Avionics Systems Development for Small Unmanned Aircraft

by

Vladislav Gavrilets

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

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998

Author .....
Department of Aeronautics and Astronautics
May 22, 1998

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

Accepted by .....
Jaime Peraire
Chairman, Department Committee on Graduate Students

Avionics Systems Development for Small Unmanned Aircraft

by

Vladislav Gavrilets

Submitted to the Department of Aeronautics and Astronautics
on May 22, 1998, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The avionics systems for two small unmanned aerial vehicles (UAVs) are considered from the point of view of hardware selection, navigation and control algorithm design, and software development. Some common challenges for many small UAV systems are addressed, including gust disturbance rejection at low speeds, control power, and systems integration. A rapid prototyping simulation framework which grew out of these efforts is described. A number of navigation, attitude determination and control algorithms are suggested for use in specific applications.

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

Acknowledgments

The work described in this thesis was a result of team effort. Here I would like to thank people who contributed to both projects described in the thesis, and otherwise provided support during my two years at MIT.

I would like to thank my advisor Professor John J. Deyst for giving me the opportunity to work on an open-ended, challenging and rewarding project. Under his guidance I was able to get real world engineering experience.

Professor Mark Drela greatly contributed to the aerodynamic modeling and analysis of the WASP flyer. Professor James Paduano helped with the analysis and design of the control augmentation system. I greatly appreciate Professor David L. Trumper's help with the vibration isolation problem. The value of the knowledge I acquired from all the above members of MIT faculty through classes and consultations is very hard to overestimate.

Draper Lab engineers Brent Appleby and John Plump provided simulation framework for the WASP project. Studying the experience of the DSAAV team, lead by Paul DeBitetto, provided an excellent source of empirical data and design guidelines.

I would like to thank Tan Trinh, Torrey Radcliffe, Jean-Marc Hauss, Cory Hallam, Josh Bernstein and other students who worked on the WASP team in 1996-98.

My thanks to MIT Aerial Robotics team. Confidence and expertise of Scott Rasmussen and Jeremy Brown made this very complicated project a reality.

I would like to thank graduate students at the MIT IC&E lab. Stimulating discussions with Jerry, Emilio, Eric and Arkadiy frequently helped me find a way to solve a problem.

Sergei, Jim and Akan were very supportive friends and roommates during my two years at MIT.

My special thanks to my family in Bishkek, Kyrgyzstan, for their caring letters, and sincere support in my endeavours.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Applications for Unmanned Aircraft

Unmanned aircraft are becoming increasingly important for both civilian and military applications. A number of situations exist in which human presence onboard the aircraft is either not necessary, too dangerous, or expensive.

Remotely piloted vehicles have been used for military purposes since the 1940s. Recent advances in sensors, communications, propulsion, flight dynamics and control algorithms have greatly increased the levels of autonomy that can be given to these vehicles. One can think of unmanned autonomous aircraft as a blend of two dreams of designers: the old dream of making flying machines, and the new dream of building smart robots.

The history of unmanned aircraft started with the Israeli Aircraft Industries decoy aircraft, used to confuse enemy anti-aircraft radars in 1972. A number of tactical surveillance aircraft from different countries followed. Currently, large long-endurance aircraft are being developed for reconnaissance, for atmospheric research and for aerial photography. There are a number of proposals for an unmanned tactical aircraft (UTA), which may replace manned combat aerial vehicles in the future.

There are also a number of applications for small unmanned aircraft, which are

the size of a hobby remote control (RC) aircraft and smaller. These include the micro UAV - a palm size surveillance aircraft that provides situation awareness for a soldier. Other applications include: traffic monitoring, search, and surveillance in the areas that are dangerous or difficult to access.

It is currently estimated [9] that there will be a $3.9 billion market for 7,900 UAVs in the next five years. The demand for unmanned aircraft will grow with the improvements in technology and, hence, reliability, of UAVs.

1.1.2 Avionics Challenges Specific to Small UAVs

There are a number of factors that are crucial for small unmanned aircraft. They can be loosely broken down into four categories: propulsion, light-weight structures, aerodynamic performance, and avionics system design and integration. The last of these includes sensors, communications, algorithms and software problems, which are considered here in application to two small UAVs.

Small aircraft fly at low Reynolds numbers, and thus it is difficult to achieve high lift to drag ratios. A number of efficient computational fluid dynamics (CFD) tools were developed that study flow behavior at this regime. Flying qualities of small aircraft are usually a serious issue due to lack of control power and vulnerability to gusts. The lack of control power arises mainly because of volume and weight constraints for control surfaces. Many designs need stability augmentation systems, and this requires high bandwidth sensor data. The flying qualities problem is coupled with the problem of avionics system design. Combining physical insights gained from new aerodynamic modeling techniques with modern control and estimation theory, can significantly aid in the design of adequate control systems using limited sensor data.

Experience has shown that the design and integration of avionics systems is one of the most complicated aspects of unmanned aircraft design. This is a problem with stringent constraints on size, weight and power consumption. The limited space available makes vibration from propulsion systems and electromagnetic interference adverse to sensor performance. Cost is a determining factor, because many small

vehicles are designed to be disposable. However, the choice of sensors, interfaces and software development practices can have a significant impact on cost. Proper use of rapid prototyping tools for software engineering and algorithm design can significantly reduce development time.

1.2 Thesis Overview

The objectives of this thesis are to summarize the development of avionics systems and some navigation and control algorithms used for two small UAVs, and to state lessons learned in the process. The field of small UAVs is very new, and the goal of this effort is to add to the common knowledge base in the area.

Specifically, Chapter 2 describes in detail the avionics system and navigation algorithms for a small unmanned helicopter. The helicopter was built as an entry in the International Aerial Robotics Competition by an MIT student team. The design of the system is based on designs made in previous years by MIT/Draper Laboratory teams, with certain changes aimed at improving performance. The thesis attempts to structure the process of hardware selection and system integration for a small UAV.

Chapter 3 describes an avionics system, model development, control and attitude determination algorithms, as well as a simulation framework for another small UAV. This vehicle is a gun launched aircraft, which is used to provide surveillance information using onboard sensors. This work has been done as a part of the MIT/Draper Technology Development Partnership Project.

Chapter 2

Avionics Architecture for a Small Unmanned Helicopter

2.1 Aerial Robotics Competition

The small unmanned helicopter system, described in this chapter, was developed as the MIT entry in the 1997 International Aerial Robotics Competition. This contest dates back to 1991. The rules of the contest change from year to year, becoming ever more challenging. However the main goal is to develop autonomous aerial robots, capable of performing complex missions without human intervention. In 1997 the competition goal was to develop a fully autonomous VTOL aircraft, that could perform search and removal of hazardous objects from a designated region. The vehicle had to take off and land autonomously, find the locations of black barrels marked with "radioactive" and "biohazardous" signs, and find and retrieve a small orange metal disc. The 1997 MIT team took second place in the contest with the helicopter based platform, shown in Figure 2-1.

The team demonstrated a fully integrated hardware system, communication, telemetry and navigation capability. The 1997 design was based on the winning design of the Draper Laboratory/MIT team of 1996. The lessons from the 1997 effort are being used in ongoing design of the 1998 MIT entry.

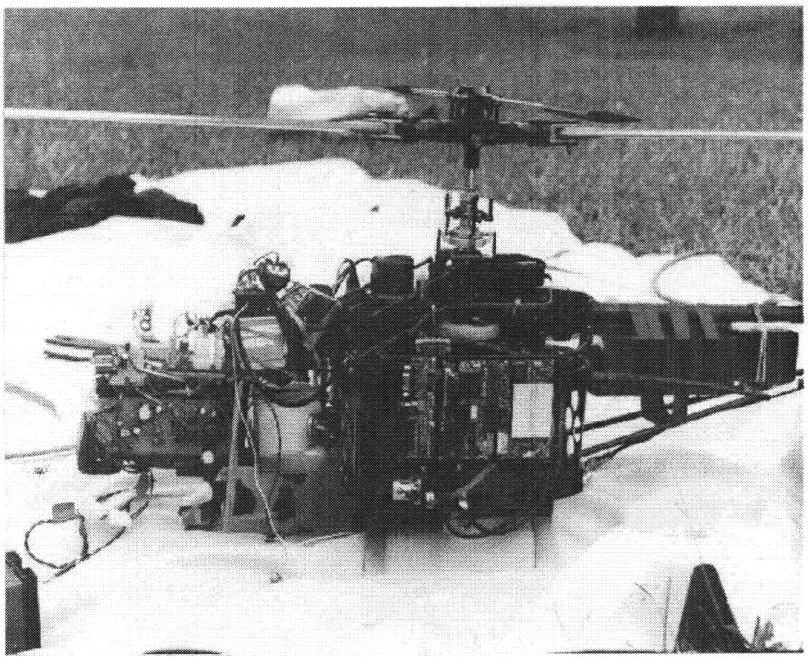

Figure 2-1: Robocopter on test field in Orlando, FLa

2.2 Hardware Architecture

2.2.1 Driving Requirements

The choice of hardware for an aircraft is determined by its mission requirements. These are derived from high-level goal(s), which in this case is to score the highest number of points in the competition. The important derived requirements are:

- an image detection and recognition system for specified targets
- navigation accuracy of 0.3 m Spherical Error Probable (SEP)
- attitude and heading accuracy of less than 3 degrees
- stable hover and forward flight
- autonomous takeoff and landing
- guidance system allowing for vision system inputs

- telemetry and monitoring system

- redundant safety system

2.2.2 Overview of Subsystems

Aerial Vehicle

The aerial platform chosen for the system was a Japanese TSK Blackstar helicopter. The TSK is a small 15 lb empty weight, 9 lb payolad, hobby helicopter, powered by a 2-cycle internal combustion engine. The payload capability of TSK was found to be marginal for the mission, and in 1998 a larger helicopter from the Bergen Machine Tool company was chosen.

Navigation Sensors

The navigation sensors included an inertial measurement unit, differential GPS, magnetic flux compass and sonar altimeter. The design of the sensor subsystem is covered in detail in the next section.

Vision System

The onboard vision system consisted of the Cognachrome vision board connected to an RGB (red-green-blue) CCD camera. The board had custom software that was trained to recognize specific colors. Since disc retrieval was a major contributor in the scoring system, the vision system was designed specifically for finding and tracking orange color.

Power System

The power system used high efficiency switching regulators. Rechargeable NiCd batteries were used to supply electrical power.

Remote Control and Safety System

Safety of the system, for the contestants and spectators, was assured by an automatic/manual switch, that would allow a remote pilot to take over the computer control in case of emergency. This switch was also used for test flights in manual mode when the navigation and telemetry systems were being tested. The auto/manual switch is one of the functions of the remote control receiver interface. The interface is a custom designed hardware element, developed by Draper Laboratory. It is based on a field programmable gate array (FPGA), and is wired to a Pulse Position Modulation (PPM) receiver. The interface provides two other important functions. It writes pilot inputs to the onboard computer via a standard RS-232C serial protocol, and reads control system commands from the computer, and converts them to servo commands. An important feature of the interface is its ability to operate in a pass-through mode, in which pilot commands are passed through the computer to the servos. This allows tuning independent control loops separately. In this case the pilot controls some channels, and computer controls the others. A separate RC receiver with its own battery is used to choke the engine in emergency situations.

Onboard Processing

The onboard computing power is provided by a 486DX2 50 MHz CPU with 8 Mb RAM and two RS-232C serial ports. The CPU is part of a PC104 stack with an ISA bus. The PC-104 is a standard format for hardware cards, ranging from CPUs to power supplies to low-cost GPS receivers. The cards are 90x96x26 mm, and can be attached to either a common PC-104 bus or the ISA bus. Other cards in the stack included a card with 4 RS-232C serial ports and an Ethernet card with the so called boot ROM. The latter is used to boot the CPU from a ground computer and upload the operating system (OS) and onboard executable code. The use of a hard drive for OS storage and data gathering on a small helicopter is unreliable because the vibration level is very high.

Several improvements to the stack hardware are suggested. The system would be

easier to use if PC104 videocard was used. This would allow the use of a monitor with the CPU to get a direct access to the onboard program without using a ground station display. Another improvement would be to use a PC104 standard power supply, which provides internal regulation for all cards in the stack. This prevents occasional CPU rebooting during spikes in the onboard power lines.

Communications Link

A two way communication link is provided by Proxim Proxlink radio modems, operating in the 900 Mhz frequency range. The modems have a range of several hundred feet. They have a serial interface with internal checking for lost packets of data. The modems are used for uploading differential GPS corrections, for uplink of guidance commands and operator requests, and for downlink of telemetry information.

