CARRIER DIFFERENTIAL GPS AS A SENSOR FOR AUTOMATIC CONTROL.

DEVELOPMENT OF A FULL STATE ESTIMATION AND FLIGHT CONTROL SYSTEM FOR AN AUTONOMOUS AIRCRAFT BASED ON THE GLOBAL POSITIONING SYSTEM

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Paul Y. Montgomery
August 1996

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bradford W. Parkinson

Bradford W. Parkinson
(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

J. David Powell

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Daniel B. DeBra

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Per Enge

Approved for the University Committee on Graduate Studies:

Abstract

This thesis is about the use of the Global Positioning System (GPS) as a sensor for automatic control. To control a vehicle outdoors in an uncontrolled (non-laboratory) environment. information on orientation and position is needed. Automotive or agricultural vehicles. aircraft and ships are possible mobile examples. Cranes or derricks are possible machine examples. To achieve control. the ability to sense position and orientation accurately. robustly and reliably is paramount. The vehicle investigated here is a small. highly maneuverable. autonomous aircraft. Having three degrees of position and three degrees of orientation freedom. it provides a challenging and suitably general platform on which to demonstrate the concept.

The thesis describes the construction, modeling, hardware and algorithms that were employed to make the autonomous aircraft pictured on the previous page fly a predetermined trajectory (from takeoff to landing) without any form of inertial instrumentation or human intervention. To the knowledge of the author, this is also the *first* time such capability has been demonstrated.

Acknowledgements

I would like to thank my thesis advisor Professor Brad Parkinson. He has influenced and inspired this research in many ways, both direct and indirect. In discussing my work, he has always had insightful comments, helpful ideas and an unerring ability to identify the key elements in complex problems. In managing this project, I have appreciated the latitude he has extended me throughout. I have also benefited from observing his skills in directing the GPS and GP-B programs, particularly his ability to understand people and manage them effectively. I would also like the thank the members of my reading committee: Professor Dan DeBra, Professor Dave Powell and Professor Per Enge. I have profited from their advice, enthusiasm and attitude toward life. Professor Debra was instrumental in inspiring me to come to Stanford, Professor Powell gave me the good advice to pursue something "fun" as well as useful for my thesis and Professor Enge has always been encouraging, cheerful, approachable and a source of thoughtful counsel. I would also like to thank Professor Tom Kane, who chaired my oral defense committee. I thank him for giving me the tools to *do* dynamics.

A debt of gratitude goes to Dr. Andrew Conway and Dr. Hiro Uematsu. I was fortunate to work closely with each of them and have profited tremendously from that interaction. Hiro helped me in making the changes to the receiver tracking loop code to bring out the phase rate estimates which we used to generate a GPS angular velocity measurement.

Andrew and I co-operated closely on many aspects of GPS signal processing. This collaboration saw the creation of code that exists today in both an autonomous helicopter that Andrew subsequently used to win the International Aerial Autonomous

Robotics contest for the first time, and the autonomous aircraft which is the major topic of this dissertation. Andrew supplied a TSR program to handle the serial interface between an IBM type computer and the GPS receivers. I also inherited a real time framework and working position code. We worked together to modify the receiver embedded code to enable a velocity estimate and to implement attitude and angular velocity determination as well as a new integer resolution technique for attitude. Andrew is perhaps the most accomplished individual I have had the pleasure of working with, providing outstanding breadth and depth of insight and learning with a practical ability to implement and problem solve.

Very sincere thanks must go to Dr Andreas Nowatzyk. Andreas was largely responsible for solving the many RFI problems that plagued the implementation of the avionics system. He was always willing to devote his time and experience, and is in no small part, responsible for the zero incidence of failure in the flight critical radio control system. Andreas also helped out during field tests and was responsible for video.

Thanks also to Dr. Brian McGeer, Gup Youngren, Siggy Zerweck and Bob Parks, all of whom were involved in the Aerosonde project that motivated this work. Brian for his help in building the aircraft A/D board and for supplying interface code. Gup for his help in building and debugging the RC switch circuit and assisting with propeller modeling. Siggy and Bob for their broad expertise in the building of model and autonomous aircraft. In this regard much thanks must also go to Dan and Reg at Pec's Hobbies and to the expert pilots Amir Neshati and Ken Gregory, who bore the responsibility of ensuring safe flight and recovery of an expensive aircraft.

Of the many people in the GPS, GP-B and ARL programs who deserve acknowledgement, I can mention only a few. Dr. Clark Cohen, Dr. Boris Pervan, Stu Cobb, Dave Lawrence and Kurt Zimmerman for their frequent assistance and advice. Students Konstantin Gromov, Gabe Elkaim, Jennifer Evans and Sherman Lowe for their great assistance in field and flight testing. Visiting student Vincent Bernatets for his assistance in building a simulink model of the aircraft servo and engine dynamics. Trimble Navigation for supplying discount hardware and access to the embedded code for the TANS receivers. The FAA for financial support under contract DOT/FAA-04

(2-WGP-110).

I also want to thank my friends at Stanford. Graduate school has been a great deal more than equations and computers, and for this I have to thank the many people I have come to know and call friends.

Finally, Birgit's love and support have meant more than I know how to say. She adds dimension and perspective to my life and it is with much anticipation and excitement for our future that I dedicate this thesis to her.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

For automatic control, the sensor employed is always a critical factor. The goal of this work was to use the precise positioning potential of the Global Positioning System (GPS) as the primary sensor for automatic control of a vehicle. In this case the vehicle was a small aircraft. More generally, it could be almost any vehicle which has a largely unobstructed view of the sky (although indoor uses of GPS have now been demonstrated). The use of the GPS carrier signal provides a suitably precise "yardstick" to enable centimeter accuracy in position and sub-degree accuracy in attitude. with corresponding accuracies in velocity and angular velocity. Traditionally, inertial sensors have been used for such measurements. Inertial sensing has an extensive and distinguished history, culminating with such achievements as manned spaceflight and submarine navigation beneath the arctic ice. This level of navigation performance has been achieved only after intensive research and painstaking attention to detail over many decades. For many purposes however, the performance provided by refined inertial instruments is not required. The promise of GPS is to provide much of the functionality of traditional inertial instruments at a fraction of the cost and complexity. Furthermore, since inertial and GPS measurements use fundamentally different physical principals, there is the opportunity to exploit the strengths of each in a marriage that provides an overall capability much greater than the sum of its parts.

The coming decade will see ever increasing use of GPS as a sensor for a wide variety of automation and vehicle control tasks. The ability to inexpensively replace traditional inertial sensors is the key to this endeavor. In the bigger picture, the triad of GPS sensing, detailed geographic databases and wireless telecommunications are the pedestals upon which a huge future industry will rest. The synergy between these three facets is fascinating. Detailed geographic databases will be very expensive to create, and even more expensive to maintain, some regions having a half life of as little as a year. They are vital, since it is not only necessary to know where you are, but where you are in relation to other things. These databases will probably be created using GPS kinematic survey, or potentially some form of GPS guided airborne reconnaissance. The key to maintaining the database will be the use of centimeter-accurate GPS data as a continual crosscheck. When the two do not correspond, either the database is in error, and can be flagged for update, or the sensor is in error, providing vital input for integrity monitoring. Wireless communications will be important to achieve automated database maintenance. For wireless telecommunications, GPS is a key source component. In the space segment, GPS will be used for orbital position and attitude sensing. In the user community, GPS will provide position and distributed time, accurate to a microsecond.

Correspondingly, the advent of robust and ubiquitous wireless telecommunications will be essential for the distribution of GPS correction data, which is necessary to achieve the positioning accuracy and integrity that will be required. Wireless service providers may provide differential correction services for a nominal charge, since they will be growing a much more lucrative business: the business of knowing where things are in real time. At a central headquarters this information has tremendous value. The cellular telecommunications industry will be uniquely positioned to exploit this lucrative opportunity. In the current race for space supremacy in wireless communications, a number of global satellite communications projects are under construction [48], each intending to place approximately fifty satellites into low earth orbit (LEO). Iridium by Motorola, GlobalStar by Loral and Odyssey by TRW and Inmarsat. Placing a *rudimentary* GPS-like ranging signal on each of these satellites could greatly

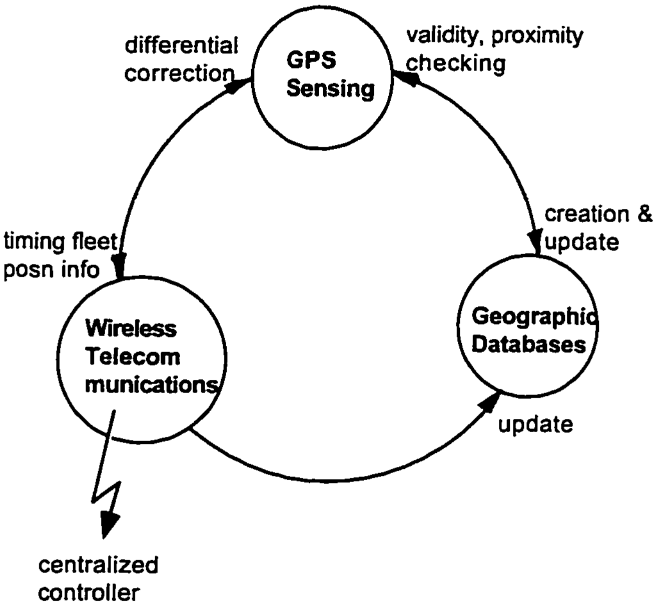


Figure 1.1: **The big picture.**
Synergy between DGPS, geographic databases and wireless communications.

aid the geometric strength and redundancy in terrestrial differential positioning, simultaneously providing much faster static survey or integer resolution. Economic viability remains an open question, but communications satellites, being much lower than GPS orbit may be able to leverage the large number of visible GPS satellites to avoid the necessity for an onboard atomic standard.

1.2 GPS for automatic control : History

The history of GPS as a *sensor for automatic control* is to date, short. Using Trimble TANS receivers, three projects at Stanford have sought to exploit the carrier phase of the GPS signal and the opportunity it presents for sensing to sub-centimeter accuracy in position and sub-degree accuracy in orientation.

In 1995, Zimmerman [63] [64], used GPS carrier phase signals generated from low power indoor pseudo-satellites (pseudolites) to control the two dimensional position and one axis orientation of a free-floating robot. This work demonstrated indoor applications of GPS for the first time and required solution of many additional problems related to multipath, non-planar wavefront geometry and signal strength variations. Also in 1995, Conway et al. [12] used GPS to stabilize a model helicopter in hover (with inner loop stability augmentation using a rate gyro), controlling its position and attitude. The third project, an autonomous aircraft, is the subject of this dissertation. Due to the close collaboration I enjoyed with Dr. Conway, there remains much commonality between the aircraft and helicopter signal processing software. Where joint contributions are justified, these are noted together with the algorithm descriptions that follow.

The application of automatic control techniques to enhance aircraft stability or to achieve closed loop regulation of an aircraft is *not* a new or novel contribution. This subject has been widely studied and an extensive literature exists. Typically, some form of inertial instrument is used for stability augmentation, and some form of inertial *navigation* is employed when accurate position or attitude is required. Recent work by McGeer [32] is an example of aircraft stability augmentation with a minimum of inertial sensors. The aircraft he describes (a small autonomous aircraft intended for remote meteorological soundings) achieves flight stabilization using air data sensors (alpha beta and pitot) and a single axis vibrating gyroscope to sense yaw rate. Slow outer control loops (waypoint tracking) are closed using *uncorrected* GPS code phase and GPS velocity measurements sampled at 1 Hz. These sensors are adequate for stability augmentation and waypoint tracking, however do not provide the accuracy or integrity required for a precision automatic landing. Furthermore, without a GPS or inertial measurement of attitude, one cannot achieve a direct measurement of the wind vector. McGeer proposes a method to estimate the average wind over a period of approximately 30 seconds by flying wind-finding maneuvers [33].

Work by Cohen et al. [42] might also be considered an application of GPS as a sensor for automatic control. In this research, centimeter accurate position measurements based on carrier differential GPS were used to emulate the localizer and glideslope beams provided by the (airport based) instrument landing system (ILS) currently used in commercial aviation for automatic approach and landing. The GPS measurements were used together with data from the inertial navigation unit (INU) on board the test aircraft, a Boeing 737, to perform 110 "autocoupled" approaches. The distinction between "autocoupled" approaches and the work described in this thesis centers on the inclusion of the INU in the 737 flight tests. Broadly speaking, one may consider the GPS components of this system to provide accurate *initialization* of the INU shortly prior to landing. The initialized INU is subsequently used as the *principal* sensor for automatic control through flare and landing[1].

This thesis describes the *first* use of an *integrated* navigation and control system based on (*12 state*) carrier differential GPS and a pitot (airspeed) sensor to fly an aircraft from takeoff to landing. Here, GPS measurements were used to replace (expensive) inertial instrumentation entirely.

1.2.1 GPS Attitude and Position : History

Each of the (three) Stanford projects listed in the previous section was based on extensive earlier work in carrier differential GPS sensing. To provide a historical perspective, a brief history of both GPS attitude and GPS position determination, highlighting earlier aircraft related work is given below.

GPS Attitude Determination

Attitude determination using GPS was originally proposed by Spinney [54] in 1976. The first published attempt to build an attitude capable receiver (based on carrier phase measurements) was in 1983, when Joseph and Deem [23] performed static tests over a single baseline[2]. The first test of attitude determination on an aircraft was performed by Purcell et al. [17], using independent receivers mounted on a DC-8 aircraft. Collected data were used to generate heading information in post-processing. In 1991, van Graas and Braasch [59] from Ohio University performed the first real time three-axis airborne attitude determination at 1 Hz, using a 24 channel receiver.

[1] This distinction is somewhat blurred by the fact that removal of the (emulated) ILS signal will be detected and result in a "missed approach" flag being raised.

[2] A baseline is the vector between two antennae between which a differential measurement is being calculated.

Practical attitude determination using GPS might be said to have reached maturity with the development of the six channel antenna multiplexing receiver described by Cohen [11]. The same receiver, with slightly modified firmware (to enable angular velocity) was used for attitude determination in this project. The receiver uses antenna multiplexing to achieve a high level of integration with a minimum of hardware. Antenna multiplexing also finesses many practical problems encountered with earlier designs. In addition to developing the receiver firmware and a number of algorithms, Cohen presents theoretical limits of performance. In subsequent work [10], Cohen, Parkinson and McNally present experimental flight data comparing GPS attitude with an inertial navigation unit (INU).

GPS angular velocity is a logical extension of GPS attitude, requiring, in effect, differentiation of the attitude signal. This possibility was first suggested by Cohen, and subsequently studied and reduced to practice as described by Uematsu and myself [44]. In this work, a number of alternative strategies for generating an angular velocity estimate were considered and experimental results from each compared with a triad of inexpensive gyroscopes.

Carrier Differential GPS Position

The concept of differential GPS using carrier phase can be traced to independent work by both Counselman and Shapiro [13] and by MacDoran [30]. These researchers extended the existing technique of very long baseline interferometry (VLBI) to GPS for static surveying. Remondi [47] expanded this concept for application to non-static (kinematic) survey. A significant literature on the application of carrier differential GPS to aircraft precision approach and landing has accumulated over the past decade. Klein and Parkinson [27] and Parkinson and Fitzgibbon [35] suggested the use of ground-based GPS pseudo-satellites (*pseudolites*) to improve ranging geometry during approach and landing. Paielli et al. [5] and Romrell et al. [22] independently applied differential carrier phase coupled with "on-the-fly" (OTF) cycle resolution algorithms and presented experimental data from flight tests using transport aircraft. Brown et al. [15] presented preliminary flight test results using carrier differential GPS together with ground-based pseudolites to provide improved redundancy for OTF integer cycle resolution. Cohen (1992) [11] used carrier differential GPS to capture the phugoid motion of a Piper Dakota aircraft. Subsequent work by Cohen, Pervan. Lawrence, Cobb, Powell and Parkinson at Stanford [38] [39] [40] [41] [37] [42] resulted in a reliable technique for cycle ambiguity resolution using pseudolites. This *Integrity Beacon Landing System* (IBLS) has been extensively flight tested and shown capable of meeting stringent ILS specifications for precision landing.

1.3 Thesis Contributions

To demonstrate the utility of GPS as a sensor for automatic control, the *primary* thesis objective was to automate the flight of a small aircraft, using only measurements from GPS and a dynamic pressure (pitot) sensor. The aircraft was to fly from takeoff to landing under automatic control and without any human intervention. Given the history presented in section 1.2, a major contribution has been assembly of a large body of existing knowledge and techniques into an integrated GPS sensor and control system.

In achieving the flight objective the principal problems were:

- *sensing the aircraft position, velocity, attitude and angular velocity*
- *integration of the flight/ground systems components and algorithms*
- *achieving adequate system reliability/robustness*
- *automatic control*

A second but related topic is the question of GPS integration with additional sensors. To date, many papers [58], [57], [49], [56] and [43] have addressed the integration of GPS with conventional inertial units. These publications have usually employed relatively expensive inertial packages. It is the goal of the second part of this work to present a converse perspective: most of the benefits of GPS/inertial integration may also be realized with extremely low cost devices which may today be rendered in silicon and incorporated directly as receiver or antenna components.

In pursuing the above objectives a number of specific research contributions have been realized. These are itemized below:

- *Demonstrated GPS as a viable full state sensor for automatic control.*
  Position, velocity, attitude and angular velocity measurements from GPS were used, unaided by inertial instruments for precise automatic control of a vehicle with six degrees of freedom.

- *Autonomous flight from takeoff to precision landing.*
  Demonstration of fully autonomous flight, from takeoff to landing rollout. This involved a number of subsidiary contributions:

  1. *GPS for angular velocity determination.*
     Two methods for deriving an angular velocity signal from GPS were considered. The superior method was developed, implemented and used in the flight control.

  2. *GPS velocity from differential phase rate.*
     Carrier differential phase rate was used to derive a velocity measurement. This signal was used in the flight control.

  3. *Attitude integer resolution techniques.*
     Two new methods for resolution of attitude cycle integer ambiguities were developed and implemented. A motion based technique for initialization during flight[3] and a static technique for runway initialization prior to takeoff.

- *Wind resolution.*
  Integration of GPS and air data sensors enables resolution of the ambient wind in a direct fashion. Although this capability is frequently employed using *inertial* measurements of attitude and velocity, this is the first report of the same technique using GPS alone. Instrumented in this manner, an aircraft becomes a wind profile sensor, providing vital data on the ambient wind in real time.

[3] Developed in collaboration with Conway [34] who was the principal author.

This capability is significant for *low cost* autonomous aircraft which require high rate/fidelity wind sensing.

- *Integration of GPS and inexpensive piezo-electric gyroscopes.*
  Addition of inexpensive gyroscopes (gyros) can be used to enhance bandwidth and provide substantial robustness for even extreme (aerobatic) maneuvers. Two methods for calibrating gyro biases are considered. The calibrated gyros were shown to be useful for providing attitude for short periods when a GPS solution is impossible. They also provide a method by which earlier data may be weighted into the current attitude estimate.

- *Receiver design for automatic control.* The receivers used in this work are in many respects unsuitable for automatic control. To remedy some of these shortcomings, a modular receiver design intended for vehicle control applications is proposed.

1.4 Quantifying Performance

The statistical steady state (SSS) performance of a system may be characterized by two broad measures. Navigation Sensor Error (NSE), which is a measure of how well the sensor measures truth (given by the statistics of the steady state sensor error), and Flight Technical Error (FTE), which is a measure of how well the steady state control system is able to regulate to a commanded trajectory. The overall SSS performance of the system, Total System Error (TSE), is given by the root sum square (RSS) of these two values, indicating that one can never control better than one can sense. This fact follows from covariance analysis of an *optimal* estimator/controller [8][4].

The dynamic performance of the system is similarly dependent upon the dynamical nature of the sensor and the plant. The overall system may be characterized in terms of system pole locations or alternatively the bandwidth and damping of the slowest system dynamics.

[4] In FAA flight tests, one measures the TSE and the NSE. The FTE is *defined* to be $FTE \equiv \sqrt{TSE^2 - NSE^2}$, even thought many of the assumptions on which the optimal covariance equation is based may be unfounded.

Other important issues for any real world implementation may be specified in terms of accuracy, integrity, availability and continuity. The importance of these concepts is noted, but little attempt will be made to quantify them in this work. Cohen et al. [37] have shown by comparing carrier differential GPS positioning with laser tracking that the two measurements agree to within the accuracy of the laser tracker. Indeed, it seems that carrier differential GPS (CDGPS) provides a way to calibrate the laser tracker rather than vice versa. End around checks are discussed in chapter 5, which, given reasonable assumptions on the nature of an error propagation model are consistent with the experimental results.

I defer to the work of Pervan [45], [3], [4] on *integrity*, noting only that the current experiments were always carried out in a manner that allowed sanity checking by a human observer. The human observer (RC pilot) had the ability to override computer control in the event of a system failure. With this caveat, the integrity issue was largely ignored. It should be noted, however, that no integrity exposures that may be attributed to the space segment were observed during the course of flight testing. Although not implemented in this work, additional onboard sensors such as the barometric altimeter could be used to implement some form of aircraft autonomous integrity monitoring (AAIM) [60], in which additional sensors could be used used to augment standard receiver autonomous integrity monitoring (RAIM) capability. AAIM of this sort enables the detection and isolation of even multiple simultaneous failures.

Availability was an issue of some concern during flight testing. Due to the nature of our integer initialization scheme, the gain pattern of the GPS antennae employed, the number and architecture of channels available and the necessity to bank during flight, operations were confined to times of the day when there were four or more satellites above a 25 degree elevation mask. This constraint could be removed by using antennae with better low elevation gain and/or a modified receiver architecture. Since availability sufficient to demonstrate the objective was achievable with the architecture described herein, this issue was not pursued. Flights were made during the window when satellite configuration was acceptable.

Continuity was not quantified in any direct manner, however continuity events caused by "drop-outs" in the differential communications link were observed. Provided the dropouts did not last more than approximately 4 seconds this did not incur a serious penalty.

1.5 Philosophy

The project involved significant flight testing. This was necessary to validate each of the control laws once basic sensing had been achieved. The approach employed was that of incremental prototyping, each control law primitive was implemented, tested and validated in simulation and then at the field. While successive loop closure has traditionally been employed for autopilot design, the availability of full state information from GPS makes the use of linear quadratic design straightforward flexible and attractive. For these reasons state feedback based on LQR was used for control synthesis. As described above, accuracy, integrity, availability and continuity were pursued to an extent sufficient for demonstration of the flight objective. More work is required to increase integrity and availability before this system can be used with broad utility.

1.6 Thesis Outline

The thesis is broken into two main topics:

- The first topic is the development of the automated flight control system and is covered in chapters 2 through 5. Chapter 2 discusses the air and ground system components and development of a linearized aircraft model. Chapter 3 covers the generation of the fundamental observable by the GPS receiver. Chapter 4 covers GPS signal processing and integer ambiguity techniques for position, velocity, attitude and angular velocity. Chapter 5 discusses control synthesis, the control hierarchy, waypoint architecture, wind estimation and presents flight test results from a number of automated flights.

- The second topic considers the augmentation of GPS with inexpensive alternate sensors. Chapters 6 considers the benefits of augmenting GPS attitude with a set of inexpensive gyros. Methods for calibrating the gyros are discussed as well as propagation of the attitude covariance given a noisy gyro measurement.

- Chapter 7 presents conclusions and suggests a number of areas recommended for future study. In this discussion, a design for an open architecture, modular, multi-antenna, receiver is discussed. The proposed receiver integrates both position and attitude in a package intended for automatic control and ongoing research and development.

Chapter 2

System Configuration and Modeling

2.1 Aircraft description

The aircraft is a heavily modified 12 foot Telemaster model, a conventional high wing design of balsa and plywood construction. The Telemaster design was chosen for its size and predictable flight characteristics, it was also the largest model aircraft that was commercially available. The fuselage was heavily modified to accommodate the avionics equipment, switch harness, batteries and tricycle landing gear. The standard Telemaster is a tail dragger, this was modified to provide a steerable nosewheel, a brake and a robust main landing gear. The entire fuselage was also "beefed up" to accommodate a takeoff weight of 65 lbs, over twice the original specification. The wings and wing struts were also reinforced for this reason. The vertical tail was modified to increase its height and provide the structure for mounting the GPS tail antenna as well as increasing the rudder area. A diagram of the aircraft construction is shown in figure 2.1, and a photograph of the aircraft is shown in figure 2.2.

The batteries are located directly over the landing gear on a sliding tray to allow for CG adjustment. The fuel tank provides sufficient fuel for approximately 40 minutes of flight time at full throttle. The wings and tail are able to be removed with a few screws to allow for transportation, although in practice the tail was rarely removed

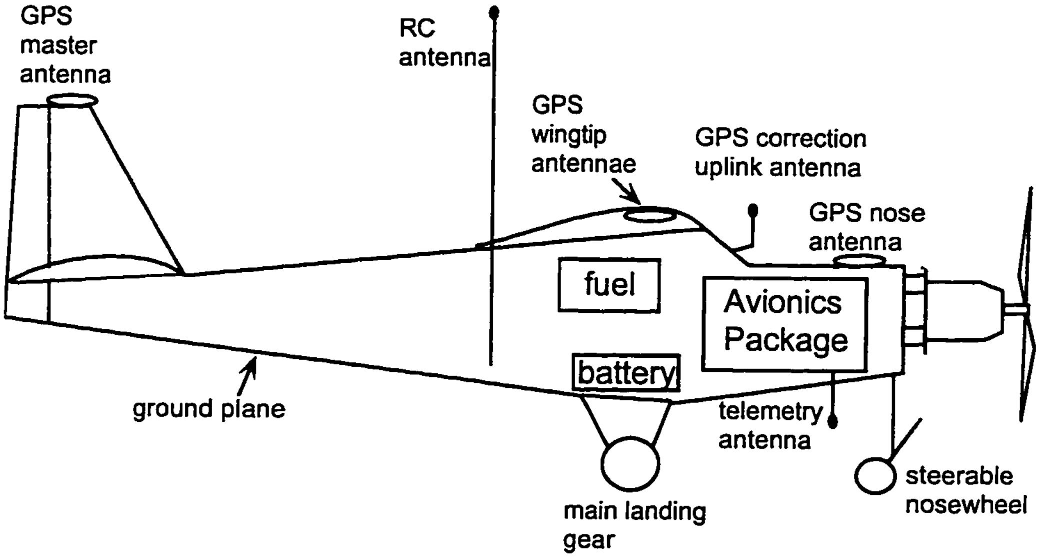


Figure 2.1: **Section through aircraft.**
Aircraft schematic showing the position of various components in the airframe.

and the aircraft was routinely transported on the roof of a car. The aircraft flies at a nominal airspeed of 19 m/s. Stall with zero flap occurs at approximately 13.5 m/s. Generally speaking, the aircraft displays stable flight characteristics, having a wing loading and response approaching that of a light aircraft such as a Cessna 152.

2.1.1 Powerplant

The powerplant is a 3.7 cubic inch, 2 stroke ignition engine from Zenoah, which is started by hand cranking. The propeller is a 22 inch fiberglass design used by model aircraft. The engine is soft-mounted to the fuselage fire wall using four vibration isolation mounts as shown in figure 2.3. The vibration isolation mounts are held in compression by a single tensioning bolt along the crankshaft axis. This design is effective in attenuating transmission of engine vibration to the airframe. The original magneto ignition system was replaced by an electronic ignition in an attempt to reduce radio frequency interference (RFI) coming from the the high tension coil and ignition lead. The ignition lead was custom built from resistance wire. It was carefully shielded

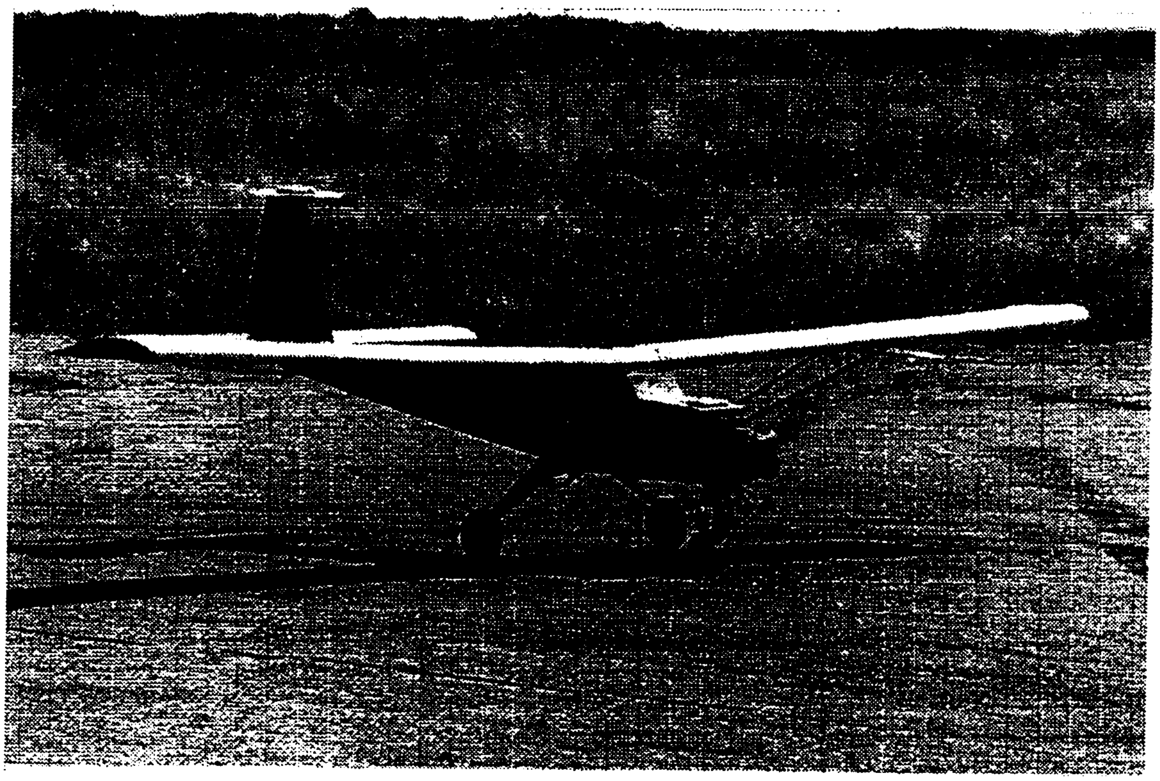


Figure 2.2: **Photograph of aircraft on runway prior to takeoff.**

and grounded to minimize radio interference. The electronic ignition is triggered by a hall effect sensor and mechanically advanced according to throttle position. This system works reasonably well, but has caused problems on a number of occasions. Engine adjustment takes patience, requiring the correct adjustment of high and low speed needles as well as ignition timing and advance sensitivity. It is hard to condense into a few paragraphs the tricks that one acquires after fiddling with these engines for many hours. A systematic approach is tedious but necessary. Above all, one must ALWAYS use caution around the propeller. As an element in the control system, the engine is difficult to characterize. Two stroke engines can demonstrate limit cycle behaviour at partial power, alternating between 2 and 4 cycling. In our case, such behaviour was evident with a period of approximately 2 seconds. This unmodeled behaviour was minimized by adjusting the carburetor and timing settings, but was

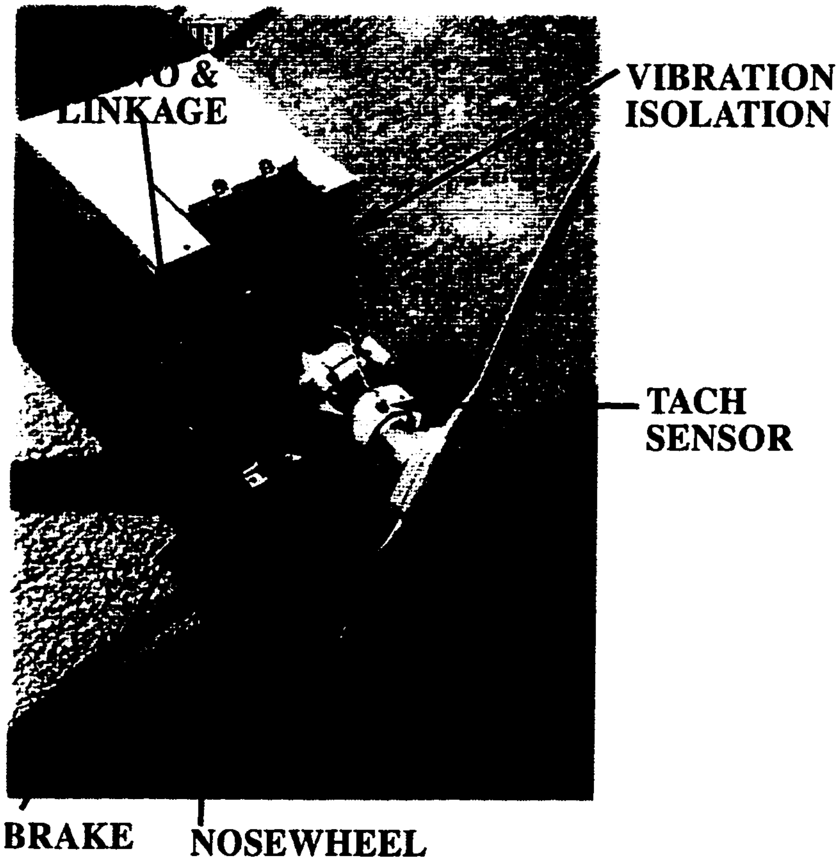


Figure 2.3: **Details of engine, propeller and nosewheel.**

never entirely eliminated on the test stand. Fortunately, the limit cycle mode was not evident during flight if the engine was correctly tuned.

The engine speed response to throttle near idle is strongly nonlinear. This is due to the appeture geometry of the butterfly valve in the carburetor near full closure. This effect was partially re-linearized by utilizing nonlinearity in the mechanical throttle linkage. Another issue worthy of note relates to the mounting of the throttle servo and linkage. The throttle servo should not be mounted on the engine itself due to vibration, and is most conveniently mounted on the firewall as shown in figure 2.3. Soft mounting the engine allows significant engine motion relative to the firewall, which can cause modulation of the fuel supply and erratic engine performance. The linkage actuation was designed to be orthogonal to the axial mode of engine vibration.

to minimize this unwanted feedback. This requirement is contrary to the desire for a short, stiff actuator linkage that will minimize deadband and other linkage nonlinearities. For these reasons, design of the engine mounting, ignition system and throttle linkage were not trivial. Indeed, they are typical of "real world" engineering concerns.

2.1.2 Servos and controls

The aircraft has conventional control surfaces: throttle, elevator, rudder aileron and flaps. The control surfaces are mechanically actuated by short linkages which are arranged for best linearity, servo authority and minimum deadband, while providing adequate control surface motion. The bandwidth of the servo/control surface system is examined in section 2.7.5.

2.1.3 Antennae

Four GPS antennae are mounted on the aircraft, located at the nose, on top of the vertical tail and on each wingtip. Each antenna has an aluminum groundplane with a radius of 15 cm. The tail antenna is used for position and velocity determination. All four antennae are used simultaneously for attitude and angular velocity determination. The aircraft also has three additional antennae. They are the RC receiver antenna and an antenna for each of two radio modems. The RC receiver was tuned to the 6 meter ham radio band. An impedence matching circuit was used to match the RC antenna to the RF front end of the receiver. The bottom of the aircraft is covered in aluminum foil to provide a suitable ground plane.

2.2 Avionics

A schematic of the system hardware is illustrated in figure 2.4. The avionics comprise most of the upper half of this diagram. The avionics were designed to fit into a tray that can be removed as a unit from the aircraft. The tray is vibration isolated from the fuselage using a soft suspension scheme, which provides further attenuation

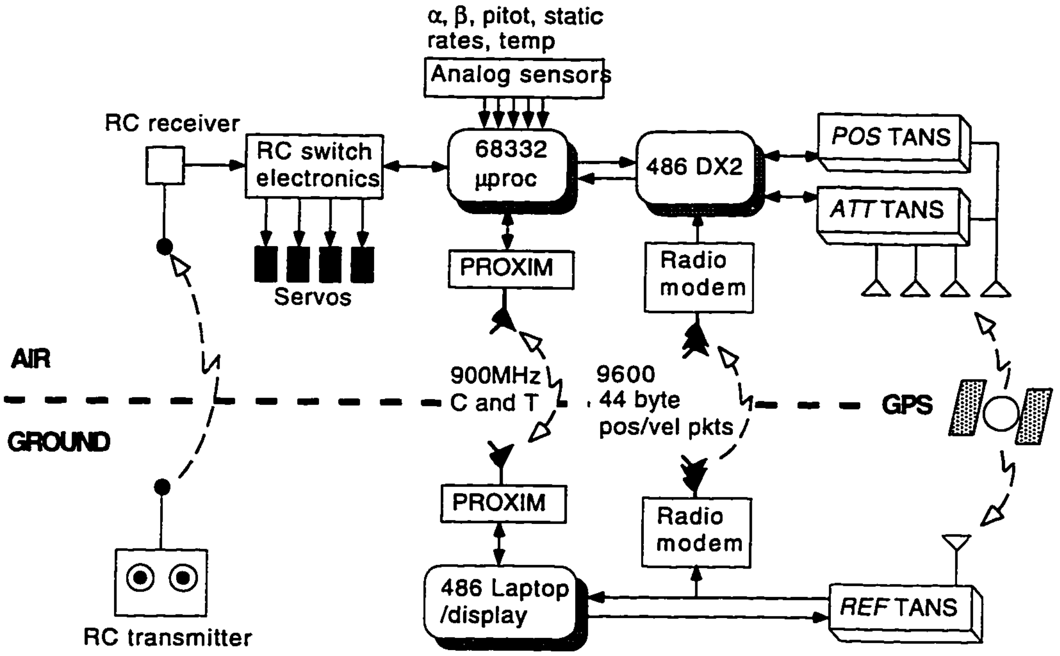


Figure 2.4: **System Schematic.**

of engine vibration at the electronics. A cutaway schematic of the avionics tray and components is illustrated in figure 2.5. This design allows one to connect the various components on the bench before placing the tray into the aircraft. The tray interfaces to the aircraft power harness and data bus via three (standard D type multi-pin) (DB) connectors. These are for servo outputs, analog inputs and power.

2.2.1 GPS receivers

The GPS receivers employed are signal processor boards from Trimble TANS QUADREX units. Two such boards are employed, one for position and the other for attitude. In principle one receiver could be used for both functions. The receivers have an embedded 68000 microprocessor and can be software modified by customizing the onboard ROMs. For the project, each receiver was customized to provide the desired output. The output is serial RS422 and is transmitted to the 486 computer via the communications card at 38400 bps. A discussion of aspects of receiver operation may

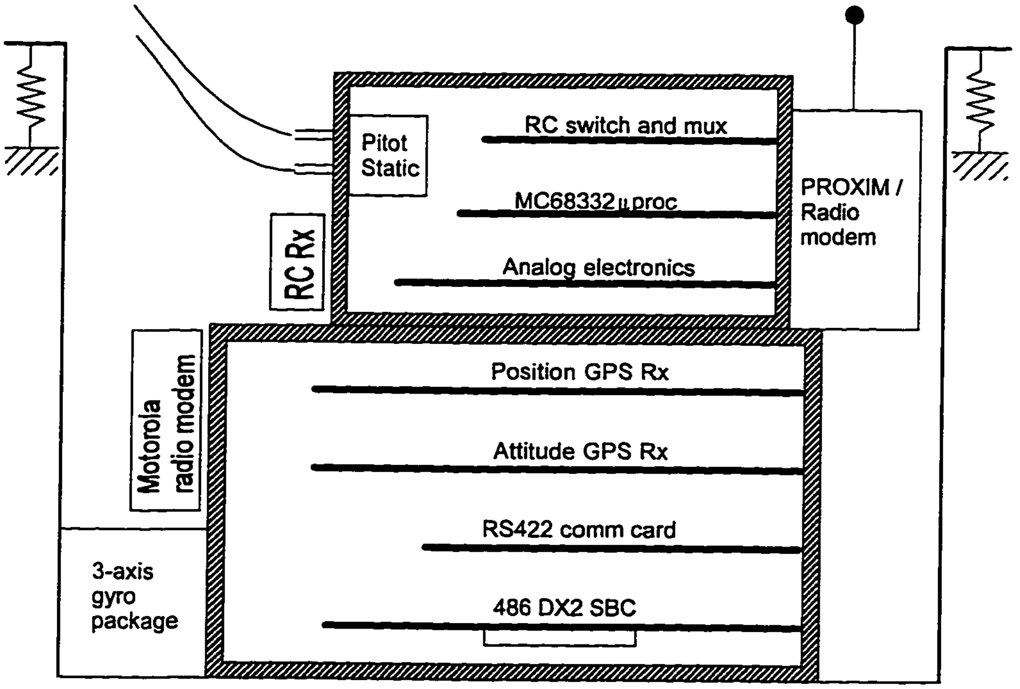


Figure 2.5: **Cross section through Avionics Tray.**

be found in chapter 3. This brief description of carrier phase fundamentals does not do justice to these units, which condense into a few chips an enormous span of human knowledge and technical endeavor. The receivers are installed in a solid aluminum box along with the 486 single board computer (SBC), a communications card, a voltage regulator and an RF splitter. These components are depicted in the lower box in figure 2.5. The box was carefully designed to accommodate the cards and provide the necessary I/O in a mechanically robust, lightweight and RF tight enclosure. Such issues as heat dissipation from the 486, ease of assembly and manufacture were also significant in the design. It was very important to filter the power supplies to prevent the 486 from interfering with the GPS receivers. Providing the ability to access and isolate each element in-situ was also important and necessary for debugging. A photograph of the GPS/486 enclosure prior to final assembly is shown in figure 2.6. On the left of this picture one may see part of the 486 SBC, and the RS-422 card. On the right, one may see the one of the GPS signal processor boards including its

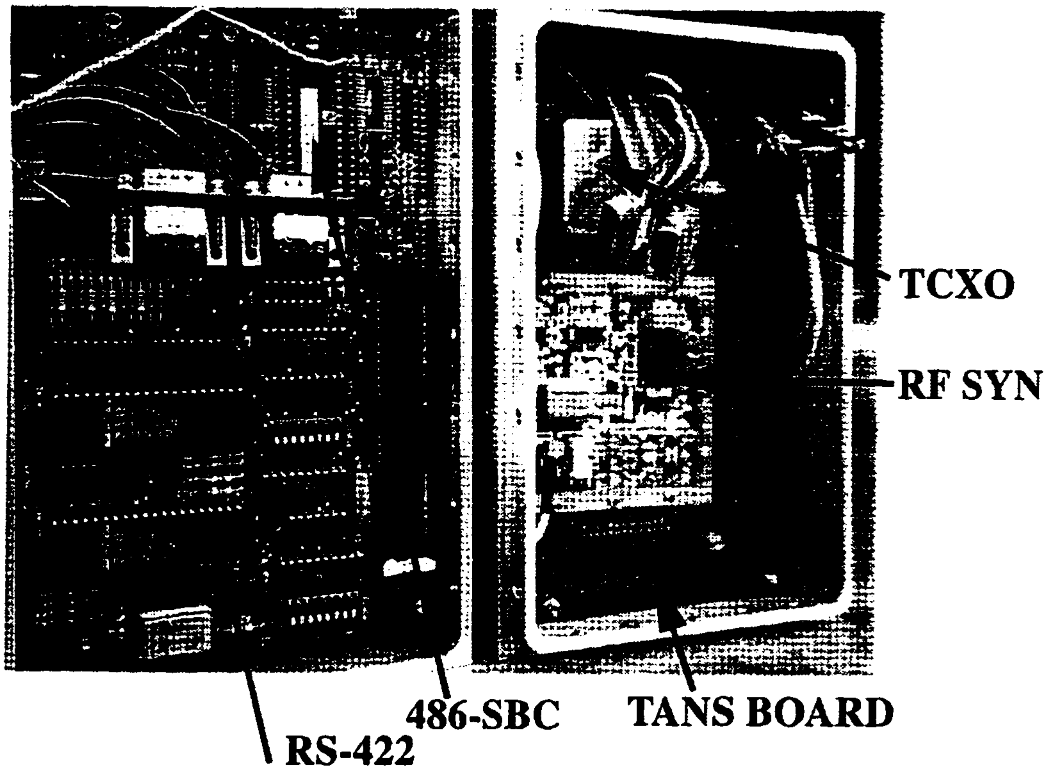


Figure 2.6: **Photograph of GPS and 468 box.**
On the left, the 486 SBC is visible beneath the RS-422 coms card.
On the right, one of the two GPS cards is visible.

TCXO and its RF synthesizer.

2.2.2 486 SBC

A 66MHz 486-DX2 single board computer (SBC) is the main processing engine. It is a standard ISA bus computer without a hard drive or display. EEPROM is used to emulate a floppy drive for program storage and 4 Mb of onboard RAM are available for execution and data storage. The unit provides two 16450 (unbuffered) serial ports and a parallel port. The computer runs DOS version 3.3 in a minimal configuration with most hardware interaction supplied by a custom Terminate-Stay-Resident (TSR) program. Conway [12] provides a detailed account of the TSR. The 486 is responsible for the following broad functions:

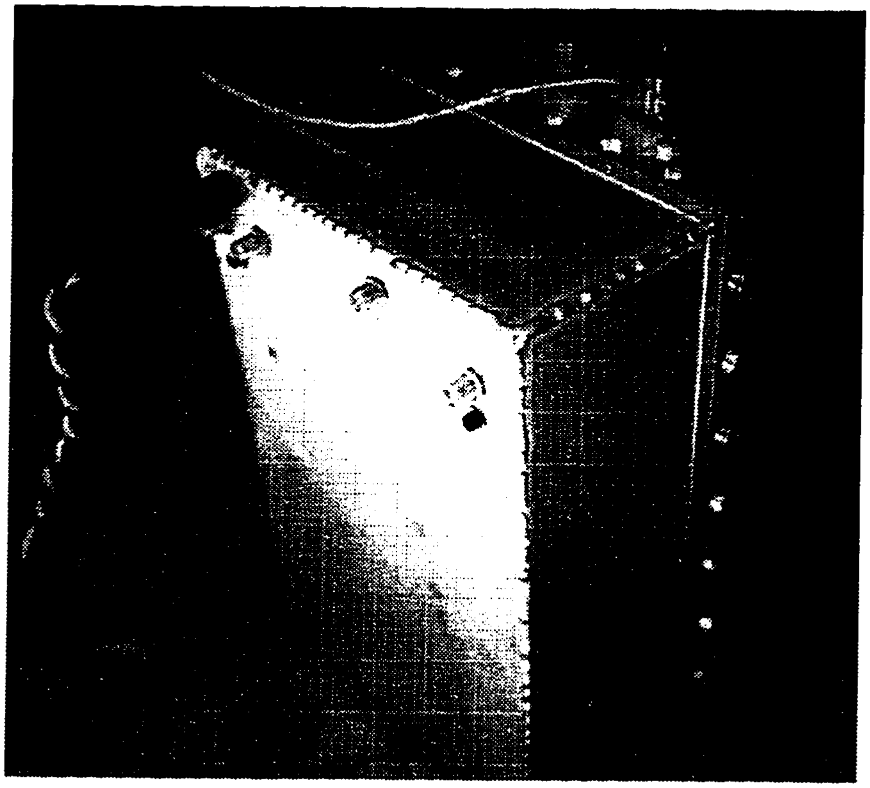


Figure 2.7: **Photograph of GPS and 468 box when sealed**
Power, serial and GPS RF ports are visible.

- Processing raw GPS data from the two onboard receivers and the remote reference receiver to generate position, velocity, attitude and angular velocity estimates at 10 Hz.

- Use the processed GPS data with additional analog and RC input data to generate an appropriate control vector to stabilize and direct the vehicle.

- Store all GPS, analog and RC data to allow for subsequent post-processing and debugging. (After landing, the accumulated flight data is transferred to a remote hard drive over the parallel port.)

- Process and handle all received commands.

The use of a 486 computer as the flight processor greatly facilitated software development and debugging. It enabled an identically configured desktop computer (with display and hard drive) and identical software to be used for development, testing and post-flight debugging. This utility was purchased for a modest increase in program complexity, by allowing flexibility in the data source and sink. This approach effectively "un-embeds" the processor and provides many advantages when compared to a heavily embedded, cross-compiled approach as was necessary for the Tattletale model 7 (TT7) and GPS receiver code development.

2.2.3 TattleTale Model 7 (TT7)

The upper box depicted in figure 2.5 contains the TT7, an electronics board and the RC switch electronics. Once again, the design of the electronics packaging had to be done with care. The TT7 is a small, low power commercial microcontroller board based on a 68332 microprocessor. The 68332 has a CPU32 instruction processing core with many useful controller features. These include a time processor unit (TPU), a Queued Serial Peripheral Interface (QSPI), and a serial communications interface (SCI). The TT7 board also has a realtime clock, a 4 input, 12 bit A-D converter (ADC) and an RS-232 driver. Each of these features is utilized in the project. Initially the TT7 was to be the sole computer for the flight system. Since it became necessary to handle the GPS signal processing, the 486 was subsequently added. This history is the reason for having two flight computers. Future iterations could usefully consolidate them into one computer handling all functions.

Code for the TT7 is written in C or 68000 assembler. The cross-compiled executable is serially downloaded and burned into EEPROM. Debugging was very tedious and time consuming since no In Circuit Emulator (ICE) or other advanced tools were available. Serial output and occasionally an oscilloscope were used for debugging. Each iteration required re-burning the EEPROM, so this type of development was very slow. The TT7 runs interrupt driven code that performs the following broad functions:

- Reads three separate A-D converters to sample measurements of alpha, beta, pitot, static pressure, three gyro axes (see section 2.2.5), temperature and system voltages. A number of additional A-D channels are available which are currently not utilized. Alpha, beta, gyro and pitot measurements are sampled at 20 Hz, other measurements are made at a lower frequency. These functions are handled by the QSPI and onboard A-D converter. The QSPI allows a queue of ADC commands to be preloaded and executed in parallel with the CPU. Relative to polling techniques, this approach greatly reduces the CPU interrupt load.

- Sample the received (multiplexed) radio control signal using the TPU Period Pulse Width Accumulate (PPWA) function. Similar to the QSPI, the TPU allows timing operations to run concurrently with the CPU, greatly reducing the interrupt load on the processor.

- Generate a signal to emulate a (multiplexed) radio control output. This signal may be used to drive the servos if enabled. This function makes uses of the TPU PWM feature.

- Asynchronous serial communications. The TT7 communicates with the ground station via a radio modem from Proxim. The Proxim connects to the 68332's on-chip SCI, allowing full duplex operation at 9600 bps. This is used for downlink of telemetry packets in real time and uplink of command or control data.

- Serial communications between the TT7 and the 486. This Universal Asynchronous Receive/Transmit (UART) is emulated by two TPU lines which allow full duplex operation at 19200 bps. This is used to route analog and RC data to the 486 along with any uplinked packets destined for the 486. The output from the 486 is a control vector destined for the servos.

- A TPU line is used to synchronize the TT7 clock by attaching it to the accurate GPS pulse per second (PPS) output from the position receiver.

- A TPU line (using the PPWA feature) is used to read the engine speed averaged over three revolutions. The input signal is derived from the engine ignition sensor.

- A TPU line is also used to read autopilot status, either enabling or disabling computer output to the servos.

- A final TPU line is devoted to a serial output for debugging. This can be enabled with a #define in the source code and provides a primitive but essential debugging tool.

2.2.4 Electronics

The analog electronics are interfaced to the TT7, allowing the 68332 to drive the data acquisition. The analog board contains two eight channel analog to digital converters (ADC's), which are additional to the 4 input, 12 bit, bus connected ADC onboard the TT7. The off-board converters are a MAX132 and an LTC1290. The MAX132 is a slow but highly accurate 18 bit device and the LTC1290 is a fast 12 bit device. The MAX132 was included since 12-bit resolution would be marginal for measurement of static pressure (ie. altitude) in high altitude operations. These ADC's are handled by the 68332's QSPI, a 4-wire synchronous link that is clocked at 4 MHz. Additional electronics exist to provide 3-pole, low-pass (anti-aliasing) filters on the alpha, beta, pitot and gyro inputs. The anti-aliasing filters have corner frequencies at approximately 3 Hz. Low frequency implies large RC values, which constrains amplifier design. Both the TT7 on-board ADC and the off-board LTC1290 have input impedences in the kΩ range. The source impedence of the 3 Hz RC filter used is too large to connect to one of these inputs directly, and an isolating op amp is necessary. The MAX132 ADC, on the other had, has MΩ inputs and doesn't need the op amp stage. High input impedence is the reason for using the 18-bit ADC to sample voltage-divider signals, even though this is overkill as far as resolution is concerned.

Accuracy The requirements for accuracy are generally modest. The most exacting are those on pitot and static pressures. The pitot measurement is incorporated directly in the longitudinal control laws and also in the wind solution. Gain selection for the analog electronics was chosen to match each sensor for the expected range of operating conditions. This was done to utilize the dynamic range of the ADC while reducing discretization increments to less than the expected sensor noise. For airspeed, a differential pressure (pitot) measurement is required and accuracy on the order of 1 Pa was desired. For static pressure an absolute pressure measurement must be used. The objective was to obtain an accuracy of a few Pa, corresponding to better than 1 m in altitude near sea level. The system achieves a comparable level of precision. Long term drift, (ie, timescales on the order of ten minutes) are several times larger, but are easy to estimate when coupled with the GPS data. The gyro data are similarly subject to bias drift with timescales on the order of minutes.

Integration issues

In systems integration of the electronics, the issue of RFI stands out as worthy of mention. Radio interference was a major issue since it was pronounced and could definitely cause the airplane to crash. The initial avionics packaging was naive in this respect, and required a total repackaging to solve the problem.

The radio control (RC) receiver on board the airplane allows control by a remote human pilot. This link is critical, and failure will cause the aircraft to crash. The receiver is sensitive to frequencies in its passband. If spurious signals are present in this band, the transmitted signal will be corrupted according to the relative power in the signals, as quantified by the signal to noise ratio (SNR). The existence of such spurious (noise) signals is evident in the servos, which respond in an erratic manner. The magnitude of the erratic motion is a measure of the ambient noise. Computers and other electronic equipment are sources of RFI, having various (often broad) spectral content. Since the noise sources are local to the RC receiver, the SNR deteriorates as the aircraft flies away from the transmitter. For this reason it is impractical to try to overcome the noise simply by overpowering it. Consequently, the noise must be prevented from reaching the RC receiver.

For broad spectrum RFI, the susceptibility of the receiver can be reduced by limiting its pass band. This was partially addressed by impedence matching the RC antenna to the receiver front end. There are a number of mechanisms by which RFI can be transmitted, additional to the design pathway through the RF front end.

These mechanisms are capacitive coupling and inductive coupling or via radiation across unintended antennae. One must remove all paths to eliminate the problem. Much effort to eliminate one suspected transmission pathway will be for nought if there is another pathway or mechanism at work in parallel. To do a thorough job, RFI issues must be addressed in the design, not as an afterthought. Ultimately the RFI problem was solved by careful shielding, filtering and grounding. Filtered DB connectors were used for all I/O. All power supplies were heavily filtered or entirely isolated. Even using separate battery packs may not work in this context if the packs are placed close together since capacitive coupling between the batteries can result. This perhaps illustrates why tracking down and eliminating all transmission pathways can be exasperating, particularly in a highly integrated assembly. Similar problems were encountered inside the GPS receiver housing. One problem was an intermittent loss of GPS signal. This was eventually traced to a coupling between the receiver and the 486 power supplies. The problem would occur when certain commands caused the 486 to became briefly loaded, resulting in an increased current drain for a short period. The problem looked like a software bug and it was difficult to trace its source to a power supply coupling. In general, all receivers were found to be sensitive to the conditioning of their power supply and this is a useful first step on any trouble shooting checklist.

RC system and Switch

The radio control subsystem is the most critical element in the avionics package. This system is responsible for supplying each of 9 onboard servos with a command signal. A variety of servos are used, each being sized according to its task. The largest servos are used for control surfaces and for steering. Less powerful servos are used for the throttle and brake. The command input to each servo is a pulse width modulation (PWM) signal at approximately 50 Hz. The pulse width is nominally 1.5 ms, with a variation of ± 0.512 ms. One way of handling the servo I/O would be to use a dedicated TPU channel for each signal. This approach would allow any type of RC receiver to be used without modification. Unfortunately, this also consumes most of the available TPU channels. To minimize this proliferation, a multiplexed signal was

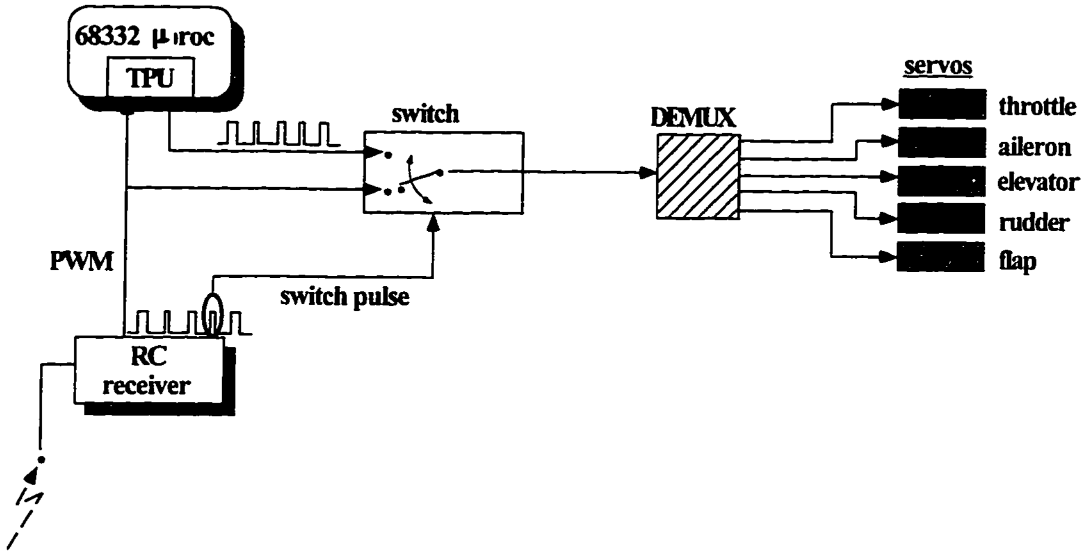


Figure 2.8: **Schematic of RC switch**

PWM signal to servos comes either from computer or directly from the RC receiver output depending on the state of the "switch pulse".

extracted from the RC receiver. This signal, shown in figure 2.8 is a pulse train with a frequency of approximately 50 Hz. Each pulse in the train is one channel of the RC signal. The information in each channel resides in the period between successive rising edges. At the end of each train is a longer framing pulse, which is used for synchronization.

The TPU module of the 68332 was used to

- demultiplex this signal using one TPU input.
- generate an analogous signal using one TPU output.

This reduces the required number of TPU channels from 10 to 2. The signal can be demultiplexed using a single shift register. Which signal is sent to the shift register and ultimately to the servos is determined by a switch circuit, located inside the TT7 enclosure. A one-shot measures the pulse width of the fifth pulse in the pulse train. This *switch pulse* is binary, having either maximum or minimum pulse width. According to the state of this pulse, either the RC signal or the TPU generated signal is routed to the demultiplexer. This feature allows the RC pilot to toggle between computer and manual control with a single switch located on the RC transmitter. The switch is rendered in hardware, ensuring that a computer failure can not prevent the RC pilot from reasserting control.

2.2.5 Analog Sensors: Characteristics and Calibration

The aircraft carries more sensors than are necessary for flight control. This redundancy provides the potential for robustness in case of sensor failure, although aircraft autonomous integrity monitoring (AAIM) and fault isolation of this type has not been implemented.

The air data probe is located on the left wing approximately 1 m from centerline. The probe extends 0.6 m forward of the leading edge and carries the pitot and static vents, and alpha and beta vanes. The pressure sensors are located in the avionics housing, and are pneumatically connected to the vents via flexible tubes that run through the wings. The alpha and beta vanes are directly attached to low friction potentiometers mounted on the shaft of the probe.

Pitot

The pitot sensor is an SCXL004 transducer from Sensym, costing approximately $80. The device is somewhat sensitive to abuse and is therefore mounted within the TT7 enclosure. The pitot provides a measurement of dynamic pressure. Airspeed V (m/s) is determined from the equation

$$\Delta p = \frac{\rho V^2}{2} \tag{2.1}$$

where Δp (Pa) is measured and ρ (kg/m^3) is assumed known. The airspeed measurement is sensitive to noise in the pressure measurement at low speed as given by

$$dV/dp = 1/(\rho V) \tag{2.2}$$

This results in a noisy airspeed measurement while sitting on the runway but is not an issue in flight. The specified device gain of 33.4 μV/Pa was used without additional calibration. The device may be reasonably modeled as unbiased with additive uncorrelated noise. Sampling occurs at 20 Hz and pitot zeroing is achieved by averaging Δp prior to takeoff with the pitot nozzle covered.

