1t)+a_1CA(t)D(t)\sin(f_1t) \qquad \textit{(eq. 3.6)}$$

$$L_2(t) = a_2P(t)D(t)\cos(f_2t) \qquad \textit{(eq. 3.7)}$$

3.2.3.1 Components of the Navigation Message

Details on several components of the Navigation Message can be found in Appendix D. The Navigation Message essentially contains information about the satellite's clock, the satellite's orbit, the satellite's health status and various correction data. The total message (consisting of 1500 bits) is divided into five subframes consisting of 300 bits each. One subframe contains 10 words of 30 bits each.

Each subframe starts with the telemetry word (TLM) containing a synchronization pattern and some diagnostic messages. The second word of each subframe is the hand-over word (HOW).

The first subframe contains the GPS week number, a prediction for the user's range accuracy, indicators on the satellite's health, on the age of the data, an estimation of the group delay of the signal, and three coefficients for a quadratic polynomial to model the satellite's clock corrections.

The second and third subframes transmit the broadcast ephemerides of the satellite as described in Section 3.2.3.2.

The contents of the fourth and fifth subframes are changed in every message and have a repetition rate of 25. The total information needs 12.5 minutes for transmission and is mostly of military use or information on the ionosphere, UTC and Almanac Data.

3.2.3.2 The Ephemerides Data

These sets of data are necessary for the determination of position and velocity vectors in a terrestrial reference frame. The data sent differ in accuracy:

- Broadcast ephemerides (SA on): 2-50 meters accuracy depending on the level of SA
- Broadcast ephemerides (SA off): 2-5 meters accuracy or better

The most recent broadcast ephemerides are used to compute a reference orbit for the satellites. These orbital data could be accurate to approximately 5 meters based on a three uploads per day [2]. For the PCUAV tests, downloads of these data are done before every test and at each new update when flying using DGPS. It is important to note that the ephemeris data take up to 1 minute to download from the time the code is ran. Therefore, the DGPS code is only accurate after that specific amount of time, and the vehicles should not be flown using DGPS information unless ephemerides data have been downloaded. The ephemeris data message format is presented in Appendix D.3.

Satellites' Position and Velocity Information Computation

Once the ephemerides data are downloaded, position and velocity are computed for each satellite in the Earth Centered Earth Fixed (ECEF) frame as described in Appendix D.3. Then the azimuth and elevation of the satellite from the receiver's position is computed knowing the receiver's position. These data are then compared to the relative position of the satellite given by the output of the binary message 33 of the Allstar GPS receiver (see Appendix D). In this message, accuracy is limited to 1 degree. Therefore the computed solution is kept as long as it does not differ from the output of message 33 by more than 3 degrees. Such an action is taken to prevent any error due to the computer internal's computation. In case of error or of no reception of a satellite's ephemerides, message 33 output is kept. In any case, the author has never observed any difference greater than 2 degrees in the output of message 33 and the computation of the ephemerides data.

3.2.4 Determination of User Geodetic Coordinates

The ephemerides data and the solutions for the determination of the GPS position and velocity are done in the Earth-Center Earth-Fixed (ECEF) coordinates [2]. This frame is interesting for computation reasons. However, for practical reasons, it is more useful to use position in geodetic coordinates, i.e. in terms of latitude, longitude and height above or below a reference surface.

The WGS-84 ellipsoidal model of the Earth [5] is used. In this Earth's model, the equatorial cross-section of the earth has radius a = 6378.137 km, which is the mean equatorial radius of the earth. The cross-sections of the Earth normal to the equatorial plane are ellipsoidal. In an ellipsoidal cross-section perpendicular to the equatorial plane, the major axis coincides with the equatorial diameter of the Earth. Therefore the semi-major axis has the same value, a, as the mean equatorial radius. The minor axis of the ellipsoidal cross-section (as shown in Figure 3.1) corresponds to the polar diameter of the Earth. The semi-minor axis, b, is therefore taken to be b = 6356.7523142 km.

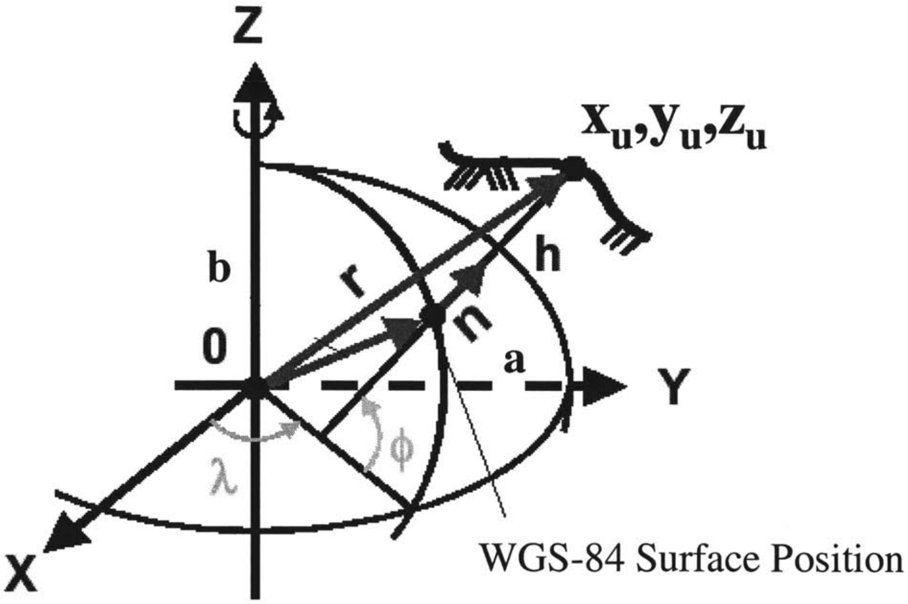


Figure 3.1 Ellipsoidal Model of Earth in the WGS-84 Representation.

The eccentricity can therefore be computed as:

$$e = \sqrt{\left(1 - \frac{b^2}{a^2}\right)} \qquad \textit{(eq. 3.8)}$$

If a position in the ECEF frame is given by the vector **u**=(x_u,y_u,z_u), the geodetic longitude (λ) can be easily computed [7] as:

$$\lambda = \begin{cases} \operatorname{atan}\left(\frac{y_u}{x_u}\right) & x_u \geq 0 \\ \pi + \operatorname{atan}\left(\frac{y_u}{x_u}\right) & x_u < 0, y_u > 0 \\ -\pi + \operatorname{atan}\left(\frac{y_u}{x_u}\right) & x_u > 0, y_u < 0 \end{cases} \qquad \text{(eq. 3.9)}$$

As it can be realized from Figure 3.1, the ellipsoid normal does not usually point towards the center of the earth. Therefore, the computation of latitude (ϕ) and height (h) is slightly more complicated than in the spherical case and is developed in Appendix D.3.

3.3 Choice of the Receiver

Thomas Jones introduced the concept of GPS based flights to the PCUAV group in 1999 by building a Payload Delivery Vehicle (PDV). In order for the PDV to be launched and hit the ground with a 100 meters accuracy, and due to the configuration of the PDV (Figure 3.2), a receiver which would send an update of a position at a rate of 5Hz had been chosen. It was the Allstar P/N 220-600944-0XX produced by Canadian Marconi Company [3].

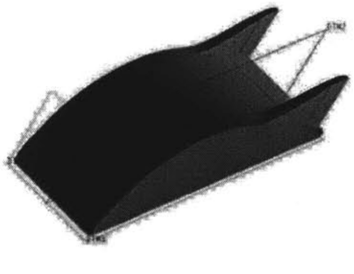


Figure 3.2 Payload Delivery Vehicle (PDV) Demonstration Vehicle

For the new applications involved by the Rendezvous, the team decided to keep the same receiver for the following reasons:

- sufficient position accuracy (Table 3.1)
- transmission of information at 5Hz
- low cost.

NAVIGATION ACCURACIES	SA INACTIVE	SA ACTIVE	DGPS
Receiver Performance 2 SIGMA (95%)			
Horizontal Position	30 meters	100 meters	2 meters
Ground Speed*	0.13 m/s	0.3 m/s	0.05 m/s
Track Angle True**	1.0 deg	3.0 deg	0.1 deg
Vertical Speed	0.16 m/s	0.6 m/s	0.1 m/s
Altitude	40 meters	160 meters	5 meters
N-S Velocity*	0.1088 m/s	0.21 m/s	0.035 m/s
E-W Velocity*	0.1088 m/s	0.21 m/s	0.035 m/s
Time***	1 usec	1 usec	1 usec

* Velocity accuracies are for straight and level motion during zero acceleration. Dynamic errors due to jerk of $2 m/s^3$ results in a maximum additive error of 4.2 m/s

** For a ground speed of 20 km/hour or greater

*** At the rising edge of Time Mark output

These accuracies are met for the following conditions:

- HDOP = 1.5
- VDOP = 2.0
- TDOP = 0.8

Table 3.1 Position and Velocity Outputs of the Allstar GPS Receiver

In the next paragraphs, different characteristics of the receiver will be discussed including some hardware properties and the accuracy of the messages sent by the receiver via its serial port.

3.3.1 GPS Receiver Hardware Profiles

The GPS receiver used is the Canadian Marconi Company Allstar Receiver Part Number 220-600944-00X as described in Appendix C. The GPS antenna used with the receiver is the active geodetic Antenna with gain +12dB part number AT575-70 by AeroAntenna Techonology Inc. It is described in Appendix C.

3.3.1.1 The Receiver Itself: Time Mark Output 1 PPS

One important aspect of the receiver is its ability to align its data output time on GPS System's Time. This property is largely used in order to synchronize communication between the Parent Vehicle and the Mini Vehicle as explained in Chapter 4.

For the GPS receiver set in its "Aligned on GPS Time" mode, the Time Mark Output of the receiver and GPS measurements will be typically aligned on GPS time (itself aligned on Coordinated Universal Time UTC) with ±200ns accuracy. Even at 5Hz, the Time Mark is output only once per second with the rising edge corresponding to the receiver's epoch (1 Hz). The timing relationship for the GPS Time Mark output is defined in Figure 3.3. The Navigation Data ID 20 (see Section 3.3.2 and Appendix D.1) defines the UTC time of the epoch and the rising edge of the Time Mark is accurate to within 1μsec of UTC.

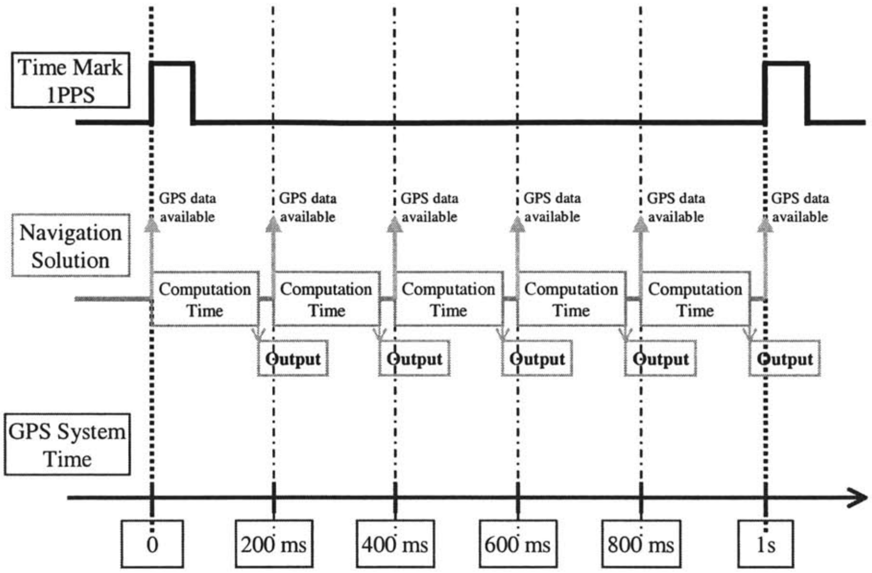


Figure 3.3 Allstar GPS Receiver Timing Relationships

3.3.1.2 The GPS Antenna

The GPS receiver includes a Low Noise Amplifier (LNA). This +20dB LNA permits reasonable performances with a passive GPS antenna. According to CMC [3], depending on the cable (the RG-80 Low Loss Cable[1] [RG-58/U LLDS80] is used by the PCUAV team) loss between the antenna element and the GPS receiver, and also the position accuracy requirements, a +12dB to +36dB Active GPS Antenna could be forecast. As the PCUAV vehicles are relatively small and as the position of the Antenna has to be close to the Center of Gravity (CG) of the vehicles for concerns linked to the dynamic model of the control system, the length of our cable never exceeds 2 meters. For this reason, a choice for an Active Antenna was limited to those with gain less than 12dB. Moreover, due to the size

---

[1] The RG-80/U LLDS80 cable is a low loss noise coax cable which is double-shielded [3]

of our vehicles and to the necessity to have low weights, the choice came down to the comparison of two antennas manufactured by AeroAntenna Technology Inc:

- Antenna Part Number AT 575-32
- Antenna Part Number AT 575-70

As the team was already in possession of an Antenna Part Number AT 575-70 and due to the great similarity of both antennas, the team decided to keep it. The Antenna Characteristics are presented in Appendix C.

3.3.1.3 GPS Receiver to Computer Communication Board

The Allstar GPS receiver is delivered with a development kit based on an interface between the GPS receiver itself and a serial port. Unfortunately, the development kit interface is too large and too heavy for the PCUAV applications. Moreover it provides lots of features which are unnecessary for the PCUAV project. Therefore, the team decided to build a new interface board.

Serial Port Hardware Properties

The serial port creates a link between a computer and the outside world. The computer's serial port plug is a DB9 Male. Within the different pins of the serial port three can be distinguished:

- The Ground
- The Transmission Line (TX)
- The Reception Line (RX)

The signal sent via these lines is of RS232 type. It presents two levels corresponding to bit levels:

- bit 0 is associated with usually -9V (between -3 and -25 V)

- bit 1 is associated with usually +9V (between +3 and +25 V)

The serial port uses an asynchronous transfer of information protocol: any piece of information is transmitted bit after bit. Bits are sent within a frame of 8 bits creating a byte. Sometimes, the information is sent over a frame of 7 bits plus one bit consisting of a parity check. For the PCUAV system, the first configuration is used. Before each byte is sent, there is a start bit. At the end of each byte, a stop bit is sent and the signal is set to the higher possible value (+9V).

Changing RS232 Signals into TTL Signals

The GPS receiver uses signal level inputs corresponding to bit levels with the following characteristics:

- bit 0 is associated with 0V
- bit 1 is associated with +5V

Therefore, in order to establish communication between a computer serial port and the GPS receiver, the RS232 signal has to be changed into a TTL signal and vice-versa. A

Max232 circuit (Figure 3.4) supplied in +5V performs this transformation.

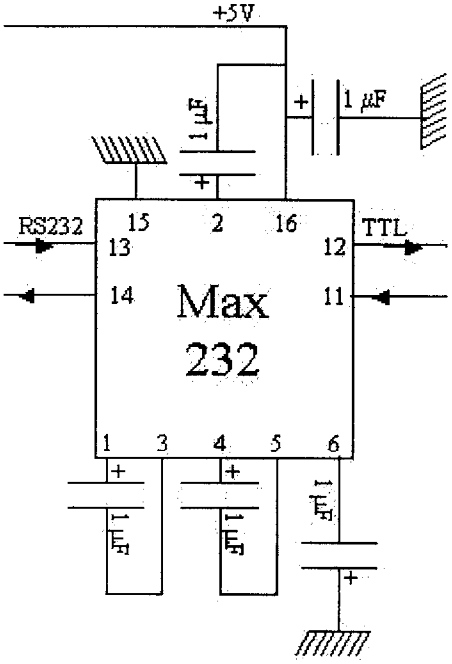


Figure 3.4 Max232 Circuit Schematic

Communication Board

The overall communication board power is +5V from a regulator with +9V input. This regulator is also necessary for the Allstar GPS receiver which requires a voltage of +5V (+10%/-5%) and an active current of typically 70mA (min:18mA, max:130mA). The overall architecture of the communication board including the pin connections is presented in Figure 3.5:

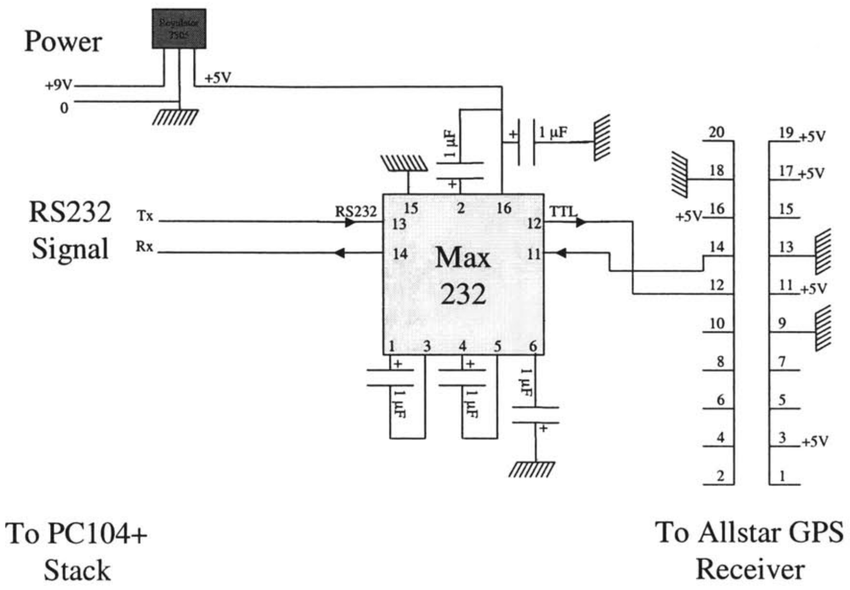


Figure 3.5 GPS Receiver/Computer Communication Board Schematic

3.3.2 The Different Messages of the Receiver

The Allstar GPS receiver [3] can handle two different sets of messages based on two different protocols:

- The NMEA Protocol
- The Binary Protocol

The NMEA protocol is the easiest one to use. Therefore it is the first one the author decided to use as he had no previous experience with GPS receivers. The different elements of the NMEA messages are presented in Appendix D.2.2 and the NMEA messages recognition procedures are presented in Appendix D.2.1. The messages sent in the NMEA mode by the CMC Allstar Receiver do not provide sufficient information and it has been necessary to switch to the Binary Protocol with 19.2 kbps baud rate. The different sets of messages used by the PCUAV team are described in Appendix D.1.

The Allstar Binary Messages

The messages are described in Appendix D.1.2 and are referred as:

- User Data (Message ID: 20): These data give time, position in terms of latitude, longitude and altitude in the WGS-84 reference frame. Velocity in all directions and DOP indications are also sent. These data are very useful and the PCUAV team used them until it realized they were filtered. This filter induces a delay of up to 5 seconds which will be discussed in Section 3.4. Since this discovery, this message has been disregarded.
- GPS Raw Data (Message ID: 21): These data give time, position and velocity in the ECEF reference frame. Compared to the previous data, these are not filtered and are just the result of computation of position in the ECEF frame, as explained in Section 3.2.4.
- Ephemeris Data (Message ID: 22): This message gives the Ephemeris data as they have been presented in Section 3.2.3.2. This message is only sent once by the receiver, and again every time the ephemeris data changed. However, it is possible to force the GPS receiver to send these data upon request. This option is used every time a PCUAV program requiring ephemeris data is used (this is the case for any flight involving DGPS).
- Measurement Block Data (Message ID: 23): This message gives time and the Signal to Noise Ratio (SNR), code phase and carrier phase of each satellite tracked. This message is mainly used for DGPS.
- Satellite Visibility Data and Status (Message ID: 33): This message is used as a backup for ephemeris data as it gives the position of the satellites relatively to the

GPS receiver's antenna. However, the accuracy is worse than what can be computed out of the ephemeris data. Hence, this information is compared with ephemeris data information. If a difference in terms of elevation and azimuth angle less than 3 degrees is observed, the ephemeris data are considered. Otherwise, message 33 output is used.

Message Block Structure

All communication is done using message blocks. Each message block consists of a header and possibly data. The data portion of the block is of variable length depending on the message ID. The header has a fixed length of 4 bytes, consisting of a Start-of-Header character (SOH), Block ID, Block ID Complement and Message Data length. Each block has a truncated 16-bit word containing the Checksum associated with the complete content of the block. It is appended at the end of the data portion of the block.

The Message Block structure is as follows:

- byte 1 [SOH] : Start of Header character (decimal 1).
- byte 2 [ID] : Byte containing the Block ID numeric value. The block ID number field is used uniquely to identify the format of the data portion of the block. Since only 7 bits are needed for the ID, the higher bit is used to encode information about start/stop of broadcast of data blocks and to set special modes for command messages. This prevents unnecessary increase in overhead by eliminating any extra bytes in the protocol.
- byte 3 [Cmpl ID] : 1's complement of the ID field. This can be calculated as Cmpl ID = 255 - ID. This field, in conjunction with the SOH, helps to synchronize the message blocks. Since the SOH character can appear within the

data, the Cmpl ID field validates the header contents and thus confirms the start of the block.

- byte 4 [Message Data Length] : One byte containing the length of the data part of the message in bytes (excluding header and checksum).
- byte 5 [Data Word 1] Least Significant Byte (LSB)
- byte 6 [Data Word 1] Most Significant Byte (MSB)
- .. ..
- [Checksum] LSB
- [Checksum] MSB

The Checksum field contains the checksum value for the complete message transmitted (it includes header and data). The 16-bit Checksum is defined as the 16-bit sum of all the unsigned 8-bit bytes starting at the beginning of the header. Any overflow or carry to the 16-bit sum is discarded immediately. Therefore, it adds unsigned bytes to produce a 16-bit result.

Message Recognition Procedure

The message recognition procedure is very much alike the one developed for the NMEA protocol (Appendix D.2). The entry is the array representing the characters received at the serial port in the arrival order:

- Identify a SOH.
- If the SOH is followed by an ID plus a CmplID, consider the message Length. Otherwise, next time the procedure is entered, start at the byte located after the SOH.
- Knowing the Length of the message, make sure the message is compatible with the Checksum (if the whole message has not arrived yet, next time the procedure is entered, start at the SOH).

  - If it is the case, compute the data and the next time the procedure is entered start at the end of the message.
  - Otherwise next time the procedure is entered, start at the byte after the SOH.

This procedure is efficient and the probability that an error occurred is very low:

- If an error occurrs during the transmission of the message via the serial port, it is be recognized by the Checksum and it is not taken into account.
- The probability that a fake message has the same characteristics as a real one with spectific ID number (taken in a limited set of ID number) after a SOH, the good Cmpl ID, a message Length compatible with the ID and a coherent Checksum has a very low probability to occur: it has never been observed in any test performed by the PCUAV team.

3.4 GPS Testing

The purpose of this chapter is to discuss the tests done by the PCUAV team regarding autonomous flight of one single vehicle using the GPS receiver described in Section 3.3. This section will describe the different progress made by the team. It will include flight tests and ground tests results. This discussion will not focus on the control part of the system developed by Sanghyuk Park or on the different steps of Phase I designed by Damien Jourdan, but the work described here is closely related to theirs.

The first autonomous flight tests were performed with an Avionics Test-bed Aircraft (ATA Figure 3.6). Flight tests using this aircraft is part of the risk management strategy of the project. The ATA is a modified version of a standard radio control kit aircraft. The goal for its use is to reduce the risk of damage to the PCUAV vehicles [6] by having a vehicle which can be rebuilt quickly and easily in case of a crash.

Figure 3.6 Avionics Test-bed Aircraft (ATA II) in Flight

3.4.1 Choice of the GPS Receiver Communication Protocol

Within the binary mode, three different sets of data could be used:

- User Data with data provided in the WGS-84 reference system in terms of longitude, latitude and altitude
- Raw data with data provided in the ECEF coordinates
- Computed Raw data from ephemerides and code phase outputs.

As they require less computation for navigation in a small local area, the first plan was to use User Data. After the first flight tests the team discovered that these data were filtered, and therefore delayed. The team decided to use Raw Data. The results provided by this set of data are sufficient and the "Computed Raw data" option (which would only provide data available faster but would require more computation within the onboard computer) has not been implemented inside the onboard computer of the UAVs.

3.4.1.1 Time Delay in the User Data and Raw Data

The first test for autonomous flight in June 2001 led the team to look more closely at the data provided by the GPS receiver. By looking at the different navigation, guidance and control data saved during the flight, Sanghyuk Park discovered that there is a delay of up to 5 seconds between the control command and the reaction of the airplane in terms of position. The controllers could not handle such a delay.

Evidence of Delay between Raw and Filtered Data

The author then compared User and Raw data from ground tests. The goal of these tests was to make sure that the computation of data in a local frame from ECEF coordinates was accurate as well as to identify unexpected delays. Two sets of ground tests were performed:

- On the roof of building 17 at MIT, where there is likelihood of multipath problems for the GPS signals, because the building is surrounded by taller buildings.
- At Briggs Field, at MIT, where multipath could be reduced due to the size of the field (500m*200m).

Tests were performed with either static or moving antennas and all tests identified a delay due to filtering in the GPS receiver. This delay was found to be of up to 5 seconds and can be identified on Figure 3.7.

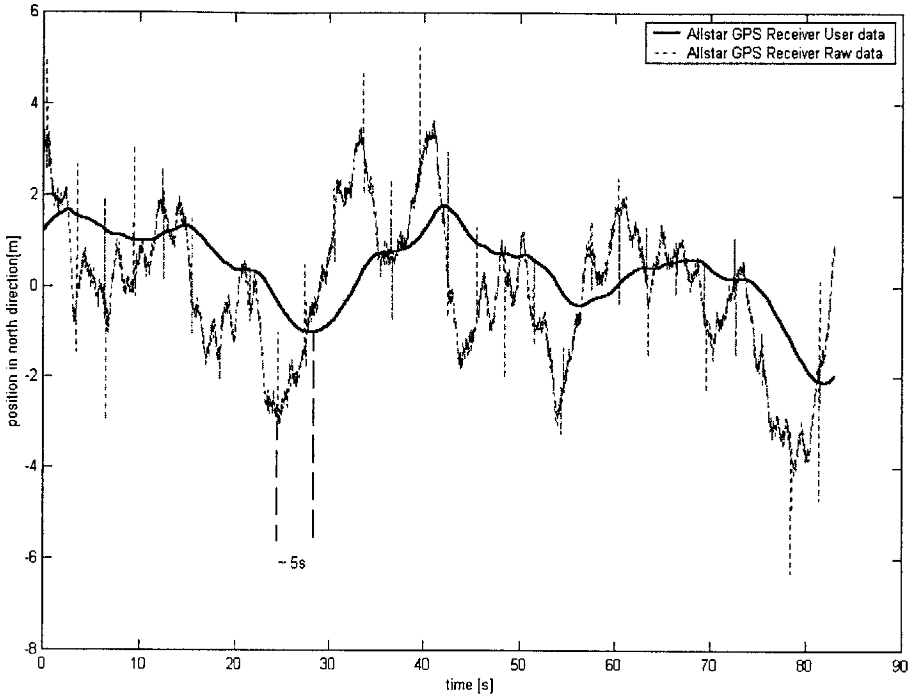


Figure 3.7 Time Delay between User Data and Raw Data

Existence of Delay in Raw Data

The author decided to contact Canadian Marconi Company (CMC). CMC confirmed the existence of this delay in the User data even though it was not mentioned in any document available to the author. CMC explained to the PCUAV team that it was due to computation and more precisely filtering. CMC also informed the PCUAV team that Raw Data were unfiltered and thus did not present such a delay.

A delay of up to 200 ms in the Raw data could be noticed in the 5Hz mode if the GPS Time Alignment option was chosen. This delay is due to computation and to alignment on GPS Time as mentioned in Section 3.3.1.1. This delay can be identified (as seen on Figure 3.8 where DGPS raw data are delayed compared to potentiometer data by up to 400ms) by tests with a pendulum performed as explained in Section 3.4.2.3. Such tests have been performed to identify delays in GPS and DGPS data for filtered and unfiltered information. The results of the tests are summarized in Table 3.2.