Ground Station

The ground station consists of a base GPS receiver, Pentium 100 MHz laptop with 8 Mb RAM and a radio modem.

Signal Flow Diagram

The signal flow diagram is given in Fig 2-2. The important feature of the architecture is use of only serial interfaces by the CPU. Each sensor requiring a more complicated interface, is sampled by a lower level processor. The IMU is sampled by a fixed point Motorola 68332 processor. The sensor interface was non-serial digital, and required an external clock rate signal at 1 MHz, which would be impossible to provide with the CPU. Basic Stamp was used to sample sonar altimeter and magnetic compass. Basic stamp is a chip which can be programmed with a simple set of commands in the Basic programming language. This architecture can be called distributed. It has an advantage with respect to a centralized architecture, that uses the CPU to sample sensors directly, because the CPU computing power is not required to perform input sampling.

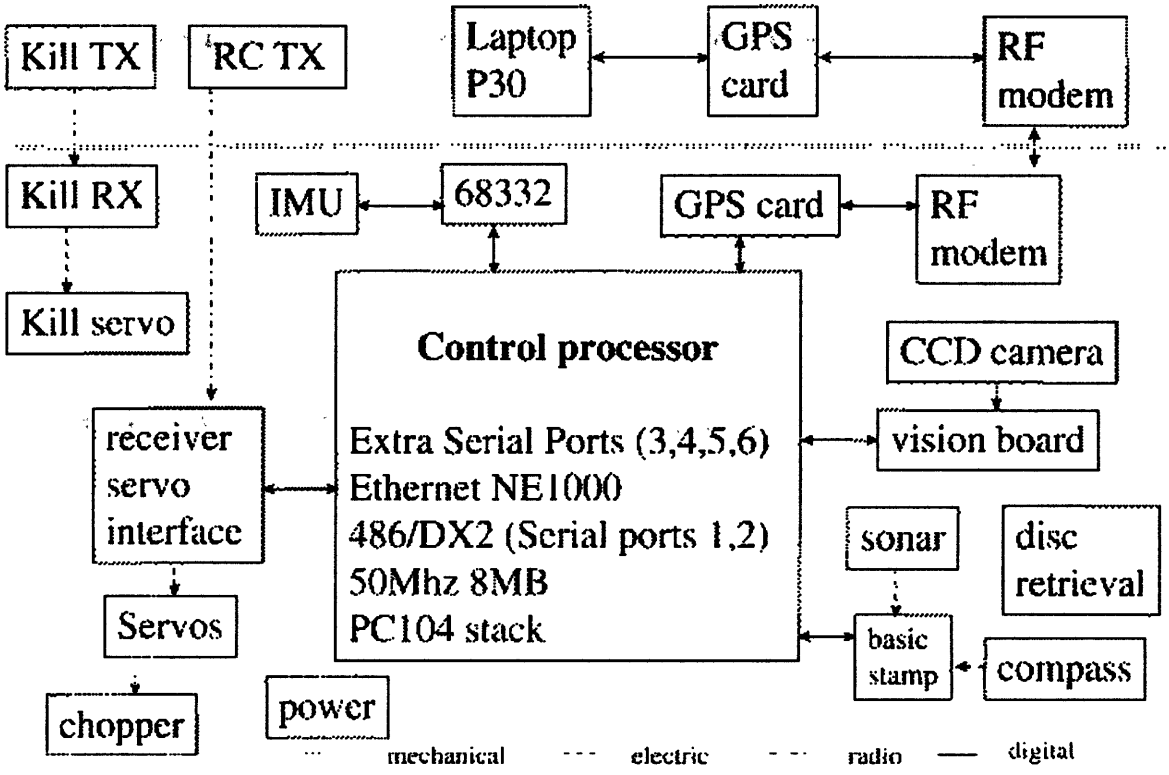

Figure 2-2: Signal flow diagram for the 1997 helicopter

Both ground based and onboard GPS receivers have two serial ports, and uplink and downlink commands are passed through from one port to another to insure the line

$$\text{laptop} < - > \text{GPS} < - > \text{modem} < - > \text{modem} < - > \text{GPS} < - > \text{CPU}$$

2.2.3 Selection Criteria for Navigation Sensors

The designer of an avionics system must go through a set of trade offs in the selection process. The main considerations, common for all sensors are:

- performance requirements
- power requirements
- weight
- size, shape and volume
- cost
- ability to operate in the environment onboard an aerial vehicle
- ease of interfacing

The performance requirements for the sensors are derived from the attitude determination and navigation accuracy requirements. The designer must determine that all of the important variables, including sensor drifts or biases, are observable from the chosen set of measurements. The sensor set should have bandwidth at least a decade above the vehicle dynamics. The correctly tuned estimation algorithm adjusts its bandwidth to take information from each sensor at the frequency range where the sensor performance is the best. The preliminary requirements on accuracy of sensors are derived from measurement equations. For example a 5 mg accelerometer bias, when used to correct low frequency gyro drifts, will result in a low frequency error in estimation of local vertical of $.005(180/\pi) \approx 0.3$ degrees. In more complicated cases,

the design of the estimation algorithm has to be performed and covariance simulation run to determine the impact of errors. In the case considered in this chapter, the set of navigation sensors consisted of the inertial measurement unit, GPS receiver with base station for differential corrections, a sonar altimeter, and a magnetic flux compass.

Inertial Measurement Unit (IMU)

The strapdown IMU provides high-bandwidth measurements of the vehicle inertial angular rate and acceleration resolved into body axes. These measurements are used for the inner loops of the control system, and for the time propagation of the navigation equations. The IMU is the heart of navigation system, and care must be taken in making its selection. A detailed study of IMU use in unmanned helicopters is given in [35]. Designs without an IMU were used in the past by Stanford University teams, which relied on multiple antenna high update rate GPS receivers for attitude information. The advantages of using an inertial measurement unit are a much higher bandwidth, higher reliability and integrity. Other sensors, such as GPS receiver can be used to compensate for IMU drifts, and to provide terrain information, in case of sonar altimeter. For application to small aircraft, with short mission times, the main performance characteristics of the IMU are bandwidth, short term bias stability, scale factor errors, and noise. The performance tables for several inertial measurement units are given in Table 3.2.

The noise level can be attenuated by choosing an A/D board with a larger number of quantization bits, shielding analog lines from EM interference, and mechanically isolating the sensor from high-frequency vibrations. A very important consideration is the full scales for both gyros and accelerometers. The full scales should not only be higher than the maximum accelerations and angular rates induced by vehicle dynamics, but also exceed the residual vibration level, not damped by the mechanical filters. The vibration level in a small aircraft can be as high as 10-100 g's, at frequencies as low as 20 Hz. The main sources are engine vibrations and rotor misalignment. The IMU should be put on flexures to reduce vibrations. An ingenious design of flexures

was suggested by Draper '96 team - the flexures were folded mousepads, which in effect creates a second order low-pass filter. The vibration sources, like an engine, should be isolated with anti-vibration mounts. In the example of vibration induced by engine piston, the mount can be rigid in directions perpendicular to piston motion, and very flexible in the direction parallel to it. If the residual vibration level does not saturate the sensors, analog filters with roll off frequencies above vehicle dynamics can be used. A higher sampling rate is beneficial to avoid aliasing of high frequency noise from vibrations into the relatively low frequency region of the rigid body dynamics.

GPS receiver

A GPS receiver provides excellent low frequency position and velocity information, especially in the horizontal plane. The receiver measures distances to the GPS satellites, and then finds its position in earth-fixed coordinates by triangulation, using accurate knowledge of satellite positions. Two sets of range measurements are used: the time interval the signal travelled from satellite to the receiver (called pseudorange), and the fractional number of the signal wavelengths (called carrier phase, or integrated Doppler). The second measurement is obtained by integrating the Doppler shift due to satellite motion, and contains the initial value of the integral (ambiguity), which has to be determined. This value is an integer, and algorithms exist for its resolution. The common error sources for GPS are the selective availability (SA), which is a pseudo random noise signal introduced by US government, multipath (a receiver mistakenly uses a reflected signal instead of the true signal), ionosphere and troposphere delays. Using correction signals from a second GPS receiver, placed at a known location, allows the elimination of SA and reduces the ionosphere and troposphere delays, provided the receivers are located close to each other. This technique is called differential GPS, or DGPS. Also, receivers that use two standard frequencies can correct for ionospheric delay, because it is frequency dependent. Choke rings should be used for ground station antennas to avoid multipath errors and geodesic tripods help as well.

The performance-based selection criteria for GPS receivers are the accuracy and

integrity (i.e. the continuity of the signal tracking without dropouts or failures). Accuracy is provided by using DGPS, carrier phase measurements with integer ambiguity resolution, and dual frequency receivers. Integrity depends on the receiver technology used to keep track of satellites, and the number of satellites tracked by the receiver. GPS/GLONASS receivers, for example, can provide better integrity than GPS-only receivers because of the increased number of satellites available at any point in the world. The satellite lock can be poor due to both antenna vibration and receiver vibration. The latter affects the performance of the oscillator, that provides frequency reference for the receiver components. Loss of GPS track leads to significant performance degradation of navigation solutions, and the IMU should be good enough to keep the system operational until reacquisition. In the case of this unmanned helicopter, the safety pilot should have several seconds to notice that GPS track is lost and take over the control of the vehicle [19]. Longer reacquisition times directly result in tighter requirements on IMU performance. The sampling rate of a GPS receiver can be important if it is used without an IMU.

GPS technology leads to a significant reduction in the cost of avionics systems, because the IMU drift requirements become much less stringent in GPS/INS systems compared to a stand-alone INS. At the current state of technology, improvement in IMU performance is very costly compared to improvement in GPS performance.

Surveys of available GPS receivers are published annually in [33].

Sonar Altimeter

The sonar gives high bandwidth altitude information by sending acoustic signal pulses and measuring the signal return time. This information is used during landing. Selection criteria are accuracy and bandwidth.

Magnetic Compass

A compass provides low frequency heading information by measuring the direction to the local magnetic north. The criteria are accuracy and availability of tilt compensation.

Non-performance based criteria

Great care should be taken in selecting avionics sensors for small UAVs based on power requirements, size, and weight. The issue of power requirements for the IMU and CPU was not given proper attention in 1997, and this led to increased battery weight, a complicated power board design and extensive weight reduction of the vehicle structure. The power requirements should be kept to as few voltage levels as possible to avoid unnecessary power regulation. This is becoming increasingly feasible, since most electronics tend to use a few standard voltages (e.g. ± 5, ± 12 V). A detailed electronic system design for a small helicopter is given in [35].

Convenient sensor interfaces are vitally important. The time spent on development of device drivers and analog prefilter circuitry directly results in program costs. The simplest interface is a serial port. It is, at the same time, often inflexible because it restricts sampling rate and does not allow analog prefiltering. The sampling rate problem is likely to be mitigated with faster electronics. The sensor bandwidth and internal prefiltering can usually be specified by the customer. In addition to saving development time, another great advantage of a serial interfaces is its adaptability for hardware-in-the-loop simulation. Reproducing analog outputs from a digital simulation can often be costly, in comparison to purchasing additional serial cards for the simulation computer. An integrated simulation framework is suggested in Appendix A.

2.3 Guidance, Navigation and Control Algorithms

2.3.1 Algorithm Reuse

The 1997 and 1998 MIT teams used the same sensor configuration as MIT/Draper Laboratory DSAAV (Draper Small Autonomous Aerial Vehicle) used in 1996. In 1997, the performance of the sensors was close to that of 1996 sensors, and the MIT team completely reused the MIT/Draper '96 team algorithms for guidance, navigation and control (GN&C). The control logic employed in the 1996 design used a set of

proportional, integral and derivative (PID) inner control loops, and waypoints were used for outer loop guidance. These algorithms were a result of 2 years of experiments with autonomous helicopters, and proved to be robust, flexible and tunable in flight. In 1997, the navigation algorithm, based on an extended Kalman filter (EKF), was tuned by changing the measurement and driving noise covariance matrices. Navigation and control were run on board, and the guidance algorithm was run in a ground station computer. The description of GN&C algorithms used in MIT/Draper 1996 helicopter system is given in [19].

In 1998 the sensor performance has changed considerably: the IMU, GPS and compass were significantly different. The '98 Crossbow IMU had much larger biases than the '96 Motion Pak and '97 GEC Marconi units. On the other hand, a Novatel RT-2 receiver showed considerable improvement in performance with respect to RT-20 due to higher reliablility of in-flight ambiguity resolution. Also, a TCMVR-20 magnetic flux compass from Precision Navigation gave better heading angle estimate and provided direct measurements of roll and pitch angles. Thus, the need to update '96 EKF has emerged and the detailed design of a new EKF for use in 1998 MIT helicopter is given in the next section.