Barometric Altitude

The static pressure sensor is an SCX15 transducer from Sensym, costing approximately $50. This is an absolute pressure sensor, from which altitude may be inferred. It has a much greater dynamic range than the differential pressure sensor used for the dynamic pressure (pitot) measurement. This device is also mounted inside the TT7 enclosure. The device gain of 0.716 μ V/Pa was found by calibration using a manometer. The pressure altitude in meters is determined according to

$$\Delta h = \frac{T_0}{C_{lapse}} \left(1 - (p/p_0)^{\frac{RC_{lapse}}{g}} \right) \tag{2.3}$$

This equation may be derived from the ideal gas law, $p = \rho RT$, the hydrostatic law $dp/dh = -\rho g$, and the assumption of linear lapse rate in temperature with altitude. $T = T_0 - C_{lapse}\Delta h$. The lapse rate C_{lapse} was taken to be 6.5 $^\circ C/km$. Sample rate is 5 Hz, and zeroing is determined by averaging measurements prior to takeoff.

Alpha and Beta vanes

The alpha and beta vanes provide measurements of the incident wind angles relative to the aircraft. Alpha is zeroed when aligned with the flat wing bottom, resulting in a nominal alpha of 7 deg in flight. Beta is zeroed when aligned with the fuselage centerline. Potentiometer sensitivity of $\sim$ $0.0029 \Delta R/R_{total}$ /deg was calibrated by direct measurement and is assumed stable. The vanes measure the angle of the incident flow at their location. These are not the same as the free stream incidence angles due to the presence of the aircraft in the flow field. This effect is corrected to first order using the aircraft aerodynamic model. Zero points are determined by averaging alpha and beta measurements prior to takeoff. The major sources of error are assumed to come from zeroing and potentiometer friction. An out the window experiment in a moving car was used for vane sizing and to observe the vane dynamics.

The experiment showed that the vane dynamics are well damped and significantly faster than relevant aircraft modes.

Murata ENV-05A rate gyros

The aircraft is equipped with a set of (three) inexpensive rate gyros, mounted orthogonally and nominally aligned with the aircraft stability axes. The Murata gyros employ piezo-electric vibrating elements which sense angular velocity through coriolis coupling. Output is in the form of a voltage which is sampled using an ADC at 20 Hz. Each axis costs approximately $130. These gyros are low-end in terms of both performance and price. They were initially used to validate the GPS angular velocity solution and were not used for flight control. The published specification for these gyros is as follows:

- max. angular velocity ± 90 deg/s
- scale factor 22.2 mV/(deg/s)
- linearity ±0.5% full scale
- bandwidth 7 Hz
- offset drift 200 mV p-p max. over temp. range
- zero point stability 20mV over 10 minutes max.
- operating temp. range -20 to + 70 ° C
- scale factor stability ± 5% max. over temp. range
- noise level 10 mV RMS max.

A photograph of the gyros mounted on the one meter baseline antenna array is shown in figure 2.9. The gyro scale factors were separately calibrated using a tri-filer suspension oscillator shown in figure 2.10. The period and amplitude of oscillation were individually measured over approximately 20 cycles. The tri-filer suspension is very linear if L/D is large and the oscillation amplitude is kept small. With a linear spring

Figure 2.9: **ENV-05A gyros mounted on attitude test array.** One meter test array was used for attitude and angular velocity testing prior to integration with the flight hardware.

and assumed sinusoidal oscillation, the relationship between amplitude, period and angular velocity is known, and may be compared with the measured gyro data. The calibrated scale factors on each axis was within 2% of specification on the test that was performed.

When mounted on the fuselage itself, the gyros were found to be sensitive to engine vibration at around 2400 rpm, resulting in greatly increased measurement noise. For this reason, the gyro package was ultimately mounted in the avionics tray to benefit from its superior vibration isolation. With this configuration, the engine vibration issue disappeared. The problem of axis alignment is accentuated by this arrangement, since the entire avionics tray can move relative to the fuselage.

Tachometer

The tachometer samples the engine speed at 10 Hz. The measurement is derived from the period required for three engine ignition pulses to occur. (once per rev for a 2-stroke). The RMS noise on the measurement was observed to be approximately 20

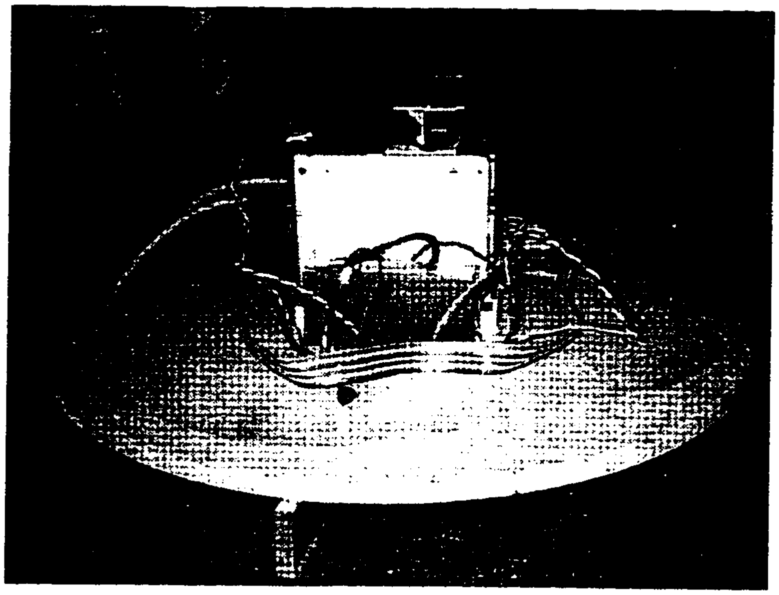


Figure 2.10: **Suspension oscillator for calibrating gyro scale factors.**

rpm. The tachometer data were used in characterization of engine steady state and frequency response out to 2 Hz. The measurement is not directly used in the flight controller.

2.2.6 Batteries

The aircraft has four separate sets of NiCd batteries. These are:

- 6.0 volt supply for the electronic ignition system.
- 4.8 volt supply for the RC receiver and servos.
- 8.4 volt supply for TT7, analog electronics and telemetry modem.
- 7.2 volt supply for the 486, GPS receivers and RNet modem.

Fully charged, the 486 batteries last for approximately 70 minutes. These batteries were strapped to the main landing gear in a position where they can be easily replaced. Other batteries do not need replacement and are fixed inside the aircraft.

A external switch panel enabled each major element in the avionics to be isolated and provides recharging connectors for the various batteries. Where the onboard equipment requires a variety of different voltages, these needs are met by voltage regulators.

2.3 Flight Software

The air software may be logically divided into the GPS receiver software, both attitude and position), the TT7 software and the 486 software. The broad functionality of each is described in sections 2.2.2 and 2.2.3. The GPS receivers and TT7 are both 68000 based embedded devices, programmed in cross-compiled C or assembler. The GPS code is compiled using a compiler and link-loader from Microtec. The TT7 code is compiled and linked using the Aztec C compiler and linker from Manx.

Embedded programming refers to the programming of devices that will ultimately exist in silicon firmware. Such code is often highly interrupted and time critical. Portions of this code, (eg. interrupt service routines) must be written for maximum efficiency. Other code must usually be interruptible and re-entrant. Writing and debugging such code can be time consuming depending upon the tools available. Both the GPS receiver code and the TT7 code run as endless foreground loops, interrupted by background processes according to a predefined set of priorities.

The 486 code is written in C++ and compiled with Borland C++ version 3.1. DOS version 3.3 is employed as the operating system, with most hardware interaction being supported by a custom TSR program, also written in C++. Multiple inheritance and use of virtual functions has been generally avoided. C++ classes have been used where appropriate to encapsulate various parts of the program and to support object abstraction. The flow of data in the program is somewhat complex and differs according to mode of usage. A detailed description of the TSR software that was used for this project can be found in Appendix B of [12]. Overall, the program runs as a foreground loop which continually checks a buffer to see if new data have arrived. If new data have arrived, they are processed according to origin and type. This is usually to update the state and calculate a new control vector. The foreground loop is continually interrupted by serial and timing events which serve to replenish the foreground buffer with data.

2.4 Telemetry

The aircraft has two onboard radio modems. The Proxim has been mentioned above. The second is an Rnet modem from Motorola. The Proxim is used for telemetry downlink at 5 Hz, and for uplink of command and control data. It emits about 500 mW spread-spectrum in the 902-928 MHz band, and has the great attraction of requiring no license from the US FCC. Range is limited, but this was not a problem since all flights remained within a 3 km radius with line of sight.

2.4.1 Uplink

Experiments with the Proxims proved them unsuitable for GPS reference phase uplink, which requires the frequent and latency critical transmission of 42 byte data packets.

Proxim modems employ a relatively sophisticated packetized protocol that contends for the airwaves. The resulting latency depends on the traffic and is variable, depending on error detection and correction. There is significant turnaround time between receiving a packet and being able to transmit a packet. This adds to the latency in these modems if duplex operation is required. Furthermore, the packetizing protocol has large overhead if one desires to transmit a large number of small packets. Finally, the radios do not send continuously, but transmission is triggered only when a buffer is full or special characters are found in the data stream.

The Rnet 9600 radio modems employ a simple mode of operation that makes them ideal for the task of reference phase uplink. They can be continuously keyed and do not introduce packetizing overhead. This means there is effectively no latency associated with the modem itself. For these reasons, Rnet modems at their maximum data rate of 9600 bps were employed. The transmitter transmits about 2 W at 432 MHz. Unfortunately, even at this power, occasional dropouts or signal errors have been observed. This is an important and perhaps surprising observation about differential GPS in general. It has been borne out in every implementation our lab has pursued to date. Simply put: *establishing a reliable, low latency and error free wireless data link over a range of kilometers is non-trivial.*

ARC Protocol

For remote transmission it is important to be able to detect erroneous messages. The packetizing protocol used by the TANS receivers does not provide this capability. so a separate protocol is used for data transmission. This protocol begins with the byte 0x55, and ends with a byte that is the exclusive OR (XOR) of all preceding bytes. followed by the byte 0x0D. Such a packet may be described as follows:

{0x55}{pkt_id}{pkt_len}{data_0}....{data_N}{XOR_byte}{0x0D}

The recipient of such a packet can accumulate the expected XOR_byte and compare it with the received XOR_byte. This provides a means of validating the correctness of a received packet. The probability of validating an incorrect message is acceptably low in practice. The final 0x0D byte is stuffing that significantly reduces the probability of losing two successive packets. It is ignored by the recipient, which is searching for the next 0x55 byte in the data stream. The 0x0D helps if an unbuffered serial connection occasionally drops a single byte. This was a problem with the unbuffered SCI and emulated TPU UART's at high data rates. Further details are documented in the code itself.

2.4.2 Command and Control

There are numerous packet types sent between the ground and air computers. These are used to maintain a common account of the state between the two. Typically, a command originating at the ground station will be sent over the data link and the ground station display updated to indicate to the operator that the command is pending. Upon receipt, the message is handled by the appropriate air computer, which will then respond with an acknowledgement packet indicating the status of the commanded operation. Upon receipt of the acknowledgement, the ground station will again update the display to inform the operator as to the success or failure of the operation. The ground station operator can remotely enable or disable controllers. send altitude or heading commands, or uplink new control laws. Different packet types are identified by their pkt_id byte, and routed to the TT7 or passed through to the 486 accordingly. Packets sent to the 486 can be handled by the TSR or passed on to the executing program. Similarly, telemetry packets originating in the 486 are passed through the TT7 en-route to the ground station.

2.5 Ground Station

The ground components of the flight system are shown below the dashed line in figure 2.4. The ground components consist of the reference GPS receiver, the ground station computer, two radio modems and the RC transmit hardware. Additional GPS ranging sources such as pseudolites or omni-markers are not required. This greatly aids the simplicity of setup and provides flexibility in the choice of flight location. To fly the aircraft in its current configuration, three people are required: the RC pilot, who is nominally on standby, the ground station operator, who is monitoring the state of the aircraft via telemetry and a third person who is pointing the radio modem directional antennae. The directional antennae are probably not required but were used to minimize the probability of telemetry glitches.

2.5.1 Computer

The ground station computer used is a ruggedized IBM PC with three serial ports. This computer was built from off the shelf components and a sunlight readable (transflective) display. One serial port is attached to channel A of the TANS Dual UART (DUART). This provides for receiver initialization and interrogation. The second serial port is attached to channel B of the TANS DUART. The TANS transmit signal on channel B is split to drive serial inputs on both the ground station computer and the differential uplink modem. This allows the ground station operator to monitor the signal strength of each satellite tracked by the reference receiver as it is transmitted over the modem. The third serial port is attached to the Proxim radio modem for receipt of real time telemetry and transmission of command and control data to the aircraft. A parallel port on the ground station computer is used for post-flight download of high density flight data stored in the flight computer RAM.

2.5.2 Antennae

The radio modems both use directional yagi antennae to maximize the antenna gain. Both antennae were mounted to a frame that allows a person to easily track the aircraft.

2.5.3 GPS receiver

The reference GPS receiver uses the same customized code as used in the flight position receiver, with the exception that the packets are formatted according to the ARC protocol and suitable for immediate uplink. The serial format is 9600,N,8,1, chosen to be compatible with the maximum transmission rate of the Rnet modems. Once initialized, 44 byte packets containing phase and phase rate data are output at 10 Hz on channel B. The reference receiver also transmits 12 byte, 0x83 packets at approximately 1 Hz. The 0x83 packets contain the time bias offset calculated from pseudorange positioning and are used for differential position initialization as described in section 4.1.4. All other packets on TANS channel B were suppressed to minimize uplink latency. The antenna for the GPS reference station was mounted at a known location at the airfield, approximately 1.5 m above the ground.

2.5.4 Software

The ground station computer uses the same TSR program used in the flight SBC to handle all serial I/O. The ground station code is written in C, compiled using Borland C++ version 3.1, and uses TSR services to handle serial I/O. DOS version 6.2 was used for the operating system. The code performs the following broad functions:

- Initialization of the reference receiver and the flight computers.
- Display of real time telemetry data from the TT7, the SBC and the onboard GPS receivers.
- Logging of received telemetry data for low fidelity post-flight analysis and playback.
- Generation of command and control law data for uplink.
- Aural and visual warning for "out of range" conditions.
- Real time graphical display of selected telemetry variables.

The code runs as an endless loop, repetitively calling functions in a circular linked list. The linked list approach allows functions to be run in a specific order with each function being executed at a frequency appropriate to its function. It also allows functions to be inserted or removed from execution in response to operator commands.

2.5.5 RC transmitter and amplifier

The RC transmitter used was a JR X-347 seven channel unit at 50.820 MHz. This is an amateur radio frequency reserved for radio control usage. The transmitter was configured to transmit in FM rather than in PCM mode, although PCM was available and is more resistant to noise. The reason for this choice was twofold. Firstly, it was more difficult to extract a multiplexed PWM signal from a PCM receiver. Secondly, the PCM mode, though resistant to noise, exhibits sudden failure as the SNR degrades. In FM mode, servo chatter will gradually increase with decreasing SNR, giving the operator a chance to detect the problem prior to flight. For example: at one point during pre-flight testing, the ignition grounding lead fatigued and broke. When the engine was started, servo chatter was observed and the flight aborted. Using PCM, such an event may not have been detected on the ground. Sufficient range would cause the SNR to drop to the point where PCM lock-up would occur, resulting in likely loss of the aircraft.

As a further precaution, the RC system was fitted with an amplifier to allow increased transmit power in case of sustained RC interference. The transmitter unit was modified to bring the RF signal out on a BNC connector on the back of the transmitter. This was then amplified prior to transmission from a ground mounted antenna. The amplifier was designed to transmit at 2 W, with a capability to boost transmit power to 18 W if enabled. The high power setting was tested to give a solid RC signal out to approximately 30 km. Fortunately, this emergency mode was never needed during flight testing.

2.6 Initialization procedures

In achieving the thesis objective, a total of nearly thirty hours of flight time were accumulated on the aircraft. To date, 6 flights have been fully autonomous. Flights were performed at the SACRC model aircraft field in Fremont California, approximately 30 minutes drive from Stanford. The aircraft and ground systems can be transported in one car and assembled by one person in approximately 40 minutes. Pre-flight checks require a further fifteen minutes if no major problems are encountered. Prior to flight, the aircraft is placed in a known attitude at a known displacement from the reference antenna. Positioning is relative to the reference antenna, so no absolute surveying is required. Flight path and traffic pattern commands are downloaded to the aircraft from the ground station, indicating the desired flight trajectory. The analog sensors are then zeroed by averaging for a number of seconds during GPS signal acquisition. Once GPS lock had been achieved, position and attitude cycle integers are resolved and the alpha beta and pitot sensors unlatched in readiness for flight. Takeoff can then proceed under either automatic or pilot control.

2.7 Aircraft modeling

A math model of the aircraft dynamics is constructed from analysis of aerodynamics, inertia properties, the propeller, the engine and the servos. Ultimately, a linearized model is desired for use in control synthesis. The model was created analytically, and used in all control synthesis. No serious attempt at system or parameter identification was used to validate the model. Inaccuracies in the modeling are adequately handled by feedback control. The use of GPS for system/parameter identification remains a fertile area for ongoing research.

In dynamic modeling, one attempts to capture the *essence* of the dynamics using a minimal state representation. Aircraft dynamics have been well studied and this aircraft is reasonably modeled by a linearized perturbation approach as detailed in [9]. Although aerodynamic analysis results were recorded for two different flight conditions, a single operating point was assumed to avoid the complexities associated with gain scheduling. The aircraft nominally flies straight and level, with bank angle during turns regulated to 25 deg. Airspeed varies between a nominal value of 19 m/s and approximately 13.5 m/s at stall. Using one model across all flight conditions seemed reasonable, provided modeling errors could be handled by a robust control design. It was anticipated that the inclusion of integral states would be necessary to handle steady state tracking errors. For small perturbations about the nominal (straight and level) flight condition, the lateral and longitudinal aircraft dynamics decouple to first order. This great simplification allows the lateral and longitudinal states to be separated, each being represented by a four element state vector. The simplified representation is convenient for analysis, simulation, estimation and control.

2.7.1 Aerodynamic modeling

Aerodynamic modeling was used to generate zero'th and first order approximations to the aerodynamic forces and moments on the aircraft. The zero'th order terms are used to derive trim, and the first order terms are used to derive aircraft stability derivatives. Stability derivatives are partial derivatives of forces and moments with respect to perturbations in aircraft state. This analysis was done using CsDeriv, a mature panel vortex code with help from Dr. Steve Morris. Details of the method may be found in [28]. The method models induced drag but not skin friction drag, so one can expect the results to underestimate damping marginally. The paneling, shown in figure 2.11 was derived from careful measurements of aircraft dimensions,

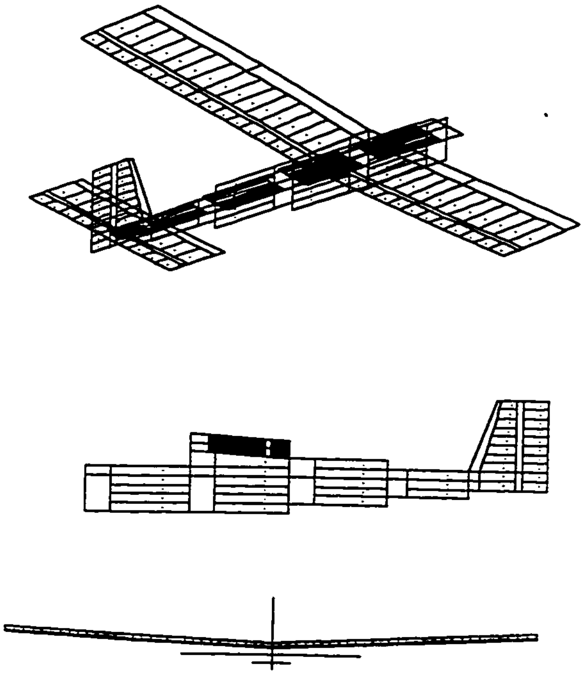


Figure 2.11: **Paneling for aerodynamic modeling.**

including all control surfaces and the fuselage. Wing and horizontal tail airfoil sections were taken from the construction blueprints. The resulting CsDeriv output is summarized in table 2.1 in terms of non-dimensional force and moment parameters.

The CsDeriv convention is x aft $(+F_x)$, y out the right wing $(+F_y)$, z up $(+F_z)$, where $F_\bullet$ represents the non-dimensional force component in the $\bullet$ direction. The moment convention is positive for pitch nose up $(+M_m)$, yaw nose right $(+M_n)$, roll right wing down $(+M_r)$, where $M_\bullet$ represents the non-dimensional moment component in the $\bullet$ direction. The tabulation gives partials of these non-dimensional force and moment components with respect to perturbations in the state (upper matrix) or the control surfaces (lower matrix). Each column of the upper matrix corresponds to a non-dimensional parameterization of α, β, roll rate $\hat{p}$, pitch rate $\hat{q}$ and yaw rate $\hat{r}$. The dimensionalization is taken directly from Ashley [1]. For example, $\frac{\partial F_x}{\partial \alpha} = -0.9478$, indicating that the non-dimensional force in the F_x direction will decrease with increase in alpha. This perhaps unexpected result is due to so called "leading edge suction". In the lower matrix, each column corresponds to angular perturbation in radians of one control surface. The sign of each column is determined by the convention CsDeriv uses to calculate what it considers a positive control surface deflection. To get the

stability derivative matrix for $\alpha = 7$ deg					
	α	β	$\hat{p}$	$\hat{q}$	$\hat{r}$
F_x	-0.9478	0.0000	0.0000	-0.1427	0.0000
F_y	0.0000	-0.1990	-0.0556	0.0000	0.2412
F_z	4.8640	0.0000	0.0000	7.4576	0.0000
M_r	0.0000	-0.0624	-0.5068	0.0000	0.1594
M_m	-0.9916	0.0000	0.0000	-12.8419	0.0000
M_n	0.0000	0.0063	-0.1284	0.0000	-0.0772

control derivative matrix for $\alpha = 7$ deg								
	flap$_1$	ailer$_1$	elev$_1$	rudder$_1$	rudder$_2$	flap$_2$	ailer$_2$	elev$_2$
F_x	-0.0200	-0.0115	-0.0063	0.0000	0.0000	0.0200	0.0115	0.0063
F_y	-0.0059	-0.0176	0.0327	-0.1495	0.0222	-0.0059	-0.0176	0.0327
F_z	0.5463	0.5251	0.2734	0.0000	0.0000	-0.5463	-0.5251	-0.2734
M_r	-0.0624	-0.1518	-0.0142	-0.0070	0.0009	-0.0624	-0.1518	-0.0142
M_m	0.1793	-0.1395	-0.7047	0.0000	0.0000	-0.1793	0.1395	0.7047
M_n	-0.0189	-0.0009	-0.0138	0.0642	-0.0096	-0.0189	-0.0009	-0.0138

Table 2.1: **CsDeriv Output.**

Non-Dimensional force and moment partial derivatives for state and control perturbations about nominal, for $\alpha = 7$ deg. The forces and moments are non-dimensional. The control surface movements are in units of radians. Where more than one panel has been used for a control surface these are subscripted 1 and 2. For example, there are two aileron panels, one on each wing. Each sub-panel provides force and moment contributions.

overall derivative for any control, it is necessary to add the columns from contributing surfaces. For example, if positive aileron is defined to give a roll rate to the right, then column 2 should be subtracted from column 7 to find the force and moment partials due to a positive aileron input. Note that the signs of the partials may be interpreted intuitively, but magnitude of the numbers has little intuitive meaning to the author in this non-dimensional representation.

The convention that is used in control synthesis is shown in figure 2.12. This is standard NASA aircraft convention, X forward, Y out the right wing and Z down. These axes are defined to be straight and level in the nominal flight condition. The conventions used for control synthesis assume that positive elevator gives a positive

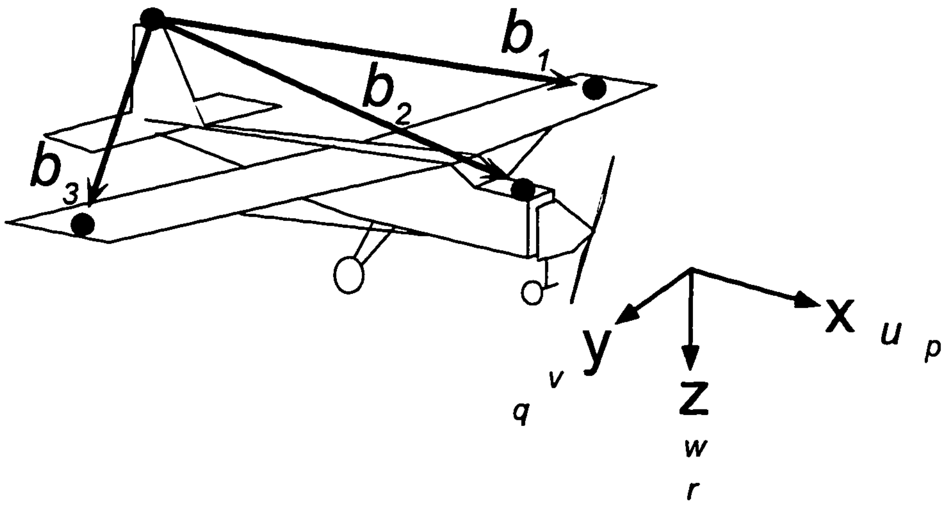


Figure 2.12: **NASA aircraft co-ordinate convention.**
Aircraft basis x, y and z axes, showing also velocity perturbations u, v and w and angular velocity components p, q and r.

pitch rate q, positive aileron gives a positive roll rate p, and positive rudder gives a positive yaw rate r. Once the signs of columns in table 2.1 have been adjusted, each component must be dimensionalized at the nominal flight conditions using aircraft physical dimensions. The nominal flight condition was determined from trim considerations to be $\alpha_0 = 7$ deg and airspeed ($U_0 = 19$ m/s). Ashley [1] describes the dimensionalization process in detail. This is somewhat lengthy and is not repeated here. Three MATLAB scripts were used to automate this process, they are t12alph7.m, t12sd_lat.m and t12sd_lon.m. These files sum, dimensionalize and rotate the contributions from each panel given in table 2.1 to build the elements of the decoupled linear state equations 2.4 and 2.5.

$$\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m - Z_{\dot w} & 0 & 0 \\ 0 & m_{\dot w} & I_{yy} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot u \\ \dot w \\ \dot q \\ \dot \theta \end{bmatrix} = \begin{bmatrix} X_u & X_w & X_q & -mg\,c\theta_0 \\ Z_u & Z_w & mU_0 + Z_q & -mg\,s\theta_0 \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix}$$

$$
+ \begin{bmatrix} 0 & X_{\delta_{thr}} \\ Z_{\delta_{elev}} & 0 \\ M_{\delta_{elev}} & M_{\delta_{thr}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{elev} \\ \delta_{thr} \end{bmatrix} \tag{2.4}
$$

$$
\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & I_{zz} & -I_{xz} & 0 \\ 0 & -I_{xz} & I_{xx} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_r - mU_0 & Y_p & mg\,c\theta_0 \\ N_v & N_r & N_p & 0 \\ L_v & L_r & L_p & 0 \\ 0 & \tan\theta_0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ p \\ \phi \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 & Y_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{ail} \\ \delta_{rud} \end{bmatrix} \tag{2.5}
$$

In the above equations, m and I_{ij} are mass and inertia properties of the aircraft. Terms such as X_u and N_p are dimensionalized stability derivatives. θ_0 is the nominal pitch of the aircraft. Equations 2.4 and 2.5 are easily manipulated to give the standard linear form $\dot{x} = Fx + Gu$. The longitudinal state vector is given by $x_{lon} = [u\,w\,q\,\theta]^T$, representing perturbations from nominal in forward airspeed, downward airspeed, pitch rate and pitch angle respectively. The lateral state vector is given by $x_{lat} = [v\,r\,p\,\phi]^T$, representing perturbations from nominal in sideslip (lateral airspeed), yaw-rate, roll-rate and roll angle. Velocity perturbation states are in meters per second, attitude states are in degrees and angular velocity states are in degrees per second.

A number of additional inputs are required to complete the above model. The mass and inertia terms must be found and the term $X_{\delta_{thr}}$ must be evaluated. Furthermore, the linkage gain for each servo and control surface must be included. The expression $X_{\delta_{thr}}/m$ is the acceleration of the aircraft due to throttle perturbations near the operating point. To evaluate $X_{\delta_{thr}}$ it is necessary to determine propeller thrust as a function of engine speed, and engine speed as a function of throttle pulse width commands.

2.7.2 Propeller modeling

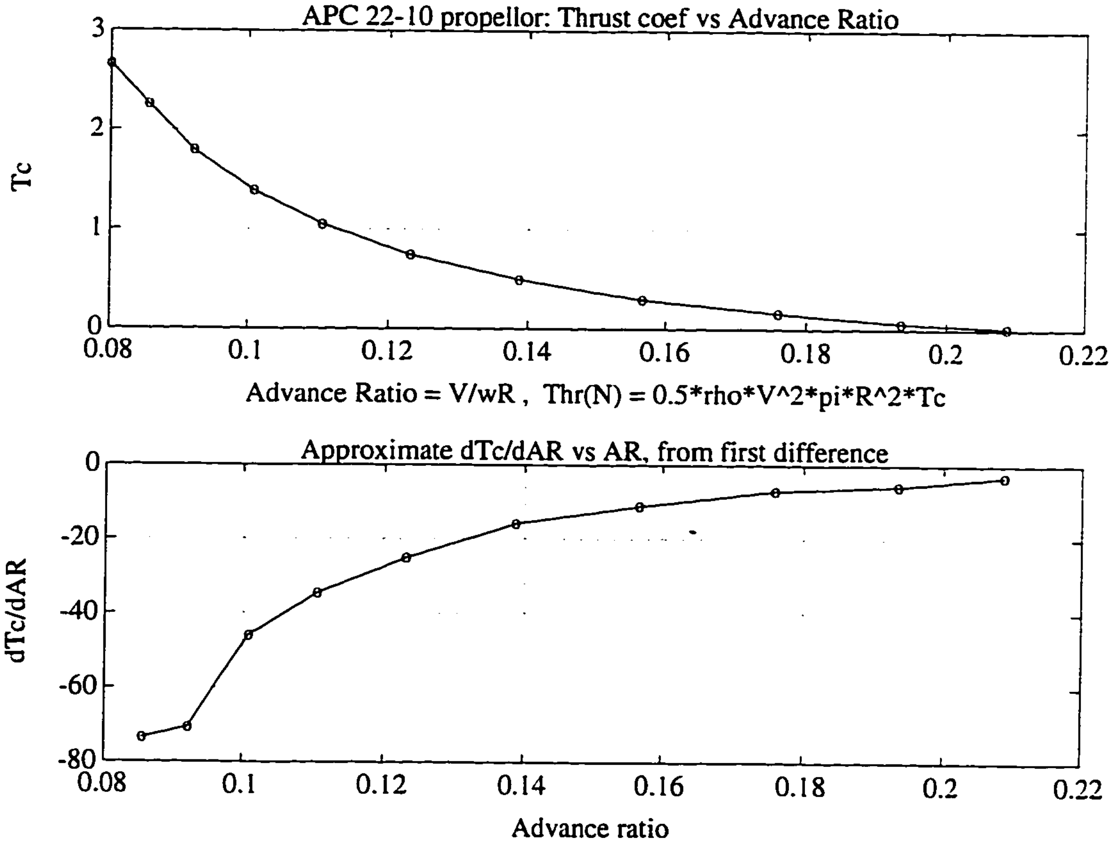


Figure 2.13: Propeller Characteristics.
Propeller specific thrust (N) vs. advance ratio.

The propeller was carefully measured to determine its chord and pitch distribution. This data was then analyzed using a vortex code similar to CsDeriv. The output of the code is plotted in figure 2.13. This data gives thrust coefficient (T_c) as a function of advance ratio (Θ). Advance ratio $\Theta \equiv \frac{U}{\Omega R_{prop}}$ is a non-dimensional ratio of propeller tip speed to flow velocity, where Ω is the engine speed in rad/s. Propeller thrust in Newtons is given by

$$T_{prop} = \frac{1}{2}\rho\left(\pi R_{prop}^2\right)U^2\,T_c(\Theta) \tag{2.6}$$

For small perturbations, we have

$$\delta T_{prop} \equiv \delta_{thr} = \left.\frac{\partial T_{prop}}{\partial \Omega}\right|_0 \delta\Omega + \left.\frac{\partial T_{prop}}{\partial U}\right|_0 u \tag{2.7}$$

It is straightforward to show that

$$\left.\frac{\partial T_{prop}}{\partial \Omega}\right|_0 = -\frac{\rho \Omega_0}{2\pi}\Theta^3 \left(\pi R^2_{prop}\right) \left.\frac{\partial T_c}{\partial \Theta}\right|_0 \tag{2.8}$$

and

$$\left.\frac{\partial T_{prop}}{\partial U}\right|_0 = \rho U_0(\pi R^2_{prop}) \left(T_{c_0} + \frac{1}{2}\Theta_0 \left.\frac{\partial T_c}{\partial \Theta}\right|_0\right) \tag{2.9}$$

$\left.\frac{\partial T_c}{\partial \Theta}\right|_0$ and T_{c_0} are taken from the propeller characteristics shown in figure 2.13. Substituting equation 2.7 into equation 2.4 changes the control vector from δ_{thr} to $\delta\Omega$, ie the control becomes engine speed rather than propeller thrust. Equation 2.9 is included in the state matrix and adds propeller damping to the phugoid mode. For a number of different airspeeds, propeller thrust vs. engine speed is depicted in figure 2.14. The plot shows that the linearization in equation 2.7 is reasonable across the operating range of the aircraft.

2.7.3 Mass and inertia modeling

Aircraft mass and CG were determined by static weighing. Determining the inertia properties was much more difficult. Initially an approximation was made by weighting each piece of the aircraft and measuring its vector from CG. This approach became overly complex as many components have irregular shapes and unknown inertia. Furthermore, the aircraft was continually being altered during development. A suspension oscillator was ultimately used to make experimental estimates of the inertia properties. The suspension spring constant is easily calculated and known to be linear for small angles. The experimental procedure involved averaging the period of oscillation over many cycles. Inertia was then determined based on the assumption of a simple oscillator. Aerodynamic surfaces that would damp the oscillation were removed and replaced with equivalent masses at the correct CG displacement. Inertia values for the three principal axes were then determined. Inertia cross products were assumed zero since there was no convenient method to measure them. It is likely that the inertia estimates represent the largest error sources in the modeling and may be in error by as much as 10%. Simple sum square checks on the inertias indicate that the results are not unreasonable. Control laws were simulated with ± 10% variation in

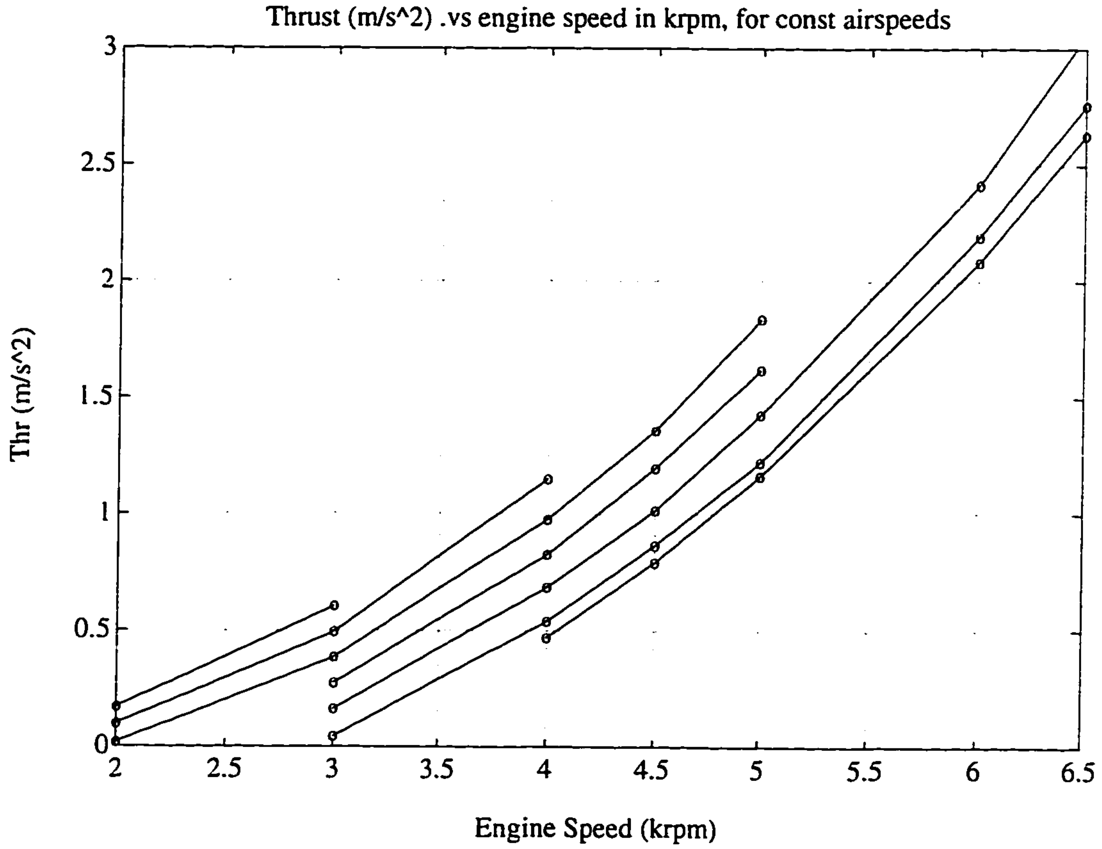


Figure 2.14: **Propeller thrust vs. engine rpm**

Lines are loci of constant airspeed, with airspeed varying from 14 m/s (lowest curve) to 19 m/s (uppermost curve) in steps of 1 m/s.

principal inertias to ensure that the control laws would be robust to modeling errors attributable to inertia modeling.

2.7.4 Engine Identification

With a known propeller characteristic, thrust can be controlled using engine speed. To understand engine performance, it is necessary to characterize engine response from throttle to engine speed near the nominal operating point. As discussed in section 2.1.1, a two stroke engine is a complex machine that is non-trivial to model. Numerous experiments were made in attempts to characterize the engine and get it

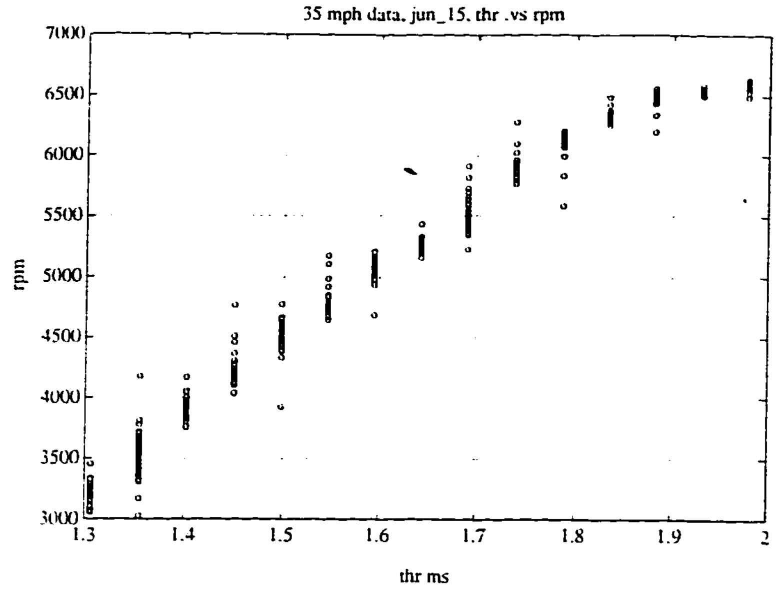


Figure 2.15: **Steady state rpm vs throttle.**

to run reliably and smoothly. The experiments involved strapping the aircraft to the top of a car and driving down the road at nominal airspeed with the engine running. A steady state plot of engine speed as a function of throttle from such an experiment is depicted in figure 2.15. This shows that above 3000 rpm, the steady state engine response is relatively linear, with saturation occurring near maximum throttle. Flight experiments showed nominal engine rpm to be approximately 4500 rpm. Data were also collected from single frequency-at-a-time experiments. The TT7 was programmed to oscillate the throttle about nominal at frequencies varying from 0.1 Hz through 2 Hz. The input/output phase and magnitude data are plotted in figure 2.16. In a separate flight experiment, the engine was manually excited over a similar range of frequencies by the RC pilot. The input and engine response data were collected. Output error identification [43] was used to arrive at a three state discretized model. When this model was excited with the input data, a reasonable correspondence with the recorded engine response was found [1]. Initially a second order model was used, but addition of a delay state greatly reduced error residuals.

[1] Except near maximum throttle, where the (linear) model consistently overestimated the peak engine rpm due to saturation evident in figure 2.15.

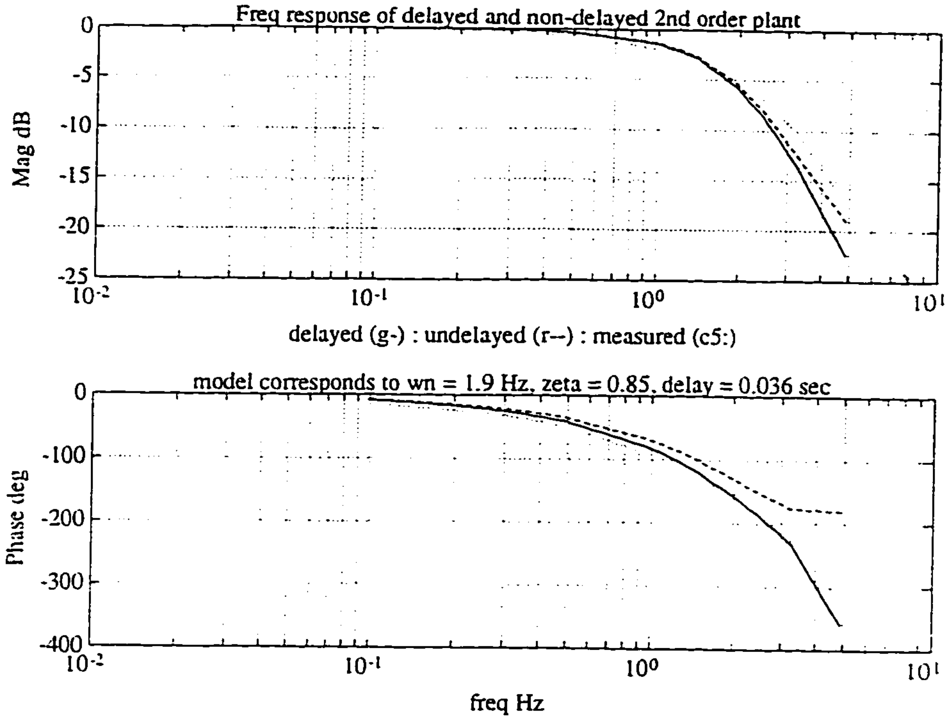


Figure 2.16: **Engine frequency response.**

Note that these experiments actually model both the throttle servo and the engine response, with input being in PWM units, and output measured in thousands of rpm (krpm). The frequency response of the 3 state (2nd order plus delay) model is plotted together with the measured frequency data in figure 2.16. A discrete engine controller was also designed and tested. This was effective in regulating the engine speed near nominal, but worked very poorly near idle when the aircraft was stationary. This was due to the tendency of the engine to alternate between 2 and 4 cycling at low power as mentioned in section 2.1.1.

The throttle is adjusted to cause idle at approximately 1500 rpm while stationary. During flight, the engine rpm at minimum throttle is observed to be approximately 2500 rpm and the oscillatory behaviour is not evident.

In flight, the engine typically runs in one of three modes. Full power on takeoff and climb, idle on descent and near nominal during altitude/airspeed regulation. For climb and descent, it makes little sense to regulate the engine rpm, so throttle is set to maximum or idle. Near nominal, one could use a fast engine control and command engine rpm, or, if the open loop engine response is fast relative to the relevant aircraft dynamics, the engine speed may be commanded open loop, using the known throttle gain. For simplicity and to avoid state inflation, the latter approach was attempted first. This was found to work adequately in practice so closed loop engine speed regulation was not used for flight control.

2.7.5 Servo modeling

RC servos are rotational devices where rotation of the actuator shaft relative to the housing is commanded by a pulse width command. Typically ±60 deg of angular motion is available. Although a math model of the servos is not supplied by the manufacturer they are known to be slew rate limited to approximately 100 deg/s and to have $\sim \pm 1$ deg deadband. Internally the DC motor drive current is proportional to the angular error up to saturation. No integral feedback is used. Faster servos are available, however the servos were sized for torque rather than speed. The surfaces are connected by pushrod linkages that are as direct and stiff as possible. The control surface deflection is approximately linear with servo rotation in the operating range. Control surface deflections were mechanically limited to approximately ±20 deg.

Single frequency at a time excitation was used to approximately characterize a servo/control surface model. The TT7 was programmed to command a sinusoidal excitation of $\pm 10°$ at frequencies varying from 0.1 Hz through 5 Hz. The data were taken for the rudder, which is the largest and presumably the slowest control surface. Rubber bands were used to provide a restoring torque, although their removal did not appear to affect the experiment. Control surface deflection was measured using a potentiometer. Plots of magnitude and phase response are depicted in figure 2.17. Assuming a second order model with delay, the frequency response data are best fit by a plant with ω_n=1.9 Hz, ζ=0.85 and delay of 0.04 s. The constant delay is attributable to the pulse width comparator circuit in the servo and accounts for $\sim$ 14

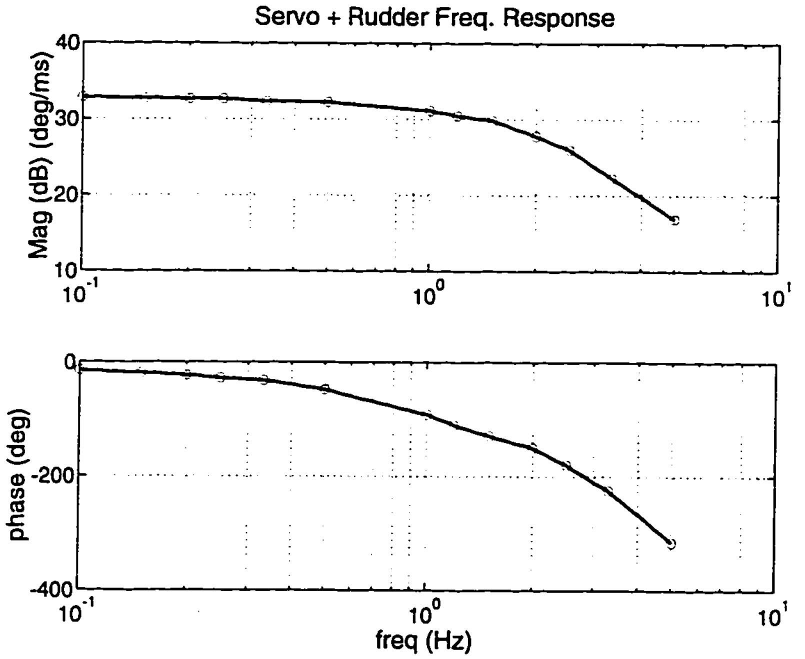


Figure 2.17: **Servo/Control surface frequency response.**

deg of delay at 1 Hz.

2.7.6 Nonlinear modeling

Since a human pilot is capable of flying the aircraft with little consideration for the engine or servo dynamics, it might be assumed that these do not interact strongly with the flight dynamics. Both pilots reported that the aircraft was easy to fly. The control laws were developed ignoring both engine and servo dynamics, so it was deemed prudent to simulate these control laws with all sensor, actuator and engine dynamics and non-nonlinearity included. Vincent Bernatets, a visiting student collaborated in developing a simulink model of the longitudinal dynamics over the summer of 1994. His report [7] details the development of this model, which also includes representative process and measurement noise effects. The conclusions lend confidence to the assumption that the ignored dynamics do not couple sufficiently to cause instability. Since LQR techniques tend to produce highly tuned compensators, model inaccuracies, or unmodeled dynamics can have deleterious effects. For this reason and as a further precaution, the initial control laws used on the airplane were relatively slow.

2.7.7 Linearized model

Linear, minimal state flight and runway models are described below. These models are discretized at 10 Hz and used for control synthesis.

Longitudinal state

Dimensionalizing equation 2.4 at the nominal flight condition yields

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.154 & 0.317 & -0.002 & -0.171 \\ -1.183 & -3.599 & 0.306 & -0.012 \\ 1.679 & -74.22 & -4.580 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.011 & 0.955 \\ 0.132 & 0.029 \\ 34.983 & -3.773 \\ 0.000 & 0.000 \end{bmatrix} \begin{bmatrix} \delta_{elev} \\ \delta_{thr} \end{bmatrix} \tag{2.10}$$

The state may be augmented with an altitude state h in meters, such that $\dot{h} \approx \left(\frac{\pi}{180} U_0\right) \theta - w$. The eigenvalues of the above matrix are

- phugoid mode: $s = -0.05 \pm j0.65$ rad/s
- short period: $s = -4.43 \pm j4.92$ rad/s

These are identifiable as the well known phugoid and short period modes. The phugoid mode is lightly damped with a period of approximately 10 seconds and the short period mode is well damped with a period of approximately 1 second. The phugoid mode requires stability augmentation.

Steady state map The steady state map gives the steady state relationship between perturbations in the controls and the state. It forms the basis for command feedforward, in that it provides the steady state controls required to achieve the operator command inputs $y_{cmd} \equiv \begin{bmatrix} u_{cmd} & \dot{h}_{cmd} \end{bmatrix}^T$. There is a *unique* combination of δ_{elev} and δ_{thr} that will achieve this. Given the state equation $\dot{x} = Fx + Gu$ and an output equation of the form $y = Hx + Ju$, one can solve for the controls required to ensure $y \equiv y_{cmd}$ at steady state. ie ($\dot{x} = 0$). In state form

$$\begin{bmatrix} F & G \\ H & J \end{bmatrix} \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} 0_{4\times2} \\ I_{2\times2} \end{bmatrix} \begin{bmatrix} u_{cmd} \\ \dot{h}_{cmd} \end{bmatrix} \tag{2.11}$$

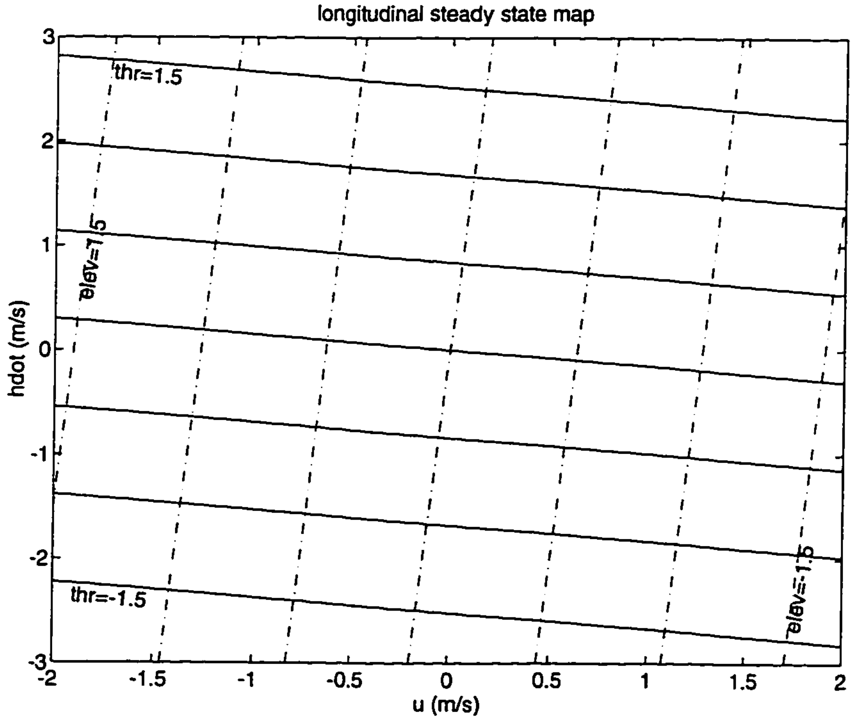


Figure 2.18: **Longitudinal steady state map.**
Solid lines are constant δ_{thr} in krpm. Dashed lines are constant δ_{elev} in degrees.

Solving gives,

$$\begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} F & G \\ H & J \end{bmatrix}^{-1} \begin{bmatrix} 0_{4\times2} \\ I_{2\times2} \end{bmatrix} \begin{bmatrix} u_{cmd} \\ \dot{h}_{cmd} \end{bmatrix} \equiv \begin{bmatrix} N_x \\ N_u \end{bmatrix} \begin{bmatrix} u_{cmd} \\ \dot{h}_{cmd} \end{bmatrix} = \begin{bmatrix} 1.000 & -0.000 \\ -0.353 & -0.004 \\ 0.000 & 0.000 \\ -1.065 & 3.004 \\ -0.787 & 0.050 \\ 0.097 & 0.538 \end{bmatrix} \begin{bmatrix} u_{cmd} \\ \dot{h}_{cmd} \end{bmatrix} \tag{2.12}$$

N_u in equation 2.12 is the longitudinal steady state map, which is also plotted in figure 2.18, showing lines of constant δ_{elev} and δ_{thr}. In this figure, throttle is in units of krpm, so variations of 1.5 around nominal approximately represent the available throttle authority. One can see that, as expected, climb rate is mostly dependent on throttle, and airspeed is mostly dependent on elevator. Holding nominal elevator and reducing throttle will cause the aircraft to slow and descend. Holding constant throttle and increasing elevator will reduce airspeed and cause the plane to climb slowly.

Lateral state

Dimensionalizing equation 2.5 at the nominal flight condition yields

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.156 & -0.326 & -0.001 & 0.171 \\ 5.618 & -1.310 & -1.048 & 0.000 \\ -49.018 & 3.834 & -13.187 & 0.000 \\ 0.000 & 0.070 & 1.000 & 0.000 \end{bmatrix} \begin{bmatrix} v \\ r \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} 0.009 & -0.045 \\ -4.363 & 10.813 \\ 79.984 & -0.038 \\ 0.000 & 0.000 \end{bmatrix} \begin{bmatrix} \delta_{ail} \\ \delta_{rud} \end{bmatrix} \tag{2.13}$$

Where the state units are [m/s deg/s deg/s deg]^T and the control deflections are in deg. The eigenvalues of the above matrix are

- roll subsidence: $s = -13.0$ rad/s
- dutch-roll mode: $s = -0.8 \pm j1.8$ rad/s
- spiral divergence: $s = -0.17$ rad/s

The dutch-roll and spiral divergence modes require stability augmentation. The state may be augmented by heading (ψ) and tracking (y_{err}) states, where $\dot{\psi} \approx r$ and $\dot{y}_{err} \approx \left(\frac{\pi}{180} U_0\right) \psi + v$. ψ is in degrees and y_{err} is in meters. The lateral steady state map gives the controls δ_{ail} and δ_{rud} required to satisfy bank angle and sideslip commands. Once again, two controls allow for a unique solution. This feedforward is used for commanding a banking turn while regulating sideslip to zero. In cases where sideslip is desired, a non-zero sideslip velocity can be commanded. For example, a forward slip approach on a crosswind landing.

Sample Rate Selection

The GPS sample rate of 10 Hz in all degrees of freedom was imposed by the I/O capabilities of the receivers employed. Although this is slow relative to sample rates commonly used for aircraft digital control [20] [21] [14] [24], this must be viewed relative to the highest frequency aircraft dynamics in need of stability augmentation. The aircraft used in this work has a short period mode at approximately 1 Hz which is naturally well damped. The fuselage flexure mode discussed in the literature was negligible with this aircraft. The dutch-roll mode has a period of approximately 3 seconds and is the highest frequency mode in need of stability augmentation (phugoid and spiral divergence are both slower). With a sample rate of 10 Hz, this gives a sampling ratio of approximately 30, which does not push the limits suggested in the literature.

2.7.8 Runway model

On the runway, the aircraft is modeled as a tricycle, see figure 2.19. Based on an assumption of zero slip, a linearized state representation for heading and crosstrack is

$$\begin{bmatrix} \dot{\psi}_{err} \\ \dot{y}_{err} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ V & 0 \end{bmatrix} \begin{bmatrix} \psi_{err} \\ y_{err} \end{bmatrix} + \begin{bmatrix} V/b \\ 0 \end{bmatrix} \delta_\theta \tag{2.14}$$

where V is forward speed, b is the wheelbase and δ_θ is the steering command in radians. A control law of the from

$$\delta_\theta = [K_\psi \quad K_y] \begin{bmatrix} \psi_{err} \\ y_{err} \end{bmatrix} \tag{2.15}$$

is sought to regulate y as V varies from zero to takeoff speed. A stabilizing controller is given by $K_\psi = 2b/V$ and $K_y = 2b/V^2$, which places the closed loop poles at $s = -1 \pm j$ rad/s. It may be shown that this feedback gives equivalent closed loop poles when the system is discretized. Steering control is inversely proportional to the speed, accounting for increased control authority as forward speed increases. Additional logic ensures that control sensitivity as $V \to 0$, is limited to that commensurate with a forward speed of 2.5 m/s and the physical actuation range of the steering

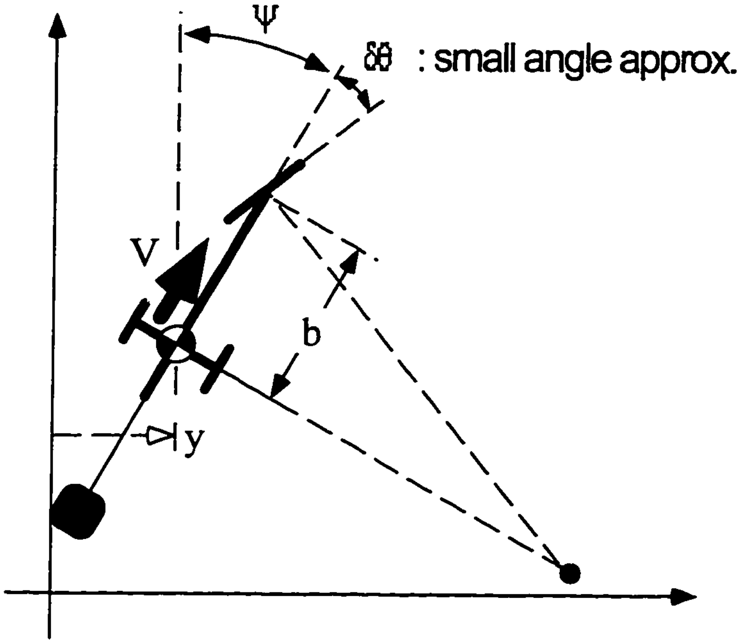


Figure 2.19: **Steering Geometry**

Runway model showing ψ_{err}, y_{err} and $\delta\theta$.

mechanism. One must also be careful to include a moment arm correction to account for the location of the tail mounted GPS antenna (position sensor). In tests on smooth concrete this control law was successful in regulating to runway centerline. On rough ground however, the control law worked less well due to violation of the no-slip assumption. Most of the aircraft weight is taken on the main landing gear, leaving the nosewheel lightly loaded. This corresponds to aircraft CG approximately 15 cm forward of the main gear. On rough ground the nosewheel tends to bounce, violating the no-slip assumption. This problem is aggravated by springs in the nosewheel assembly, but is unavoidable if the elevator is to have sufficient authority to raise the nose at takeoff. During periods when the nosewheel is off the ground the dynamics are altered considerably. During these periods the rudder becomes the active control [2]. Note that aircraft CG forward of the main wheels results in heading stability even when the nosewheel is not in continual contact with the ground. The rudder and

[2] Both nosewheel and rudder are driven with the same servo commands.

nosewheel steering servos are commanded with the same signal, so the rudder exerts some authority ($\propto V_{air}^2$) due to the propeller wash and forward airspeed.

In practice, it was found adequate to control runway heading rather than centerline tracking during takeoff roll. This was accomplished with heading error feedback inversely proportional to runway speed. This worked well even on the bumpy airfield used during flight testing.