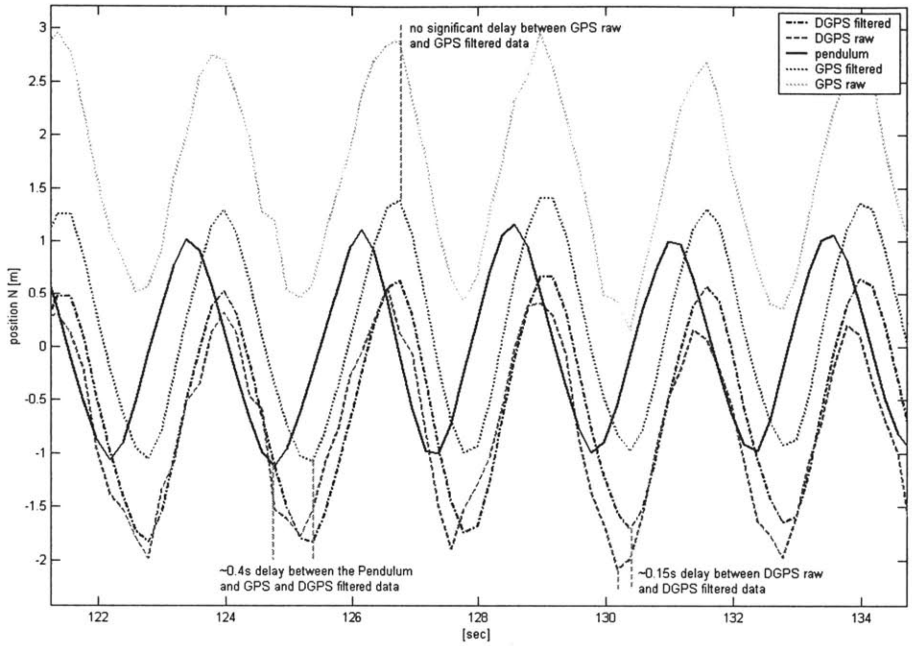


Figure 3.8 Time Delay in the Computation of GPS Position

Test	Data	Total Delay	Delay to reception of data and computation (expected delay)	Due to Kalman filter delay	Unexplained Delay (refers to delay not specified by CMC)
2 Receivers	Horizontal Velocity	400 ms	100-150 ms	0 ms	250 ms
1 Computer	Vertical Velocity	400 ms	100-150 ms	0 ms	250 ms
GPS	Horizontal Position	400 ms	100-150 ms	0 ms	250 ms
	Vertical Position	400 ms	100-150 ms	0 ms	250 ms

Table 3.2 Time Delay in GPS Data

DGPS	Horizontal Position	400 ms	100-150 ms	200 ms	50 ms
	Vertical Position	400 ms	100-150 ms	200 ms	50 ms
2 Receivers **2 Computers**	Horizontal Velocity	400 ms	100-200 ms	0 ms	250 ms
	Vertical Velocity	400 ms	100-200 ms	0 ms	250 ms
GPS	Horizontal Position	400 ms	100-200 ms	0 ms	250 ms
	Vertical Position	400 ms	100-200 ms	0 ms	250 ms
DGPS	Horizontal Position	400 ms	100-200 ms	200 ms	50 ms
	Vertical Position	400 ms	100-200 ms	200 ms	50 ms
1 Receiver	Horizontal Velocity	300 ms	50 ms	0 ms	250 ms
	Vertical Velocity	300 ms	50 ms	0 ms	250 ms
	Horizontal Position	400 ms	50 ms	0 ms	300 ms
	Vertical Position	400 ms	50 ms	0 ms	300 ms

Table 3.2 Time Delay in GPS Data

The observed and "unexplained" delays are different if one or two receivers are concerned. But, their origins can be clearly identified:

- Velocity is always computed inside the GPS receivers. Therefore the "unexplained" delay of 250 ms that is always observed is due to computation and setup of the messages within the receivers.
- Position is computed inside the receiver in the "one receiver" case. The "unexplained" delay is therefore due to computation and setup of the messages inside the receiver.
- When 2 receivers are used, position is computed inside the receiver if the test is referred as a GPS test and in the PC104+ stack if it is referred as a DGPS test.

Delay due to computation can be of up to 200ms and is due to communication synchronization.

  - The "unexplained" delay is only on the order of 50 ms for position computation in the DGPS case. In this case, code phase is used to compute the position and there is no time consuming process inside the GPS receiver itself. Because velocity outputs (delayed by 250 ms) are used by the Kalman filters, 200 ms delay are due to the Kalman filters for the DGPS position output.
  - In the GPS case, if Raw data are used, positions are computed inside the GPS receiver and the "unexplained" delays are on the same order as for the one receiver case, that is 250ms.

- When 2 computers are used, GPS data are transferred from one computer to the over one using Maxtream 9XStream Transceivers. No significant delay is added due to the use of the transceivers.

3.4.1.2 Computation of Data in a Local Frame from ECEF Coordinates

In order to reduce the amount of data transmitted by the PCUAV system, the author decided to reduce the GPS data information sent by each vehicle. As the demonstration of reintegration takes place within at most 2 miles of a departure airbase, the author decided to have all the GPS information reduced to a small framework surrounding the airbase. Details of the communication procedure to set this framework are given in Chapter 4, but its specificities (Figure 3.9) are set as follows:

- The center of the framework is the point at which the parent is flying when the program starts. In order to prevent any communication mistake the following process is used:
  - The Parent averages its positions over one second

  - At each step, the Parent compares the result with the former position and makes sure it is within 100m (the speed range of our UAVs is less than 50m/s), or the process is restarted.

- The different axis are given by:
  - The West-East Direction at the center of the new frame
  - The South-North Direction at the center of the new frame
  - The Altitude Direction at the center of the new frame

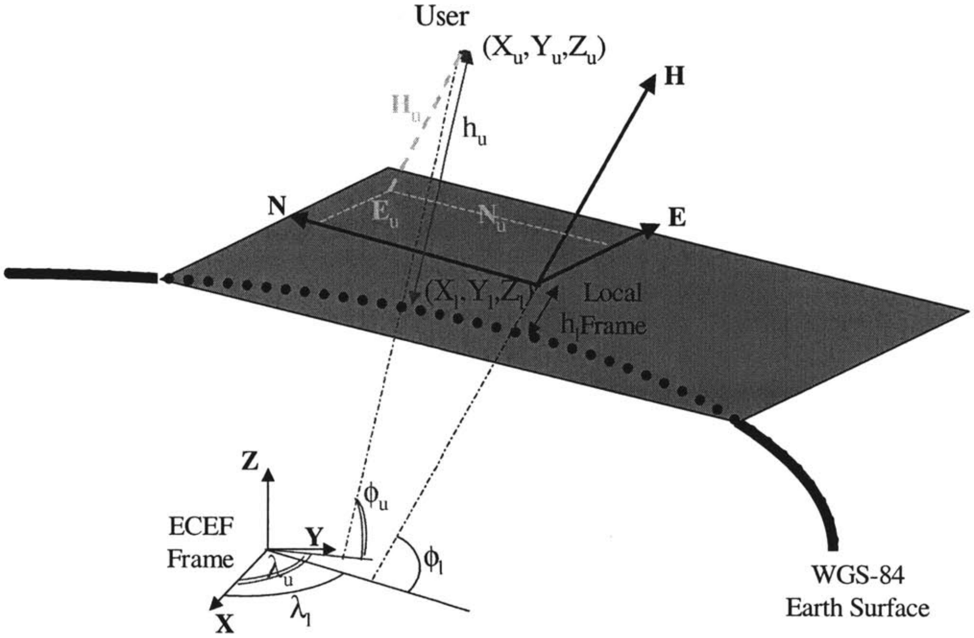


Figure 3.9 Comparison between ECEF, WGS-84, and Local Frame

The first step in order to determine the position in the local frame is to transform the Raw data (X_l,Y_l,Z_l) at a certain time t_0 from ECEF coordinates to WGS-84 coordinates as detailed in Section 3.2.4. This position is then set as the Center of the Local Frame. The base vectors of the Local Frame **(N,E,H)** are also computed in the ECEF Frame.

Then a simple difference between the User's Position at time t (X_u,Y_u,Z_u) in ECEF coordinates and the Center of Local Frame (X_l,Y_l,Z_l) in ECEF coordinates is performed. This difference is then projected on the Local Frame base vectors to give the position in the Local Frame in terms of Local Frame coordinates (N_u,E_u,H_u).

It is in particular interesting to notice that the altitude H_u, in the Local Frame, is different from the Altitude h_u, given by the WGS-84 geodetic coordinates, as Figure 3.9 shows.

3.4.2 Filtering GPS Raw Data

Unfortunately, GPS Raw data do not present smooth characteristics (Figure 3.10) and therefore can not be used directly for guidance and navigation.

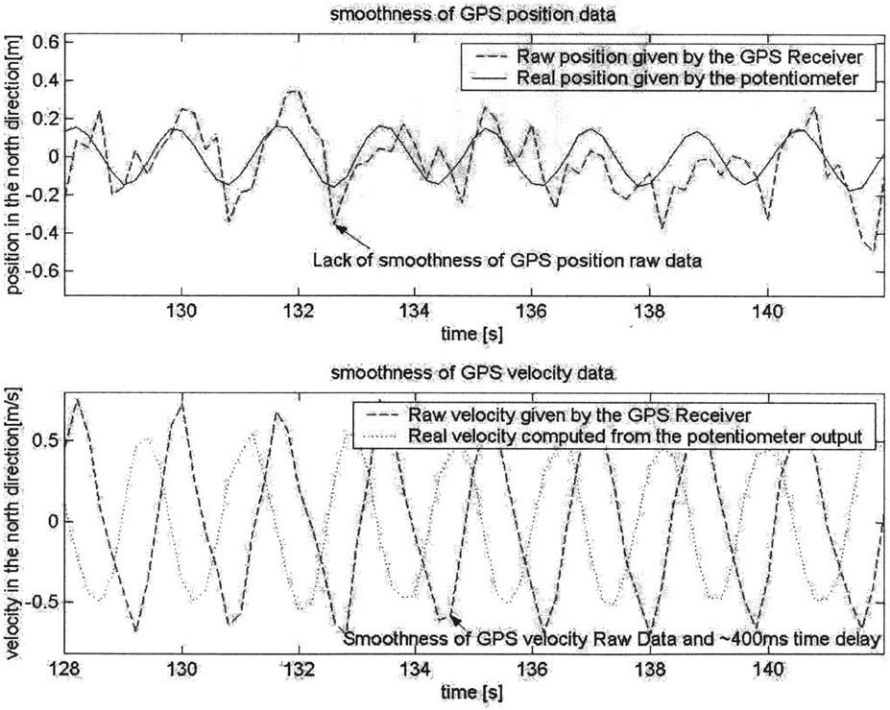


Figure 3.10 GPS Raw Data in a Pendulum Test: Position and Velocity

Therefore, Sanghyuk Park decided to implement a GPS Kalman Filter using velocity and position input based on Richard P. Kornfeld's Ph.D. Thesis [13] and reference [12].

3.4.2.1 Constraints on the Filter

The goal is to get smooth position and velocity estimates that are not delayed. The inputs to the filter are GPS position and velocity and the outputs are estimates of position, velocity and acceleration.

The GPS Kalman filter provides the system with a filtered position and velocity but also with acceleration data. It has given good results in both ground and flight tests using either GPS or DGPS data inputs.

3.4.2.2 GPS Kalman Filter Theory

The system dynamics can be represented by:

$$\dot{\mathbf{x}} = \mathbf{F} \bullet \mathbf{x} + \mathbf{w} \qquad \textit{(eq. 3.10)}$$

where **x** is a column vector of states of the system, **F** the system dynamics matrix and **w** white noise process. In fact, three filters are used, one each for north, east and altitude directions. Each such filter then estimates four states: position, velocity, acceleration and jerk in the direction associated with that filter. **F** is equal to:

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \textit{(eq. 3.11)}$$

The derivative of jerk is modeled as white process noise. Thus

$$E[w(t) \cdot w(t+\tau)] = \mathbf{Q} \cdot \delta(\tau) \qquad \textit{(eq. 3.12)}$$

where the matrix **Q** represents the strength of the process noise and:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{44} \end{bmatrix} \qquad \text{(eq. 3.13)}$$

where q_{44} is chosen as $100\mathrm{m}^2/\mathrm{s}^5$. The Kalman filter formulation also requires a linear formulation of the measurements.

$$\mathbf{z} = \mathbf{H} \bullet \mathbf{x} + \upsilon \qquad \text{(eq. 3.14)}$$

where **z** is the measurement (GPS position and velocity) and ν is white measurement noise.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \text{(eq. 3.15)}$$

The measurement noise matrix R is the covariance of the measurement noise vector:

$$\mathbf{R} = E[\upsilon \bullet \upsilon^T] \qquad \text{(eq. 3.16)}$$

R is chosen diagonal with variances of position and velocity on the diagonal. These values are chosen differently with regards to horizontality or verticality and have been computed thanks to ground tests at Briggs Field and over Building 17 at MIT:

- standard deviation position horizontally = 5.0 m
- standard deviation velocity horizontally = 0.05 m/s
- standard deviation position vertically = 10.0 m
- standard deviation velocity vertically = 0.08 m/s

In case of a lot of multipath and/or the presence of few satellites, standard deviations for position are multiplied by 100. The continuous system of (eq. 3.10) must be discretized before a discrete Kalman Filter can be implemented. As the measurement is taken every T_S seconds with T_S=0.2 (5Hz), the first step consists in finding a state transition matrix Φ

that reflects discretization. Such a matrix is given by a Laplace inverse transform which, after linearization, can be treated as the linearization of the exponential of **Ft**:

$$\Phi(t) = e^{\mathbf{F}t} = \mathbf{I}_4 + \mathbf{F}t + \frac{(\mathbf{F}t)^2}{2!} + \frac{(\mathbf{F}t)^3}{3!} \qquad \textit{(eq. 3.17)}$$

where $\mathbf{I}_4$ is the 4*4 identity matrix. The transition matrix for an interval T_S is then:

$$\Phi_k = \Phi(T_S) = \begin{bmatrix} 1 & T_S & \frac{T_S^2}{2} & \frac{T_S^3}{6} \\ 0 & 1 & T_S & \frac{T_S^2}{2} \\ 0 & 0 & 1 & T_S \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textit{(eq. 3.18)}$$

The Kalman filter is then:

$$\mathbf{x}_k = \Phi_k \cdot \mathbf{x}_{k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\Phi_k\mathbf{x}_{k-1}) \qquad \textit{(eq. 3.19)}$$

where:

$$\begin{aligned} \mathbf{M}_k &= \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k \mathbf{H}^T (\mathbf{H}\mathbf{M}_k\mathbf{H}^T + \mathbf{R}_k)^{-1} \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}) \cdot \mathbf{M}_k \end{aligned} \qquad \textit{(eq. 3.20)}$$

$\mathbf{P}_k$ is the covariance matrix of errors in the state estimates after an update, $\mathbf{M}_k$ is the covariance matrix of errors in the state estimates before the update. The discrete process noise matrix $\mathbf{Q}_k$ can be found from the continuous process noise matrix **Q** and the fundamental matrix according to:

$$Q_k = \int_0^{T_S} \Phi(\tau) Q \Phi(\tau)^T d\tau = 100 \cdot \begin{bmatrix} \frac{T_S{}^7}{252} & \frac{T_S{}^6}{72} & \frac{T_S{}^5}{30} & \frac{T_S{}^4}{24} \\ \frac{T_S{}^6}{72} & \frac{T_S{}^5}{30} & \frac{T_S{}^4}{24} & \frac{T_S{}^3}{6} \\ \frac{T_S{}^5}{30} & \frac{T_S{}^4}{24} & \frac{T_S{}^3}{6} & \frac{T_S{}^2}{2} \\ \frac{T_S{}^4}{24} & \frac{T_S{}^3}{6} & \frac{T_S{}^2}{2} & T_S \end{bmatrix} \qquad (eq.\ 3.21)$$

3.4.2.3 Ground Tests with the Pendulum

Ground tests have been done on top of building 17 and at Briggs Field at MIT with and without moving the GPS receiver's antenna. The team discovered that dynamic tests were much more accurate (and more relevant for our applications) than static tests. It has been noticed that even without a Kalman filter, position and velocity would diverge during a static test but not during dynamic tests. It is believed that these bad results during static tests are due to sudden jumps in the estimation of the code phase of some satellites by the

GPS receiver.

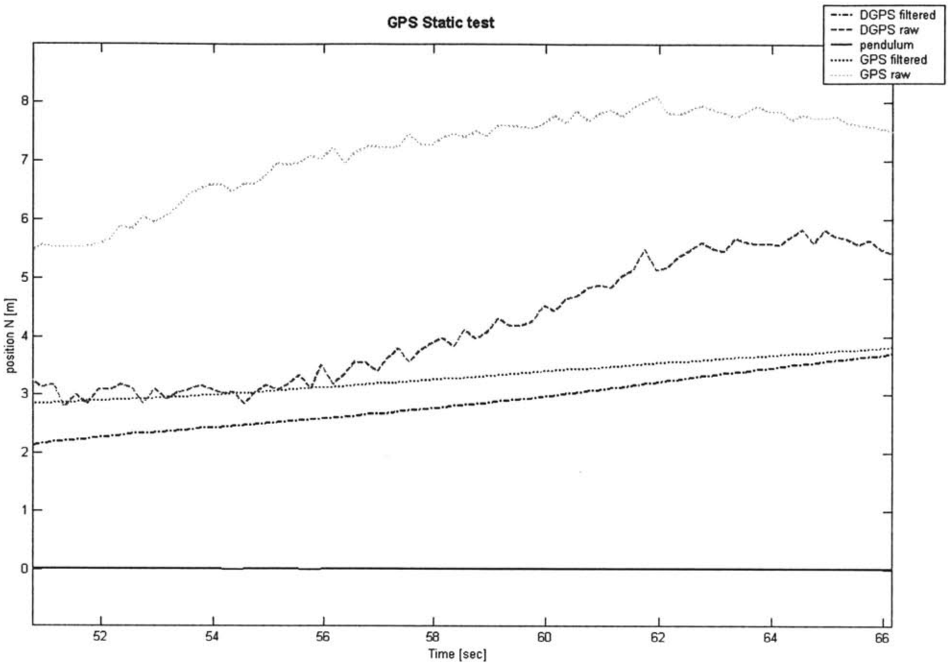


Figure 3.11 GPS Static Tests

Due to this discovery, the team decided to implement a GPS test involving movement and which would give position results that could be compared to known positions. Professor Jonathan How's GPS team at MIT has implemented a pendulum test and offered to let us use their pendulum. However, due to the weight of the pendulum and to its location on top of another building at MIT, the PCUAV team decided to build its own pendulum.

Description of the Pendulum

The pendulum is made of one main arm rotating around a fixed point at which a potentiometer is attached in order to measure the angle of rotation. The pendulum has two positions at which a GPS receiver's antenna can be attached. These positions are set on weighted arms so that the GPS antenna would always point to the sky (Figure 3.12).

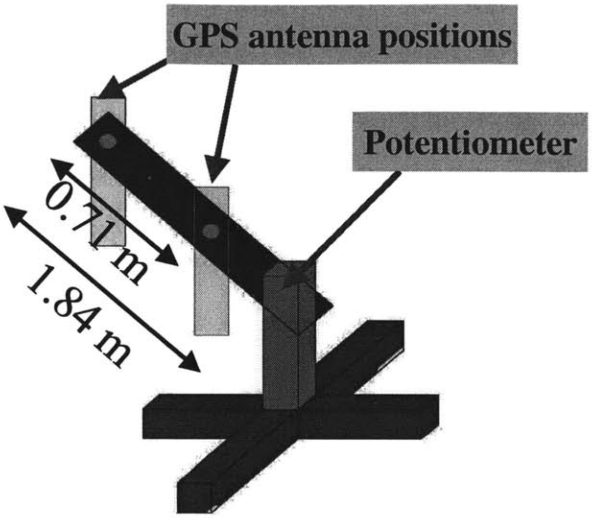


Figure 3.12 Pendulum with two Locations for GPS Receiver

Velocity Concerns and Pendulum Modifications

The first set of results showed that the velocity was not as smooth as expected (Figure 3.13). This was due to pendulum movements of the arms on which the receivers were mounted and can be identified at the maxima and minima of each cycle.

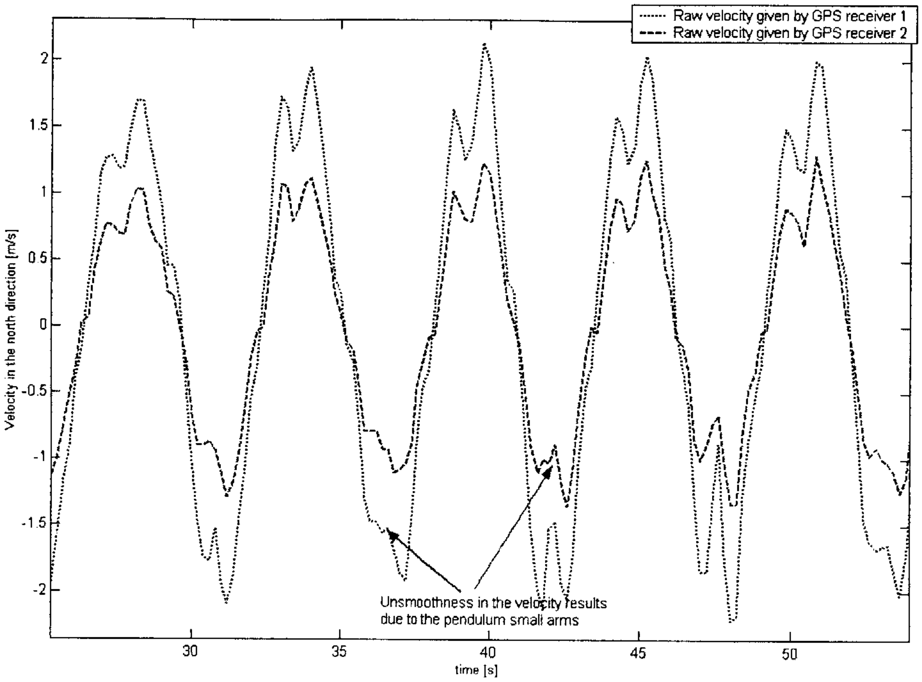


Figure 3.13 First Pendulum Velocity Results

In order to get rid of these movements, the arms on which the antennas are mount have been attached to the main arm (Figure 3.14). The angles of rotation of the main arm have to be maintained small enough so that at least five satellites are still tracked at the maximum angle.

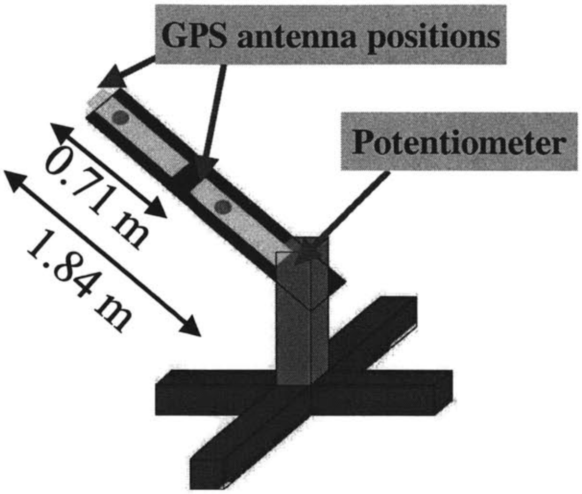


Figure 3.14 Pendulum Modifications

Test Results

The tests proved the smoothness of the velocity. The accuracy of the absolute position in the WGS-84 frame of reference is not tested. The main purpose of the test is to demonstrate that the filtered data would remain within a certain range within time. This range has been estimated to be of:

- 5 meters horizontally within 1 minute at 95%.

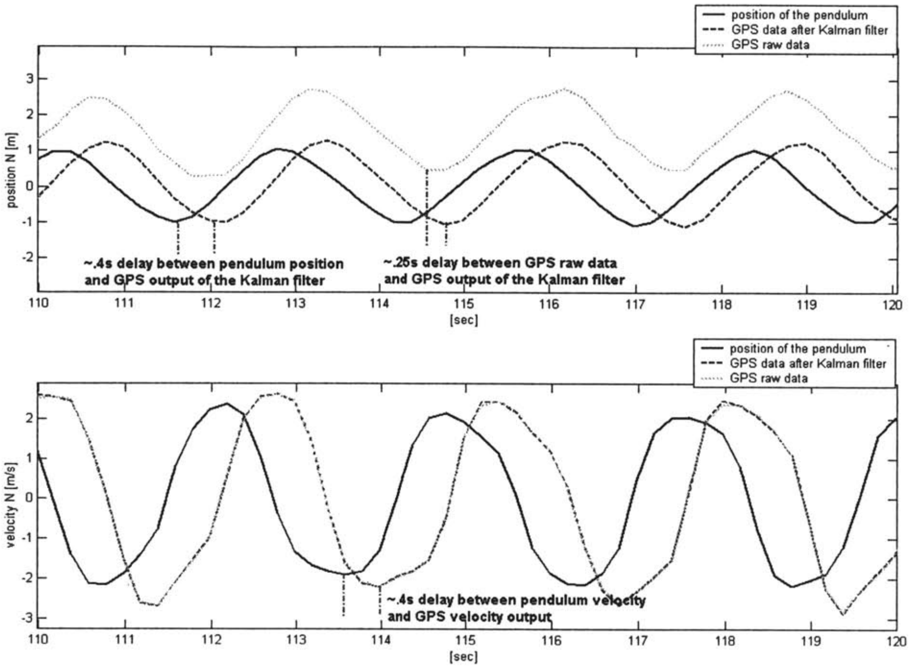


Figure 3.15 Horizontal Position and Velocity in a GPS Pendulum Test

Figure 3.15 presents the Raw data, the output of the Kalman filter, and the estimated horizontal position and velocity from the potentiometer output, for a pendulum test with one GPS receiver and 6 satellites tracked. As expected:

- Raw and filtered positions and velocities are delayed by respectively 150 ms and 400ms compared to the potentiometer data.
- Raw and filtered velocities are almost equal.
- Over 20 seconds, the absolute error on GPS raw data does not increase by more than 50cm, and the error on absolute GPS filtered data does not increase by more than 20 cm.

- 10 meters vertically within 1 minute at 95%

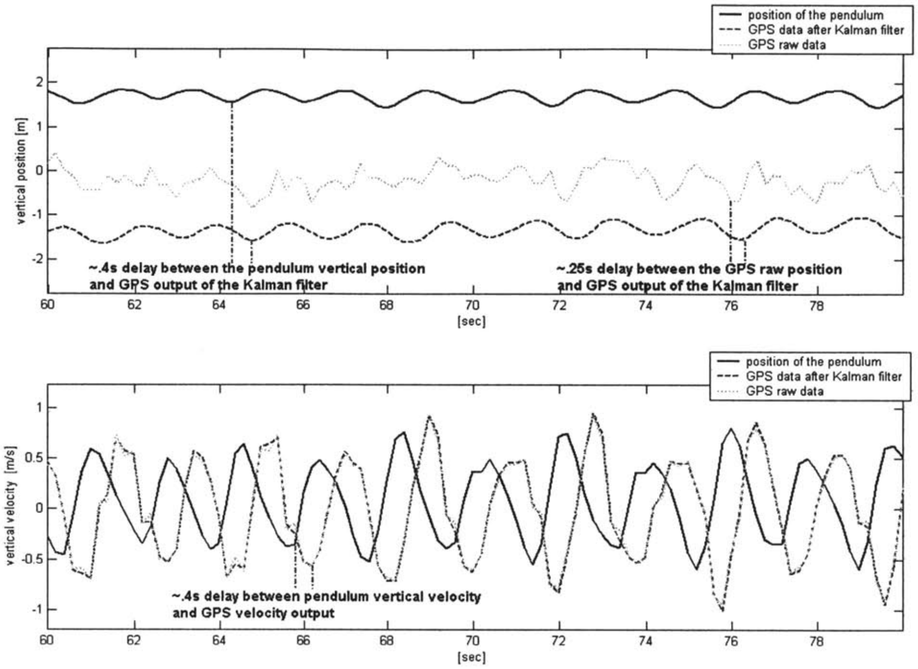


Figure 3.16 Vertical Position and Velocity in a GPS Pendulum Test

Figure 3.16 presents the Raw data, the output of the Kalman filter, and the estimated vertical position and velocity from the potentiometer output, for a Pendulum test with one GPS receiver and 6 satellites tracked. As expected:

- Raw and filtered positions and velocities are delayed by respectively 150 ms and 400ms compared to the potentiometer data.
- Raw and filtered velocities are almost equal.
- Over 20 seconds, the absolute error on GPS raw data does not increase by more than 3m, and the error on absolute GPS filtered data does not increase by more than 50cm.

3.4.3 Flight Tests

Flight tests have been performed in order to demonstrate the possibilities of autonomous flight and to make sure that the path designed for reintegration could be followed. Tests have been performed at Medfield and at Shirley airport.

3.4.3.1 Preparation of Flight Tests

In order to avoid any unexpected event once at the field, ground tests are always performed before any flight tests. During these ground tests, the airplane is set as if it would fly. In particular batteries are incorporated. The main difference with flight tests (beside the fact that the UAV stays on the ground) is that the engine is most of the time not ran during these tests. The plane is driven around the field and the different options of the flights are tested. The main goal is to make sure that the computer program does not have any bug. Most bugs can be found in the lab with hardware in the loop simulations [15], but unexpected situations can always occur, and the closer the configuration can be to the flight configuration, the better chances the vehicles have to perform their expected tasks.