2.3.2 Extended Kalman Filter Design for Navigation of The 1998 Helicopter

Selection of States

The navigation computations are performed using two reference frames: a local north-east-down (NED) frame, with the center of coordinate system chosen at the location of the GPS base station, and an aircraft body frame with its origin at the aircraft center of mass, X axis forward and Z axis down. The DSAAV navigation algorithm [19] used 11 states. Three more states, corresponding to gyro biases were added to account for poorer IMU performance. The resulting state vector is

$$x = \begin{bmatrix} x_L & y_L & z_L & u_B & v_B & w_B & q_0 & q_1 & q_2 & q_3 & g & \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \qquad (2.1)$$

where x_L, y_L, z_L are vehicle positions relative to the ground station in NED frame, u_B, v_B, w_B are body axis velocity vector components; q_0, q_1, q_2, q_3 are a quaternion representation of attitude, g is a gravity acceleration estimate, γ_1, γ_2, γ_3 are gyro bias states. Gravity acceleration state implicitly accounts for accelerometer biases and thus improves the attitude estimate [29]. The quaternion is a minimal representation of attitude which is free of singularities. However, there is a recurring need to normalize the quaternions after each time propagation or measurement update step:

$$\overline{q} := \frac{\overline{q}}{\| \overline{q} \|_2} \tag{2.2}$$

where $\overline{q} = [q_0 \; q_1 \; q_2 \; q_3]$, and $\| \overline{q} \|_2 = \sqrt{\overline{q} \cdot \overline{q}}$.

Time Propagation Equations

Between measurement updates the state estimate is propagated using the nonlinear differential equation:

$$\dot{\hat{x}} = f(\hat{x}, \overline{\omega}, \overline{a}) \tag{2.3}$$

where $\overline{\omega}$ are the rate gyro measurements and $\overline{a}$ are the accelerometer measurements. The inertial sensors are precalibrated before flight to account for biases. The inertial unit should be given a 20 to 40 minute warm up time to avoid transient temperature dependent bias and scale factor changes, at which point the calibration procedure could be started. Estimates of body rates and accelerations are formed from sensor outputs in flight:

$$\omega_B = \omega_{IMU} \times SF_\omega - \omega_{bias}$$
$$a_B = a_{IMU} \times SF_a - a_{bias}$$

where SF_ω and SF_a are gyro and accelerometer biases. Accelerometer measurements must be transferred to c.g. location to correct for the IMU moment arm (Appendix B). Define the following vector arrays $p_L = [x_L \; y_L \; z_L]$, $v_B = [u_B \; v_B \; w_B]$, $q = [q_0 \; q_1 \; q_2 \; q_3]$, $\overline{g} = [0 \; 0 \; g]$, $\gamma = [\gamma_1 \; \gamma_2 \; \gamma_3]$. Define a coordinate transformation matrix:

$$C = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.4}$$

Define 3D and 4D skew-symmetric matrices for 3D vectors:

$$\Omega_4 = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \\ -\omega_x & 0 & -\omega_z & \omega_y \\ -\omega_y & \omega_z & 0 & -\omega_x \\ -\omega_z & -\omega_y & \omega_x & 0 \end{bmatrix} \Gamma_4 = \begin{bmatrix} 0 & \gamma_x & \gamma_y & \gamma_z \\ -\gamma_x & 0 & -\gamma_z & \gamma_y \\ -\gamma_y & \gamma_z & 0 & -\gamma_x \\ -\gamma_z & -\gamma_y & \gamma_x & 0 \end{bmatrix} \tag{2.5}$$

$$\Omega_3 = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \Gamma_3 = \begin{bmatrix} 0 & -\gamma_z & \gamma_y \\ \gamma_z & 0 & -\gamma_x \\ -\gamma_y & \gamma_x & 0 \end{bmatrix} \tag{2.6}$$

Then nonlinear propagation equations can be written as:

$$\begin{aligned} \dot{\overline{p}}_L &= C\overline{v}_B \\ \dot{\overline{v}}_B &= \overline{a} + C^{-1}\overline{g} - (\Omega_3 - \Gamma_3)\,\overline{v}_B \\ \dot{\overline{q}} &= -\frac{1}{2}(\Omega_4 - \Gamma_4)\,\overline{q} \\ \dot{g} &= 0 \\ \dot{\overline{\gamma}} &= 0 \end{aligned} \tag{2.7}$$

These equations can be numerically integrated at the IMU sampling rate using a first or second order integration routine. The IMU sampling rate of 25 Hz proved to be sufficient using a first order integration routine.

The nonlinear propagation equations can be linearized, with the Jacobian matrix

$$A(\hat{x}, \overline{\omega}) = \left. \frac{\partial f(x, \overline{\omega}, \overline{a})}{\partial x} \right| x = \hat{x} \tag{2.8}$$

This matrix is used to propagate the error covariance matrix between measurement

updates:

$$\dot{P} = A(\hat{x}, \overline{\omega})P + PA^T(\hat{x}, \overline{\omega}) + Q \tag{2.9}$$

where Q is the process noise intensity matrix, which is used to model errors of inertial sensors. Typically the Q matrix is chosen to be diagonal. The suggested initial guess for main diagonal values is derived from sensor covariances:

$$Q_d = \frac{1}{\tau}\left[\begin{array}{cccc} 0_{1\times3} & \sigma_a^2 I_{1\times3} & \sigma_g^2 I_{1\times4} & \sigma_{ab}^2 \quad \sigma_{gb}^2 I_{1\times3} \end{array}\right] \tag{2.10}$$

where σ_a, σ_g are the estimated accelerometer and gyro noise rms values, and σ_{ab}, σ_{gb} are bias stability rms values, and τ is the sampling rate of inertial sensors. The scaling by $1/\tau$ is needed to account for the fact that the equations are integrated numerically with time step τ [14]. The σ parameters can be used to tune the system for maximum performance.

A first order integration of equations 2.9 proved to be satisfactory. The continuous propagation equations can be discretized, and a discrete version of propagation equations can be used [14]. The Jacobian matrix A is a sparse matrix, given in Appendix B.

Measurement equations

At the instances of time when the updates of DGPS, sonar altimeter or compass occur, the measurement update equations are used. Let $h(x)$ be the nonlinear measurement function of state, and let

$$H = \frac{\partial h(x)}{\partial x} \tag{2.11}$$

Then the EKF measurement update equations are given by [14] :

$$\begin{aligned} K &= PH^T\left[HPH^T + R\right]^{-1} \\ \hat{x}^+ &= \hat{x} + K\left[z - h(\hat{x})\right] \\ P^+ &= P - KHP \end{aligned} \tag{2.12}$$

where K is the filter gain, R is the measurement noise covariance matrix, and z is the measurement. In some instances the inversion of the symmetric matrix in equation 2.12 may become close to singular, so the computer may not have a sufficient number of bits (16 would be marginal) to assure positive eigenvalue. In this case the computations are unstable, so the square root implementation of the KF should be used [14].

DGPS is used to correct low-frequency drifts in position, velocity and attitude estimates. The sampling rate of the RT-2 receiver is 4 Hz. DGPS provides anntenna position and velocity information in the local NED frame. This data must be transferred to the vehicle c.g. location (Appendix B). The linearized equations for GPS measurements are:

$$H_p = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times1} & 0_{3\times3} \end{bmatrix} \tag{2.13}$$

$$H_v = \begin{bmatrix} 0_{3\times3} & C & H_{vq} & 0_{3\times1} & 0_{3\times3} \end{bmatrix} \tag{2.14}$$

The block H_{vq} is given in Appendix B. Standard deviations of position and velocity measurement are provided as outputs from the GPS receiver. These estimates take into account the current state of the internal receiver KF, whether ambiguities are resolved, the number of satellites etc. These values are used to determine the measurement noise covariance. Under normal operating conditions, when the estimate has converged, the standard deviation of horizontal position measurement error is $\sigma_p = 0.02$ m, and horizontal velocity measurement error standard deviation is $\sigma_v = 0.003$ m/s for RT-2 receiver. Vertical position and velocity error standard deviations maybe 3-4 times larger than horizontal. In the data sheets for RT-2 receiver the specified time to convergence after GPS acquisition is around 12 min if at least 6 satellites are available, thus the receiver should be started during the IMU warm-up.

DGPS provides altitude information relative to the location of the base receiver. To take into account terrain information during landing, the sonar altimeter is used. Sampling rate for the sonar is chosen to be 5 Hz. The altimeter measurement updates

are used only if the measured altitude is between 0.5 m and 6 m. The altimeter sends a pulse and shuts down so that it does not mistakenly take the emitted signal for the reflected one. This explains the lower limit on sonar measurements. Sonar measurements are also transferred to c.g. location (Appendix B). The standard deviaton of sonar measurement error increases with altitude, at low altitude it is $\sigma_h \approx$ 10 cm due to grass and other surface non-uniformity. The linearized measurement equation is:

$$H_h = \begin{bmatrix} 0 & 0 & 1 & 0_{1\times 3} & 0_{1\times 4} & 0_{1\times 1} & 0_{1\times 3} \end{bmatrix} \tag{2.15}$$

For the case of a non-maneuvering vehicle, the heading angle is unobservable in the GPS velocity measurements (see Section 3.7.2). Magnetic compass updates are used to correct low-frequency drifts of the heading angle estimate. A sampling rate of 1 sec is sufficient. The TCMVR-20 compass also provides roll and pitch angle estimates. These are used inside the compass to correct heading information for tilt. However, if CPU time allows, these measurements can be used in the EKF as well. The compass measurement is corrected for magnetic variation, i.e. local direction to magnetic North with respect to true North. The accuracy for compass measurement is $\sigma_\psi \approx 2$ degrees. The linearized measurement equation for heading measurement is

$$H_\psi = \begin{bmatrix} 0_{1\times 3} & 0_{1\times 3} & H_{\psi q} & 0_{1\times 1} & 0_{1\times 3} \end{bmatrix} \tag{2.16}$$

where $H_{\psi q}$ is given in Appendix B.

2.4 Software Architecture and Reuse

A major part of the onboard and ground station software from the 1996 system was reused. Both ground station and onboard software were implemented in the C programming language under the QNX operating system (OS). The QNX OS is a multitasking operating system, allowing several independent processes to be run simultaneously.

In retrospect, the decision to reuse the software was not justified. A summary of the major functions of the '97 software is presented and the reasons why it was hard to reuse the '96 software are given in the following paragraph. Ways to improve software design for an unmanned helicopter are suggested as well. Detailed descriptions of the '96 onboard and ground station software is given in [19].

The three major functions of the onboard software were navigation, control and communication. The ground station functions include mission planning, waypoint guidance, processing of vision information, communication, and the operator interface. The vision processing code takes information from the Cognachrome vision board, indicating the location of orange pixels in the camera field of view. It decides whether there is a disc, and estimates the disc coordinates based on the current state of the helicopter. The mission planner adds this location as a new entry into waypoint list. The guidance logic converts the waypoint list into low-level position, velocity and heading commands for the onboard control loops. Implementing the guidance logic in the ground station introduces a delay, and it is recommended that it be incorporated onboard the vehicle.

Figure 2-3 illustrates a high level flow chart for the onboard software. The onboard processor communicates with the external devices via serial ports, using non-blocking I/O. This means that the main program will not hang up in case the device being sampled fails. This is an important safety feature. However, it would be logical to use the multitasking capability of QNX OS to have separate processes for interfacing with devices, for control and navigation logic, and for communication. This would allow better timing for the processes, and easier debugging due to higher modularity.

A good rule for software engineering is to keep different functions in separate files, otherwise the code becomes very hard to debug. With the exception of interfaces, all code that implemented the flow chart above was contained in a single file, which made this software difficult to reuse.

A simple example of the multitasking design approach to the onboard software for unmanned aircraft is given in Section 3.4.