Chapter 3

GPS Receiver Hardware

This chapter briefly covers the GPS position and attitude receivers used in this work. It discusses the implementation of the TANS receivers and the generation of carrier phase and phase rate measurements. These measurements are output serially at 10 Hz and are used in subsequent signal processing to generate position, velocity, attitude and angular velocity estimates. Further details of the TANS implementation may be found in the work of Cohen [11].

The GPS receiver hardware that was used in this work is based on the Trimble architecture. This uses an efficient one-bit signal quantization scheme that allows most of the signal processing to be performed in digital electronics. A schematic of this receiver architecture is illustrated in figure 3.1 showing a single antenna and a single channel.

GPS signal structure. As described by Spilker [53], the coarse/acquisition (C/A) signal, $s(t)$, transmitted with amplitude A from each GPS satellite is a product of three components:

1. $c(t)$, the predefined PRN code

2. $d(t)$, the 50 bits per second data message

3. $\sin(1540 f_0 2\pi t)$, the narrow-band GPS carrier

such that $s(t) = A\,c(t)\,d(t)\,\sin(1540 f_0\, 2\pi t)$. The parameter f_0 is a fundamental frequency, defined to be 1.023 MHz. The PRN sequence $c(t)$ can be either ±1 and is chosen for its orthogonality properties from a family of related "gold codes". The code changes value at 1.023 MHz (the "chipping frequency") and is repeated every millisecond. When the PRN sequence is modulated onto the GPS carrier, it has the effect of spreading the signal power over a bandwidth roughly equal to the chipping rate (1 MHz), so that each signal looks like broadband noise. The near orthogonal properties of the PRN codes allow signals from multiple satellites to continuously share the same electrical pathway and frequency range without interference. The signal from a desired satellite can be reconstructed by mixing the received signal with the code for that specific satellite. Coded signals from other satellites still resemble broad band noise and are greatly attenuated in the correlation process. The 50 bps data modulation (by ±1) that remains after decorrelation can be detected because its frequency is slower than typical correlator integration times and faster than vehicle dynamics.

3.1 Receiver Fundamentals

As shown in figure 3.1, the GPS signal received at the antenna is amplified and down-converted to an intermediate frequency of 4 MHz. At this point it is amplified again so that it saturates the output stages of the amplifier and hard-limits. The result is to quantize both the signal and the noise into one of two states: 0 or 5 volts. All subsequent sampling, mixing and correlation operations are efficiently implemented in digital electronics. After mixing with the PRN code, $c(t)$, for the desired satellite, (shown in figure 3.1 as an exclusive-or gate), the digital correlator is implemented in the form of a digital up/down counter. The binary input to the correlator determines whether the counter increments or decrements when and in-phase or quadrature (I or Q) clock pulse is applied. After approximately 4096 clock pulses accumulated over one millisecond, the value of each counter reads out the average value of the I and Q signal amplitudes. In effect, the carrier has been down-converted to baseband by sampling. The final sums are read by a microprocessor and the counters reset for the

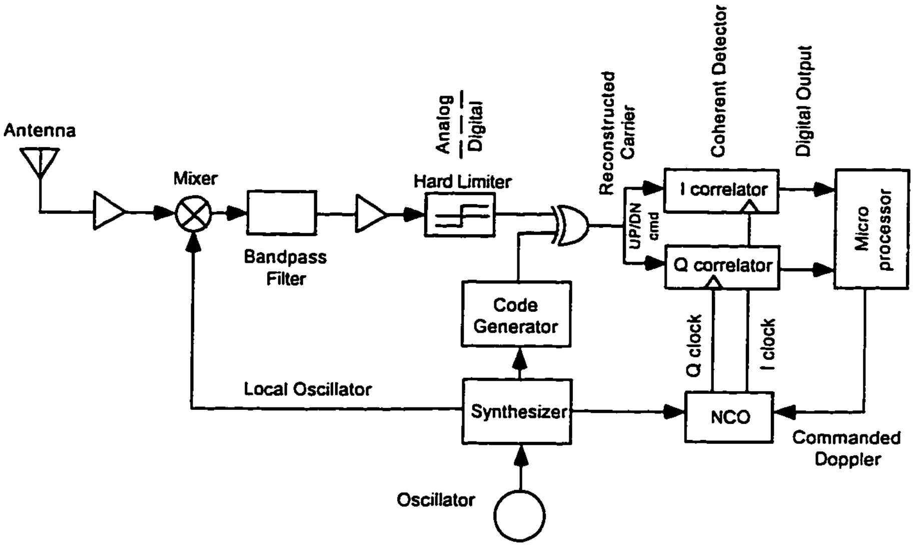


Figure 3.1: **Trimble GPS Receiver Architecture**
Single channel analog/digital schematic of hard limiting GPS receiver.

subsequent interval. Using a fast table lookup to implement the arctangent function, the I and Q measurements are interpreted as the relative phase error between the Numerically Controlled Oscillator (NCO) and the received signal.

$$\delta\phi_{IQ} = \arctan(Q/I) \tag{3.1}$$

The carrier to noise ratio (C/N_0) is proportional to $\sqrt{I^2 + Q^2}$, the length of the vector in the I-Q plane. Based on the error, $\delta\phi_{IQ}$, a new commanded frequency is calculated and latched into the NCO to drive the phase tracking error to zero. This describes the operation of one "channel", where a channel is the hardware and software required to track a single satellite. The receivers used in this work have 6 channels, allowing up to 6 satellites to be tracked simultaneously. Spilker [52], shows that because the log signal-to-noise ratio (SNR) of the GPS signal is negative, the single bit quantization is viable and the power loss incurred is only a factor of $2/\pi$, or about 2 dB.

Figure 3.2 illustrates how the I and Q averages are interpreted as phase tracking error with respect to the nominal phase, $I = C/N_0$ and $Q = 0$. In operation, the NCO frequency is continuously regulated by the control law to drive the phase tracking error

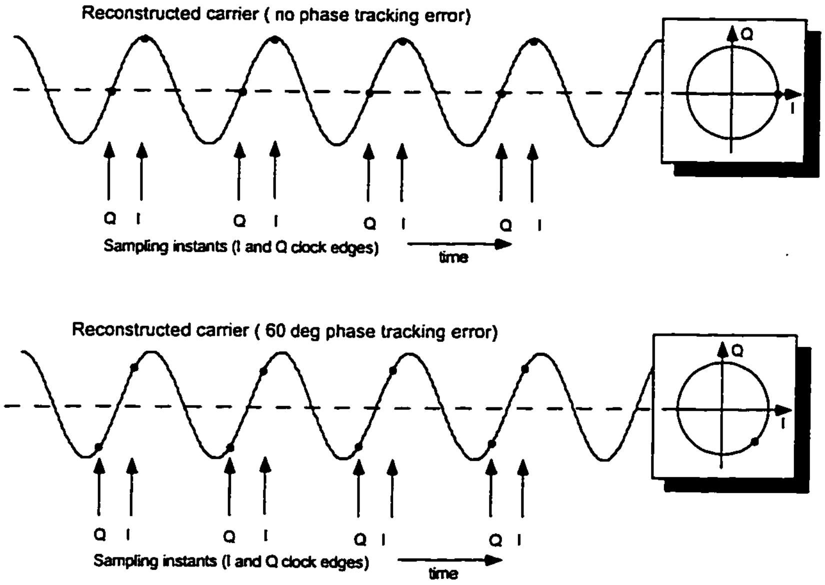

Reconstructed carrier (no phase tracking error)
Q
I
Q I Q I Q I Q I
Sampling instants (I and Q clock edges)
time
Reconstructed carrier (60 deg phase tracking error)
Q
I
Q I Q I Q I Q I
Sampling instants (I and Q clock edges)
time

Figure 3.2: **I and Q correlator sampling**
Upper diagram shows zero tracking error, lower diagram shows a 60 deg phase tracking error.

to zero. The "carrier-beat-phase" is then a measure of integrated Doppler, and the phase-rate is a measurement of the instantaneous Doppler shift of the satellite signal.

3.1.1 Absolute Carrier Phase Tracking

The TANS employs a 68000 based microprocessor running highly optimized assembly code for state estimation, and to close the carrier tracking loop. Based on a simple kinematic model, a measurement-update/time-update implementation of a steady state Kalman filter is employed. Using a sample period, T, and assuming a third order kinematic model, the state transition equation is

$$\begin{bmatrix} \phi \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix}_{k} + \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix} w_k \qquad (3.2)$$

where a discrete, unbiased disturbance, w_k, with variance W is applied to the acceleration state. The estimator measurement update equation is then

$$\begin{bmatrix} \hat{\phi} \\ \hat{\dot{\phi}} \\ \hat{\ddot{\phi}} \end{bmatrix}_k = \begin{bmatrix} \bar{\phi} \\ \bar{\dot{\phi}} \\ \bar{\ddot{\phi}} \end{bmatrix}_k + \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} \delta\phi_{IQ_k} \tag{3.3}$$

where $\delta\phi_{IQ_k}$ is the phase error determined from the I-Q table lookup, and K_1, K_2 and K_3 are steady state estimator gains. The estimator gains are chosen based on the assumed process and measurement noise statistics to provide minimum-variance estimates of the phase-state. As described by Cohen [11], the NCO is commanded according to

$$u_k = \hat{\dot{\phi}}_k + \frac{T}{2}\hat{\ddot{\phi}}_k + \frac{K_1\,\delta\phi_{IQ_k}}{T} \tag{3.4}$$

which, in effect, commands the receiver clock to the expected phase at the next time step. Note that the NCO command is therefore *not* the Doppler estimate, although it is very close in practice. The overall transfer function from $\delta\phi_{IQ_k}$ to u_k, is given by $C(z)$ as shown in figure 3.4.

The nature of the measurement noise, v_k, is well understood, being dominated by thermal noise with variance V. The process noise however, is entirely determined by the dynamics of the vehicle to which the antenna is attached. The resulting disturbances, w_k, will be correlated from one time to the next. Since there exists a direct mapping between the the ratio of V/W and the estimator poles, (as given by the symmetric root locus), the estimator gains are chosen to provide bandwidth sufficient to track the expected dynamics. A faster tracking loop will track a more agile plant at the cost of increased noise in the filtered state estimates. For this reason, the tracking loop bandwidth is one parameter that may be configured by the user.

Figure 3.3 shows the frequency response of the phase estimator transfer function from $\delta\phi_{IQ_k}$ to both the phase $\hat{\phi}_k$, (solid line) and the phase-rate $\hat{\dot{\phi}}_k$, (dashed line) based on a sample period of 6 ms. The phase estimator bandwidth based on the -3 dB point is approximately 6 Hz. The magnitude response of a perfect differentiator is shown for comparison as a dash-dotted line. A serious consequence of insufficient

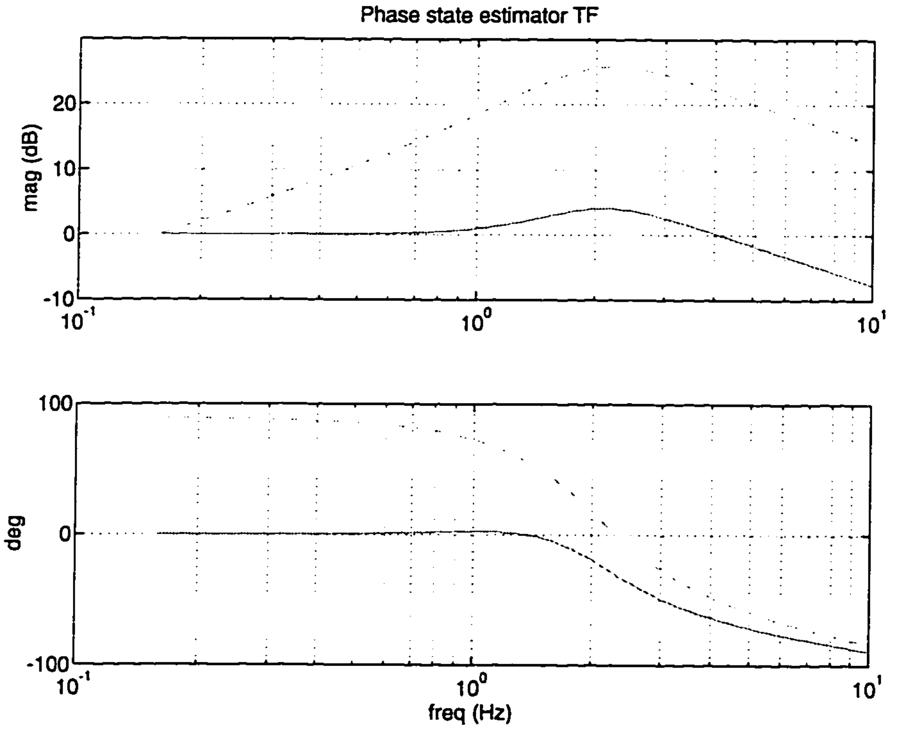


Figure 3.3: **Frequency Response of Absolute Phase Estimator.** Frequency response of estimator transfer function from $\delta\phi_{IQ_k}$ to $\hat{\phi}_k$ and $\hat{\dot{\phi}}_k$.

tracking bandwidth is an increased probability of "cycle slips", which if undetected, can severely impact the accuracy of downstream signal processing.

Another significant parameter to be chosen in the software implementation is the pre-detection interval (PDI), the number of milliseconds over which the correlator outputs are averaged prior to the phase discrimination operation. In the current work, the PDI was set to 2 ms throughout. This choice affects the base sample rate and noise in the correlation operation.

It is clear that noise in the raw phase error measurement ($\delta\phi_{IQ}$), is related to noise in the I and Q accumulations. A hardware specific unit employed is the "amplitude measurement unit" (AMU). This is defined such that $1\,AMU \equiv I/64$, where $64 \approx \sqrt{4092}$, is approximately the square root of the number of samples accumulated by the correlator over 1 ms. Since each sample has amplitude ±1, the standard deviation of the correlator output with random white noise inputs is approximately 64 if there is *no* signal present. (ie, given a pure noise signal, the output of the correlators. if plotted over many samples, would look like a circular cloud at the origin of the I-Q plane with a 1 σ radius of 64). Nominal signal strength in AMU for a satellite in the middle of the antenna pattern is in the low-twenties. This equates to 1 σ noise in $\delta\phi_{IQ}$ of arctan(1/20) $\approx$ 3 deg. For the TANS antenna/correlator hardware combination, one finds that the receiver will track satellites down to an antenna-relative elevation mask of approximately 15 deg, corresponding to a minimum signal strength of approximately 5 *AMU* [1]. For signal strength above this limit, the 1 σ noise in $\delta\phi_{IQ}$ is well approximated as inversely proportional to the signal strength in *AMU*.

The phase measurements which are available on the serial output are strobed from the low level estimator state $[\hat{\phi} \quad \hat{\dot{\phi}}]^T$, at 10 Hz. The noise in these estimates is proportional to $1/AMU$, with the proportionality factor depending on the tracking loop bandwidth. Scaled accordingly, this provides a natural metric to use in weighted least squares estimation.

Absolute carrier phase tracking as described in this section was used to generate the phase and phase-rate outputs from the stationary reference receiver and the airborne position receiver.

3.1.2 Differential Carrier Phase Tracking

GPS based attitude determination relies on the accurate measurement of *differences* in the GPS carrier phase across antennae attached to an (assumed) rigid object. The locality of all antennae enables *differential carrier phase tracking* to be performed by the receiver firmware. The technique developed by Cohen [11], uses antenna multiplexing to finesse many difficulties in achieving a practical implementation. As depicted in figure 3.4, antenna multiplexing refers to the regular and sequential cycling

[1] This is an empirical fact that was observed repeatedly with the hardware and firmware employed. Modern receivers can typically track much lower satellites, often down to better than 5 degrees above the antenna-relative horizon.

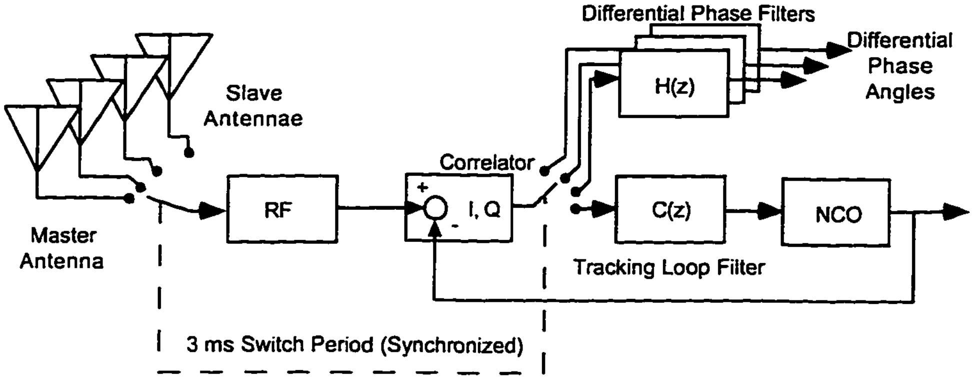

Differential Phase Filters
Differential Phase Angles
Slave Antennae
H(z)
Correlator
RF
I, Q
+
-
C(z)
NCO
Master Antenna
Tracking Loop Filter
3 ms Switch Period (Synchronized)

Figure 3.4: **Antenna Multiplexing Schematic**
One channel of a four antenna multiplexed receiver, showing filters for absolute and differential carrier phase tracking.

of signals from multiple antennae into the R.F. front end of the receiver. Antenna switching is achieved under microprocessor control using a PIN diode R.F. switch. One antenna, designated master, serves as the carrier phase reference. In this work, the master antenna for attitude was mounted on the tail of the aircraft and was also used for position and velocity determination.

When the master antenna signal passes through the R.F. path, the carrier tracking loop discussed in section 3.1.1 is used to track the absolute carrier phase. The output of the correlators at this time is the carrier phase tracking error $(\delta\phi_{IQ})$. In effect this loop removes the translational motion of the vehicle relative to each satellite. When signals from any of the other, "slave antennae", pass through the R.F. path, the NCO runs *open-loop*. Since the I and Q sampling is referenced to the same NCO, the output of the correlators at these times, $(\Delta\phi_i)$, corresponds to the differential phase angle between a slave antenna, (i), and the master antenna. As illustrated in figure 3.5, the differential phase measurements from each slave antenna are processed by a separate estimator with transfer function $H(z)$. In effect, $H(z)$ tracks differential phase changes due to variation in the vehicle orientation. The differential phase estimator is implemented as a steady state Kalman filter based on a second order kinematic model. As with $C(z)$, the estimator gains are chosen to provide bandwidth sufficient to track the expected dynamics, and the same noise versus bandwidth tradeoff

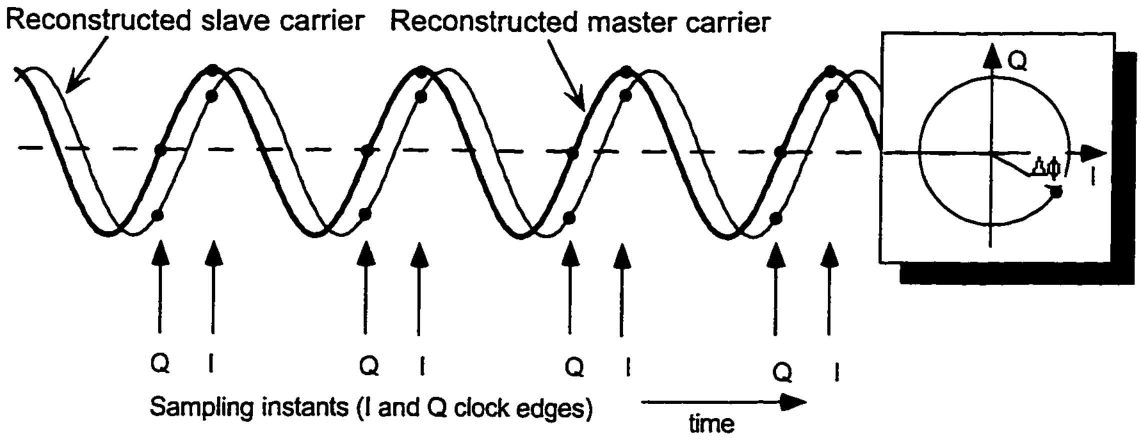

Reconstructed slave carrier
Reconstructed master carrier
Q
I
Δφ
Q I Q I Q I Q I
Sampling instants (I and Q clock edges)
time

Figure 3.5: **Coherent Phase Detector**
Generation of $\Delta\phi_i$, the raw differential phase measurement relative to the local oscillator across baseline i.

applies.

The antenna switching frequency is chosen to be as rapid as possible to track differential phase dynamics and to minimize the impact of oscillator drift during the period of open-loop operation. As described in [11], a conservative bound on the ranging error due to oscillator stability over the open-loop interval is on the order of 1 cm. It would be interesting to study the effect of vehicle dynamics on local oscillator stability and the effect this has on oscillator drift over the open-loop interval.

Figure 3.4 depicts one channel of the four antenna multiplexed receiver that was used for attitude determination. As discussed in section 3.1, there are a total of six identical channels in the TANS receiver. It is impressive to realize that each of these six channels is running four separate estimators on a heavily interrupted 68000 microprocessor. Of necessity, the signal processing is implemented very efficiently in assembly code using binary shifts and adds.

The PDI used here is again 2 ms, however an additional code epoch (1 ms) is lost with each antenna switch. The resulting dwell time on each antenna is 3 ms. With four antennae enabled, the overall cycle time is 12 ms. Clearly there is a significant power reduction associated with antenna multiplexing, which increases with the number of enabled antennae. Other problems inherent to antenna multiplexing are related to the software complexity associated with multiple potential aliasing modes, data bit transitions and initial signal acquisition. These issues are covered in detail in [11].

Phase-Rate software modifications. To access the signals required for velocity and angular velocity determination, the embedded code for both the attitude and position receivers was modified to enable output of phase-rate estimates in addition to phase estimates. In generating the phase-rate estimates, the fundamental (I and Q) measurements have been statistically differentiated at the lowest level of signal processing. Although the estimates are minimum variance based on the model and assumed noise statistics, the phase-rate estimates are necessarily more noisy than the associated phase estimates. Unfortunately, the assumption of white process noise is *not* realistic. The process noise for a plant with significant dynamics out to $1/\tau$ Hz, will be correlated with a minimum time constant of approximately τ seconds. The filter, believing the process noise to be white, (extremely jagged), weights the noisy measurements more heavily than is optimal. The obvious way to address this problem is to filter against a more realistic model at this level of signal processing. Unfortunately, this is infeasible for a number of practical reasons.

1. The receiver becomes plant specific.

2. Inflating the state increases the signal processing requirements beyond the capability of the existing microprocessor.

3. Additional information such as deterministic plant control inputs become necessary in the depths of the receiver.

Consider the filtering of differential phase (attitude) for an aircraft where accurate, decoupled, linearized models of the lateral and longitudinal dynamics are known. Assuming a four state model for each, it is clear that significant state inflation is required. Furthermore, both lateral and longitudinal dynamics will influence the phase seen across any baseline, destroying the traditional decoupling. Finally, the attitude is required to determine the line of sight vector to any satellite in the aircraft reference frame. This and the requirement for the deterministic control inputs makes the inclusion of detailed plant dynamics impractical.

An alternative approach is to model the process noise as first order markovian. with a correlation time inversely proportional to the highest frequency associated with the vehicle dynamics. The magnitude of the white noise input to the augmented state is then chosen to give estimator bandwidth matched to the fastest vehicle dynamics. This approach requires only one additional state, however MATLAB studies indicate that significant improvements in filtered performance are not realized.

A final approach is discussed in chapter 7.2, which suggests the construction of an integrated position and attitude receiver where tightly coupled accelerometers are used in a model replacement mode. The process noise is then attributable to the (hopefully well known) noise characteristics of the accelerometers.

In the current work, phase-rate estimates were made based on the simple kinematic models and white noise assumptions detailed in sections 3.1.1 and 3.1.2. Time stamped samples $\Delta\hat{\phi}_{ij}$ and $\Delta\hat{\dot{\phi}}_{ij}$, for the i'th baseline and j'th satellite were strobed from the low level (filtered) estimates at 10 Hz, together with the associated signal strength. Because these data are stored internally in fixed point, the quantization level of the phase measurements was fixed at 1/256 of a cycle, or approximately 1.4 deg. To minimize latency associated with serial output of the data, the packets were designed to minimize overhead and the maximum serial output rate was employed.

Chapter 4

GPS Signal Processing

This chapter describes the algorithms used to process the phase and phase-rate outputs described in the preceding chapter into estimates of vehicle position, velocity, attitude and angular velocity. A description of each algorithm is included in this chapter for completeness and because each was implemented for the flight system. An effort is made to differentiate those contributions which are mine from those which were developed in collaboration with others or which were inherited from others entirely. There remains much commonality between the software that was implemented by Conway [12] for his automatic helicopter and the software used in this project. Real time software for both projects was based on a TSR program written by Conway to make a DOS based computer handle interrupt driven serial communications with the Trimble GPS receivers. I was fortunate to inherit this and working position code from Conway, who in turn inherited positioning software from the GPS lab at Stanford. The original positioning software was principally authored by Lawrence [29] for the Integrity Beacon Landing System (IBLS) [3], [38], [39], [40], [41], [4], [37] and [42].

The algorithm for operational GPS attitude[1] was originally presented by Cohen [11]. Software to implement this algorithm and integrate it with a real-time system was written from scratch, principally by Conway with coding/debugging/testing assistance from myself. Cohen's method for attitude integer resolution (which must

[1] Called the "normal method" in section 4.4.1

be accomplished before operational attitude calculations may commence) proved unsuitable for reasons described in section 4.4.1. This prompted the development of a new technique for attitude integer resolution as reported in a paper co-authored by Conway and myself [34]. I attribute the core idea in this algorithm (repetitively minimizing the attitude in one dimension) to Conway. The algorithm presented in the paper is also detailed in section 4.4.2.

Conway and I also collaborated in enabling a 10 Hz GPS velocity estimate. This required changes to the receiver firmware and extension of the time alignment algorithm used for position calculation. The algorithm is described in section 4.2 and should be considered an equal contribution.

The mathematical descriptions of the position and velocity algorithms given in sections 4.1 and 4.2 differ slightly from those presented by Conway. The differences result from a different choice for time stamp alignment but result in functionally identical software[2].

The algorithm for GPS angular velocity determination (section 4.5) was developed by Uematsu and myself [44] following a suggestion by Cohen [11]. The flight software implementation was coded by myself.

Carrier Differential Fundamentals. Considering the GPS carrier, the action of the phase locked loop within the receiver can be thought of as freezing the carrier wavefronts in space. With this viewpoint, a general picture of carrier differential techniques is shown in figure 4.1. The diagram shows two antennae, both of which receive the signal from satellite j, (with unit line of sight vector $\hat{\boldsymbol{s}}_j$). By differencing their ***absolute*** phase measurements one generates the quantity $\Delta\phi_j \equiv \phi_j^a - \phi_j^g$, (not to be confused with $\Delta\phi_{ij}$, the *differential* phase from satellite j, across baseline i) which has the physical interpretation indicated in the diagram. A scalar equation in units of wavelengths, involving the cartesian displacement of the two antennae $\Delta\boldsymbol{x}$, is given by

$$\Delta\phi_j = -\hat{\boldsymbol{s}}_j \bullet \Delta\boldsymbol{x} + \tau - O_j \tag{4.1}$$

[2] The differences are in second order terms which are "in the noise" and are consequently ignored in the software implementation.

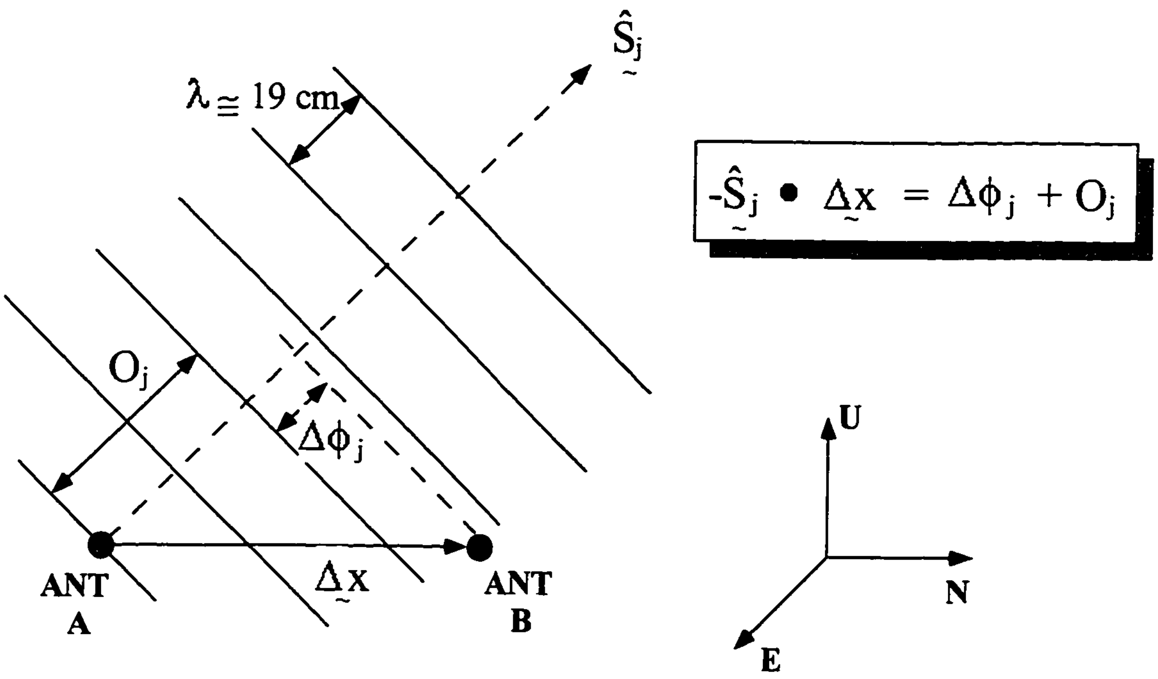


Figure 4.1: **Carrier Differential Fundamentals**
The fractional portion of the phase difference between two antennae is observable when two absolute phase measurements are differenced. An integer number of cycles O_j is not directly observable.

where O_j is an integer number of carrier wavelengths and τ, the *clock offset* expressed in units of carrier cycles, comes from clock errors at both receivers. (Note that time and distance are used interchangeably, where the proportionality factor is the vacuum speed of light, c). Reliable determination of the quantity O_j is known as *integer cycle ambiguity resolution*, and is a major difficulty associated with all carrier differential techniques. The quantity τ (assumed zero in the diagram) is not, in general, an integer number of wavelengths. Since one can reliably determine the quantity $\Delta\phi_j$ to better than 5% of a wavelength, the benefit of carrier differential techniques is the extraordinary accuracy that may be achieved.

In the description of the position and attitude algorithms that follows, the operational algorithms (which assume the integers are know) are presented first. Methods for obtaining the integers are subsequently discussed.

4.1 Position Determination

Position is calculated relative to the reference receiver in a differential manner. The absolute position accuracy of the mobile antenna is consequently limited by the absolute positioning accuracy of the reference antenna. For many purposes, however, only the relative position is important.

In addition to the carrier phase outputs detailed in chapter 3, the receivers also provide unit line-of-sight (LOS) vectors, $\hat{s}_j$, at approximately 30 second intervals and pseudo-range position information. The pseudo-range information is the absolute position and receiver clock offset, τ^r, based on C/A code measurements (conventional GPS). The LOS vectors are provided in the local East North Up (ENU) reference frame. The accuracy of the pseudo-range solution is reduced by factors such as Selective Availability (SA) and ionospheric and trophospheric effects to approximately 100 meters[3]. For receivers close together however, most of these error sources are common mode and cancel out to within a few meters. This is the basis for traditional code-differential GPS. The timestamps on the 10 Hz phase outputs are as accurate as the local clock, which may vary by up to 0.5 msec from exact GPS time. It becomes important to correct for the fact that the two receivers will therefore produce phase information corresponding to slightly different instants of exact time (although the timestamps will be identical). If not accounted for, this effect introduces errors on the order of 1 meter in the single difference phase equation (4.1).

4.1.1 Nomenclature

Following the notation used by Conway [12] with insight from Pervan [45], we define

- X^r is quantity X for receiver r.
- X_j is quantity X for satellite j.
- t_p is absolute GPS time.
- t^r is local time for a GPS receiver r.

[3] Horizontal position: 100 m 2σ

- ϕ_j^r is the measured phase for satellite j at receiver r relative to the local clock.
- $\dot{\phi}_j^r$ is the measured doppler for satellite j at receiver r relative to the local clock. $\dot{\phi}_j^r \equiv \frac{d}{dt^r}\phi_j^r$
- τ^r is the time offset for receiver r, *in cycles*, such that $t^r(t_p) = t_p + \tau^r(t_p)/L_1$
- t_p^r is that instant of t_p when receiver r made its measurement. One can therefore write $t^r(t_p^r) = t_p^r + \tau^r(t_p^r)/L_1$.
- $\boldsymbol{x}_j^r$ is the vector from satellite j to receiver r in cycles.
- $\hat{\boldsymbol{s}}_j^r$ is the unit line of sight vector from receiver r to satellite j. It points towards the satellites (up in practice).
- $\boldsymbol{\Delta x}^{rs}$ is the 3-displacement of receiver r relative to receiver s.
- $\boldsymbol{\Delta u}^{rs}$ is the 3-velocity of receiver r relative to receiver s.

4.1.2 Timestamp Alignment

The analysis deals with a receiver, r, that outputs measurements timestamped at time t^r. We consider the case where the timestamp is in error by a small amount (less than a 0.5 msec). The measurement made by receiver r was *actually* made at a time t_p^r, where the two times are related by

$$t^r(t_p^r) = t_p^r + \tau^r(t_p^r)/L_1 \tag{4.2}$$

The notation t_p^r, refers to the time that receiver, r, made its measurement, as recorded by a perfect clock. The clock offset at receiver, r, in wavelengths, at this instant is denoted $\tau^r(t_p^r)$. For example, the timestamp might record $t^r = 34.200000$ seconds when the measurement was actually made at $t_p^r = 34.200432$ seconds based on the perfect clock.

At any instant, t_p, ignoring common mode error sources and noise, the phase from satellite j, measured at receiver r, is given by

$$\phi_j^r(t_p) = -\boldsymbol{x}_j^r(t_p) \bullet \boldsymbol{s}_j^r(t_p) + \tau^r(t_p) - O_j^r \tag{4.3}$$

where O_j is some constant integer offset. This says that the carrier phase will increase as the receiver moves away from the satellite and will increase with increasing receiver time bias.

Consider two receivers, r=(a)ir and r=(g)round, each of which produces an identical timestamp, $t^a(t_p^a) = t^g(t_p^g)$, but where both timestamps are in error by unknown amounts. The measurements were ***actually*** made at different instants of precise time, t_p^a and t_p^g respectively. Differencing these measurements and using equation (4.2) we have

$$t^a(t_p^a) - t^g(t_p^g) \equiv 0 = t_p^a - t_p^g + \left(\tau^a(t_p^a) - \tau^g(t_p^g)\right)/L_1 \tag{4.4}$$

from which

$$\begin{aligned} t_p^a - t_p^g &= -\left(\tau^a(t_p^a) - \tau^g(t_p^a)\right)/L_1 - \left(\tau^g(t_p^a) - \tau^g(t_p^g)\right)/L_1 & (4.5)\\ &\approx -\left(\tau^a(t_p^a) - \tau^g(t_p^a)\right)/L_1 - \frac{d\tau^g}{dt_p}(t_p^a - t_p^g)/L_1 & (4.6) \end{aligned}$$

where the last term is on the order of 10^{-8} sec. This term is safely neglected since it contributes error on the order of only 10^{-4} wavelengths when substituted into equation (4.10) in the following analysis.

To compare the two measurements, we wish to extrapolate one measurement to the instant that the other occurred. Although other choices are equally valid, for convenience we choose to align the reference phase measurement with t_p^a, (the instant that the air receiver made its measurement). At this common instant, the two measurements can be differenced and interpreted in terms of physical antenna separation. Expanding ϕ_j^g in a Taylor series about t_p^g gives

$$\phi_j^g(t_p^a) = \phi_j^g(t_p^g) + \frac{d}{dt_p}\left(\phi_j^g(t_p^g)\right)(t_p^a - t_p^g) + \frac{1}{2}\frac{d^2}{dt_p^2}\left(\phi_j^g(t_p^g)\right)(t_p^a - t_p^g)^2 + h.o.t. \tag{4.7}$$

the second and higher order terms contribute less than 10^{-5} wavelengths and can be neglected.

Applying the chain rule to equation (4.2), gives the operator

$$\frac{d}{dt_p}(\bullet) = \left[1 + \frac{1}{L_1}\frac{d\tau^r}{dt_p}\right]\frac{d}{dt^r}(\bullet) \tag{4.8}$$

Expanding equation (4.7) then gives

$$\begin{aligned} \phi_j^g(t_p^a) &\approx \phi_j^g(t_p^g) + \left[1 + \frac{1}{L_1}\frac{d\tau^g(t_p^g)}{dt_p}\right] \dot{\phi}_j^g(t_p^g)\,(t_p^a - t_p^g) && (4.9) \\ &\approx \phi_j^g(t_p^g) + \dot{\phi}_j^g(t_p^g)\,(t_p^a - t_p^g) && (4.10) \end{aligned}$$

Once again the neglected term contributes only a sub-millimeter correction. Substituting for $t_p^a - t_p^g$ from equation (4.6) gives

$$\phi_j^g(t_p^g) \approx \phi_j^g(t_p^a) + \frac{\dot{\phi}_j^g(t_p^g)}{L_1}\left(\tau^a(t_p^a) - \tau^g(t_p^a)\right) \qquad (4.11)$$

Differencing the air and reference measurements at their respective timestamps, and using equation (4.3) then gives

$$\begin{aligned} \Delta\phi_j(t_p^a) &\equiv \phi_j^a(t_p^a) - \phi_j^g(t_p^g) && (4.12) \\ &= -\boldsymbol{x}_j^a(t_p^a) \bullet \hat{\boldsymbol{s}}_j^a(t_p^a) + \left(1 - \frac{\dot{\phi}_j^g(t_p^g)}{L_1}\right)\left(\tau^a(t_p^a) - \tau^g(t_p^a)\right) - O_j \\ &\quad + \boldsymbol{x}_j^g(t_p^a) \bullet \hat{\boldsymbol{s}}_j^g(t_p^a) && (4.13) \\ &= -\left(\boldsymbol{x}_j^a(t_p^a) - \boldsymbol{x}_j^g(t_p^a)\right) \bullet \hat{\boldsymbol{s}}_j^a(t_p^a) + \left(1 - \frac{\dot{\phi}_j^g(t_p^g)}{L_1}\right)\tau(t_p^a) - O_j \\ &\quad + \boldsymbol{x}_j^g(t_p^a) \bullet \left(\hat{\boldsymbol{s}}_j^g(t_p^a) - \hat{\boldsymbol{s}}_j^a(t_p^a)\right) && (4.14) \\ &= -\boldsymbol{\Delta x}^{ag} \bullet \hat{\boldsymbol{s}}_j^a + \left(1 - \dot{\phi}_j^g/L_1\right)\tau - O_j + \boldsymbol{x}_j^g \bullet \Delta\hat{\boldsymbol{s}}_j^{ga} && (4.15) \end{aligned}$$

Note that the exact value of O_j^r is meaningless; only the difference $O_j = O_j^a - O_j^g$ has significance.

The last term is *not* neglected because it can lead to a significant error term if the baseline between the mobile and reference station is large. Back of the envelope calculations indicate that a 1 cycle error will result from this effect with a baseline of approximately 3 km. The last term may be used to give an upper bound on the error.

$$\begin{aligned} \boldsymbol{x}_j^g \bullet \Delta\hat{\boldsymbol{s}}_j^{ga} &\approx \rho_{min}\left(1 - \cos(\Delta x/\rho_{min})\right) && (4.16) \\ &\approx \Delta x^2/2\rho_{min} && (4.17) \end{aligned}$$

where $\rho_{min} \approx 20{,}000$ km, is the range to a satellite at closest approach, and Δx is the baseline length. The change of line of sight vector is related to the fact that, although far away, the satellites project spherical, not planar wavefronts. This correction therefore accounts for the difference between a planar assumption and a spherical assumption on incident wavefronts. Equation (4.16) is basis independent, however if including the correction, one must also account for the fact that the line of sight vectors are output from each receiver in the local ENU basis, ie. different bases as the receivers are separated.

Ignoring the last term, the single difference equation accounting for small timing errors is given by

$$\Delta\phi_j = -\hat{\boldsymbol{s}}_j^a \bullet \boldsymbol{\Delta x}^{ag} + \left(1 - \dot{\phi}_j^g / L_1\right) \tau - O_j \qquad (4.18)$$

Stacking up n measurements gives a matrix equation that can be solved in a least-squared sense

$$\begin{bmatrix} -\hat{\boldsymbol{s}}_1^T & 1 - \dot{\phi}_1^g(t_p^g)/L_1 \\ -\hat{\boldsymbol{s}}_2^T & 1 - \dot{\phi}_2^g(t_p^g)/L_1 \\ \vdots & \vdots \\ -\hat{\boldsymbol{s}}_n^T & 1 - \dot{\phi}_n^g(t_p^g)/L_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta x}^{ag} \\ \tau \end{bmatrix} = \begin{bmatrix} \Delta\phi_1(t_p^a) \\ \Delta\phi_2(t_p^a) \\ \vdots \\ \Delta\phi_n(t_p^a) \end{bmatrix} + \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_n \end{bmatrix} \qquad (4.19)$$

The values on the right hand side of equation (4.19) are first differences of the receiver outputs after adjustment by the integers, which are assumed known at this point. The matrix on the left hand side contains the satellite line-of-sight unit vectors, which are extrapolations of the line-of-sight vectors produced by the TANS. The phase-rate measurements are uplinked from the reference station or may be estimated from the uplinked phase as described in section 4.1.3. The equation is expressed in wavelengths and is time referenced to the instant that the air receiver made its phase measurement.

4.1.3 Kinematic Positioning

For simplicity of notation, equation (4.19) is rewritten for timestep k as

$$H_k x_k = \Phi_k + O + \nu_k \qquad (4.20)$$

where ν, an $n \times 1$ vector of disturbances has been added. The disturbances are attributable to phase noise and multipath. Other error sources (ionospheric, trophospheric, SA) are negligible for differential baselines of only a few kilometers. For the aircraft used here, multipath was not expected to contribute significant errors except perhaps when steeply banked over a reflective surface (such as tranquil seawater). As described in chapter 3, the receiver phase noise over all channels is well modeled as white, unbiased and independent, with RMS value in wavelengths given by $\sigma_{\Delta\phi_j} = \sqrt{\sigma^2_{air_j} + \sigma^2_{ref_j}}$. And we assume, $E\{\nu\,\nu^T\} = V$, where $V = diag(\sigma^2_{\Delta\phi_j})$.

The fourth column of the observation matrix, H, is a vector of unity elements, perturbed by terms on the order of 10^{-6}. For this reason, the vector of integers, O, is determined only to within an integer constant. In effect, one can add any (reasonably small) integer number of wavelengths to τ if one also adds this number of wavelengths to each element of O.

Clearly, at least four first-difference measurements are necessary to solve for the four unknowns in $\mathbf{\Delta x}$ and τ. This in turn requires that at least four *common* satellites are simultaneously in track on both the airborne and reference receivers. Since the two stations are separated, it further requires that a wireless datalink is available. For real time applications, latency is also an important consideration.

The weighted least-squares solution vector $\hat{x}$, (in carrier wavelengths) is given by

$$\hat{x}_k = (H_k^T V_k^{-1} H_k)^{-1} H_k^T V_k^{-1} (\Phi_k + O) \tag{4.21}$$

where $\mathbf{\Delta x}$ is the vector from the reference station to aircraft in the ENU basis, and τ is the relative clock offset between the two receivers.

The error covariance is given by

$$E\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\} = (H_k^T V_k^{-1} H_k)^{-1} \tag{4.22}$$

from which measures of *dilution of precision*, DOP, are often derived. The quality of the solution is therefore related to the number of satellites, their signal strength and the location of each satellite relative to the user (satellite geometry).

If more than four measurements are available, the solution residual provides an indication of self-consistency among the measurements. A residual norm within the expected noise bounds is a necessary (but insufficient) condition for a correct position fix.

The requirement to hold at least four common satellites was found to be a limiting factor for position robustness with the model aircraft. After experimenting with a number of satellite selection schemes, it was found that the most reliable results were obtained when tracking the highest elevation satellites. The problem of maintaining position integrity was compounded by a number of factors.

1. Kinematic position integrity had to be maintained from takeoff to landing since there was no capability for reacquisition of accurate position just prior to touchdown.

2. It is necessary to bank the aircraft to fly the desired traffic pattern. In banking, low elevation satellites become occluded. Consequently, in flying the chosen pattern, *every* satellite lower than approximately 30 deg of elevation is occluded multiple times.

3. The receiver/antenna combination employed in both the airborne and reference locations is unable to track satellites below approximately 15 deg of elevation relative to the antenna horizon.

4. No additional ranging sources such as pseudolites were employed.

5. The reference receiver had only 6 channels, which is not enough to track all the (low elevation) satellites that are seen by the airborne receiver.

In view of these issues, we chose to fly at times when there were 5 or more satellites above a 25 deg elevation mask. At these times, it was possible to keep at least 4 and usually 5 common satellites in view at all times. The process of bringing a newly acquired signal online is called *patching*, and is required many times during each flight to handle banking induced occlusion events.

Reference Phase Prediction

For a control system, the punctual availability of sensor data is critical. For an onboard sensor this does not pose a major problem, however, since the reference station phase measurements are also required for position calculation, this places similar requirements on the wireless communications channel. The aircraft suffered one crash during testing, and it is likely that a brief transmission outage was responsible for this incident. The software architecture was subsequently altered to remove such a stringent dependence on the data link integrity. In the former system, reference phase and phase-rate data were transmitted to the aircraft at 10 Hz. Position and velocity calculations were not attempted until the validated reference phase data was received. This design was susceptible to dropouts or errors in data transmission which would prevent the position estimate from being updated. This was a serious problem, since occasional transmission dropouts proved difficult to avoid in practice.

Unlike the aircraft antenna, the reference station antenna is fixed, and consequently its phase is relatively predictable for a brief duration. Work by [31] in reference phase prediction suggested that propagation of the reference phase by a simple quadratic extrapolation is adequate for up to 10 seconds. With this approach, the last few seconds of received reference data is stored in a ring buffer in the aircraft computer. Fitting a quadratic to this data allows timestamps to be matched with timestamps from the local receiver as soon as local data becomes available. This removes the latency associated with data transmission and is robust to brief transmission outages. If a transmission outage occurs, the position accuracy degrades gracefully over time rather than suddenly stopping entirely, and full accuracy is recovered as soon as the next reference packet arrives. The minimum update rate is driven by error accumulation in the predicted reference phase, which is largely a function of satellite clock error (SA).

This architecture also relaxes the latency requirements on the communications channel and allows its capacity to be reduced if one chooses to transmit reference data less frequently. This does not compromise the ability of the mobile station to calculate fixes at high rate, although transmission at less than 2 Hz will degrade the positioning accuracy. A further factor is related to the detection and handling of phase wraparound (due to finite word length internal to the receiver) and leap milliseconds. These events and satellite constellation switches become more troublesome as the reference station update frequency decreases. A practical lower limit is probably 2

Hz. Assuming update at 5 Hz, a 6 channel reference receiver transmitting phase and phase-rate will require approximately 2400 bits per second in datalink capacity.

In predicting the reference phase with a quadratic extrapolation, one is effectively finding three coefficients. Assuming reference update at 5 Hz, the number of points for data fitting and the separation of these points must be selected. At least three measurements are required, and noise in the solution will be reduced by including additional data. Plotting coefficient noise versus number of data points shows that the incremental benefit of including more than approximately 7 data points is minimal. To save computation 6 data points is a reasonable compromise. At 5 Hz update, this involves fitting the phase data over the previous 1.2 seconds, which is also reasonable given the spectral content of the reference phase data. In effect, this filters the reference phase data with a time constant of approximately 1 second, although the filter is not expressible as a discrete transfer function. One way to proceed is to form the normal equations each time a new phase datum arrives and solve the resulting toeplitz matrix for each channel to find the desired coefficients. The coefficients are then used to propagate the state forward in time. The phase-rate estimate is given by the second (linear) coefficient and may be used for the time alignment correction if the phase rate is not available. An alternative to forming the normal equations is to use a recursive filter, however the benefit of this approach is not large for such a small system.

Latency

The latency discussed above is due to the transmission of reference data across the data link. There are three other sources of position latency.

1. Estimator phase lag internal to the receiver. Tracking a 2 Hz signal, with an assumed estimator bandwidth of 6 Hz, the phase lag, (as illustrated in figure 3.3) is approximately 20 deg, or equivalently 28 msec.

2. Serial output latency. At 38400 bps, it takes approximately 12 msec to receive a 46 byte data packet from the receiver.

3. Processing latency. The processing time with the current computer is less than 1 msec.

Together these account for position latency of up to 40 msec.

Cycle Slips

A cycle slip results when the phase locked loop inside the receiver fails to track a one wavelength increase in the phase on a particular channel. This can happen if low signal strength (or strong multipath) is momentarily encountered, but is *not* a common occurrence at typical signal strengths (> 5 AMU). Imminent cycle slips are usually detected at the signal processing level of the receiver (when low signal strength is encountered) and are flagged when the *integer valid* bit is reset.

An undetected cycle slip is identical to acquiring an integer error, and will result in increased residual and a biased position estimate. Although uncommon, cycle slips become more likely with

- Antenna multiplexing
- Low signal strength
- Faster dynamics

4.1.4 Integer Initialization

Thus far it has been assumed that the integer cycle ambiguities are known. There are a number of different approaches to acquiring the integers, which can be classified as

- search methods
- geometry based methods
- straightforward declaration

Search methods attempt to find the correct cycle ambiguities using a redundant (overdetermined) set of measurements. Trying different integers combinations, an algorithm attempts to find the set of integers that minimizes some cost function. This has the disadvantages of being slow and often inconclusive, although for certain constrained applications, (such as attitude integer determination for short baselines) it is very effective.

Geometric methods rely on geometry change over time between the baseline (the vector between a pair of antennae) and the satellite line of sight vectors. One approach is to wait for the satellites to move overhead. This has long been used by surveyors but it is not ideal for automatic control as it involves a lengthy delay. A second approach is to achieve large line of sight changes quickly, by moving the mobile antenna relative to a pair of ground based pseudo-satellites (*pseudolites*). This works well for certain applications and has the advantage of supplying additional ranging sources when the satellite geometry is poor. A drawback is the additional hardware, setup and surveying that is required.

In this work the integers were initialized by straightforward declaration. This was achieved by placing the aircraft in a known initial position. The vector, $\boldsymbol{\Delta x}$, between the reference antenna and the aircraft position antenna was surveyed and recorded for initialization. Two initialization points were surveyed, one at either end of the runway. Pseudo-range (code) estimates of airborne and reference receiver time bias, τ^r, were differenced to provide an estimate of τ. Given $\boldsymbol{\Delta x}$ and τ, equation (4.20) was used to solve for the vector of "floating integers" O_{flt}. The elements of O_{flt} were then rounded to integer elements of O. Any fractional component common to all elements of O_{flt} was subsequently absorbed into τ. Due to inaccuracies in the pseudo-range estimate of τ, this initialization technique may result in slightly incorrect integer acquisition. This potential is mitigated by the following factors:

- The major error sources in code positioning are spatially correlated to a high degree and the receivers were approximately 10 meters apart during initialization.

- Identical receivers tracking the same satellites were used.

- Initialization took place in a multipath benign area.

Anticipating differential code accuracy on the order of a meter, one can argue that the acquired integers are accurate to within a small number of wavelengths. The resulting least-squares residual will be small by design.

Consequence of initialization errors.

Consider the case where the cycle ambiguities are initialized in error using the vector $x_0 = x + x_{err_0}$, where x is the correct (but unknown) initialization vector as described by equation (4.20). We first consider the case where the initialization error is distributed among all elements of x_{err_0}. After initialization, the resulting vector of "floating integers" is given, ignoring noise, by

$$O_0 = H_0\,x_0 - \Phi_0 \tag{4.23}$$

Breaking O_0 into true and error components, $O_0 = O + O_{err}$, it follows that the vector of "floating integer" errors is given by

$$O_{err} = H_0\,x_{err_0} \tag{4.24}$$

Assuming an overdetermined set of equations, ($n \geq 4$), and equal weighting ($V = \sigma^2 I_n$), the position will subsequently be found as

$$\begin{aligned} \hat{x}_k &= (H_k^T H_k)^{-1} H_k^T(\Phi_k + O_0) \\ &= (H_k^T H_k)^{-1} H_k^T(\Phi_k + O) + (H_k^T H_k)^{-1} H_k^T O_{err} \\ &= x_k + (H_k^T H_k)^{-1} H_k^T H_0\,x_{err_0} \end{aligned} \tag{4.25}$$

Writing H_k as

$$H_k = H_0 + \begin{bmatrix} \Delta s_1^T & \epsilon_1 \\ \vdots & \vdots \\ \Delta s_n^T & \epsilon_n \end{bmatrix} \tag{4.26}$$

which is valid even over constellation switches and where ϵ_j is on the order of 10^{-6}. Consequently,

$$x_{err_k} \equiv \hat{x}_k - x_k \;=\; (H_k^T H_k)^{-1} H_k^T H_0\,x_{err_0}$$

$$
\begin{aligned}
&= (H_k^T H_k)^{-1} H_k^T H_k \, x_{err_0} - (H_k^T H_k)^{-1} H_k^T \begin{bmatrix} \Delta s_1^T & \epsilon_1 \\ \vdots & \vdots \\ \Delta s_n^T & \epsilon_n \end{bmatrix} x_{err_0} \\
&= x_{err_0} - (H_k^T H_k)^{-1} H_k^T \begin{bmatrix} \Delta s_1^T & \epsilon_1 \\ \vdots & \vdots \\ \Delta s_n^T & \epsilon_n \end{bmatrix} x_{err_0} \qquad (4.27)
\end{aligned}
$$

The error varies from x_{err_0} with changing satellite geometry, but will remain reasonably constant over a period of minutes provided constellation switches are specifically excluded. This means that even over a number of minutes, the errors resulting from incorrect initialization are not necessarily catastrophic, but grow with satellite motion proportional to the initial error. For brief station-keeping tasks (on the order of a few minutes), where the objective is to "stay there until I tell you to move", approximate integer initialization from code differential data is potentially adequate.

Conway [12], notes that significant errors accumulate only after sufficient satellite motion has occurred, and that satellite motion is precisely what is required for integer resolution in the static case. Assuming more than 4 satellite measurements are available, he proposes an algorithm to successively refine the correct integers from an approximate initialization. This provides an initial capability, with greater accuracy being attained over time. For example, after approximate integer initialization on the runway just prior to takeoff, the measurements are good enough over the time required for takeoff, that this could be done under automatic control. Further integer refinement could be done over time while airborne.

Considering the case where the initialization error is limited to the τ element of x_0, with τ in error by τ_{err_0}, one has:

$$
x_{err_k} = \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - (H_k^T H_k)^{-1} H_k^T \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \right) \tau_{err_0} \qquad (4.28)
$$

with ϵ_j on the order of 10^{-6}, errors in τ_0 (based on differential code phase measurements) do not significantly corrupt the position state over time.

Patching

Patching is the procedure used to bring new measurements online as they become available. The algorithm used was very simple. Assuming a known position, $\mathbf{\Delta x}_k$ and receiver clock offset, τ_k, the integer for any new measurement, j, was resolved using equation (4.18) to get a floating estimate. This estimate was then rounded to the nearest integral value. This technique has the potential to introduce errors since rounding to an incorrect integer is possible. This can occur if the position covariance (DOP) in the direction of the new satellite is large. In this case, it is possible for the instantaneous position estimate to be in error by greater than half a wavelength in the direction of the new satellite. Assuming phase noise standard deviation of 0.05 cycles on all measurements, a DOP of 10 in the direction to the new satellite will result in a 32% probability of incorrect integer aquisition. This can be reduced if a number of samples are averaged prior to the acquisition execution provided the solution is unbiased. Of course the previous absence of the measurement being acquired is likely to cause the position covariance to be large in this direction.

If an incorrect integer patch does occur, the residual norm will increase and the position estimate will become biased. With satellite motion, the error will potentially increase over time as described by equation (4.27). Unfortunately, with continuing patching, such errors will tend to increase the probability of subsequent patching errors. We chose to fly at times where there were many satellites high in the sky, this choice was made to minimize the problems with satellite occlusion and subsequent patching.

As described earlier, the problem of losing low elevation satellites was compounded by the poor low elevation gain on the (survey) antennae that were used. This situation can be improved by using antenna with better low elevation gain and by providing additional ranging sources such as pseudolites. Pervan [45], suggests a method in which integers are not rounded to integral values, but allowed to "float" with an associated covariance matrix. The covariance associated with the integers then enters into the position and position covariance calculations.

End Around Checks

No attempt was made to compare the accuracy of the GPS position measurement with an alternate position reference. As documented in [37], careful comparisons of CDGPS against laser tracker measurements indicate that it is more appropriate to calibrate the laser tracker using GPS than vice versa. For this reason "end-around" checks were used to gain confidence in the position integrity over the duration of a typical flight. This was achieved by bringing the aircraft back to the surveyed initialization point after a flight, and comparing the sensor output with the known

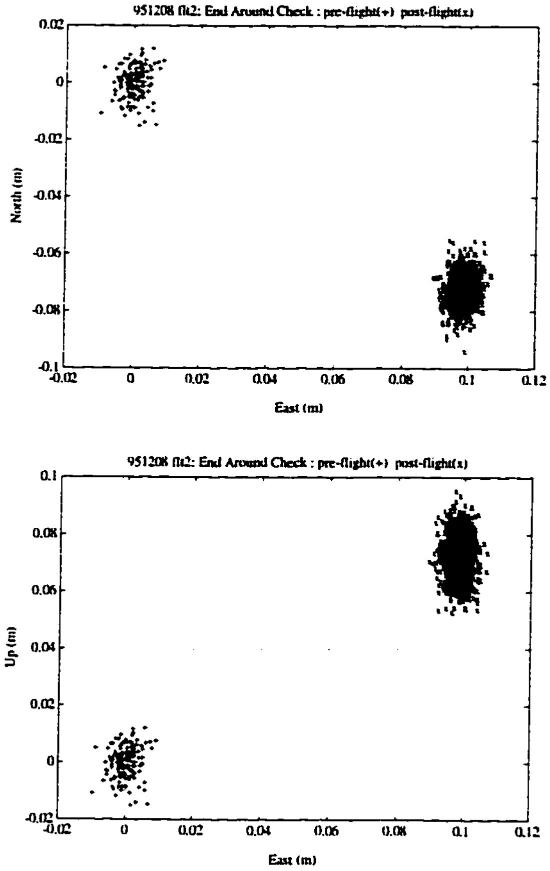


Figure 4.2: **Position "End-Around" Check**

Horizontal and vertical position error accumulated over a 7 minute flight in *marginal* satellite conditions. The "+" symbols represent data points taken pre-flight. The "x" symbols represent data points taken post-flight with the aircraft positioned identically in both cases.

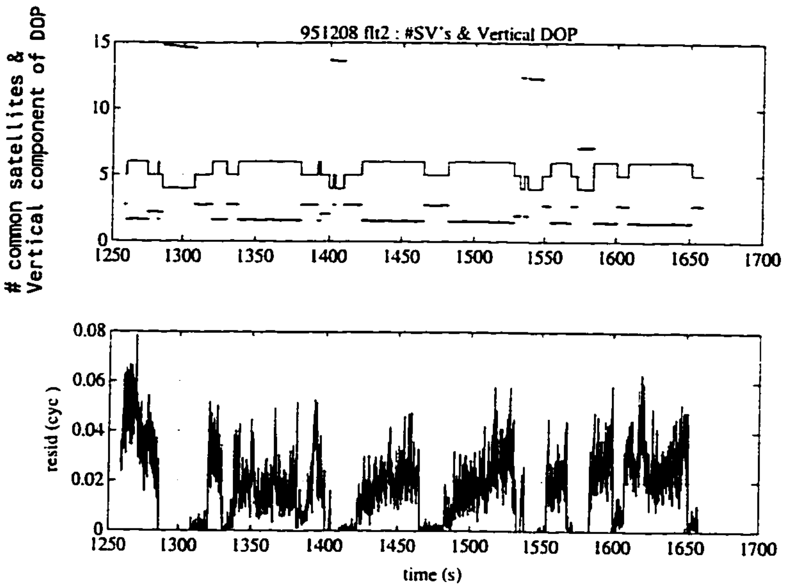


Figure 4.3: **Vertical component of DOP and Position Residual**
The upper plot shows a time history of the number of satellites used in the position calculation and the vertical component of DOP (divided by 2 to improve legibility). The lower plot shows the corresponding position residual in units of carrier wavelengths.

position. Reasonable models of error propagation indicate that the position error at any point during the flight is likely to be less than that accumulated over the entire flight. Results of an end-around check are plotted in figure 4.2. This figure shows the position before and after a 7 minute flight on a day where the satellite geometry was not favorable. The multiple points plotted represent the differing measurements made during pre- and post-flight periods when the aircraft was stationary. Figure 4.3 shows the number of common satellites in track, the vertical dilution of precision (DOP) and the position residual over the flight. It may be seen that approximately 14 cm of position error have accumulated over the duration of the flight. It seems likely that one of the patching events that occurred numerous times over the flight caused an incorrect integer to be acquired. This *imperfect* result is presented because it represents a limiting case on the conditions we chose to fly under. The accumulation of 14 cm of position error, although undesirable, is tolerable, being approximately the diameter of the main landing gear wheels.