In particular, these tests have demonstrated that the position of the GPS receiver's antenna had an influence on the number of satellites tracked. The best position for the antenna is over the Center of Gravity (CG) to simplify the navigation controller's code. Unfortunately, the onboard computer, which produces considerable electro-magnetic interference, is also near the CG on the ATA. The GPS antenna is sensitive to magnetic effects and must therefore be separated as much as possible from the onboard computer. Tests have identified unacceptable positions for the antenna (which would induce a loss of the number of satellites). The GPS antenna has been set on top of the vertical stabilizer on the Mini Vehicle (Figure 3.17).

Figure 3.17 Position of the GPS Antenna on top of the Mini Vehicle

These tests have also revealed that the GPS Receiver would stop tracking satellites if too much acceleration is experienced. This situation can occur during flight tests when the pilot is in control of the airplane. In particular, it can occur during steep turns. It is also believed that loss in the number of satellites is due to multipath and from noise due to the engine for the ATA. However, in the pusher configuration of the Mini, noise from the engine is significantly reduced. For the Parent, the antenna is located on top of the MPIM and therefore far away from the engine and the onboard computer. No loss of satellites tracked has been observed with the Mini and the Parent

3.4.3.2 Description of Flight Tests for Autonomous Flights

Two sets of autonomous flight tests involving only one vehicle have been performed at

Medfield and Shirley Airport with the ATA, Mini and Parent:

- A flight along a circle (to simulate the parent's trajectory)
- A flight simulating the Mini's Trajectory during Phase I.

Autonomous Flight Around a Circle

For this flight, the onboard computer is turned on after the UAV is airborne. When the program starts, the actual position of the Parent is set as the center of the local frame. The Computer in Control (CIC) Mode (autonomous navigation) is engaged when the ground team decides that conditions are appropriate. Such a decision is based on the following criteria:

- Communication between the ground and the UAV is established
- The UAV's GPS receiver is tracking enough satellites (five or more, for more than 3 seconds)
- Successful calibration of onboard inertial sensors has been accomplished
- The Copilot estimates that the airplane's attitude is appropriate to engage the CIC mode
- The Pilot estimates that the flight is safe and that recovery can be performed in case of CIC failure

At the beginning of the CIC mode, the UAV computes the circle trajectory by plotting the center of the circle at its actual altitude on its left hand side. The circle has a 250 meter radius so that the Parent's bank angle is small enough to stay in linearity of the control system model [15]. The circle is also as small as possible, so that the airplane remains within an acceptable range from the pilot. It gives the following forecasted trajectory:

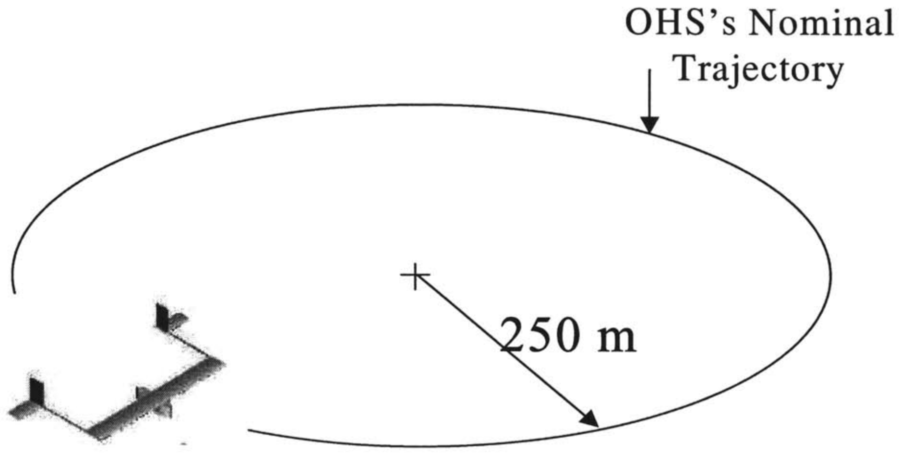


Figure 3.18 Parent's Nominal Trajectory

Phase I Autonomous Flight Test

In this configuration, the pilot has to fly over a designated point. Once the UAV reaches this point, the operator of the ground station's laptop sets it as the center of the local frame. This point is set so that it represents the center of the circle around which the parent would circle anticlockwise. Then the vehicle is flown outside of this circle. The position and forecasted trajectory can be followed on the ground station laptop. The Computer in Control (CIC) mode is only engaged if the expected trajectory is not too long.

At the beginning of the CIC mode, the UAV computes its trajectory (Figure 3.19). The trajectory is especially designed for this test by removing the synchronization option present in the test involving two vehicles. The trajectory is followed thanks to proportional navigation.

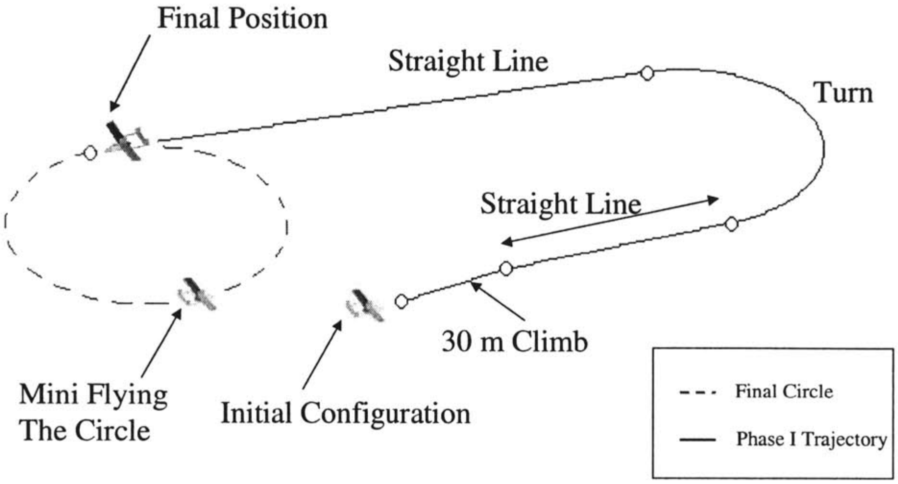


Figure 3.19 Phase I Test Nominal Trajectory

3.4.3.3 Results of Flight Tests

The tests performed with the ATA, the Mini and the Parent were very satisfactory for autonomous flight. A good design of the control system that couples the GPS output with an inertial navigation system could handle the existing inaccuracy of a single GPS receiver, even for the vertical controllers. During these flight tests many satellites (6 to 8 most of the time) were tracked.

The Parent circle path could be followed as shown in Figure 3.20.

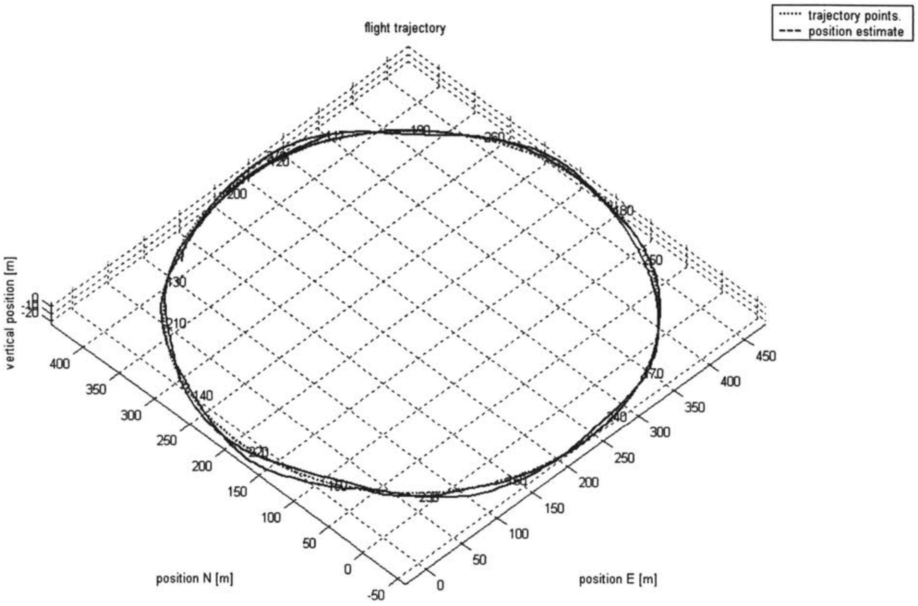


Figure 3.20 Circle Trajectory Results

The Phase I Trajectory could be followed as shown in Figure 3.21 where trajectory points linked to proportional navigation and the estimate of the position are shown. The different steps (including climb, straight line, turn, straight line and then turn around the parent's circle) were performed efficiently.

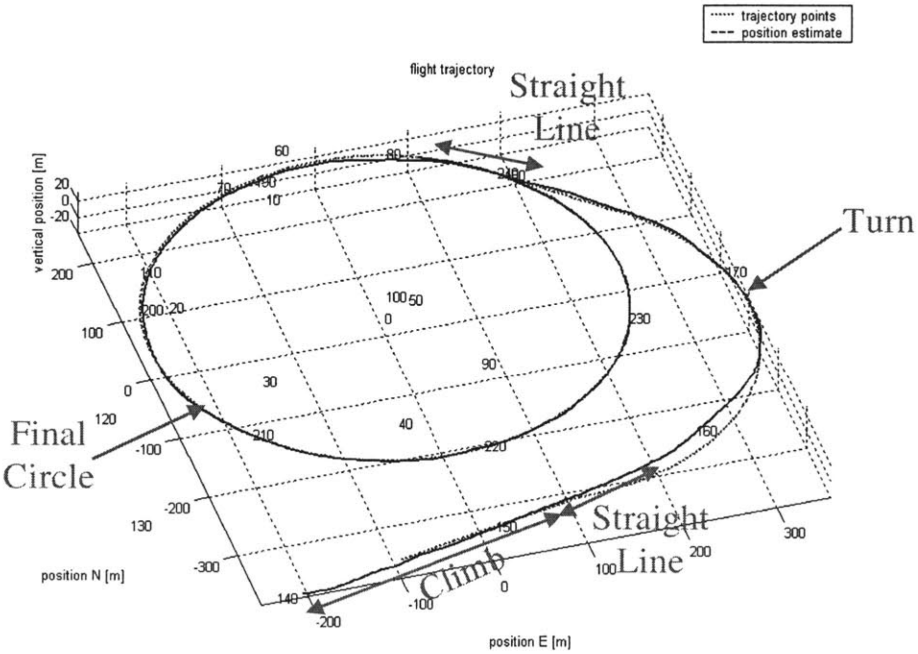


Figure 3.21 Phase I Trajectory Results

3.5 Differential GPS

Differential GPS (DGPS) involves at least two vehicles in the PCUAV Project. Due to the possibilities of operations in unfriendly contexts, the PCUAV team decided not to rely on systems such as WAAS. Moreover, as the only accuracy required is for a vehicle with respect to another vehicle, the team decided to create a system that would only involve GPS receivers onboard vehicles. Different concepts have been implemented in order to reach the expected accuracy requirements. Most of them are discussed within this section.

3.5.1 Difference of GPS Position

The difference of GPS position concept models the difference of positions between the Parent and the Mini. The team first believed that such an approach would provide the expected accuracy (less than 5 meters in all directions). The results obtained were close to the expected values but did not reach them as it will be shown in Section 3.5.1.3.

3.5.1.1 Difference of GPS Position Concept

In this situation the Parent is controlled using filtered GPS Raw data. The GPS Raw data (unfiltered) from the parent is also sent to the Mini. The Mini's onboard Computer computes the difference between its own GPS Raw data and the Parent's GPS Raw data. The result of this difference is then filtered in the Mini's computer and subtracted from the Parent's filtered data, which is also transmitted from the Parent to the Mini (Figure 3.22).

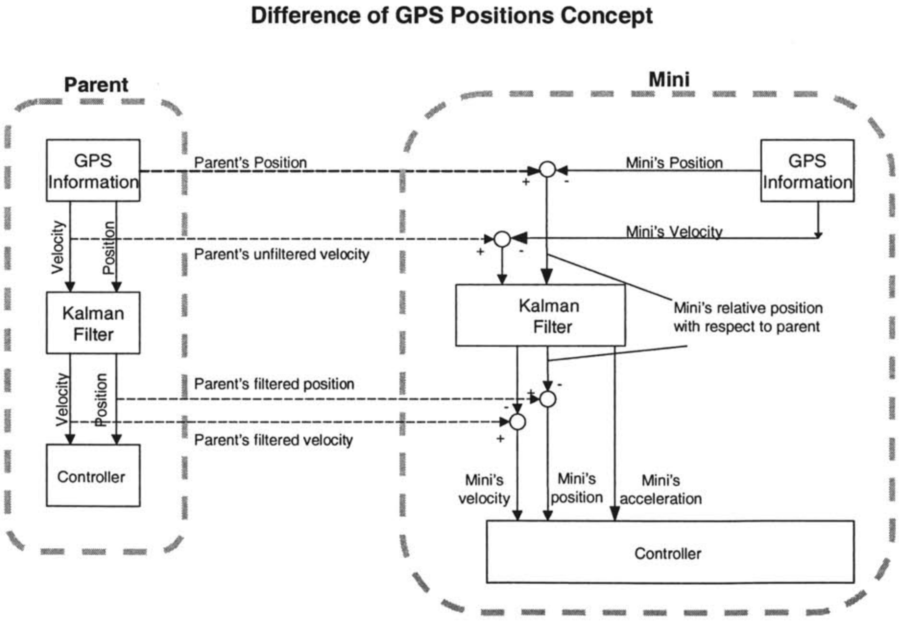


Figure 3.22 Architecture of the Difference in GPS Positions Concept

This concept is based on an easy way to build a DGPS system without too much communication and computation. One-way transmission of GPS Raw data and filtered estimates from Parent to Mini, and one Kalman filter onboard the Mini provides it with filtered absolute and relative (to the Parent) position and velocity. The expected accuracy of 7 meters horizontally and 10 meters vertically is on the difference of positions:

$$\mathbf{dX}_{MP} = X_M - X_P \qquad (eq.\ 3.22)$$

The accuracy is obtained as the errors in the absolute position of two GPS receivers close to each other (within a 2 kms range) have the same origins, and therefore, the effects are similar for each GPS signal received. Such errors include atmospheric errors, geometric architecture of the GPS satellites network over the user, or multipath.

The data used for the mini navigation are just recomposed and filtered data:

$$X_{FM} = \mathrm{Filter}(\mathbf{dX}_{MP}) + X_{FP} \qquad (eq.\ 3.23)$$

where the Filter() application alludes to the GPS Kalman Filter.

As it can be seen from (eq. 3.23), if data did not have to be filtered for navigation, no computation would be necessary for the Mini to have an accurate position relatively to the parent and navigation could be performed with the Mini's position. This method is recommended if the expected accuracy is on the order of 10 meters horizontally and vertically.

3.5.1.2 Difference in GPS Position Final Architecture

Based on the architecture developed in Section 3.5.1.1 and on the fact that the Parent's position is mostly used by the Mini for trajectory planning during Phase I, the architecture of the difference in position can be simplified as presented in Figure 3.23:

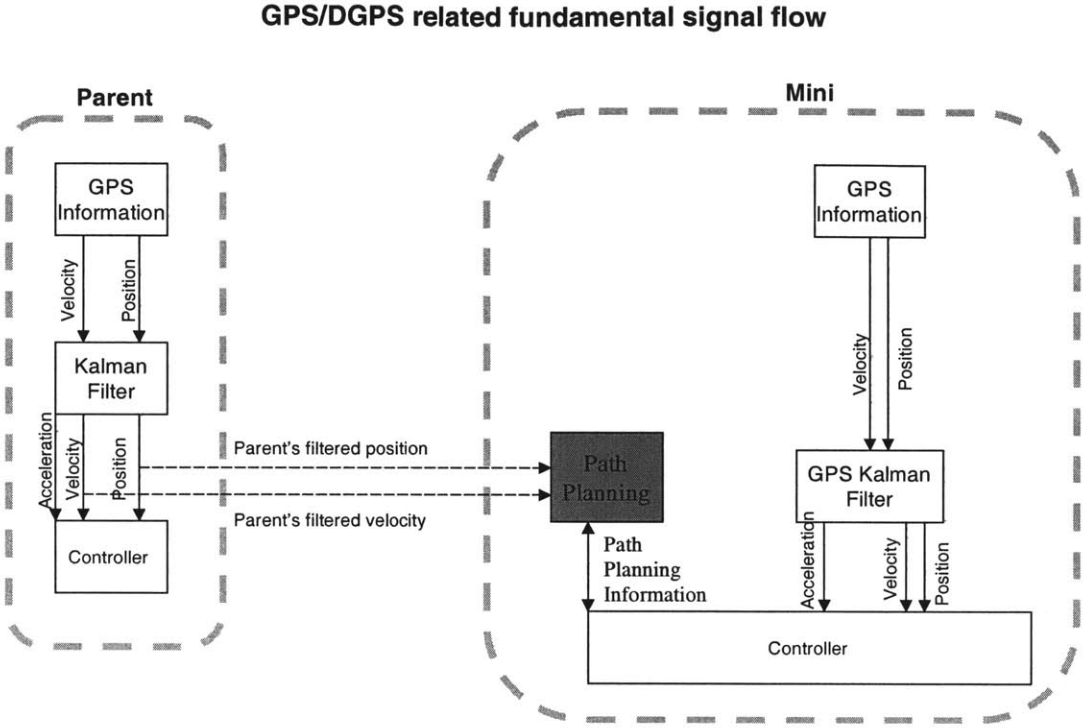


Figure 3.23 Difference in GPS Position Reduced Architecture

This architecture has been the first one tested to demonstrate Phase I with the Mini's final position 30 meters behind the Parent.

3.5.1.3 Results of the Ground Tests

The relative accuracy obtained in ground tests was 7 meters horizontally and 10 meters vertically with 95% probability. The team then believed that a major part of the error was due to computation and decided to implement a differential GPS (DGPS) system between both vehicles.

3.5.1.4 Results of Phase I tests

Due to bad weather conditions in the Boston area during the month of may 2002, Phase I flight test could not be performed at the time this Master's Thesis was written.

3.5.2 Differential GPS (DGPS)

The idea on which Code Phase Differential GPS is based is the same one as for the difference in GPS position concept, which is to get rid of the code phase noise. The solution developed is more accurate as computation is performed only once (compared to twice for the Difference in GPS Position system).

3.5.2.1 DGPS Concept

The equation used to solve the position of one single GPS receiver can be derived from [2]:

$$\Pi_i = H_i \cdot X_u + (t_u - t_0) + \varepsilon_i + N_e \qquad \textit{(eq. 3.24)}$$

where Π_i is the satellite code phase, H_i the line of sight from the receiver to the satellite, X_u the position of the GPS receiver, t_u time since a certain time origin t_0, ε_i the error in the observation of the pseudo-range, and N_e the error due to linearization. When looking at a set of satellites, the equation can be written with vectors as:

$$\Pi = \mathbf{H} \cdot \mathbf{X} + (\mathbf{t} - \mathbf{t_0}) + \varepsilon + \mathbf{N_e} \qquad \textit{(eq. 3.25)}$$

or, if **I** is the column vector with only ones:

$$\Pi = \begin{bmatrix} \mathbf{H} & \mathbf{I} \end{bmatrix} \bullet \begin{bmatrix} \mathbf{X} \\ t - t_0 \end{bmatrix} + \varepsilon + \mathbf{N_e} \qquad \textit{(eq. 3.26)}$$

Such an equation can be solved for two receivers. Each result will be biased due to the code phase noise and to errors in computation. Subtracting both results gives the difference in GPS position concept. As code phase noise are similar for each receiver in the same area, the result is accurate. However, noise due to computation can be added resulting in an inappropriate error.

In order to get rid of these errors the following equation can be derived when two receivers are involved and are close enough to each other so that the line of sight to each satellite is the same (or very similar as the satellites are far away):

$$\Pi_1 - \Pi_2 = \begin{bmatrix} \mathbf{H} & \mathbf{I} \end{bmatrix} \bullet \begin{bmatrix} \mathbf{X}_1 - \mathbf{X}_2 \\ t_1 - t_2 \end{bmatrix} + \delta\varepsilon + \mathbf{N_e} \qquad \textit{(eq. 3.27)}$$

which by neglecting δε compared to N becomes:

$$\Delta\Pi = \begin{bmatrix} \mathbf{H} & \mathbf{I} \end{bmatrix} \bullet \begin{bmatrix} \Delta\mathbf{X} \\ \Delta t \end{bmatrix} + \mathbf{N_e} \qquad \textit{(eq. 3.28)}$$

As **H** is known from a receiver's position and the satellites' positions (given by the ephemeris data), (eq. 3.28) can be solved as long as there are more than 4 satellites tracked.

In order to solve this equation when more than four satellites are involved, a least square method combined with a Gauss inversion is used and the following set of equations is used:

$$\begin{bmatrix} \mathbf{H}^T \\ \mathbf{I}^T \end{bmatrix} \bullet \Delta\Pi = \begin{bmatrix} \mathbf{H}^T \\ \mathbf{I}^T \end{bmatrix} \bullet \begin{bmatrix} \mathbf{H} & \mathbf{I} \end{bmatrix} \bullet \begin{bmatrix} \Delta\mathbf{X} \\ \Delta t \end{bmatrix} \qquad \textit{(eq. 3.29)}$$

$$\begin{bmatrix} \Delta\mathbf{X} \\ \Delta t \end{bmatrix} = \left(\begin{bmatrix} \mathbf{H}^T \\ \mathbf{I}^T \end{bmatrix} \bullet \begin{bmatrix} \mathbf{H} & \mathbf{I} \end{bmatrix} \right)^{-1} \bullet \begin{bmatrix} \mathbf{H}^T \\ \mathbf{I}^T \end{bmatrix} \bullet \Delta\Pi \qquad \textit{(eq. 3.30)}$$

3.5.2.2 DGPS Kalman Filter

A Kalman filter similar to the one described in Section 3.4.2.2 has been implemented at the output of the DGPS computation. This filter prevents errors from occuring in the reception or transmission of code phase data from one or another vehicle. The inputs of the filter are the DGPS data (position and velocity).

In order to validate the DGPS code, the position obtained is compared with the GPS position of the mini filtered as explained in Section 3.4. The output of a GPS method is also used for the Mini in order to provide the control system with appropriate accelerometer inputs.

3.5.2.3 Pendulum Ground Tests

A set of ground tests has been performed in order to validate the DGPS code.

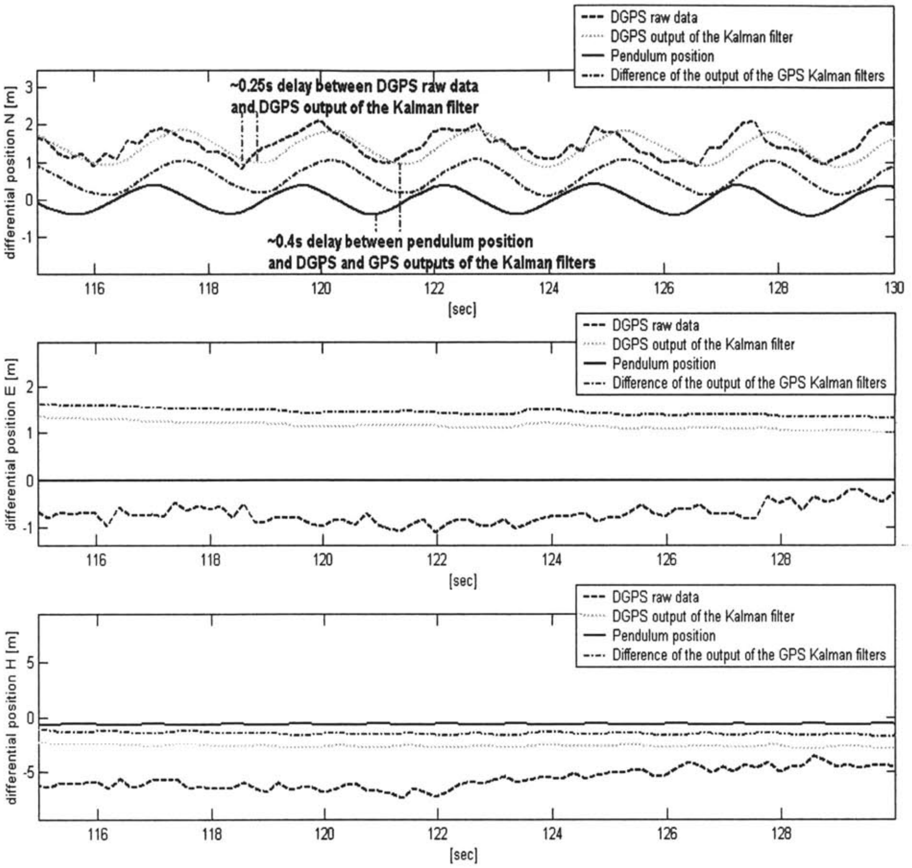


Figure 3.24 DGPS Pendulum test Results

It can be observed that the time delay between the pendulum and the GPS data is 400ms as it has been explained in Section 3.4.1.1. The tests helped figuring out the impact of the DGPS Kalman filter on data. As one might expect, it mostly prevents the results from diverging, but it also smooths them if an error occurs. However, the impact is not as clear as for one single GPS signal. These results can be compared to the difference in position results. No significant delay between the difference in position and DGPS computation can be identified.

As a result of our tests, the accuracy of position (within one minute with probability 95%) thanks to DGPS is:

- 2 meters horizontally
- 4 meters vertically.

3.5.2.4 Flight Tests with DGPS

Flight tests using DGPS information have not yet been performed at the time this Master's Thesis was written.

3.5.3 Combination of GPS and DGPS information

Using DGPS improves the accuracy of GPS data information but requires more information to be sent from one vehicle to the other. Moreover, if two vehicles are not tracking the same satellites or if they are too far away from each other, DGPS information might be either irrelevant or even impossible to compute. Therefore, a strategy had to be established in order to determine the most accurate piece of information to be used:

- if one vehicle is tracking strictly less than 5 satellites:
    - if position is updated inside the GPS receiver (at least 4 satellites), this information is used if autonomous flight is performed using GPS, but the Kalman filter noise standard deviation is multiplied by 100. If this situation lasts for more than 1 second, Phase I is aborted.
    - if the position is not updated, Phase I is aborted.
- if one vehicle is tracking at least 5 satellites:
    - if the vehicles are using GPS information, a warning is sent to the ground station but Phase I continues to be performed

- if the Mini is using DGPS information and receives code phase information from at least 5 same satellites from the parent and the mini, navigation is performed using DGPS. If this number becomes less than 5 (and GPS information is still computed using 5 satellites inside the Mini's GPS receiver), then the navigation of the Mini is performed using GPS information. The switch is performed by using the difference of position between the output used for navigation (most likely DGPS) and the GPS output at the time (t_0) of satellite drop: **Distance**(t_0). This difference is then multiplied by the function of time DGPStoGPS() and subtracted to the GPS information. DGPStoGPS() performs as follows:

$$\mathrm{DGPStoGPS}(t) = \begin{cases} 0 & \text{if } t > t_0 + 1 \\ 1 - (t - t_0) & \text{if } t_0 \le t \le t_0 + 1 \\ 1 & \text{if } t < t_0 \end{cases} \qquad \textit{(eq. 3.31)}$$

- if the Mini is supposed to use DGPS information, but is using GPS information because of a former low number of satellites, and has recovered at least 5 code phase pieces of information from the same satellites for the Mini and the Parent for at least 1 second, it switches back to DGPS at time t_0 by using the difference between the used information and the newly computed DGPS information: **Distance**(t). This difference (that varies with time) is multiplied by the function GPStoDGPS() and is subtracted to DGPS information:

$$\mathrm{GPStoDGPS}(t) = \begin{cases} 0 & \text{if } t > t_0 + 1 \\ 1 - (t - t_0) & \text{if } t_0 \le t \le t_0 + 1 \\ 1 & \text{if } t < t_0 \end{cases} \qquad \textit{(eq. 3.32)}$$

3.5.4 Communication Requirements induced by DGPS Constraints

The code for DGPS implies communication of a certain amount of data from the Parent to the Mini:

- Parent's code phase for the DGPS code
- Parent's Raw velocity in all directions for the DGPS Kalman filter.