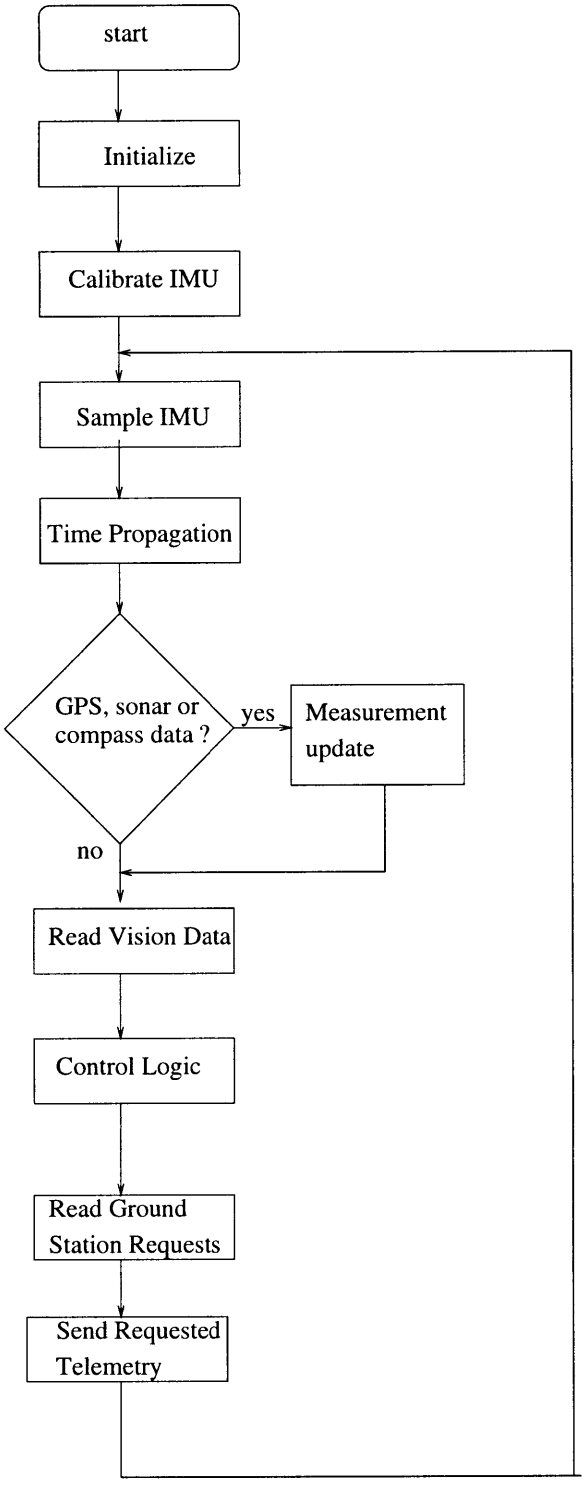

Figure 2-3: Flow chart for onboard software

Chapter 3

Gun-Launched Surveillance Aircraft

3.1 Wide Area Surveillance Projectile

3.1.1 Project Background and Goals

The avionics system development for an autonomous surveillance aircraft is presented in this chapter. The effort described here was done under the sponsorship of Draper Laboratory as a part of the MIT/Draper Technology Development Partnership Project. The major goal of the project was to develop entrepreneurship and innovation in engineering students. The primary requirement was to create an innovative product that would meet a major US national need.

During the fall term of 1996 a team of students, MIT faculty, and engineers from Draper Laboratory proposed several ideas, all related to aerospace. The group performed market assessments and preliminary technical evaluations of the concepts, and finally selected the Wide Area Surveillance Projectile (WASP) concept. WASP is a small gun launched aircraft, designed to meet a need for fast-response surveillance capability. A set of requirements for the vehicle was determined with potential customers (Navy, Army, Defence Airborne Reconnaissance Office) and the sponsor (Draper Laboratory). The requirements were further adjusted as the design pro-

gressed, and final estimates of the required performance are:

- 15,000 g launch environment
- 15 min. loiter time
- 15 km range
- 250 m radius of camera field-of-view, still frame or video image
- fit inside a 5" shell diameter
- 2-way datalink capability
- GPS-based waypoint guidance

In a typical scenario, WASP is launched from a 5" Navy gun, reaches its destination point (15 km range, and altitude of 500 m), is deployed from its shell, and starts sending still images or a videostream, and positioning information of designated points to a ground station. A ground operator designates new waypoints for the vehicle and commands its search pattern. Recognition of points of interest is performed by the operator, and positions of points of interest are determined using a combination of onboard navigation and camera pointing information.

3.1.2 Overall Design

This section provides a brief description of the overall vehicle design. A detailed discussion of the structural design, deployment mechanisms and aerodynamic design are given in [20] and [6].

The flyer is stowed inside a 5" fin-stabilized artillery shell, derived from a re-designed illuminating round. At a predetermined time after gun launch an onboard timer triggers separation of the shell base, and a parachute is deployed. The parachute decelerates the vehicle to a sufficiently low dynamic pressure to allow deployment of the wings, tail surfaces and a propeller. The vehicle is powered by a 2-stroke internal combustion engine, which is started by a preloaded spring device. A drawing of the

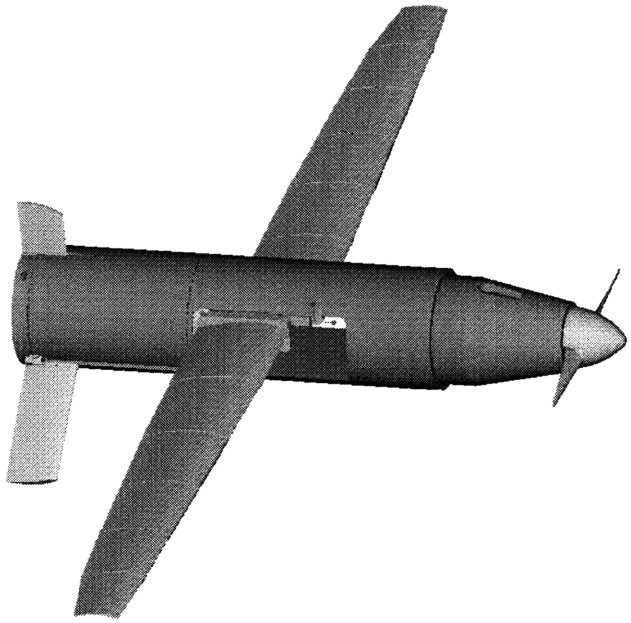

Figure 3-1: Deployed WASP flyer

flyer is given in Figure 3-1. There are two static vertical fins, one on the bottom and one on top of the vehicle; and two downward pointing control surfaces, in a vee-tail configuration, which provide lateral-directional and longitudinal control. The fins were added to improve lateral-directional weathercock stability. The wings have a nominal dihedral angle of 3 degrees.

The onboard avionics system includes a tightly coupled GPS/INS system, a 2 way datalink and a charge-couple display (CCD) camera. A detailed description of the avionics is given in Section 3.2.

3.1.3 Testing Approach

To demonstrate the key capabilities of the proposed system, two prototypes were developed: a high-g vehicle and a flight test vehicle (FTV). The requirements for the high-g vehicle were to demonstrate the survival of the airframe and shell in the gun environment, and to show operation of the deployment mechanisms. The high-g vehicle does not have any avionics onboard. Micromechanical high-g-qualified avionics systems for use in smart munitions programs are currently under development in

Draper Laboratory, but are not available for the WASP flight tests because of the high cost. The WASP design relies on the future availability of these systems.

The other prototype, the FTV, was designed to demonstrate flying qualities and aerodynamic performance of the deployed flyer. The preliminary stability analysis of the WASP flyer showed that the flying qualities will present a challenge for the design of a stability augmentation system. This is due to minimal lateral control authority because there are no ailerons and small control surface area, and a near-stall incidence angle of the wings.

The FTV is a scaled up test version of the operational flyer, equipped with off-the-shelf avionics. It is not launched from a gun but rather deployed from a conventional ultralight aircraft. It has a remote control receiver with an interface to the onboard computer, and can be flown by a remote pilot with a control augmentation system. The parameters of the prototype vehicle were chosen to match important stability characteristics of the WASP (Section 3.5.10). The "scaling-up" approach, used in the project to achieve dynamics similarity of the vehicles, can be a useful tool for development of a small unmanned aircraft, it provides a relatively low-cost and high-fidelity solution to testing and data gathering.

3.2 Hardware Architecture for the Operational Vehicle

A set of specifications for the WASP avionics systems was derived from the perceived customer requirements, stated in Section 3.1.1.

- digital flight control system
- GPS/INS navigation and control
- onboard imaging system
- 2-way encrypted datalink

All avionics must survive 15,000 g. The Draper Laboratory developed high g-qualified integrated INS/GPS navigation system for Extended Range Guided Munitions (ERGM) program and another follow on advanced technology demonstrator (ATD) program, which will provide the technology for the core avionics for the WASP. Both the ERGM and ATD systems use a micromechanical inertial sensor assembly (MMISA), and a DSP-based GPS receiver. The flight control computer is also a g-hardened DSP chip. The fact that these systems are being developed at Draper Laboratory was a major advantage in pursuing the gun-launched surveillance aircraft concept, because of the ready access the team has had to information and advice from Draper engineers.

A CCD camera will be used as an imaging sensor. A high-g CCD camera is being developed by Xybion Corporation. Two COTS CCD cameras (Black & White Pro-Video CVC-50PH, Black & White Watec WAT-660) survived air-gun testing at 15,000 g's. Thus there will potentially be a choice for a high-g qualified imaging sensor.

Separate C-band receiver and transmitter are suggested for the two-way datalink. A message forming encryption chip, that takes telemetry information from the flight control computer and video information from the CCD camera will relay data to the ground via the downlink transmitter.

The signal flow diagram for the operational WASP is presented in Figure 3-2.

3.3 Hardware Architecture for the Flight Test Vehicle (FTV)

3.3.1 Requirements and Constraints

The initial set of requirements for a prototype vehicle included a demonstration of autonomous flight, with a minimal requirement to demonstrate that the vehicle can be flown by a remote control pilot, with the help of an onboard stability and control augmentation system. Although the FTV uses off-the-shelf electronics its aerodynamic

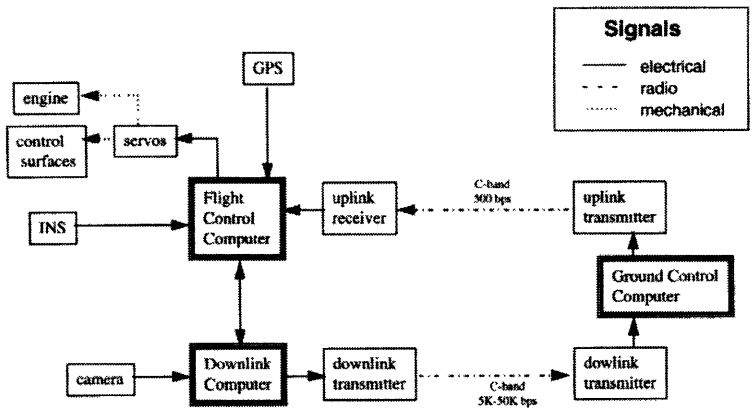

Figure 3-2: Signal flow for operational WASP

shape is kept as close as possible to the deployed configuration of the operational vehicle. FTV is not required to demonstrate deployment mechanisms. The same servo gears and a similar 2 cycle internal combustion engine are used in the FTV. Also, both the FTV and the operational WASP do not have landing gear. The FTV deployment is done from a remotely piloted, powered, ultra-light aircraft; under command from a ground based operator. The retrieval is done by deploying a parachute from the base of the FTV. A digital control system allows tuning of control algorithms in flight and gathering flight test data.

The vehicle was a $\times 1.28$ scaled up copy of the operational WASP. The volume constraint was very important for FTV. The allocation of weights inside the FTV was done to keep the dynamics of the flyer similar to that of the operational WASP. The vibrations induced by the engine piston motion and propeller unbalance are important factors in development of the FTV.

3.3.2 Subsystems

Sensors

The inertial measurement unit was chosen to provide high-bandwidth information for the stability augmentation system and attitude information for the autopilot. To

meet the requirement for autonomous flight, the Systron-Donner Motion Pak was chosen. The main performance characteristics of the unit are given in Table 3.2. Motion Pak has analog outputs and requires an A/D card onboard the aircraft to provide an interface to the onboard computer. A mistake was made during the selection of accelerometer full scale ranges: they were chosen to match vehicle dynamics only. As was mentioned in the Section 2.2, it is necessary to choose the full scale of accelerometers higher than the vibration level that can be experienced in the vehicle. Because the sensitivity was chosen to be quite high, relative to the vibration created by the engine, the team had to design special anti-vibration mounts to reduce vibration level below .1 g peak-to-peak. These mounts were used to isolate the engine from the nose cone, and the nose cone from the fuselage. However, in spite of the anti-vibration mounts a 100 Hz vibration component, from propeller unbalance, still remained a problem. To fix it, the IMU was mounted on a series of folded rubber sheets. This effectively creates a flexure that acts a fourth order lowpass filter, and guarantees a -80 dB/decade slope at high frequencies. Finally, when saturation of IMU was no longer a problem an active analog low pass filter, with 8 hz bandwidth, was used to filter out residual high frequency vibration. The sampling rate was increased from 50 hz to 100 hz to avoid aliasing.