4.2 Velocity Determination

Absolute phase rate (Doppler) estimates are output from the TANS ten times per second, quantized in units of 0.03 L1 carrier cycles per second, or approximately 6 mm per second. Unfortunately, as discussed in chapter 3, the level of noise in these estimates is only a little better than that which would be achieved by differencing the phase measurements directly. The resulting velocity estimate has RMS noise on the order of 0.1 m/s in ENU coordinates. This is similar to that which would be achieved by taking a first difference of the calculated (ENU) position estimates, but has the advantage of avoiding the sample delay associated with this approach. A more detailed analysis of the noise in the phase-rate estimates as a function of bandwidth is included in section 4.5.2, which discusses the analogous use of differential phase-rate for GPS angular velocity determination.

Taking the time derivative of the position equation involves the velocity vector $\boldsymbol{\Delta u}^{ag} = \frac{d}{dt_p}\boldsymbol{\Delta x}^{ag}$. The signal produced by the receiver, $\dot{\phi}_j^r$, is the derivative of the measured phase of satellite j with respect to the *local* clock in receiver r. Again, as with position, the difference between local and exact time complicates the algebra somewhat. Differentiating equation (4.18), and applying the chain rule as earlier, one has

$$\begin{aligned}\left(1+\frac{\dot{\tau}^a(t_p^a)}{L_1}\right)\dot{\phi}_j^a(t_p^a)-\left(1+\frac{\dot{\tau}^g(t_p^g)}{L_1}\right)\dot{\phi}_j^g(t_p^g) &= -\hat{\boldsymbol{s}}_j^a \bullet \boldsymbol{\Delta u}^{ag} - \dot{\hat{\boldsymbol{s}}}_j^a \bullet \boldsymbol{\Delta x}^{ag} \\ &\quad -\left(1+\frac{\dot{\tau}^g(t_p^g)}{L_1}\right)\frac{\ddot{\phi}_j^g(t_p^g)}{L_1}\tau(t_p^a) \\ &\quad +\left(1-\frac{\dot{\phi}_j^g(t_p^g)}{L_1}\right)\dot{\tau}(t_p^a) \qquad (4.29)\end{aligned}$$

where $\ddot{\phi}_j^g \equiv \frac{d^2\phi_j^g}{dt^{g2}}$. Rearranging the left hand side, gives

$$\begin{aligned} LHS &= \dot{\phi}_j^a(t_p^a)-\dot{\phi}_j^g(t_p^g)+\left(\dot{\tau}^a(t_p^a)-\dot{\tau}^g(t_p^g)\right)\frac{\dot{\phi}_j^g(t_p^g)}{L_1}+\frac{\dot{\tau}^a(t_p^a)}{L_1}\left(\dot{\phi}_j^a(t_p^a)-\dot{\phi}_j^g(t_p^g)\right) \\ &\approx \left(1+\dot{\tau}^a(t_p^a)/L_1\right)\left(\dot{\phi}_j^a(t_p^a)-\dot{\phi}_j^g(t_p^g)\right)+\dot{\tau}(t_p^a)\dot{\phi}_j^g(t_p^g)/L_1 \qquad (4.30)\end{aligned}$$

Ignoring the $\dot{\tau}^g\,\ddot{\phi}_j^g\tau/L_1^2$ term in equation (4.29), and combining with equation (4.30), gives the single difference velocity equation after correction for small timing errors:

$$\begin{aligned} -\hat{\boldsymbol{s}}_j^a \bullet \boldsymbol{\Delta u}^{ag} + \left(1 - 2\dot{\phi}_j^g(t_p^g)/L_1\right)\dot{\tau}(t_p^a) \;\approx\; & \left(1 + \dot{\tau}^a(t_p^a)/L_1\right)\left(\dot{\phi}_j^a(t_p^a) - \dot{\phi}_j^g(t_p^g)\right) \\ & +\dot{\hat{\boldsymbol{s}}}_j^a \bullet \boldsymbol{\Delta x}^{ag} + \frac{\ddot{\phi}_j^g(t_p^g)}{L_1}\tau(t_p^a) \end{aligned} \tag{4.31}$$

Defining $\Delta\dot{\phi}_j \equiv \dot{\phi}_j^a(t_p^a) - \dot{\phi}_j^g(t_p^g)$ and stacking up n equations gives a matrix equation that can be solved in a least squares sense:

$$\begin{bmatrix} -\hat{\boldsymbol{s}}_1^T & 1 - 2\dot{\phi}_1^g/L_1 \\ -\hat{\boldsymbol{s}}_2^T & 1 - 2\dot{\phi}_2^g/L_1 \\ \vdots & \vdots \\ -\hat{\boldsymbol{s}}_n^T & 1 - 2\dot{\phi}_n^g/L_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta u}^{ag} \\ \dot{\tau} \end{bmatrix} = (1 + \dot{\tau}^a/L_1) \begin{bmatrix} \Delta\dot{\phi}_1 \\ \Delta\dot{\phi}_2 \\ \vdots \\ \Delta\dot{\phi}_n \end{bmatrix} + \begin{bmatrix} -\dot{\hat{\boldsymbol{s}}}_1^T & \ddot{\phi}_1^g/L_1 \\ -\dot{\hat{\boldsymbol{s}}}_2^T & \ddot{\phi}_2^g/L_1 \\ \vdots & \vdots \\ -\dot{\hat{\boldsymbol{s}}}_n^T & \ddot{\phi}_n^g/L_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta x} \\ \tau \end{bmatrix} \tag{4.32}$$

The rate of change in line of sight comes from interpolation. The second derivative of phase comes from first differences of $\dot{\phi}_j^g$. The position and time bias come from the position calculation, which is performed immediately prior to the velocity calculation. With the TANS receivers used on the airplane, it is reasonable to ignore the factor of $(1 + \dot{\tau}^a/L_1)$. The net result is an error term that is small relative to the noise for anything subsonic. It is also reasonable to ignore the factor of 2 which appears in the leftmost matrix in equation (4.32), as the term it multiplies also has a very small effect relative to the current level of noise. This apparently pointless approximation is mentioned since there exists a significant intermediate computation (the Cholesky factorization) from the position calculation. This factorization can be reused to reduce execution time if the matrices in equations (4.19) and (4.32) are identical. The equation is expressed in units of cycles/sec and is time referenced to the instant that the air receiver made its phase measurement.

Figure 4.4 shows East North and Up components of the velocity estimate over one minute when the aircraft was held stationary. Six satellites were in view during this interval and the ENU components of DOP (square roots of diagonal components of the position covariance matrix) were 1.0, 1.0 and 2.8 respectively. The data gives an indication of the noise in the signal, but does not include satellite switches. Provided more than four measurements are available, such switches are not catastrophic. The

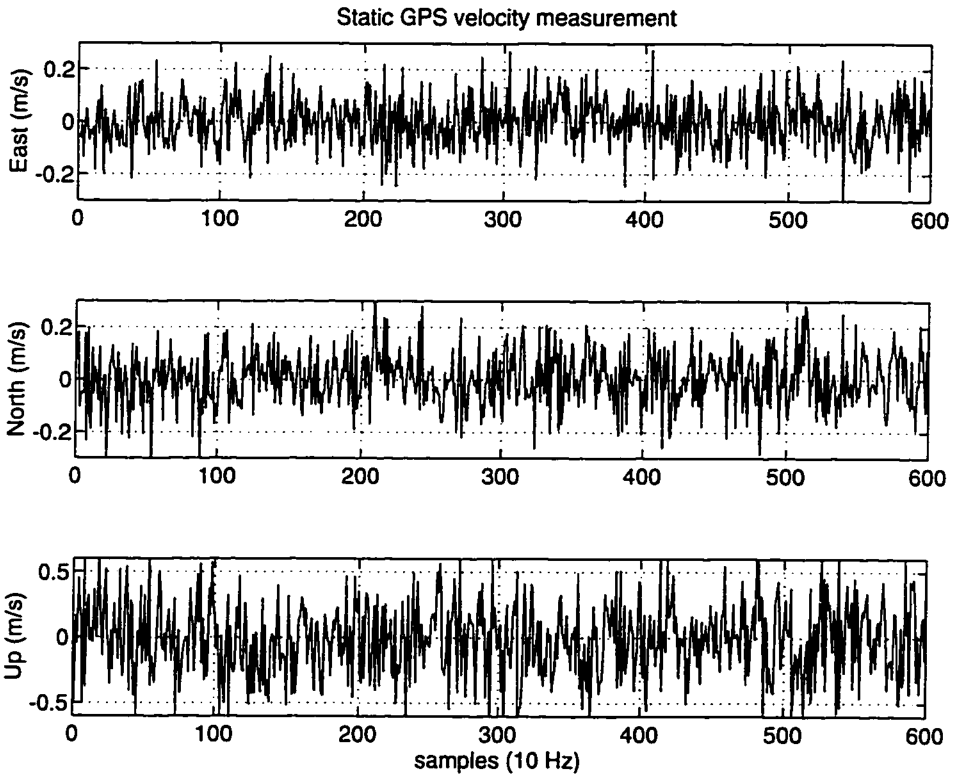


Figure 4.4: **GPS Velocity Measurement while Stationary**
10 Hz samples of East North and Up components of GPS velocity while stationary. The ENU components of DOP are 1.0, 0.9 and 2.8 respectively. Each component is zero mean with standard deviation of 0.1, 0.1 and 0.29 m/s respectively. Both reference and mobile receivers had tracking bandwidth set to 2 Hz.

result is a sudden change in the covariance of the velocity noise, but the velocity estimate remains unbiased.

A final point of note is that the cycle ambiguities are not required for the velocity solution. Since the corrections on the right hand side of equation (4.32) are small relative to the current noise, a reasonable velocity estimate can be attained independent of accurate position.

Filtering at either receiver

The quantity $\Delta\dot{\phi}_j$ is a first difference of two noisy measurements, with standard deviation given by $\sigma_{\Delta\dot{\phi}} = \sqrt{\sigma^2_{\dot{\phi}^a} + \sigma^2_{\dot{\phi}^g}}$. To minimize the combined noise, the estimator bandwidth at each receiver should be restricted to that which is required to admit the signal frequencies of interest at that receiver. In cases where the reference receiver is largely stationary, one should therefore employ a lower bandwidth estimator than is used by the mobile receiver.

As an alternative to uplinking the doppler outputs from the reference receiver. it is feasible to use the outputs from the reference phase predictor described in section 4.1.3. This obviates the necessity to uplink the reference doppler and reduces the required uplink bandwidth.

4.3 Attitude Fundamentals

Prior to discussing attitude determination, this section is included to define the nomenclature and conventions that will be used later. It introduces a number of well known relations that will be of subsequent utility.

4.3.1 Nomenclature

In the following work, vectors in $\mathbf{R}^3$ are written in boldface, ie, $\boldsymbol{v} = [v_1\, v_2\, v_3]^T$. Standard definitions for vector addition, inner product and cross product are applicable. Vectors themselves are basis independent, however, when actual computation is required, the concept of basis becomes necessary. The numbers v_1, v_2 and v_3 (called vector measure numbers [26]) *are basis dependent.* Where necessary for clarity, the vector will be subscripted with a mnemonic for the basis in which the measure numbers are known. For example, $\boldsymbol{v}_A$, indicates the measure numbers of vector $\boldsymbol{v}$ are known in the A basis. Unit vectors aligned with a labeled vector are indicated in boldface with a circumflex (ˆ), and the vector magnitude operator is indicated with vertical bars around the argument. For an arbitrary vector $\boldsymbol{v}$ one can therefore write

$$\boldsymbol{v} \equiv |\boldsymbol{v}|\ \hat{\boldsymbol{v}} \qquad (4.33)$$

Vectors are also considered to be column matrices with 3 rows, so matrix operations such as transposition make sense. This convention allows vectors to be included in vector-matrix expressions such as matrix multiplication. Orthonormal transformation between bases may be achieved by premultiplication by a direction cosine matrix, denoted $\mathcal{R}$. The direction cosine matrix to transform a vector in basis A to the equivalent vector in basis B, is denoted ${}^{B}\mathcal{R}^{A}$, so successive matrix multiplications conform in a natural way, and

$$\boldsymbol{v}_B = {}^{B}\mathcal{R}^{A}\, \boldsymbol{v}_A \tag{4.34}$$

It is also useful to define the vector cross product operator in matrix form. For an arbitrary 3 vector $\boldsymbol{v}$, the skew-symmetric cross product matrix $[[\boldsymbol{v}]]$ is defined as

$$[[\boldsymbol{v}]] \equiv \begin{bmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{bmatrix} \tag{4.35}$$

With this notation we have

$$\boldsymbol{u} \times \boldsymbol{v} = -[[\boldsymbol{u}]]\,\boldsymbol{v} \tag{4.36}$$

Some authors define instead

$$[\boldsymbol{u}\times] \equiv -[[\boldsymbol{u}]] \tag{4.37}$$

but the matrix defined in equation (4.35) turns out to be more convenient overall. Many useful matrix identities may be developed which are the matrix equivalent of well known vector identities. For example,

$$[[\boldsymbol{u}]]\,\boldsymbol{v} = -[[\boldsymbol{v}]]\,\boldsymbol{u} \tag{4.38}$$

$$\boldsymbol{v}^T\,[[\boldsymbol{u}]] = -\boldsymbol{u}^T\,[[\boldsymbol{v}]] \tag{4.39}$$

$$[[\mathcal{R}\,\boldsymbol{u}]] = \mathcal{R}\,[[\boldsymbol{u}]]\,\mathcal{R}^T \tag{4.40}$$

$$[[\boldsymbol{u}]]\,[[\boldsymbol{v}]] = -\boldsymbol{u}^T\boldsymbol{v}\,I_3 + \boldsymbol{v}\boldsymbol{u}^T \tag{4.41}$$

$$[[\boldsymbol{u}]]\,[[\boldsymbol{v}]] - [[\boldsymbol{v}]]\,[[\boldsymbol{u}]] = \boldsymbol{v}\boldsymbol{u}^T - \boldsymbol{u}\boldsymbol{v}^T = -[[\boldsymbol{u} \times \boldsymbol{v}]] \tag{4.42}$$

$$\boldsymbol{u}\boldsymbol{v}^T[[\boldsymbol{w}]] + [[\boldsymbol{w}]]\boldsymbol{v}\boldsymbol{u}^T = -[[\boldsymbol{u} \times (\boldsymbol{v} \times \boldsymbol{w})]] \tag{4.43}$$

4.3.2 Attitude Representations

Numerous different parameterizations of attitude have been studied and an extensive literature on the subject exists. A useful summary of this body of knowledge in a unified notation may be found in a recent survey paper by Shuster [51]. His notational conventions are discussed briefly below and are followed in the sequel. In this introductory section, some of the equations are transcribed directly.

In essence, attitude representation and manipulation is fundamentally nonlinear. This nonlinearity expresses itself in varying forms depending on the representation. For this reason, some representations are more advantageous than others for a particular purpose. Representation that are of interest in this work are:

- *Euler angles.* The Euler angles commonly used for aircraft work may be represented by the vector $[\psi\ \theta\ \phi]^T$ representing successive rotations about body yaw (ψ), pitch (θ) and roll (ϕ) axes. This is commonly referred to as an Euler body 321 rotation sequence. There are numerous alternative rotation orderings with different but similar characteristics. One can construct an orthogonal *direction cosine* or rotation matrix $\mathcal{R}(\psi, \theta, \phi)$, as the product of these three simple rotations as

$$\mathcal{R}(\hat{\boldsymbol{n}}_3, \hat{\boldsymbol{n}}_2, \hat{\boldsymbol{n}}_1, \psi, \theta, \phi) \equiv \mathcal{R}(\hat{\boldsymbol{n}}_1, \phi)\, \mathcal{R}(\hat{\boldsymbol{n}}_2, \theta)\, \mathcal{R}(\hat{\boldsymbol{n}}_3, \psi) \tag{4.44}$$

  where the notation $\mathcal{R}(\hat{\boldsymbol{n}}_3, \psi)$ represents a rotation by amount ψ about the vector $\hat{\boldsymbol{n}}_3$ in a right-handed sense. With Euler angles, three parameters are used to represent the three degrees of attitude freedom. Nonlinearity inherent in attitude appears as sums and products of transcendental functions in the direction cosine matrix $\mathcal{R}(\psi, \theta, \phi)$ and in the resulting kinematic relations. A serious problem with the Euler angle parameterization for unconstrained attitude is "gimbal lock". That is, there are combinations of ψ, θ , ϕ, for which the representation becomes singular. A further problem is that there is no *simple* formula to compose successive rotations, when each rotation is expressed as a three vector of Euler angles.

- *direction cosine (rotation) matrix.* The direction cosine matrix representations uses nine parameters to represent the three degrees of attitude freedom. Nonlinearity is inherent in the six constraints on matrix orthonormality, where by orthonormality is implied the property

$$\mathcal{R}^T\mathcal{R} = \mathcal{R}^T\mathcal{R} = I_{3\times3} \tag{4.45}$$

- *axis/angle representation.* Euler's theorem states that every direction cosine (*rotation*) matrix may be constructed as a rotation about a given vector, $\hat{\boldsymbol{n}}$, by a given angle, ξ, such that

  $$\mathcal{R}(\hat{\boldsymbol{n}},\xi) = cos\xi I_{3\times3} + (1 - cos\xi)\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T + sin\xi[[\hat{\boldsymbol{n}}]] \tag{4.46}$$

  The representation $(\hat{\boldsymbol{n}},\xi)$ is not unique, since $(\hat{\boldsymbol{n}},\xi)$ and $(-\hat{\boldsymbol{n}},-\xi)$ (or, equivalently $(-\hat{\boldsymbol{n}},2\pi-\xi)$) correspond to the same rotation. An important related representation is the *rotation vector* $\boldsymbol{\xi}$, defined by

  $$\boldsymbol{\xi} \equiv \xi\,\hat{\boldsymbol{n}} \tag{4.47}$$

  Of particular importance is the case where the angle of rotation, ξ, is small. In that case one writes the angle of rotation as $\Delta\xi$, and notes that $\sin(\Delta\xi) \approx \Delta\xi$, and $\cos(\Delta\xi) \approx 1$. Euler's formula, equation (4.46), becomes

  $$\mathcal{R} = I + [[\boldsymbol{\Delta\xi}]] + O(|\boldsymbol{\Delta\xi}|^2) \tag{4.48}$$

  where

  $$\boldsymbol{\Delta\xi} \equiv \Delta\xi\,\hat{\boldsymbol{n}} \tag{4.49}$$

  is the *infinitesimal rotation vector*, and $O(x^c)$ denotes a quantity which becomes proportional to x^c as x tends to zero. The components of $\boldsymbol{\Delta\xi}$ are termed the *infinitesimal angles.* Unlike finite rotations, successive infinitesimal rotations commute when quadratic and higher-order terms can be neglected.

- *Euler-Rodrigues symmetric parameters.* The Euler-Rodrigues symmetric parameters are a four-parameter representation, where nonlinearity appears as a single inner product constraint. The Euler-Rodrigues symmetric parameters have several advantages over the direction cosine matrix as an attitude representation. First, there are fewer elements (4 instead of 9), so that less storage is required. Secondly, there are fewer constraints (1 instead of 6), and the single constraint is very easy to enforce. With both direction cosine and Euler symmetric representations, both composition and attitude kinematics become linear in the representation parameters. Stupelnagel [55], proves that there is no three-parameter representation that can be both global and nonsingular. Having four parameters, the Euler-Rodrigues symmetric parameters are therefore a *minimal*, global, nonsingular representation, which also have many interesting symmetries and advantageous properties. These details are the subject of section 4.3.3.

Numerous other attitude representations exist, however the above are the most useful for our purposes.

4.3.3 Euler-Rodrigues Symmetric Parameters

Often referred to simply as *quaternions*, the Euler-Rodrigues symmetric parameters, η_1, η_2, η_3 and η_4, are usually arranged in a four-dimensional column matrix, $\bar{\eta}$, given by

$$\bar{\eta} \equiv \begin{bmatrix} \boldsymbol{\eta} \\ \eta_4 \end{bmatrix} = [\eta_1 \quad \eta_2 \quad \eta_3 \quad \eta_4]^T \tag{4.50}$$

which satisfies the normalization constraint

$$\bar{\eta}^T \bar{\eta} = |\boldsymbol{\eta}|^2 + \eta_4^2 = 1 \tag{4.51}$$

One advantage mentioned previously, is simple (*and least squares optimal*) re-normalization in cases where the constraint is not satisfied.

$$\bar{\eta} \longrightarrow (\bar{\eta}^T \bar{\eta})^{-1/2} \, \bar{\eta} \tag{4.52}$$

This may become necessary over the course of a lengthy calculation due to the accumulation of roundoff or other numerical errors.

The Euler-Rodrigues symmetric parameters are in fact a special case of a more general object, the quaternion. Every column matrix of Euler-Rodrigues symmetric parameters is a quaternion (*a quaternion of rotation*). The converse is not true, since a general quaternion will have non-unity norm. For each column matrix of Euler-Rodrigures symmetric parameters there exists a unique direction cosine matrix given by

$$\begin{aligned}\mathcal{R}(\boldsymbol{\eta},\eta_4) &= (\eta_4^2-|\boldsymbol{\eta}|^2)I_3+2\boldsymbol{\eta}\boldsymbol{\eta}^T+2\eta_4[[\boldsymbol{\eta}]] \\ &= \begin{bmatrix} \eta_1^2-\eta_2^2-\eta_3^2+\eta_4^2 & 2(\eta_1\eta_2+\eta_4\eta_3) & 2(\eta_1\eta_3-\eta_4\eta_2) \\ 2(\eta_2\eta_1-\eta_4\eta_3) & -\eta_1^2+\eta_2^2-\eta_3^2+\eta_4^2 & 2(\eta_2\eta_3+\eta_4\eta_1) \\ 2(\eta_3\eta_1+\eta_4\eta_2) & 2(\eta_3\eta_2-\eta_4\eta_1) & -\eta_1^2-\eta_2^2+\eta_3^2+\eta_4^2 \end{bmatrix}\end{aligned} \tag{4.53}$$

where I_3 is the 3×3 identity matrix and,

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \equiv \sin(\xi/2)\hat{\boldsymbol{n}}, \qquad \eta_4 \equiv \cos(\xi/2), \tag{4.54}$$

showing also the relationship to the well known axis/angle ($\hat{\boldsymbol{n}}$ and ξ) interpretation of the representation. The Euler-Rodrigues symmetric parameters may equivalently be extracted from the direction cosine matrix within a sign. This non-uniqueness is related to the non-uniqueness of the axis/angle representation. Unlike the Euler angles, which are not easily combined for successive rotations, the Euler-Rodrigues symmetric parameters have a very simple composition rule, where

$$\bar{\eta}'' = \bar{\eta}' \otimes \bar{\eta} \tag{4.55}$$

The composition operator (denoted by $\otimes$), is defined so that it is also true that

$$\mathcal{R}(\bar{\eta}'') = \mathcal{R}(\bar{\eta}')\,\mathcal{R}(\bar{\eta}) \tag{4.56}$$

Composition can also be written in terms of 4×4 matrix multiplication in two different ways

$$\bar{\eta}' \otimes \bar{\eta} = \{\bar{\eta}'\}_L\,\bar{\eta} = \{\bar{\eta}\}_R\,\bar{\eta}' \tag{4.57}$$

where

$$\{\bar{\eta}\}_L \equiv \begin{bmatrix} \eta_4 & \eta_3 & -\eta_2 & \eta_1 \\ -\eta_3 & \eta_4 & \eta_1 & \eta_2 \\ \eta_2 & -\eta_1 & \eta_4 & \eta_3 \\ -\eta_1 & -\eta_2 & -\eta_3 & \eta_4 \end{bmatrix} = \eta_4 I_4 + \begin{bmatrix} [[\boldsymbol{\eta}]] & \boldsymbol{\eta} \\ -\boldsymbol{\eta}^T & 0 \end{bmatrix} \tag{4.58}$$

and

$$\{\bar{\eta}\}_R \equiv \begin{bmatrix} \eta_4 & -\eta_3 & \eta_2 & \eta_1 \\ \eta_3 & \eta_4 & -\eta_1 & \eta_2 \\ -\eta_2 & \eta_1 & \eta_4 & \eta_3 \\ -\eta_1 & -\eta_2 & -\eta_3 & \eta_4 \end{bmatrix} = \eta_4 I_4 + \begin{bmatrix} -[[\boldsymbol{\eta}]] & \boldsymbol{\eta} \\ -\boldsymbol{\eta}^T & 0 \end{bmatrix} \tag{4.59}$$

Note that the 4×4 matrices $\{\bar{\eta}\}_L$ and $\{\bar{\eta}\}_R$ defined above are both orthonormal. Given an arbitrary vector $\bar{\eta}$, the implication is that one can generate an orthogonal basis in $\mathbf{R}^4$ by inspection, something that is not true in $\mathbf{R}^3$. The 4×3 submatrix $\Xi(\bar{\eta})$, is defined such that

$$\{\bar{\eta}\}_R \equiv [\Xi(\bar{\eta}) \mid \bar{\eta}] \tag{4.60}$$

and spans the orthogonal subspace to $\bar{\eta}$ in $\mathbf{R}^4$. The matrix $\Xi(\bar{\eta})$ has many interesting representational and kinematic properties and will later prove to be the key to the Kalman filter measurement update equations. It is easily shown that

$$\Xi^T(\bar{\eta})\,\Xi(\bar{\eta}) = I_{3\times3} \qquad \Xi(\bar{\eta})\,\Xi^T(\bar{\eta}) = I_{4\times4} - \bar{\eta}\bar{\eta}^T \tag{4.61}$$

and

$$\Xi^T(\bar{\eta})\,\bar{\eta} = \mathbf{0} \tag{4.62}$$

A further useful relation involving $\Xi(\bar{\eta})$ is given here for later reference, where the operator $\Omega_L(\boldsymbol{v})$ is defined in equation (4.76).

$$\Xi(\bar{\eta})\,\boldsymbol{v} = \Omega_L(\boldsymbol{v})\,\bar{\eta} \tag{4.63}$$

The column matrix of Euler-Rodrigues symmetric parameters has the the *unique* identity element

$$\bar{1} \equiv [0 \quad 0 \quad 0 \quad 1]^T \tag{4.64}$$

and inverse

$$\bar{\eta}^{-1} \equiv \begin{bmatrix} -\boldsymbol{\eta} \\ \eta_4 \end{bmatrix} \tag{4.65}$$

It will later be useful to note that equation (4.53) may be written as

$$\{\bar{\eta}\}_L \{\bar{\eta}^{-1}\}_R = \{\bar{\eta}^{-1}\}_R \{\bar{\eta}\}_L = \begin{bmatrix} \mathcal{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4.66}$$

It can also be shown that

$$\Xi^T(\bar{\eta})\,\Omega_L(\boldsymbol{v})\,\Xi(\bar{\eta}) = [[\boldsymbol{v}]] \tag{4.67}$$

for an arbitrary vector $\boldsymbol{v}$.

4.3.4 Kinematics

Kinematics relates the time derivative of the attitude representation to the *angular velocity vector*, $\boldsymbol{\omega}(t)$.

The Direction Cosine Matrix

If the attitude is changing with time, then $\mathcal{R}(t + \Delta t)$ will differ from $\mathcal{R}(t)$. One may write

$$\mathcal{R}(t + \Delta t) = \Phi(t + \Delta t, t)\,\mathcal{R}(t) \tag{4.68}$$

where $\Phi(t + \Delta t, t)$ is also a direction cosine matrix. For Δt sufficiently small, one may expand $\Phi(t + \Delta t, t)$ as

$$\Phi(t + \Delta t, t) = I + [[\boldsymbol{\Delta\xi}(t)]] + O(|\boldsymbol{\Delta\xi}(t)|^2), \tag{4.69}$$

where $\boldsymbol{\Delta\xi}(t)$ is a small three-vector which tends to zero as Δt tends to zero. Forming

$$\frac{1}{\Delta t}\left(\mathcal{R}(t + \Delta t) - \mathcal{R}(t)\right) = \frac{1}{\Delta t}[[\boldsymbol{\Delta\xi}(t)]]\,\mathcal{R}(t) + O(|\boldsymbol{\Delta\xi}(t)|) \tag{4.70}$$

and taking the limit as Δt tends to zero leads to the well known equation

$$\frac{d}{dt}\mathcal{R}(t) = [[\boldsymbol{\omega}(t)]]\,\mathcal{R}(t) \tag{4.71}$$

where $\omega(t)$ is defined as

$$\omega(t) \equiv \lim_{\Delta t - 0} \frac{\boldsymbol{\Delta\xi}(t)}{\Delta t} \tag{4.72}$$

The angular velocity appearing in equation (4.72) is necessarily referred to the body basis. Thus $\boldsymbol{\Delta\xi}(t)$ is the instantaneous rotation vector for the associated infinitesimal rotation and is also referred to the body basis. This fact becomes important because perturbations in $\boldsymbol{\Delta\xi}(t)$ relate to perturbations in the local body reference frame. This is convenient and physically meaningful for attitude error analysis.

The Euler-Rodrigues Symmetric Parameters

The relation for the Euler-Rodrigues symmetric parameters is very similar to that for the direction cosine matrix. Note that

$$\begin{aligned} \bar{\eta}(t+\Delta t) &= \delta\bar{\eta}(t) \otimes \bar{\eta}(t) \\ &= \{\delta\bar{\eta}(t)\}_L \bar{\eta}(t), \end{aligned} \tag{4.73}$$

where, since $\delta\bar{\eta}(t)$ describes a small rotation, it follows from the axis/angle ($\hat{\boldsymbol{n}}$ and ξ) interpretation (equation (4.54)) that

$$\delta\bar{\eta}(t) = \begin{bmatrix} \boldsymbol{\Delta\xi}(t)/2 \\ 1 \end{bmatrix} + O(|\boldsymbol{\Delta\xi}(t)|^2). \tag{4.74}$$

and

$$\{\delta\bar{\eta}(t)\}_L = I_{4\times4} + \frac{1}{2}\Omega_L(\boldsymbol{\Delta\xi}(t)) + O(|\boldsymbol{\Delta\xi}(t)|^2), \tag{4.75}$$

where

$$\Omega_L(\boldsymbol{v}) \equiv \begin{bmatrix} 0 & v_3 & -v_2 & v_1 \\ -v_3 & 0 & v_1 & v_2 \\ v_2 & -v_1 & 0 & v_3 \\ -v_1 & -v_2 & -v_3 & 0 \end{bmatrix} \tag{4.76}$$

for some arbitrary vector $\boldsymbol{v}$.

Carrying out the steps to calculate the time derivative of the Euler-Rodrigues symmetric parameters (as above for the direction cosine matrix) leads directly to

$$\frac{d}{dt}\bar{\eta} = \dot{\bar{\eta}} = \frac{1}{2}\Omega_L(\omega(t))\bar{\eta} = \frac{1}{2}\Xi(\bar{\eta})\,\omega(t) \tag{4.77}$$

The right hand equality may be verified by direct substitution of $\Xi(\bar{\eta})$ from equation (4.60) and is a direct consequence of equation (4.63).

If the direction of $\boldsymbol{\omega}(t)$ is constant over the time interval of interest, or if the *rotation vector* defined by

$$\boldsymbol{\Delta\theta} = \int_{t}^{t+\Delta t} \boldsymbol{\omega}(t')\,dt' \tag{4.78}$$

is small, then the solution of equation (4.77) is [16]

$$\bar{\eta}(t+\Delta t) = M(\boldsymbol{\Delta\theta})\,\bar{\eta}(t) \tag{4.79}$$

where

$$M(\boldsymbol{\Delta\theta}) = \cos(|\boldsymbol{\Delta\theta}|/2)\,I_{4\times 4} + \frac{\sin(|\boldsymbol{\Delta\theta}|/2)}{|\boldsymbol{\Delta\theta}|}\,\Omega_L(\boldsymbol{\Delta\theta}) \tag{4.80}$$

Equation (4.77) may be readily inverted to give

$$\bar{\omega} \equiv \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} = 2\frac{d\bar{\eta}}{dt} \otimes \bar{\eta}^{-1} \tag{4.81}$$

where the notation $\bar{\omega}$ indicates the construction of a general *quaternion* (four-vector) from an arbitrary three-vector. This can be useful for manipulation since it allows expressions such as equation (4.77) to be rewritten as

$$\dot{\bar{\eta}} = \frac{1}{2}\{\bar{\omega}\}_L\,\bar{\eta} = \frac{1}{2}\bar{\omega}\otimes\bar{\eta} \tag{4.82}$$

Similarly, the orthonormality of $\Xi(\bar{\eta})$ allows the easy inversion of the right hand of equation (4.77) to give

$$\begin{aligned}
\boldsymbol{\omega} &= 2\,\Xi^T(\bar{\eta})\,\dot{\bar{\eta}} && (4.83)\\
&= 2\begin{bmatrix} \eta_4\dot{\eta}_1 + \eta_3\dot{\eta}_2 - \eta_2\dot{\eta}_3 - \eta_1\dot{\eta}_4 \\ -\eta_3\dot{\eta}_1 + \eta_4\dot{\eta}_2 + \eta_1\dot{\eta}_3 - \eta_2\dot{\eta}_4 \\ \eta_2\dot{\eta}_1 - \eta_1\dot{\eta}_2 + \eta_4\dot{\eta}_3 - \eta_3\dot{\eta}_4 \end{bmatrix} && (4.84)\\
&= 2\,(\eta_4\dot{\boldsymbol{\eta}} - \dot{\eta}_4\boldsymbol{\eta} - \boldsymbol{\eta}\times\dot{\boldsymbol{\eta}}), && (4.85)\\
\omega^2 &= 4\,\dot{\bar{\eta}}^T\,\dot{\bar{\eta}} && (4.86)
\end{aligned}$$

which suggests a technique for differentiating successive attitude estimates to yield an angular velocity estimate.

4.3.5 A New Inner Product Identity

While studying the GPS attitude problem I discovered an interesting identity involving the inner product of two vectors, where each vector is known in a different basis. The identity relates the inner product of the vectors to the Euler Rodrigues symmetric parameters representing the rotation between the two bases. The identity is given below together with a simple derivation. The inner product arises in connection with the GPS attitude problem, where the objective is to determine the direction cosine matrix (or the related vector of Euler Rodrigues symmetric parameters). I am not aware of its previous publication and discussions with several experts in the field (Kane [26], Shuster [51], Bar-Itzhack [6]) lead me to believe that it is original. Unfortunately, I have been unable to exploit the identity to solve problems that are not more easily solvable in an alternate form. The identity is of independent (although perhaps academic) interest in that it has a perfect quadratic form in the elements of $\bar{\eta}$, and the matrix $\mathcal{Q}$ has a number of interesting properties. The following formula takes the inner product between two vectors $\boldsymbol{s}$ and $\boldsymbol{b}$, where the measure numbers of each are known in different bases, say A and B. To be explicit, one could write $\boldsymbol{s}_A$ and $\boldsymbol{b}_B$ throughout, however most of the subscripts are omitted in equations (4.87) and (4.88) to aid clarity. The notation ${}^B\mathcal{R}^A(\bar{\eta})$ indicates that the direction cosine matrix ${}^B\mathcal{R}^A$ is related to a column matrix of Euler-Rodrigues symmetric parameters $\bar{\eta}$, as given by equation (4.53). The identity is given as follows:

$$\boldsymbol{b}_B \cdot \boldsymbol{s}_A = |\boldsymbol{b}|\,|\boldsymbol{s}|\cos(\theta) = \boldsymbol{b}^T\,{}^B\mathcal{R}^A(\bar{\eta})\,\boldsymbol{s} = |\boldsymbol{b}|\,|\boldsymbol{s}|\;\bar{\eta}^T\mathcal{Q}(\hat{\boldsymbol{b}},\hat{\boldsymbol{s}})\,\bar{\eta} \tag{4.87}$$

where

$$\mathcal{Q}(\hat{\boldsymbol{b}},\hat{\boldsymbol{s}}) \equiv \begin{bmatrix} [[\hat{\boldsymbol{s}}]]\,[[\hat{\boldsymbol{b}}]] + \hat{\boldsymbol{s}}\hat{\boldsymbol{b}}^T & [[\hat{\boldsymbol{s}}]]\hat{\boldsymbol{b}} \\ \hat{\boldsymbol{s}}^T[[\hat{\boldsymbol{b}}]] & \hat{\boldsymbol{s}}^T\hat{\boldsymbol{b}} \end{bmatrix} = \begin{bmatrix} [[\hat{\boldsymbol{b}}]]\,[[\hat{\boldsymbol{s}}]] + \hat{\boldsymbol{b}}\hat{\boldsymbol{s}}^T & -[[\hat{\boldsymbol{b}}]]\hat{\boldsymbol{s}} \\ -\hat{\boldsymbol{b}}^T[[\hat{\boldsymbol{s}}]] & \hat{\boldsymbol{b}}^T\hat{\boldsymbol{s}} \end{bmatrix} \tag{4.88}$$

and θ is the angle between the two vectors $\boldsymbol{b}$ and $\boldsymbol{s}$. Note that $\mathcal{Q}$ is a very unusual matrix, being both symmetric and orthogonal, ie,

$$\mathcal{Q} = \mathcal{Q}^T = \mathcal{Q}^{-1} \tag{4.89}$$

The implication is that raising $\mathcal{Q}$ to an even power gives identity, while raising $\mathcal{Q}$ to an odd power returns itself. This is analogous to a reflection or rotation by 180 deg in $\mathbf{R}^4$. To show the identity, note that

$$\boldsymbol{b}^T \, {}^B\mathcal{R}^A(\bar{\eta}) \, \boldsymbol{s} = \begin{bmatrix} \boldsymbol{b}^T & 0 \end{bmatrix} \begin{bmatrix} {}^B\mathcal{R}^A(\bar{\eta}) & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ 0 \end{bmatrix} \tag{4.90}$$

using equation (4.66) we have,

$$\boldsymbol{b}^T \, {}^B\mathcal{R}^A(\bar{\eta}) \, \boldsymbol{s} = \begin{bmatrix} \boldsymbol{b}^T & 0 \end{bmatrix} \{\bar{\eta}\}_L \{\bar{\eta}^{-1}\}_R \begin{bmatrix} \boldsymbol{s} \\ 0 \end{bmatrix} \tag{4.91}$$

from equations (4.65) (4.58) and (4.59),

$$\begin{bmatrix} \boldsymbol{b}^T & 0 \end{bmatrix} \{\bar{\eta}\}_L \{\bar{\eta}^{-1}\}_R \begin{bmatrix} \boldsymbol{s} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}^T & 0 \end{bmatrix} \begin{bmatrix} [[\boldsymbol{\eta}]] + \eta_4 I_3 & \boldsymbol{\eta} \\ -\boldsymbol{\eta}^T & \eta_4 \end{bmatrix} \begin{bmatrix} [[\boldsymbol{\eta}]] + \eta_4 I_3 & -\boldsymbol{\eta} \\ \boldsymbol{\eta}^T & \eta_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ 0 \end{bmatrix} \tag{4.92}$$

and rearranging,

$$\begin{bmatrix} \boldsymbol{b}^T & 0 \end{bmatrix} \{\bar{\eta}\}_L \{\bar{\eta}^{-1}\}_R \begin{bmatrix} \boldsymbol{s} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}^T & \eta_4 \end{bmatrix} \begin{bmatrix} -[[\boldsymbol{b}]] & \boldsymbol{b} \\ \boldsymbol{b}^T & 0 \end{bmatrix} \begin{bmatrix} -[[\boldsymbol{s}]] & \boldsymbol{s} \\ \boldsymbol{s}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \eta_4 \end{bmatrix} \tag{4.93}$$

Multiplying the two central matrices and factoring out the scalar magnitudes of $\boldsymbol{b}$ and $\boldsymbol{s}$ gives equation (4.88). Equation (4.41) may be used to transfer to the alternate form given. Orthonormality and symmetry of $\mathcal{Q}$ can be shown by direct substitution into the definitions and manipulation using the cross product matrix identities.

4.4 Attitude Determination

The techniques used for attitude and angular velocity determination are very similar to those employed for position and velocity. In a sense, the attitude problem is easier, since all measurements are made relative to a common clock, and the small timing corrections used in the position and velocity calculations are not required. With all measurements made locally, a further bonus is that the datalink component can be removed entirely. On the negative side, since there are many more measurements involved, the integer resolution problem becomes more complex.

4.4.1 Attitude

As discussed in section 3.1.2, the primary output of the multiplexed attitude receiver are the differential phase measurements $\Delta\phi_{ij}$. Unit line of sight vectors $\hat{\boldsymbol{s}}_j$, are also output at 30 second intervals, expressed in the ENU basis. Vehicle attitude is related

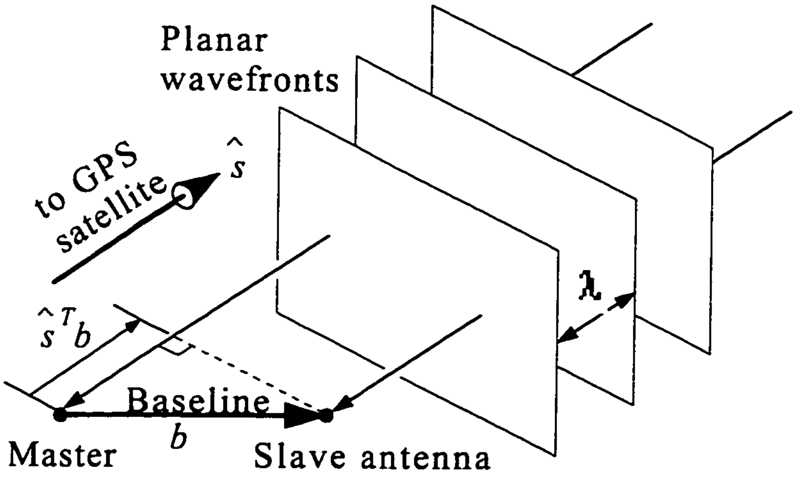


Figure 4.5: **Differential phase Inner Product Geometry.**

Geometry of planar wavefronts from satellite $\hat{j}$, across baseline b.

to differential phase measurements according to the scalar equation

$$\boldsymbol{b}_i^T \mathcal{R}(\bar{\eta})\hat{\boldsymbol{s}}_j = \Delta\phi_{ij} + N_{ij} + \beta_i + \nu_{ij} \tag{4.94}$$

Here, the vector $\boldsymbol{b}_i$, called baseline i, is the vector between the master antenna and the i'th slave antenna in units of wavelengths as shown in figure 2.12. N_{ij} is an integer number of carrier wavelengths that must be determined before operational attitude calculations can commence. In our case, three baseline vectors (i=1..3), are known in the body basis from pre-survey, together with associated line biases, β_i, (attributable to the electrical length of each baseline). To reduce clutter, it will be assumed in all subsequent discussion that the line bias β_i has been included with the phase measurement, (i.e. $\Delta\phi_{ij} \leftarrow \Delta\phi_{ij} + \beta_i$). This is done as a first step in any computation.

We desire to find the *rotation matrix* $\mathcal{R}(\bar{\eta})$, which represents the aircraft attitude. The notation indicates the 1 to 1 mapping between the orthogonal matrix $\mathcal{R}$ and a column matrix of Euler-Rodrigues symmetric parameters, $\bar{\eta}$. The equation is subject to additive noise ν_{ij}, as discussed in section 3.1.2. Figure 4.5 shows a diagram of the differential phase geometry for a single baseline and satellite. This diagram shows a differential phase measurements made at a "slave" antenna, relative to another antenna designated "master". Baseline $\boldsymbol{b}_i$ is the vector from the master to the slave antenna. The "master" tracks the absolute GPS phase and removes the phase changes due to translation. The "slave" antennae track carrier phase variations due to changes in orientation.

Operational Attitude

To calculate the vehicle attitude, represented by $\mathcal{R}$ (or $\bar{\eta}$), one wishes to solve a set of equations of the form given in equation (4.94), where the integers N_{ij} are assumed known. Since $\mathcal{R}$ is a function of three independent variables, one needs at least three independent measurements. In practice, there tend to be many more than three measurements available, and one solves the set of equations in a least-squares sense.

One wishes to find $\mathcal{R}$ that minimizes the cost function

$$J(\mathcal{R}) = \sum_{i \in \{baselines\}} \sum_{j \in \{satellites\}} \left(\boldsymbol{b}_i^T \mathcal{R}(\bar{\eta}) \hat{\boldsymbol{s}}_j - \Delta\phi_{ij} - N_{ij} \right)^2 \tag{4.95}$$

This is a nonlinear equation in three variables. Given an initial guess that is close to the minimizer, the process of linearizing about the current solution and solving, works well in practice. This is the "normal" method presented by Cohen [11]. In operation at 10 Hz, the attitude solution at a given time step is close enough to the solution at the subsequent time step to provide a suitable starting point for that iteration. Cohen also presented Wahba's method in a form suitable for the efficient solution of the GPS attitude problem. This technique is applicable when the antenna array is non-coplanar (at least four antennae) and each antenna has good satellite visibility. This fast algorithm was not feasible on our aircraft due to a nearly planar baseline configuration.

"Normal" Method: Local Linearize and Solve

Perturbing equation (4.94) about the current best guess $\mathcal{R}_k$, one has

$$\boldsymbol{b}_i^T(\mathcal{R}_k + \delta\mathcal{R})\hat{\boldsymbol{s}}_j = \Delta\phi_{ij} + N_{ij} + \nu_{ij} \tag{4.96}$$

Defining $\delta\phi_{ij} \equiv \Delta\phi_{ij}+N_{ij}-\boldsymbol{b}_i^T\,\mathcal{R}_k\,\hat{\boldsymbol{s}}_j$, and ignoring noise, gives the scalar perturbation equation

$$\begin{aligned} \boldsymbol{b}_i^T\,\delta\mathcal{R}\,\hat{\boldsymbol{s}}_j &= \delta\phi_{ij} && (4.97)\\ \boldsymbol{b}_i^T\,[[\boldsymbol{\Delta\xi}]]\,\mathcal{R}_k\,\hat{\boldsymbol{s}}_j &= \delta\phi_{ij} && (4.98)\\ \hat{\boldsymbol{s}}_j^T\,\mathcal{R}_k^T\,[[\boldsymbol{b}_i]]\,\boldsymbol{\Delta\xi} &= \delta\phi_{ij} && (4.99) \end{aligned}$$

where the row matrix $\hat{\boldsymbol{s}}_j^T\,\mathcal{R}_k^T\,[[\boldsymbol{b}_i]] = \partial\delta\phi_{ij}/\partial\boldsymbol{\Delta\xi}$, is the phase sensitivity to angular perturbations about roll, pitch and heading axes, relative to the local orientation. Defining a matrix $\mathcal{S}_i$ with a column for each satellite $i_a..i_n$ seen across baseline i as

$$\mathcal{S}_i \equiv [\hat{\boldsymbol{s}}_{i_a}\hat{\boldsymbol{s}}_{i_b}\cdots\hat{\boldsymbol{s}}_{i_n}] \tag{4.100}$$

one can construct a matrix equation to be solved for $\boldsymbol{\Delta\xi}_k$ in a least squares sense

$$\begin{bmatrix} \mathcal{S}_1^T\,\mathcal{R}_k^T\,[[\boldsymbol{b}_1]] \\ \mathcal{S}_2^T\,\mathcal{R}_k^T\,[[\boldsymbol{b}_2]] \\ \mathcal{S}_3^T\,\mathcal{R}_k^T\,[[\boldsymbol{b}_3]] \end{bmatrix} \boldsymbol{\Delta\xi}_k = \begin{bmatrix} \delta\phi_{1_a} \\ \vdots \\ \delta\phi_{1_n} \\ \delta\phi_{2_a} \\ \vdots \\ \delta\phi_{2_n} \\ \delta\phi_{3_a} \\ \vdots \\ \delta\phi_{3_n} \end{bmatrix}_k \tag{4.101}$$

where the *sensitivity matrix* on the left is defined as H. Updating the attitude estimate with the perturbation solution $\boldsymbol{\Delta\xi}$, conceptually as, $\mathcal{R}_{k+1} = \mathcal{R}_k + [[\boldsymbol{\Delta\xi}]]\,\mathcal{R}_k$, the iteration continues until the residual stops decreasing. This usually happens within two or three iterations if the initial guess is close. The solution vector may be augmented with an aero-elastic (wing flexure) state if the baselines are perturbed in flight more than a significant fraction of a wavelength. It should be noted that this algorithm uses data from the current sample only, and does not weight earlier data into the current estimate. Given the degree of redundancy ($\leq$ 18 measurements to determine 3 unknowns), GPS attitude determination was found to be robust relative to the position solution, where the degree of data redundancy is greatly reduced ($\leq$ 6 measurements to determine 4 unknowns).

After convergence, the attitude covariance about the solution, in terms of body referenced angular perturbations is given by

$$P_{\Delta\xi\Delta\xi} \equiv E\{\boldsymbol{\Delta\xi}\,\boldsymbol{\Delta\xi}^T\} = (H^T\,V^{-1}\,H)^{-1} \tag{4.102}$$

where V is a diagonal matrix based on the signal strength of each measurement and H is constructed from the converged attitude estimate.

Due to nonlinearity in the cost function, it is not surprising that the above technique does not work well in cases where the initial attitude estimate is greatly in error. To address this problem, but *primarily* as an introduction to the more difficult problem of attitude integer resolution, an alternate approach is presented. This "pseudo-global" technique does work well with large initialization errors and in cases where the baseline array is nearly (or entirely) co-planar or when satellite visibility across baselines is poor.

Pseudo-Global Minimization

In this algorithm (described in detail in a paper by Conway and myself [34]), we consider an estimate of the attitude $\mathcal{R}$, and allow variation in *one* degree of freedom while holding the other two degrees of freedom fixed. Consider the problem of finding the best value of ψ, such that an updated matrix $\mathcal{R}'$ defined as

$$\mathcal{R}' = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathcal{R} \tag{4.103}$$

has the smallest cost function $J(\mathcal{R})$. Although we have considered ψ to be the free parameter above, an analogous *simple rotation matrix* is applicable to the other degrees of attitude freedom. All give a cost function of the form

$$\begin{aligned} J(\mathcal{R},\psi) &= \sum_{\substack{i\in\{\text{baselines}\}\\ j\in\{\text{satellites}\}}} \left(\boldsymbol{b}_i^T \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathcal{R}\,\hat{\boldsymbol{s}}_j - \Delta\phi_{ij} - N_{ij} \right)^2 \\ &= \sum_{\substack{i\in\{\text{baselines}\}\\ j\in\{\text{satellites}\}}} \left(a_1\cos\psi + a_2\sin\psi + a_3\right)^2 \\ &= c_1\cos^2\psi + c_2\sin^2\psi + c_3\cos\psi\sin\psi \\ &\quad +c_4\cos\psi + c_5\sin\psi + c_6 \\ &= d_1\cos(2\psi+\xi_1) + d_2\cos(\psi+\xi_2) + d_3 \end{aligned} \tag{4.104}$$

where a_i, c_i, d_i, and ξ_i are numbers that can be readily and efficiently calculated for given $\mathcal{R}$, phases $\Delta\phi_{ij}$, integers N_{ij}, baselines $\boldsymbol{b}_i$ and line-of-sight unit vectors $\hat{\boldsymbol{s}}_j$. One wishes to find the value of ψ that minimizes $J(\mathcal{R},\psi)$. Equivalently, one wants to find the minimum of $d_1\cos(2\psi+\xi_1)+d_2\cos(\psi+\xi_2)$ as d_3 is irrelevant to the minimization. One can use simple trigonometric identities and manipulations of ξ_1 and ξ_2 to ensure that $d_1 \geq 0$ and $d_2 \geq 0$ without loss of generality. If d_1 is near (or equal to) zero, and d_2 is not, then the solution is trivial. If both are near zero, then the result is not important anyway as ψ has no real effect on the cost. Otherwise, dividing by d_1, and making the substitution $x = \psi + \frac{1}{2}\xi_1$, $B = d_2/d_1 \geq 0$ and $\phi = \xi_2 - \frac{1}{2}\xi_1$, results in a cost function that requires solution in x to minimize

$$C(x) = \cos 2x + B\cos(x+\phi) \tag{4.105}$$

The value of x that minimizes this equation can be found efficiently using a two dimensional table lookup function. This function, described in Appendix B, returns x_{min} given arguments B and ϕ. (i.e. $x_{min} = \mathcal{F}(B,\phi)$). Given x_{min}, one can apply the above coordinate transformations in reverse to obtain the optimal value of ψ, which is then used to update the attitude estimate.

This process is repeated for rotations around the other two axes, with appropriate changes of variable. Together, these are considered to be a single, "pseudo-global" iteration of the new algorithm. Because the new algorithm can jump over ridges in the cost function, one iteration is generally superior to an iteration of the linearize-and-solve method when the estimate is far from the solution. Near the solution. linearize-and-solve provides quadratic convergence and is therefore superior for later steps of iteration. For this reason, two iterations of the global algorithm followed by linearize-and-solve until convergence works well and was implemented in operation for the case with known integers.

In practice, the addition of the global algorithm is not a large improvement when the integer ambiguities are known (as is the case here), since unaided linearize-and-solve works adequately if the attitude change between successive updates is small[4]. The reason for introducing the new algorithm is its value in solving the more complex cost function that appears when the integers are not known.

4.4.2 Motion Based Attitude Integer Resolution

Assuming the integers are known, with $n \geq 3$ independent measurements, there is sufficient information to determine the 3 unknowns represented by $\mathcal{R}$. If the integers are not known, there are effectively n equations in $n + 3$ unknowns, which is clearly insufficient for any n. The solution to this problem is to observe that, provided carrier lock is maintained, the integers are not time dependent. If N measurements sets are taken over a period of time (the data collection interval), there will be Nn equations in $n + 3N$ unknowns, which is sufficient for reasonably large N and n (in practice, we used $N = 20$ and required n to be at least 5). The technique is termed *motion based*, since the orientation of the antenna array must change between measurements to provide observability.

Cohen [11], describes an elegant motion based algorithm (using the solution to Wahba's problem) to determine an initial estimate of the attitude history and cycle integers over the data collection interval. Unfortunately, the algorithm assumes that the antenna array is significantly non-coplanar, and it requires each of the slave antennae to observe phase measurements from at least three satellites in common with the master antenna. These restrictions render the technique unusable with a

[4] Small implies less than 10 degrees.

planar (or nearly planar) array or in cases where satellite occlusion due to mechanical mounting constraints is a problem.

This section describes an alternative algorithm that works in conditions of poor satellite visibility and with antennae arrays which may be coplanar (possibly having only three antennae). The objective of the algorithm is to provide an estimate of the attitude history and cycle integers over the data collection interval. The results of the algorithm are then used to initialize a subsequent refinement step, which iterates using linearize-and-solve until convergence.

We form a cost function similar to the integer-known case, but which includes N measurement sets from different times in the collection interval. Unknowns are now the attitude $\mathcal{R}_k$ at each timestep and the cycle integers N_{ij} which are constant across all timesteps. The cost function is a sum of squared errors over all measurement sets.

$$J\left(\{\mathcal{R}_k\}, \{N_{ij}\}\right) = \sum_{k \in \{\text{times}\}} \sum_{i \in \{\text{baselines}\}} \sum_{j \in \{\text{satellites}\}} \left(\boldsymbol{b}_i^T \mathcal{R}_k \hat{\boldsymbol{s}}_j - \Delta\phi_{ij_k} - N_{ij}\right)^2 \quad (4.106)$$

This is a more difficult nonlinear minimization than the integer-known case, having the same nonlinearities but with many more variables (typically on the order of 70). In this case, linearize-and-solve from an arbitrary starting point tends to work poorly or not at all. The problem has been tackled for a single baseline in [46], and improved in [11] by using the known constraints between multiple baselines. The algorithm presented in [11] was used to find an initial estimate of the attitude history and the cycle integers over the data collection interval which was subsequently refined using linearize-and-solve.

The pseudo-global solution for the integer-known case (described in detail above) is readily generalized from minimizing equation (4.104) to minimizing equation (4.106). Using the current estimate of the integers, one iteration of the pseudo-global algorithm now entails pseudo-global minimization with respect to the three attitude parameters for each time step sequentially, followed by minimization with respect to the integers for each satellite-baseline pair.

Minimizing with respect to an attitude degree of freedom for a given time step follows almost exactly the steps described in the previous section. Equation (4.104) becomes

$$\begin{aligned} J(\psi_k) &= \sum_{\substack{i\in\{\text{baselines}\}\\ j\in\{\text{satellites}\}}} \left(\boldsymbol{b}_i^T \begin{bmatrix} \cos\psi_k & \sin\psi_k & 0 \\ -\sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathcal{R}_k\, \hat{\boldsymbol{s}}_j - \Delta\phi_{ij_k} \right)^2 \\ &\qquad + \text{costs independent of } \psi_k \\ &= \begin{array}{c} c_1 \cos^2\psi_k + c_2 \sin^2\psi_k + c_3 \cos\psi_k \sin\psi_k \\ +c_4 \cos\psi_k + c_5 \sin\psi_k + c_6 \end{array} \end{aligned} \tag{4.107}$$

Equation (4.107) is now in the same form as equation (4.104), and the optimal value of ψ_k can be found in the same fashion. After processing the $3N$ angle variables, it is necessary to minimize over the integers. This is achieved by differentiating the cost with respect to the integers and equating to zero. The best estimate of an integer N_{ij} is therefore given by

$$N_{ij} = \frac{1}{N} \sum_{k\in\{\text{times}\}} \boldsymbol{b}_i^T \mathcal{R}\, \hat{\boldsymbol{s}}_j - \Delta\phi_{ij} \tag{4.108}$$

Cycle slips or other inconsistencies flagged during the data collection interval will result in the corresponding baseline-satellite pair being removed from the minimization.

To summarize: one iteration of the pseudo-global integer resolution algorithm consists of minimizing with respect to $3N$ attitude variables and then with respect to a set of integers, one integer for each baseline-satellite pair for which measurements were valid over the collection interval. As previously, the pseudo-global algorithm works better than linearize-and-solve if the initial guess is poor, but linearize-and-solve converges more rapidly when the estimate is close (within 10 deg) to the solution.

The algorithm is initialized with $\mathcal{R}_k$ set to the identity for all k, and with each integer N_{ij} set to the mean of $\Delta\phi_{ij}$ over the collection interval.

When the integers are not known, a screening step prevents the algorithm from commencing until sufficient phase change has been observed across each baseline. This process attempts to ensure that sufficient angular change has accumulated during the preceding interval to guarantee solution observability.

Two slight variations are useful in special cases:

1. When it is known that most of the attitude error is in azimuth, one should perform a global minimization with respect to azimuth alone on the first iteration.

This would be appropriate for an aircraft on the runway since pitch and roll are known to be approximately zero.

2. One can modify the local iteration step to do a pseudo-global step if the matrices used during refinement ever become numerically ill-conditioned.

The overall algorithm to minimize equation (4.106) therefore consists of several iterations of the pseudo-global or local algorithms. Simulations show that approximately 20 iterations of the pseudo-global algorithm are needed to find a reasonable first estimate for the subsequent refinement algorithm, which is then applied until the cost function stops decreasing (an increase on the first local iteration is common and is not counted). A slightly better algorithm consists of alternating pairs of pseudo-global and local iteration steps. Dividing into pairs is beneficial since a local step may cause the solution to jump along a valley, greatly increasing the cost function. A second local iteration is usually better at repairing this damage than a pseudo-global step.