The team has also decided to send the Parent's filtered data in order to compute the Mini's position in the local frame, which involves:

- Parent's filtered position
- Parent's filtered velocity

The flow of information is summarized below:

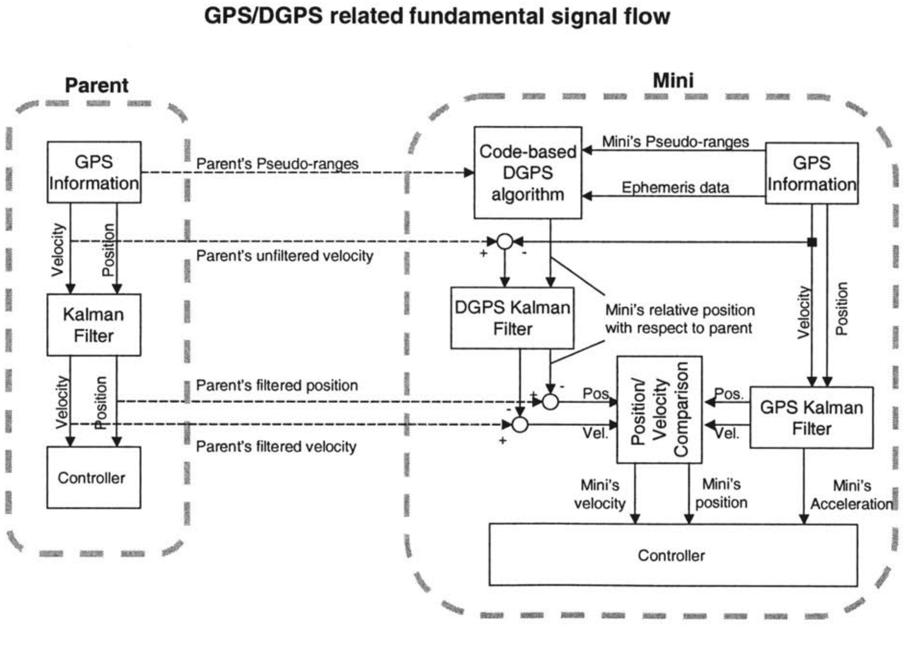


Figure 3.25 Flow of Information to Perform DGPS Navigation

3.6 Chapter Summary

This chapter presented the different approaches used by the PCUAV team to deal with

GPS data. It implies the use of a Kalman filter in order to perform navigation with one single receiver and a Code Phase DGPS code to get more accuracy in the difference of position between two vehicles.

It also gave the results of different tests performed on the ground and in the air and introduced different requirements for communicating data from one vehicle to another one.

Chapter
4

Communication

4.1 Chapter Overview

This chapter presents an in-depth discussion of the communication aspects of the PCUAV project involved in the navigation process. The transceiver the team decided to use based on the requirements presented in Chapter 2 and Chapter 3 is first introduced. Then serial port communication is discussed, followed by the presentation of data compression and reduction for the system.

4.2 Communication requirements

The communication requirements are closely linked to the GPS accuracy requirements and have therefore changed as the project was evolving. The main task is to provide data from the Parent to the Mini to enable computation of the Mini's trajectory depending on the Parent's trajectory.

4.2.1 First Set of Requirements: Parent-Mini-Ground Communication

The first reintegration concept developed by the team implies the Parent is following a circular trajectory. At the beginning of Phase I, the Parent sends its position to the Mini. From that point on, both vehicles know what path to follow and communication is not necessary until the beginning of Phase II (Vision system). As a matter of fact, the difference in GPS position concept does not require continuous knowledge of both positions if the two trajectories are followed precisely.

4.2.1.1 The Overall Communication Concept

In order to make sure that nothing goes wrong in the air, the team decided to follow the position of both vehicles on a ground station laptop. Therefore communication involves:

- The Parent sending its position to the Mini and the Ground station
- The Mini sending its position to the Parent and the Ground station: Figure 4.1 (for the final project, the Mini's position would be sent to the ground station via the Parent: Figure 4.2)

In addition, the team has decided to maintain control over several variables from the ground station, which implies communication from the Ground station to the Parent. Most of the time, any order from the ground station to the Parent implies an action, and the Parent retransmits the order to the Mini. If an order from the Ground station has to be performed without delay, it can be sent directly to the Mini. Therefore, the communication paths can be presented as follows:

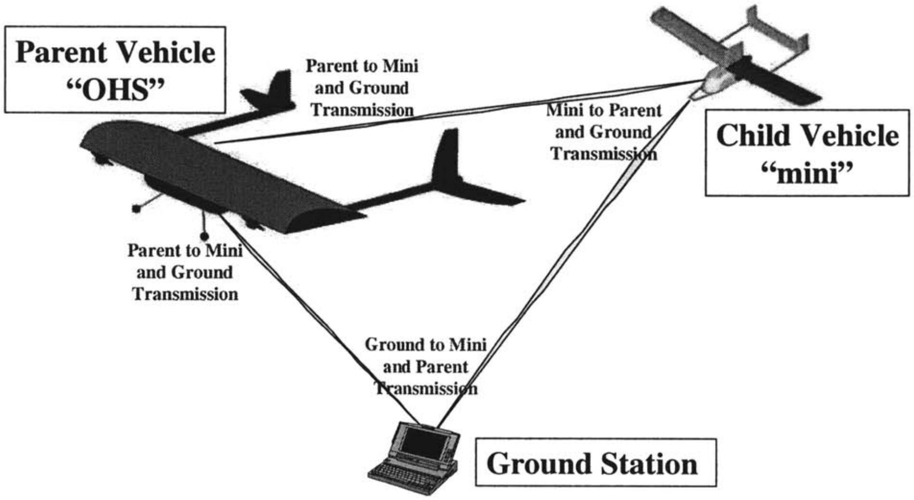


Figure 4.1 Demonstration Communication Channels within the PCUAV Fleet

This communication has to be performed at 5 Hz.

For a final project with UAVs flying far away from the home base, all communications between the ground station and the UAVs would transit via the Parent, which would then transmit to the ground station its own information and the mini's information (Figure 4.2). For demonstration purposes, in order to have less data in the air, less delay for the display of the Mini's position on the Ground station, and the ability to track Mini's data if the Parent is not present, this feature has not been implemented.

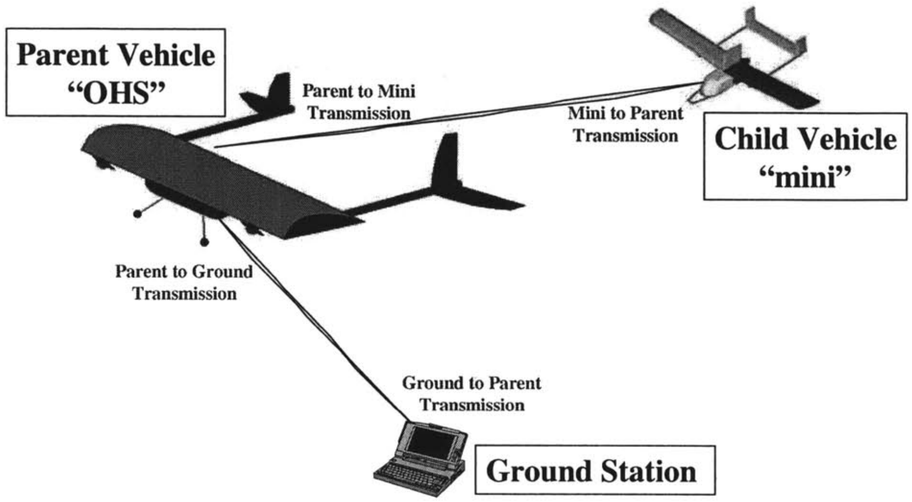


Figure 4.2 Final Project Communication Channels within the PCUAV Fleet

4.2.1.2 The Interference Problem: The Token Ring like Protocol

With such a system, a solution had to be found to avoid interference. The solution has been developed around a Token Ring like procedure: during a 200ms lapse time (communication has to be performed at 5Hz -as GPS and DGPS data are computed at 5Hz), communication would be performed as follows:

- The Parent would send its information when it has received all its GPS information from the GPS receiver (which is aligned on GPS Time)
- Once the Mini has received the Parent's Data (and its own GPS data which arrive at the same time as the Parent's one -as the Mini's GPS receiver is aligned on GPS Time too), it sends its own data
- Once the Ground station has received data from the Mini, it sends data if it has data to send. An option can be set on the Ground Station in the case where only one vehicle is present to enable transmission after reception of the Parent's data.

Another option can be set to enable transmission at any time if the data to be sent is less than 16 bytes long.

The receiver used creates some delay between each operation of this process. Such a delay will be discussed in Section 4.3.

4.2.2 Second Set of Requirements: DGPS data

As developed in Section 3.5.4, the DGPS code places a lot of constraints on the communication process. In particular, the following data must be sent at a rate of 5Hz from the Mini to the Parent:

- Time of GPS data reception
- Satellite number and code phase for each satellite
- Raw velocity in all directions
- Filtered position
- Filtered velocity

4.2.2.1 The Parent's Bandwidth

The Raw velocity requires 4 bytes per axis, as does the filtered velocity. The filtered position is computed so that it can be sent using 4 bytes per axis. Time is computed so that it can be sent using 2 bytes. The total is 34 bytes.

The code phase is coded using 4 bytes, the satellite number using 1 byte, which makes a total of 5 bytes per satellite. With a maximum of 12 satellites, the total amount of data transmitted can go up to 60 bytes (it is usually 40 to 45 bytes with 8 to 9 satellites actually tracked). In addition, 2 bytes are added to make sure that time is correct and 1 byte is sent to give the number of satellites.

During each process, the Parent would send a total amount of data in the range 40 to 100 bytes, with a high probability to be within the range of 57 to 82 bytes (4 to 9 satellites tracked).

At the same time, the Mini would send 26 bytes including filtered position, velocity and time.

4.2.2.2 The Local Frame

In order to reduce the amount of data sent for the filtered position, it was decided to work in the local frame as presented in Section 3.4.1.2. The major requirement for this local frame is to make sure that both vehicles have the same center of the frame of reference in the ECEF coordinates. To make sure that all the vehicles have the same position of reference, the following protocol has been developed for the Parent:

- When the Parent's computer is turned on, it sets its position as the position of reference with the procedure discussed in Section 3.4.1.2.
- Once the position is set, it is sent to the Mini and the Ground Station.

The Mini's protocol works as follows if the Parent is also present in the air:

- It asks the Parent for the position of reference and keeps on requesting it at 5Hz until reception. Upon reception, the message is acknowledged with the whole received information retransmitted. Therefore, if the Parent's computer is not on, the Mini cannot fly autonomously. This is not a constraint, as the Mini needs to know the Parent's position and trajectory for its own trajectory to be decided. However, if the Mini is alone in the air, it can set up its own position of reference.
- Once the Parent has received the request from the Mini to send the position of reference, it sends it and waits for the Mini acknowledgement and sends the information again, as long as the acknowledgement has not been received. Upon reception

of the acknowledgement, the Parent checks the compatibility with its own position of reference and sends the information again if necessary.

The Ground station is also involved in the position of reference setting. The position of reference is first set when the Parent's code starts running. As this position is the center around which the autonomous Phase I test trajectory is based, the ground station has been given some control over its setting:

- The Ground Station can ask the Parent to send the position of reference.
- The Ground Station can give the Parent the order to set its current position as the position of reference at any time if the Parent is not in Computer in Control (CIC) mode:
    - The Parent would reset the position of reference by averaging its positions over 1 second.
    - Then the Parent sends the new position of reference.

In this process, the position sent is in the ECEF format and requires 8 bytes per axis, which makes a total of 24 bytes.

4.2.3 Third Set of Requirements: Optional Data

Mostly for safety reasons, it is necessary to send additional information. This includes optional messages that are sent only once (like different pieces of information sent after the trajectory design), data of particular interest (i.e. velocity relative to the air), or data that can be changed from the ground station (i.e. the fact of being in Computer in Control -Autonomous flight- or Pilot in Control).

The total amount of information sent this way can grow rapidly so it has always been a challenge to reduce the amount of such information (Section 4.5). The pieces of information used are:

- for all vehicles, at 5Hz:
    - Air Speed
    - Air Speed Command
    - Ground Speed Command
    - State of Computer/Pilot in Control
- for the Parent:
    - position of the center of its circle sent when autonomous flight start and until acknowledgement from the mini
    - upon request, the IMU information can be sent to the ground station at 5Hz. Such information contains the accelerometer and gyro outputs.
- for the Mini:
    - Mini's trajectory points are sent at 5Hz to ensure the display of the Mini's forecasted trajectory on the ground station.

4.2.4 Structure of the Communication Signal

The author decided to use the same protocol as the one used for the GPS receiver (regardless of the one used for GPS). As the Binary protocol (Section 3.3.2) is used for GPS messages, the communication messages structure and recognition procedure presents the same characteristics as the one presented in Section 3.3.2.

4.2.4.1 Message Block Structure

The main difference with the GPS receiver's block messages is the Start of Header (SOH) byte which identifies the transmitter of the message. The SOH corresponds to byte one of the communication messages and the following hexadecimal codes are used for the different components of the system:

- Mini: 23h
- Parent: 24h
- Ground station: 25h

On the receiver side, these SOH identify the sender of the message. The next byte of the messages is the message identification number (ID). Different messages with different ID numbers have been designed in order to enable transmission of different pieces of information. These messages are presented in Appendix E.3. Byte 3 of each message is the complementary identification number (Cmpl ID) computed so that SOH+ID+CmplID=256. Byte 4 is the Length of the message, excluding the header (the first 4 bytes) and the Checksum (last 2 bytes). Each message ends with a two bytes Checksum.

The main reason that explains the use of this procedure with all its elements is to prevent any misunderstanding in the nature of the data sent. Thanks to the receiver's characteristics, packet loss is very unlikely, unless the transmitter is out of range.

4.2.4.2 Data Recognition Procedure

The message recognition procedure is the same one as for the GPS receiver and explained in Section 3.3.2. The only difference for the receiver lies in the SOH byte finding process, as the SOH can refer to any component of the system (but the receiver itself).

4.3 Choice of the Transmitter/Receiver (Transceiver)

The choice of the receiver has been based on the following requirements:

- it has to perform serial port communications
- it has to be small and of low cost (one of the main requirements of the project)
- it has to perform reliable communication at more than 1 mile (hopefully 3 miles)

- it needs to operate at a sufficient baud rate.

4.3.1 Computation of Minimum Baud Rate

Communication is performed at 5Hz and therefore the results obtained in Section 4.2 have to be multiplied by 5 to compute the minimum baud rate in bytes per second. The Parent sends up to 100 bytes and the Mini sends 26 bytes, making a total of 126 bytes. Also, the communication signal structure requires 12 bytes for the Parent and 6 for the Mini, making a total of 144 bytes. Multiplied by 5, it makes 720 bytes per second for Parent and Mini navigation.

In addition, communication can involve the Ground station that might send up to 10 bytes. Moreover 20 optional bytes can be sent by the Parent, and 10 by the Mini. The grand total is therefore 920 bytes.

Hence, if data are sent with no parity bit, 920*8=7360 bits have to be sent every second. This is without taking into account the time delay involved by the different components of the system:

- The communication structure can imply time delay if data sent by the Mini are not sent as soon as the Parent's data are received.
- The serial port use can create significant delay (see Section 4.4).
- The transceiver itself can account for significant delay (see Section 4.3.3).

These delays have been estimated to count for at least half of the process and therefore, the minimum baud rate has been estimated at 19.2kbps.

4.3.2 Comparison of Different Transceivers

The author looked at different transceivers available on the market in January 2001. The main requirements for the choice of the transceiver were:

- Compatibility of the transceivers within the different components of the PCUAV system (to perform communication efficiently)
- Serial port communication possible (for interaction with a PC)
- Low price (as a main requirement of the project)
- Small size (to embed them in the UAVs)
- Minimum range of 1 nautical mile (to demonstrate Rendezvous locally)
- Minimum baud rate of 19.2 kbps (to satisfy the amount of data to be transferred)

From these requirements, three different sets of transceivers came out:

- Maxstream 9XStream
- Coness WIT2410
- Radiometrix TX1 and RX1

Systems involving one transceiver have been preferred for their simplicity to the Radiometrix choice (which is composed of one transmitter and one receiver). Moreover the Maxstream 9XStream and the Coness WIT2410 transceivers use a Frequency Hopping Spread Spectrum (FHSS) technique (Section 4.3.3.3) which reduces the risk of interference. The 9XStream has finally been chosen as its price was lower and as Draper Laboratory had already used it (hence a base of experience was available).

4.3.3 The 9XStream Transceiver

The 9XStream 192 module is a 100-milliwatt, FHSS wireless module that allows wireless communication between equipment using a standard asynchronous serial data stream.

4.3.3.1 Technical Characteristics

The half-duplex transmission of the 9XStream can sustain a continuous data stream at 19.2 kbps. The frequency range goes from 902 to 928 Mhz and the transceiver uses a

FHSS technique. It can hop through 25 channels and features 7 different hop sequences. The reception sensitivity is –107 dBm. The range is:

- Indoor: 150m to 400m
- Outdoor: 8km with dipole, over 14 miles with high gain antenna

The supply voltage is 5 VDC +/-0.25V. The current consumption is 150 mA nominal for transmission and 50 mA nominal for reception. The board's size is 4.06cm x 6.86cm x .89cm, it weights 24g, and operates in a temperature range from 0ºC to 70ºC.

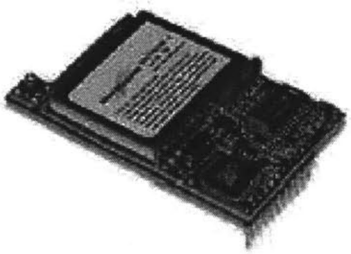


Figure 4.3 MaxStream 9XStream

4.3.3.2 Choice of the Antenna

The 9XStream comes with a 1/4 wave length wire monopole leading to an outdoor range of only 400m. Therefore, the team had to upgrade to a 1/2 wave length antenna Part Number A09-HBMM-7-P6I mounted with an MMCX connection. The accessible range was then tested for communications of up to 1km with no packet loss in an outside environment (the airport fields were used for this test).

Position of the antenna

No real interference or loss of efficiency has been observed with the antenna used. Therefore, the position of the antenna in or on the airplane has not been much of a concern. However, the polarities of the dipoles have to be coherent within all the components of the system.

4.3.3.3 Data Transmission Procedure

The 9XStream presents several characteristics to be taken into account while transmitting data through the air. It mainly consists of:

- a Frequency Hopping Spread Spectrum (FHSS) technique
- a protocol to capture reception and transmission

These techniques are discussed in Appendix E.

The constraints on the transmission procedure stem mainly from the fact that it induces delay when switching from Receive to Transmit Mode. This delay is on the order of 35ms and is due to the header size that is used for synchronization due to the FHSS technique. This delay reduces the bandwidth if switches occur often, as it is the case for the PCUAV system, because communication is based on a Token Ring like procedure.

4.3.4 Communication Constraints and Maximum Rate

When 2 UAVs are used, the Parent sends data at a 5Hz rate and so does the Mini. Therefore 2 transitions from Receive to Transmit Mode occur which can induce a 70ms delay during the 200ms slot time (5Hz) available for transmission. If, in addition, communication is required from the Ground station, this time delay goes up to 105 ms dividing the bandwidth by more than 2. Such observations explain why the minimum baud rate has to be 19.2 kbps, and give the main reason why non necessary data have to be reduced to a minimum.

It has been demonstrated that communication could be performed at 5Hz with the Parent and the Mini in the DGPS case with up to 9 satellites (no situation of more satellites tracked has been observed by the PCUAV team). Unfortunately, in this case, the Parent's IMU optional information (30 bytes * 5 (Hz) = 150 bytes per second) cannot be sent.

4.4 Serial Port Software

In this communication process, as time is a major constraint, serial communication has to be optimized. This optimization has to occur at both transmission and reception of data, for the transceiver and for the GPS receiver.

4.4.1 PCUAV System Serial Ports Addresses

This discussion will mainly focus on the software control part of a serial port. This control is closely linked to the addresses of the Computer's Serial Ports. The team uses four serial ports on a PC104+ stack:

Port	Link	Address	IRQ
COM 1:	GPS Receiver	3F8	4
COM 2:	Transceiver	2F8	3
COM 3:	SBC2000 #1	3E8	Not used
COM 4:	SBC2000 #2	2E8	Not used

Table 4.1 PCUAV System Serial Port Addresses and IRQs

Two serial ports are mounted on the CPU board. Four additional serial ports are available on a PC104 stack card (CM312 LAN and quad serial port Board). Only two of these are used (for the SBC2000s used by the team). The different registers are presented in Appendix F.

4.4.2 Programming the Serial Port

When writing a communication program two methods are available. The Universal Asynchronous Receiver Transmitter (UART) can be polled to see if any new data is available or an interrupt handler can be used to remove the data from the UART when it generates an interrupt. Polling the UART is a much slower method, which is very CPU intensive, and thus, can only have a maximum speed of around 34.8 kbps before data are lost. However, it is the most intuitive way. The other option is using an interrupt handler, which is safer and faster.

4.4.2.1 Polling Method

The Polling Method has been the first one used by the author. It was efficient as long as the amount of data sent from each vehicle was less than 32 bytes. The code is presented in Appendix F.2.

The Maxtream 9XStream transceiver has its own buffer of size 32 bytes, which would occasionally clear the serial port's buffer. At the same time, the serial port's buffer size is only of 16 bytes and to send data over it with a polling system, only two solutions are possible in the PCUAV context:

- send no more than 16 bytes
- arrange data in packets of 16 bytes and wait a certain amount of time between each packet transmission

Both solutions are time consuming, inefficient and the interrupt handled method is preferred.

Results of the Polling Method

Reliable communication has been performed using the Polling Method. The impact of the method on the code was its dependency on serial port communications:

- When GPS data was expected, the code would wait to read the GPS receiver's serial port
- When communication had to be established, the Parent would send data and then wait for the Mini's data

- The Mini would get its GPS data and then wait for Parent's data before sending its own data over the air.

This method, on top of its time consumption, would also fail if communication would not be reliable. Therefore, time-out constants had to be established all over the code to ensure its efficiency.

4.4.2.2 Interrupt Handled Method

The interrupt handled method is much more efficient and avoids wondering about the presence of data in the serial port's buffer as an interrupt takes place when data is received. The code is presented in Appendix F.3. The most important feature of the interrupt handler method is its capability to handle a large amount of data to be transmitted. Moreover, the method is very reliable, and does not consume as much processor time as the Polling method (Section 4.4.2.1).

4.5 Data Compression and Reduction

In order to be able to transmit as much data as possible, different methods of compression have been tested. As transmission occurs at a 5Hz rate with the Parent and the Mini transmitting simultaneously, and due to the transmission time spent by the receiver to switch from the reception mode to the transmission mode, the total amount of data that can be sent in total by both vehicles is on the order of 200 bytes. This total amount can be divided into two categories:

- data required for navigation
- optional data to track the performance of the UAVs from the Ground station

In order to decouple these data, each category corresponds to a specific message independent from the other one. Such a division takes place for the Mini and for the Parent. Therefore, the number of bytes sent per message is not more than 100 bytes. Compression has been tried according to these remarks.

4.5.1 Data Compression

Different data compression methods have been tested and are developed in Appendix E.4. However, none of them gave a good compression ratio for the small amount of data the project is dealing with.

4.5.2 Data Reduction: Working in a Local Frame

The fact of working locally enables reductions in the amount of data sent. In particular, position on the earth has to be sent over a double format (8 bytes) to be precise enough. Working locally reduces the amount of bytes sent for position to 4 bytes. Moreover, a byte can take value in the range 0-255. Therefore 128 Booleans can be saved in a byte. Thanks to this, lots of different pieces of information could be coded and sent through the communication channel.

4.6 The Different Sets of Transmission and Results of the Tests

Tests have been performed to demonstrate the capability of the system. They include performance of the Token Ring like system, reliability of the position of reference setup and range limitations.

4.6.1 Token Ring System

Communication involving the Parent, Mini and Ground station has been performed and no packet loss was observed. The Parent would communicate its information, then the Mini would give its position and finally the Ground station could perform tasks such as resetting the position of reference. With up to 9 satellites tracked by the Parent, all the pieces of information were sent in time with no packet loss during DGPS ground tests. However, the Parent's IMU data (30 optional bytes sent at 5Hz) could not be sent to the Ground station if the DGPS code was ran. In tis case, the limits of the bandwidth of the communication system are reached in this case. The procedure described in order to establish the position of reference and the trajectory points works efficiently.

4.6.2 Range

Range performance ground tests have been performed using the following procedure:

- Computer A sends a byte to computer B
- Computer B acknowledges by returning the message to A
- A sends the next message to B on reception of the acknowledgement or after a time-out.

The range of a monopole was 100 meters on the ground. With a dipole, the range has been estimated to be at least 500 meters (limits of the field). No packet loss was observed during flight tests unless the airplane was right above the Ground station (in this situation, antennas were in the worse possible configuration). Such a configuration is very rare and does not affect airplane to airplane communication.

4.7 Chapter Summary

This chapter has described the different communication procedure used by the PCUAV team to enable autonomous flight and docking. It includes communication requirements due to the specificity of the PCUAV system, the characteristics of the transceiver used, the software process used to enable communication, and results of data compression tests.

Chapter

5

The Overall System

5.1 Chapter Overview

This chapter presents the overall GPS and communication system of the PCUAV navigation and docking part. It then introduces modifications that could be done to improve GPS accuracy and to link navigation communication with other communication concepts.

5.2 The GPS/Communication System

GPS has been presented in Chapter 3, and communication related to GPS has been emphasized in Chapter 4. This section presents a summary of all the GPS linked communication involved in the navigation process and gives details on how this procedure is linked to the control system and Rendezvous planning software.

5.2.1 GPS related Communication Flow

The requirements for GPS communication have been introduced in Section 4.2. However, it is necessary to link them closely to the required GPS data for all the vehicles. Figure 5.1 gives the GPS data flow necessary for computation of the DGPS position and velocity.

These data are then reused by the controller system implemented by Sanghyuk Park.

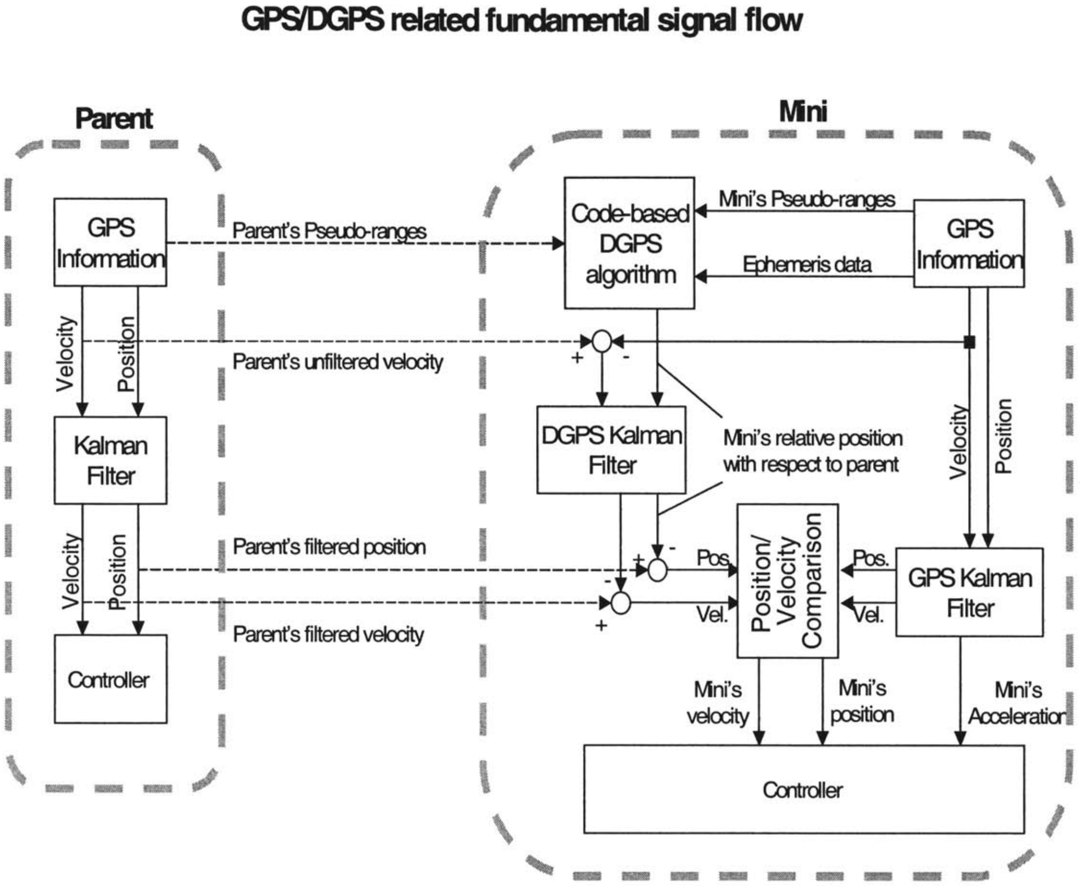


Figure 5.1 GPS Data Flow

The main communication flow is from the Parent to the Mini (as the Mini's trajectory is designed knowing the Parent's position and trajectory). In order for the DGPS code to be processed, Parent's pseudo-ranges have to be sent to the Mini. The result of the code is a relative position of the Mini with respect to the Parent. From this point, different alternatives have been followed:

- The Mini computes its absolute position by adding the Parent's position, and then a Kalman filter is used (Figure 5.2).