To achieve autonomous operation, differential GPS was chosen to correct for IMU drifts and provide heading information indirectly. Section 3.7.2 describes an attitude and heading estimation algorithm that uses IMU data and GPS velocity information. The G12 GPS receiver provides 1 m SEP position accuracy and .01 m/sec velocity accuracy in the differential mode. It has a 2 second reacquisition time. The inertial system has sufficient performance to provide adequate navigation during short GPS dropouts.

Communication

Radio modems were chosen to provide two way communication links over a 1 mile range. The modems have serial interfaces, and they are either directly attached to the CPU, or their outputs go through the GPS receiver serial ports.

Computer Stack

A PC-104 stack, described in Section 2.2, was used as the communication bus for the onboard electronics. The stack included a 486DXi 100 MHz CPU from Ampro Computers, a 12 bit A/D card from WinSystems, a power supply from Tri-M engineering and ethernet card with boot ROM from Florida DataMation. The 100 MHz processor speed gives a significant advantage in onboard computing power and allows IMU output sampling at more than 100 Hz. By comparison, the 50 MHz processor used in the unmanned helicopter project, described in Chapter 2, could not handle more than a 25 Hz sampling rate, resulting in marginal performance.

The VGA PC104 video card was used in table top testing for troubleshooting of the stack electronics. As mentioned in Section 2.2, it allows one to connect the CPU to a monitor and look at the program outputs directly, not via Ethernet link with ground station computer.

Remote Control Unit

Standard model aircraft remote control servos were chosen to drive the differential stabilizers. The torque requirements, weight and size were the main drivers in the choice of the servos. The Futaba PPM receiver and interface, as described in Section 2.2, were used to deliver pilot commands to the computer, and for computer commands to the servos.

Retrieval Mechanism

A parachute deployment, to allow safe retrieval of the FTV, is commanded by an RC servo via computer or pilot command.

Power System

During the process of choosing components special attention was given to minimizing the number of voltage levels used, and the total power requirements as well, so as to minimize battery weight. The complete design of the power system is given in [17].

Table 3.1 lists power requirements for the subsystems.

Component	Power (W)	Voltage (V)	Tolerance (%)
GPS receiver	1.40	5	5
GPS antenna	0.75	5-15	N/A
IMU	3.50	15	20
	3.50	-15	20
CPU	4.90	5	5
Ethernet	2.00	5	5
Serial Ports	2.00	5	5
A/D	0.05	5	5
	0.12	12	5
	0.12	-12	5
Modem	1.00	5	5
Total			19.64
	Battery Efficiency		×1.2
	Conversion Efficiency		×1.33
	Total Power Required		31.34

Table 3.1: Power Requirements For Prototype Vehicle

The RC receiver and FPGA draw power from a separate battery. The cards in the PC-104 stack are powered by a PC-104 power supply, which draws voltage in a 10-20 V range from the main battery. The GPS antenna is powered by the receiver.

Figure 3-3 shows the signal flow for the FTV.

3.4 Software Design for the Prototype Vehicle

3.4.1 Functionality

A good software design is essential for system integrity and safety. For example, the prototype vehicle retrieval mechanism is a parachute, and the deployment of the parachute is commanded via the CPU. If the parachute is not deployed the vehicle will be destroyed, and could pose a hazard to people or the environment. An effective and well documented software design increases productivity of the software development, reduces the chances of unpredicted software behavior, and can save considerable de-

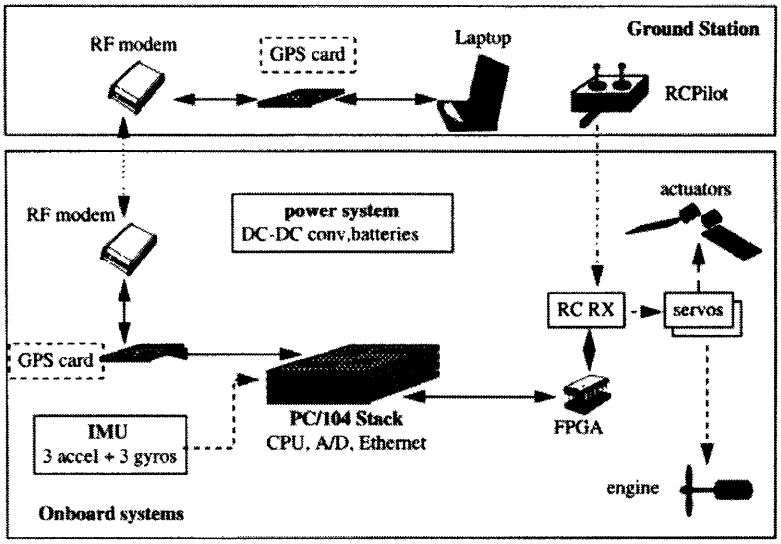

Figure 3-3: Signal flow for prototype Vehicle

bugging time and effort. The onboard and ground station software are implemented in the C programming language under the QNX real-time operating system (OS).

One important requirement for system safety is that the parachute should be deployed if one of the following is true:

- pilot commanded deployment
- ground operator commanded deployment
- communication link is lost
- RC link is lost

The multitasking capability of QNX OS is used to implement this logic. The program is split into three processes (Figure 3-4): control, communication, and "watchdog". The control process functions are:

- read IMU data from A/D card at 100 Hz
- read battery voltage from A/D card at 1 Hz

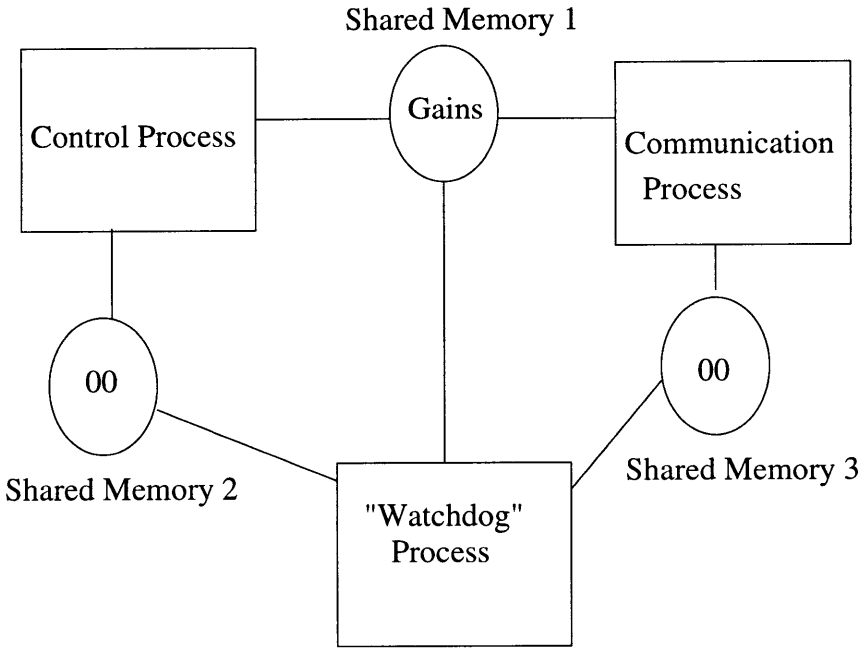

Figure 3-4: Software diagram for FTV

- read pilot inputs from receiver interface (serial port) at 5 Hz
- compute control outputs and command them to receiver interface at 50 Hz
- perform attitude calculation at 100 Hz
- update parameters in shared memory location 1 (see below) at 5 hz
- control parachute deployment bits in shared memory location 2 (see below) at 10 Hz

The memory location 1 is shared by the communication and control processes. It contains the control system gains, that can be commanded to the communication process by the ground operator, and telemetry information, that the control process gives to the communication process.

The role of the communication process is to

- read requests from ground computer via serial interface with modem
- update parameters in shared memory location 1

- write requested and default telemetry to modem port
- update chute deployment bits in shared memory location 3

The watchdog process has access to shared memory locations 2 and 3. Both contain only 2 bits. The first bit is the command from the other process sharing location to deploy the parachute. The command can come from the pilot through the control process, or from the ground operator through the communication process. The second bit in each location is a health bit. The watchdog process resets both health bits to 0 each time it runs. The other process sets it back to 1. In the event that the health bit remains 0 more than a certain threshold time, the watchdog process deploys the chute.

3.4.2 Implementation Issues

Shared Memory

The use of shared memory can pose a serious danger if not handled properly. Each process, before reading from or writing to a shared memory location, must lock the location, and unlock it after update is made. If this is not done, a race condition may occur in which one process may read data while it is being changed by another process. Thus, the value of the data can be completely erroneous. The FTV shared memory is implemented using "shmem_open" and "mmap" functions of the Watcom C compiler.

Timing

The timing can be done using "signals", which have the same meaning in QNX as in UNIX, or by "proxies", which are QNX versions of interrupt. Each of these requires a per-process timer. The timer is periodically reloaded, and when it expires, it sends either a proxy or a signal, which triggers the run of the process.

In this design, all functions inside a process are run at multiples of the time period the process itself is run. Thus simple counters can be used to trigger the lower rate

functions. The timer is created by the "create_timer()" routine.

Creation of Processes

Processes are created by the "fork" routine. For example, the "watchdog" is a parent process, it creates a copy of itself using "fork". Then this copy is substituted with the child process, which is either the communication process or the control process. The watchdog timer is chosen to be a parent process for both of the other processes because it should have access to shared memory locations with each of the processes.

3.5 Model and Simulation Development

3.5.1 Approach to modeling

This section describes the development of a common simulation framework for both vehicles, provides a discussion of the linearized dynamics, and a description of the stability and control augmentation system design.

It was decided to create a 6 degree of freedom (DOF) nonlinear simulation of both flyers, find nominal trim conditions, and linearize numerically around the trim conditions to obtain linear lateral-directional and longitudinal vehicle dynamic models. The following features were incorporated into the simulation:

- 6 degree of freedom rigid body equations of motion
- aerodynamic forces and moments
- simplified engine thrust and torque model
- non-linear actuator models
- non-linear sensor models
- wind gust models

A description of these items is provided in the following subsections. The next subsection gives inertial and geometric properties for both vehicles.

3.5.2 Geometric and Inertial Properties of WASP and FTV

The geometric data for the operational WASP vehicle are:

$$\begin{array}{lll} \text{wing area:} & S = 0.0582 & m^2 \\ \text{wingspan:} & b = 1.01 & m \\ \text{aspect ratio:} & A = 15.5 & \\ \text{mean aerodynamic chord:} & \bar{c} = 0.06 & m \end{array} \tag{3.1}$$

and for the FTV all linear dimensions are increased by a factor of 1.28. The inertial properties of the WASP vehicle, weighing 7 kg, are:

$$\begin{array}{lll} J_x = 0.0406\, kg\, m^2 & J_y = 0.1103\, kg\, m^2 & J_z = 0.1125\, kg\, m^2 \\ J_{xz} = 0.0017\, kg\, m^2 & J_{xy} = 0\, kg\, m^2 & J_{yz} = 0.0016\, kg\, m^2 \end{array} \tag{3.2}$$

The correponding inertial properties of the FTV, which weighs 5 kg, are:

$$\begin{array}{lll} J_x = 0.0233\, kg\, m^2 & J_y = 0.0918\, kg\, m^2 & J_z = 0.1060\, kg\, m^2 \\ J_{xz} = 0.0054\, kg\, m^2 & J_{xy} = 0\, kg\, m^2 & J_{yz} = 0.0002\, kg\, m^2 \end{array} \tag{3.3}$$

3.5.3 Equations of Motion

It was assumed that the asymmetric wing locations on the fuselage do not contribute significantly to inertial coupling, and aerodynamic coupling will be accounted for by appropriate trim deflections of the control surfaces. Thus standard 6-DOF equations of motion [32, table 2.4-1] in body axes can be employed:

Force Equations

$$\begin{aligned} \dot{U} &= RV - QW - g\sin\theta + \frac{F_x}{m} \\ \dot{V} &= -RU + PW + g\sin\phi\cos\theta + \frac{F_y}{m} \\ \dot{W} &= QU - PV + g\cos\phi\cos\theta + \frac{F_z}{m} \end{aligned} \tag{3.4}$$