It should be noted that this algorithm is *not* guaranteed to converge to the correct answer. The collected data may not contain sufficient information, either due to inadequate baseline motion or poor satellite geometry. However, once a solution has been found one may apply a number of "sanity" checks

1. The value of the cost function must be sufficiently small that it is reasonably attributed to measurement noise.

2. The "integers" N_{ij} which are estimated as continuous values should be close to integer values.

3. Subsequent attitude solutions, which depend on the integers should have reasonable residuals.

To increase reliability further, two alternatives are proposed. First, one could store the phase measurements for the short duration preceding and during integer resolution (the integer resolution process takes approximately half a second on a 66MHz 486), and use them for verification. Secondly, by (numerically) computing the partial derivative of changes in N_{ij} to the measured phases one can bound the expected noise in the integer estimates. This can be used to decide if a non-integer estimate of N_{ij} is reasonable in a statistical sense.

The algorithm has been successfully applied in four distinct cases: In simulation, on a 1 meter test array, on a model helicopter and on the 12 foot aircraft described here. It is reasonable to believe that the technique is equally applicable to longer baselines. One 360 degree rotation in azimuth was usually sufficient to solve for the integers on the test array and on the model helicopter. In flight, the aircraft attitude integers were successfully reacquired on numerous occasions by slowly rolling and pitching the aircraft through approximately ± 20 deg. Although not extensively tested, the algorithm was observed to correctly acquire the integers with a three antenna (coplanar) array. This occurred on one occasion when the nose antenna was inadvertently disconnected.

4.4.3 Static Technique for Attitude Integer Resolution

Due to its physical size and weight, it was inconvenient to manually provide sufficient angular motion for attitude initialization of the aircraft while it was on the ground. A static technique was developed to aid in this problem. Using this technique the aircraft was placed in an approximately known attitude ($\hat{\bar{\eta}}$) and the integers resolved prior to takeoff using equation (4.94). The algorithm for finding the integer estimate $\hat{N}_{ij}$ is simply to round to the nearest integer

$$\hat{N}_{ij} = floor\left(0.5 + \boldsymbol{b}_i^T\, \mathcal{R}(\hat{\bar{\eta}})\, \hat{\boldsymbol{s}}_j - \Delta\phi_{ij}\right) \tag{4.109}$$

Clearly any phase residual will be < 0.5 cycles with this approach. If a residual close to zero is obtained then one might feel confident that the correct integer has been obtained. One way to increase this probability is to set a tolerance on the acceptance of each integer, requiring that the phase residual be smaller than some specific value, eg. $\Delta\phi_{ij_{resid}} < 0.3$ carrier wavelengths. Of course it is still possible for an incorrect integer to be obtained if the baselines are long enough and the initial attitude estimate is in error. An alternate approach is to use the attitude covariance together with the phase sensitivity (evaluated at the attitude estimate) to determine if the residual is acceptable in a probabilistic sense. With this approach, one requires that

$$|\delta\phi_{ij}| = \left| \left.\frac{\partial\Delta\phi_{ij}}{\partial\boldsymbol{\Delta\xi}}\right|_{\hat{\bar{\eta}}} \begin{bmatrix} \sqrt{P_{\Delta\xi\Delta\xi_{1,1}}} \\ \sqrt{P_{\Delta\xi\Delta\xi_{2,2}}} \\ \sqrt{P_{\Delta\xi\Delta\xi_{3,3}}} \end{bmatrix} \right| < \frac{0.5}{n\sigma} \qquad (4.110)$$

where $P_{\Delta\xi\Delta\xi}$ is defined in equation (4.102) and n is the sigma-confidence number required. The 3 × 1 vector is created from the square roots of the diagonal elements of $P_{\Delta\xi\Delta\xi}$. Setting n=2 gives a confidence of 95 percent that the integer has been correctly determined. This technique was used to initialize the aircraft attitude on the runway prior to takeoff. A motion based method could also have been used but the method described above was found to be faster and less physically demanding. Since the runway is essentially flat, the pitch and roll estimates were set to zero and their standard deviation estimates to 1 deg. Since the azimuth was set by the pilot prior to takeoff, its estimate was set to the runway direction, and its standard deviation set to 4 deg. It was assumed that the pilot was capable of aligning the aircraft with the runway centerline to within a few degrees. This was not difficult in practice. With up to 18 measurements and the baselines of the aircraft used here, it was found that one could always accept enough integers to run through the attitude code once, thereby refining the estimate and reducing the covariance sufficiently that additional integers could be brought online. Since the requirements on initialization accuracy increase with baseline length, the technique is limited by this factor as described by equation (4.110). However, since the least sensitive measurements are acquired first, and others become available after subsequent refinement, the algorithm was found to be more robust than anticipated.

The technique is equally applicable in cases where one wishes to bring integers online during flight when an estimate of the attitude and its covariance are known. The addition of inexpensive gyros makes such estimates available even after brief but total GPS outages and is described in chapter 6.

To summarize, with baselines on the order of a wavelength, integer acquisition is feasible using a constrained search and with no initial knowledge. For somewhat larger baselines, it is possible to acquire the integers by placing the array in a known orientation and backing out the integers as described above. For large baselines, this technique may become infeasible due to the constraints on initial accuracy, and motion based techniques become necessary. If there are are sufficient antennae that both short and long baselines can be used, one of the shorter baseline techniques can be used in a coarse acquisition mode to acquire the integers for the longer baselines.

Patching

As with the position solution, when a new measurement becomes available it can be included in the least squares attitude determination only after its cycle ambiguity has been determined. This can be done as described above, making use of the known attitude covariance to establish probabilistic bounds on integer acceptance. Although improbable, it is possible to acquire an incorrect integer. This would result in a biased attitude estimate and an increase in the solution residual. With a largely overdetermined set of measurements, it is relatively straightforward to determine which integer is incorrect and to exclude it using the parity-space methods described by Pervan [45] for position RAIM[5].

4.4.4 Special Case: Rotation about a known Axis

In the special case where motion takes place about a known axis, determination of the cycle integers becomes easier. An example of applicability might be a car or aircraft on level ground performing a complete or partial rotation. Alternatively, the technique could be employed for a spacecraft rotating quickly relative to GPS line of sight changes. A brief description of the algorithm follows.

It is clear that after rotating through 360 degrees, the differential-phase measurements ($\Delta\phi_{ij}$) will return to their initial values, and it is easily shown that the phase for each ij-pair varies sinusoidally with the rotation parameter. Each sinusoid is offset from zero by the unknown cycle ambiguity plus other *known* terms. Subtracting the known terms, the value of the sinusoid averaged over an entire rotation is the desired

[5] Receiver Autonomous Integrity Monitoring

integer ambiguity. To eliminate the necessity of a full rotation, consider plotting phase measurements from two separate ij-pairs against each other after subtracting off the known terms. The locus of points generated over a full rotation then traces a closed ellipsiodal path. This can be transformed into a circle of known radius by an appropriate coordinate transformation. The cartesian coordinates of the center of the circle are the unknown integers for the two measurements being plotted. It is straightforward to determine this centerpoint from the data given even a partial rotation, particularly since the center point is known to have integer coordinates. This observation forms the basis of the algorithm for finding the cycle ambiguities. One proceeds to find the integers in pairs, or knowing one integer to find all other integers sequentially. Since many potential error terms average out over a full rotation, simulation suggests this technique works well even under adverse conditions or when the rotation axis in known only approximately, as might occur when rotating on ground that is not level.

4.4.5 Self Survey and Reference Attitude Datum

Prior to operational attitude, the quantities $\boldsymbol{b}_i$ and β_i must be determined. This was done by surveying the stationary antenna array over many hours, using satellite motion to provide observability. One input to this process is the attitude of the vehicle while data are collected. This then forms the reference attitude datum. The reference attitude was defined to be zero roll and the flat bottoms of the wings level in pitch. The aircraft was placed in this attitude prior to attitude self survey using careful measurements with a spirit level. The attitude reference in pitch was chosen to provide a convenient reference for gyro axis alignment and for zeroing the alpha vane. In calculating attitude perturbations for feedback control, one must however correct for the difference between this reference frame and the aircraft "stability" axes. For the model aircraft tested, the pitch axes differ by 7 deg at the nominal flight condition.

4.5 GPS Angular Velocity

To the knowledge of the author, GPS has not previously been used as an angular velocity sensor. The possibility was briefly discussed in [11] and was first reduced to practice as a result of collaborative work with Uematsu [44]. Angular velocity completes the battery of sensor capabilities available from GPS and provides a useful signal to add damping (*lead*) in feedback control. The signal is similar to GPS velocity in that differential phase-rate $\Delta\dot{\phi}_{ij}$, is used in its calculation. The angular velocity signal, although unbiased, is consequently more noisy than the analogous attitude output.

Two techniques for angular velocity generation were considered and experimentally compared. These were:

1. Differentiate the GPS attitude signal directly

2. Use the phase-rate signal from the differential phase estimator

The first technique employed equation (4.83). The quantity $\dot{\bar{\eta}}$ was formed as a first difference of the attitude quaternion, subject to the constraint $\bar{\eta}^T\dot{\bar{\eta}} = 0$. This and other techniques for differentiating the attitude signal had the disadvantage of introducing significant noise and delay (c.f. velocity). The second approach was more natural to implement, and provided a method to balance the compromise between the resultant noise and the delay introduced. This technique was implemented in operation and is presented in the following section.

4.5.1 Operational Angular Velocity

Differentiating equation (4.94) one has

$$\boldsymbol{b}_i^T \dot{\mathcal{R}} \, \hat{\boldsymbol{s}}_j + \boldsymbol{b}_i^T \mathcal{R} \, \dot{\hat{\boldsymbol{s}}}_j = \Delta\dot{\phi}_{ij} \tag{4.111}$$

Noting that $\dot{\mathcal{R}} = [[\boldsymbol{w}]] \, \mathcal{R}$, it follows that

$$\hat{\boldsymbol{s}}_j^T \mathcal{R}^T [[\boldsymbol{b}_i]] \, \omega = \Delta\dot{\phi}_{ij} - \boldsymbol{b}_i^T \mathcal{R} \, \dot{\hat{\boldsymbol{s}}}_j \tag{4.112}$$

As earlier, stacking up many measurements of this form gives a matrix equation that can be solved for $\boldsymbol{\omega}$, the body referenced angular velocity. The term $\boldsymbol{b}_i^T \mathcal{R}\, \dot{\boldsymbol{s}}_j$ is neglected as small relative to the noise.

$$\begin{bmatrix} \mathcal{S}_1^T \mathcal{R}_k^T [[\boldsymbol{b}_1]] \\ \mathcal{S}_2^T \mathcal{R}_k^T [[\boldsymbol{b}_2]] \\ \mathcal{S}_3^T \mathcal{R}_k^T [[\boldsymbol{b}_3]] \end{bmatrix} \boldsymbol{\omega}_k = \begin{bmatrix} \Delta\dot{\phi}_{1_a} \\ \vdots \\ \Delta\dot{\phi}_{3_n} \end{bmatrix}_k \tag{4.113}$$

Note that although the angular velocity is apparently independent of the cycle ambiguities, it is dependent on knowledge of the vehicle attitude, which in turn requires knowledge of the integers. The angular velocity calculation is performed immediately after the attitude calculation and reuses the Cholesky factorization of the squared observation matrix, $H^T H$ just completed for this purpose. For this reason, the additional computational burden of GPS angular velocity is minor. Again, it is reasonable to use the signal strength as a metric for weighted least squares, and the diagonal elements of $(H^T V^{-1} H)^{-1}$ have the interpretation of angular velocity error covariance. One can define measures of attitude dilution of precision (ADOP) on this basis.

As with attitude determination, it is common for the angular velocity calculation to be heavily overdetermined, having up to 18 measurements to determine 3 unknowns with the current receiver. Unfortunately, for the same reason that GPS velocity is noisier than position, so GPS angular velocity is noisier than attitude.

Flexure

In cases where aero-elastic effects are significant, "angular velocity" is defined as the angular rate of an imaginary (*rigid*) wire frame connecting the nominal locations of the antennae in the array. In performing the calculation for flexible structures, it becomes important to account for flex-rate terms which may be large. One derives a state element for wing flex-rate by differentiation of the attitude state in which wing flexure has been included. The first, symmetric flex-rate mode is observable, but to first order the second, antisymmetric mode (which will be excited by aileron) is unobservable, being indistinguishable from roll rate (in fact this is a legitimate definition of roll rate).

4.5.2 Limits of Performance

Factoring out baseline length, angular velocity performance is limited by the noise in the filtered phase-rate estimates. Assuming nominal signal strength (~ 20 AMU), one can expect RMS noise in the differential phase I-Q measurements ($\Delta\phi_i$) of approximately 5 degrees of phase. It is clear from the definition of the error covariance matrix that the standard deviation of both attitude and angular velocity estimates vary inversely with baseline length. In effect, angular uncertainty is diluted with increasing baseline length. The differential phase estimator has a transfer function

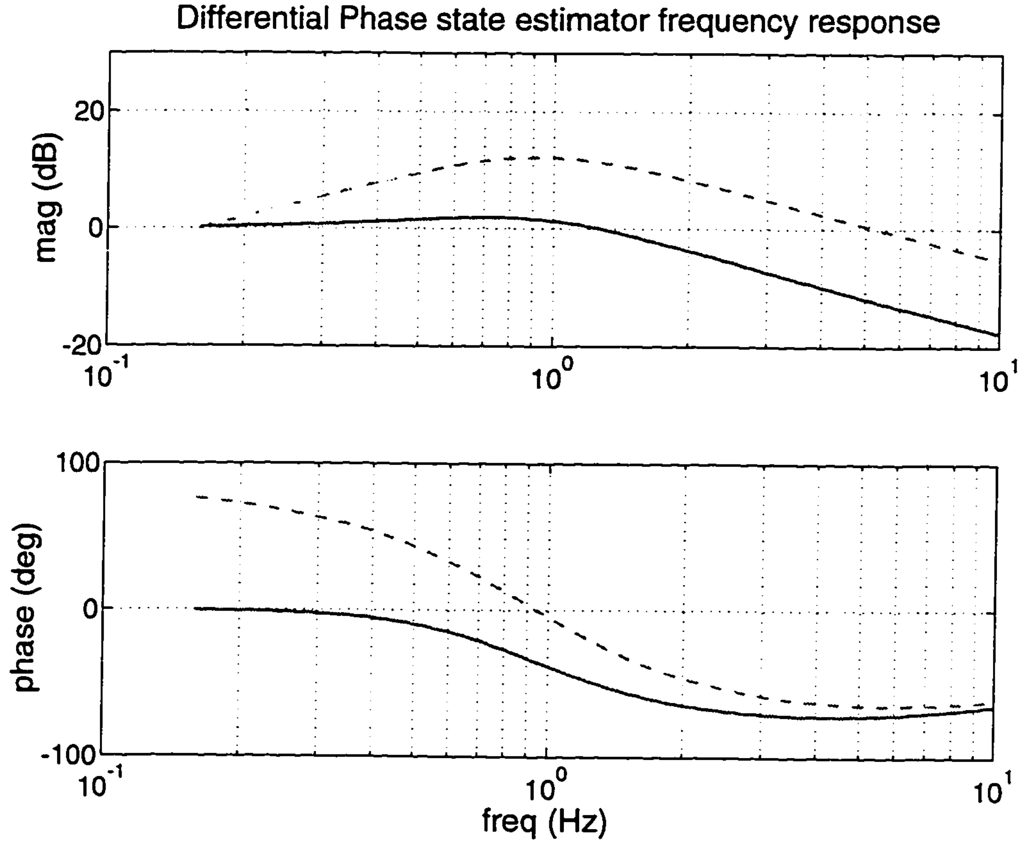


Figure 4.6: **Differential Phase Estimator Frequency Response.** Based on a sample period of 12 msec, and a bandwidth of 2 Hz. The solid line is the frequency response from the $\Delta\phi_i$ to $\hat{\phi}_i$, and the dashed line is the frequency response from $\Delta\phi_i$ to $\dot{\hat{\phi}}_i$.

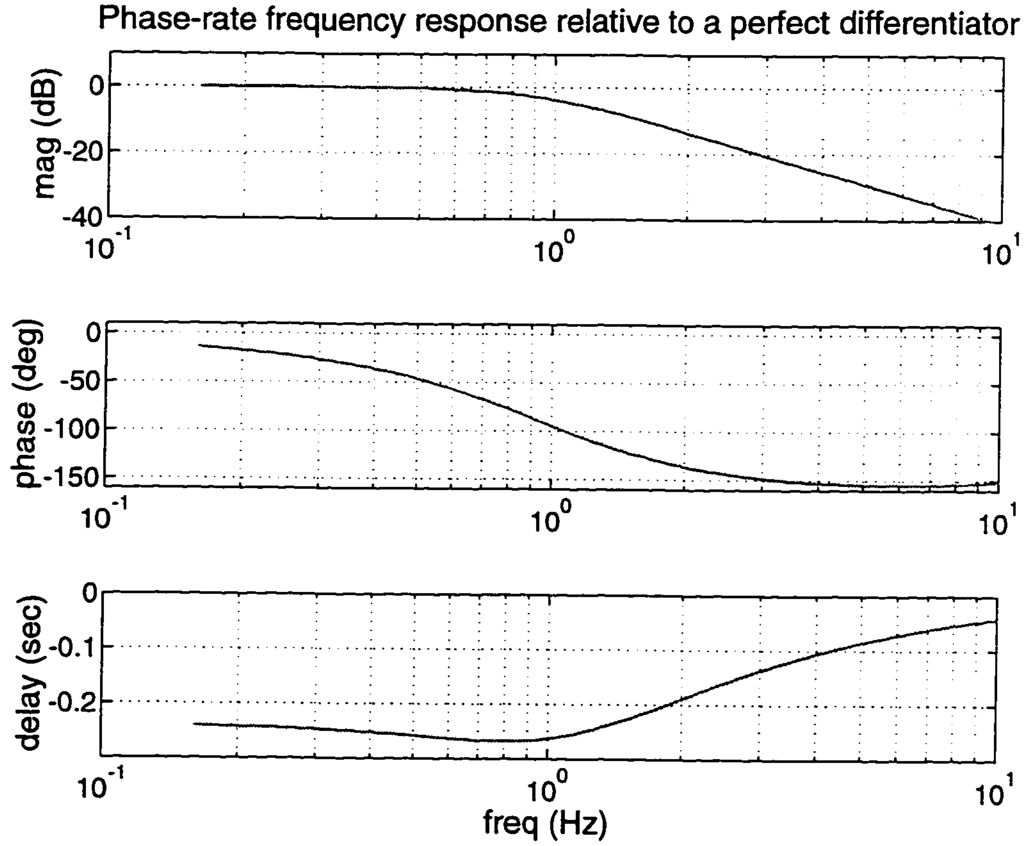


Figure 4.7: **Phase-Rate Frequency Response relative to a Differentiator.** Gain and phase-lag of the 2 Hz phase-rate estimator relative to a perfect differentiator.

matrix, from the I-Q measurement $\Delta\phi_i$ to the posteriori phase state that is given by

$$\begin{bmatrix} H_1(z) \\ H_2(z) \end{bmatrix} \equiv \begin{bmatrix} \Delta\hat{\phi}_i(z)/\Delta\phi_i(z) \\ \Delta\hat{\dot{\phi}}_i(z)/\Delta\phi_i(z) \end{bmatrix} = (I - KH_z)\left[zI - (\Phi - \Phi K H_z)\right]^{-1}\Phi K + K \tag{4.114}$$

where $H_z = [1 \;\; 0]$. Based on a sample period, T, of 12 msec and assuming white process noise, the estimator gain $K = [0.09 \;\; 0.38]^T$ was chosen using MATLAB's *dlqe* function to provide estimator bandwidth of 2 Hz (based on -3 dB crossover). The 2 Hz estimator was used for all flight testing. The *dlqe* function also returns the steady state posteriori covariance matrix $\hat{P}$, from which $\sqrt{diag(\hat{P})} = [0.9 \;\; 5.4]$ (deg deg/s) given the assumed measurement noise. The estimator frequency response is plotted in

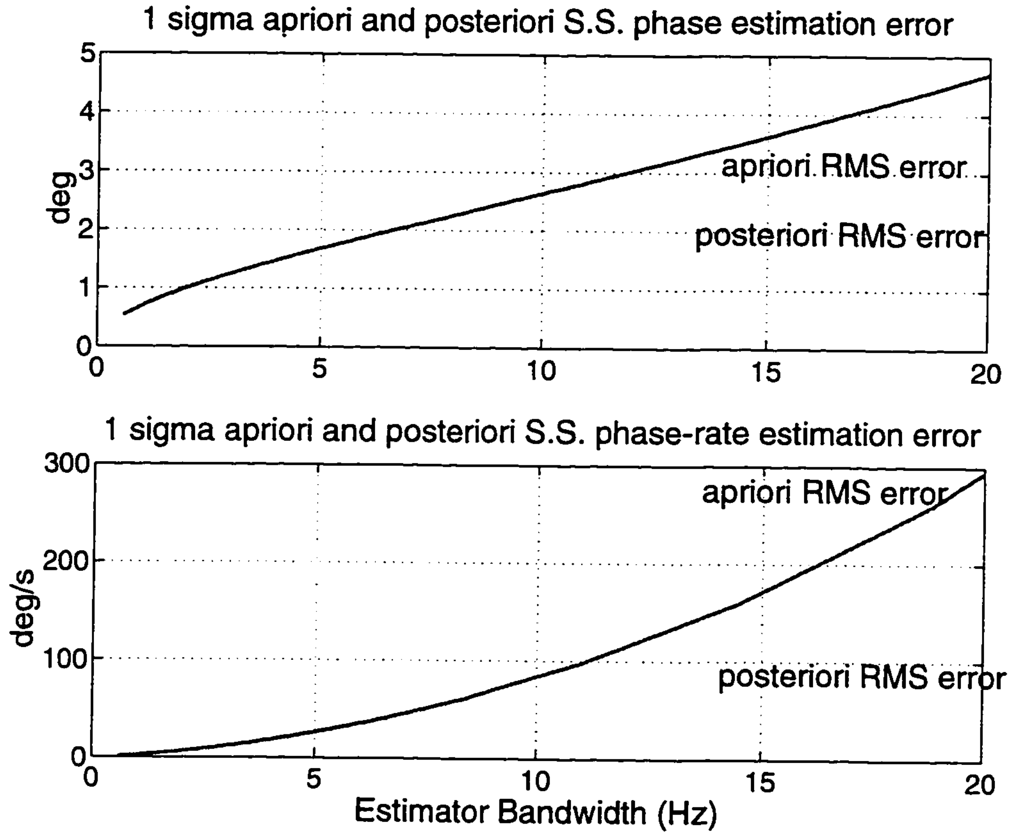


Figure 4.8: **Estimator Steady State RMS Phase and Phase Rate Performance vs. Bandwidth.**

Based on a sample period of 12 msec and RMS phase measurement noise of 3 deg (~ 20 AMU).

figure 4.6. The upper plot also shows the magnitude response of a pure differentiator for comparison. Note that noise attenuation has been achieved only at the cost of significant phase lag. Figure 4.7 shows how the estimator frequency response from I-Q measurement to phase-rate ($H_2(e^{j\omega})$) differs from a perfect differentiator. Since each measurement has been filtered by this transfer function, the resultant GPS determined angular velocity also has this character relative to the true angular velocity.

Based on the assumed process and measurement noise statistics, figure 4.6 and the previous paragraph detail the estimator gains and steady state performance commensurate with a bandwidth specification of 2 Hz. It is interesting to study the

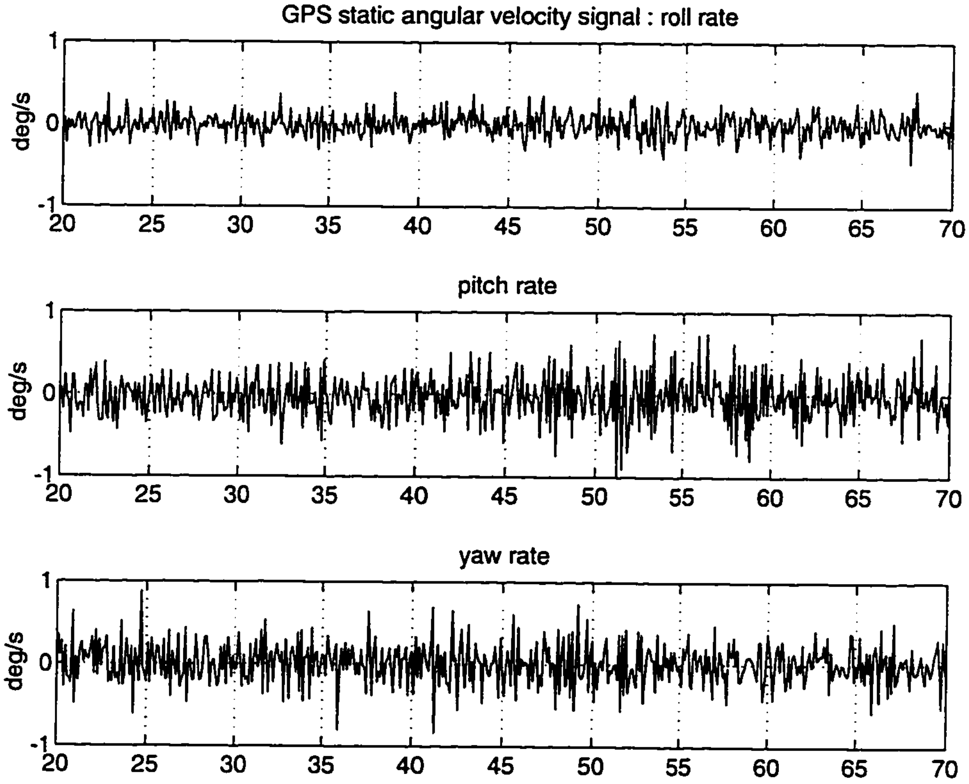


Figure 4.9: **Static measurement of GPS angular velocity.** Unbiased with standard deviations of 0.15 0.25 and 0.25 deg/sec respectively.

variation in steady state performance and phase lag as a function of the bandwidth specification. Writing out the Kalman filter covariance equations for this case, one has six coupled nonlinear equations. Even for such a simple state equation, finding an analytic expression for the positive definite solution is non-trivial. Resorting to numerical evaluation, figure 4.8 shows the result of solving these equations numerous times. The upper plot shows the predicted steady state RMS error in signal phase, $\Delta\hat{\phi}$, as a function of the desired bandwidth. The lower plot shows analogous data for signal phase-rate, $\Delta\hat{\dot{\phi}}$. Because the process noise is not white, the predicted phase-rate covariance is somewhat conservative in practice, however the phase lags are present and noticeable when the angular velocity is compared with onboard rate gyros. With estimator bandwidth set to 2 Hz and using baselines of 10 cycles, the steady state RMS phase-rate error (using a single measurement) translates to RMS angular velocity noise of approximately 0.1 deg/sec. Using a 10 Hz estimator bandwidth, the RMS angular velocity noise increases to approximately 1 deg/sec although the phase lag is greatly reduced. As quantified by the ADOP matrix, the availability of a redundant set of measurements can reduce the noise in the angular velocity estimate to some extent.

Figure 4.9 shows the GPS angular velocity about the aircraft roll, pitch and yaw axes while stationary with the engine off. The measurement is unbiased, with RMS noise of 0.15, 0.25 and 0.25 deg/sec respectively.

Figures 4.10 and 4.11 show a comparison of GPS angular velocity about the yaw axis compared with the equivalent measurement from a piezo-electric rate gyro during flight. Figure 4.11 is a more detailed view of figure 4.10 for $60.0 < t < 65.0$ seconds. Gyro measurements were made at 20 Hz and the GPS measurements were made at 10 Hz. The difference between the two measurements is also plotted in each case. One can see that there is phase delay of approximately 100 ms at the dominant frequency of approximately 0.5 Hz. This is less delay than predicted by figure 4.7.

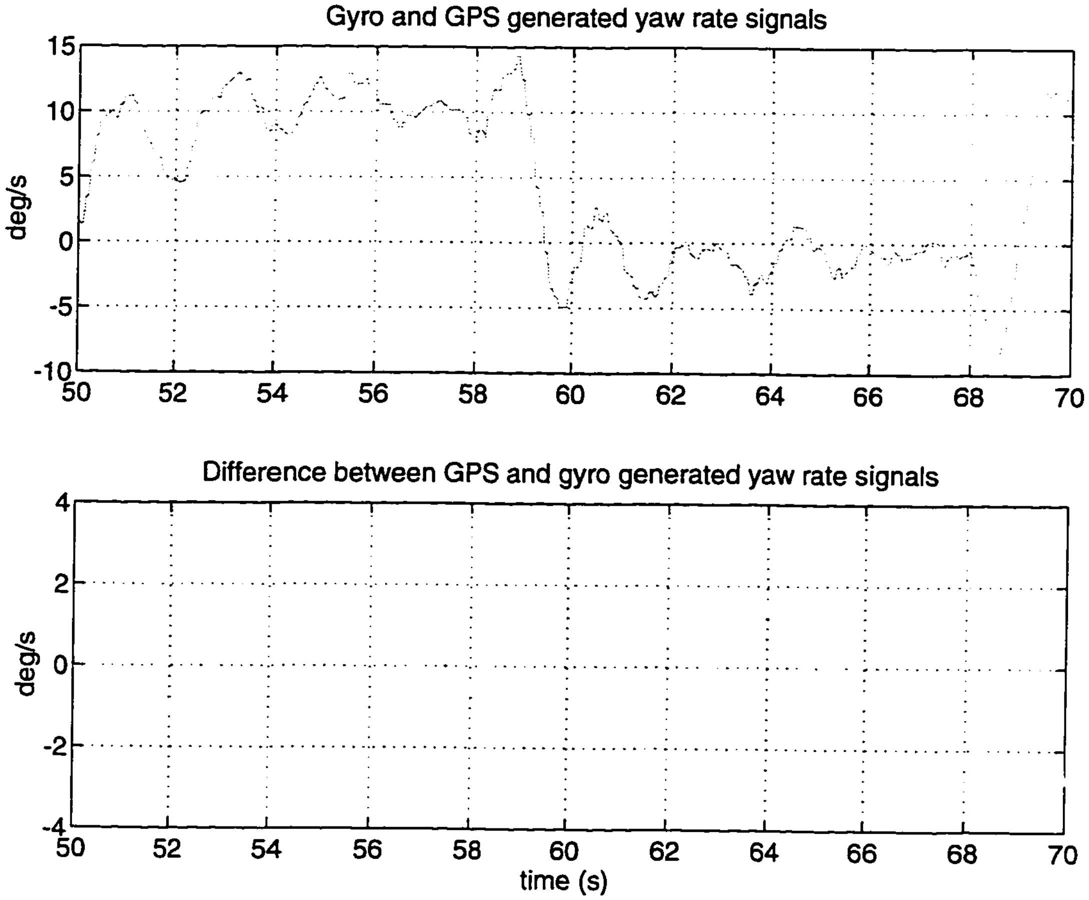


Figure 4.10: **Comparison of GPS and gyro generated yaw rate signal.** The GPS yaw rate estimate (darker line) is delayed by approximately 100 ms relative to the gyro measurement (lighter line).

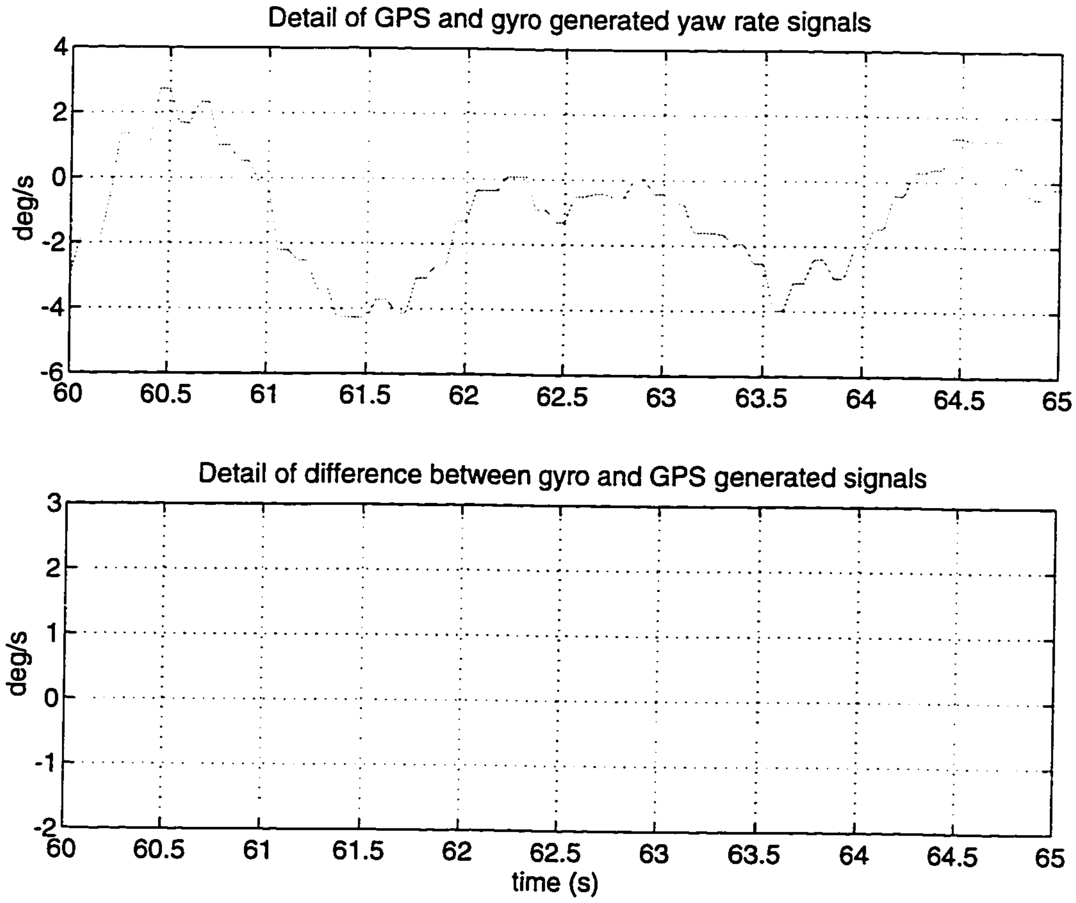


Figure 4.11: **Detail of figure 4.10**

Chapter 5

Automatic Control

Although successive loop closure is perhaps a more traditional technique for aircraft control synthesis, the *full state information* available from GPS enables direct state feedback without requiring additional estimators. For this reason, linear quadratic (LQ) synthesis was chosen for control design. With LQ design, one has a discretized linear plant model given by

$$x_{k+1} = \Phi x_k + \Gamma u_k \tag{5.1}$$

$$y_k = H x_k + J u_k \tag{5.2}$$

where x_k is the discretized state vector at time k, and u_k is the control vector at time k. A controller of the form $u_k = -C x_k$ is chosen to minimize a cost function given by

$$J(C) = \sum_{k=1}^{\infty} (y_k^T Q y_k + u_k^T R u_k) \tag{5.3}$$

The cost function is a weighted sum of squared plant excursions and controls. By varying the ratio of Q to R, one can influence the closed loop dynamics. As R is increased holding Q constant, the cost of using control increases relative to the cost associated with plant excursions. The optimum controller is therefore less willing to "spend" control effort to reduce plant excursions. The task of specifying the system performance is therefore analogous to selecting reasonable weighting matrices Q and R.

The linearized aircraft models (F_{lon}, G_{lon} and F_{lat}, G_{lat}) described in section 2.7.7 were discretized for a sample rate of 10 Hz to give discrete linear models (Φ_{lon} Γ_{lon} and Φ_{lat} Γ_{lat}). MATLAB was used to design command feedforward and control feedback gains for many different controllers. This approach is modular and flexible, allowing a general code to implement different control laws simply by switching to the appropriate set of matrices.

Each control law was tested in simulation using the linearized aircraft model, and also against the more comprehensive (simulink) model described by Bernatets [7]. If the control law proved effective in simulation, the same matrices were uplinked (transmitted to the aircraft over the telemetry channel) for flight validation.

The flight hardware allowed the longitudinal and lateral controls to be separated and debugged in isolation. For example, to test a lateral control law, say heading hold, the RC pilot would maintain control over elevator and throttle but allow the computer to drive the rudder and aileron servos. Data were collected to allow detailed post flight analysis. Through incremental prototyping, all control laws were ultimately validated.

The control laws employed are of the general form

$$u_k = -C\,(x_k - N_x\,r_k) - C_i\,x_{i_k} + N_u\,r_k \tag{5.4}$$

where u_k is control vector output to the servos at time k, C is the state feedback matrix, x_k is the state vector at time k, N_x is the state feedforward matrix, r_k is the command input at time k, C_i is the integral state feedback matrix, x_{i_k} is the integral state vector at time k and N_u is the command feedforward matrix. The elements of the control and state vectors for the linearized longitudinal and lateral models are detailed in section 2.7.7. State feedback is performed on the state elements taken from the full state GPS estimates discussed in chapter 4. It is interesting to note that most of the state elements are velocity or angular velocity components. Velocity perturbation components are generated by using the attitude estimate to express the GPS (ENU) velocity measurement in the aircraft basis. The aircraft CG velocity must also be corrected for angular velocity terms due to the moment arm to the tail mounted position/velocity receiver.

5.1 Control Primitives

Four control primitives are used in both lateral and longitudinal control. These are

- Lateral primitives
  1. Bank angle hold: coordinated turn at a commanded bank angle
  2. Heading hold: regulate to a commanded heading and sideslip velocity
  3. Track hold: regulate to a commanded ground track with wings level
  4. Runway tracking: regulate to runway centerline
- Longitudinal primitives
  1. Airspeed hold: regulate to a commanded airspeed using elevator
  2. Altitude/Airspeed hold: regulate to a commanded altitude and airspeed
  3. Glideslope hold: regulate to commanded glideslope and airspeed
  4. Flare: hold altitude using elevator with throttle idle

5.1.1 Lateral Control Primitives

All lateral control objectives were achieved using the rudder and ailerons, with the exception of ground steering, which used nosewheel steering.

Bank angle

The bank angle controller uses command feedforward with no integral states. The matrices N_x and N_u are the lateral steady state map of the aircraft for commands of bank angle and sideslip velocity respectively. Solving for the lateral steady state, N_x and N_u are given by

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} I_{4\times4} - \Phi_{lat} & \Gamma_{lat} \\ H_{lat} & 0_{2\times2} \end{bmatrix}^{-1} \begin{bmatrix} 0_{4\times2} \\ I_{2\times2} \end{bmatrix} \tag{5.5}$$

$$\text{where} \qquad H_{lat} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Defining a coordinated turn to mean zero sideslip ($v = 0$), the sideslip command is always set to zero. Feedback gain C is calculated using MATLAB's *dlqr* command, weighting errors on sideslip velocity and bank angle. Using this control law, by commanding a bank angle of $\phi_{cmd} = \pm 25$ deg, the aircraft will bank to this angle and regulate to zero sideslip. This causes the aircraft to circle slowly to the left or right, with a yawrate given by the steady state kinematic relationship

$$r_{ss} = \frac{g \tan \phi_{cmd}}{U_0} \tag{5.6}$$

where r_{ss} is the steady state yaw rate, g is gravitational acceleration and U_0 is the airspeed. This gives a yawrate of approximately 14 deg/s at the nominal airspeed of 19 m/s.

Heading

For heading control, the lateral state is augmented with a heading state ψ_{err}, the heading perturbation from nominal. We have

$$\psi_{err} = \psi_{gps} - \psi_{cmd} \tag{5.7}$$

where ψ_{gps} is determined from GPS attitude and ψ_{cmd} is the commanded heading in degrees from true North. The lateral state becomes $x_{lat} = [v \quad r \quad p \quad \phi \quad \psi_{err}]^T$ and F_{lat} is augmented with $\dot{\psi}_{err} = r$. Command feedforward is similar to that described for bank angle, however the N_x and N_u matrices are now formed for command inputs of heading and sideslip velocity. Typically, the controller regulates to a given heading, ψ_{cmd}, with the potential to command a non-zero sideslip velocity. This can be used to create a forward slip, as might be required on a crosswind landing. Feedback matrix, C, is calculated using MATLAB's *dlqe* function with a weighting on heading error and bank angle error.

Track

For crosstrack control, or localizer tracking, the lateral state vector is further augmented with a crosstrack state y_{err}, the tracking error in meters. The lateral state becomes $x_{lat} = [v \quad r \quad p \quad \phi \quad \psi_{err} \quad y_{err}]^T$ and F_{lat} is augmented with

$$\dot{y}_{err} = v + \frac{\pi U_0}{180} \psi_{err} \tag{5.8}$$

where it is assumed that ψ_{cmd} is aligned with the desired localizer heading. An integral state is introduced such that

$$x_{i_{k+1}} = x_{i_k} + y_{err_k} \tag{5.9}$$

where y_{err_k} is calculated from GPS position relative to the localizer at each time step. Command feedforward is not used and one regulates the integral tracking error to zero. This achieves the desired ground track in the presence of disturbances. Feedback control matrices C and C_i are generated using MATLAB's *dlqr* command. weighting the integral tracking error, bank angle and sideslip velocity. In the presence of crosswind the integral state is necessary to eliminate steady state tracking error. With its inclusion, at steady state, the aircraft is crabbed into the crosswind with zero bank angle and sideslip.

Ground Track

Figure 5.1 illustrates the steering geometry for the tricycle undercarriage used on the aircraft. The nosewheel steering angle in radians is denoted by $\delta\theta^1$, and the wheelbase in meters by b. Crosstrack and heading errors (y_{err} and ψ_{err}) in meters and radians are shown on the diagram. Note that the antenna sensing position is located at the tail of the aircraft and a moment arm correction must therefore be applied. The forward speed is given by V in m/s. The ground track controller has a state model based on a non-slip assumption given by

$$\begin{bmatrix} \dot{\psi}_{err} \\ \dot{y}_{err} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ V & 0 \end{bmatrix} \begin{bmatrix} \psi_{err} \\ y_{err} \end{bmatrix} + \begin{bmatrix} V/b \\ 0 \end{bmatrix} \delta\theta \tag{5.10}$$

A control law of the form $\delta\theta_k = -C\,[\psi_{err_k} \quad y_{err_k}]^T$ is used. It may be shown that choosing the control law as

$$C = \begin{bmatrix} \frac{2b}{V} & \frac{2b}{V^2} \end{bmatrix} \tag{5.11}$$

[1] Not to be confused with a perturbation in θ, which is used to denote pitch angle.

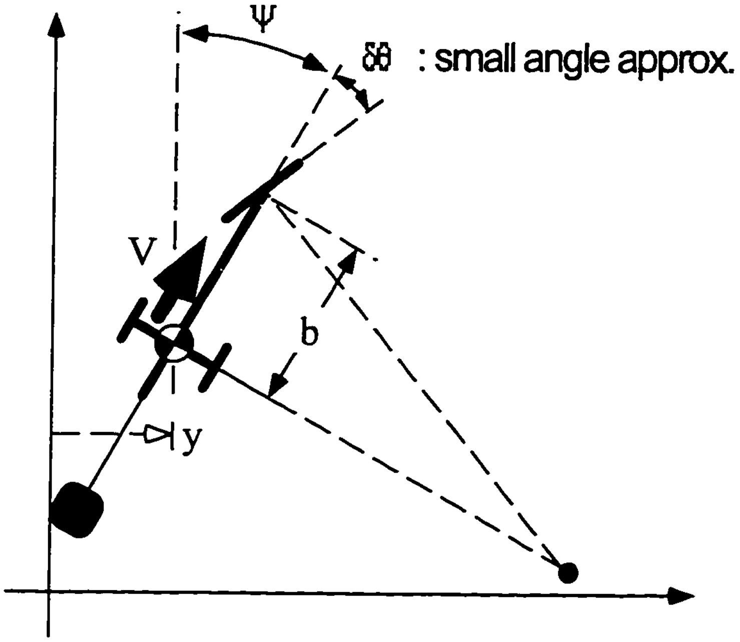


Figure 5.1: **Steering Geometry**

Runway model showing ψ_{err}, y_{err} and $\delta\theta$.

will place the system poles at $-1 \pm j$ in the s plane. This control law effectively linearizes the state equation at each forward speed. One must also be careful to limit $\delta\theta$ as $V \rightarrow 0$, to prevent large control output at low speed. The logic employed simply sets V to 2.5 if $V < 2.5$ m/sec.

5.1.2 Longitudinal Control Primitives

All longitudinal control objectives were achieved using elevator and throttle only. Flaps, although present were not used at all.

Climb and Descent

Due to the difficulty of modeling and regulating the engine at off-nominal conditions, it was easiest to simply set the engine speed to either full power or idle. Airspeed was then regulated using elevator to achieve climb or descent. Maximum climb rate was achieved by regulating airspeed to approximately 16 m/s at full power.

To accommodate engine thrust variations, state feedback is augmented with feedback on the integral of airspeed error. The airspeed command, u_{cmd}, is introduced into the integral time update equation as

$$x_{i_{k+1}} = x_{i_k} + (u_k - u_{cmd_k}) \tag{5.12}$$

where u_k (the first element in the longitudinal state vector) is the airspeed perturbation from nominal in m/s. The matrices C and C_i were generated using MATLAB's *dlqr* function, weighting only the integral airspeed error and elevator servo commands.

Altitude/Airspeed

This control primitive is used to regulate altitude and airspeed for level flight at a commanded altitude and airspeed. The longitudinal state vector is augmented with h_{err}, the altitude error in meters. The longitudinal state becomes $x_{lon} = [u \quad w \quad q \quad \theta \quad h_{err}]^T$ and F_{lon} is augmented with

$$\dot{h}_{err} = -w + \frac{\pi U_0}{180}\theta \tag{5.13}$$

Note that the pitot measurement of airspeed was used for u and not a GPS measurement of velocity. Integral states on both airspeed and altitude are used to account for variations in engine performance and to ensure zero steady state error. The integral on airspeed error is given by equation (5.12). The integral altitude error is updated using

$$x_{i_{k+1}} = x_{i_k} + (z_{gps_k} - z_{cmd_k}) \tag{5.14}$$

where z_{gps_k} is GPS determined altitude at time k and z_{cmd_k} is the commanded altitude. The matrices C and C_i are formed using *dlqe*, weighting both the integral airspeed and integral altitude errors.

Glideslope

The glideslope controller is identical to the altitude/airspeed controller. The only difference is that the altitude command z_{cmd_k} in equation (5.14) is replaced at each time step with a value that is a function of the approach geometry and $\dot{h}_{err}$ is is modified by the steady state pitch angle associated with the glideslope. Typically the airspeed command is also reduced as one descends the glideslope on final approach.

Flare

The flare autopilot is used just prior to landing for transition between glideslope regulation and touchdown. The control law uses an integral on altitude error to drive the elevator only. The throttle is set to idle and there is no attempt to regulate airspeed. Attempting to regulate altitude with minimum throttle cannot work, and the integral state slowly winds up, saturating the elevator while slowing and pitching the aircraft nose up. Ultimately this control law will stall the aircraft. This is the intention with landing flare. One wishes to stall the aircraft just as the wheels touch the ground, having zeroed the vertical descent rate and minimized the airspeed. The control uses an integral on altitude error as described by equation (5.14). The matrices C and C_i are formed using *dlqe*, weighting only the integral altitude error. Typically the commanded altitude is set to approximately 0.4 m above the ground. This allows the integral to wind up before touchdown occurs.

5.2 Switching Logic

To enable commands at a higher level of abstraction, one needs switching logic to tie together the control primitives detailed above. With this switching logic in place, it becomes possible to issue a command such as "go to location (x,y,z)" or to specify an entire flight path in terms of waypoints. To achieve this capability, both the lateral and longitudinal control laws are organized as a hierarchy.

Lateral Hierarchy

The lateral hierarchy has the bank angle primitive at its root. Moving up the hierarchy, first heading control and then tracking control primitives are enabled. If one gives a command to fly to location (x,y,z), the bank angle command is invoked at the lowest level in the hierarchy to turn toward the heading required to fly to the commanded location. When within a tolerance band around the correct heading, the heading hold controller is invoked. This is used for small heading corrections to bring the aircraft onto the ground track (localizer) which will intersect the desired location. Finally, when within a tolerance band around the desired ground track, the highest level in the hierarchy is attained and the track primitive is invoked.

Longitudinal Hierarchy

The longitudinal hierarchy is simpler than the lateral, having just two levels. The lowest level is a climb/descent primitive, in which airspeed hold is invoked with either maximum or idle throttle. This is used to climb or descend to the commanded altitude. One level higher is the altitude/airspeed hold autopilot, which regulates to the commanded altitude and airspeed when within a few meters of the commanded altitude.

Hysteresis

At each time step, logic is checked to see if the vehicle is within the required tolerance band for the active control law. If it is not, a new primitive is initialized and enabled. A potential problem with this architecture is the possibility for limit cycles, where the system oscillates between two levels in a hierarchy. This can happen if in changing to a higher level in the hierarchy, a transient pushes the vehicle once again outside of the required tolerance envelope. To prevent this occurrence some hysteresis in the tolerance bands was employed. With hysteresis, once a higher level primitive has been enabled, a large excursion is necessary to force a lower level mode to depose the active control law.

Integral states and mode switching

In changing between control modes, care must be taken to initialize any integral states appropriately. One possibility is to initialize each integral state to zero, however this has the potential to excite large transients. To eliminate this problem, integral states were initialized to values chosen to ensure continuity of the control outputs across the mode switch. Consider the transition from airspeed regulation during ascent to altitude and airspeed regulation once the commanded altitude has been reached. We wish to ensure that $u_{k+1} \equiv u_k$ over a transition where the control matrices change from C_k and C_{i_k} to C_{k+1} and $C_{i_{k+1}}$. From equation (5.4) it follows that the new integral state must be initialized according to:

$$x_{i_{k+1}} = C_i^{-1} \left[u_k + C_{k+1}\, x_{k+1} - (C_{k+1} N_{x_{k+1}} + N_{u_{k+1}})\, r_{k+1} \right] \tag{5.15}$$

With this initialization, the controls move smoothly across mode transitions and large transients are avoided. This is particularly important on the mode change from glideslope hold to flare as this transition occurs close to the ground.

5.3 Wind Estimation

In addition to GPS measurements of velocity and attitude, additional sensors are available to measure airspeed and the air incidence angles α and β. The combination of pitot, α and β allow one to measure the velocity of the aircraft relative to the air in the aircraft basis. GPS gives the aircraft velocity relative to the ground in the ENU basis. The difference, ignoring noise, is due to the ambient, *three dimensional wind*

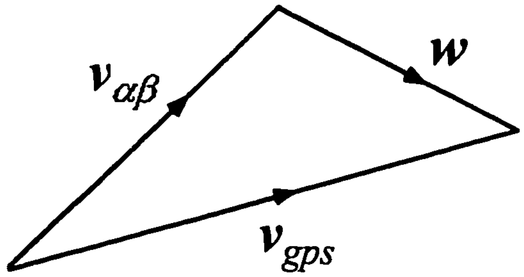


Figure 5.2: **Wind Triangle**
GPS measured ground velocity + wind velocity = aircraft velocity relative to the air (plus noise).

at the aircraft location[2]. In vector form and as illustrated in figure 5.2, one can write

$$\boldsymbol{v}_{gps} = \boldsymbol{w} + \boldsymbol{v}_{\alpha\beta} \tag{5.16}$$

where $\boldsymbol{v}_{gps}$ is the GPS determined aircraft velocity, $\boldsymbol{v}_{\alpha\beta}$ is the aircraft velocity relative to the air measured using the pitot, α and β sensors and $\boldsymbol{w}$ is the wind velocity at the aircraft location. Using the known attitude, $\mathcal{R}$, to express the aircraft relative airspeed in the ENU basis, to first order one has the wind vector equation

$$\boldsymbol{w}_{enu} = \boldsymbol{v}_{gps} - \mathcal{R}^T \begin{bmatrix} 1 \\ \beta \\ \alpha \end{bmatrix} V_{pitot} \tag{5.17}$$

where V_{pitot} is the airspeed measured at the pitot. In practice one must also correct for the fact that the alpha vane is perturbed by the local flow field in the vicinity of the aircraft. A first order correction can be calculated based on elementary lifting line theory as

$$\Delta\alpha = \frac{L}{2\pi\rho U_0^2\, b\, l} \tag{5.18}$$

where L is the lift, b is the wingspan and l is the distance from the quarter chord to the α vane. Another correction can be applied proportional to the pitch component of angular velocity multiplied by the moment arm from CG to the alpha vane. With these corrections, one has a *noisy* but unbiased estimate of the ambient wind at 10 Hz. Noise contributions come from each element on the right hand side of equation (5.17). In designing the α and β vanes it is important to ensure that their dynamics are fast relative to the aircraft dynamics and that nonlinear effects such as wiper friction in mechanical potentiometers is minimized.

Wind magnitude and direction are subsequently filtered by a first order filter with a time constant of 1 second to achieve some noise reduction. The filter time constant was chosen based on the expected frequency content of the wind variation at the nominal airspeed. No serious attempt was made to quantify or otherwise characterize

[2] This form of wind estimation is frequently used in commercial aviation using *inertial* velocity and attitude measurements. In this case GPS alone was used to derive the velocity and attitude data.

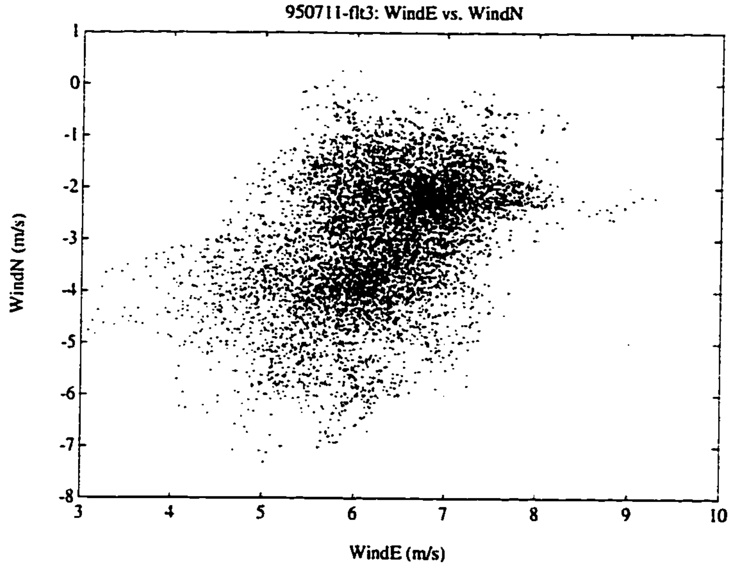


Figure 5.3: **East-North Wind Scatter Plot**
Filtered wind scatter plot over 5 minutes of flight. Wind strength is ~7 m/s. Samples at 10 Hz (3000 samples).

the nature of the wind estimate. Conservatively, it is reasonable to expect RMS noise of approximately 0.15 m/s prior to filtering. [3]

Figure 5.3 shows East and North components of the wind from 5 minutes of flight at a constant altitude. Figure 5.4 shows the East and North components of wind in altitude bins of 20 meters over a typical flight. The latter data shows significant increase in wind strength with altitude and a significant wind shift at approximately 200 m.

Instrumented in this manner, an aircraft becomes a sensor that can detect the wind profile of the atmosphere through which it flies. This can be of value to air traffic controllers, and has potential application as a sensor in a downburst detection algorithm. Wingrove [61] provides an example of this type of wind estimation using attitude and velocity measurements from the onboard INU. In this paper, Wingrove presents wind profiles that were experienced by commercial aircraft when

[3] The variation shown in figure 5.3 is largely attributable to wind variations over the data collection period. This is not noise, but it the true signal is obscured by noise.

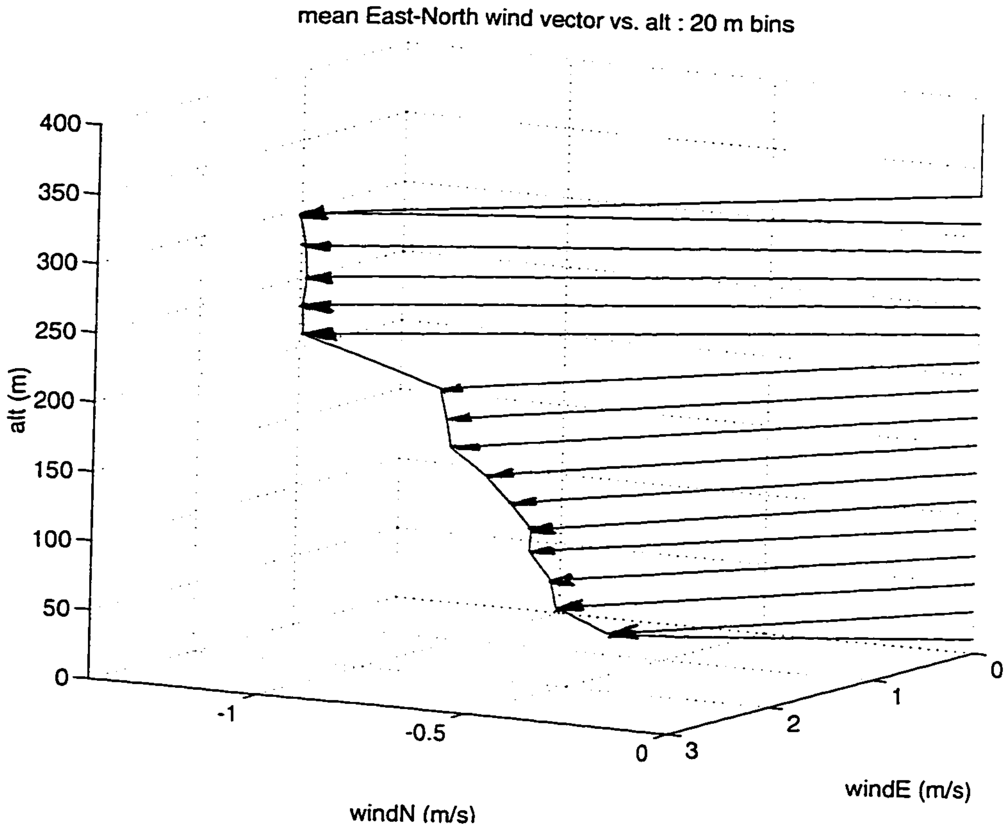


Figure 5.4: **Wind Profile with Altitude**

Wind direction and strength in 20 meter altitude bins. Wind strength increases with altitude and direction change is evident at approximately 200 meters.

flying through downburst activity[4]. The fidelity and sample rate is also of interest to people attempting meteorological soundings. At 10 Hz, even relatively small scale turbulence (with characteristic length of $\sim 2U_0/10$) can be observed.

The wind estimate described here was not used in the inner loop automatic control primitives, however it was used by higher level logic to determine a first order correction to turn commencement when flying a predetermined trajectory in the presence of wind.

[4] In one case the aircraft crashed. The data was recovered from the flight recorder.

5.4 Finite State Machine

To automate the flight of an aircraft from takeoff to landing, it is useful to conceive of a number of logical *states* through which the system passes. Each state becomes active for some duration over the course of a flight, as illustrated in figure 5.5. The state machine proceeds sequentially from "holding", on the runway prior to takeoff, through "flight plan", where the waypoint sequence is executed and ultimately to "roll out", where the aircraft is decelerated on the runway after landing. This concept is implemented in operation as a very simple finite state machine which provides the *glue logic* uniting the various control primitives with the pre-specified trajectory.

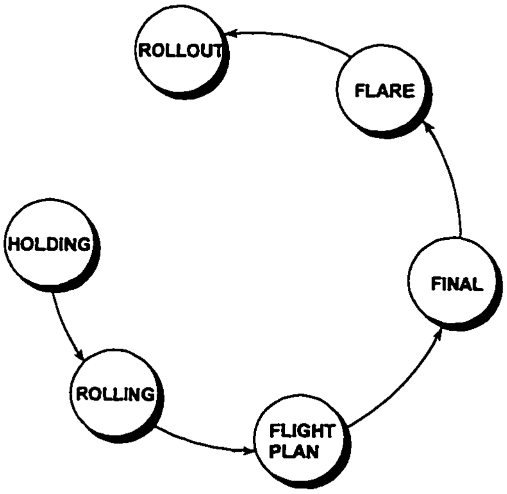


Figure 5.5: **Auto Takeoff to Landing State Machine**

The current section describes the waypoint architecture and each phase of the state machine in detail.

5.4.1 Waypoint Architecture

Waypoints are cartesian points (x,y,z), specified in the runway relative basis as shown in figure 5.6. A sequence of waypoints was used to define a runway relative flight path consisting of three rectangular circuits aligned with the runway on one edge. Waypoint 0 is the takeoff initialization and glideslope-aim-point. Given n waypoints, the vector from waypoint n to waypoint 0 is the specification of the glidepath and localizer on final approach. This closed-path architecture allows for touch and go if desired.