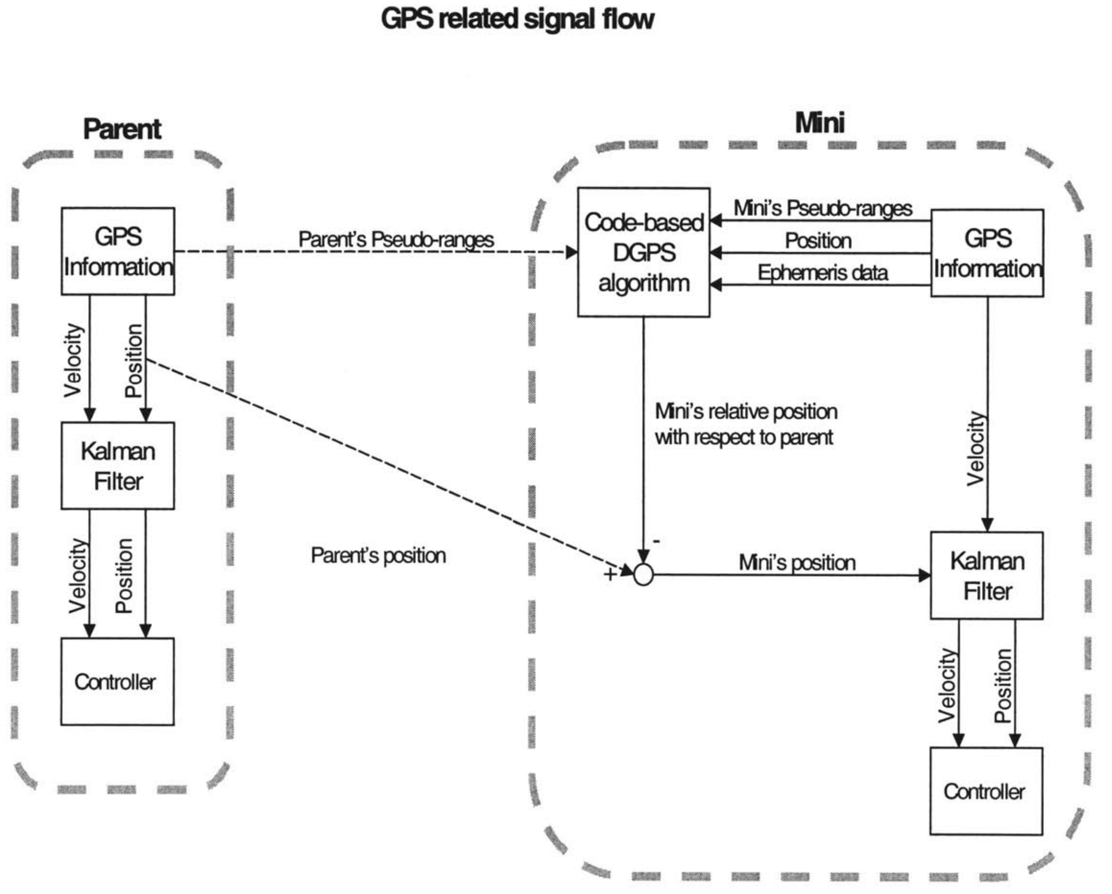


Figure 5.2 Mini's Absolute Position Computation

- The Mini vehicle uses its own position and the Parent's position is used for trajectory planning (Figure 5.3).

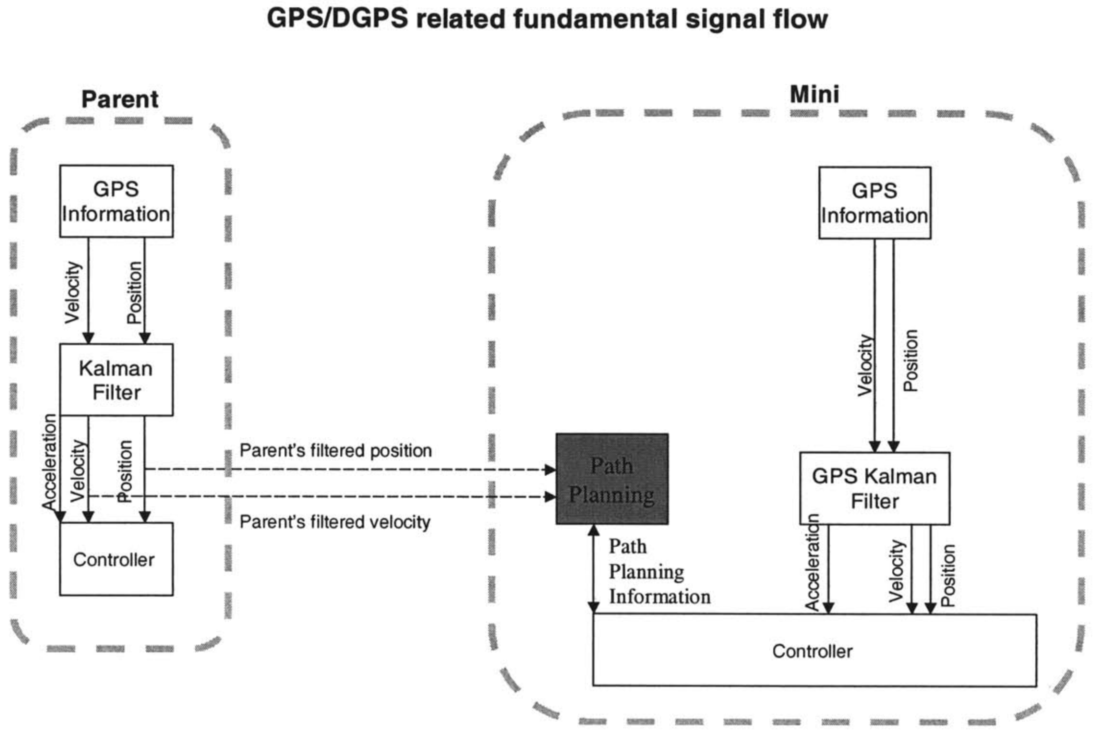

GPS/DGPS related fundamental signal flow

Figure 5.3 Mini's Position and Difference in Position Concept

- The Mini vehicle uses its relative position (Figure 5.1):
  - For autonomous navigation purposes (the controller side), it is necessary to have a local absolute position of the Mini. Therefore, the output of the DGPS Kalman filter is recomputed with the Parent's filtered position and velocity (which then have to be sent to the Mini).
  - The control system uses the acceleration output of the GPS Kalman filter. Having the Parent's acceleration transmitted and recomputed with the DGPS Kalman filter output is not really relevant and consumes a lot of the

communication bandwidth. Therefore, it has been decided to compute this acceleration directly from the Mini's position in the same way as it is done for the Parent. This option also gives the capacity to compare the output of the DGPS Kalman filter with the output of a GPS Kalman filter.

5.2.2 GPS, Communication, Control, and Trajectory Planning

The code that enable the PCUAV team vehicles to perform autonomous navigation and Rendezvous has been built by three different people: Sanghyuk Park for the Controller System Design, Damien Jourdan for the Trajectory Planning and Synchronization and the author, Richard Poutrel, for the GPS and communication inputs. Hence, it has been decided to divide the code into different parts, each of which would be developed by one person.

5.2.2.1 Natural Division of the Vehicles' Code

In the process of dividing the vehicles' code in parts that would only interact via interfaces, the computed GPS data can be considered as inputs of the system. These data are used by:

- The control system for navigation
- The trajectory planner which generates target points for the control system

Communication is a result of the different parts of this process and can be viewed as on the side of this process. Figure 5.4 summarizes these remarks.

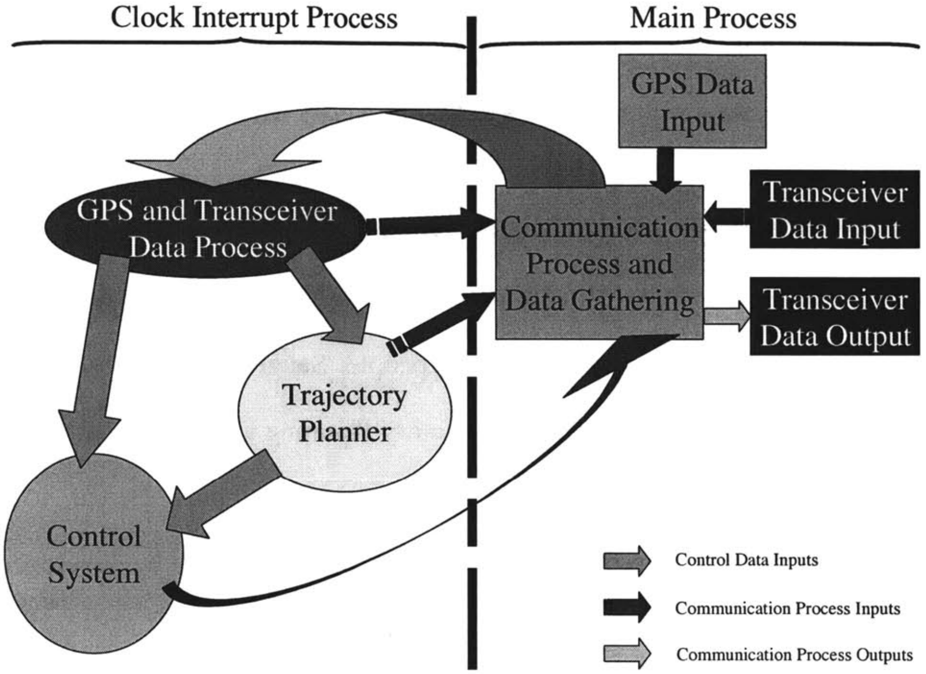


Figure 5.4 Natural Division of the Vehicles' Code

5.2.2.2 Structures of the Vehicles' Code

Based on the natural division of the codes between the project members, a code structure has been decided for all the vehicles. Slight modifications take place for each vehicle according to the progress but the main framework remains the same.

Structure of the Vehicles' "main" Code in C

In the software, the "main" code deals with the accumulation of data. It is mostly composed of a loop that would compute new data if received at a serial port. The loop also deals with:

- file saving
- screen display

- a LED.

These features are necessary to save the flight's data information, to debug and to make sure that the code is running as expected.

```
void main(void){
initialization_steps(); // sets the interfaces with the hardwares and resets data to nominal values
do{
    get_gps_data(); // get the gps data, save new information if available from the GPS.
    handle_transceiver_data(); // get the communication data, save new information if available from the Transceiver.
    handle_keyboard_copilot_inputs(); // read copilot commands or keyboard commands.
    display_LED(CIC,BIAS,PROGRAM_EXIT,SV_computed_for_solution); // display a LED signal
    if(((float)(GPS_time[0])<10.1)||(DEBUG_PRINT_SCREEN==1)){
        handle_screen(ScreenMode); // print data to screen if it's a lab test
        }
    if(ready_to_print==1){
        handle_file(); // save data to file
        ready_to_print=0;
        }
    }while(PROGRAM_EXIT!=1);
    display_LED(CIC,BIAS,PROGRAM_EXIT,SV_computed_for_solution);
    closing_steps(); // stop the serial ports and close files
}
```

The efficiency of this code is ensured by reading data with an interrupt function. The two functions *get_gps_data()* and *handle_transceiver_data()* test if data have been put into the reception buffers, and most of the code is only run in case of reception of a whole message at either serial port. Within these functions, global variables are set in order to communicate with other parts of the program. This information is then used in a clock interrupt function ran at 40Hz. The clock interrupt function performs the following GPS and communication related tasks:

- Computation of GPS or DGPS data
- Filtering (Kalman filter)

Such operations are only performed when data has been treated by the *get_gps_data()* and *handle_transceiver_data()* as shown by Figure 5.4. Then, the controllers use the information and the Token Ring Procedure is initialized.

Thanks to this structure, serial port communications are decoupled from the controllers and the synchronization process. Therefore modifications on either parts of the code can be done more efficiently.

Structure of the Ground Station's Code

The code is very similar to the vehicles' code. The main difference is that there is no need for clock interrupt functions at the ground station as its status is only to provide information about the flight. Another difference comes from the interaction that can occur between the Ground station operator and the program. It can be decided to perform operations such as resetting the position of reference. In order to accomplish this task, an *handle_keyboard_input()* function has been implemented in an outer loop of the main loop.

5.3 Improvements to be Considered

Improvements regarding GPS and communication can be considered for the PCUAV project.

5.3.1 CDGPS

The accuracy of the Differential GPS methods can be improved by using GPS satellites' carrier phase data. It will first be shown how phase is determined and can be related to the distance to satellites, then how it can be used for differential positioning and eventually how it can be integrated to the PCUAV project reintegration concept.

5.3.1.1 Carrier Phase Estimation

If $\varphi^S(t)$ represents the phase of the received and reconstructed carrier with frequency f^S and $\varphi_R(t)$ the phase of a reference carrier generated in the receiver with frequency f_R then, as the phase is the derivative of the frequency for each signal, it can be related to the distance to the origin of a signal if t is an epoch in the GPS Time reckoned from an initial epoch t_0 assumed to be $t_0=0$. The following phase equations can be obtained:

$$\varphi^S(t) = f^S t - f^S\frac{\rho}{c} - \varphi_0{}^S \qquad (eq.\ 5.1)$$

$$\varphi_R(t) = f_R t - \varphi_{0R} \qquad (eq.\ 5.2)$$

where the initial phases $\varphi_0{}^S$ and $\varphi_{0R}$are caused by clock errors and are equal to:

$$\varphi_0{}^S = f^S\delta^S \qquad (eq.\ 5.3)$$

$$\varphi_{0R} = f_R\delta_R \qquad (eq.\ 5.4)$$

Hence the phase difference is given by:

$$\begin{aligned}\varphi_R{}^S(t) &= \varphi^S(t) - \varphi_R(t)\\ \varphi_R{}^S(t) &= -f^S\frac{\rho}{c} - f^S\delta^S + f_R\delta_R + (f^S - f_R)t\end{aligned} \qquad (eq.\ 5.5)$$

The deviation on the frequencies $f^S$and f_R from the nominal frequency f is only on the order of some fractional parts of Hertz and such an error can be neglected. Therefore (eq. 5.5) can be written:

$$\varphi_R{}^S(t) = -f\frac{\rho}{c} - f(\delta^S - \delta_R) \qquad (eq.\ 5.6)$$

If a receiver is switched on at time t_0, the instantaneous fractional beat phase is measured. The initial integer number M of cycles between satellite and receiver is unknown. However, when tracking continuously, this number, caller integer ambiguity, remains the same and the beat phase at epoch t is given by:

$$\varphi_R{}^S(t) = \Delta\varphi_R{}^S \cdot (t - t_0) + M \qquad (eq.\ 5.7)$$

where $\Phi = \Delta\varphi_R{}^S$ denotes the measurable fractional phase at epoch t augmented by the number of integer cycles since the initial epoch t_0. Therefore the equations given the code phase can be summarized by:

$$\Phi = \frac{1}{\lambda}\rho + \frac{c}{\lambda}\Delta\rho + M \qquad \textit{(eq. 5.8)}$$

It is then to be noticed that the phase of the carrier can be measured to better than 0.01 cycles, which corresponds to millimeter precision. Unfortunately, the integer ambiguity M remains unknown. However, if two receivers are considered, this ambiguity can be lifted as shown in Section 5.3.1.2.

5.3.1.2 Carrier Phase Differential GPS

If two receivers are considered (eq. 3.28) gives the relationship between the difference in code phase related to the line of sight of each satellite tracked. This relationship can also be written for carrier phases by taking into account the integer ambiguity vector for each satellite **M**, which is an unknown constant:

$$\Delta\Phi = \begin{bmatrix}\mathbf{H} & \mathbf{I}\end{bmatrix} \bullet \begin{bmatrix}\Delta\mathbf{X} \\ \Delta t\end{bmatrix} + \mathbf{N_e} + \mathbf{M} \qquad \textit{(eq. 5.9)}$$

If **M** would be known, then computation of position could be performed in the same way as discussed in Section 3.5.2.1 and would provide results accurate for up to a millimeter. However, this constant has therefore to be determined, for which different methods can be used.

5.3.1.3 Determination of the Relative Integer Ambiguity

If computation is started with two GPS receivers' antennas at the same position, then $\Delta\mathbf{X}$ is equal to 0, and therefore (eq. 5.9) can be reduced to:

$$\Delta\Phi = \Delta t + N_e + M \qquad (eq.\ 5.10)$$

By knowing **Δt** from code phase positioning computation, N_e+**M** can be determined precisely. In order to improve the result, integration over time can be performed. This result can then be used as a value of **M** by neglecting N_e.

Unfortunately, this method requires the computers to be switched on the ground with the planes positioned next to each other. The PCUAV team prefers not to have planes take off with the computers on to prevent from interference between the computer and the RC signals. Therefore, the receivers would have to remain switched on, the value of **M** saved in a file and then reused. Due to the complication induced by this method and regarding the performances of other systems such as the Vision system, this method has not been implemented

5.3.2 Communication Improvements

One of the main drawbacks of the PCUAV system so far is the inability to combine the communication system developed by Alexander Omelchenko and the control system. This is mainly due to the fact that both systems run on different Operation Systems (OS). The image communication part runs under Linux where the control system is ran under DOS. Moreover, due to timing constraints in the control system and the necessity to have rapid reactions, combining both codes might be a real issue.

However, the author believes that both communication systems should be combined. In order for this operation to be accomplished serial communication would have to be replaced by Wireless LAN Modem communication. The baud rate would not be an issue then, as Wireless LAN communication can provide a larger Baud rate with available Wireless cards, and as a communication range of more than 500 meters has been demonstrated by Alexander Omelchenko.

5.4 The Author's Input

During the last two years of the PCUAV (Parent Child Unmanned Air Vehicle) project, many areas have been explored by the team. The author participated in three aspects.

The first aspect explored is the way GPS information could be used. The author explored different ways of getting accurate relative positioning between two UAVs. In particular he demonstrated that two UAVs could be positioned with a horizontal accuracy of 7 meters by only using their own GPS position. He also demonstrated that thanks to communication, this relative accuracy could be improved.

A reliable communication protocol has been designed and implemented by the author. This first of a kind system enables reliable communication between independent systems. In particular, it avoids the possibilities of interference between the different components of the system thanks to a token-ring like protocol. Moreover, it enables continuous transmission of information from one UAV if another vehicle is absent.

The author was involved in ground and flight tests. In particular, he was in charge of the Ground station with Damien Jourdan. By looking at the information available on the Ground station laptop during flight tests, the author gave the order to abort several automated flight attempts, and thus, he prevented several possibilities of crash. He also provided the UAVs' copilots with real time information on the flights' status.

5.5 Chapter Summary

This chapter has shown how the PCUAV team could combine all the different components of the control system. It has also introduced interesting concepts regarding Differential GPS improvements and new paths of Communication that could be followed. It has finally summarized the author's work within the PCUAV project.

References

[1] Alexander Omelchenko, *Communication and Video Surveillance in the Parent and Child Unmanned Air Vehicle*, Master of Science in Aeronautics and Astronautics, Massachusetts Institute of Technology, 2001.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, *Global Positioning System, Theory and Practice, Third, revised edition*, 1994.

[3] Canadian Marconi Company *Allstar User's Manual*, Supersedes Publication No. 1200-GEN-0101, 1998.

[4] Craig Peacock, *http://www.beyondlogic.org/serial/serial.htm*, August 19, 2001.

[5] Eurocontrol and IfEN, *WGS Implementation Manual*, http://www.wgs84.com, February 12, 1998.

[6] Francois Urbain, *Vehicle Design, Flight Control Avionics, and Flight Tests for the Parent and Child Unmanned Air Vehicle*, Master of Science in Aeronautics and Astronautics, Massachusetts Institute of Technology, 2001.

[7] Elliott D. Kaplan, *Understanding GPS: Principles and Applications*, chapter 2: Fundamentals of Satellite Navigation, Artech House, February 1996.

[8] Jason Kepler, *Parent Child Unmanned Air Vehicles and the Structural Dynamics of an Outboard Horizontal Stabilizer Aircraft*, Master of Science in Aeronautics and Astronautics, Massachusetts Institute of Technology, 2002.

[9] MaxStream Incorporated, *9XStream Wireless Modem specification*, MaxStream Incorporated, 2000

[10] MaxStream Incorporated, *9XStream Frequency Hopping Data Radio Operating Manual v2.1*, MaxStream Incorporated, 2002.

[11] Navstar GPS Space Segment/Navigation Interface. *Global Positioning System Standard Positioning Service Signal Specification*, 1995.

[12] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, Progress in Astronautics and Aeronautics, Vol 190, chapter 4, pages 129-182, 2000.

[13] Richard P. Kornfeld, *The Impact of GPS Velocity Based Flight Control on Flight Instrumentation Architecture*, Doctor of Philosophy in Aeronautics and Astronautics, Massachusetts Institute of Technology, 1999.

[14] Richard P. Kornfeld, R. John Hansman, John J. Deyst, *Single-Antenna GPS-Based Aircraft Attitude Determination*, Navigation: Journal of the Institute of Navigation Vol 45, No. 1, Spring 1998.

[15] Sanghyuk Park, *Integration of Parent-Child Unmanned Air Vehicle Focusing on Control System Development*, Master of Science in Aeronautics and Astronautics, Massachusetts Institute of Technology, 2001.

Appendix

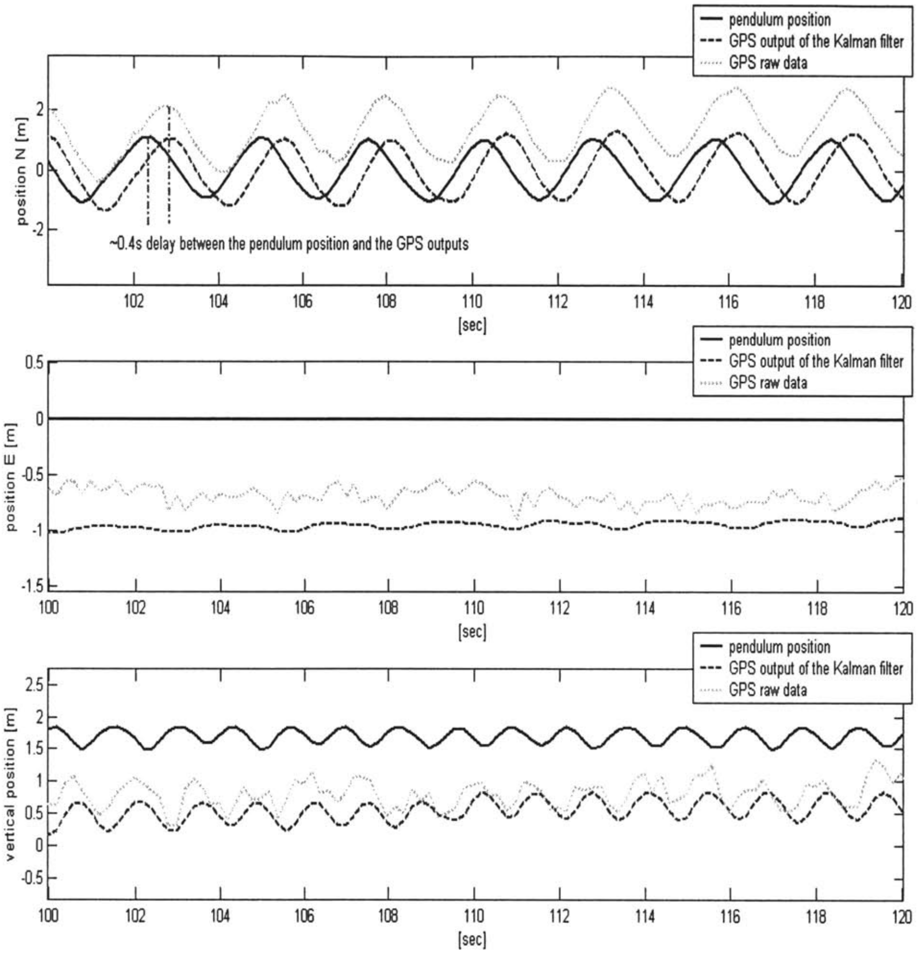
A

Ground Tests Results

Results of the pendulum ground tests are provided for one or two GPS receivers.

A.1 One GPS Receiver Pendulum Test Results

A.1.1 Position



~0.4s delay between the pendulum position and the GPS outputs

Figure A.1 Position Information for 1 GPS Receiver

A.1.2 Velocity

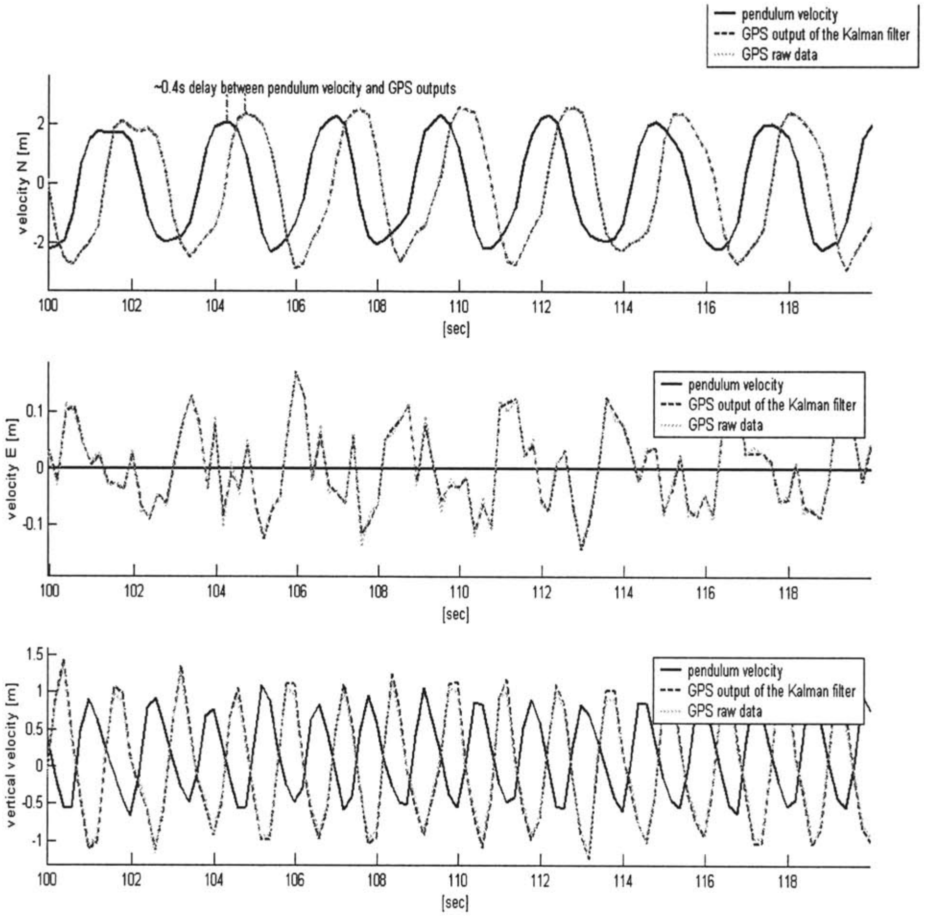


Figure A.2 Velocity Information for 1 GPS Receiver

A.2 Two GPS Receivers Pendulum Tests Results with One Computer

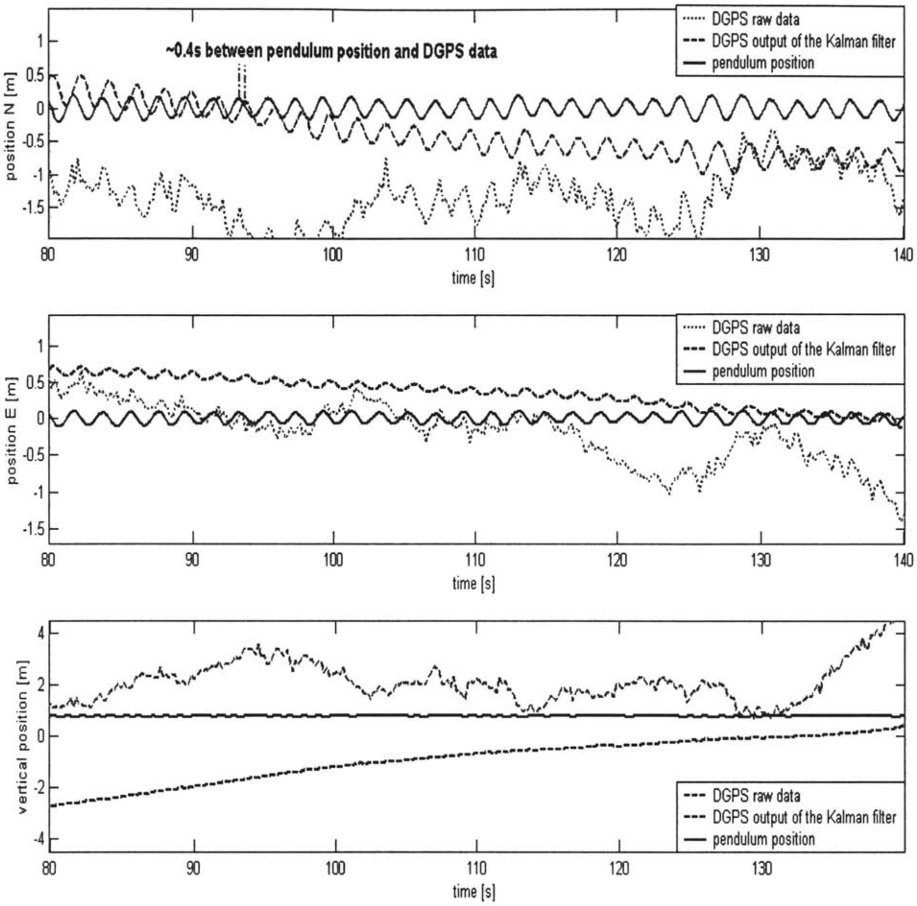


Figure A.3 DGPS Position

A.3 Two GPS Receivers Pendulum Tests Results with Two Computers

A.3.1 Position

A.3.1.1 Differential Position

In this test the position using DGPS is computed for the Mini. The GPS data are computed using the Mini's and the Parent's GPS information.