Kinematic Equations

$$\begin{aligned}
\dot{\phi} &= P + \tan\theta(Q\sin\phi + R\cos\phi) \\
\dot{\theta} &= Q\cos\theta - R\sin\phi \\
\dot{\psi} &= \frac{Q\sin\phi + R\cos\phi}{\cos\theta}
\end{aligned} \tag{3.5}$$

Moment Equations

$$\begin{aligned}
\dot{P} &= (c_1 R + c_2 P)Q + c_3\bar{L} + c_4 N \\
\dot{Q} &= c_5 PR - c_6(P^2 - R^2) + c_7 M \\
\dot{R} &= (c_8 P - c_2 R)Q + c_4\bar{L} + c_9 N
\end{aligned} \tag{3.6}$$

Navigation Equations

$$\begin{aligned}
\dot{p_N} &= U\cos\theta\cos\psi + V\left(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi\right) \\
&\quad + W\left(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi\right) \\
\dot{p_E} &= U\cos\theta\sin\psi + V\left(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi\right) \\
&\quad + W\left(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi\right) \\
\dot{h} &= U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta
\end{aligned} \tag{3.7}$$

where the constants c_1 through c_9 are functions of the inertial properties of the vehicle, given as follows:

$$\begin{aligned}
&\Gamma = J_x J_z - J_{xz}^2, && c_1 = \tfrac{(J_x - J_z)J_z - J_{xz}^2}{\Gamma} \\
&c_2 = \tfrac{(J_x - J_y + J_z)J_{xz}}{\Gamma}, && c_3 = \tfrac{J_z}{\Gamma} \\
&c_4 = \tfrac{J_{xz}}{\Gamma}, && c_5 = \tfrac{J_z - J_x}{J_y} \\
&c_6 = \tfrac{J_{xz}}{J_y}, && c_7 = \tfrac{1}{J_y} \\
&c_8 = \tfrac{J_x(J_x - J_y) + J_{xz}^2}{\Gamma}, && c_9 = \tfrac{J_x}{\Gamma}
\end{aligned} \tag{3.8}$$

3.5.4 Aerodynamic Forces and Moments

Non-dimensional aerodynamic coefficients were estimated using a variety of tools. The complete set of coefficients was broken down into 4 subsets:

- constant force and moment coefficients calculated using potential flow theory
- control derivatives
- drag coefficients
- forces and moments that depend non-linearly on angle of attack

Both the operational WASP and FTV fly at a low subsonic Mach number ($M < 0.15$), and the flight altitude does not change significantly during operation. Thus, as long as the aircraft does not stall, the only parameter that affects the aerodynamic coefficients is center of gravity (cg) location. The cg changes with time, as fuel is burned, and its change of location primarily affects the pitching moment due to angle of attack coefficient - C_{m_α}. The center of gravity changes from 20.3 cm forward of the back end, with a full fuel tank, to 20.0 cm forward of the back end, with an empty fuel tank. This change represents 5 percent of the mean aerodynamic chord and it is accounted for in the simulation. Also, it is reasonable to assume that both FTV and WASP have the same values of aerodynamic coefficients, because Mach number is not a relevant parameter at $M < 0.15$, and drag derivatives do not affect stability.

Coefficients calculated using potential flow theory

These coefficients were estimated using the Athena Vortex Lattice program (AVL). The value for C_{m_α} given here is at the most aft cg location, and it is calculated at each timestep of the simulation, using the current estimate of the fuel weight. The simulation equations are integrated with a variable time step, determined by Simulink to better represent continuous dynamics. It is assumed that the fuel weight is changing according to the following ODE:

$$\dot{x} = \begin{cases} -K_{fuel}T & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where T is a throttle setting in Watts and

$$K_{fuel} = \frac{W_{fuel}}{T_{max} t_{exp}} \tag{3.9}$$

where W_{fuel} is initial fuel weight, T_{max} is maximum throttle setting, and t_{exp} is time (in seconds) it takes to burn a full tank of fuel at full throttle. The c.g. location of the vehicle, due to fuel burn, is determined from the following equation:

$$X_{vf} = \frac{m_v X_v + m_f X_f}{m_v + m_f} \tag{3.10}$$

where m_v - weight of the vehicle without fuel, X_v - c.g. of empty vehicle, m_f - fuel weight, X_f - fuel tank c.g. The change in C_{m_α} is given by:

$$\delta C_{m_\alpha} = \frac{m_f(X_f - X_v)}{(m_v + m_f)\bar{c}} C_{L_\alpha} \tag{3.11}$$

The remaining coefficients are all assumed to be constant and their values are as follows:

Longitudinal Coefficients

$$C_{L_\alpha} = 6.261 \quad C_{L_q} = 9.866$$
$$C_{m_\alpha} = -0.163 \quad C_{m_q} = -13.188$$

Lateral Coefficients

$$C_{Y_\beta} = -1.007 \quad C_{l_\beta} = -0.085 \quad C_{n_\beta} = 0.015$$
$$C_{Y_p} = -0.119 \quad C_{l_p} = -0.729 \quad C_{n_p} = -0.162$$
$$C_{Y_r} = 0.488 \quad C_{l_r} = 0.420 \quad C_{n_r} = -0.083$$

The accuracies of these estimates are consistent with currently available methods for predicting stability derivatives based on both purely analytical methods and numerical finite element calculations, where appropriate. For example, the value of lift curve slope $C_{L_\alpha} = 6.261$, obtained with a vortex lattice method, is probably an overestimate, since this slope is not likely to be so close to 2π, for a three-dimensional vehicle. However, the value for the static margin $\frac{C_{m_\alpha}}{C_{L_\alpha}}$, obtained using an estimation formula, was verified independently with wind tunnel tests and gave very close results. For most aircraft configurations, the vortex-lattice method tends to give stability derivatives within 15 percent of their true values. However, the AVL representation of WASP contains a crude model of the fuselage, which is represented by two intersecting plates rather than a body of revolution. A number of these kind of approximations were made, which should be verified by wind tunnel and flight testing.

Control derivatives

The control derivatives were estimated using two independent methods. Analytical values, based on traditional vehicle models, were compared with AVL values computed for various surface defelctions. It was found that both methods produce consistent results for the longitudinal control derivatives. There was less consistency for the lateral directional derivatives. The smaller values, obtained with the analytical techniques, implied poorer flying qualities (smaller values of control derivatives correspond to lower control power), and these values were used in the simulation.

$$C_{m_{i_h}} = 1.27 \quad C_{L_{i_h}} = 0.5591$$

$$C_{Y_{i_h}} = -0.1 \quad C_{l_{i_h}} = 0.009 \quad C_{n_{i_h}} = 0.01$$

Here longitudinal derivatives correspond to symmetric deflection of the differential stabilizer, in which case the stabilizer acts like an elevator, and lateral derivatives correspond to antisymmetric deflection, in which case the stabilizer acts like a rudder.

Drag derivatives

Drag was estimated both analytically and in the wind tunnel. The analytical prediction tended to overestimate drag, and wind tunnel data are given here for drag at zero angle of attack: $C_{D_0} = 0.0881$. The induced drag formula is used for computation at each timestep, because induced drag depends on lift coefficient, which varies with angle of attack.

Coefficients with non-linear dependence on angle of attack

At each timestep during the simulation, angle of attack α, angle of sideslip β, as well as projections of angular rates onto aircraft stability axes are computed. After the lift coefficient is computed, its value is used to compute induced drag. The induced drag component stems from empennage drag, induced drag on the wings, and induced drag on fuselage. Induced drag on wings is given by

$$C_{D_{L_w}} = \frac{{C_L}^2}{\pi A e}, \tag{3.12}$$

where e = 0.8 - efficiency factor, A = 15.5 - aspect ratio, C_L - lift coefficient, given by equation 3.17. Induced drag on fuselage is calculated as

$$C_{D_{L_{fus}}} = 2\alpha^2 \frac{S_{b_{fus}}}{S} + \eta C_{D_c} \alpha^3 \frac{S_{pl_{fus}}}{S}, \tag{3.13}$$

where $S_{b_{fus}} = 0.0125\ m^2$ - fuselage base area, and $S_{pl_{fus}} = 0.0672\ m^2$ - fuselage planform area, $\eta = 0.65$ - ratio of the drag of a finite cylinder to the drag of an infinite cylinder, C_{D_c} - experimental steady state crossflow drag. To determine induced drag on the tail it is necessary to obtain lift generated by the tail:

$$C_{L_h} = C_{L_{\alpha_h}} (\alpha + \frac{i_r + i_l}{2}), \tag{3.14}$$

where $C_{L_{\alpha_h}} = 4.71$ - empennage lift slope, i_r and i_l - right and left stabilizer deflections (positive down), so $(i_r + i_l)/2 = \delta_e$ - effective elevator action. The induced drag on

the tail is:

$$C_{D_{L_{emph}}} = \frac{C_{L_h}^2 S_t \cos\delta_t}{\pi A_h e_h S}, \tag{3.15}$$

where $S_t = 0.0119\,m^2$ - tail area, $\delta_t = 35\,deg$ - tail anhedral, $A_h = 3.89$ - tail aspect ratio, $e_h = 0.7$ - tail planform efficiency factor.

Dimensional Forces and Moments

At each simulation timestep, aerodynamic forces and moments are computed using standard dimensionalizing formulae. First, angular rates are transformed to stability axes:

$$\begin{aligned} P_s &= P\cos\alpha + R\sin\alpha \\ Q_s &= Q \\ R_s &= -P\sin\alpha + R\cos\alpha \end{aligned} \tag{3.16}$$

The lift coefficient is obtained as:

$$C_L = \begin{cases} C_{L_0} + C_{L_\alpha}\alpha - C_{L_{i_h}}(\frac{i_r+i_l}{2} + \epsilon_t) + C_{L_q}\frac{Q_s c}{2Vcr} & \text{if } C_L < C_{L_{max}} \\ C_{L_{max}} & \text{otherwise} \end{cases} \tag{3.17}$$

where $C_{L_0} = 1.4726$ - zero angle of attack lift coefficient, $C_{L_{i_h}} = 0.5591$ - tail lift curve slope, V_{cr} - current velocity, ϵ_t - downwash angle, assumed to be constant at 0.1 deg. $C_{L_{max}} = 1.91$ - maximum lift coefficient used for crude modeling of stall. The lift force is obtained from $F_z = -C_L\bar{q}S$.

The total drag coefficient is:

$$C_D = C_{D_0} + C_{D_{L_w}} + C_{D_{L_{fus}}} + C_{D_{L_{emph}}} \tag{3.18}$$

The aerodynamic force in the X-direction in stability axes is $F_x = -C_D\bar{q}S$.

The aerodynamic force coefficient in the Y direction is:

$$C_Y = C_{Y_\beta}\beta + C_{Y_p}\frac{P_s b}{2V_{cr}} + C_{Y_r}\frac{R_s b}{2V_{cr}} + C_{Y_{i_h}}\frac{i_r - i_l}{2} \tag{3.19}$$

and the Y-axis force is $F_Y = C_Y \bar{q} S$.

Then the forces are transformed from stability axes into body axes:

$$\begin{aligned} F_{x_b} &= F_x \cos\alpha - F_z \sin\alpha \\ F_{y_b} &= F_y \\ F_{z_b} &= F_z \sin\alpha + F_z \cos\alpha \end{aligned} \tag{3.20}$$

Similar equations are developed for aerodynamic moments: The rolling moment coefficient is:

$$C_l = C_{l_\beta}\beta + C_{l_p}\frac{P_s b}{2V_{cr}} + C_{l_r}\frac{R_s b}{2V_{cr}} + C_{l_{i_h}}(\frac{i_r - i_l}{2}) \tag{3.21}$$

Rolling moment in stability axes is $L_s = C_l \bar{q} S b$.

The pitching moment coeffcient is given by:

$$C_m = C_{m_0} + \frac{C_{m_\alpha}}{C_{L_\alpha}}C_L + C_{m_q}\frac{Q_s c}{2V_{cr}} + C_{m_{i_h}}(\frac{i_r + i_l}{2}), \tag{3.22}$$

where $C_{m_0} = -0.0688$ - pitching moment coefficient at zero lift, zero stabilizer deflection. The aerodynamic pitching moment in stability axis: $M_s = C_m \bar{q} S \bar{c}$.

The yawing moment equation is:

$$C_n = C_{n_\beta}\beta + C_{n_p}\frac{P_s b}{2V_{cr}} + C_{n_r}\frac{R_s b}{2V_{cr}} + C_{n_{i_h}}(\frac{i_r - i_l}{2}) \tag{3.23}$$

The aerodynamic yawing moment in stability axis: $N_s = C_n \bar{q} S b$.