. Given the runway azimuth, (which can vary with airfield and wind direction) each waypoint is rotated into the local ENU basis. Associated with each waypoint is a quantity, $turninit_k$, and the unit vector $\hat{\boldsymbol{h}}_k$, aligned with the ground track from the previous waypoint. $Turninit_k$ is that distance prior to waypoint k where the aircraft should commence a turn towards waypoint $k+1$ in the absence of wind. The *turninit* distance is calculated for a given bank angle command to ensure that the turn will terminate with the aircraft aligned with the ground track and heading for the subsequent waypoint. In making the turn toward each successive waypoint, the lateral hierarchy is exercised as described in section 5.2. As illustrated in figure 5.6, it is with respect to the ground track vector $\hat{\boldsymbol{h}}_k$, that the lateral state elements ψ_{err} and y_{err} are defined.

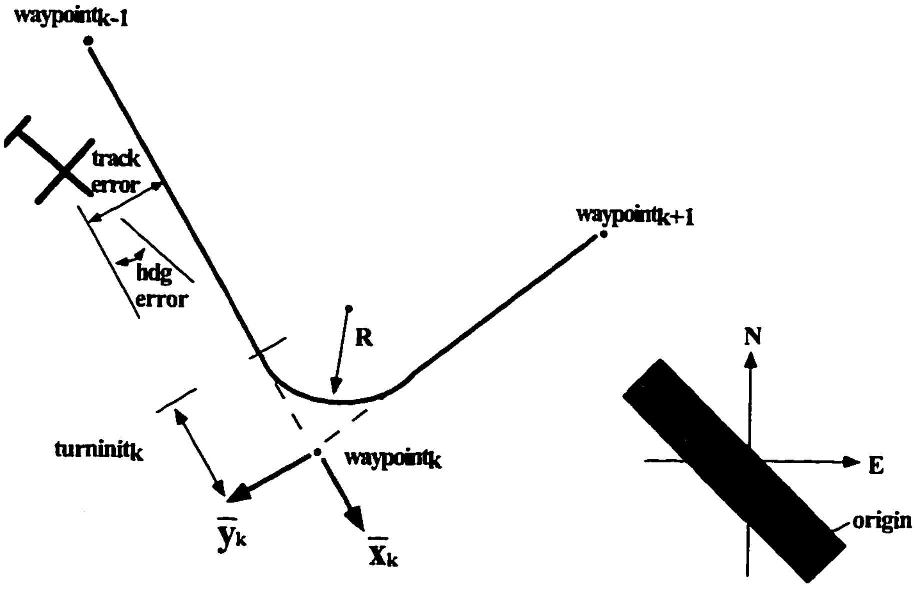


Figure 5.6: **Waypoints Architecture**
Successive waypoints showing $turninit_k$, heading error ψ_{err} and crosstrack error y_{err} for leg k. The runway East and North axes plus origin point are also shown.

Effect of Wind

Since there is always *some* wind, its effect must be anticipated when turning toward a new waypoint. The time taken for each turn is estimated based on the angle through which the aircraft must turn and the expected yaw rate (as given by equation (5.6)). The quantity $turninit_k$ is calculated prior to flight assuming zero wind, based on a turn radius, R, consistent with the commanded bank angle.

As shown in figure 5.7, with ambient wind $\boldsymbol{W}$, commencing a turn at the *turninit* point will result in a tracking error on the subsequent leg equal to the time taken for the turn multiplied by the wind velocity perpendicular to $\hat{\boldsymbol{h}}_{k+1}$. This is an "open loop" prediction of the cross track error on the subsequent leg based on the current wind estimate. This error is canceled to first order by either anticipating or delaying turn commencement relative to *turninit*.

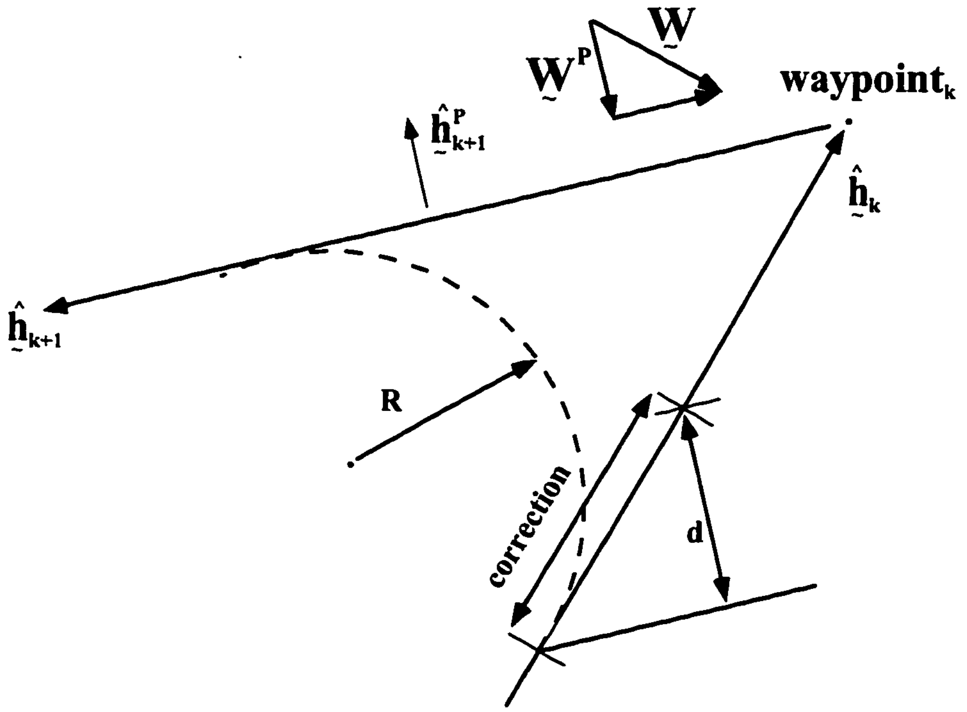

$\underset{\sim}{\mathbf{W}}$
$\underset{\sim}{\mathbf{W}}^P$
waypoint$_k$
$\hat{\underset{\sim}{\mathbf{h}}}^P_{k+1}$
$\hat{\underset{\sim}{\mathbf{h}}}_k$
$\hat{\underset{\sim}{\mathbf{h}}}_{k+1}$
R
correction
d

Figure 5.7: **Wind Correction to Turn Commencement**

Delayed turn commencement due to crosswind on subsequent leg.

Figure 5.7 shows the geometry for a turn at waypoint k. Each waypoint is associated with a unit vector $\hat{\boldsymbol{h}}_k$ aligned with the ground track and its perpendicular $\hat{\boldsymbol{h}}_k^P$. The wind vector is resolved into crosswind, $\boldsymbol{W}^P$, and headwind components for the subsequent leg. The nominal *turninit* point and turn radius R are shown, together with a correction to account for the anticipated cross track error d. In this case the waypoint logic must delay turn commencement by the amount shown by "correction", to account for the crosswind on the subsequent leg. For a 90 degree turn to a leg with a crosswind component of 5 m/s, the correction amounts to approximately 32 m. Accounting for the correction usually places the aircraft within 6 meters of the correct ground track, close enough to enable the tracking primitive immediately upon turn completion. In the case where the aircraft is off track by more than the tracking primitive tolerance (6 m), the heading primitive is enabled. A heading command is chosen, accounting for the crosswind component to effect convergence with the desired ground track at an angle of 10 degrees.

5.4.2 Automatic Takeoff to Landing

Auto Takeoff

After initialization of the position and attitude cycle ambiguities, transition from the "holding" to the 'rolling" state occurs when the human (RC) pilot enables computer control. In the "rolling" state, full throttle is applied and runway heading is maintained with nosewheel/rudder steering. Elevator is set to neutral and ailerons are used in direct feedback on roll angle. The aircraft accelerates down the runway and airspeed increases.

When the measured airspeed is greater than 13.5 m/s, a transition from the "rolling" to the "flight plan" state occurs. The airspeed integral state is initialized to give an elevator pulse of approximately +10 degrees. At 13.5 m/s the elevator pulse is sufficient to lift the nose of the aircraft, generating lift and causing a clean transition from runway roll to flight.

Flight Plan

In the "flight plan" state, the waypoint tracking mode is engaged. Control primitives are enabled as necessary to fly each leg to successive waypoints. Waypoint 1 is specified to be upwind and inline with the airfield at an altitude of 150 meters. At the instant the elevator pulse occurs, the "flight plan" state is enabled and the longitudinal control enters airspeed regulation at full throttle, in order to climb to the specified altitude. The takeoff airspeed is low relative to nominal climb airspeed so the integral state quickly winds the elevator down to a setting appropriate for climbout. With the correct initialization, this results in a clean transition to flight and steady increase in altitude. Incorrect initialization results in undesirable phugoid excitation. The magnitude of the initial elevator pulse was determined by emulation of manual takeoffs and by simulation using the aircraft dynamic model. Since the aircraft is correctly aligned with the upwind ground track at takeoff, the tracking primitive is immediately enabled for lateral control.

Base and Final

Figure 5.8 shows the geometry for base turn and final approach. Having reached $turninit_{n-1}$, waypoint n becomes active and the base turn is commenced. Once this point is reached, the state machine transitions from "flight plan" to "final". In the "final" state, waypoint tracking remains active for lateral control but the glideslope primitive is engaged for longitudinal control. With glideslope control the altitude and airspeed commands become linear functions (affine to be precise) of the parameter θ [5] shown in figure 5.8, as θ varies from $-\pi/2 \rightarrow \pi/2$. The glideslope angle was chosen to be 6 degrees, requiring a steady state engine setting of $\sim$ 15% full power with zero flaps.

The curving approach geometry was chosen as a precaution against engine failure on final (when close to the ground). This choice increases the probability of successful "dead-stick" recovery relative to the trajectory required for a "straight-in" approach.

[5] In this context the parameter θ is used to track progress through base and final turns. A naming conflict exists with the aircraft pitch angle but the two should not be confused.

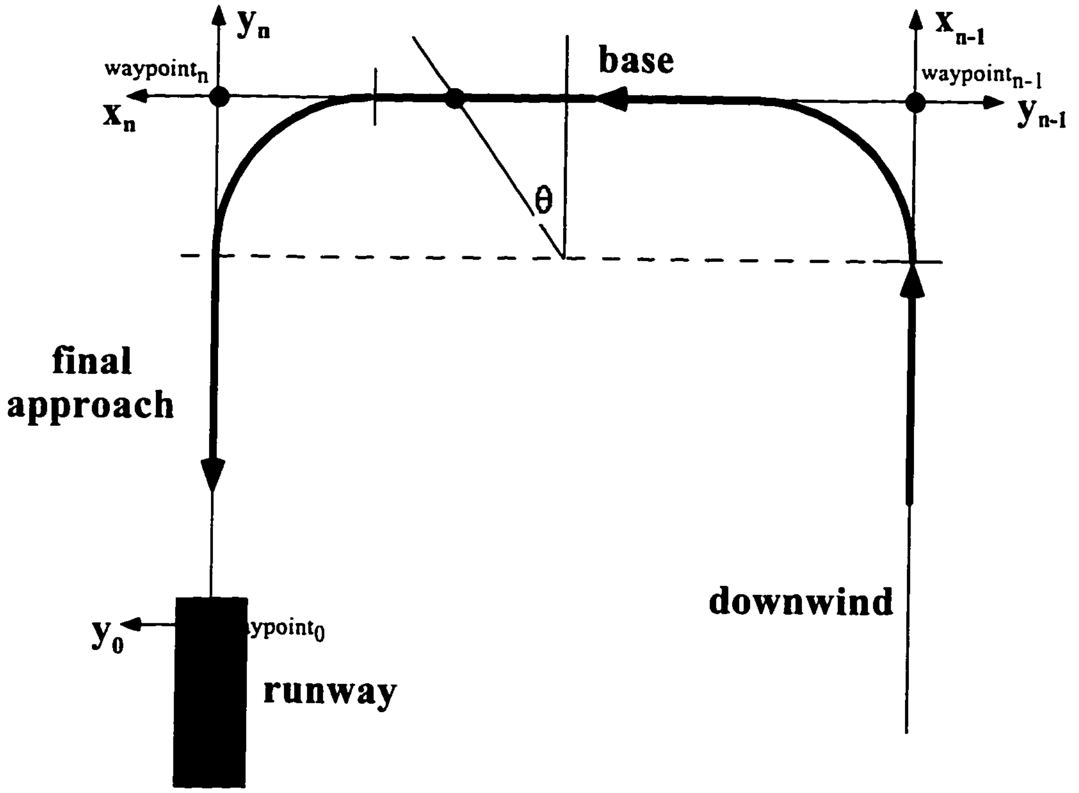


Figure 5.8: **Base and Final Approach**
Plan view of approach geometry, showing base and final relative to the runway.

When the turn to final has been completed, the airspeed command has been reduced to 16 m/s and altitude command now becomes a linear function of the distance to the aim-point.

Flare and Touchdown

A quadratic segment is used to transition between the glideslope and the touchdown point. (An equivalent quadratic segment is also used to transition between the constant altitude downwind leg and the beginning of the glideslope). This quadratic is chosen to be tangent to the glideslope at the *glideslope-intersection-point*, and to have zero gradient at the touchdown point. In practice, the tangent point and the touchdown point occur 50 meters equidistant on either side of the glideslope-aim-point

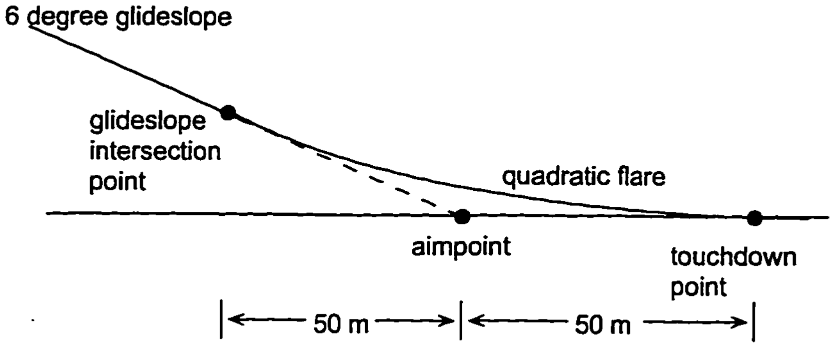


Figure 5.9: **Transition from Glideslope to Touchdown**

Quadratic flare trajectory.

as shown in figure 5.9. When the aircraft passes the glideslope-intersection-point, transition from the "final" state to the "flare" state occurs.

In the "flare" state, the quadratic altitude is used for altitude command to the longitudinal flare primitive and throttle is set to idle. The lateral controller continues to use the tracking primitive throughout. As described earlier, the flare autopilot slowly stalls the aircraft by attempting to hold altitude using elevator with throttle set to idle. Once touchdown has occurred, transition to the "roll out" state occurs, which currently zeroes all controls and holds throttle at idle. It would be possible to re-engage the 'rolling" state at this point to effect a go-around. In current practice, the

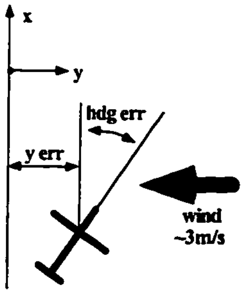


Figure 5.10: **Landing Conventions**

Landing heading error (ψ_{err}), crosstrack error (y_{err}) and crosswind direction for figure 5.15.

RC pilot recovers control when the aircraft is rolling along the runway and applies the brake. This was not automated since application of the brake also involves applying full down elevator.

5.5 Flight Results

This section presents a number of flight results taken from a flight on November 27th 1995. All data comes from the same flight, in which a standard three circuit traffic pattern was flown entirely under automatic control. On this day there was a smooth right crosswind of approximately 3 m/s at groundlevel. Figure 5.11 shows the takeoff

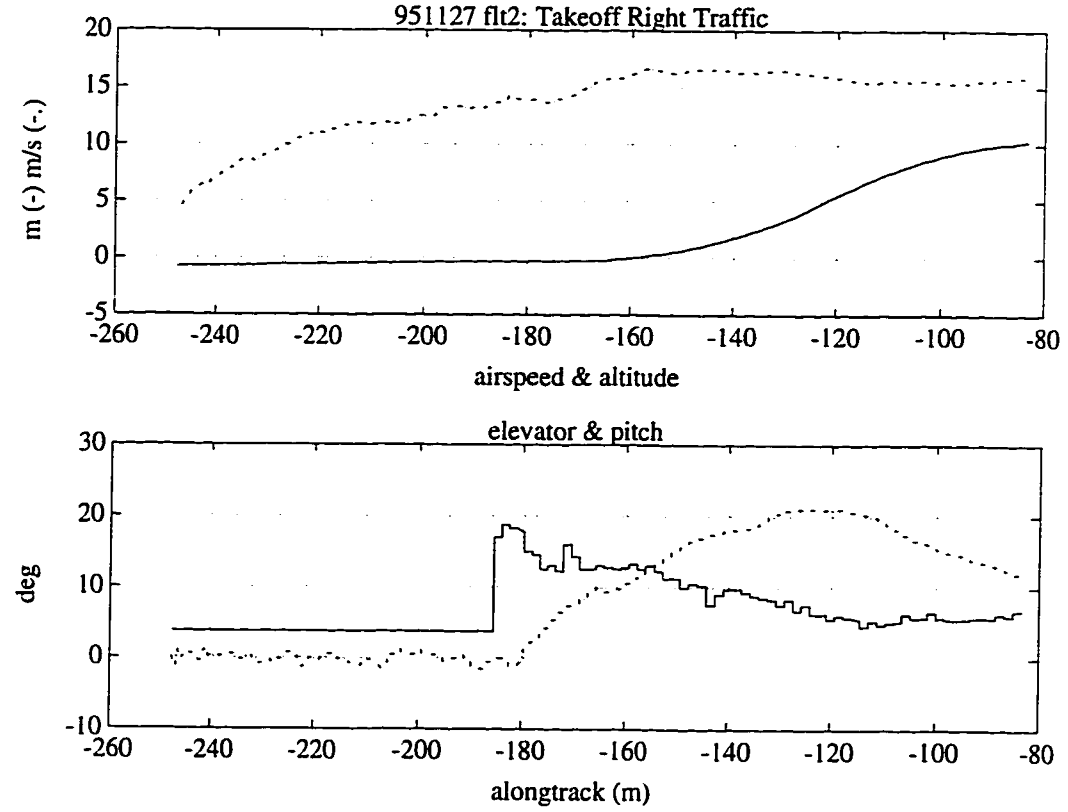


Figure 5.11: **Automatic Takeoff**

Runway roll and and liftoff. The upper plot shows altitude (solid line) and airspeed (dashed line) vs. alongtrack distance. The lower plot shows the control command to the elevator servo (solid line) and the aircraft pitch angle (dashed line). The elevator perturbations are relative to elevator trim of approximately 4 deg.

roll from stationary to a few seconds after lift off versus alongtrack distance. The takeoff roll is approximately 80 meters and is slightly uphill. An elevator pulse of approximately 13 degrees (relative to trim, which was set to 4 deg) lifts the nose and causes a clean transition to flight. The elevator subsequently winds down and airspeed levels out at the commanded value of 16 m/s.

Figure 5.12 shows East-North (plan) and East-Up (elevation) plots over an autonomous flight from takeoff to landing. The runway azimuth was +120 degrees from true north and right traffic was active. Waypoints specifying the desired trajectory are shown as circles. The runway (not shown) is near (0,0) on the upper plot. The larger circuit is 1200 m long and 400 m wide. After takeoff, the aircraft first flies the close pattern while gaining altitude. Commanded altitude of 150 m has been achieved prior to passing over the runway for the first time. This altitude is held throughout the large rectangular pattern. At the uppermost waypoint shown in the upper plot, the altitude command is reduced to $\sim$ 60 m. This altitude is held during another close circuit prior to final approach and landing. Figure 5.13 shows the direction and elevation[6] of each satellite used for position determination over the flight duration. On two occasions a satellite signal was occluded during the flight as show in figure 5.18. On both occasions this was PRN 28 which was setting at the time. The elevation of PRN 28 changed from 18 degrees above the horizon to 14 degrees above the horizon during the flight and was occluded on two occasions when the aircraft banked away from it. This occurred for the first time turning downwind on the first circuit and the second time turning base on the third circuit. Figure 5.14 shows steady state performance along the long downwind leg which extends from coordinates (150,-630) to coordinates (-780,-30) (meters). The upper plot shows airspeed and altitude perturbations from the commanded values over approximately 1100 meters. Once the initial transients have damped out, the RMS errors in altitude and airspeed are 0.21 m and 0.23 m/s respectively. The lower plot shows the cross track error. The error has approximately zero mean in the presence of a steady crosswind from the left. The steady state RMS tracking error had a value of 0.44 m.

[6] The length of the vectors in this figure is of no significance.

Figure 5.15 shows final approach and landing data. The first plot shows the airspeed, the altitude and the altitude-command versus alongtrack distance relative to the glideslope-aim-point. Touchdown occurs at approximately 35 meters past the aim point, with an airspeed of approximately 14 m/s. The second plot shows the command to the elevator servo and the aircraft pitch angle. The aircraft pitch slowly increases throughout the flare maneuver, increasing to approximately 10 deg just prior to touchdown. At touchdown, the pitch is suddenly reduced to zero when the main wheels contact the runway. A photograph just prior to touchdown is shown in figure 5.16. (A similar picture is also shown on the abstract page). One can see the command to the elevator servo is increasing throughout the flare maneuver, reaching its mechanical limit ($\sim$ 30 deg) just as touchdown occurs. The third plot shows crosstrack and heading errors. Figure 5.10 shows the geometry and sign conventions for the heading and cross-track signals plotted. Throughout final, the aircraft is crabbed to the right by approximately 10 deg to counter the steady crosswind component from the right. The crosswind is shown in the fourth subplot in figure 5.15. The cross track error remains less than 1 meter throughout. The final plot also shows the bank angle history in degrees through final and touchdown, which varies by only a few degrees.

Figure 5.17 shows aggregate data from four right traffic automatic landings. These data was taken over two different days, in variable wind conditions. The diagram is included to give some idea of the spread in the flight technical error.

Figure 5.18 shows the number of satellites seen by both the air and reference receivers, the vertical component of dilution of position squared (VDOP^2) and the position residual over the entire flight. The vertical component of DOP is seen to be increasing over the flight, with jumps when PRN 28 is occluded. The lower plot shows the position residual over the flight in units of wavelengths squared. This plot is somewhat distorted due to quantization of the residual measurement to only two decimal places.

To summarize, seven successful automatic flights, from takeoff to landing were performed on two successive days. These flights were done using GPS and air data sensors only. No inertial instruments were employed. In each case[7], the aircraft landed within one meter of runway centerline after performing a controlled flare prior to touchdown. The alongtrack dispersion was not measured. Weather conditions varied over the two days of flight testing, requiring both runway directions to be used and demonstrating operation in cross winds at the ground of up to approximately 3 m/s.

[7] There were eight flight attempts in total. The fourth attempt on the first day was aborted by ground control when the number of satellites simultaneously in track at both the aircraft receiver and the reference receiver dropped below four during flight. At that time there were only three satellite above a 30 degree elevation mask.

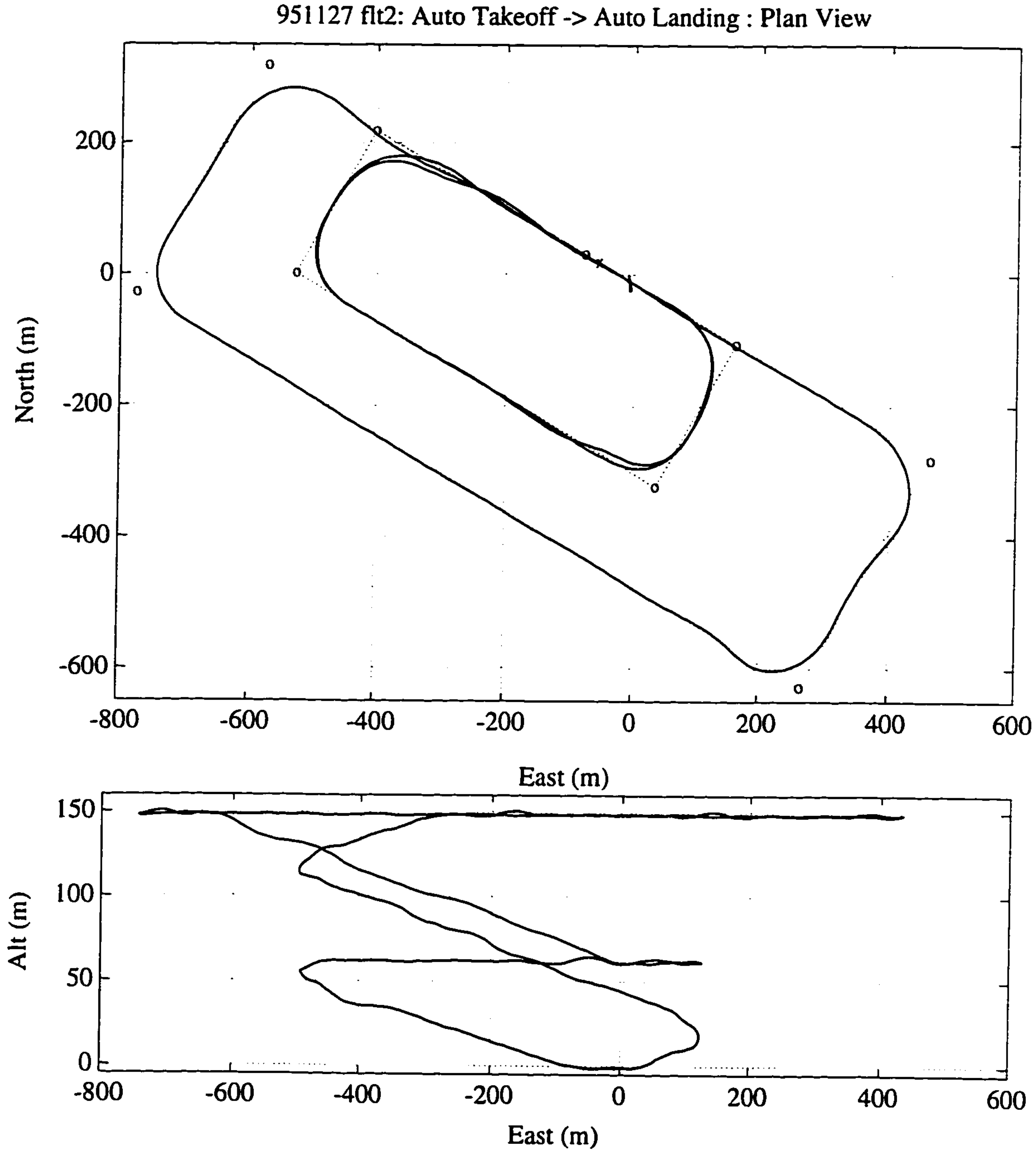


Figure 5.12: **Plan and Elevation Views**

Autonomous flight from takeoff to landing. Right traffic is active. Airfield (not shown) is near (0,0) on the plot.

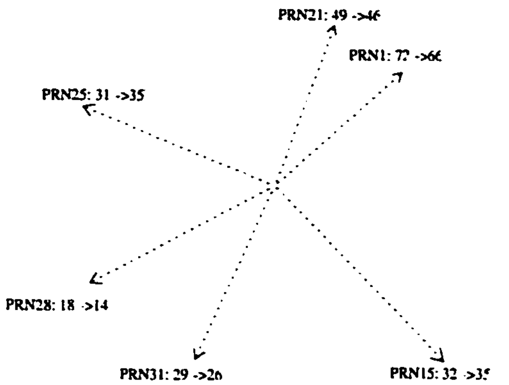


Figure 5.13: **PRN's Elevation and Azimuth**

Direction and elevation of each satellite over the flight duration. North is up and East is to the right. Elevation at the beginning and end of the flight is noted beside each direction vector.

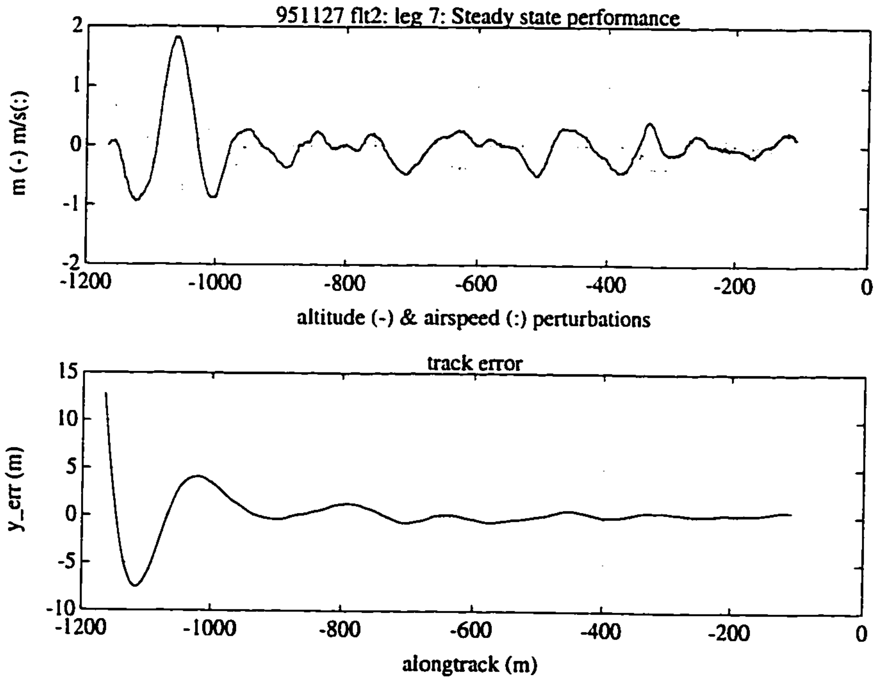


Figure 5.14: **Steady State Performance**

Detail of figure 5.12 showing flight technical error on the long downwind leg. The upper plot shows airspeed and altitude perturbations from nominal. The lower plot shows crosstrack error.

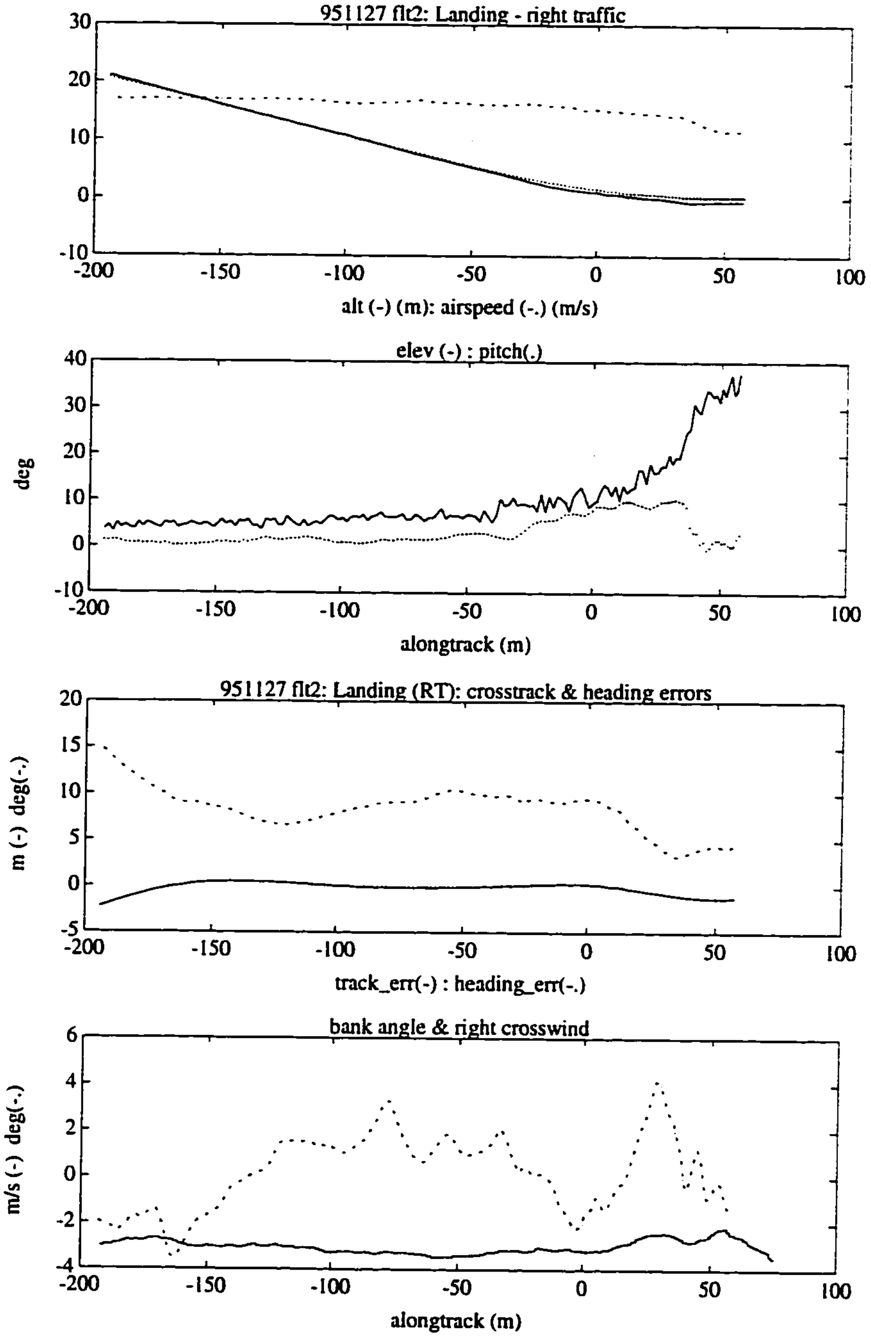


Figure 5.15: **Automatic Landing**

Final approach and touchdown vs. distance alongtrack relative to the glideslope-aim-point. The first plot shows airspeed altitude and the quadratic altitude command. The second plot shows the command to the elevator servo and pitch angle. The third plot shows heading and cross track errors. The final plot shows bank angle and crosswind component of wind. A schematic is shown in figure 5.10 showing the interpretation of these quantities.

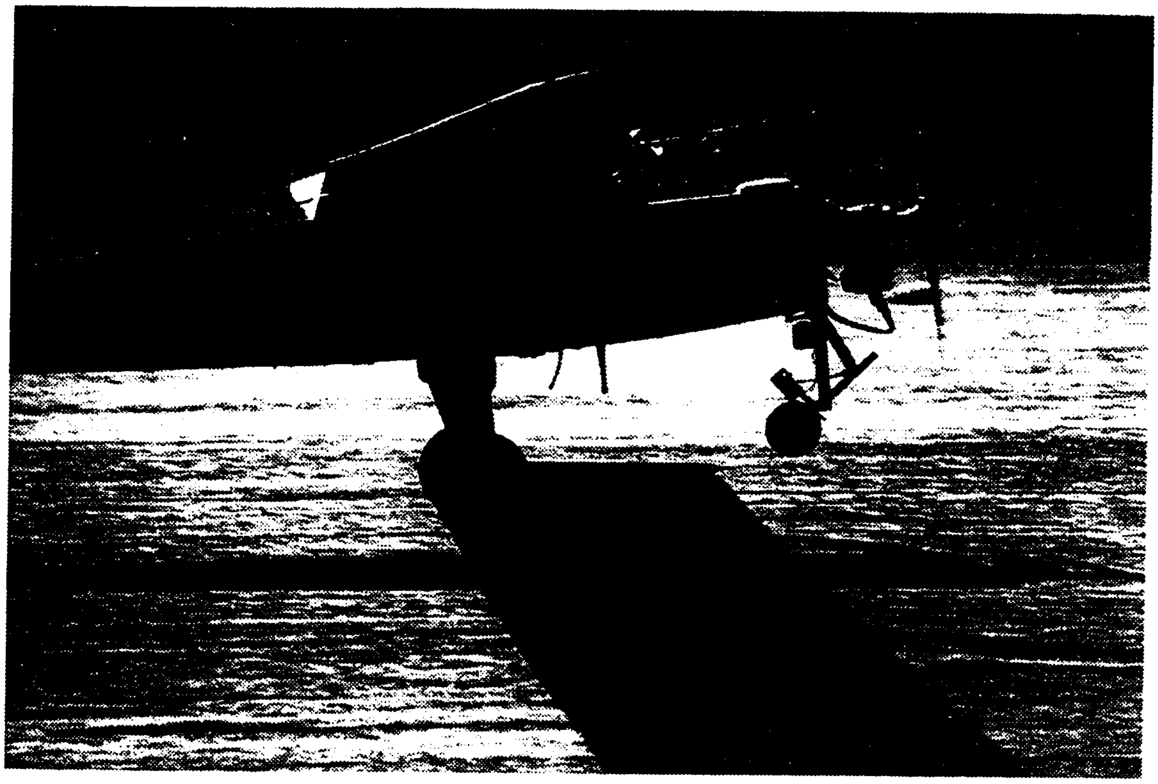


Figure 5.16: **Photograph Of Flare**

Flare with zero flap, just prior to touchdown. Note the position of the alpha vane, the pitch attitude and the aileron position. The nosewheel brake and telemetry antenna are also clearly visible in this photograph.

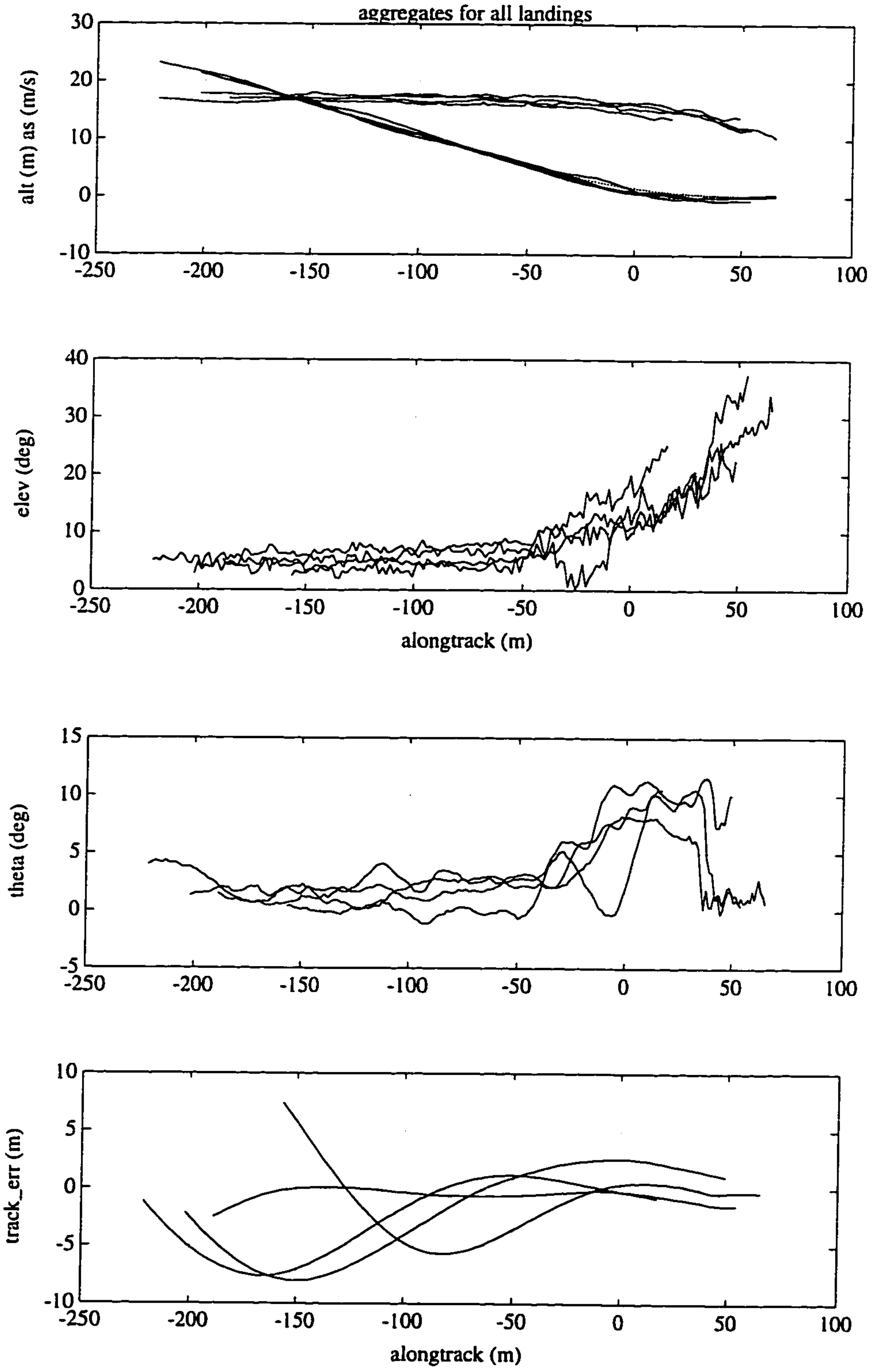


Figure 5.17: **Aggregate Landing Data**

Landing data from 4 right traffic landings taken on different days.

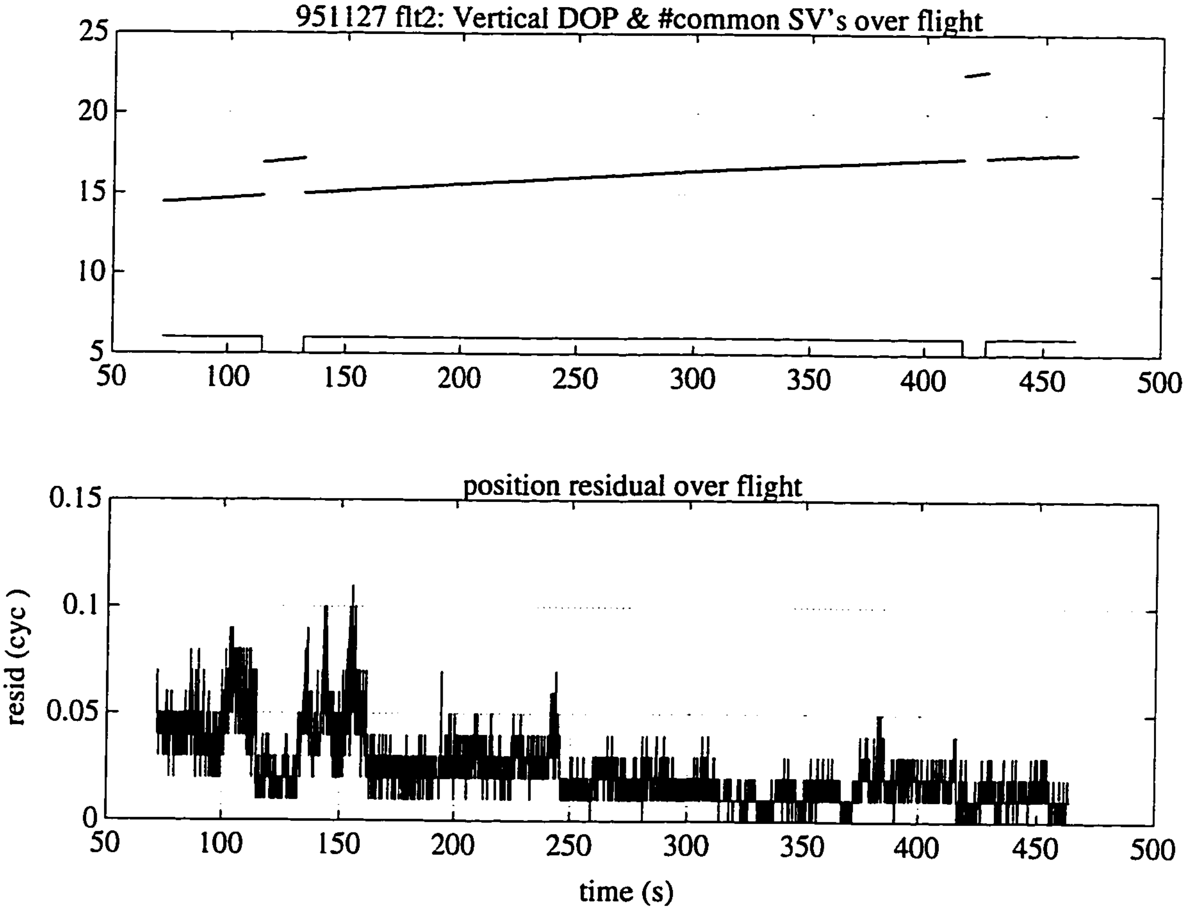


Figure 5.18: **Position Residual and DOP**

The upper plot shows the VDOP^2 and the number of satellites common to the air/ground receivers. Six satellites are held throughout most of the flight, dropping to five on two occasions due to occlusion of PRN 28. The lower plot shows the norm of the position residual in cycles^2 over approximately 7 minutes.

Chapter 6

Gyroscope Integration

This chapter investigates the use of inexpensive piezo-electric rate gyros integrated with GPS attitude measurements. The objectives are as follows:

- *Calibrate gyroscope biases.*
- *Provide a method by which earlier data may be weighted into the current attitude estimate.*
- *Enhance bandwidth and robustness.*
- *Provide attitude integrity monitoring.*
- *Rapidly reacquire attitude integers after a brief GPS outage.*

These advantages were the result of an investigation into how GPS attitude data might be used to calibrate biases on a triad of inexpensive gyros. Two techniques are presented. The first uses the GPS attitude solution or differential phase measurements in an extended Kalman filter. The second filters both the GPS angular-velocity solution and the gyro measurements in a steady state Kalman filter.

There has been much earlier work in this domain. The current contribution is an extension of a method presented by Lefferts, Markley and Shuster [50], to use measurements available from a GPS receiver. The algorithm was originally presented to integrate vector line of sight measurements from star sensors with onboard gyroscopes and was applied to spacecraft attitude estimation. The algorithm itself is *not* my contribution and is included here for clarity only. A number of simulation and experimental results are presented which use either raw GPS differential phase data or the GPS attitude estimate based on these data and filter these together with the gyro measurements from inexpensive Murata rate gyros.

6.1 Gyro bias estimation using GPS attitude

This section summarizes the derivation of the reduced order filter presented in [50], and its adaption for use with GPS measurements. The development uses the extended Kalman Filter equations which are summarized in appendix A.

6.1.1 Gyro Model

The onboard gyros consist of three, single axis piezo-electric (*tuning-fork*) rate-gyros mounted orthogonally and aligned with the aircraft attitude reference axes. The output of each axis is a voltage that is low-pass filtered by the electronics and subsequently sampled at 20 Hz. This (anti-aliasing) filter has a bandwidth of 5 Hz based on the -3 dB gain. The gyros are inexpensive and suffer from zero-point bias drift and noise in the output as described in section 2.2.5. Each axis is assumed independent and misalignment errors are assumed negligible. This is easily generalized by assuming that gyro outputs and biases are linear combinations of three (or more) gyros. A very simple model is used for each axis, which accounts for the bias drift and the measurement noise. The aircraft angular velocity vector is assumed to be related to the vector of scaled gyro outputs $\boldsymbol{z} = [z_x \; z_y \; z_z]^T$ by

$$\boldsymbol{\omega} = \boldsymbol{z} - \boldsymbol{b} - \boldsymbol{\nu}_1 \tag{6.1}$$

The vector $\boldsymbol{b}$ is the column matrix of gyro biases and $\boldsymbol{\nu}_1$ is a zero-mean, white Gaussian process with spectral density $Q_1(t)$ chosen to match the measurement noise characteristics specified in section 2.2.5.

$$E\{\boldsymbol{\nu}_1(t)\} = \mathbf{0} \qquad \text{and} \qquad E\{\boldsymbol{\nu}_1(t)\,\boldsymbol{\nu}_1^T(t')\} = Q_1(t)\,\delta(t-t') \tag{6.2}$$

The zero-point bias is assumed to be driven by a second Gaussian white-noise process with spectral density $Q_2(t)$ chosen to match the worst case time rate of change of the gyro bias.

$$\dot{\boldsymbol{b}} = \boldsymbol{\nu}_2 \tag{6.3}$$

$$E\{\boldsymbol{\nu}_2(t)\} = \mathbf{0} \qquad \text{and} \qquad E\{\boldsymbol{\nu}_2(t)\,\boldsymbol{\nu}_2^T(t')\} = Q_2(t)\,\delta(t-t') \tag{6.4}$$

It might be argued that this (*random walk*) model is contrived, since it implies bias variance growing linearly with time. As justification, one may consider an exponentially correlated model for bias drift, where

$$\dot{\boldsymbol{b}} = -\frac{1}{\tau}\boldsymbol{b} + \boldsymbol{\nu}_2 \tag{6.5}$$

Since persistent drifts are observed, the time constant must be large compared to other time constants in the system. Letting τ be infinite leads to the random walk equation (6.3) and is adequate in this context. Finally, the two noise processes are assumed uncorrelated

$$E\{\boldsymbol{\nu}_1(t)\,\boldsymbol{\nu}_2^T(t')\} = 0_{3\times3} \tag{6.6}$$

6.1.2 State Equation in $\mathbf{R}^7$

The attitude state vector is defined to be the concatenation of vector of Euler-Rodrigues symmetric parameters and the vector of gyro biases[1]

$$\boldsymbol{x}(t) \equiv \begin{bmatrix} \bar{\eta}(t) \\ \boldsymbol{b}(t) \end{bmatrix} \tag{6.7}$$

Substituting equation (6.1) into equation (4.77) gives the coupled differential equations

$$\begin{aligned} \dot{\bar{\eta}}(t) &= \frac{1}{2}\Omega_L\left(\boldsymbol{z}(t) - \boldsymbol{b}(t) - \boldsymbol{\nu}_1(t)\right)\bar{\eta}(t) && (6.8) \\ \dot{\boldsymbol{b}}(t) &= \boldsymbol{\nu}_2(t) && (6.9) \end{aligned}$$

Using equation (4.63) one may therefore rewrite equation (6.8) as

$$\dot{\bar{\eta}}(t) = \frac{1}{2}\Omega_L\left(\boldsymbol{z}(t) - \boldsymbol{b}(t)\right)\bar{\eta}(t) - \frac{1}{2}\Xi(\bar{\eta}(t))\,\boldsymbol{\nu}_1(t) \tag{6.10}$$

[1] The notation for the vector of gyro biases, $\boldsymbol{b}$, should not to be confused with baseline vectors $\boldsymbol{b}_i$.

This equation is linear in $\bar{\eta}$ and time varying. It is interesting to note that the gyro *measurement noise* $\boldsymbol{\nu}_1(t)$, now appears as *process noise* in the state equation. Such an equation may be solved using a transition matrix approach as given by equation (A.7).

Time Update

Taking expectations on both sides of equations (6.9) and (6.10), leads, within the approximation inherent in equation (A.6) to

$$\dot{\hat{\bar{\eta}}} = \frac{1}{2}\Omega_L(\hat{\boldsymbol{\omega}})\,\hat{\bar{\eta}} \qquad (6.11)$$

$$\dot{\hat{\boldsymbol{b}}} = \mathbf{0} \qquad (6.12)$$

where

$$\hat{\boldsymbol{\omega}} = \boldsymbol{z} - \hat{\boldsymbol{b}} \qquad (6.13)$$

is the estimated angular velocity. From equation (6.12) one infers that $\hat{\boldsymbol{b}}$ is constant over the update interval. Thus, $\hat{\boldsymbol{\omega}}$ depends only on $\boldsymbol{z}(t)$ and the initial value of the state vector. Therefore, equation (6.11) can be integrated directly to give

$$\hat{\bar{\eta}}(t) = \Theta(t,t_k)\,\hat{\bar{\eta}}(t_k) \qquad (6.14)$$

with

$$\frac{\partial}{\partial t}\Theta(t,t_k) = \frac{1}{2}\Omega_L(\hat{\boldsymbol{\omega}}(t))\,\Theta(t,t_k) \qquad \text{and} \qquad \Theta(t_k,t_k) = I_{4\times 4} \qquad (6.15)$$

Noting the similarity with equation (4.55), one may show that

$$\Theta(t,t_k) = \{\bar{\eta}(t,t_k)\}_L \qquad (6.16)$$

Measurement Update in $\mathbf{R}^7$

An obvious way to define the state-error vector and covariance matrix is to express these in terms of the state vector defined in equation (6.7). Unfortunately, this is not useful for filtering due to the necessary singularity of the error-state covariance

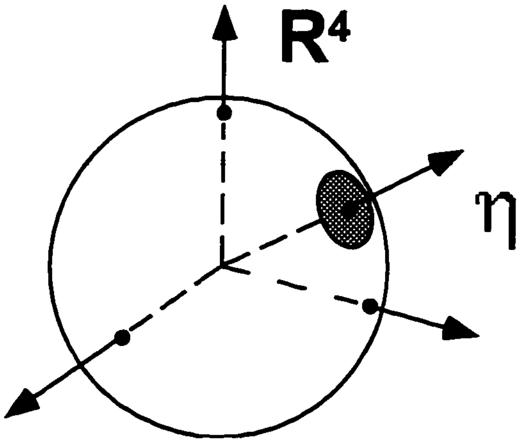


Figure 6.1: **Unit circle in $\mathbf{R}^4$**
P is necessarily singular since $\Delta\bar{\eta}$ must lie in the shaded section, and is perpendicular to $\bar{\eta}$.

matrix $P(t)$, and the consequent failure of the filter equations. Singularity results from the constraint on the norm of $\bar{\eta}$, such that

$$\Delta\bar{\eta}^T \hat{\bar{\eta}} = 0 \tag{6.17}$$

hence the vector $\left[\hat{\bar{\eta}}^T(t) \quad 0^T\right]^T$ is always a null vector of $P(t)$. To view this another way, the uncertainty of $\hat{\bar{\eta}}$ in the direction of $\bar{\eta}$ is zero since both are constrained to lie on the surface of the unit sphere in $\mathbf{R}^4$. Figure 6.1 attempts to illustrate a unit circle in $\mathbf{R}^4$ to show this graphically. For time update, the singularity of $P(t)$ is also problematic due to the accumulation of round-off error. This can lead to $P(t)$ becoming full rank and potentially indefinite.

6.1.3 Body Fixed Covariance Representation

One way to finesse the singularity problem is to project the state-space in $\mathbf{R}^7$ onto a subspace where the state-error covariance matrix is full rank. This can be achieved using an approximate (*body-referenced*) representation of the state-vector and covariance matrix. In this representation the elements of the covariance matrix have a natural interpretation, being body referenced angular errors and gyro biases respectively.

In the reduced representation, the error in $\bar{\eta}$ is *not* expressed as $\Delta\bar{\eta}$, the arithmetic difference between $\bar{\eta}$ and $\hat{\bar{\eta}}$, but as the quaternion, $\delta\bar{\eta} \equiv [\boldsymbol{\delta\eta}^T \quad \delta\eta_4]^T$, that must be *composed* with $\hat{\bar{\eta}}$ in order to obtain $\bar{\eta}$, ie.

$$\bar{\eta} = \delta\bar{\eta} \otimes \hat{\bar{\eta}} \overset{(4.57)}{=} \{\hat{\bar{\eta}}\}_R\, \delta\bar{\eta} = \left[\Xi(\hat{\bar{\eta}}) \mid \hat{\bar{\eta}}\right] \delta\bar{\eta} \tag{6.18}$$

Since this incremental quaternion represents a small rotation, the fourth component, $\delta\eta_4$, will be close to unity (to second order in the vector components). To first order, all the attitude information is contained in the three components of $\boldsymbol{\delta\eta}$, as may be inferred from the axis/angle interpretation. Indeed, from equation (4.54), $\boldsymbol{\delta\eta} = \boldsymbol{\Delta\xi}/2$.

Multiplying equation (6.18) from the left by $\{\hat{\bar{\eta}}\}_R^T$, it follows that

$$\boldsymbol{\delta\eta} = \Xi^T(\hat{\bar{\eta}})\, \bar{\eta} \tag{6.19}$$

and

$$\delta\eta_4 = \hat{\bar{\eta}}^T\, \bar{\eta} \approx 1 \tag{6.20}$$

Defining the reduced state (denoted by $\tilde{\boldsymbol{x}}$) as the six element vector

$$\tilde{\boldsymbol{x}} \equiv \begin{bmatrix} \boldsymbol{\delta\eta} \\ \boldsymbol{b} \end{bmatrix} \tag{6.21}$$

provides the desired (non-redundant) representation of the state. One may therefore define the 6 × 7 projection matrix, $S^T(\hat{\bar{\eta}})$, that maps $\boldsymbol{x}$ onto $\tilde{\boldsymbol{x}}$

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \Xi^T(\hat{\bar{\eta}}) & 0_{3\times 3} \\ 0_{3\times 4} & I_{3\times 3} \end{bmatrix} \begin{bmatrix} \bar{\eta} \\ \boldsymbol{b} \end{bmatrix} \equiv S^T(\hat{\bar{\eta}})\, \boldsymbol{x} \tag{6.22}$$

where $\boldsymbol{x}$ is the original seven element state vector. Taking expectations on both sides and using equation (4.62) it then follows that

$$\hat{\tilde{\boldsymbol{x}}} \equiv E\{\tilde{\boldsymbol{x}}\} = S^T(\hat{\bar{\eta}})\, \hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{0} \\ \hat{\boldsymbol{b}} \end{bmatrix} \tag{6.23}$$

and

$$\Delta\tilde{\boldsymbol{x}} \equiv \tilde{\boldsymbol{x}} - \hat{\tilde{\boldsymbol{x}}} = S^T(\hat{\bar{\eta}})\,\Delta\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\delta\eta} \\ \Delta\boldsymbol{b} \end{bmatrix} \tag{6.24}$$

By design, we have

$$\Delta\boldsymbol{x} = S(\hat{\bar{\eta}})\,\Delta\tilde{\boldsymbol{x}} \tag{6.25}$$

however, it happens that

$$\hat{\boldsymbol{x}} \neq S(\hat{\bar{\eta}})\,\hat{\tilde{\boldsymbol{x}}} \tag{6.26}$$

Prediction

From equation (4.82) and taking expectations we have

$$\frac{d}{dt}\hat{\bar{\eta}} = \frac{1}{2}\hat{\bar{\boldsymbol{\omega}}} \otimes \hat{\bar{\eta}} \tag{6.27}$$

and from the definition (equation (6.18)), $\delta\bar{\eta} \equiv \bar{\eta} \otimes \hat{\bar{\eta}}^{-1}$, it follows that

$$\frac{d}{dt}\,\delta\bar{\eta} = \frac{d}{dt}(\bar{\eta}) \otimes \hat{\bar{\eta}}^{-1} + \bar{\eta} \otimes \frac{d}{dt}\left(\hat{\bar{\eta}}^{-1}\right) = \frac{1}{2}\left[\bar{\boldsymbol{\omega}} \otimes \delta\bar{\eta} - \delta\bar{\eta} \otimes \hat{\bar{\boldsymbol{\omega}}}\right] \tag{6.28}$$

where use has been made of the fact that

$$\frac{d}{dt}\left(\bar{\eta}^{-1}\right) = -\bar{\eta}^{-1} \otimes \frac{d}{dt}(\bar{\eta}) \otimes \bar{\eta}^{-1} \tag{6.29}$$

hence

$$\frac{d}{dt}\delta\bar{\eta} = \frac{1}{2}\left[\hat{\bar{\boldsymbol{\omega}}} \otimes \delta\bar{\eta} - \delta\bar{\eta} \otimes \hat{\bar{\boldsymbol{\omega}}}\right] + \frac{1}{2}\Delta\bar{\boldsymbol{\omega}} \otimes \delta\bar{\eta} \tag{6.30}$$

where

$$\Delta\bar{\boldsymbol{\omega}} \equiv \begin{bmatrix} \Delta\boldsymbol{\omega} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} = \begin{bmatrix} -\Delta\boldsymbol{b} - \boldsymbol{\nu}_1 \\ 0 \end{bmatrix} \tag{6.31}$$

Now, by direct multiplication it may be shown that

$$\frac{1}{2}\left[\hat{\bar{\boldsymbol{\omega}}} \otimes \delta\bar{\eta} - \delta\bar{\eta} \otimes \hat{\bar{\boldsymbol{\omega}}}\right] = \begin{bmatrix} -\hat{\boldsymbol{\omega}} \times \boldsymbol{\delta\eta} \\ 0 \end{bmatrix} \tag{6.32}$$

and

$$\Delta\bar{\boldsymbol{\omega}} \otimes \delta\bar{\eta} = \Delta\bar{\boldsymbol{\omega}} + O(|\Delta\boldsymbol{\omega}|\ |\boldsymbol{\delta\eta}|) \tag{6.33}$$

Thus, neglecting second-order terms

$$\frac{d}{dt}\boldsymbol{\delta\eta} = -\hat{\boldsymbol{\omega}} \times \boldsymbol{\delta\eta} - \frac{1}{2}\Delta\boldsymbol{b} - \frac{1}{2}\boldsymbol{\nu}_1 \tag{6.34}$$

$$\frac{d}{dt}\delta\eta_4 = 0 \tag{6.35}$$

from which the following state-error equation may be written

$$\frac{d}{dt}\Delta\tilde{\boldsymbol{x}}(t) = \tilde{F}(t)\Delta\tilde{\boldsymbol{x}} + \tilde{G}(t)\boldsymbol{w}(t) \tag{6.36}$$

where

$$\tilde{F}(t) = \begin{bmatrix} [[\hat{\boldsymbol{\omega}}(t)]] & -\frac{1}{2}I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \neq S^T(\hat{\bar{\eta}})\,F(t)\,S(\hat{\bar{\eta}}) \tag{6.37}$$

$$\tilde{G}(t) = \begin{bmatrix} -\frac{1}{2}I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} = S(\hat{\bar{\eta}})\,G(t) \tag{6.38}$$

The transition matrix my be written as

$$\tilde{\Phi} = \begin{bmatrix} \tilde{\Theta} & \tilde{\Psi} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{6.39}$$

with

$$\frac{\partial}{\partial t}\tilde{\Theta}(t, t_0) = [[\hat{\boldsymbol{\omega}}(t)]]\,\tilde{\Theta}(t, t_0) \tag{6.40}$$

$$\frac{\partial}{\partial t}\tilde{\Psi}(t, t_0) = [[\hat{\boldsymbol{\omega}}(t)]]\,\tilde{\Psi}(t, t_0) - \frac{1}{2}I_{3\times3} \tag{6.41}$$

subject to

$$\tilde{\Theta}(t_0, t_0) = I_{3\times3} \qquad \text{and} \qquad \tilde{\Psi}(t_0, t_0) = 0_{3\times3} \tag{6.42}$$

it then follows immediately that

$$\tilde{\Theta}(t, t_0) = \Lambda(t, t_0) \tag{6.43}$$

$$\tilde{\Psi}(t, t_0) = -\frac{1}{2}\int_{t_0}^{t} \Lambda(t, t')\,dt' \equiv K(t, t_0) \tag{6.44}$$

so that

$$\tilde{\Phi}(t, t_0) = \begin{bmatrix} \Lambda(t, t_0) & K(t, t_0) \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{6.45}$$

The covariance matrix is defined as

$$\tilde{P}(t) = E\{\Delta\tilde{\boldsymbol{x}}(t)\,\Delta\tilde{\boldsymbol{x}}^T(t)\} \tag{6.46}$$

and from equation (6.25) it follows that

$$\tilde{P}(t) = S^T(\hat{\bar{\eta}}(t))\,P(t)\,S(\hat{\bar{\eta}}(t)) \tag{6.47}$$

$$P(t) = S(\hat{\bar{\eta}}(t))\,\tilde{P}(t)\,S^T(\hat{\bar{\eta}}(t)) \tag{6.48}$$

where $\tilde{P}(t)$ satisfies the Riccati equation

$$\frac{d}{dt}\tilde{P}(t) = \tilde{F}(t)\tilde{P}(t) + \tilde{P}(t)\tilde{F}^T(t) + \tilde{G}(t)Q(t)\tilde{G}^T(t) \tag{6.49}$$

the integral form of which is analogous to equation (A.14).