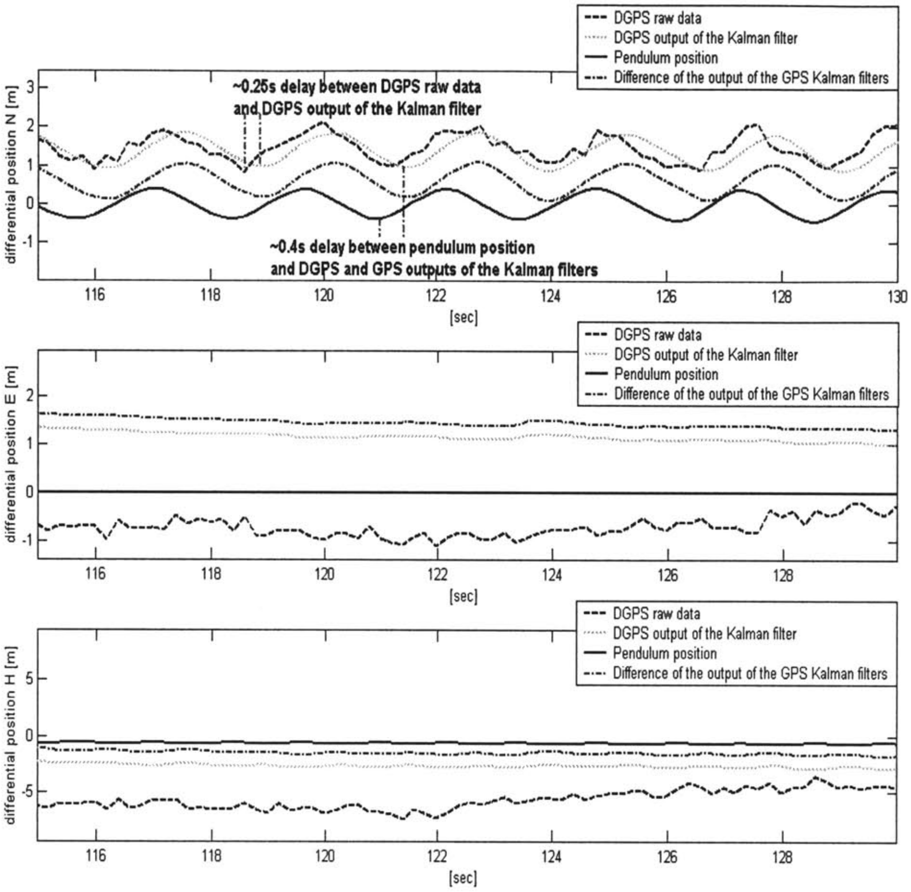


Figure A.4 Differential Positioning using DGPS or GPS Information

A.3.1.2 Absolute Position

In this test the position using DGPS is computed using the DGPS information of the Mini and the GPS information of the OHS. The GPS data are computed using the Mini's GPS information.

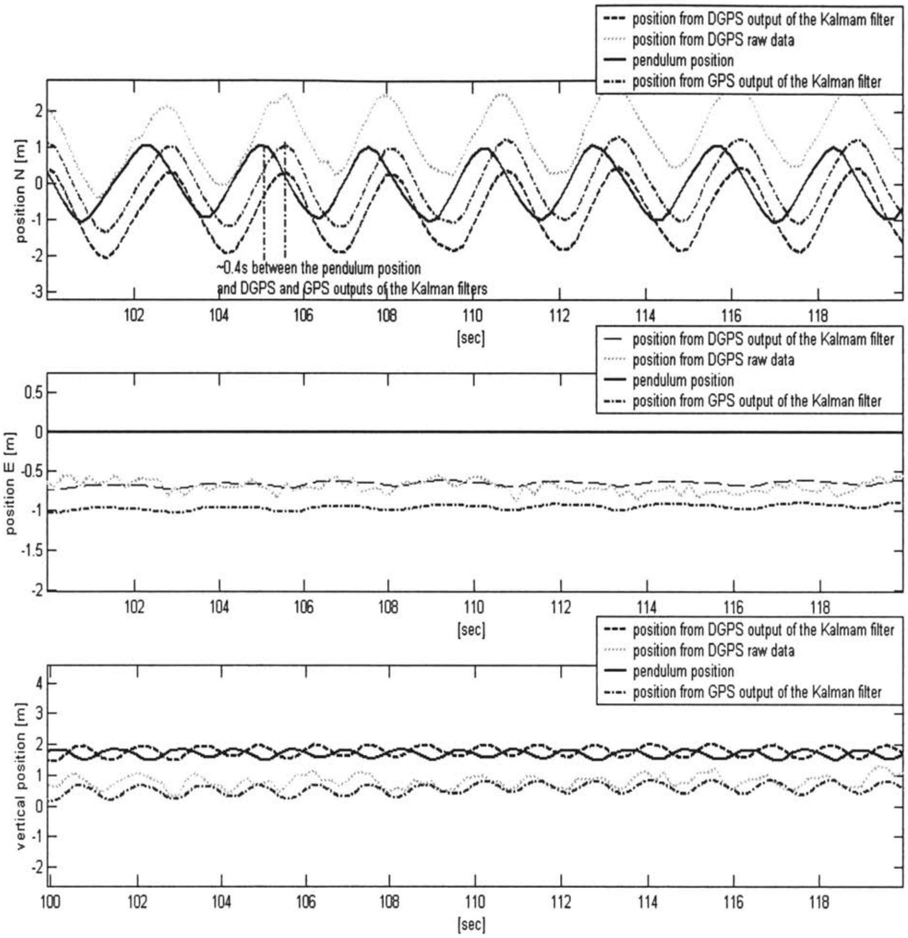


Figure A.5 Absolute Positioning using GPS or DGPS Information

A.3.2 Velocity

For velocity, the information used is only the one sent by the GPS receiver. Therefore no

Differential computation is presented.

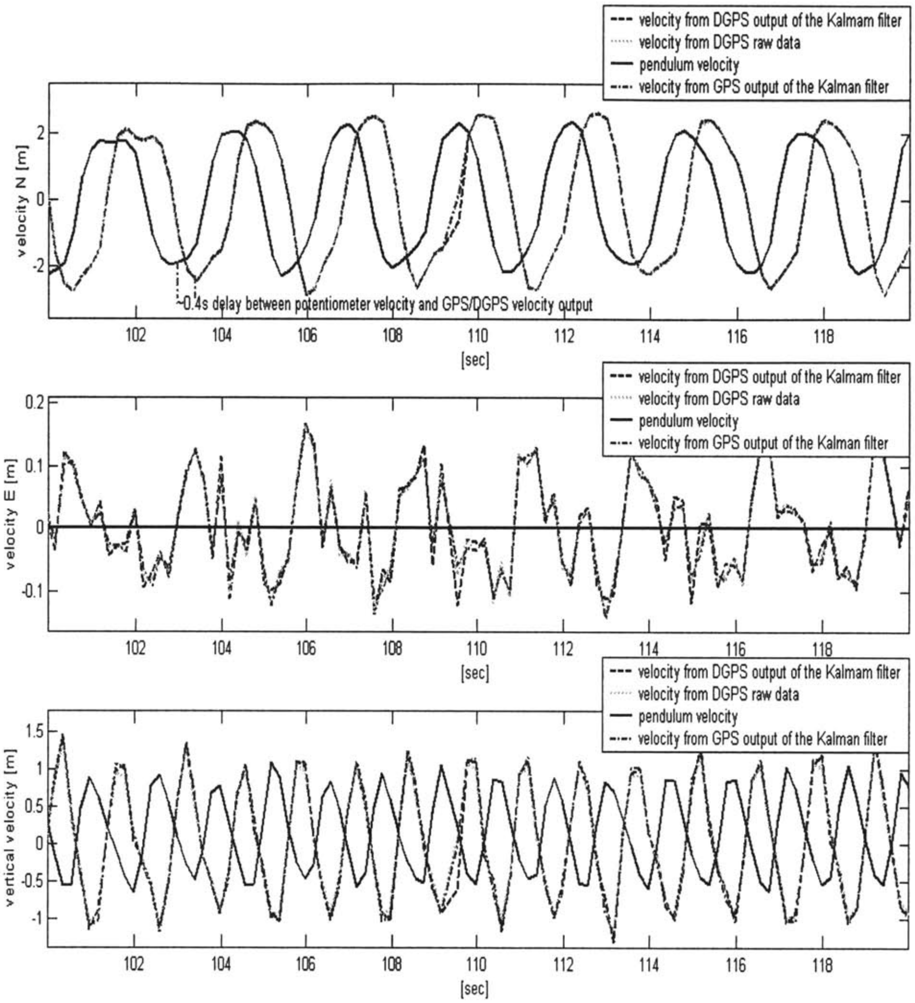


Figure A.6 Velocity Information for the Pendulum Test

Appendix

B

Flight Tests Results

B.1 Autonomous Flight around a circle

The first sets of results present the autonomous flights of the Mini and the Parent along a circular trajectory.

B.1.1 Mini's Flight around a circle

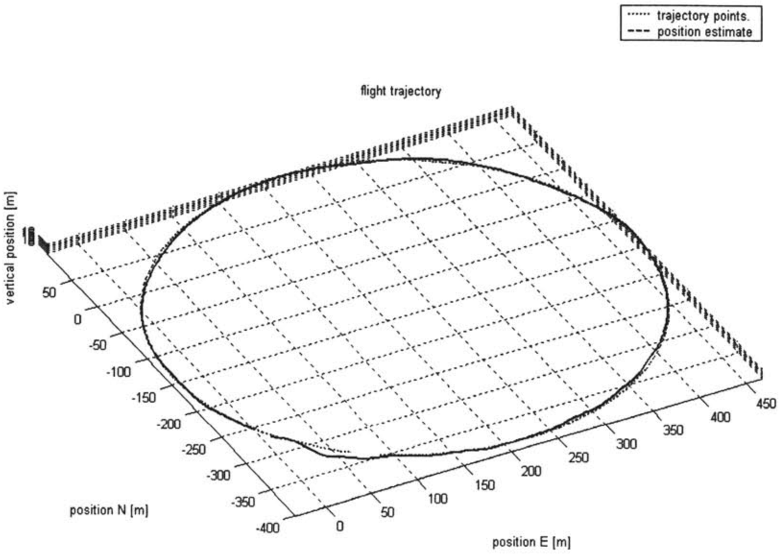


Figure B.1 Mini's Circle Trajectory Results

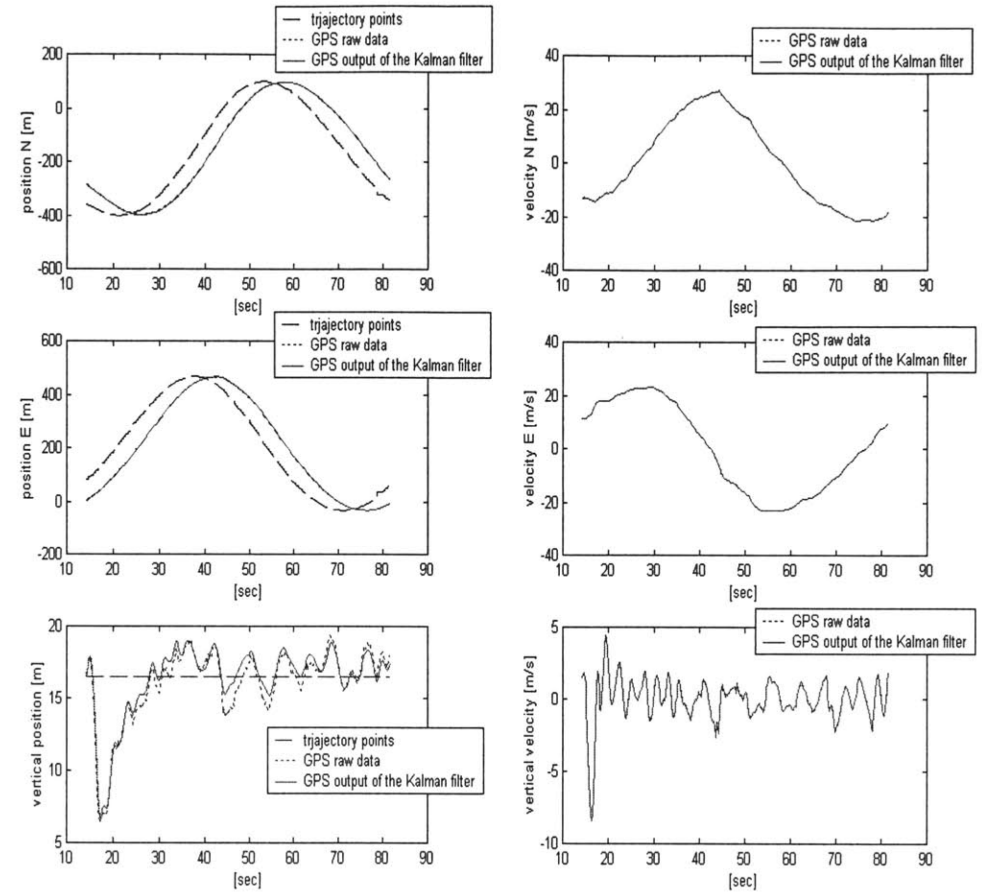


Figure B.2 Mini's Circle Trajectory Data

The trajectory points are designed to be 100 meters away from the Parent, at any time during the flight.

B.1.2 Parent's Flight around a circle

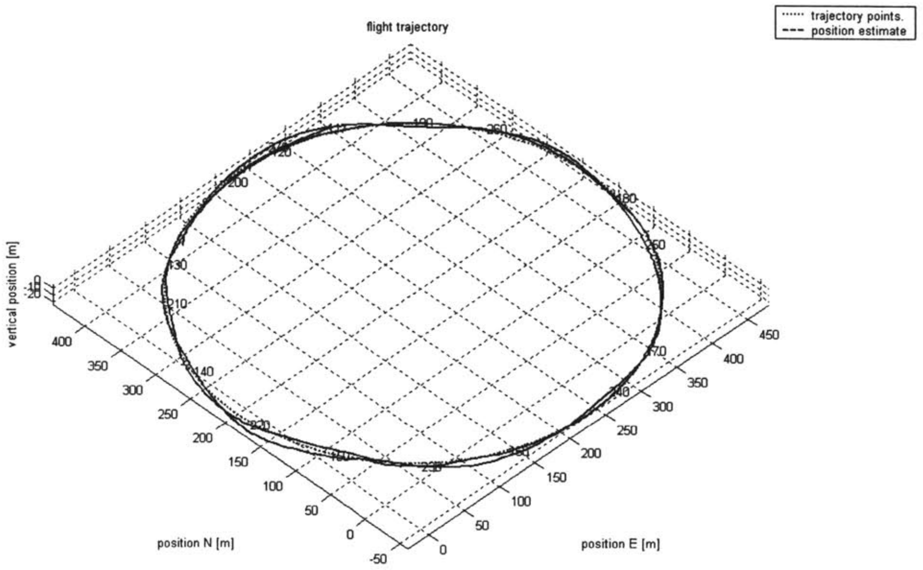


Figure B.3 Parent's Circle Trajectory Results

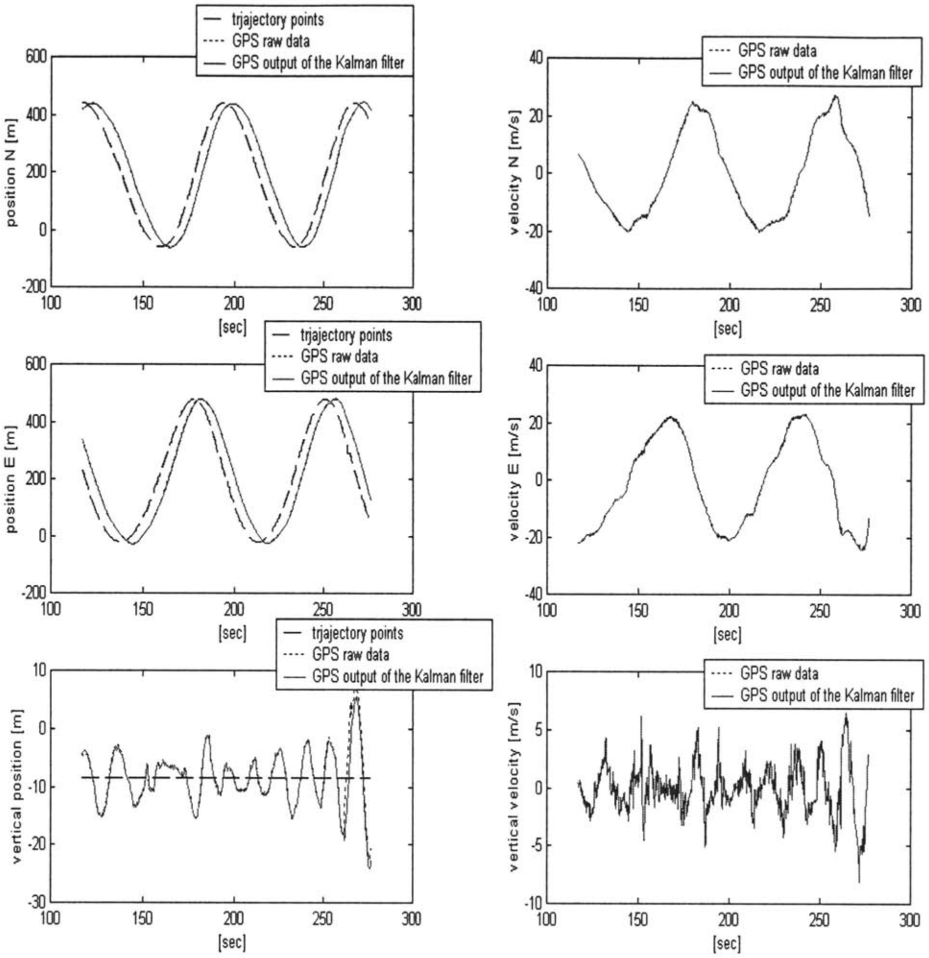


Figure B.3 Parent's Circle Trajectory Results

The trajectory points are designed to be 100 meters away from the Parent, at any time during the flight.

B.2 Autonomous Phase I Test for the Mini

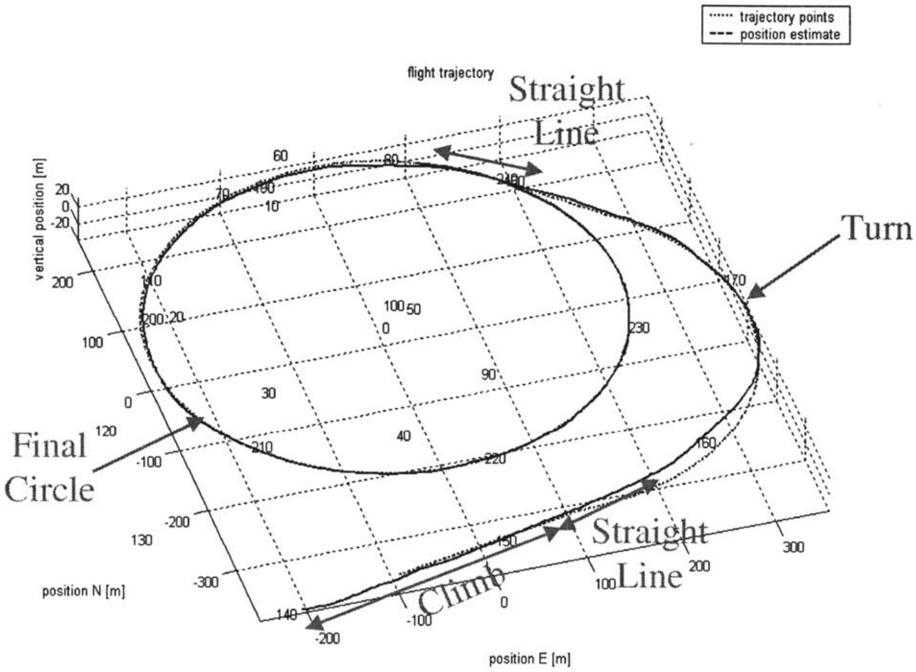


Figure B.4 Autonomous Phase I Trajectory Results

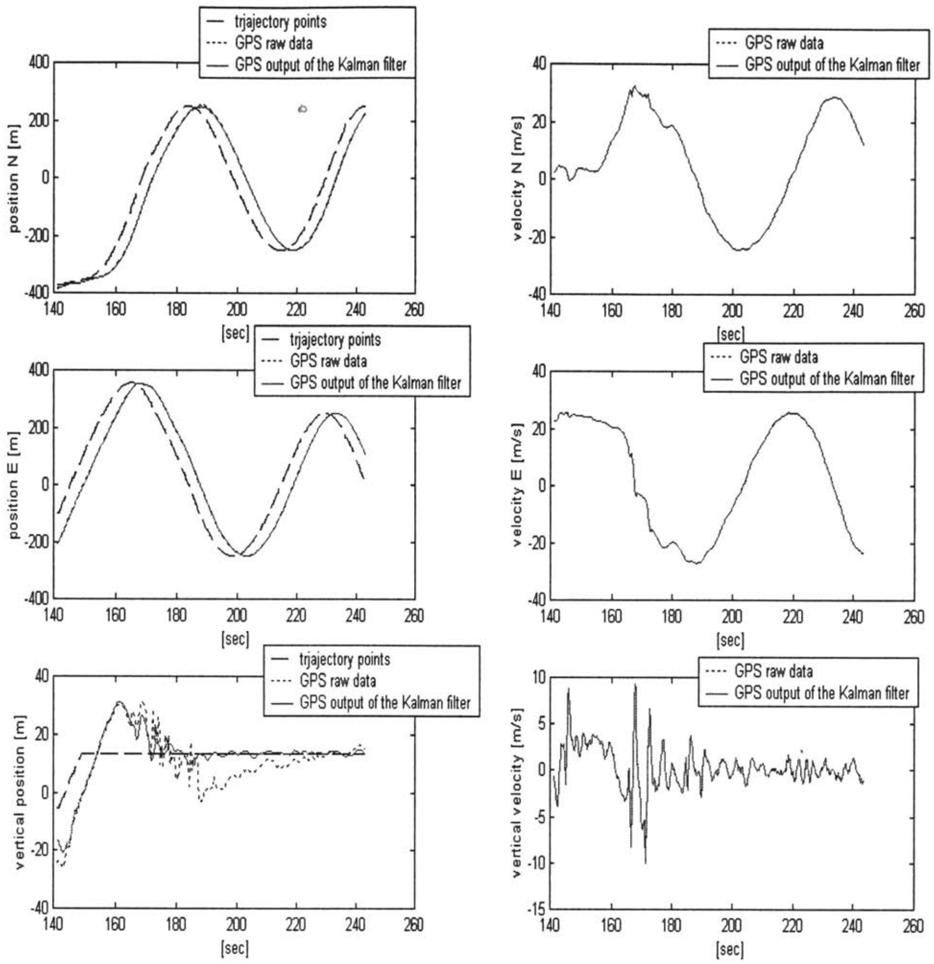


Figure B.4 Autonomous Phase I Trajectory Results

The trajectory points are designed to be 100 meters away from the Mini, at any time during the flight.

B.3 GPS based Phase I with Parent and Mini.

Due to bad weather conditions in the Boston area during the month of may 2002, Phase I flight test could not be performed at the time this Master’s Thesis was written.

B.4 DGPS based Phase I with Parent and Mini.

Flight tests using DGPS information has not yet been performed at the time this thesis was written.

Appendix C

GPS Receiver Characteristics

C.1 Hardware Profile

The GPS receiver (Figure C.1) presents the following main characteristics [3]:

- Single 5V input operation
- Operation under standard temperature range (-30°C to +75°C)
- 1 PPS Output Aligned on GPS Time ±200 ns
- 5 Hz Measurement Output Aligned on GPS Time
- Code and Carrier tracking of L1 GPS frequency
- Serial Input/Output Data Ports with Baud Rate from 300 to 38400 bps
- Supports Active and Passive Antennas
- 12 channel correlator for all-in-view satellite tracking

- The Receiver uses WGS-84 as its geographic reference.

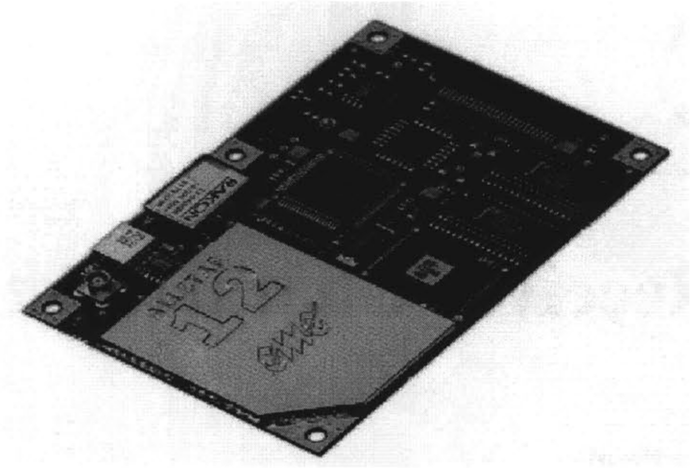


Figure C.1 Allstar GPS Receiver Single Board

On top of its accuracy performances, the receiver shall provide valid and useful outputs in less than 50 seconds (95%) after completion of a self-test if the following criteria are met:

- Valid time (±10 min) and position data (±100 km) from actual real position available.
- Valid almanac data (less than 1 year old)
- HDOP<6

Under certain circumstances [3], this time delay can be reduced.

Receiver Performance

The receiver meets the performance requirements defined below under conditions of vehicle operating speeds up to 514m/s, acceleration up to ±4.0g, jerk of up to 2 m/s^3, the range temperature specified herein and the minimum carrier-to-noise ratios specified herein.

- The receiver is meant to operate using the L1 GPS signal as described in [11].

- There is no disruption of navigation data output when a satellite signal is lost, for reasons other than a receiver power interrupt, for a period of less than or equal to 200 milliseconds. When a satellite signal is lost, for reasons other than a receiver power interrupt, for a period greater than 200 milliseconds but less than 5 seconds, the receiver reacquires the satellite signal within 0.3 seconds after the satellite visibility has been restored. When a satellite signal has been lost due to signal masking, the signal is typically reacquired within 2-3 seconds after the satellite signal meets the minimum input levels. The vehicle dynamics during the masking period are assumed to be less than or equal to 0.5g acceleration and 100 m/s velocity. When total signal masking occurs, navigation will resume within 3-5 seconds.
- The impedance is 50 ohms.
- The receiver has the following noise figure characteristics in the temperature range of -40°C to +85°C and supply voltage range 5V ±5%:
    - Typical: 3.8 dB
    - Maximum: 4.8 dB
- The receiver is capable of acquiring satellite signals with a minimum input carrier-to-noise density ratio (C/N_0) to the correlator of 34 dB-Hz.
- Once a signal has been acquired, the receiver is capable of tracking satellite signals with a minimum input carrier-to-noise density ratio (C/N_0) to the correlator of 31 dB-Hz.
- The receiver is capable of withstanding a signal level not exceeding +15 dBm at L1+/- 50 MHz without damage.
- The receiver, in a suitable system configuration, is capable of continuous operation under interference conditions specified in Figure C.2

- The receiver is suitable for operations if the temperature variation does not exceed 4°C per minute.
- The receiver can some vibrations [3].