Then the moments are transformed from stability axes to body axes:

$$\begin{aligned} L_b &= L_s \cos\alpha - N_s \sin\alpha \\ M_b &= M_s \\ N_b &= L_s \sin\alpha + N_s \cos\alpha \end{aligned} \tag{3.24}$$

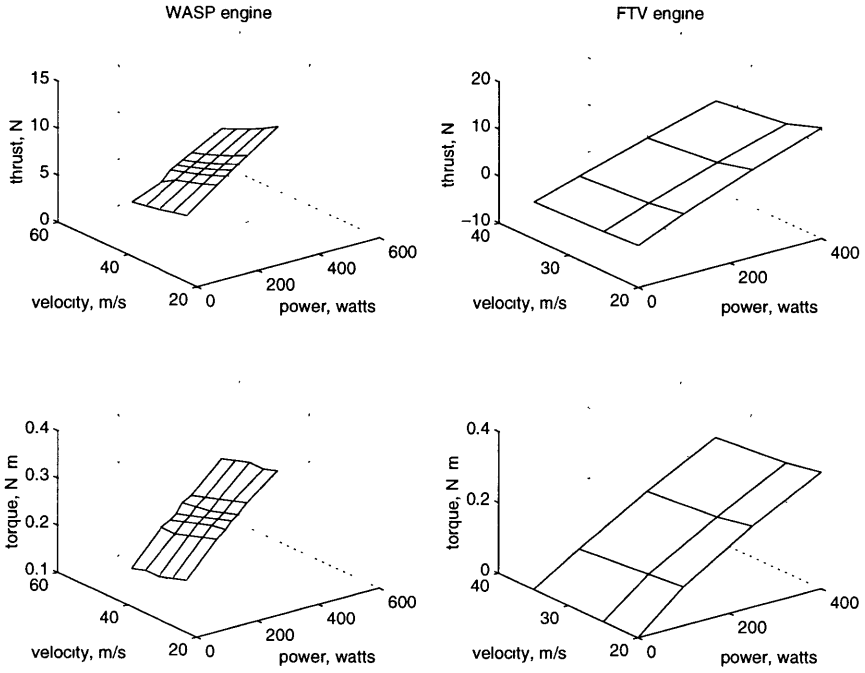

Figure 3-5: Engine thrust and roling torque for WASP and FTV.

3.5.5 Engine Model

A simplified propeller model was built using a computer program called Xrotor, available on Athena (MIT computer network), which employs a potential flow calculation method. Aircraft velocity and power setting are given as inputs to the program, and thrust and rolling torque are produced as outputs. This is a fairly crude model for the engine, but it will suffice for longitudinal stability analysis, because the only mode affected by powerplant properties is a slow phugoid mode, which is not crucial for flying qualities. The WASP and FTV engine models are given here as two dimensional tables which are illustrated in Figure 3-5.

3.5.6 Actuator Models

Simple actuator models, based on slew rate data for standard radio control model servos were assumed for both vehicles. Each of the differential stabilizers was modeled as follows:

- first order low pass filter with bandwidth of 15 rad/sec

- rate limit of 200 deg/sec

- deflection limits of ± 10 degrees

The bandwidth of 15 rad/sec was chosen according to

$$\text{actuator bandwidth} \approx 1.5 \frac{\text{full range}}{\text{slew rate}} \tag{3.25}$$

3.5.7 Sensor Models

The position of the inertial measurement unit was determined for each vehicle, and accelerometer measurement equations were defined. For stability augmentation purposes, the important sensor properties are bandwidth, non-linearities, and noise. For attitude determination and navigation, biases are also important. The following properties were modeled for both systems: first-order anti-aliasing filter, noise, saturation at full scale and short term biases. The sources of noise are A/D board quantization, EM noise in wires and inherent sensor noise, all of which is modeled as band-limited white noise. Short term bias is usually modeled as a first order Markov process with a large time constant. In cases when the time constant for bias stability is much larger than mission time, a random constant model is assumed. Short term bias is a residual bias after pre-calibration. If it is not specified in manufacturer specifications, it must be measured. Three sets of sensors were modeled:

- micromechanical inertial sensor assembly (MMISA), projected for use in WASP

- Systron-Donner Motion Pak, used in FTV

- Crossbow DMU-6, used in FTV

Biase and noise 1 σ rms value estimates for these units are given in table 3.2. Other important considerations for sensor selection are highlighted in Section 2.2.

Error	Units	Crossbow	Motion Pak	MMISA
gyro bias	deg/sec	0.01	0.004	0.003
gyro noise	deg/sec	0.1	0.1	0.1
accel bias	mg	5	1	1
accel noise	mg	5	1	1

Table 3.2: IMU Models

3.5.8 Gust Models

Stochastic gust models were taken from [5]. The gust models take into account the dependence of the probability of gust occurrence on altitude. The estimate of probability of encountering gusty air at sea level, given in [5], is 80%. The root mean square gust velocity is assumed to be normally distributed with standard deviation $\sigma = 2.3$. Gust levels of 1 σ were assumed for the model. Forward, vertical, and lateral velocity gusts in aircraft body axis, as well as rolling gust were modeled as first and second order Markov processes. The gust model shaping filters take into account altitude, wing span and airspeed. These models were based on larger aircraft, and their validity for a small aircraft as WASP or FTV is uncertain. The frequency content of rolling gusts seems to be exaggerated according to this model. Thus, the same shaping filter was used for rolling gusts and side velocity gusts. Shaping filters for gusts are given in Fig 3-6.

3.5.9 Trim Analysis and Linearization

The non-linear 6-DOF simulation of the aircraft dynamics was built using Simulink S-functions, and standard Simulink blocks. A detailed description of the simulation is given in Appendix A. It was assumed that the aircraft does not enter steep turns (bank angle of less than 30 degrees), and it is sufficient to consider linearization around steady level flight trim condition. Thus, only the longitudinal equations had to be trimmed. A trim routine was developed using Matlab optimization toolbox function 'fmins'. The $\|\cdot\|_2$ norm of the scaled vector of derivatives of longitudinal states was chosen as a cost function. The results were verified by running the simulation with

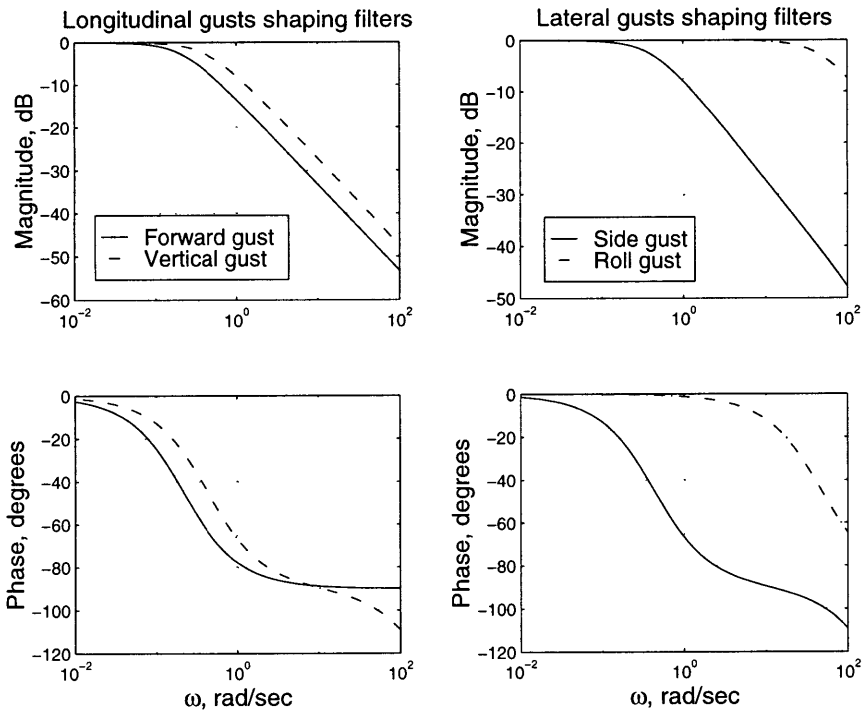

Figure 3-6: Shaping filter for gust inputs

trim inputs, and observing steady trim output states. Initial conditions for the state space blocks representing actuator bandwidth were chosen to be actuator trim states to avoid transients.

It was found that tuning Simulink internal 'trim' function is a lengthy and unreliable method. The reason is that 'trim' function uses a constrained optimization method, which is a numerically sensitive procedure. Incorporating constraints implicitly and using a non-constrained optimization method like 'fmins' leads to better convergence.

Steady state level flight trim points were found for several cg locations for both vehicles. The trim values for most forward and most aft cg locations are given in Table 3.3.

The FTV center of gravity is inside its fuel tank, so the FTV cg excursion is quite small. For both vehicles, actuator deflection does not exceed 6 degrees, and control surfaces stall occurs at 11 degrees. The wing incidence angle is 7 degrees, which is only 4 degrees below stall. This high incidence angle of the wings was chosen to keep

Parameter		Units	WASP		FTV	
X_{cg}	cg location forward of base	cm	20.3	20.0	25.7	25.6
W_{fuel}	fuel weight	kg	0.07	0	0.02	0
T	throttle	Watts	301.57	299.86	192.34	192.0
$i_r = i_l$	actuator deflection	deg	4.89	4	5.15	4.82
α	vehicle angle of attack	deg	0.53	0.45	0.53	0.50

Table 3.3: WASP and FTV Trim Conditions

the trim angle of attack of the fuselage low in order to reduce drag. These conditions impose significant constraints on the longitudinal control system.

The non-linear equations of motion were numerically linearized around trim points using Simulink linearization program 'linmod'. To insure correct results minimal realization of state space model given by the linearization routine was found, before obtaining transfer functions from the state space model. The numerically derived transfer functions were compared with analytical approximations, thus providing a check on implementation of the simulation. Also, the modes and frequencies were analyzed from a physical point of view, which helped to gain insight into aircraft dynamics. The linearized dynamics of WASP and FTV were compared, and it is shown in the following section that the two vehicles' natural frequencies are very close, with FTV slightly more heavily damped than WASP.

3.5.10 Dynamics of Operational and Prototype Vehicle

According to the model, the FTV dynamics is very close to that of WASP due to the combination of lower speed and higher linear scale factor for the FTV. The factors that make the models dynamically similar are stated in this section.

The linear aircraft dynamics are completely described by dimensional stability derivatives. For example, consider the crucial M_α derivative:

$$M_\alpha = \sqrt{C_{m_\alpha} \frac{\overline{q} S \overline{c}}{I_{yy}}} \tag{3.26}$$

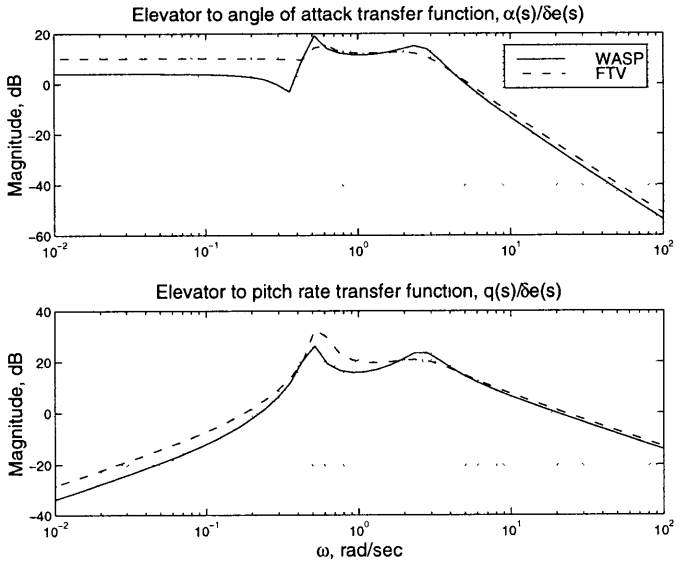

Figure 3-7: Open loop WASP and FTV longitudinal dynamics

The ratio of M_α for the WASP and for the FTV is

$$\frac{{M_\alpha}^{WASP}}{{M_\alpha}^{FTV}} = \frac{V^{WASP}}{V^{FTV}}\sqrt{\kappa^3 {I_{yy}}^{FTV}/{I_{yy}}^{WASP}} = \frac{37.5}{24.8}\sqrt{(1/1.28)^3 0.092/0.11} = 0.95 \tag{3.27}$$

where κ is an inverse of scale-up factor for FTV. Thus inertia tensor elements provide some freedom in tuning the dynamics of FTV by appropriately distributing some of its mass. Figure 3-7 shows the comparison of longitudinal WASP and FTV dynamics for the most forward cg location in both vehicles. It can be seen that short period frequencies are closely matched while the phugoid frequency of FTV is larger since it flies slower, which is consistent with the phugoid approximation. Both transfer functions are minimum-phase, thus magnitude plots convey all pertinent information.