Measurement Update

It may be shown by direct substitution into the filter equations in $\mathbf{R}^7$ that

$$\tilde{K}_k = S^T\left(\hat{\bar{\eta}}_k(-)\right)\,K_k \qquad \text{and} \qquad \tilde{H}_k = H_k\,S\left(\hat{\bar{\eta}}_k(-)\right) \tag{6.50}$$

The measurement update is performed on the reduced state as

$$\Delta\hat{\tilde{\boldsymbol{x}}}(+) = \begin{bmatrix} \delta\hat{\boldsymbol{\eta}}(+) \\ \Delta\hat{\boldsymbol{b}}(+) \end{bmatrix} = \tilde{K}\,\left[\boldsymbol{z} - \boldsymbol{h}\left(\hat{\bar{\eta}}(-)\right)\right] \tag{6.51}$$

where K, H and P are replaced in equations (A.20) and (A.22) with $\tilde{K}$, $\tilde{H}$ and $\tilde{P}$ respectively. The posteriori[2] estimate of $\boldsymbol{x}$ is then calculated as

$$\hat{\bar{\eta}}(+) = \begin{bmatrix} \delta\hat{\boldsymbol{\eta}}(+) \\ 1 \end{bmatrix} \otimes \hat{\bar{\eta}}(-) \tag{6.52}$$

$$\hat{\boldsymbol{b}}(+) = \hat{\boldsymbol{b}}(-) + \Delta\hat{\boldsymbol{b}}(+) \tag{6.53}$$

As measurements, one can potentially use either: (a), the attitude estimate, $\bar{\eta}_{gps}$, from the linearized batch attitude calculation described in section 4.4.1, or (b), the raw GPS differential phase data, $\Delta\phi_{ij}$.

[2] In the following *posteriori* is used in place of *a posteriori*.

Method (a). Measurement Update from GPS Attitude. The body referenced covariance matrix is a by-product of the batch method and may be used as the measurement covariance matrix, R, in the update. As described in section 4.4.1, this covariance may include a weighting according to the signal strength of each measurement. With this approach, and using equation (6.19), one has the body-referenced perturbation

$$\delta\boldsymbol{\eta}_{gps} = \Xi^T\left(\hat{\bar{\eta}}(-)\right)\,\bar{\eta}_{gps} \tag{6.54}$$

with equation (6.51) becoming simply

$$\Delta\hat{\tilde{\boldsymbol{x}}}(+) = \begin{bmatrix} \hat{\delta\boldsymbol{\eta}}(+) \\ \Delta\hat{\boldsymbol{b}}(+) \end{bmatrix} = \tilde{K}\,\delta\boldsymbol{\eta}_{gps} \tag{6.55}$$

The measurement is not dependent on the bias states, and the sensitivity matrix, $\tilde{H}$, reduces to $[I_{3\times3}\,0_{3\times3}]$. The equations for $\tilde{K}$ and $\tilde{P}(+)$ simplify accordingly, requiring only a 3×3 matrix inversion. A numerical example follows in section 6.1.5.

Method (b). Measurement Update from Differential Phase. Alternatively, ignoring multipath, (or having removed it by prior calibration), the raw phase measurements can be assumed white and uncorrelated with variance $\sigma^2_{\Delta\phi}$ and may be used as the measurements in the filter. Once again, a weighting based on signal strength could easily be included. With three baselines and six channels, there are up to eighteen scalar differential-phase measurements to include in the measurement update. Each scalar measurement is related to the attitude by the nonlinear measurement equation

$$h_{ij}(\tilde{\boldsymbol{x}}) = \Delta\phi_{ij} = \boldsymbol{b}_i^T\,\mathcal{R}(\bar{\eta}(-))\;\hat{\boldsymbol{s}}_j. \tag{6.56}$$

Again, there is no dependence on the bias states, and the (1×6) sensitivity matrix for the ij'th measurement is given by

$$\tilde{H}_{ij} = \left[\frac{\partial\Delta\phi_{ij}}{\partial\,\delta\boldsymbol{\eta}}\Big|_{\hat{\bar{\eta}}(-)} \qquad 0^T\right] \overset{(4.99)}{=} \left[2\,\hat{\boldsymbol{s}}_j^T\,\mathcal{R}^T\left(\hat{\bar{\eta}}(-)\right)\,[[\boldsymbol{b}_i]] \qquad 0^T\right] \tag{6.57}$$

Stacking up $\tilde{H}_{ij}$ over all ij pairs, one may construct the $(n \times 6)$ sensitivity matrix $\tilde{H} \equiv \left[\tilde{H}_{\delta\eta} \quad 0_{n\times3}\right]$. The equations for $\tilde{K}$ and $\tilde{P}(+)$ then involve the inversion of an $n \times n$ matrix, where n is the number of measurements available. This may be simplified using the Woodbury-Shur matrix inversion lemma [25] to require only two. 3×3 inversions, giving

$$\tilde{K} = \frac{-1}{\sigma^2_{\Delta\phi}} \begin{bmatrix} P_{\delta\eta\delta\eta}(-) \\ P_{\delta\eta\Delta b}(-) \end{bmatrix} \tilde{H}^T_{\delta\eta} \left[\tilde{H}_{\delta\eta} \left(\tilde{H}^T_{\delta\eta} \tilde{H}_{\delta\eta} + \sigma^2_{\Delta\phi} P^{-1}_{\delta\eta\delta\eta}(-) \right)^{-1} \tilde{H}^T_{\delta\eta} - I_{n\times n} \right] \quad (6.58)$$

and

$$\tilde{P}(+) = \tilde{P}(-) - \tilde{K}\,\tilde{H}_{\delta\eta}\,[P_{\delta\eta\delta\eta}(-) \quad P_{\Delta b\delta\eta}(-)] \quad (6.59)$$

This additional computational burden is not evident using method (a). Overall however, the methods have similar computational requirements since an initial iteration is required to solve for the GPS attitude solution and its covariance matrix. This iteration can usefully be initialized with $\hat{\bar{\eta}}(-)$ rather than the previous attitude solution (as is currently done), particularly if the attitude may have changed significantly during the update period.

For initialization, gyro bias covariance is set to a large value. Even if there is no attitude motion, the gyro biases converge rapidly, typically reaching statistical steady state in a few seconds. It is clear that attitude integers are required for the measurement update, however, with an estimate of the attitude, $\hat{\bar{\eta}}(-)$, the detection and isolation of even multiple cycle slips becomes relatively straightforward. This is not true when using the solution residual as a cycle-slip detector and parity methods for isolation.

Unbiased Gyro Integration.

A simple integration, ignoring bias errors shows that, with the noise characteristics of inexpensive gyros and the baselines of the autonomous aircraft, propagation for approximately ten seconds is feasible. Starting with the time varying linear differential equation

$$\frac{d}{dt}\boldsymbol{\delta\eta}(t) = [[\hat{\omega}(t)]]\,\boldsymbol{\delta\eta}(t) - \frac{1}{2}\,\boldsymbol{\nu}_1 \quad (6.60)$$

where $E\{\boldsymbol{\nu}_1\,\boldsymbol{\nu}_1{}^T\} = I_3\,\sigma^2$. Integrating formally, we have

$$\boldsymbol{\delta\eta}(t) = \Theta(t, t_0)\,\boldsymbol{\delta\eta}(t_0) \quad (6.61)$$

where

$$\frac{\partial \Theta(t,t_0)}{\partial t} = [[\hat{\omega}(t)]]\,\Theta(t,t_0) \qquad \text{and} \qquad \Theta(t_0,t_0) = I_{3\text{x}3} \tag{6.62}$$

from which one concludes

$$\Theta(t,t_0) = \mathcal{R}(t,t_0) \tag{6.63}$$

where $\mathcal{R}(t,t_0)$ is the direction cosine matrix relating the attitude at time t_0 to that at time t. The integrated Riccati equation is then

$$P_{\delta\eta\delta\eta}(t) = \frac{1}{4}\,P_{\Delta\xi\Delta\xi}(t) = \mathcal{R}(t,t_0)\,P_{\delta\eta\delta\eta}(t_0)\,\mathcal{R}^T(t,t_0) + \frac{1}{4}\int_{t_0}^{t} \mathcal{R}(t,t')\,\left(I_3\,\sigma^2\right)\mathcal{R}^T(t,t')\,dt' \tag{6.64}$$

hence

$$P_{\Delta\xi\Delta\xi}(t) = \mathcal{R}(t,t_0)\,P_{\Delta\xi\Delta\xi}(t_0)\,\mathcal{R}^T(t,t_0) + (t-t_0)I_3\,\sigma^2 \tag{6.65}$$

indicating that, ignoring bias terms, the angular covariance grows linearly with time.

If the intention is to back out the cycle integers, as described in section 4.4.3, then the differential phase error due to the attitude uncertainty must be less than 0.5 cycles of the GPS carrier along the line of sight to the satellite. Assuming perfect initial knowledge, a worst case satellite orientation and 99% (3σ) probability of successful integer determination, the maximum propagation time[3] ΔT is bounded by

$$\Delta T < \frac{1}{2^2\,3^2\,|b_{max}|^2\,\sigma_{gyro}^2} \tag{6.66}$$

For a baseline length of 15 ($\sim$ 3m) carrier cycles and assuming $\sigma_{gyro} = 0.1$ deg/s, one finds that approximately 40 seconds of propagation may be achieved. If $\sigma_{gyro} = 0.2$ deg/s then the propagation time drops to only 10 seconds. Since gyro noise is typically specified in (deg/s)/$\sqrt{Hz}$ it is clear that the noise power depends on the the bandwidth of the anti-aliasing filter employed. The Murata gyros used in this work do not provide a noise power spectral density (PSD) statistic. Based on observed one sigma noise of approximately 0.14 deg/s and assuming a perfect low pass filter with cutoff at 5 Hz[4] one calculates an approximate noise density of 0.044 (deg/s)/$\sqrt{Hz}$. This is close to the stated noise statistic of 0.05 (deg/s)/$\sqrt{Hz}$ for the less expensive of

[3] Period of time after which the attitude covariance is too large to "back-out" attitude integers based on knowledge of the attitude.

[4] A three pole antialiasing filter rolling off at 5 Hz was used while collecting experimental data.

two commercially available Systron-Donner piezo-gyros which use similar technology and for which noise PSD statistics are provided. For many GPS attitude outages. even 10 seconds of "flywheeling" is more than adequate, indicating that it may be feasible to regain GPS attitude integers after a total loss of carrier lock. The major error source is attributable to bias drift, however the analysis without biases places a useful upper bound on the maximum tolerable GPS outage duration.

6.1.4 Summary

To summarize the above, propagation of the attitude state is performed using equations (6.11) and (6.12). In this, the four components of the attitude quaternion are treated as independent variables, with the normalization constraint being maintained. within numerical error by the structure of equation (6.11). This normalization can be destroyed by the accumulation of round-off error and by the small rotation approximation inherent in the update equation. The constraint is however simply *and (least squares) optimally* restored by scaling to unity norm. Measurement update is achieved in an error space where the error covariance matrix is full rank, and has a natural interpretation in terms of body-relative angular errors and gyro-bias errors. The update may be done in terms of an attitude estimate from the current GPS attitude algorithm, or using the raw GPS relative-phase measurements. The transition matrix and covariance matrix are calculated as 6 × 6 matrices throughout, which avoids the numerical problems associated with rank deficiency of the equivalent matrices in $\mathbf{R}^7$. Computationally, the largest burden is imposed by the evaluation of the transition matrix in equation (6.45), and the contribution of process noise to the state covariance matrix in equation (A.14). It should be noted that the given equations are in the continuous domain. Since the time update must be evaluated in a computer, some form of integration technique is required. Assuming sampling at a rate sufficient that $M(\boldsymbol{\Delta\theta})$ is small, then equation (4.79) provides a suitable method for updating the attitude state. As discussed in section 6.1.5, a more accurate integration technique becomes necessary if a lower gyro sampling rate is desired.

This section has shown how attitude and attitude covariance may be propagated using an inexpensive set of onboard rate gyros. In addition to providing the ability to coast through GPS outages and increase the sample rate and bandwidth of the attitude system, the gyros enable the optimal inclusion of previous data in the current estimate. Two different methods for using GPS data as measurements in a Kalman filter have been described. In both, GPS data are used to calibrate the gyro biases, which are assumed to vary much more slowly than the vehicle dynamics. Both methods have similar computational requirements and yield similar results in simulation.

If one uses the existing (batch) attitude solution, the additional computational requirements for the measurement update are small, involving a (3×3) matrix inverse in the calculation of $\tilde{K}$ at each time step. The major cost is associated with propagation of the attitude covariance matrix.

One can conceive of an integrated "strap-down" GPS attitude receiver that includes a set of inexpensive gyros rendered in silicon. This marriage has the potential to offer bandwidth, robustness and sample rate far greater than is available from current GPS attitude implementations. Finally, for suitably short GPS outages where cycle integer knowledge is lost, knowledge of the attitude error covariance matrix indicates whether it is feasible to recover the cycle integers directly without resorting to the much more laborious integer resolution techniques discussed earlier.

6.1.5 Simulation

Four separate computer simulations were performed in MATLAB to validate and compare the filters. The matlab code that was used to run these simulations is given in appendix C. The first was an implementation of filter update using the GPS batch attitude estimate $\bar{\eta}_{gps}$ (method a). Once this filter was working, the second simulation was a Monte-Carlo simulation of 25 runs to compare simulated results against the predicted covariance envelopes. This simulation includes sections of initial convergence to steady state, steady state and then a section where the GPS measurements were disabled to simulate a GPS attitude failure. The intention of the last step was to understand how lengthy a dropout might be tolerated before direct recovery of the attitude integers became impossible. The third simulation implements the filter as described in method (b). The final simulation uses an array with short baselines to demonstrate the accuracy improvement that may be realized.

The simulations start with an assumed angular velocity history and generate the corresponding attitude history by integrating the kinematic relations with an accurate numerical integration routine. These then form the *reference* angular velocity $\omega_{ref}(t)$, and attitude $\bar{\eta}_{ref}(t)$, profiles against which filter performance is measured. The body-referenced angular velocity history was (arbitrarily) specified as

$$\omega_{ref}(t) = \begin{bmatrix} \sin(\pi t + \pi/6) \\ \sin(\pi t + \pi/3) \\ \sin(\pi t/2 + \ pi/2) \end{bmatrix} \tag{6.67}$$

The reference attitude history, represented as Euler angles is plotted in figure 6.2. Note that this (sinusoidal) choice of angular velocity causes the body to oscillate rather than move continuously. Other angular velocity histories have been tested in simulation, including the stationary case, $\omega_{ref}(t) = \mathbf{0}$. In all cases the simulation results were similar in character to those reported below. The reference history has relatively high frequency oscillations. This was chosen to exercise the accuracy of the integration routine and not as an attempt to model a typical flight profile. The final simulation uses a stationary array.

The antenna baseline vectors used were similar to the autonomous aircraft. Six satellite line of sight vectors were arbitrarily chosen, but with geometry representative of GPS satellite coverage. These were assumed motionless over the simulation duration of 20 seconds. A GPS update rate of 10 Hz was assumed throughout, but gyro sample rate was varied to attain acceptable results. After some experimentation it was found that the Euler update step given by equation (4.79) gave poor filter performance even with a sample rate of 40 Hz. The delay inherent in Euler integration caused the attitude estimate residuals to correlate strongly with the angular velocity history. For this reason, a 4'th order Runge-Kutta integration scheme was adopted for time update of the attitude state. This routine has stable numerical characteristics, is self-starting, and provides much greater accuracy in the attitude time update. The cost is an increase in the number of function evaluations from one

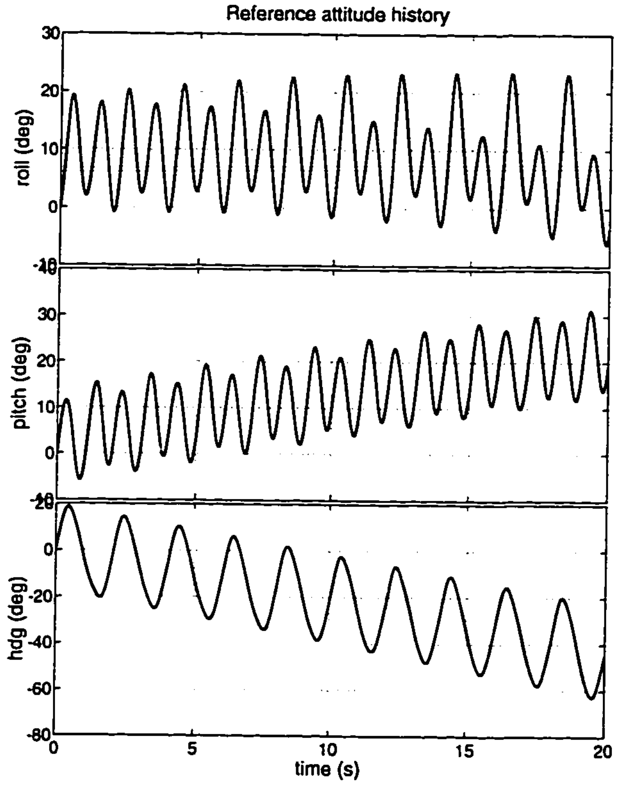


Figure 6.2: **Reference Attitude History.**

Euler angle representation of integrated angular velocity history.

to four. Fortunately, the attitude rate function is relatively inexpensive to evaluate and the additional computation is justified. A small latency cost is also inherent, in that the angular velocity sample at t_k is necessary to evaluate the apriori estimate of the attitude at t_k. Since most of the time update calculation can be pre-computed, this is not a major impediment. The Runge-Kutta algorithm for the attitude time update from t_k to t_{k+1} may be written

$$k_1 = T\Xi\left(\bar{\eta}_{t_k}(+)\right)\hat{\omega}_{t_k} \tag{6.68}$$

$$k_2 = T\Xi\left(\bar{\eta}_{t_k}(+) + \frac{k_1}{2}\right)\hat{\omega}_{t_{k+\frac{1}{2}}} \tag{6.69}$$

$$\begin{aligned}
k_3 &= T\Xi\left(\bar{\eta}_{t_k}(+) + \frac{k_2}{2}\right)\hat{\omega}_{t_{k+\frac{1}{2}}} && (6.70)\\
k_4 &= T\Xi\left(\bar{\eta}_{t_k}(+) + k_3\right)\hat{\omega}_{t_{k+1}} && (6.71)\\
\bar{\eta}_{t_{k+1}}(-) &= \bar{\eta}_{t_k}(+) + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) && (6.72)
\end{aligned}$$

where $T = t_{k+1} - t_k$, and $t_{k+\frac{1}{2}}$ refers to the sample midway between t_k and t_{k+1}. Note that the measurement update occurs after calculation of $\bar{\eta}_{t_{k+1}}(-)$ and before the subsequent evaluation of k_1. Simulation shows that a second order accurate Runge-Kutta algorithm does adequately well, requiring with only two function evaluations instead of four.

Covariance time update is evaluated at the intersample interrupt, thereby spreading the computational load over the available interrupts and minimizing latency at measurement update.

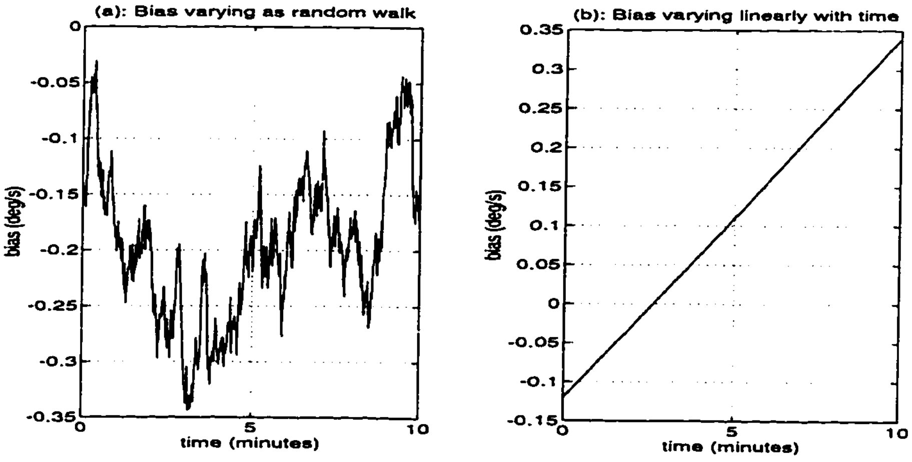


Figure 6.3: **Gyro Bias Modeling**

Plot (a) shows a random walk model of gyro bias over time. Plot (b) shows gyro bias varying linearly with time with an initial bias of -0.12 deg/s

The simulations use GPS differential-phase noise samples which are white and uncorrelated among measurements, with an RMS value of 1 cm. The noise, ν_1, on the gyro measurement samples are similarly white and uncorrelated, with an RMS value of 0.1 deg/s on each axis. These values approximate the expected noise values

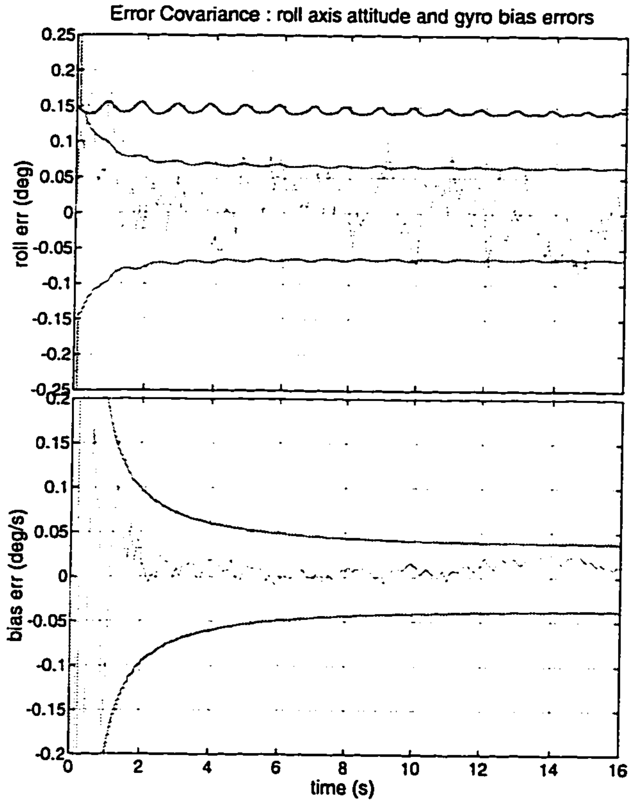


Figure 6.4: **Filter Simulation 1.**

Attitude filter using batch GPS attitude for measurement update.

found with the GPS attitude receiver and gyros used in this work. These values are also assumed by the filter.

The filter assumes the noise on the gyro bias states is white and uncorrelated, with spectral density Q_2, chosen such that the 1-sigma random walk envelope on the bias state grows at a rate which matches the maximum time rate of change of the gyro bias as quoted by the manufacturer. For the Murata gyros, the manufacturer quotes zero point stability to be better than 1 deg/s over 10 minutes. This (conservative) statistic accommodates the large time rates of change seen in the gyro biases after initial power up and before thermal equilibrium has been attained. This *random walk* model of the gyro bias is mathematically tractable, however is not representative of how (Murata) gyros actually drift. That is, the bias on a (Murata) rate gyro does not have a time history that resembles a statistical random walk such as that shown in figure (6.3a). If one observes the bias on a stationary gyro from the moment power is applied, one typically finds an initial (unknown) offset (bias), which may subsequently be observed to change smoothly over time as shown in figure (6.3b)[5]. For this reason the simulation uses biases which change over time as shown in figure (6.3b). The filter is unaware of the initial offset and must estimate this together with any subsequent drift that may occur over time. The *simulated* drift rates (gradient of the line in figure (6.3b)) are chosen from a uniform distribution with a maximum value equal to 1 deg/s over 10 minutes.

The state error covariance is initialized with large values on the diagonal elements and with zeros elsewhere. The bias offsets are initialized to be as much as 1 deg/s on each axis, this being the largest offset expected in practice. Gyro scale factor or alignment errors have not been modeled.

In simulation, the filter proceeds to estimate the unknown gyro biases and to reduce the state error covariance until a statistical steady state is achieved. The posteriori values are compared against the truth state and the errors (innovations) plotted together with the square root of the corresponding diagonal element of the covariance estimate (both apriori and posteriori). The attitude state errors may be interpreted as body referenced angular perturbations, with one plot being generated for each axis. Each simulation run takes a different path depending on the initialization seed used by the random number generator, so the state innovations history (plotted as a fine line) for any individual run is not necessarily representative. The predicted (1σ) covariance envelopes are plotted with thick lines. On average, one expects approximately 66% of the innovation samples to lie within the covariance envelope, and the error distribution to be uncorrelated over time.

[5] After a gyro has thermally stabilized the bias generally settles down to an (almost) constant value.

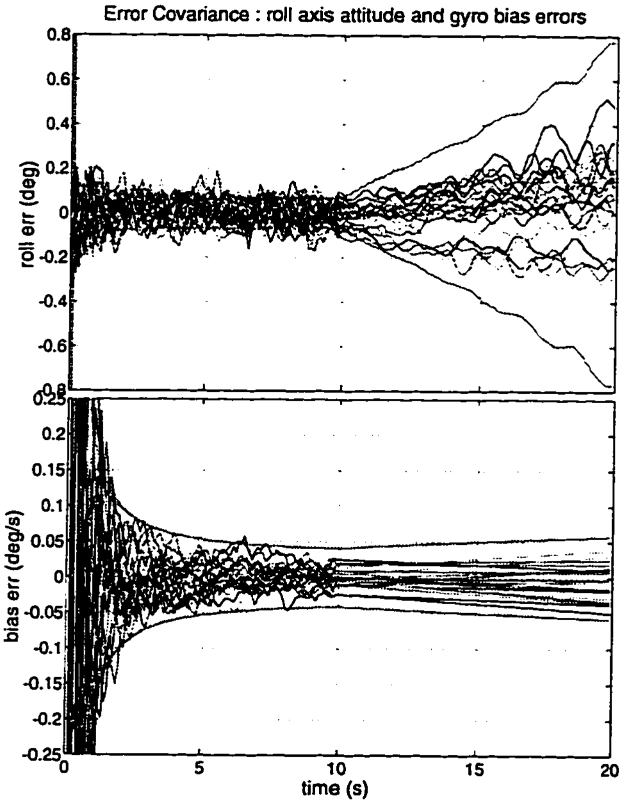


Figure 6.5: **Filter Simulation 2a.**
Attitude filter Monte Carlo with GPS measurement update failure.
Bias process simulated as a random linear ramp.

Simulation 1.

Figure 6.4 shows the outcome of one simulation run over 16 seconds using method (a). The plot shows the roll axis innovations history for the angular and bias errors. For economy, and because the roll axis plot displays behaviour that is largely representative of the pitch and yaw axes, plots of pitch and yaw innovations are omitted. The heavy wavy line above the covariance envelope is the roll axis 1σ covariance estimate from the GPS batch attitude calculation. One can see that addition of crude gyros improves accuracy by almost a factor of two. It is worth noting that the solution converges within a few seconds. This was found to be largely independent of the

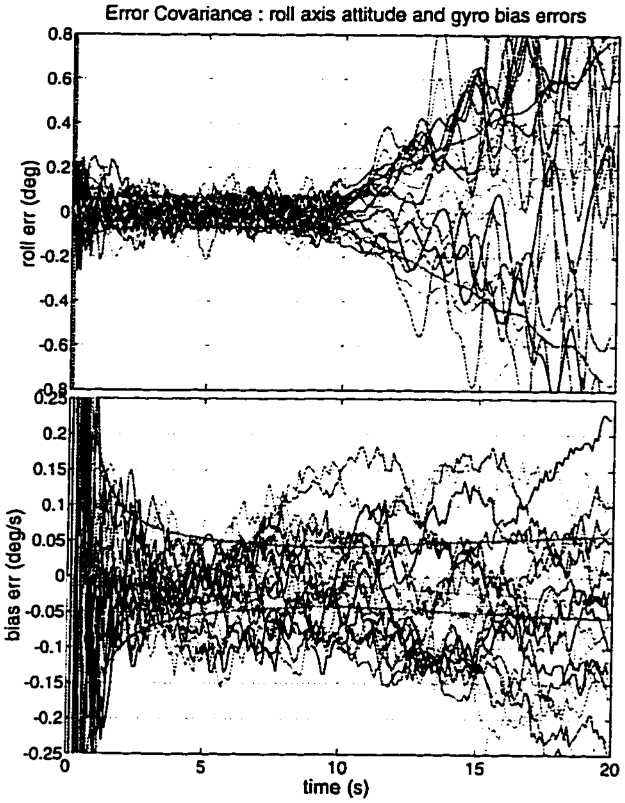


Figure 6.6: **Filter Simulation 2b.**
Attitude filter Monte Carlo with GPS measurement update failure.
Bias process simulated as a random walk.

specific attitude history chosen. The residuals appear white and unbiased and have approximately the correct variance. The gyro bias error converges to better than 0.05 deg/s within 8 seconds.

Simulation 2.

Figure 6.5 shows the outcome of a Monte Carlo simulation with 25 separate runs using method (a). The data plotted are identical to those for simulation 1, except that after 10 seconds, measurement updates are inhibited to simulate a GPS failure. The 1σ error bounds appear overly conservative in this plot. This may be attributed

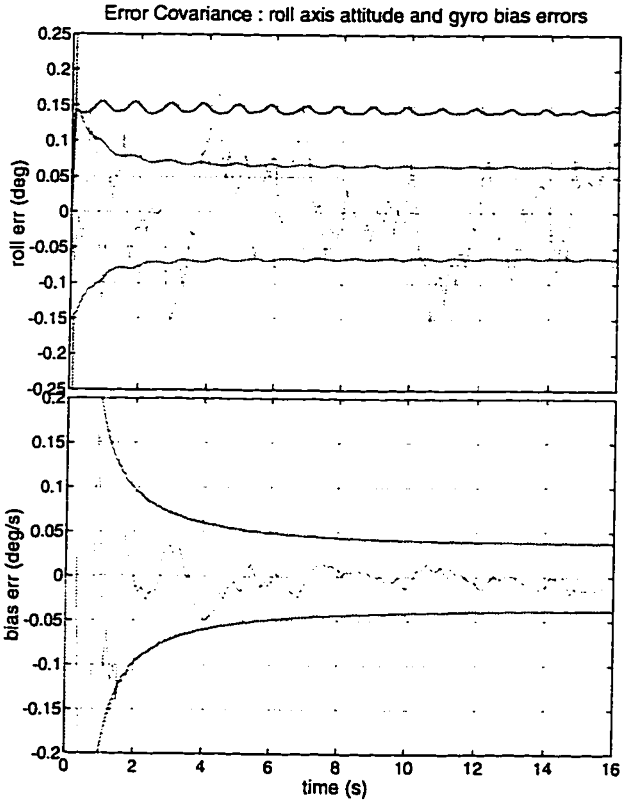


Figure 6.7: **Filter Simulation 3.**

Attitude filter using GPS differential phase for measurement update.

to the difference between the modeled and the simulated bias process. Figure 6.6 is an identical simulation where the bias process has been replaced by a true random walk as assumed by the filter. In this case, the covariance estimates appear to be underestimated. The actual bias process is less smooth than a linear ramp and less variable than a random walk, so covariance growth in reality is likely to be close to the predicted. After the GPS failure, the predicted angular error standard deviation envelope grows at approximately 0.07 deg/s about each axis. Using the same logic as earlier (section 6.1.3), with a worst orientation, one concludes that approximately 9 seconds of "flywheeling" are possible, with a 99% (3σ) probability of correct cycle integer resolution for this antenna configuration and the assumed gyro

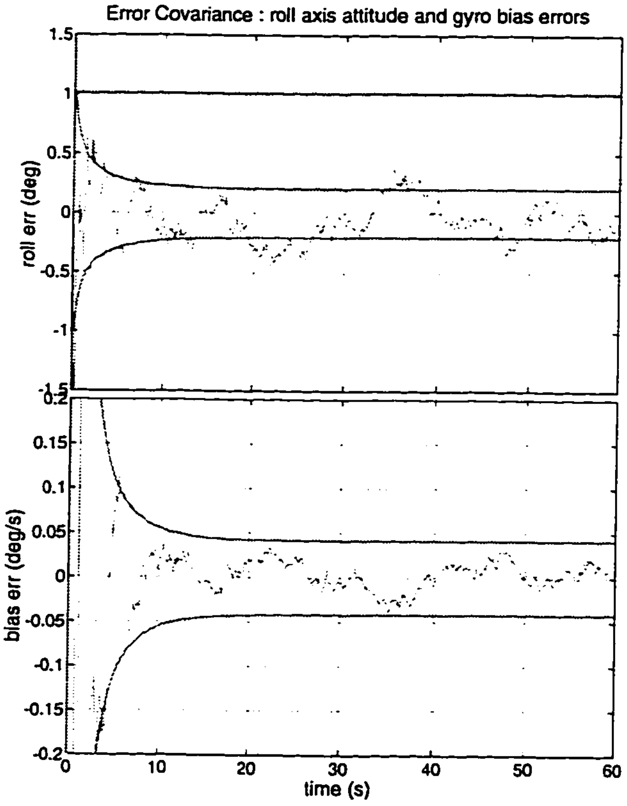


Figure 6.8: **Short Baseline Simulation.**
Attitude filter for a stationary, short baseline array using method (a).

error characteristics. This number can be increased by solving for the least sensitive integers first, as described in section 4.4.3. Finally, as expected, the observed error accumulation is faster than that attributable to gyro rate noise alone.

Simulation 3.

Figure 6.7 shows the outcome of one simulation run over 16 seconds using method (b). Since all axes display similar error characteristics, only the roll axis is plotted. This is directly comparable with figure (6.4). It may be seen that both figures show very similar convergence and steady state performance. This is not surprising in the converged filter since identical data is available to each. One might however expect the two results to differ if the initial guess was very poor. The batch formulation will iterate as necessary until the residual stops decreasing. The measurement update approach allows for only one iteration.

Short Baseline Simulation.

Figure 6.8 shows the outcome of one simulation run over 60 seconds using method (a). This simulation uses the same satellite geometry as all previous simulations, however the antenna array is held stationary throughout and the baselines have been scaled so that the longest baseline is 2 carrier cycles in length. The shorter baselines each have a length of 1.36 cycles. This simulation shows filter convergence in the absence of attitude motion and indicates that accuracy is improved by a factor of approximately four in this case. Pitch and yaw axes show similar behaviour and are not plotted. It is interesting to consider an array with baselines of one or two carrier cycles, as suggested by Xia [62], who also notes that very short baselines greatly reduce the problems associated with attitude integer resolution and array flexure. Short baselines limit the accuracy of any single attitude fix, however the gyros allow numerous earlier fixes to be weighted into the current filtered solution. Accuracy is enhanced by this averaging. In addition to reducing the problem of integer resolution, short baselines are not significantly influenced by array flexure, which becomes a limiting factor on large structures. The short baseline simulation indicates that attitude accuracy suitable for many applications is available with baselines of only a few carrier cycles. The stationary case (or any constant angular velocity case) is also interesting because the differential equations describing the errors become time invariant. The filter poles and steady state covariance performance are then readily found using MATLAB's *dlqe* command. This allows one to quantitatively study the impact of gyro noise and zero-point stability (gyro quality/cost) against expected improvements in steady state performance.

6.2 Bias estimation from GPS angular velocity

Instead of using attitude information to infer the vector of gyro biases, an alternative is to filter each axis independently against the equivalent component of the GPS determined angular velocity. As detailed in section 4.5, the GPS angular velocity signal is generated from the differential phase-rate outputs from the attitude receiver. The internal filter used to achieve the differentiation is necessarily a compromise between the resultant noise and phase lag. The effective frequency response of the filter employed, $H_2(e^{j\omega})$, is shown in figure 4.7.

The least squares angular velocity solution is *linear* in the measurements, $\Delta\dot{\phi}_{ij}$, each of which has been filtered by $H_2(e^{j\omega})$. The GPS angular velocity signal is therefore given by the sum of the *true* angular velocity filtered by $H_2(e^{j\omega})$ plus additive noise.

Ignoring gyro alignment and scale factor errors, one has three identical filters, one for each axis. For each axis, the objective is to resolve the unknown gyro bias given both the GPS and the gyro measurements. Since $H_2(e^{j\omega})$ is known, it is possible to account for the effects of the low level filter. In effect one is reversing the filtering that was achieved by the receiver tracking loops, and is the reason why signal processing is most appropriately applied at the most fundamental level, i.e., the raw phase measurements.

6.3 Flight Experiments

As an *acid* test of gyro aided attitude robustness, the aircraft was used to perform a series of barrel-rolls. These maneuvers were performed under pilot control during which GPS differential phase data was saved at 10 Hz to allow for post-flight evaluation. Since all GPS antennae were attached to the upper surface of the aircraft, simultaneous signal outages on all antennae for all satellites occurred during the period of inverted flight. Attitude integrity was lost shortly before the aircraft became completely inverted. Although the receiver was able to re-establish carrier lock after approximately 6 seconds, a further 16 seconds were required to collect and process sufficient data to complete integer reacquisition. Figure 6.9 shows the pitch, roll and yaw outputs from the unaided GPS attitude code during one such maneuver. Signal strength and attitude integrity was lost when the roll angle of the aircraft reached approximately 120 deg. The interval where the aircraft attitude is unknown is shown in the plot as blank.

Knowing the integers after reacquisition, the attitude code was run backward in time to discover the point where signal strength was again lost. The sum of the forward and backward time histories is plotted in figure 6.10. The blank area indicates the true signal outage experienced by the GPS receiver, and is sufficiently brief (approximately 6 sec) that gyro aiding is feasible. Figure 6.11 shows the number of differential phase measurements that were available for the attitude calculation over the interval.

Figure 6.12 shows the result of propagating the attitude during the interval when GPS measurements were not available (shown as a fine line on the plot). Using the integrated attitude estimate, integer reacquisition was attempted at $t = 2972.5s$. A total of 9 integers were initially determined using the technique outlined in section 4.4.3. The resolved values are known to be correct from the subsequent integer resolution step. The angular errors about pitch, roll and yaw axes at the time of integer reacquisition were [-1.5 -1.6 -2.5] deg respectively. In this case, the effects of scale factor, linearity and alignment errors were sufficiently small that they did not dominate the error budget over the outage interval. In figure 6.12, the integrated attitude estimate is plotted for approximately 19 seconds after receiver lock was reacquired. The integrated estimate may be compared with the attitude truth. The divergence gives a rough indication of the rate of error growth with flight data.

Although further testing is required and the scale factor and axis alignment estimation problems remain, this extreme example demonstrates the potential for inexpensive gyros to greatly increase the robustness of GPS attitude.

The ability to reacquire the attitude cycle ambiguities after a period of signal outage is related to the angular errors and the length of the baselines between antennae. Smaller baselines allow a larger angular error to accumulate before integer backout becomes infeasible. For example, a baseline length of 2 carrier cycles with a worst orientation would demand 1σ angular accuracy of only 4.7 deg to recover the integer successfully with $> 99\%$ probability. It is also feasible to explicitly search the integer space for the vector that minimizes the least squares residual when the baselines are small (or even the space near predicted, if the baselines are large).

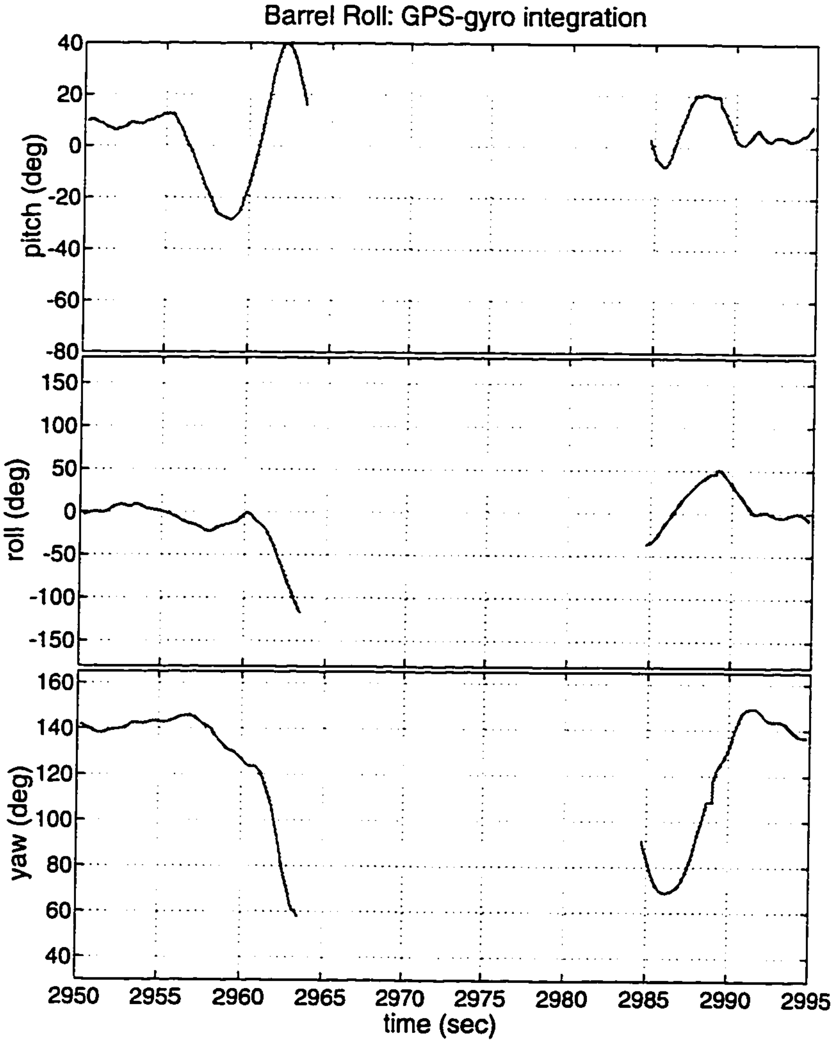


Figure 6.9: **Barrel Roll: Unaided GPS.**
Heading pitch and roll measurements from unaided GPS during a barrel roll.

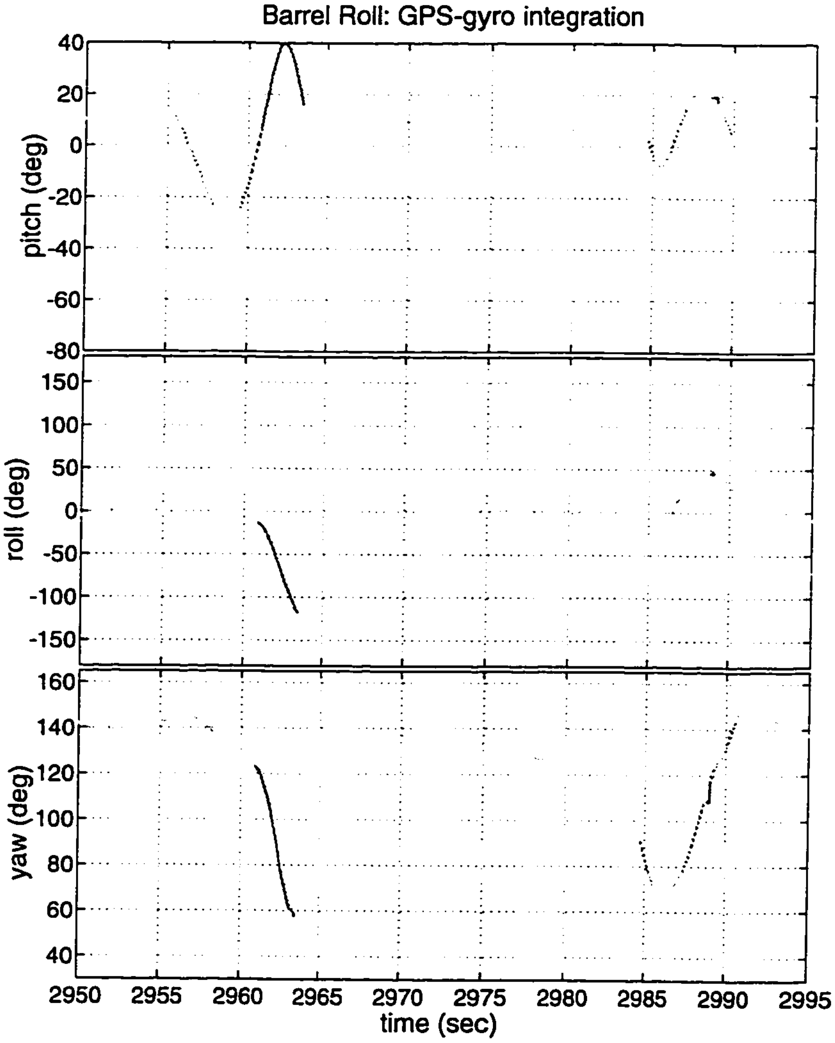


Figure 6.10: **Barrel Roll: True GPS Signal Outage.**
Post processed flight data. Sum of forward and backward data processing.

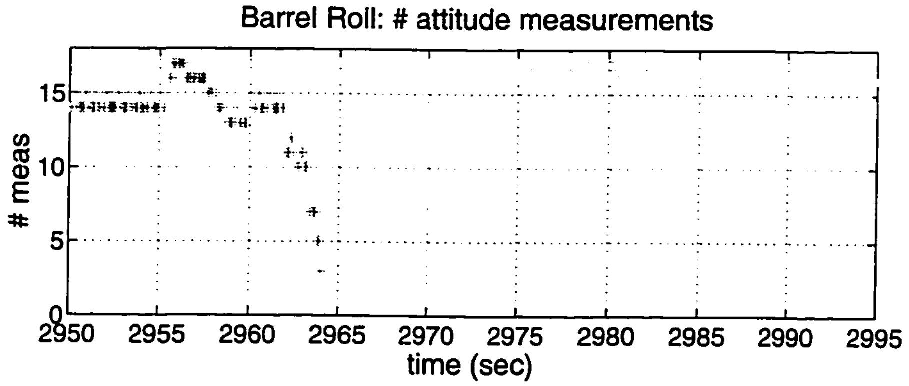


Figure 6.11: **Barrel Roll: Number of attitude measurements.** Post processed flight data. Sum of forward and backward data processing.

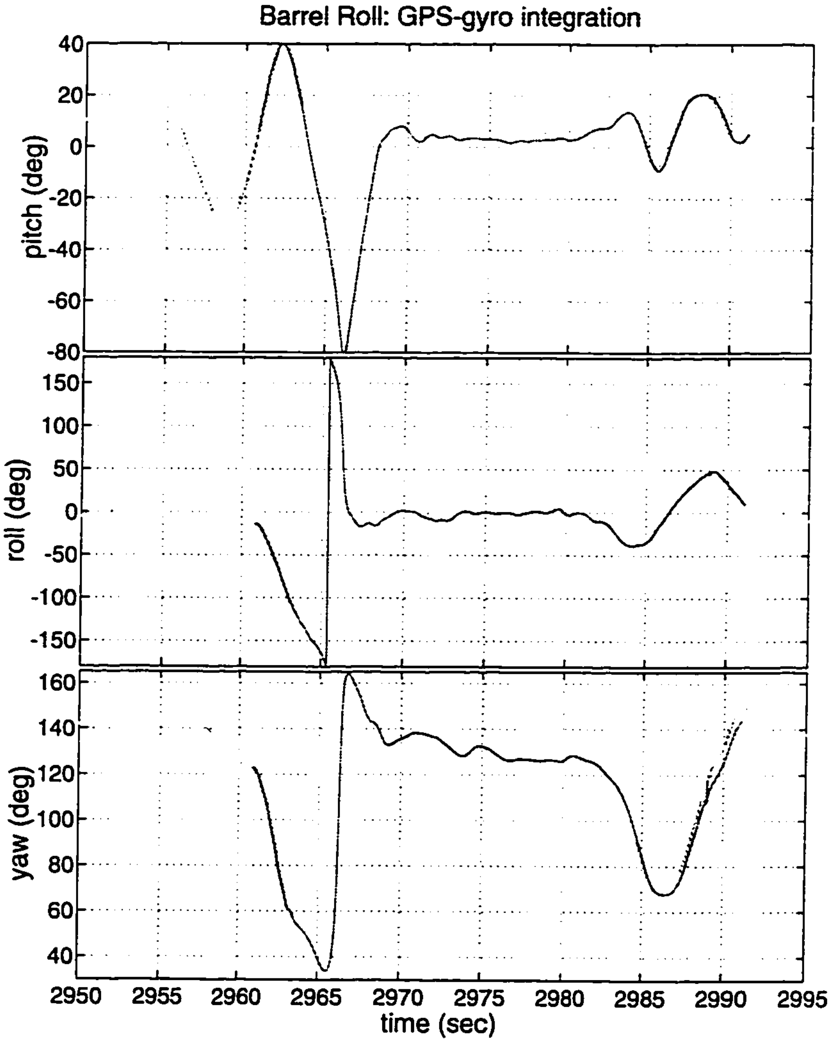


Figure 6.12: **Barrel Roll: GPS attitude with gyro aiding.** Heading pitch and roll measurements during a barrel roll.

Chapter 7

Conclusions and Future Work

This chapter highlights significant conclusions and details ideas for future study, particularly those that make sense on a model aircraft and which extend the ideas developed to date.

7.1 Conclusions

7.1.1 GPS as a sensor for automatic control

Demonstration of the thesis objective (automatic takeoff to landing) provides an "existence proof" to the utility of CDGPS as a sensor for automatic control. GPS estimates of position, velocity, attitude and angular velocity were used at 10 Hz, aided only by an airspeed sensor. Position and attitude measurements are inferred from differential carrier phase measurements directly. Velocity and angular velocity signals require differentiation of the fundamental receiver measurements and are consequently more noisy. They are, however, unbiased and can be usefully employed to provide damping (rate feedback) in automatic control. In cases where the system to be controlled is sensitive to phase lag, one must be careful to model the delays and latencies inherent in the GPS sensing system.

The bullets below highlight *major points* related to the integrated GPS sensor and control system that was assembled for this project. Some conclusion items are not original but are included to provide a more comprehensive overview. Appropriate citations are made in these cases.

Position and Velocity Sensing

Position

- CDGPS gives a position measurement relative to the reference antenna. As reported by Pervan [45], the relative position accuracy with a tracking loop bandwidth of 10 Hz is conservatively 10 mm, with spatial distribution as quantified by the position covariance matrix (DOP). This assumes the cycle integers are correctly resolved and four or more *common* satellites are simultaneously in track at both the reference and mobile receivers.

- The requirement for at least four common satellites between reference and mobile receivers is a limitation in practice. Aircraft banking occludes satellites. Bringing satellites online after an occlusion event requires integer patching, which can result in accumulation of position errors over time. With more channels and a receiver/antenna combination that can track satellites to a lower antenna relative elevation mask this *availability* limitation should be greatly reduced.

- Differential positioning requires robust wireless communications to provide differential correction information at a minimum of 1-2 Hz. This translates to a differential correction bandwidth of at least 1200 bps.

- With the largest (deterministic) error sources accounted for, position accuracy degrades with increasing separation between the reference and mobile receivers due to the accumulation of trophospheric and ionospheric errors. With the current system, a practical range for carrier differential positioning is on the order of 10 km based on related experimental work at Stanford [45] [29].

- In some cases, the requirements for position integer initialization can be relaxed, since position errors accumulate due to satellite motion, in proportion with the initial error. Where station keeping for a few minutes is required, approximate integer resolution (eg. from code differential) is probably acceptable.

- With more than the minimum of 4 common satellites, one can use the position residual as a metric for detection of *fault* conditions (RAIM) [45]. The effectiveness of this technique in detecting and isolating erroneous measurements is related to the number of redundant measurements available. Using additional sensors to aid the GPS data can add greater reliability (AAIM) [60].

Velocity

- The fundamental carrier signals produced by the receiver are related to displacement. To generate a velocity signal requires differentiation, which amplifies noise in proportional with its frequency content. For this reason, rate signals are necessarily more noisy than the related position signals, although they are unbiased.

- As with position, there is a tradeoff between tracking loop bandwidth and the resultant noise in the signal. With the tracking loop bandwidth set to 2 Hz, the resultant noise was found to be approximately 0.1 m/s RMS, with spatial distribution given by the position covariance matrix (DOP). This is approximately that which would be obtained by a first difference of the position estimate but avoids the sample delay.

- To minimize noise in the position and velocity estimates it is important to apply filtering at each receiver appropriate to the dynamics seen by its antenna. The reference receiver can typically be filtered much more heavily than the mobile receiver for this reason.

Attitude and Angular Velocity Sensing

Attitude

- GPS attitude determination is related to position in that it uses differential phase measurements. Similar to position, it requires the determination of the cycle ambiguities across each baseline. For integer resolution, earlier motion based algorithms have required three or more satellites to be observed across each baseline and the antenna array to be significantly non-planar, e.g. Cohen [11]. By demonstrating an algorithm without these constraints [34], this work has shown that the constraints are artifacts of the integer resolution algorithm rather than intrinsic to the problem itself.

- Attitude is inherently more robust than position due to the greater number of redundant measurements that are available at each instant. The occlusion of a single satellite is therefore less of a problem than with position determination.

- Ignoring flexure, Cohen [11] showed that (RMS) attitude errors scale inversely with baseline length. This is also true for angular velocity. The ultimate accuracy is limited by flexure in the structure and the extent to which this can be modeled and observed. For an aircraft, baseline perturbations due to the first symmetric mode (symmetric wing flexure) may be resolved, however higher order modes are not observable with the four antennae system used here.

- With tracking loop bandwidth set to 2 Hz, phase noise of approximately 10 mm is typical in the differential phase measurements. This results in attitude accuracy of approximately 0.2 deg with the (approximately 3 meter) baselines of our aircraft. The spatial distribution of the attitude error is given by the attitude covariance matrix (ADOP).

Angular Velocity

- As with velocity, GPS angular velocity requires time differentiation of the fundamental phase signal. Identical tradeoffs between bandwidth/latency and resultant noise apply. With the baselines of our aircraft, the resultant angular velocity signal has RMS noise of approximately 0.2 deg/s about each axis multiplied by ADOP.

- Unlike velocity which is independent of the position integers, the angular velocity signal requires knowledge of the attitude cycle ambiguities.

7.1.2 Integration with inexpensive inertial instruments

Chapter 6 discusses the use of inexpensive piezo electric gyros integrated with GPS attitude. The objective is to produce a system that combines the low frequency stability of GPS with the high frequency performance of inexpensive inertial instruments. Ultimately, one wishes to (quantitatively) understand the impact that factors such as GPS and gyro sample rate, gyro noise, gyro zero point stability, gyro scale factor stability, gyro alignment and the dynamics of the vehicle (frequency content of vehicle motion) have on the overall performance of the system. One area of particular interest is the rate at which attitude errors grow over time in the absence of GPS measurement updates. This is important since it provides a measure of the longest GPS outage that can be tolerated. The work presented in chapter 6 examines via simulation, a *test case* using simple noise models with values representative of the GPS receiver and rate gyros used in this work. These simulations ignore gyro scale factor errors and errors due to gyro axis misalignment. The test case examines one possible combination of GPS/gyro sample rates with a 4th order Runge-Kutta algorithm for performing kinematic integration and one possible algorithm for performing the GPS/gyro filtering.

Based on the simulations and the given assumptions, one can draw the following conclusions:

- It is feasible to calibrate rate gyro biases using either the GPS attitude estimate or by using low level GPS differential phase measurements directly. Simulation results indicate it is possible to estimate biases to better than 0.05 deg/s (1 sigma) given the assumed noise and gyro bias processes.

- Given updates at 10 Hz, the bias estimates converge to statistical steady state in approximately 10 seconds independent of the specific attitude history.

- Given the noise assumptions employed, removal of GPS measurement updates (simulating a GPS attitude failure), results in growth in the (1 sigma) attitude error covariance envelopes at a rate of approximately 0.07 deg/s about each axis. Propagating the attitude with this error growth rate suggests that approximately 9 seconds of "flywheeling" (attitude propagation in the absence of measurement update) is possible before an antenna array with 3 meter baselines[1] has accumulated sufficient error to preclude integer "back-out" with 3 sigma probability[2].

- In the absence of other error sources (e.g. biases, scale factor errors, alignment errors etc.) and independent of the specific angular velocity history, integrating angular velocity measurements with additive white noise results in attitude errors (body referenced attitude error covariance) that grow linearly with time in proportion with the noise variance. This *analytic* result is useful in understanding how gyro measurement noise impacts "flywheeling" duration.

- The inclusion of inexpensive gyros increases attitude *accuracy* relative to a solution using GPS data alone. The improvement is due to the weighted inclusion of current plus earlier GPS data in the filtered estimate. The observed accuracy improvement is more pronounced as the GPS attitude measurement becomes more noisy relative to the gyro measurements, i.e., the accuracy improvement is greater for arrays with shorter baselines. For the aircraft studied here, accuracy improvement by approximately a factor of two is indicated in simulation.

To summarize, one concludes that benefits of gyro integration with GPS attitude (relative to unaided GPS attitude) are:

1. *Improved robustness*

2. *Greater accuracy*

3. *Increased bandwidth*

The work described in chapter 6 falls in the category of "loose" inertial integration. Tight inertial integration is discussed in section 7.2.3 and uses inertial instruments at the level of the tracking loops within the receiver. This has the potential to generate

[1] Approximate baseline length of antenna array used on the aircraft.