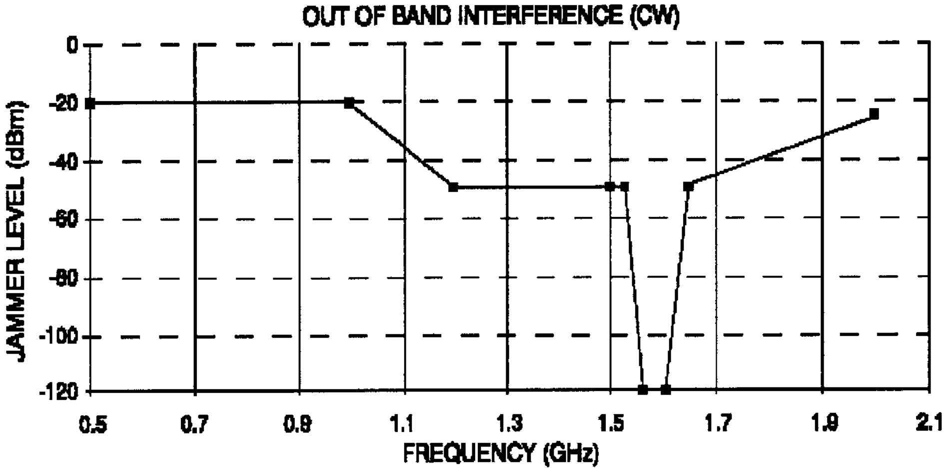


Figure C.2 Out of Band Interference for the Allstar GPS Receiver

C.2 Characteristics of the Antenna AT575-70

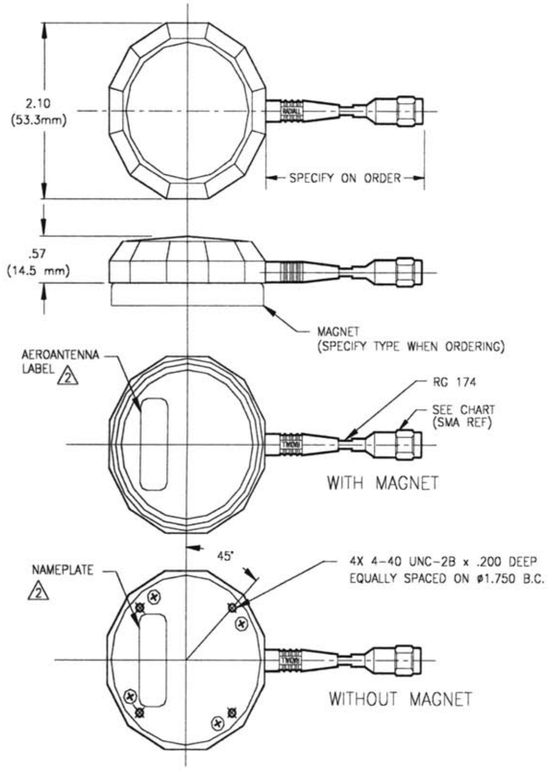
PROPRIETARY NOTICE: This document contains proprietary information of AAT Inc. neither the document nor said proprietary information shall be published, reproduced, copied, disclosed or used for any purpose other than consideration of this document without the express written permission of a duty authorized representative of said Company.

REVISIONS			
SYM	DESCRIPTION	DATE	APPROVED
D	SEE ECO 4498	07/26/00	

AT575-70

NOTES: UNLESS OTHERWISE SPECIFIED

1. LOGO OPTIONAL.
2. LABEL SIZE, SHAPE AND CONTENTS SUBJECT TO CHANGE WITHOUT NOTICE.
3. TOLERANCES: .XX = ±.03
   .XXX = ±.010



SPECIFICATION:

FREQUENCY:	1575 MHz ±5 MHz
POLARIZATION:	RIGHT HAND CIRCULAR
AXIAL RATIO:	3 dB MAX
RADIATION COVERAGE:	4.0 dBic $0° = 0°$ −1.0 dBic $0° < \theta < 75°$ −2.5 dBic $75° \leq \theta < 80°$ −4.5 dBic $80° \leq \theta < 85°$ −7.5 dBic $\theta = 90°$
AMPLIFIER:	
GAIN:	SPECIFY WHEN ORDERING
NOISE FIGURE:	2.5 dB MAX
VOLTAGE:	SPECIFY WHEN ORDERING
IMPEDANCE:	50 OHM
VSWR:	≤2.0:1
BAND REJECTION:	35 dB
POWER HANDLING:	1 WATT
FINISH:	WEATHERABLE POLYMER
CONNECTOR:	SPECIFY WHEN ORDERING
WEIGHT:	5 OUNCES MAX
ALTITUDE:	20,000'
OPERATING TEMP:	−40° TO +70°C
DESIGNED TO:	D0160C

PART NO. DESIGNATION:

AT575-70X-XXXX-XXX-XX-XX-XX

COLOR
W = WHITE
S = SMOKE GRAY
O = OLIVE DRAB

CONNECTOR
TNCM = TNC MALE
TNCF = TNC FEMALE
BNCM = BNC MALE
BNCF = BNC FEMALE
MCXM = MCX MALE
MCXF = MCX FEMALE
SMAM = SMA MALE
SMAF = SMA FEMALE
• NTPM = N TYPE MALE
• NTPF = N TYPE FEMALE
• 0000 = NO TERMINATION

CABLE LENGTH
000 = IN INCHES

MAGNET
NM = NO MAGNET
IM = INTERNAL
RM = REMOVABLE
HD = HEAVY DUTY

GAIN (± 2 db)
00 = PASSIVE
12 = 12 db (20 mA)
26 = 26 db (35 mA)
• 36 = 36 db (50 mA)
XX = OTHER

VOLTAGE
00 = PASSIVE
05 = 5 VDC
• RG = 5-18 VDC
XX = OTHER

• = OPTION NOT AVAILABLE WITH THIS MODEL

STANDARD MODEL= AT575-70W-SMAM-120-05-26-RM

RIGHT ANGLE CONNECTOR ADD "R" TO CONNECTOR DESIGNATION.
THIS OPTION AVAILABLE FOR CABLED ANTENNAS ONLY.

DO NOT SCALE THIS DRAWING
ALL DIMENSIONS ARE IN INCHES
DIMENSIONING & TOLERANCING PER ANSI Y14.5M-1982
TOLERANCES: ~~.XX = ±.01~~
~~.XXX = ±.005~~
FRACT = ±1/32
ANG. = ±1/2°
√ SURFACE ROUGHNESS PER ANSI B46.1
REMOVE BURRS AND BREAK SHARP EDGES:
PART TO BE CLEAN AND OIL FREE
UNLESS OTHERWISE SPECIFIED

USED ON

DRAWN	S. REESE	08/17/94
CHECKED	G. SHACHAR	11/28/95
ENGR	J. AMADO	12/01/95
Q.A.		
APPROVED	J. KLEIN	12/02/95
DRAWN IN	ACAD	

AeroAntenna Technology Inc

TITLE
GPS ANTENNA

SIZE	CAGE CODE	DRAWING NO.	REV
B	0UVG2	AT575-70	D
SCALE NONE	AT575-70D	SHEET 1 OF 1	

Appendix

D

GPS Navigation Messages

D.1 The Allstar Binary Messages

Different sets of messages are used or have been used by the PCUAV team. They are described in Appendix D.1.2. The ordering of data within message blocks is such that the least significant bit (LSB) is the first bit received; moreover, the most significant bit (MSB) is the last bit in the sequence. This ordering is applied to all data formats presented in Appendix D.1.1, which include integer values, fixed point values, floating point values, and character strings.

D.1.1 The Binary Messages Data Structure

Compared to the NMEA mode where all the characters involved are in a classical handwriting format, numbers and characters are sent in numerical formats. There are five sets of data structure involved:

- Characters: unsigned by default
- Integer values: presented in two's complement form
- Fixed point values : set as needed
- Floating point values : stored in IEEE format using "little endian" method to store data which are longer than one byte. Words are stored in two consecutive bytes

with the low-order byte at the lowest address and the high-order byte at the highest address. The same convention applies for 32 bit and 64 bit values.

Short Float (32 bits)	Double Float (64 bits)
• MSB (bit 31) = Sign	• MSB (bit 63) = Sign
• Bit 30..23 = Exp	• Bit 62..52 = Exp
• Bit 22..00 = Mantissa	• Bit 51..00 = Mantissa
2exp(-1*bit22) + 2 exp(-2*bit21).......	2exp(-1*bit51) + 2 exp(-2*bit50).......
$\text{Value}= \text{Sign} \times 1.\text{mantissa} \times 2e^{\text{Exp} - 127}$	$\text{Value}= \text{Sign} \times 1.\text{mantissa} \times 2e^{\text{Exp} - 1023}$

Table D.1 Floating Point Format Definition

D.1.2 The Different Sequences of the Allstar Binary Messages

MESSAGE	BYTE	DESCRIPTION
20 Navigation Data (user coordinates)		The message is output 5 times per second in the 5Hz mode. The latency on this message is less than 0.2 seconds. The latency defined here refers to the time difference between the time tag of the computed position and the time of transmission of the first message byte. Message Length : 77 bytes
	5..14	UTC Time; units: HR:MN:SC
	5	Time not corrected by UTC parameters (1=True, 0=False)
	6-7	Reserved
	15..18	Date;byte 15, bits 5-7: Reserved
	19..26	Latitude;range: -PI/2 .. PI/2
	27..34	Longitude;range: -PI .. PI
	35..38	Altitude
	39..42	Ground Speed
	43..46	Track Angle;range: -PI .. PI
	47..50	Velocity North
	51..54	Velocity East
	55..58	Vertical velocity
	59..62	HFOM
	63..66	VFOM
	67..68	HDOP; resolution: 0.1 units
	69..70	VDOP;resolution: 0.1 units
	71	bits 0-4: NAV Mode

		0 -> Init. Required, 1 -> Initialized, 2 -> Nav 3-D, 3 -> Alt. Hold (2-D), 4 -> Diff. 3-D, 5 -> Diff. 2-D, 6 -> Dead. Reckoning bit 5: Solution Confidence Level 0-> Normal (NAV solution from less than 5 SVs) 1 -> High (NAV solution from at least 5 SVs) bits 6: Reserved bit 7 : GPS Time Alignment mode 1-> Enable 0-> Disable
	72	bits 0..3 : Number of SVs used to compute this solution
	73	System Mode and Satellite tracking mode (c.f. msg #49, byte 5); bit 7: Reserved
	72	bits 4..7 : Coordinate system (lowestnibble)
	73	bits 4,5: Coordinate system (highest nibble). Datum number= B73 b5,b4,B72 b7,b6,b5,b4 (B=byte, b=bit).
	74..75	Reserved
21 Navigation Data (GPS coordinates)	5..12	GPS Time; range: 0.0 .. 604800.0
	13..14	Week
	15..22	X Position in GPS units
	23..30	Y Position in GPS units
	31..38	Z Position in GPS units
	39..42	X Velocity in GPS units
	43..46	Y Velocity in GPS units
	47..50	Z Velocity in GPS units
	51..58	Clock Bias
	59..66	Clock Drift
	67..70	HFOM
	71..74	VFOM
	75..76	HDOP; resolution: 0.1 units
	77..78	VDOP; resolution: 0.1 units
	79	NAV Mode (see message #20, byte 71 for thedescription)
	80	bits 0..3 : Nb of SV used to compute this solution bits 4..7 : Reserved
	81..83	Reserved
22		This message contains ephemeris data for one Satellite.

Ephemeris Data		It is transmitted at a rate of one message per second until the ephemeris data list completed, and then it is transmitted only if new ephemeris occurs.
	5	bits 0..4 : SV Number bits 5..7 : reserved
	6..77	Ephemeris sub-frame 1-3/words 3-10MSB of byte 6 is the Bit 61 of subframe 1
23	5-6	Reserved
Measurement Block Data (1, 2, 5, 10 Hz)	7	Number of measurement blocks (N)
	8..15	Predicted GPS Time
	16	bits 0..5 : SV # (0..31) bit 6 : reserved bit 7 : Toggle at each Ephemeris Transmission
	17	SNR
	18..21	Code Phase; range : 0 .. 2095103999
	22..25	Integrated Carrier Phase bit 0-1 : 0 : Ready 1 : Phase Unlock 2 : Cycle Slip Detected 3 : Not Ready bits 2-11 : Carrier Phase; range: 0-1023 bits 12-31: Integrated Number of Cycles range: natural roll over
	26	Cycle_Slip Counter Increment by 1 every time a cycle slip is detected during a 10ms period; range: natural roll over
		Measurement block #2 . . Measurement block #N
33	5	bit 0...3: Total number of Satellites in view bit 4..7: reserved
Satellite Visibility Data and Status		Data transmission of up to 12 satellites in view listed in decreasing elevation order. Satellite visibility data of the 1st SV:
	6	Computed data bit map bit 0..4 : SV Number bit 5..6 : SV Status 0 = In View 1 = Tracking 2 = Measurement Ready 3 = Used by Navigation

	bit 7 : Differential Corrections available
7	Elevation;range : -90..90
8-9	Azimuth;range : 0..360
	bits 9-15 : Reserved
10	SNR;range : 0..90
11..15	Satellite visibility data of the 2nd SV
16..20	Satellite visibility data of the 3rd SV
21..25	Satellite visibility data of the 4th SV
26..30	Satellite visibility data of the 5th SV
31..35	Satellite visibility data of the 6th SV
36..40	Satellite visibility data of the 7th SV
41..45	Satellite visibility data of the 8th SV
46..50	Satellite visibility data of the 9th SV
51..55	Satellite visibility data of the 10th SV
56..60	Satellite visibility data of the 11th SV
61..65	Satellite visibility data of the 12th SV

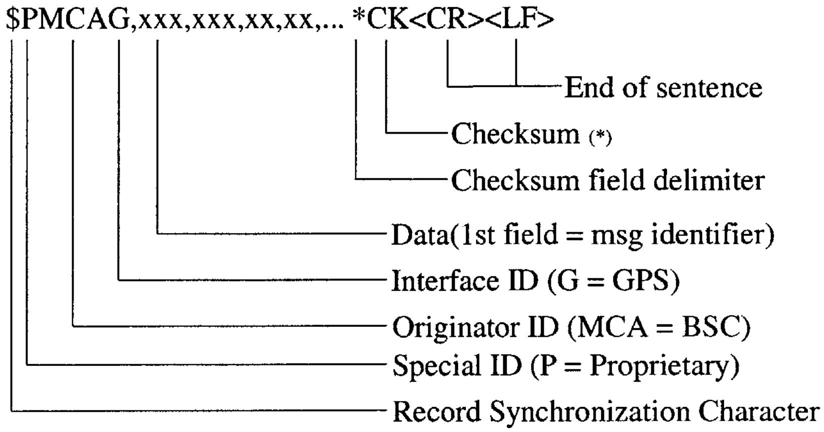
D.2 The Allstar NMEA Messages

In this protocol, the baud rate can be set from 300 to 38400 bps.

D.2.1 The NMEA Message Recognition Procedure

Message Block Structure

Data information is divided into several NMEA approved or CMC proprietary sentences having the following structure:



(*) The checksum is a 8-bit exclusive OR of all characters in the sequence, including "," delimiters, between but not including the "$" and the "*" delimiters.

Figure D.1NMEA Sentences Structure

Out of all the messages sent by the receiver [3], only three of them were used and are described in Appendix D.2.2:

- PMGGA is the main message and is sent at a rate of 5Hz. It is composed of time, position (in the WGS-84 reference frame), HDOP and number of satellites used for solution computation information. The total length of the message is 82 characters.
- PMGSV is sent every second and is composed of up to three different messages giving information on the satellites in view consisting of their azimuth and elevation with reference to the receiver, and of the signal to noise ratio of their signal. The maximum length of a message is 70 characters (*3).
- PMVTG is sent at 5Hz and provides the velocity of the user. The total length of the message is 37 characters.

Message Recognition Procedure

In order to recognize the messages, the code would proceed as follows:

- Recognize the “$” character
- Recognize the type of message received
- Save the incoming data into an array up to the “*” character unless a new “$” arrives before a “*”or unless the message’s length is passed.
- Make sure that the Checksum is correct
- Recognize a set of characters in between “,” delimiters: for this recognition, as each sentence is specifically formatted, the code would make sure that the format of the characters and the number of characters in between “,” is respected.
- From the determined sets of characters, the actual value of the expected data is computed.

The advantage of this method is to reduce the number of possible errors with first making sure that the Checksum is correct and then that the format is respected. The implications are to be sure with a high probability that the received message is correct and that there will not be any bad data input (due to transfer of information) for the controllers. If a piece of data is incorrect, it is not taken into account (which can be handled by the controllers as the minimum required rate is 2.5Hz and as the used rate is 5Hz).

D.2.2 The Different Sequences of the Allstar NMEA Messages

HEADER	CONTENTS OF DATA FIELDS
$GPGGA	

```
,hhmmss.ss,1111.1111,a,yyyyy.yyyy,a,x,xx,xx.x,
 |         |         | |          | | |  |
 |         |         | |          | | |  HDOP
 |         |         | |          | | SVs in use4
 |         |         | |          | Quality indicator3
 |         |         | |          E/W - East or West
 |         |         | Longitude2
 |         |         N/S - North or South
 |         Latitude1
 UTC of position

 ±xxxxx.x,M,xxxx,M,xxxx,xxxx*hh<CR><LF>
 |        | |    | |    |
 |        | |    | |   Differential reference station
 |        | |    | |   ID(0000-1023)
 |        | |    | Age of Differential GPS Data7
 |        | |    Units of geoidal separation, meters
 |        | Geoidal Separation6
 |        Units of antenna altitude, meters
 Altitude5
```

1 Latitude with respect to WGS-84.
- 2 digits of degrees, 2 digits of minutes, 4 digits of decimal fraction of minutes.

2 Longitude with respect to WGS-84.
- 3 digits of degrees, 2 digits of minutes, 4 digits of decimal fraction of minutes

3 GPS Quality indicator,
0 = fix not available or invalid
1 = GPS fix
2 = Differential GPS fix

4 May be different from number in view.

5 Altitude with respect to mean sea level.

6 Geoidal separation: the difference between the WGS-84 earth ellipsoid and mean-sea-level (geoid). "-" mean-sea-level below ellipsoid.

7 Time in seconds since last SC104 Type 1 or 9 update, empty field when DGPS is not used.

```
HEADER      CONTENTS OF DATA FIELDS

$GPGSV
            ,x,x,xx,xx,xx,xxx,xx..........,xx,xx,xxx,xx.x*hh<CR><LF>
             | | |  |  |  |   | |         | |            |
             | | |  |  |  |   | |         | -------------4th SV
             | | |  |  |  |   | ----------2nd - 3rd SV
             | | |  |  |  |   SNR1
             | | |  |  |  Azimuth, degrees2
             | | |  |  Elevation, degrees3
             | | |  Satellite PRN number
             | | Total number of satellites in view
             | Message number, 1 to 3
             Total number of messages, 1 to 3
```

1 SNR (C/No) 00-99 dB, null when not tracking.
2 Azimuth, range 000 to 359 degrees.
3 Elevation, range 00 to 90 degrees.

```
HEADER      CONTENTS OF DATA FIELDS

$GPVTG
            ,xxx.x,T,,,xxx.x,N,xxx.x,K*hh<CR><LF>
             |     |   |     | |     |
             |     |   |     | Speed, km/hr
             |     |   Speed, knots
             Track, degrees True
```

D.3 The Different Parts of the Ephemeris Data Message

The messages from the receiver start with a Telemetry Word and a Handover Word which are not reproduced by the Allstar Ephemeris Data Message.

Telemetry Word (TLM)

TLM word is 30 bits long, occurs every six seconds in the data frame, and is the first word in each subframe. Each TLM word begins with a preamble, followed by 16 reserved bits and six parity bits.

Handover Word (HOW)

The HOW is 30 bits long and is the second word in each subframe, immediately following the TLM word. A HOW occurs every 6 seconds in the data frame. The HOW begins with the 17 MSBs of the time-of-week (TOW) count. (The full TOW count consists of the 19 LSBs of the 29-bit Z-count). These 17 bits correspond to the TOW-count at the 1.5 second epoch which occurs at the start (leading edge) of the next following subframe. Apart from a subframe identification, this word contains a number which, when multiplied by four, gives the time of week (TOW) count for the epoch at the start (leading edge) of the next subframe. The TOW count is a multiple number of 1.5 second intervals since the beginning of the current GPS week (based on UTC).

Broadcast Ephemerides Parameters

For each received ephemeris parameter, a scale factor has to be adapted as summarized in Table D.2:

Parameters	Number of Bits	Scale Factor (LSB)	Effective Range (**)	Units	Name
IODE	8				Issue of Data (Ephemeris)
C_{RS}	16*	2^{-5}		m	Amplitude of the Sine Harmonic Correction Term to the Orbit Radius
Δ_N	16*	2^{-43}		semi-circles/s	Mean Motion Difference from Computed Value
M_0	32*	2^{-31}		semi-circles	Mean Anomaly at Reference Time
C_{UC}	16*	2^{-29}		radians	Amplitude of the Cosine Harmonic Correction Term to the Argument of Latitude
e	32	2^{-33}	0.03		Eccentricity
C_{US}	16*	2^{29}		radians	Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude

Table D.2 GPS Satellites Broadcast Ephemerides Parameters

$\sqrt{A}$	32	2^{-19}		$m^{1/2}$	Square Root of the Semi-Major Axis
t_{oe}	16	2^{4}	604,784	s	Reference Time Ephemeris
C_{IC}	16*	2^{-29}		radians	Amplitude of the Cosine Harmonic Correction Term to the Angle of Inclination
Ω_0	32*	2^{-31}		semi-circles	Longitude of Ascending Node of Orbit Plane at Weekly Epoch
C_{IS}	16*	2^{-29}		radians	Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination
i_0	32*	2^{-31}		semi-circles	Inclination Angle at Reference Time
C_{RC}	16*	2^{-5}		semi-circles	Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius
ω	32*	2^{-31}		m	Argument of Perigee
ω	24*	2^{-43}		semi-circles/s	Rate of Right Ascension
$\dot{i}$	14*	2^{-43}		semi-circles/s	Rate of Inclination Angle

* Parameters so indicated are two's complement, with the sign bit (+ or -) occupying the MSB

** Unless otherwise indicated in this column, effective range is the maximum range attainable with indicated bit allocation and scale factor.

Table D.2 GPS Satellites Broadcast Ephemerides Parameters

Some constants specific to the WGS-84 are also used in order to compute the satellites' Position and velocity:

- μ=3.986005.10^{14}: WGS-84 value of the Earth's universal gravitational parameter
- Ω_e=7.2921151467.10^{-5}: WGS-84 value of the Earth's rotation rate.

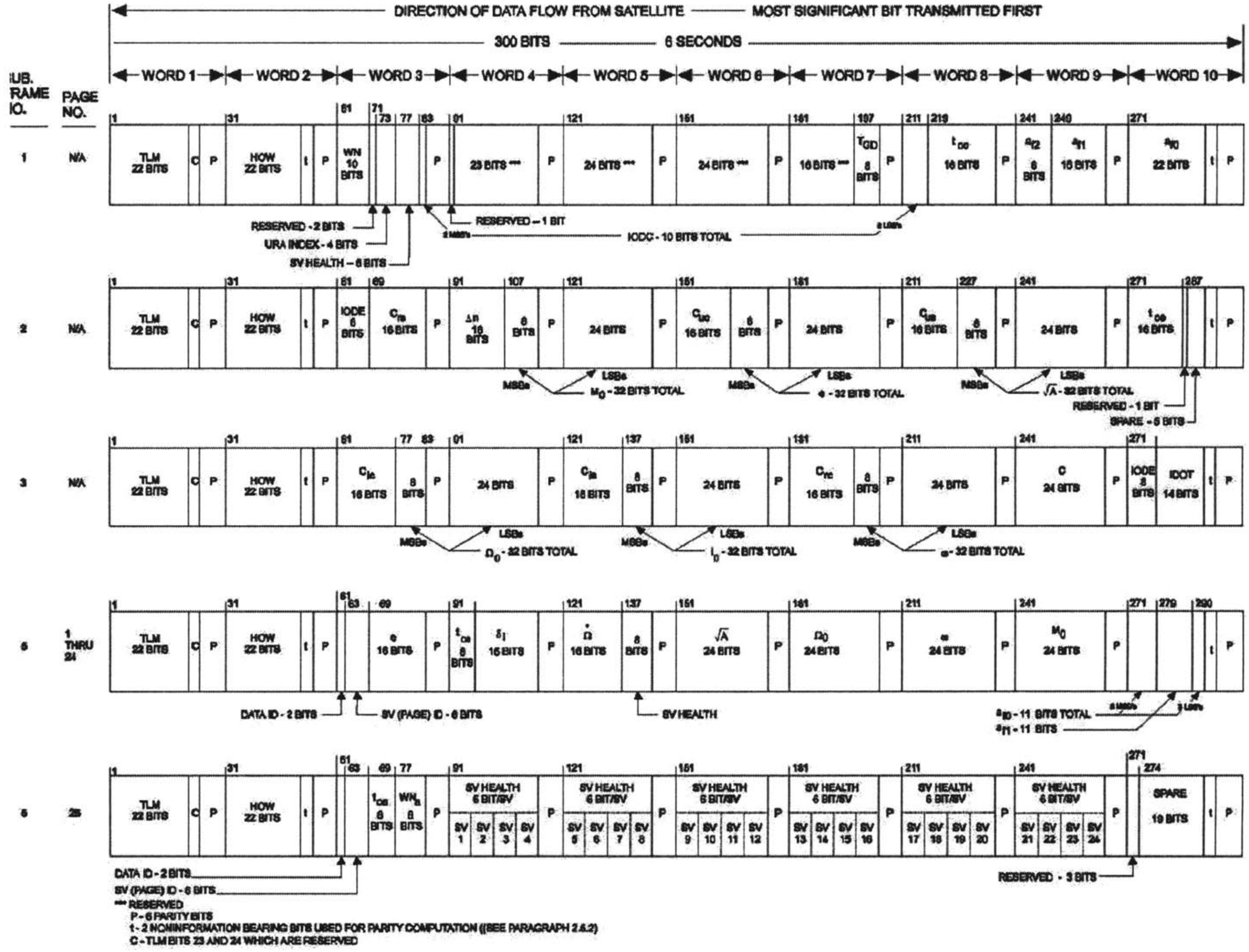


Figure D.2 Ephemeris Data Message

Satellites' Position and Velocity Information Computation

Thanks to the set of equations (eq. D.1) where t is the GPS Time at time of transmission, i.e. corrected for transit time and where t_k shall be the actual time difference between the time epoch t and the time epoch t_{oe} and must account for week crossovers (that is: if t_k is greater than 302,400s, 604,800s shall be subtracted from t_k, and if t_k is less than -302,400s, 604800s shall be added to t_k):

$A = (\sqrt{A})^2$	Semi-major Axis
$n_0 = \sqrt{\frac{\mu}{A^3}}$	Computed mean motion: rad/s
$t_i = t - t_{oe}$	Time from ephemeris reference epoch
$n = n_0 + \Delta n$	Corrected mean motion
$M_i = M_0 + n t_i$	Mean anomaly
$M_i = E_i - e \sin E_i$	Keplers equation for eccentric anomaly solved first by Newton,then by iteration -rad
$\upsilon_i = \operatorname{atan}\left(\frac{\sin \upsilon_i}{\cos \upsilon_i}\right) = \operatorname{atan}\frac{\frac{\sqrt{1-e^2}}{1 - e\cos E_i}\sin E_i}{\frac{\cos E_i - e}{1 - e\cos E_i}}$	True anomaly
$E_i = \operatorname{acos}\left(\frac{e + \cos \upsilon_i}{1 + e\cos \upsilon_i}\right)$	Eccentric anomaly
$\Phi_i = \upsilon_i + \omega$	Argument of latitude
	Second Harmonic Perturbations
$\delta u_i = C_{US}\sin 2\Phi_i + C_{UC}\cos 2\Phi_i$	–Argument of latitude correction
$\delta r_i = C_{RS}\sin 2\Phi_i + C_{RC}\cos 2\Phi_i$	–Radius correction
$\delta i_i = C_{IS}\sin 2\Phi_i + C_{IC}\cos 2\Phi_i$	–Correction to inclination
$u_i = \Phi_i + \delta u_i$	Corrected argument of latitude
$r_i = A(1 - e\cos E_i) + \delta r_i$	Corrected radius
$i_i = i_0 + \delta i_i + IDOT \cdot t_i$	Corrected inclination
$\left.\begin{aligned} x'_i &= r_i \cos u_i \\ y'_i &= r_i \sin u_i \end{aligned}\right\}$	Positions in orbital plane
$\Omega_i = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)t_i - \dot{\Omega}_e t_{oe}$	Corrected longitude of ascending node
$\left.\begin{aligned} x_i &= x'_i \cos\Omega_i - y'_i \cos i_i \sin\Omega_i \\ y_i &= x'_i \sin\Omega_i - y'_i \cos i_i \cos\Omega_i \\ z_i &= y'_i \sin i_i \end{aligned}\right\}$	Earth-Centered, Earth-Fixed Coordinate

(eq. D.1)

Determination of User Geodetic Coordinates

If a position in the ECEF frame is given by the vector $\mathbf{u}=(x_u,y_u,z_u)$, the computation of latitude (ϕ) and height (h) is given by (eq. D.2):

$$r = \sqrt{x_u^2 + y_u^2}$$

$$E^2 = a^2 - b^2$$

$$F = 54b^2 z_u^2$$

$$G = r^2 + (1 - e^2)z_u^2 - e^2E^2$$

$$c = \frac{e^4Fr^2}{G^3}$$

$$s = \sqrt[3]{1 + c + \sqrt{c^2 + 2c}}$$

$$P = \frac{F}{3\left(s + \frac{1}{s} + 1\right)^2 G^2}$$

$$Q = \sqrt{1 + 2e^4P}$$

$$r_0 = -\frac{Pe^2r}{1+Q} + \sqrt{\frac{a^2}{2}\left(1 + \frac{1}{Q}\right) - \frac{P(1-e^2)z_u^2}{Q(1+Q)} - \frac{Pr^2}{2}} \tag*{(eq. D.2)}$$

$$U = \sqrt{(r - e^2r_0)^2 + z_u^2}$$

$$V = \sqrt{(r - e^2r_0)^2 + (1 - e^2)z_u^2}$$

$$z_0 = \frac{b^2z_u}{aV}$$

$$h = U\left(1 - \frac{b^2}{aV}\right)$$

$$\phi = \operatorname{atan}\left(\frac{z_u + \left(\frac{a}{b}e\right)^2 z_0}{r}\right)$$

$$\lambda = \operatorname{atan}\left(\frac{y_u}{x_u}\right)$$

where h is simply the minimum distance between the user and the reference ellipsoid. ϕ is the angle between the ellipsoid normal vector and the projection of this vector into the equatorial (xy) plane.