[2] Based on a worst case satellite/baseline orientation.

phase rate estimates which are much cleaner (less noisy) than those derived from GPS measurements alone and to enhance receiver bandwidth greatly.

7.2 Future Work

7.2.1 Airplane/Helicopter Collaboration

The model helicopter experiment developed by Andrew Conway and the ARL, currently share similar GPS hardware and software with the autonomous aircraft described here. It is attractive to maintain this cooperation for many reasons.

- *Exchange of expertise.* Both the ARL and the GPS lab can benefit from the expertise and learn from the experiences of the other. The GPS lab should emulate the computer management and software engineering practices employed at the ARL. This would greatly help in documentation, sharing of software, maintenance and protection of shared computer resources and transfer of knowledge to new students.

- *Leverage.* As both labs are developing systems that require common core functionality, it is reasonable to leverage this investment to the maximum. By developing a flexible base system, improvements made by anyone can then benefit everyone.

- *Knowledge retention.* The more people who have studied and are knowledgeable about the system, the easier it is to educate new students and to retain the knowledge when students graduate.

Further development with the current system is possible, but it makes sense to integrate some fundamental changes into a second generation implementation at this juncture. There are two areas where it makes sense for the ARL and GPS groups to collaborate.

1. *Replace DOS with an RTOS.* Using DOS as a real time operating system (RTOS) has serious drawbacks. DOS is single tasking, generally non-reentrant and does

not give deterministic response or provide debugging support. Its sole advantage is low cost. These drawbacks greatly complicate the design of software where different tasks must coexist and share computer resources on a priority basis. Another major drawback is the fact that software errors can crash the whole system, making it very difficult to track down bugs since all context is lost. This is particularly true on an embedded system where there is no display or hard drive. These are strong arguments to port the software to a more robust real time operating system. The ARL has significant experience with VxWorks and will be migrating the helicopter software onto this platform. It would be advantageous to follow their lead in this direction. This will provide a much firmer base on which to build future enhancements, reduce the learning curve and maintain software commonality.

2. *Consolidate to a single receiver for all GPS functions.* Assuming that the TANS architecture will continue to be used in the near term, it makes sense to consolidate all GPS functions in a single receiver. This would save weight cost and power, reduce hardware complexity and I/O requirements and simplify timing and debugging issues. The TANS firmware should be modified to output a timestamped packet containing both absolute (master), and relative phase and phase rate data at 10 Hz. If carefully designed, this packet would be approximately 100 bytes long. This is within the serial capability of the current TANS DUART, and would not incur latency greater than is currently present.

The current aircraft architecture could be streamlined by removal of the TT7. Inexpensive PC-104 cards could be used to replace the required ADC and TPU functions. This would make data available to the 486 on interrupt rather than serially and would generally reduce complexity. It also makes sense to build real time video capability into the next generation aircraft. Since a video datalink will be required, downlink telemetry could be modulated onto the voice channel allowing removal of the telemetry modems. This would also greatly increase range[3] and enable remote

[3] Because the current system requires an RC pilot for safety, out of sight operations are precluded. Video would allow the RC pilot to assert manual control if required in out of sight operations.

piloting or out of sight operations if desired.

Airframe

The current airframe is fragile and complex to build and repair. An updated airframe designed for simplicity of manufacture and low cost could greatly reduce the workload associated with building and maintaining an aircraft. The proposed design has similar dimensions and aerodynamic performance to the current airplane but somewhat greater payload capacity and increased time aloft. A concept diagram of a possible design is shown in figure 7.1. The fuselage is an aluminium tube with a 2-stroke petrol engine hard mounted to the front end. The flying surfaces are of composite construction and come out of the mould almost finished, requiring only control surface actuation and GPS antennae to be added. The surfaces attach directly to the fuselage tube in a high wing conventional configuration. Control surfaces are standard but flaps are not used. The vertical and horizontal stabilizers are staggered, with the rudder also acting as a tailwheel to eliminate the complexity of the nosewheel construction. A generic pod, holding the avionics, is suspended below the fuselage tube and the wings. This pod is application specific depending on desired function. The objective of the design is a very inexpensive, robust and easily transportable airframe that can be successfully operated by an inexperienced operator with minimal airfield requirements.

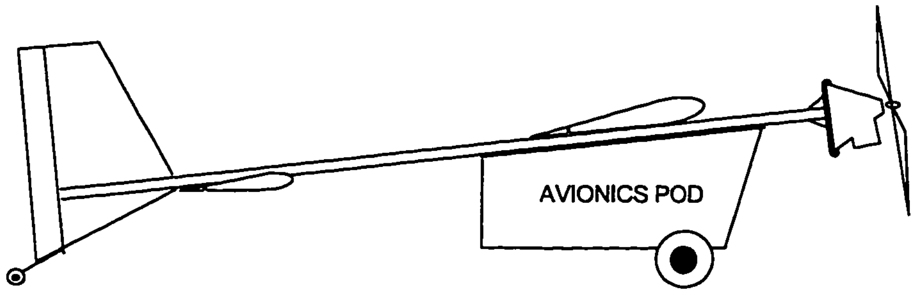


Figure 7.1: **Concept Airframe**

Ugly but functional airframe for a variety of airborne tasks.

7.2.2 GPS receiver for research and automatic control

As a tool for research and development and as a sensor for automatic control, the receivers used in this work were lacking in some respects. This section attempts to highlight these deficiencies from the perspective of a control engineer and to propose requirements for a future design.

Issues with the current receiver design.

- *Antenna multiplexing limits bandwidth, reduces robustness and adds complexity.* With 4 antennae enabled, the current receiver processes each antenna for one sixth of the time. Between samples, the phase estimator runs open loop. This reduces signal strength and is likely to cause the receiver to lose lock on low elevation satellites sooner than would otherwise occur, something that has been observed in practice. Furthermore, cycle slips are more likely to occur in dynamic environments than with a non-multiplexed architecture. Antenna multiplexing also adds a great deal of complexity to the receiver code. This is required to handle data bit transitions and multiple potential aliasing modes. These problems are compounded with each additional antenna, with special logic being required in some configurations but not others. The advantages of multiplexing are a minimum of hardware and a single RF path once the signal enters the receiver. The latter is perhaps the more significant, since crosstalk and RF path variability can cripple receiver performance.

- *CPU limitations.* The current CPU is "maxed out". The 68000 processor at the heart of the current receiver is heavily interrupted and close to 100% cycle utilization. To make the receiver work, the core signal processor code is carefully crafted assembly code that has been highly optimized and pared to a bare minimum. For example, the Kalman filters are implemented in approximately 10 lines of assembly code. This code is certainly fast, but it is not easy to read, understand or maintain. Enhancement or modification is exceedingly tedious, and portability to a non-68000 microprocessor is untenable. Usually, attempts to make other than minor changes will break the existing code or introduce new bugs. In short, the existing implementation is cycle constrained and highly fragile. In a research environment this hurts everyone. The learning time for a new participant is great, new code is difficult to debug and various forms of monitoring are not feasible as the associated code can change the timing sufficiently to cause erroneous results.

- *I/O limitations.* The current serial I/O implementation does not provide adequate bandwidth to access the correlator level of the hardware in real time. Such access is certainly desirable in many cases, since signal processing is most appropriately applied at the level of the fundamental observable. This begs a further question. What is the appropriate level of integration in a GPS receiver? The answer depends on what one wishes to achieve. In this work, the receivers were treated as sources of filtered phase data. This is one level removed from the raw (I and Q) observables. The receiver output is strobed from the low level Kalman filters at 10 Hz. At this level, the data still requires substantial processing to generate position and attitude estimates. Assuming that the GPS receiver is viewed as a generic sensor and not as the control processor, one is ultimately interested in getting data from the receiver into the control computer. It is desirable that this coupling be as direct and latency free as possible. For high rate and low latency, the current serial communications are inappropriate.

- *Black Box Problem* The black box problem refers to the fact one can only extract from the receiver that which the manufacturer has enabled and disclosed in the interface control document (ICD). Usually the details of internal signal processing are not available.

Requirements for a future receiver.

Most of the above shortcomings result from a design philosophy that views the receiver as a stand alone unit rather than as a component in a larger system. When contemplating the ideal receiver for automatic control, a broader "system" perspective is required.

The key to increasing the robustness and noise/bandwidth performance of GPS sensing is tight integration of *inexpensive* inertial instruments at the GPS carrier tracking level. To enable this next step, basic requirements are:

1. access to the fundamental (correlator) measurements in the receiver
2. interrupt driven, high bandwidth I/O
3. increased signal processing horsepower
4. a flexible and modular hardware architecture
5. a flexible and modular software architecture

Together, these create a foundation for a highly capable integrated position/attitude receiver.

Hardware Architecture. To eliminate multiplexing one must increase the number of channels if multiple antennae are desired. The objective is to eliminate the complexities and limitations associated with antenna multiplexing but retain the functionality provided by multiple distributed antennae. A receiver that looks something like the conceptual diagram illustrated in figure 7.2 is proposed. This diagram shows a GPS receiver which interfaces to a generic (VME or PCI) bus to provide a modular implementation of channels and antennae. This also facilitates migration to any bus compatible processor. The receiver motherboard consists of a synthesizer module which generates signals which are distributed among an arbitrary number of hardware channels. Hardware addresses are directly accessed by the CPU, which can also access other cards (sensors) on the bus. A design perturbation might place dedicated signal processing capability local to each channel or group of channels. This approach matches a dedicated set of channels with each active antenna.

Challenges associated with this hardware approach are issues related to line bias variation and interchannel crosstalk. A prototype implementation is required to determine viability in this respect.

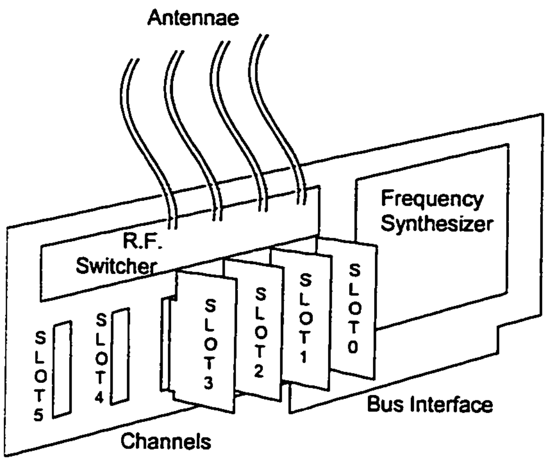


Figure 7.2: **Receiver Hardware Concept**

Modular, open architecture receiver hardware concept.

Software Architecture. Exposing the receiver hardware in this way eliminates the "black box" problem, but places the software burden on the user. Accepting this challenge, the problem becomes one of software engineering. The objective is to make use of the enhanced hardware architecture to make the process of *experimenting* with the receiver easy.

The requirements for the software design become

- provide a framework which allows a software product to be built/customized from well tested software components. This supports the hardware modularity being proposed.
- decouple software from hardware by abstracting the functionality and encapsulating hardware dependencies in device drivers
- make use of a commercial real time operating system and state of the art debugging tools
- use a high level language which is portable and self documenting
- provide a pathway by which application specific user enhancements can be easily incorporated in the software. This is similar to the toolbox philosophy used by

MathWorks.

The benefits of these steps are numerous, including

- reduced time to develop new products
- a higher quality and (FAA) certifiable product
- improved maintainability and reduced learning curve
- leverage users with application specific knowledge

Integrated position/attitude receiver. In the current work, separate receivers were used to generate data for attitude/angular velocity and data for position/velocity. With the proposed receiver design, position and attitude are integrated. This saves weight cost and power and reduces hardware complexity and I/O requirements while simplifying timing and interfacing issues.

The proposed receiver design enables enhancements such as tight inertial integration and roving antennae, both of which require an integrated attitude capability. With these facilities, one can conceive of a "strap-down" GPS receiver with multiple distributed antennae. When mounted on a highly dynamic aircraft, the *goal* is a sensor capable of delivering accurate and robust measurements of the aircraft position, velocity, attitude and angular velocity throughout aerobatic maneuvers.

Open vs. Closed Architecture. It is interesting to contemplate future trends in the satellite positioning industry. No doubt GPS receivers or their equivalent will become smaller and more capable, even as economies of scale drive prices down. The story is a familiar one in the computer industry and the same economics apply. As the technology matures, the availability of off-the-shelf GPS chipsets will transform the industry from the hardware dominated industry we see today to a software industry. Recent history suggests that majority market share will fall to the company that controls an "open" architecture. The open architecture will be in the form of a non-proprietary chipset that will allow users to enhance "vanilla" solutions in a manner that makes sense in their application domain. If a pathway exists to license and incorporate such application specific enhancements into the core product as software options, it may be possible to leverage the expertise of users in a highly synergistic manner.

7.2.3 Tight integration with inertial instruments

Current tracking loop design is based on a tradeoff between measurement and process noise, where these are assumed white and uncorrelated. The measurement noise is well understood, being largely attributable to thermal noise. The process noise, however, is entirely determined by the antenna array dynamics. It is not white or uncorrelated in a dynamic environment. As discussed in chapter 3, the current approach to estimator design is to set the phase estimator gains to achieve tracking loop bandwidth sufficient to track the expected dynamics. Insufficient bandwidth will result in phase lag and an increased probability of cycle slips. The cost of opening up the tracking bandwidth is increased noise in the resulting phase and phase-rate estimates.

To remove these limitations, it is necessary to inform the estimator about deterministic inputs. As discussed in section 3.1.2, it is undesirable to use a plant model to achieve this feedforward. An alternate approach is to use inertial sensors in a model replacement mode by *measuring* accelerations (caused either by deterministic control inputs or real process noise) and inferring phase accelerations from the measurements. With this approach, one is effectively creating a "strap down GPS receiver". The process noise is now given by the (hopefully well known) measurement noise characteristics of the inertial sensors employed. This in turn allows the estimator bandwidth and the resulting measurement noise to be greatly reduced. The sensors required for this purpose need not be of "inertial" or "navigation" quality, however sensor stability, noise characteristics and dynamic range are important factors.

For position, one approach is to measure accelerations as close as possible to the antenna phase center using a triad of inexpensive accelerometers inside the antenna enclosure. The latest generation of automotive accelerometers rendered in silicon (see appendix D), have performance and form factor that may be appropriate for this purpose. In principle, the accelerometer signals could be frequency separated and modulated on to the antenna coax to provide an integrated solution. Ignoring various error terms and noise, the measured acceleration $\boldsymbol{z}_{meas}$, is given by

$$\boldsymbol{z}_{meas} = \mathcal{R}(\boldsymbol{a} - \boldsymbol{g}) \tag{7.1}$$

where $\boldsymbol{a}$ is acceleration of the antenna in the ENU basis, $\mathcal{R}$ represents the rotation of the accelerometer package relative to ENU, and $\boldsymbol{g}$ is acceleration due to gravity. Taking the inner product with the unit LOS vector to satellite j gives the desired feedforward correction to the absolute phase tracking loop to account for antenna acceleration. This gives

$$\ddot{\phi}_j = \hat{\boldsymbol{s}}_j^T \boldsymbol{a} = \hat{\boldsymbol{s}}_j^T \left(\mathcal{R}^T \boldsymbol{z}_{meas} + \boldsymbol{g}\right) \tag{7.2}$$

where previously this term was modeled as white noise! Note that the attitude estimate is required and that gravity is a *large* disturbance that must always be removed.

For rigid arrays, an orthogonal triad of inexpensive gyros inside the receiver housing could be used to add a similar feedforward correction to phase-rate at the level of the differential phase estimator. This would greatly increase tracking loop performance in highly dynamic environments. The mathematics for attitude error propagation has been developed in chapter 6.

Although the potential for tight GPS/inertial integration is great, there are a number of potential problems and untested assumptions.

- Inexpensive inertial instruments have poor zero-point stability, particularly across temperature. Such biases would need to be estimated by adding bias elements to the estimator state.

- Scale factor nonlinearity or stability may be an issue.

- Sensor dynamic range must be adequate. Since accelerations at antennae can be large, the dynamic-range to noise tradeoff must be acceptable with the sensors employed.

- Sensor alignment and orthogonality of sensor axes is an important potential error source. Since these are unlikely to vary after installation, it is conceivable that a survey procedure similar to attitude self survey could be used to estimate these parameters after installation.

- Attitude must be accurately known to remove the large effect of gravity from accelerometer readings. Attitude noise of 0.5 deg RMS about each axis will result in acceleration noise terms on the order of 0.1 m/s^2. This implies that attitude uncertainty, and consequently baseline length would probably dominate the process noise budget.

7.2.4 Roving Master

The model aircraft is a suitable testbed for a roving antenna implementation. The aircraft can readily perform aerobatic maneuvers that will test the robustness of any such design. The objective is to utilize multiple antennae distributed over the aircraft to demonstrate continuous position and attitude tracking as individual antennae move in and out of occlusion during an aerobatic maneuver. Work to date relies on inexpensive gyros to propagate attitude for short periods when GPS signals are lost. Position integrity is entirely lost during such maneuvers. The proposed receiver architecture was conceived with the potential for a roving implementation in mind. It is reasonable to believe that this approach will allow centimeter accurate position and sub-degree accurate attitude to be maintained during highly dynamic aerobatic maneuvers. Interesting questions to consider in this context are:

- *What is the optimal number of antennae?*

- *What is the optimal placement and orientation of each antenna?*

- *What is an appropriate algorithm for selecting active antennae?*
  Candidate criteria are:

  - signal strength.
  - GDOP (ie. maximize geometric strength).
  - greatest number of satellites in view.

  - minimize antenna switching.
  - multipath mitigation.

- *Multipath mitigation.* Reflected signals can be very strong, particular over water. This can cause large errors even for carrier differential systems. The multipath issue has not been acute to date due to the use of upward facing antennae. With the potential to switch to an antenna that has strong gain in a downward facing direction the problem of multipath is greatly amplified.

The flexible hardware/software receiver design proposed above is intended to simplify the problem of antenna switching. Knowing the vehicle attitude, the antenna gain pattern and the satellite LOS, an algorithm can switch the R.F. input to an antenna which is certain to "see" a given satellite. To drive the hardware, a flexible software architecture is also required. One conceives of a software implementation where the concept of a "channel" is highly abstracted and a scheduling algorithm is used to map antenna/PRN combinations onto available channel resources.

7.2.5 System/Parameter Identification

The model aircraft is a suitable testbed for research related to system or parameter identification. One can program the flight computer to apply a deterministic input, designed to excite a specific mode or response. The response can be measured and accurately time-tagged. In addition to GPS data, one also has access to measurements from gyro, alpha, beta and pitot sensors. This should be adequate to identify an accurate model of the aircraft dynamics. The model may be fed back into the control synthesis process to increase controller performance or to look at alternative control methodologies. Optimal control theory could potentially be applied to demonstrate recovery to an airfield with engine loss or terminal guidance on landing [8].

Ultimately, real time parameter identification and adaptive control may be contemplated. Parameter identification is probably a more appropriate term than system identification for the aircraft, since a good deal is known about generic aircraft dynamics. A first cut might be to assume decoupled dynamics and attempt to refine the model discussed in section 2.7. In line with the discussion in section 7.2.2, one might conceive of automating the parameter identification process for aircraft (self-parameter-id), and supplying this capability as an optional software enhancement to a GPS receiver.

7.2.6 Wind Mapping and Downburst Detection

Wind mapping remains a very important area of ongoing research. As discussed in section 5.3 the instrumented aircraft provides the ability to measure the vector wind direction with unprecedented fidelity. This has great utility in meteorology, particularly if one is interested in measuring turbulence with sufficient resolution to estimate atmospheric mixing or transport phenomena. The Aerosonde program [2] is interested in utilizing GPS attitude and velocity measurements for this reason. Aerosondes have recently flown missions into thunderheads and weather fronts and sent back detailed meteorological data [33]. In flying near and beneath thundercloud activity, Aerosondes also have the potential to collect valuable downburst data [61]. These data are vital in developing systems that can detect the wind signature indicative of imminent downburst activity.

7.2.7 Takeoff at A and land at B

Current work has focused on achieving takeoff and landing at the same airfield. This has been accomplished using a differential system that provides measurements in airport relative coordinates. An ambitious extension would be to attempt a landing on a naval airfield at sea. The navy and coastguard are both interested in deploying UAV's for numerous purposes from naval platforms. The proposed takeoff and landing area is often very confined, with existing helicopter landing platforms being an obvious candidate. The problem is further complicated by turbulence in the vicinity of the landing area, particularly for vehicles with low wing loading. Ship motion is an additional challenge, although aircraft and ship dynamics have sufficient frequency separation that it should be possible to estimate the important ship dynamics and account for them in the control system design. Carrier differential position measurement facilitates this, being a direct relative measurement between the shipborne reference antenna and the aircraft antenna.

It is likely that local area differential integrity will be lost during a long flight far from the takeoff point. For in-flight navigation and return to base, standard GPS code or code differential systems are adequate, but carrier positioning is required to attempt any sort of precision landing. A practical approach would be to overfly a pair of pseudolites to regain cycle integers prior to landing. On a ship, pseudolites could be placed at the bow and stern. The aircraft would overfly the ship to resolve cycle ambiguities and then perform a 270 degree turn to bring it onto final. A similar approach would be practical for landing at a conventional airfield. As a demonstration of integrated WAAS/LAAS systems at Stanford, the model aircraft could autonomously takeoff from Shoreline Ampitheater carpark, fly across San Francisco Bay, a distance of approximately 12 miles and land at SACRC airfield in Fremont. These locations are also appropriate because the range is within the current airplane capability and the route does not require flight over built up areas.

A further observation is that the feedback control laws for flight stabilization do not require accurate position data. Most of the aircraft states are velocity components for which the cycle ambiguities are not required. For en-route flight, code-differential or even raw GPS position is adequate in many applications. Attitude, angular velocity and velocity components are still available even when the position integers required for accurate positioning are not. Even for takeoff, accurate position is not required as described in section 4.1.4.

7.2.8 Quantify GPS Angular Velocity Performance

GPS measurement of angular velocity needs to be compared with a navigation quality inertial measurement unit (IMU) to quantify its performance. The GPS lab has recently acquired such a unit. Initial comparison with crude rate gyros provided sufficient data to indicate that GPS angular velocity might be useful for automatic control. A serious attempt to quantify bias, noise, latency and drift characteristics was not attempted but is necessary.

With a flexible antenna array, one must be careful to define what is meant by "angular velocity". The angular velocity measured at the fuselage will differ from the angular velocity of a rigid wireframe connecting antennae mounted on flexible wings or appendages. The first (symmetric) mode may be accounted for with the addition of a symmetric wing flexure state. The second (anti-symmetric) mode is unobservable by definition. Aileron inputs that would excite such modes may be expected to result in disparity between the INU and GPS measurements so pitch and yaw perturbations are probably more appropriate for this experiment.

7.2.9 Aerial Photography

Integration of accurate position and attitude information enable accurate photography. At an altitude of 100 m, an attitude accuracy of 0.5 deg translates to a positional uncertainty on the order of 1 m. Referred to a standard coordinate frame, photographic data at this level of accuracy is of great utility in many applications. To generate a large database of significant geographic or man-made features, aerial photography coupled with feature recognition software is likely to be the most cost effective approach. The prospect of using aerial photography from a UAV, coupled with GPS controlled farm machinery has exciting potential application in precision farming.

7.2.10 Aircraft Autonomous Integrity Monitoring (AAIM)

Work by Pervan [36], describes fault detection and isolation when the position solution is overdetermined. Using the left nullspace of the observation matrix (*parity space*), he shows it is possible to unambiguously identify and remove single channel failures. The underlying assumption is that single channel failure modes are uncorrelated, so multiple simultaneous failures are extremely unlikely. This may be true, but it is unproven. A potential consequence of multiple simultaneous failures under this scheme is detection and removal of a *good* measurement, with potentially catastrophic results. For example, with 6 measurements and faults on two channels resulting in the removal of a good measurement, three good and two corrupt measurements remain. The resulting solution is in error and the residual may be small. In an aircraft where there are many additional sources of information, dependence on RAIM alone is unnecessary. Greater integrity may be obtained with *aircraft autonomous integrity monitoring* (AAIM). The model aircraft provides barometric altitude data in addition to GPS data. This signal alone, if integrated with RAIM provides a minimal AAIM implementation that can detect multiple failures. Additional signals such as those from an onboard IMU can further reduce the probability of integrity events.

Appendix A

Extended Kalman Filter Summary

A brief summary of the major assumptions and results for a sequentially linearized Kalman filter, taken from chapter 6 of [19] are given below for reference. We are interested in a general state equation of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + g(\boldsymbol{x}(t), t)\, \boldsymbol{w}(t) \tag{A.1}$$

where $\boldsymbol{x}(t)$ is the state vector and $\boldsymbol{w}(t)$, the process noise, is a Gaussian white-noise process with mean and covariance as given by

$$E\{\boldsymbol{w}(t)\} = \boldsymbol{0} \qquad \text{and} \qquad E\{\boldsymbol{w}(t)\, \boldsymbol{w}^T(t')\} = Q(t)\, \delta(t - t') \tag{A.2}$$

The initial mean and covariance of the state vector are given by

$$E\{\boldsymbol{x}(t_0)\} \equiv \hat{\boldsymbol{x}}(t_0) = \boldsymbol{x}_0 \qquad \text{and} \qquad E\{(\boldsymbol{x}(t_0) - \boldsymbol{x}_0)\, (\boldsymbol{x}(t_0) - \boldsymbol{x}_0)^T\} \equiv P(t_0) = P_0 \tag{A.3}$$

Given the initial conditions on the state vector and the state covariance matrix, the minimum-variance estimate of the state vector at a future time is, in the absence of measurements given by the conditional expectation

$$\hat{\boldsymbol{x}}(t) = E\{\boldsymbol{x}(t) \mid \hat{\boldsymbol{x}}(t_0) = \boldsymbol{x}_0\} \tag{A.4}$$

Note that the ($\hat{}$) notation in this context is used to indicate the minimum variance estimate, not a unit vector. This predicted estimate satisfies the differential equation

$$\dot{\hat{\boldsymbol{x}}}(t) = E\{\boldsymbol{f}(\boldsymbol{x}(t), t)\} \equiv \hat{\boldsymbol{f}}(\boldsymbol{x}(t), t) \tag{A.5}$$

which is written approximately as

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t), t) \tag{A.6}$$

This expression may be formally integrated to give

$$\hat{\boldsymbol{x}}(t) = \phi(t, \hat{\boldsymbol{x}}(t_0), t_0) \tag{A.7}$$

The state-error vector $\boldsymbol{\Delta x}(t)$ and the state-error covariance matrix $P(t)$ are defined as

$$\boldsymbol{\Delta x}(t) \equiv \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \qquad \text{and} \qquad P(t) \equiv E\{\boldsymbol{\Delta x}(t)\, \boldsymbol{\Delta x}^T(t)\} \tag{A.8}$$

Neglecting terms higher than first order in the state vector and process noise, the state-error vector satisfies the differential equation

$$\boldsymbol{\Delta}\dot{\boldsymbol{x}}(t) = F(t)\, \boldsymbol{\Delta x}(t) + G(t)\, \boldsymbol{w}(t) \tag{A.9}$$

where

$$F(t) \equiv \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, t) \,|_{\hat{\boldsymbol{x}}(t)} \qquad \text{and} \qquad G(t) \equiv g(\hat{\boldsymbol{x}}(t), t) \tag{A.10}$$

Equation (A.9) may be integrated formally to give

$$\boldsymbol{\Delta x}(t) = \Phi(t, t_0)\, \boldsymbol{\Delta x}(t_0) + \int_{t_0}^{t} \Phi(t, t')\, G(t')\, \boldsymbol{w}(t')\, dt' \tag{A.11}$$

where $\Phi(t, t_0)$ is the transition matrix, which satisfies

$$\frac{\partial}{\partial t} \Phi(t, t_0) = F(t)\, \Phi(t, t_0) \qquad \text{and} \qquad \Phi(t_0, t_0) = I \tag{A.12}$$

The predicted covariance matrix satisfies the continuous Riccati equation

$$\dot{P}(t) = F(t)\, P(t) + P(t)\, F^T(t) + G(t)\, Q(t)\, G^T(t) \tag{A.13}$$

which may be integrated to give

$$P(t) = \Phi(t, t_0)\, P(t_0)\, \Phi^T(t, t_0) + \int_{t_0}^{t} \Phi(t, t')\, G(t')\, Q(t')\, G^T(t')\, \Phi^T(t, t')\, dt' \tag{A.14}$$

That equation (A.14) is indeed a solution to equation (A.13) may be verified by direct substitution and use of Lebnitz's rule for differentiating an integral. Discretizing the above equations to form the prediction-step update equations over a time step, one has, in obvious notation

$$\hat{\boldsymbol{x}}_{k+1}(-) = \phi(t_{k+1}, \hat{\boldsymbol{x}}_k(+), t_k) \tag{A.15}$$

$$P_{k+1}(-) = \Phi_k \, P_k(+) \, \Phi_k^T + N_k \tag{A.16}$$

where $\Phi_k = \Phi(t_{k+1}, t_k)$ and N_k is the integral term on the right side of equation (A.14). This term can be expensive to evaluate numerically.

Measurement Update. The measurement vector at time t_k is related to the state vector by

$$\boldsymbol{s}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{\nu}_k \tag{A.17}$$

where $\boldsymbol{\nu}_k$, the measurement noise, is a discrete Gaussian white-noise process, with

$$E\{\boldsymbol{\nu}_k\} = \mathbf{0} \qquad \text{and} \qquad E\{\boldsymbol{\nu}_k \, \boldsymbol{\nu}_{k'}^T\} = R_k \, \delta_{kk'} \tag{A.18}$$

The optimal (*minimum-variance*), estimate of $\boldsymbol{x}_k$ immediately following a measurement update is given by the covariance form of the Kalman filter equations

$$\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + K_k \left[\boldsymbol{z}_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k(-)\right)\right] \tag{A.19}$$

where the Kalman gain matrix K_k is given by

$$K_k = P_k(-) \, H_k^T \left[H_k \, P_k(-) \, H_k^T + R_k\right]^{-1} \tag{A.20}$$

and the measurement sensitivity matrix H_k is given by

$$H_k = \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} \big|_{\hat{\boldsymbol{x}}_k(-)} \tag{A.21}$$

The covariance matrix immediately after measurement update is given by

$$\begin{aligned} P_k(+) &= (I - K_k \, H_k) \; P_k(-) & \text{(A.22)} \\ &= (I - K_k \, H_k) \; P_k(-) \; (I - K_k \, H_k)^T + K_k \, R_k \, K_k^T & \text{(A.23)} \end{aligned}$$

Appendix B

Minimizing equation 4.105

One knows that $B \geq 0$ from the way B is determined in practice[1]; and one can clearly assume that ϕ is between 0 and 2π. If $\phi > \pi$ one can make the transformations $x \to -x$ and $\phi \to 2\pi - \phi$ and one ends up solving the same equation form except with ϕ between 0 and π, so one can assume without loss of generality that $0 \leq \phi \leq \pi$. Furthermore, if $\phi > \frac{\pi}{2}$, we can make the further substitutions $x \to \pi - x$ and $\phi \to \pi - \phi$ leaving the same form of equation again, except with $0 \leq \phi \leq \frac{\pi}{2}$.

If B is zero, then there are two identically correct solutions : $x = \pm\frac{\pi}{2}$. Assume hereafter that $B > 0$. So now the problem is reduced to solving equation (4.105) for $B > 0$ and $0 \leq \phi \leq \frac{\pi}{2}$.

It can be shown through straightforward mathematics that there is then a solution in the range $(\frac{\pi}{2}, \pi - \phi)$, and that this can be reliably obtained by the bisection algorithm.

The bisection algorithm is reliable but slow to converge, and speed is very important as this equation has to be solved many times. For almost all values of B and ϕ, a solution can be obtained using Newton one dimensional iteration from a starting point obtained via a table lookup (with a mapping like $B/(1+B)$ to convert the infinite range of B values to a convenient range for the look up table). Due to the nasty behaviour of the solution near $B = 4$ and $\phi = 0$ (where three of the turning points of equation 4.105 merge), Newton iteration does not work reliably here. This

[1] In any case one could change the sign of B via adding π to ϕ.

is not a problem, as there exists a reliable alternative algorithm, the bisection algorithm, which can be used if the Newton iteration fails. Logged experimental data indicates that this happens rarely, so that this does not reduce the average speed of the algorithm significantly.

Away from the point $B = 4$ and $\phi = 0$, the solution of equation 4.105 is a smooth function of B and ϕ, so the table look-up does not have to be large. In practice, two separate table look ups were used in order to treat the portion of the $B - \phi$ space near $B = 4$, $\phi = 0$ in greater detail. With a table look-up occupying a few kilobytes of memory, typically only 1 or 2 Newton iteration steps are needed to get accuracy far greater than needed, making this process very fast.

In summary, it is possible to find the minimum of equation (4.105) in a computationally efficient manner.

Appendix C

GPS/Gyro Simulation Code

The following is the matlab v4.2 code that was used to generate the simulations given in section 6.1.5.

```
% ekf3.m                                     P. Montgomery April 1996
% .m file to calculate attitude KF from simulated data, assuming GPS
% attitude solutions are available at 10 hz
% The angular velocity is sampled at 20 Hz, and a 4th order
% Runge-Kutta step is used to propagate the attitude.
% The angular velocity at the intermediate time step is used for the
% transition matrix update and the process noise addition calculation.
% This also distributes the computational load between the timesteps.
% The integrated attitude is used to calculate the GPS differential
% phase history given assumed SV directions and baseline array vectors.
% The attitude covariance matrix is calculated for each time step
% so its covariance can be used in the kalman filter with correct
% noise added to the GPS measurements.
% The biases have an initial offset and are drifting at rms 1 deg/s
% over 10 min.
clear
load integav.1
T=0.05;                                % sample rate is 20 hz
T2=2*T;                                % covariance T.U. at 2*T
d2r=pi/180;
N=1:length(integav);                   % number of samples to filter
t=integav(N,1);
w=[integav(N,2)'; integav(N,3)'; integav(N,4)'];
q=[integav(N,5)'; integav(N,6)'; integav(N,7)'; integav(N,8)'];
clear integav

% calculate the gps phase information phi_ij
% for each time step ignoring ints
PHI=[]; R=[]; m_noise=[];
```

```
 phi_sig=0.05;                              % phase noise of 5% ~= 0.9 cm
 B=[5 12 5;-7 0 7;3 6 3];                   % make up some baseline vectors
 S=[0  1  1 -1 -1 -1 ;
    0  1 -1  1  1  0 ;
   -1 -1 -1 -2 -2 -2];                      % make up some LOS vectors
 for i=1:6,
 S(:,i)=S(:,i)/norm(S(:,i)); end;           % make unit LOS vectors
 for i=1:2:length(N),                       % measurements at i=1,3,5 etc
  A=ep2dc(q(:,i)');
  PHI(:,i*6-5:i*6)=B'*A*S;
  H=[S'*A'*cross(B(:,1)); S'*A'*cross(B(:,2)); S'*A'*cross(B(:,3))];
  R=0.25*phi_sig^2*((H'*H)\eye(3));
  % note: 0.25 is necessary, since the measurement noise and the
  % measurement update is in delta_eta, not Delta_xi
  % where delta_eta=0.5*Delta_xi
  Rgps(:,i*3-2:i*3)=R;
  m_noise(:,i)=sqrtm(R)*randn(3,1);         % add ang. noise to the GPS soln
 end;

 bias=[1 -1 0.4]'*d2r;                      % initial gyro biases in rad/s
 ramp=d2r*2*sqrt(3)*(rand(3,1)-0.5);        % random ramp rates
                                            % (max 1 deg/s over 10 min)
 I3=eye(3); O3=zeros(size(I3));
 p_nRMS=0.1*d2r;                            % gyro rate noise RMS (rad/s)
 p_noise=p_nRMS*randn(3,length(N));         % gyro rate noise samples
 Q1=I3*p_nRMS^2;                            % gyro noise covariance matrix
 % since bias drift is modeled as a random walk, the covariance
 % grows linearly with time.
 % assuming the biases may drift by 1 deg/s over 10 minutes with 1 sigma
 % probability, then P(t)=q*(600)=1^2 => q=0.001667: gelb eqn 3.8-11
 % use cvrtq() to find the amplitude of noise at sample times
 % to simulate this
 b_nRMS=d2r*sqrt(cvrtq(0,1,0.001667,T2));
 % b_noise=b_nRMS*randn(3,length(N));       % bias process noise samples
 Q2=I3*b_nRMS^2;                            % bias process covariance matrix

 % initialization for just after measurement update @ t=0
 eta=q(:,1);
 b=[0;0;0];
 Qplus=eta;
 Qminus=eta;
 P=[I3 O3; O3 I3];                          % initial uncertainty is large
 Pplus=P;
 Pminus=P;
 Bplus=b;
 Tplus=t(1);
 AngErrPlus=zeros(3,1);
 k1=T*xi(eta)*(w(:,1)+bias+p_noise(:,1)-b);
```

```
for i=2:length(N)-1,                         % run KF!
ieven=rem(i+1,2);
bias=bias+ramp*T/600;                        % model bias drift

% Time Update : all quantities are (+) here
w_meas=w(:,i)+bias+p_noise(:,i);             % what the gyro measures
if (~ieven),
 k4=T*xi(eta+k3)*(w_meas-b);
 eta=eta+(k1+2*k2+2*k3+k4)/6;                % 4th order Runge-Kutta update
 % save (-) state
 Qminus=[Qminus eta];                        % save (-) attitude estimate
 Pminus=[Pminus P];                          % save (-) covariance estimate
else
 k2=T*xi(eta+k1/2)*(w_meas-b);
 k3=T*xi(eta+k2/2)*(w_meas-b);
 % calculate the transition matrix
 A= I3-T2*cross(w_meas-b);                   % A(t_k+1,t_k) ~= I+[[D_theta]]
 K=-T2*(8*I3+5*A-A')/24;                     % K(t_k+1,t_k)
 Phi=[A K;O3 I3];                            % transition matrix
 % calculate the process noise contribution
 N11=T2*((4*I3+A-A')*Q1+2*A*Q1*A')/48 ...
 + T2^3*((30*I3+11*A-7*A')*Q2+6*A*Q2*A')/960;
 N11=N11+N11';
 N12=-T2*T2*(10*I3+3*A-A')*Q2/48;
 N22=T2*Q2;
 P=Phi*P*Phi'+[N11 N12;N12' N22];            % perform the covariance T.U.
end % if-else T.U.

% Measurement Update : all quantities are (-) here
if (~ieven),                                 % M.U. @ 10 hz: i=1,3,5 etc
 Xi43=xi(eta);                               % form Xi(eta(-))
 P_deta=P(1:3,1:3);
 R=Rgps(:,i*3-2:i*3);                        % get GPS meas. cov. matrix
 K=[P_deta;P(4:6,1:3)]/(P_deta+R);           % calc meas. update matrix
 Delta_x=K*(Xi43'*q(:,i)+m_noise(:,i));      % add  meas. noise in body frame
 deta=[Delta_x(1:3);1];                      % form perturbation quaternion
 eta=qml(deta)*eta;                          % perform attitude state update
 eta=eta/norm(eta);                          % normalize attitude estimate
 b=b+Delta_x(4:6);                           % update bias vector estimate
 P=(eye(6)-[K zeros(6,3)])*P;                % covariance measurement update
 k1=T*xi(eta)*(w_meas-b);
 % save (+) state
 Qplus=[Qplus eta];                          % save (+) attitude estimate
 Bplus=[Bplus b-bias];                       % save (+) bias error estimate
 Pplus=[Pplus P];                            % save (+) covariance estimate
 Xi43=xi(q(:,i));
 AngErrPlus=[AngErrPlus 2*Xi43'*eta];        % save (+) angular err hist(rad)
```

```
  Tplus=[Tplus; t(i)];                 % save measurement update times
 end; % measurement update

 end; % EKF


% clean up
clear N11 N12 N22 A K Phi P deta eta b Delta_x Xi43
clear P_deta I3 O3 N Q1 Q2 R T T2
clear b_nRMS b_noise w_meas d2r i k1 k2 k3 k4 ieven k H phi_sig

% plot some stuff out
plot_ek3;

%%%%%%%%%% end ekf3.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% plot_ek3.m
% to be run after ekf3.m to plot out stuff in the workspace
% this will plot the k'th diagonal element of the (+) and (-) covariance
% matrices in figure(k)
% the sigma locus at (+) and (-) are plotted and the error innovations
% history is also plotted.  The batch GPS 1 sigma bounds are also plotted

r2d=180/pi;
rootopts; % set up root defaults
ylabels=['roll err (deg)  ';'pitch err (deg) ';'yaw err (deg)   ';
         'bias err (deg/s)'];
titles=['Error Covariance : roll axis attitude and gyro bias errors ';
        'Error Covariance : pitch axis attitude and gyro bias errors';
        'Error Covariance : yaw axis attitude and gyro bias errors  '];

for k=1:3,
 figure(k);
 subplot(211);
 % 1. Plot the covariance bounds
 j=k:6:length(Pminus);
 tmp=r2d*2*sqrt(Pminus(k,j));
 plot(Tplus,tmp,'c5-',Tplus,-tmp,'c5-'); grid; hold on;
 j=k:6:length(Pplus);
 tmp=r2d*2*sqrt(Pplus(k,j));
 plot(Tplus,tmp,'c7-',Tplus,-tmp,'c7-');
 j=k:6:length(Rgps);
 plot(t(1:2:length(t)),2*r2d*sqrt(Rgps(k,j)),'g-');
 h=plot(Tplus,r2d*AngErrPlus(k,:),'c7-');
 set(h,'linewidth',0.5);
 hold off;
 axis([0 16 -0.25 0.25]);
 set(gca,'position',[0.13 0.52 0.77 0.41],'xticklabels',[]);
```

```
  title(titles(k,1:59));
  ylabel(ylabels(k,1:15));

  subplot(212);
  k=k+3;
  j=k:6:length(Pminus);
  tmp=r2d*sqrt(Pminus(k,j));
  plot(Tplus,tmp,'c5-',Tplus,-tmp,'c5-'); grid; hold on;
  j=k:6:length(Pplus);
  tmp=r2d*sqrt(Pplus(k,j));
  plot(Tplus,tmp,'c7-',Tplus,-tmp,'c7-');
  h=plot(Tplus,r2d*Bplus(k-3,:),'c7-');
  set(h,'linewidth',0.5);
  hold off;
  axis([0 16 -0.2 0.2]);
  set(gca,'position',[0.13 0.10 0.77 0.41]);
  xlabel('time (s)'); ylabel(ylabels(4,1:16));
  set(gcf,'paperunits','inches','paperposition',[0.25 1.75 8.0 9.75]);
  figure(k-3);
end;

clear j k r2d tmp h

%%%%%%%%%% end plot_ek3.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Xi43] = xi(q)
% [Xi43] = xi(q)
% return the first 3 columns of the right-quaternion
% matrix given a quaternion
Xi43=[q(4) -q(3)  q(2);
      q(3)  q(4) -q(1);
     -q(2)  q(1)  q(4);
     -q(1) -q(2) -q(3)];

%%%%%%%%%% end xi.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [qm] = qml(q)
% [qm] = qml(q)
% return the left quaternion matrix given a quaternion
qm=[q(4)  q(3) -q(2)  q(1);
   -q(3)  q(4)  q(1)  q(2);
    q(2) -q(1)  q(4)  q(3);
   -q(1) -q(2) -q(3)  q(4)];

%%%%%%%%%% end qml.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [W,Phi]=cvrtq(F,Ga,Q,Ts)
% [W,Phi]=cvrtq(F,Ga,Q,Ts)      : typed PYM 950615
% Determine covariance W of full rank random sequence wd(k)
% that is equivalent to white noise w(t) having sqectral
% density Q and distribution matrix Ga, with sampling
% period Ts, i.e. xdot = F*x + Ga*w is equivalent to
% x(k+1) = Phi*x(k) + wd(k) @ t = k*Ts
% W=integral[Phi(t)*Ga*Q*Ga'*Phi'(t)] dt over 0 -> Ts
[ns,nd]=size(Ga);
S=[-F,Ga*Q*Ga';zeros(ns),F'];
C=expm(S*Ts);
G2=C([1:ns],[ns+1:2*ns]);
F3=C([ns+1:2*ns],[ns+1:2*ns]);
Phi=F3';
W1=F3'*G2;
W=(W1+W1')/2;

%%%%%%%%%% end cvrtq.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function dc=ep2dc(q)
% syntax :   dc=ep2dc(q)
% function returns dc,  an orthogonal direction cosine
% matrix given a vector of euler parameters q=[x y z w]
% where x,y,z define the vector about which
% rotation occurs, and w defines the rotation amount.
% q is a row vector of euler parameters or quaternions
% of assumed unity norm.

s=2/(q*q');
qs=q*s;
wx=q(4)*qs(1);wy=q(4)*qs(2);wz=q(4)*qs(3);
xx=q(1)*qs(1);xy=q(1)*qs(2);xz=q(1)*qs(3);
yy=q(2)*qs(2);yz=q(2)*qs(3);zz=q(3)*qs(3);
dc(1,1)=1-(yy+zz);
dc(1,2)=xy-wz;
dc(1,3)=xz+wy;
dc(2,1)=xy+wz;
dc(2,2)=1-(xx+zz);
dc(2,3)=yz-wx;
dc(3,1)=xz-wy;
dc(3,2)=yz+wx;
dc(3,3)=1-(xx+yy);
dc=dc';

%%%%%%%%%% end ep2dc.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function omega=cross(w);
% syntax : omega=cross(w);
% provides vector cross product or skew symmetric omega matrix;
% returns 3x3 matrix omega, given a 3 vector w;
% b = w (cross) a : can be achieved by  b=cross(w)*a;
% a is a 3x1 vector;
% d/dt(DC) = (DC) * cross(w);
% where DC is a direction cosine matrix

for i=1:3 omega(i,i)=0.0; end;
omega(1,2)=-w(3);
omega(1,3)= w(2);
omega(2,1)= w(3);
omega(2,3)=-w(1);
omega(3,1)=-w(2);
omega(3,2)= w(1);

%%%%%%%%%% end cross.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Appendix D

Accelerometer Spec. Sheet

ANALOG DEVICES

±1g to ±5g Single Chip Accelerometer with Signal Conditioning

PRELIMINARY

ADXL05*

FEATURES

- 5 milli-g Resolution
- Noise Level 12X Less Than the ADXL50
- User Selectable Full Scale From ±1g to ±5g
- Output Scale Selectable from 200mV/g to 1V/g
- Complete Acceleration Measurement System On a Single Chip IC
- Self Test on Digital Command
- +5V Single Supply Operation
- 1000g Shock Survival

GENERAL DESCRIPTION

The ADXL05 is a complete acceleration measurement system on a single monolithic IC. The ADXL05 will measure accelerations with full-scale ranges of ±5g to ±1g or less. Typical noise floor is 500ug√Hz, (12X less than the ADXL50), allowing signals below 5 milli-g to be resolved. The ADXL05 is a force balanced capacitive accelerometer with the capability to measure both ac accelerations, (typical of vibration) or dc accelerations, (such as inertial force or gravity).

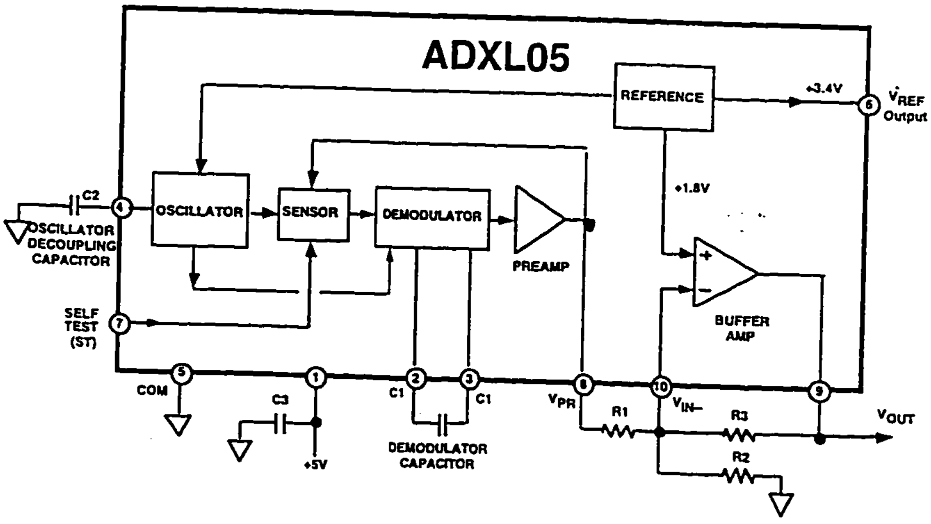
Three external capacitors and a +5 volt regulated power supply are all that is required to measure accelerations up to ±5 g. Three resistors are used to configure the output buffer amplifier to set scale factors from 200mV/g to 1V/g. External capacitors may be added to the resistor network to provide 1 or 2 poles of filtering. No additional active components are required to interface directly to most analog to digital converters (ADCs).

The ADXL05 uses a capacitive measurement scheme. The analog output voltage is directly proportional to acceleration, and is fully scaled, referenced and temperature compensated, resulting in high accuracy and linearity over a wide temperature range. The device features a TTL compatible self-test function which can electrostatically deflect the sensor beam at any time to verify that the sensor and its electronics are functioning correctly.

The ADXL05 is available in a hermetic 10-pin TO-100 metal can, specified over the 0°C to +70°C commercial, and -40°C to +85°C industrial temperature ranges. Contact factory for availability of devices specified for operation over the -40°C to +105°C automotive and -55°C to +125°C military temperature ranges.

APPLICATIONS

- Low cost sensor for vibration measurement
- Tilt Sensing with faster response than electrolytic or mercury sensors.
- More sensitive alarms and motion detectors
- Affordable inertial sensing of velocity and position



Bibliography

[1] H. Ashley, *Analysis of Flight Vehicles*, Dover, New York, 1974.

[2] G. J. Holland B. McGeer, "Small Autonomous Aircraft For Economical Oceanographic Observations on a Wide Scale," *OCEANOGRAPHY*, Vol. 6, 1993, no. 3, pp. 129–135.

[3] B. W. Parkinson B. S. Pervan, C. E. Cohen, "Integrity Monitoring for Precision Approach using Kinematic GPS and a Ground Based Pseudolite," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 41, Summer 1994, no. 2.

[4] B. W. Parkinson B. W. Pervan, C. E. Cohen, "Integrity in Cycle Ambiguity Resolution for GPS-Based Precision Landing," *Presented DSNS 1994, April 18-24, London*, 1994.

[5] R. Paielli B. McNally R. Bach and D. Warner, "Carrier Phase Differential GPS for Approach and Landing: Algorithms and Preliminary Results," *ION GPS-93, Salt Lake City, Utah*, September 1993.

[6] I. Bar-Itzhack, "The Piogram: An Easy Tool for Angular Position and Rate Computations," *Journal of Astronautical Sciences*, Vol. 41, 1993, no. 4.

[7] V. Bernatets, "Longitudinal Control of a Model Airplane Using GPS Signals", Visiting Scholar Summer Research Report, September 1994, Stanford University Dept. Aeronautics and Astronautics.

[8] A. E. Bryson, *Optimal Estimation: E207C Classroom Notes*, Stanford University, Dept. Aeronautics and Astronautics, 1992.

[9] ———, *Control of Spacecraft and Aircraft*, Hemisphere, New York, 1995.

[10] B.W. Parkinson C.E. Cohen and B. McNally, "Flight Tests of Attitude Determination Using GPS Compared Against and Inertial Navigation Unit," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 41, Spring 1994, no. 1.

[11] C. E. Cohen, "Attitude Determination using GPS," Ph.D. Thesis, Stanford University, 1992.

[12] A. R. Conway, "Autonomous Control of a Model Helicopter using Carrier Phase GPS," Ph.D. Thesis, Stanford University, 1995.

[13] C. Counselman and I. Shapiro, "Miniature Interferometer Terminals for Earth Surveying," *Bulletin Geodesique*, Vol. 53, 1979, no. 2, pp. 139–163.

[14] R.F. Stengel D. Atzhorn, "Design and Flight Test of a Lateral-Directional Command Augmentation System," *Journal of Guidance and Control*, Vol. 7, 1984, no. 3, pp. 361–368.

[15] A. Brown F. Van Diggelen and C. LaBerge, "Test Results of a GPS/Pseudolite Precision Approach and Landing System," *ION GPS-93, Salt Lake City, Utah*, September 1993.

[16] S. D. Wertz (editor), *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, Dordrecht, 1978.

[17] G. Purcell et al., "Measurement of Aircraft Position, Velocity and Attitude using Rogue GPS Receivers," *Fifth International Geodetic Symposium on Satellite Positioning, Las Cruces, New Mexico*, March 1989.

[18] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 2nd ed., Addison-Wesley, Reading, MA, 1990.

[19] A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

[20] J.D. Powell H. Hirata, "Sample Rate Effects on Disturbance Rejection for Digital Control Systems," *ACC, San Diego, Ca.*, 1990.

[21] P. Katz J. D. Powell, "Sample Rate Selection for Aircraft Digital Control," *AIAA Journal*, Vol. 13, August 1975, no. 8.

[22] G. Romrell R. Brown G. Johnson and D. Kaufman, "FAA/FEDSIM E-Systems Cat IIIB Feasibility Demonstration Flight Test Preliminary Results," *ION GPS-95, Palm Springs, Ca*, September 1995.

[23] K. Joseph and P. Deem, "Precision Orientation: A New GPS Application," *International Telemetering Conference, San Diego, Ca.*, October 1983.

[24] J.M. Johnson Jr., "Digital Filters Using Observers Applied to ICBM Control System Design," *Journal of Spacecraft*, Vol. 11, 1974, no. 7, pp. 498–504.

[25] T. Kailath, *Linear Systems*, 2nd ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1988.

[26] T. R. Kane, *Dynamics: Theory and Applications*, McGraw Hill, 1985.

[27] D. Klein and B. Parkinson, "The Use of Pseudo-Satellites for Improving GPS Performance," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 31, Winter 1984-85, no. 4.

[28] I. Kroo, Notes from Stanford Class AA-200A. Stanford University, September 1992.

[29] D. G. Lawrence, "Aircraft Landing Using the Global Positioning System," Ph.D. Thesis, Stanford University, 1996.

[30] P. MacDoran, "Satellite Emission Radio Interferometric Earth Surveying (SERIES) - GPS Geodetic System," *Bulletin Geodesique*, Vol. 53, 1979, no. 2, pp. 117–138.

[31] A. Masson, "Prediction of Reference station GPS Carrier Phase Measurements," *AA290 Project Report, Stanford University Dept. Aeronautics and Astronautics*, 1994.

[32] B. McGeer, "Avionics Specification for the Phase I Aerosonde," *Insitu Whitepaper*, 1994, (Insitu Group, internet: tmcg@netcom.com).

[33] ———, "Meteorological Instrumentation for the Aerosonde Autonomous Sounding Aircraft," *Final tech report under SBIR N00014-95-C-0106*, 1996, (Insitu Group, internet: tmcg@netcom.com).

[34] A. Conway P. Montgomery R. Cannon B. Parkinson, "A New Motion Based Algorithm for GPS Attitude Integer Resolution," *NAVIGATION. Journal of the Institute of Navigation*, Summer, 1996, (not yet published).

[35] B. Parkinson and K. Fitzgibbon, "Aircraft Automatic Landing Systems using GPS," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 42, Summer 1989, no. 1.

[36] B.S. Pervan D.G. Lawrence C.E. Cohen B.W. Parkinson, "Parity Space Methods for Autonomous Fault Detection and Exclusion using GPS Carrier Phase," *Presented at IEEE PLANS-96, Atlanta, GA*, 1996.

[37] C.E. Cohen B.W. Pervan H.S. Cobb D. Lawrence A.K. Barrows J.D. Powell B.W. Parkinson, "Flight Test Results of Autocoupled Approaches using GPS Integrity Beacons," *Presented ION GPS-94, Salt Lake City UT*, September,1994.

[38] C.E. Cohen B.W. Pervan H.S. Cobb D. Lawrence J.D. Powell B.W. Parkinson, "Real-Time Cycle Ambiguity Resolution using a Pseudolite for Precision Landing of Aircraft with GPS," *DSNS 1993, Amsterdam*, 1993.

[39] ———, "Real-Time Flight Test Evaluation of the GPS Marker Beacon Concept for Category III Kinematic GPS Precision Landing," *ION GPS-93, Salt Lake City, Utah*, 1993.

[40] ———, "Real-Time Flight Testing Using Integrity Beacons for GPS Category III Precision Landing," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 41, 1993, no. 2.

[41] ———, "Achieving Required Navigation Performance using GNSS for Category III Precision Landing," *Presented DSNS 1994, April 18-24, London*, 1994.

[42] ———, "Preliminary Results of Category III Precision Landing with 110 Automatic Landings of a United Boeing 737 using GNSS Integrity Beacons," *Presented ION NTM 1995, Anaheim Ca.*, 1995.

[43] Ping-Ya Ko J. Powell P. Enge B. Parkinson, "Continuity Improvements via Inertial Augmentation of GPS-Based Landing System," *PLANS, Atlanta Georgia*, April 22-26 1996.

[44] P.Y. Montgomery H. Uematsu B.W. Parkinson, "Analysis of Angular Velocity Determination using GPS," *Proceedings, ION GPS 1994, Salt Lake City*, 1994.

[45] B. S. Pervan, "Navigation Integrity for Aircraft Precision Landing using the Global Position System," Ph.D. Thesis, Stanford University, 1996, Sudaar 677.

[46] P. Ward R. Brown, "A GPS receiver with built-in precision pointing capability," *IEEE PLAN Symposium Proceedings*, Vol. March, 1990, pp. 83–93.

[47] B. Remondi, "Performing Centimeter-level Surveys in Seconds with GPS Carrier Phase: Initial Results," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 40, Winter 1985-86, no. 4, pp. 139–163.

[48] R. Rusch, "The Market and Proposed Systems for Satellite Communications," *Applied Microwave and Wireless*, Vol. Fall, 1995, pp. 10–34.

[49] B. McNally D. Warner Jr. D. Hegarty T. Schultz, "Flight Test Evaluation of Precision-Code Differential GPS for Terminal Approach and Landing," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 39, Summer 1992, no. 2.

[50] E. Lefferts F. Markley M. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," *Presented at AIAA 20th Aerospace Sciences Meeting, Orlando, Florida*, 1982.

[51] M. D. Shuster, "A Survey of Attitude Representations," *Journal of Astronautical Sciences*, Vol. 41, 1993, no. 4.

[52] J. Spilker, *Digital Communications by Satellite*, Prentice Hall, New Jersey, 1977.

[53] ———, "GPS Signal Structure and Performance Characteristics," *Navigation, Vol 1, pp. 29-54*, 1980.

[54] V. Spinney, "Applications of GPS as an Attitude Reference for Near Earth Uses," *Institute of Navigation. Warminster, PA.*, April 1976.

[55] J. C. Stupelnagel, "On the Parameterization of the Three-Dimensional Rotation Group," *SIAM Review*, Vol. 6, 1964.

[56] H. Sun, "Integration of INS with Multiple GPS Antennas for Airborne Applications," *Student Paper, ION GPS-94, Salt Lake City*, September 1994.

[57] G. Schanzer T. Jacob, "Integrated Flight Guidance System using Differential GPS for Landing Approach Guidance," *AGARD Guidance and Control Panel, Lissabon*, 1989.

[58] M. Dieroff T. Jacob, "Integrated Navigation for Approach Guidance using Differential GPS," *Proceedings, ION GPS-90, Colorado Springs, September*, 1990.

[59] F. van Graas and M. Braasch, "GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 38, Fall 1991.

[60] S. Vieweg, "Using Low Cost Inertial Sensors for Integrated Satellite-Inertial Navigation," *Proceedings ION GPS 94*, 1994.

[61] R. C. Wingrove and R.E. Bach, "Severe Winds in the Dallas Fort-Worth Microburst Measured from Two Aircraft," *Proceedings, AIAA GNC conference 1987, Monterey, Ca*, 1987.

[62] Ren Xia, "Short Baselines for GPS Attitude Determination, Stanford University Aeronautics and Astronautics Affilitates Presentation, April 1996.

[63] K. Zimmerman, "Experiments in the Use of the Global Positioning System for Space Vehicle Rendezvous," Ph.D. Thesis, Stanford University, 1995.

[64] K. Zimmerman and R. Cannon, "Differential Carrier Phase GPS Techniques for Space Vehicle Rendezvous," *Proceedings ION GPS-95, Palm Springs, Ca*, 1995.