Appendix

E

Communication Techniques

E.1 Frequency Hopping Spread Spectrum Technique

Frequency hopping uses the data signal and modulates it with a carrier signal that hops from frequency to frequency as a function of time over a wide band of frequencies. With frequency hopping spread spectrum, the carrier frequency changes periodically. The frequency hopping technique reduces interference because an interfering signal from a narrow band system will only affect the spread spectrum signal if both are transmitting at the same frequency at the same time. Thus, the aggregate interference will be very low, resulting in little or no bit errors.

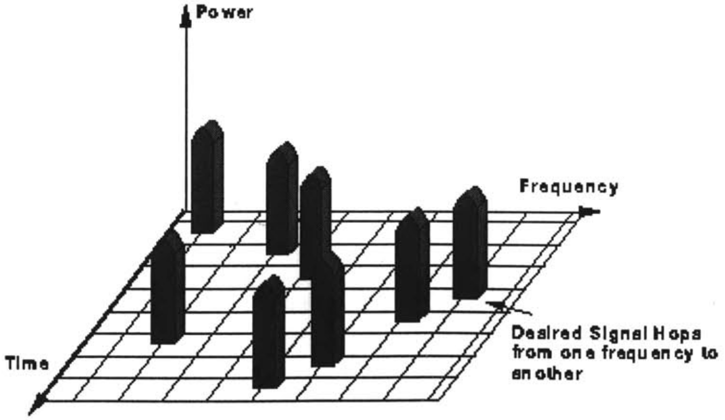


Figure E.1 Illustration of the FHSS Concept

A hopping code determines the frequencies the radio will transmit and in which order (Figure E.1). To properly receive the signal, the receiver must be set to the same hopping code and listen to the incoming signal at the right time and correct frequency. FCC regulations require manufacturers to use 75 or more frequencies per transmission channel with a maximum *dwell time* (the time spent at a particular frequency during any single hop) of 400 ms. If the radio encounters interference on one frequency, then the radio will retransmit the signal on a subsequent hop on another frequency.

Thanks to this technique, data communication can occur at the same time as a detection finder radio built by Thomas Jones which works at 908 MHz using Radiometrix Tx1 and Rx1.

E.2 Maxstream 9XStream Transmission and Reception Protocol

The 9XStream wireless module features several modes of operation that allow the module to be responsive to data and yet utilize minimum power. Figure E.2 shows these modes. In order for the 9XStream module to transition from one mode to another, several time constraining procedures presented on Figure E.2 are explored (The Sleep Mode and Command Mode are not presented in this study as they are not used during the PCUAV tests).

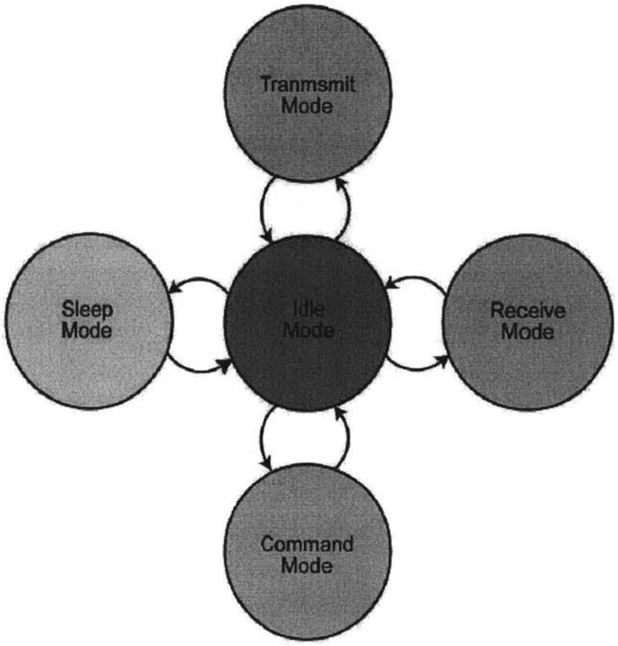


Figure E.2 9XStream Different Modes

The 9XStream module operates in Idle Mode when there is no data being transmitted or received. The module transitions to Transmit Mode once data is presented at the serial port. If valid data is detected at the antenna, the module will switch from Idle Mode to Receive Mode. When no longer transmitting or receiving, the module returns to Idle Mode.

When the first byte arrives in the data buffer, the module leaves Idle Mode and transitions to Transmit Mode. This transition happens instantaneously from the moment the first byte of data arrives in the data buffer. In Transmit Mode, a header is sent out and is followed by the first data packet, which has a CRC (Cyclic Redundancy Check) attached (Figure E.4). The first data packet contains all bytes that accumulated in the data buffer while the header was being sent. After the first data packet is sent, another header will be sent if data is available in the buffer. The header is followed by another data packet. The second data packet (and all subsequent data packets) will consist of data that accumulated in the buffer while the previous data packet and header were being sent (Figure E.3). The size of each data packet can vary up to 64 bytes.

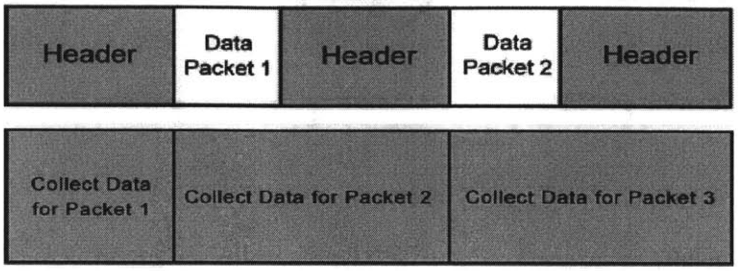


Figure E.3 9XStream Transmission Generation of Data Packets

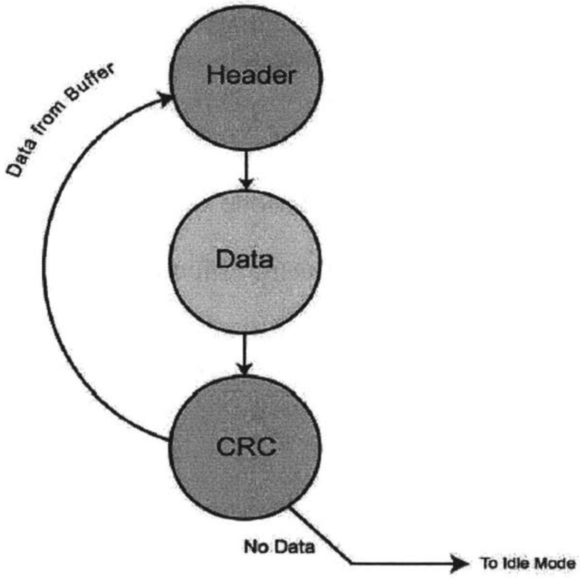


Figure E.4 9XStream Transmission Mode Description

If over-the-air data is present at the RF receiver when the module is in Idle Mode, it will transition to Receive Mode and start receiving packets. Once a packet is received, it goes through a CRC (cyclic redundancy check) to ensure that the data was transmitted correctly. If the CRC data bits on the incoming packet are invalid, the packet is discarded.

If the CRC is valid, the packet is sent to the serial port. This process is shown in Figure E.5. The module will remain in Receive Mode until an error is detected in the received data, or data is no longer transmitted, at which point, the module transitions to Idle Mode. If serial data was stored in the data buffer while the module was in Receive Mode, the data will be transmitted after the module returns to Idle Mode.

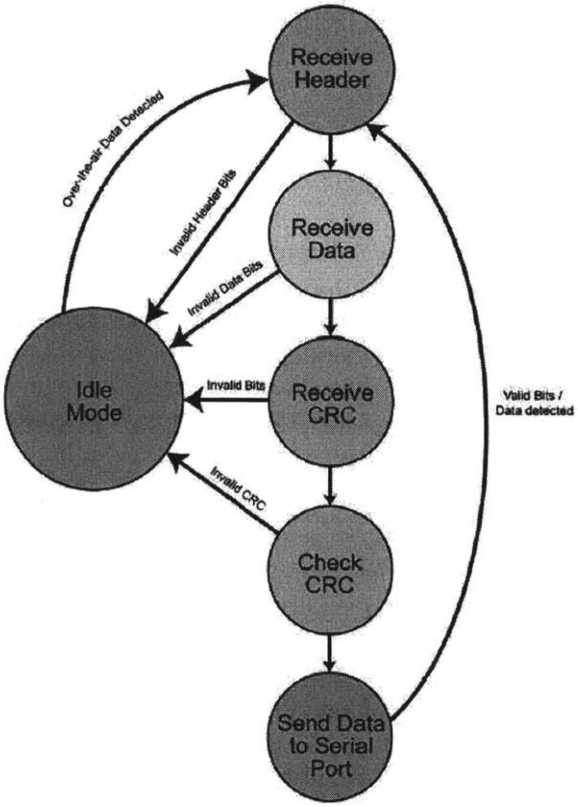


Figure E.5 9XStream Receive Mode Description

E.3 Transceiver Communication: Binary Protocol Messages

The different sets sent by each component of the PCUAV system are described below

E.3.1 Parent's Communication Messages

The different binary messages sent by the Parent are composed of the following pieces of information:

Parent's Message 0: Code Phase Information

The message ID is 30h. Its Length is variable and depends on the Number of Measurement Block: the 5 bytes corresponding to each satellite number and code phase are only sent from byte 21 to byte 21+5*Number_of_measurement_block.

Byte	Format	Name	Characteristics
5-6	Unsigned Integer	GPS Time to transmit	=(int)(10*GPS Time)
7-10	Short Float	Vx	Velocity in the X direction in the ECEF Frame
11-14	Short Float	Vy	Velocity in the Y direction in the ECEF Frame
15-18	Short Float	Vz	Velocity in the Z direction in the ECEF Frame
19	Unsigned Char	SV computed for solution	range: 0..12
20	Unsigned Char	Number of Measurement Blocks	range: 0..12
21	Unsigned Char	Satellite Number	range: 0..32
22-25	Short Float	Code Phase	specific code phase of the satellite
26	Unsigned Char	Satellite Number	range: 0..32
27-30	Short Float	Code Phase	specific code phase of the satellite
31	Unsigned Char	Satellite Number	range: 0..32
32-35	Short Float	Code Phase	specific code phase of the satellite
36	Unsigned Char	Satellite Number	range: 0..32
37-40	Short Float	Code Phase	specific code phase of the satellite
41	Unsigned Char	Satellite Number	range: 0..32
42-45	Short Float	Code Phase	specific code phase of the satellite
46	Unsigned Char	Satellite Number	range: 0..32
47-50	Short Float	Code Phase	specific code phase of the satellite
51	Unsigned Char	Satellite Number	range: 0..32
52-55	Short Float	Code Phase	specific code phase of the satellite
56	Unsigned Char	Satellite Number	range: 0..32
57-60	Short Float	Code Phase	specific code phase of the satellite
61	Unsigned Char	Satellite Number	range: 0..32

Table E.1 Parent's Message 0

62-65	Short Float	Code Phase	specific code phase of the satellite
66	Unsigned Char	Satellite Number	range: 0..32
67-70	Short Float	Code Phase	specific code phase of the satellite
71	Unsigned Char	Satellite Number	range: 0..32
72-75	Short Float	Code Phase	specific code phase of the satellite
76	Unsigned Char	Satellite Number	range: 0..32
77-80	Short Float	Code Phase	specific code phase of the satellite

Table E.1 Parent's Message 0

Parent's Message 3: Position of Reference Information

The message ID is 33h. Its Length is 24 bytes.

Byte	Format	Name	Characteristics
5-12	Long Float	X ref	Position of Reference in the X direction in the ECEF Frame
13-20	Long Float	Y ref	Position of Reference in the Y direction in the ECEF Frame
21-28	Long Float	Z ref	Position of Reference in the Z direction in the ECEF Frame

Table E.2 Parent's Message 3

Parent's Message 4: Acknowledgement Message

The message ID is 34h. Its Length is 1 byte.

Byte	Format	Name	Characteristics
5	Unsigned Char	Ack	Information used to acknowledge requests. Value of "Ack" depends on the message to acknowledge.

Table E.3 Parent's Message 4

Parent's Message 5: Filtered and General Information

The message ID is 35h. Its Length is 40 bytes.

Byte	Format	Name	Characteristics
5-6	Unsigned Integer	GPS Time to transmit	=(int)(10*GPS Time)

Table E.4 Parent's Message 5

7-10	Short Float	pN	Filtered position in the North direction in the Local Frame
11-14	Short Float	pE	Filtered position in the East direction in the Local Frame
15-18	Short Float	pH	Filtered position in the Vertical direction in the Local Frame
19-22	Short Float	vN	Filtered velocity in the North direction in the Local Frame
23-26	Short Float	vE	Filtered velocity in the East direction in the Local Frame
27-30	Short Float	vH	Filtered velocity in the Vertical direction in the Local Frame
31-34	Short Float	Vair	Air Speed
35-38	Short Float	Vair cmd	Air Speed Command
39-42	Short Float	Vgnd cmd	Ground Speed Command
43	Unsigned Char	Satellite Number	range: 0..32
44	Unsigned Char	CIC, Choice, Whois	bit 0 -> CIC bits 1-3 -> CHOICE bits 4-5 -> WHOIS

Table E.4 Parent's Message 5

Parent's Message 7: Center of the Parent's Circle

The message ID is 37h. Its Length is 12 bytes.

Byte	Format	Name	Characteristics
5-8	Short Float	Nc	Center of the Parent's circle position in the North direction in the Local Frame
9-12	Short Float	Ec	Center of the Parent's circle position in the East direction in the Local Frame
13-16	Short Float	Hc	Center of the Parent's circle position in the vertical direction in the Local Frame

Table E.5 Parent's Message 7

Parent's Message 9: IMU Information

The message ID is 39h. Its Length is 24 bytes.

Byte	Format	Name	Characteristics
5-8	Short Float	p	
9-12	Short Float	q	
13-16	Short Float	r	

Table E.6 Parent's Message 9

17-20	Short Float	Ax
21-24	Short Float	Ay
25-28	Short Float	Az

Table E.6 Parent's Message 9

E.3.2 Mini's Communication Messages

The different binary messages sent by the Mini are composed of the following pieces of information:

Mini's Message 0: Filtered and General Information

The message ID is 35h. Its Length is 40 bytes.

Byte	Format	Name	Characteristics
5-6	Unsigned Integer	GPS Time to transmit	=(int)(10*GPS Time)
7-10	Short Float	pN	Filtered position in the North direction in the Local Frame
11-14	Short Float	pE	Filtered position in the East direction in the Local Frame
15-18	Short Float	pH	Filtered position in the Vertical direction in the Local Frame
19-22	Short Float	vN	Filtered velocity in the North direction in the Local Frame
23-26	Short Float	vE	Filtered velocity in the East direction in the Local Frame
27-30	Short Float	vH	Filtered velocity in the Vertical direction in the Local Frame
31-34	Short Float	Vair	Air Speed
35-38	Short Float	Vair cmd	Air Speed Command
39-42	Short Float	Vgnd cmd	Ground Speed Command
43-46	Short Float	N1	Trajectory Point
47-50	Short Float	E1	Trajectory Point
51-54	Short Float	DeltaL	Trajectory Variable
55-56	Unsigned Integer	Satellite Number, Number of Measurement blocks, Size of the Code Phase Matrix (H)	bits 0-4 -> SN: range: 0..32 bits 5-9 -> Meas: range: 0..32 bits 10-14 -> H: range: 0..32
44	Unsigned Char	CIC, Choice, Whois	bit 0 -> CIC bits 1-3 -> CHOICE bits 4-5 -> WHOIS

Table E.1 Mini's Message 0

Mini's Message 4: Acknowledgement Message

The message ID is 34h. Its Length is 1 byte.

Byte	Format	Name	Characteristics
5	Unsigned Char	Ack	Information used to acknowledge requests. Value of "Ack" depends on the message to acknowledge.

Table E.2 Mini's Message 4

E.3.3 Ground Station's Communication Messages

The different binary messages sent by the Ground Sation are composed of the following pieces of information:

Ground Station's Message 0: Position of Reference Information

The message ID is 30h. Its Length is 24 bytes.

Byte	Format	Name	Characteristics
5-12	Long Float	X ref	Position of Reference in the X direction in the ECEF Frame
13-20	Long Float	Y ref	Position of Reference in the Y direction in the ECEF Frame
21-28	Long Float	Z ref	Position of Reference in the Z direction in the ECEF Frame

Table E.1 Ground Station's Message 0

Ground Station's Message 4: Acknowledgement Message

The message ID is 34h. Its Length is 1 byte.

Byte	Format	Name	Characteristics
5	Unsigned Char	Ack	Information used to acknowledge requests. Value of "Ack" depends on the message to acknowledge.

Table E.2 Ground Station's Message 4

E.4 Data Compression Techniques

Different methods have been tested and are exposed next.

E.4.1 Probabilistic Methods.

The Probabilistic Method can't be applied to the problem as data sent are always changing and as the amount of data to transmit is very low (up to 100 bytes) because they require a long header file to link probabilities with data.

However, tests have been performed for random data (floats) with total size up 120 bytes. The results gave a mean of 300 bytes for 100 bytes sent which is too much for our model. Of course, results could have been improved as some data (position, velocity) look alike within time. But the probability that a certain message is too large was too high for the path to be followed.

E.4.2 Lempel-Ziv Method (Improved)

For large codes, the compression ratio can become high. Unfortunately for small codes, as the address number increases the number of bytes of the first bytes, the method is not as efficient.

Tests have been preformed by creating random bits and for 80, 120, 200 bytes and 400 bytes. The mean results where always on the order of the number of bytes tested with small variance.

E.4.3 Lempel-Zif-Welsh (LZW) Method

The idea is to develop a dictionary as data are sent. The dictionary is built at the transmitter and rebuilt at the receiver as data arrive. The difficulties are first that there has to be no errors (because an error propagates in the dictionary creating more and more errors). It also implies that there is no packet loss, or that every transmission is acknowledged, which is too restrictive.

However, if the dictionary would be reconstructed between each transmission, the number of bytes per character is linked to the number of words in the dictionary. It induces that at least 2 bytes per character would have to be sent. Then, the benefits of the method are lost. Tests have shown that once again, the number of bytes after compression is on the same order as before compression.

Appendix

F

The Serial Ports

F.1 The Registers

The different registers permit control over the serial port communication protocol. In Table F.1 DLAB stands for Divisor Latch Access Bit. When DLAB is set to '1' via the line control register, two registers become available and it is possible to set the speed of communications measured in bits per second.

Base Address	DLAB	Read/Write	Abr.	Register Name
+0	=0	Write	-	Transmitter Holding Buffer
	=0	Read	-	Receiver Buffer
	=1	Read/Write	-	Divisor Latch Low Byte
+1	=0	Read/Write	IER	Interrupt Enable Register
	=1	Read/Write	-	Divisor Latch High Byte
+2	-	Read	IIR	Interrupt Identification Register
	-	Write	FCR	FIFO Control Register
+3	-	Read/Write	LCR	Line Control Register
+4	-	Read/Write	MCR	Modem Control Register
+5	-	Read	LSR	Line Status Register
+6	-	Read	MSR	Modem Status Register
+7	-	Read/Write	-	Scratch Register

Table F.1 The Different Serial Port Registers

For communication at 19.2 kbps, the divisor latch high bytes is 0x00 and the low bytes is 0x06 in hexadecimal with DLAB set to 1.

Interrupt Enable Register (IER)

The Interrupt Enable Register could possibly be one of the easiest registers on a UART (Universal Asynchronous Receiver / Transmitter) to understand. Setting bit 0 high enables the Received Data Available Interrupt, which generates an interrupt when the receiving register/FIFO contains data to be read by the CPU. Bit 1 enables Transmit Holding Register Empty Interrupt. This interrupts the CPU when the transmitter buffer is empty. Bit 2 enables the receiver line status interrupt. The UART will interrupt when the receiver line status changes. Likewise for bit 3 which enables the modem status interrupt.

Interrupt Identification Register (IIR)

The interrupt identification register is a read only register. Bits 6 and 7 give status on the FIFO (First In, First Out) buffer. When both bits are "0" no FIFO buffers are active. If bit 7 is active but bit 6 is not active then the UART has it's buffers enabled but are unusable. If both bits are '1' then the FIFO buffers are enabled and fully operational.

Bit 0 shows whether an interrupt has occurred. If an interrupt has occurred it's status will shown by bits 1 and 2. These interrupts work on a priority status. The Line Status Interrupt has the highest Priority, followed by the Data Available Interrupt, then the Transmit Register Empty Interrupt and then the Modem Status Interrupt which has the lowest priority.

First In / First Out Control Register (FCR)

The FIFO register is a write only register. This register is used to control the FIFO buffers.

Line Control Register (LCR)

The Line Control register sets the basic parameters for communication. Bit 7 is the DLAB. Bit 3 controls parity. That is, if it is set to '0' then no parity is used, but if it is set to '1' then parity is used.

Line Status Register (LSR)

Bit 0 shows data ready, which means that a byte has been received by the UART and is at the receiver buffer ready to be read.

F.2 Polling Method Concept and C Code

The method looks at data in the serial port reception buffer and then transmits data in the transmission buffer.

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define PORT1 0x3F8 // Defines Serial Ports Base Address
void main(void) {
    // PORT 1 - Communication Settings
    int c;
    int ch;
    outportb(PORT1 + 1 , 0); // Turn off interrupts - Port1
    outportb(PORT1 + 3 , 0x80); // SET DLAB ON
    outportb(PORT1 + 0 , 0x06); // Set Baud rate - Divisor Latch Low Byte 0x06 = 19,200 BPS
    outportb(PORT1 + 1 , 0x00); // Set Baud rate - Divisor Latch High Byte
    outportb(PORT1 + 3 , 0x03); // 8 Bits, No Parity, 1 Stop Bit
    outportb(PORT1 + 2 , 0xC7); // FIFO Control Register
    outportb(PORT1 + 4 , 0x0B); // Turn on DTR, RTS, and OUT2
    do {
        c = inportb(PORT1 + 5); // Check to see if char has been received.
        if (c & 1) {ch = inportb(PORT1); // If so, then get Char
        printf("%c",ch);} // Print Char to Screen
        if (kbhit()){ch = getch(); // If key pressed, get Char
        outportb(PORT1, ch);} // Send Char to Serial Port
    } while (ch !=27); // Quit when ESC (ASC 27) is pressed
}
```

F.3 Interrupt Handled Method and C Code

The following codes are presented for PORT2 = 0x2F8.

Initialization C Code

INITIALIZE_SERIAL_PORT2 initializes the serial port to interrupt when data is received at the serial port. The Program ran when interrupt occurs is PORT2INTERRUPT.

```
void INITIALIZE_SERIAL_PORT2(void){
    outportb(PORT2+1,0); // Turn off interrupts - Port1
    old2=getvect(INTVECT2); //Save old Interrupt Vector of later
    setvect(INTVECT2,PORT2INTERRUPT); // Set Interrupt Vector Entry
    outportb(PORT2+3,0x80); // SET DLAB ON
    outportb(PORT2+0,0x06); // Set Baud rate - Divisor Latch Low Byte
    outportb(PORT2+1,0x00); // Set Baud rate - Divisor Latch High Byte
    outportb(PORT2+3,0x03); // 8 Bits, No Parity, 1 Stop Bit
    outportb(PORT2+2,0xC7); // FIFO Control Register
    outportb(PORT2+4,0x0B); // Turn on DTR, RTS, and OUT2
    outportb(0x21,(inportb(0x21)&0xF7)); // Set Programmable Interrupt Controller
```

```
    outportb(PORT2+1,0x01); // Interrupt when data received
}
```

Reception C Code

```
void interrupt PORT2INTERRUPT() // Interrupt Service Routine (ISR) for PORT2
{
    int c;
    do {
    c=inportb(PORT2+5);
    if(c&1){
        received_from_transceiver[read_trans_up_to]=inportb(PORT2);
        read_trans_up_to++;
        if(read_trans_up_to>=index_array_max) read_trans_up_to=0;
        }
    }while(c&1);
    outportb(0x20,0x20);
}
```

When an interrupt occurs, data are saved in the array *received_from_transceiver* at the index *read_trans_up_to* that is smaller than *index_array_max*. These data can then be used at any time.

Transmission Set Up C Code

Transmission is a little bit more difficult as the serial port's buffer has not to be full when data are written to it. Data are only sent to the buffer if it is empty.

```
void SET_TRANSMISSION(void)
{
    outportb(PORT2+1,0);      // Turn off interrupts - Port1
    old2=getvect(INTVECT2);   //Save old Interrupt Vector of later
    setvect(INTVECT2,TRANSMIT); // Set Interrupt Vector Entry
    outportb(0x21,(inportb(0x21)&0xF7)); // Set Programmable Interrupt Controller
    outportb(PORT2+1,0x02); // Interrupt when data received and when transmitter buffer is empty
}
```

TRANSMIT is executed when data are received and when transmitter buffer is empty. However, it is not executed as long as data is sent to the transmitter buffer if no data is received. Hence transmission has to be initialized:

```
void START_TRANSMISSION(void)
{
    // transmission_index is equal to zero
```

```
    CAN_I_TRANSMIT=1;
    outportb(PORT2,data_to_transmit[transmission_index]);
     transmission_index++;
}
```

Data to be sent are available in the *data_to_transmit* array, with index from zero to *transmit_up_to*. CAN_I_TRANSMIT is a variable set by the Token Ring like protocol and to make sure that communication occurs at different time specified by the reception of GPS data at the GPS serial port.

Transmission C Code

```
void interrupt TRANSMIT() // Interrupt Service Routine (ISR) for PORT2
{
    int c;
    // THIS IS THE RECEPTION PART
    do {
        c=inportb(PORT2+5);
        if(c&1){
            received_from_transceiver[read_trans_up_to]=inportb(PORT2);
            read_trans_up_to++;
            if(read_trans_up_to>=index_array_max) read_trans_up_to=0;
        }
    }while(c&1);
    // THIS IS THE REAL TRANSMISSION PART
    if(CAN_I_TRANSMIT==1){
         if(transmission_index<transmit_up_to) {
             outportb(PORT2,data_to_transmit[transmission_index]);
             transmission_index++;
             }
        else {
             STOP_TRANSMISSION();
             initialize_regular_data_to_transmit();
             }
        }
    outportb(0x20,0x20);
}
```

Stop Interrupt C Code

Two options can be selected:

- STOP_TRANSMISSION : when only reception is expected
- STOP_SERIAL_PORT2 : to stop the interrupt routines

```
void STOP_TRANSMISSION(void)
{
    outportb(PORT2+1,0);        // Turn off interrupts - Port1
```

```
    old2=getvect(INTVECT2);     //Save old Interrupt Vector of later
    setvect(INTVECT2,PORT2INTERRUPT); // Set Interrupt Vector Entry
    outportb(0x21,(inportb(0x21)&0xF7)); // Set Programmable Interrupt Controller
    outportb(PORT2+1,0x01); // Interrupt when data received and when transmitter buffer is empty
}


void STOP_SERIAL_PORT2(void)
{
    outportb(PORT2+1,0);        // Turn off interrupts - Port1
    outportb(0x21,(inportb(0x21)|0x08)); // MASK IRQ using PIC
    setvect(INTVECT2, old2); // Restore old interrupt vector
}
```

Sending Information to Two Different Serial Ports

This method can be used to send to two different serial ports. Unfortunately, the setting of the interrupt controller by the commands:

```
outportb(0x21,(inportb(0x21)&0xEF)); // for COM1: 0x3F8

outportb(0x21,(inportb(0x21)&0xF7)); // for COM2: 0x2F8
```

can only be used simultaneously due to the same address used: 0x21. Therefore, either different addresses have to be used for different serial ports, or communication has to be performed for via one serial port and then the other one. The second option has been chosen.

4068 - 25