The FTV dutch roll and spiral modes are close to those of WASP, because of the same kind of scaling reasons described above for the longitudinal dynamics. The roll subsidence pole, however, is significantly faster for FTV, because the FTV's wings are made of carbon fiber, and are considerably lighter than the WASP wings, which are aluminum. Rolling inertia of WASP is three times larger than for the FTV. The important modes for lateral stability are dutch roll and spiral, thus FTV fulfills its purpose of simulating WASP dynamics in terms of these modes. Figure 3-8 shows

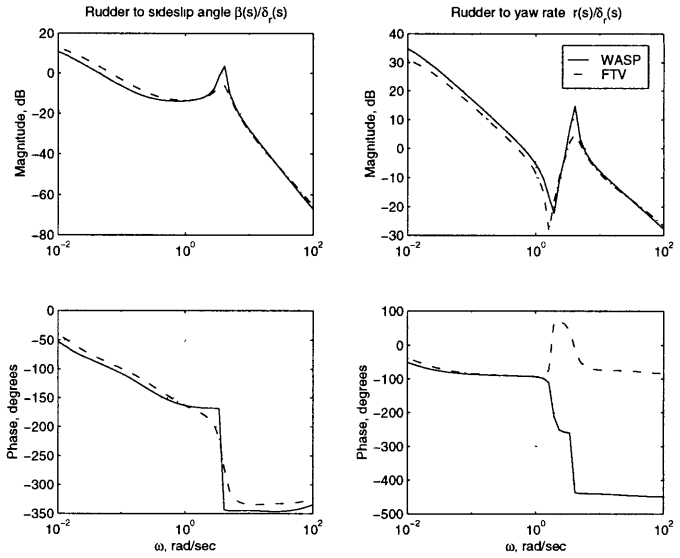

Figure 3-8: WASP and FTV open loop lateral dynamics

transfer functions from rudder (i.e. differential stabilizer deflection) to yaw rate and sideslip angle for WASP and FTV. The discrepancy in the phase plot of $r(s)/\delta_r(s)$ transfer function is a numerical artifact, caused by 360 degrees phase wraparound.

3.6 Control System Design

3.6.1 Goals and Challenges

Gust disturbance versus inertial attitude stiffness

For a small aircraft control system the major challenge is design to accomodate gust disturbances. Vertical and side gusts are only an order of magnitude smaller than the speed of the vehicle, thus α and β responses to gusts are severe. To make a gust rejection control system, the natural ability of aircraft to turn itself into the wind (weathercock stability) must be augmented. Thus, the goal is to keep aircraft attitude, with respect to wind axes, constant. For surveillance purposes it is desirable to keep aircraft attitude with respect to inertial space fixed. These goals obviously contradict each other, and a reasonable trade off must be achieved. The first priority for any aircraft is a stable flight, so it may be necessary to use an actively controlled

gimbal system to stabilize the camera.

Surface allocation

For both WASP and FTV, large angle of attack responses to vertical gust inputs are prohibitive, because they cause the vehicles to fly in a near-stall regime. FTV is designed to fly slower and is more vulnerable to gusts than WASP. The vehicles do not have ailerons, thus there are only two independent control effectors to control longitudinal, lateral and directional control variables. It is natural to decouple the longitudinal channel from lateral and directional by considering only symmetric and antisymmetric actuator inputs. Thus, "elevator" and "rudder" commands are calculated and then allocated to physical surfaces:

$$\delta_e = \frac{i_r + i_l}{2} \quad \delta_r = \frac{i_r - i_l}{2} \tag{3.28}$$

3.6.2 Prototype Vehicle Command Augmentation System (CAS)

Longitudinal CAS

The choice of a longitudinal control variable depends on the pilot's perception of the vehicle's motion, i.e. the variable that the pilot observes. The prototype vehicle is a remotely piloted aircraft, and the pilot observes the vehicle from the ground. It is assumed that the pilot observes flight path angle, which for a small angle of attack is close to pitch attitude. Unless the vehicle is very agile, the pilot prefers to command the rate of the state, not the state itself. For a highly maneuverable RC aircraft, a pilot might prefer controlling pitch attitude directly. FTV is a sluggish aircraft, thus pitch rate was chosen as the longitudinal control variable. Another reason for this choice is turn entry, for which the appropriate non-zero pitch rate should be commanded.

Phasor diagrams for the short period and phugoid mode of WASP are shown in Figure 3-9 and Figure 3-10. These diagrams correspond to the trim condition with

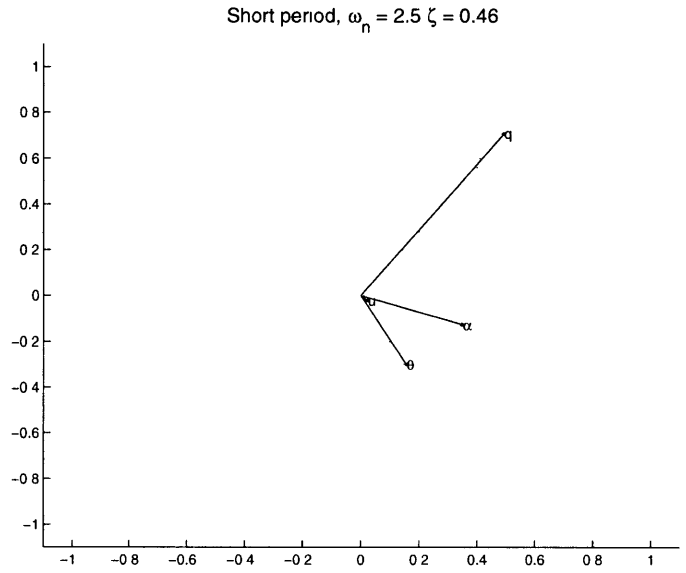

Figure 3-9: Short period phasor diagram

the fuel tank full. In the longitudinal phasor diagrams, the component corresponding to forward velocity was divided by the trim value U_0 to have non-dimensional scaling, so as to be consistent with other variables. As can be seen, the modes are consistent with traditional trends for aircraft: the short period dynamics are concentrated in pitch rate q, angle of attack α, and pitch angle θ, with almost no effect on forward velocity. Similarly, the phugoid dynamics are exhibited in pitch, forward velocity and pitch rate, with almost no contribution to angle of attack.

The light damping ($\zeta = 0.26$) of the short period mode can be accounted for by the small tail area and short moment arm between c.g. and tail surfaces. The natural frequencies of phugoid and short period are separated by less than a decade. This can be explained using phugoid and short period approximations. The phugoid frequency can be approximated as

$$\omega_{n_p} = \sqrt{\frac{2gC_{m_a}C_L}{\frac{1}{2}\bar{c}C_{m_q}C_{L_a} + 2mC_{m_a}/(\rho S)}} \tag{3.29}$$

For WASP, which is a slow and heavy vehicle, the second term in the denominator, containing mass, dominates, and the following approximation can be used:

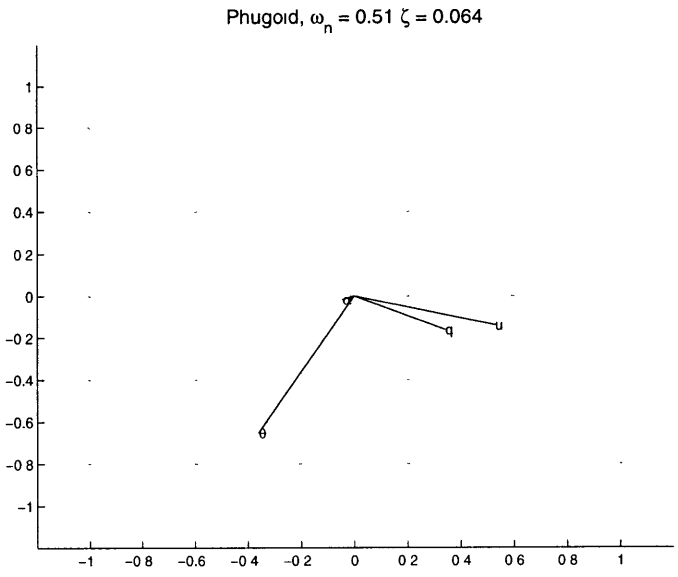

Figure 3-10: Phugoid phasor diagram

$$\omega_{n_p} = \sqrt{\frac{g\rho S}{m} C_L} \tag{3.30}$$

Substituting for lift coefficient at level flight $C_L = m/(\bar{q}S)$, yields a simple approximation for phugoid frequency:

$$\omega_{n_p} = \frac{g}{V_T}\sqrt{2} \tag{3.31}$$

For the nominal FTV model, Equation 3.31 yields a phugoid frequency of 0.5 rad/sec. This approximation shows that phugoid frequency depends mainly on airspeed. On the other hand, natural frequency for short period approximation is

$$\omega_{n_{sp}} = \sqrt{\frac{M_q Z_\alpha}{V_T} - M_\alpha} \tag{3.32}$$

The M_α term dominates in the above expression for WASP and it can be further simplified to

$$\omega_{n_{sp}} \approx \sqrt{-M_\alpha} = \sqrt{-C_{m_\alpha} \frac{\bar{q}S\bar{c}}{I_{yy}}} \tag{3.33}$$

Since vehicle velocity V_T is relatively small, and pitching inertia I_{yy} is relatively

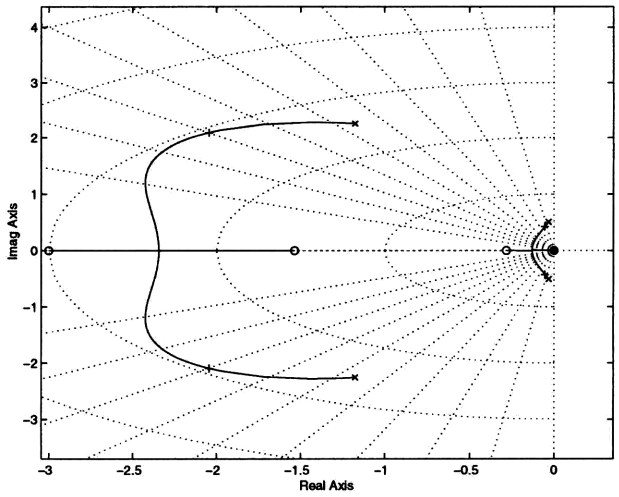

Figure 3-11: Root locus for inner pitch rate loop with lead, $K_q = 0.06$

large, the nominal FTV parameters yield a short period frequency approximation of 2.5 rad/sec, which is only 5 times higher than the phugoid frequency. Hence, the controllers should be designed using the complete longitudinal dynamics. The high phugoid frequency gives an opportunity to use pitch rate and vertical accelerometer feedback for stabilization of the phugoid mode.

In designing the longitudinal controller the pitch rate SAS loop was closed first to augment the short period frequency and damping. A lead compensator, with zero at 3 rad/sec and pole 4.5 rad/sec, was added in the feedback path to change the root locus (Figure 3-11), and gain was chosen to put the short period poles of the augmented system at $\omega_n = 4.0$ rad/sec, with a damping ratio of $\zeta = 0.8$.

The n_z SAS loop was closed next. The purpose of n_z feedback is to increase aircraft angle of attack stiffness. The accelerometer measurement for low-speed aircraft contains mostly u and α. Consider the n_z measurement equation:

$$n_z = \frac{F_z}{m} - x_a\dot{Q} = a_z - x_a\dot{Q} \tag{3.34}$$

where x_a is location of the accelerometer in front of the cg, and a_z is the specific force at the cg. The vehicle is trimed at low θ and α, thus neglecting the difference

between stability and body axes,

$$n_z \approx \frac{C_L \bar{q} S}{m} - x_a C_m \bar{q} S \bar{c} \tag{3.35}$$

Using equations for C_L and C_M from section 3.5.4, linearizing about the trim point and using dimensional derivatives, obtains

$$n_z \approx (Z_\alpha - x_a M_\alpha)\alpha + (Z_q - x_a M_q)q + \frac{2g}{U_0}u + (Z_{\delta_e} - x_a M_{\delta_e}) \tag{3.36}$$

The term $2g/U_0$ is significant for a low-speed vehicle. The fact that u is observable in accelerometer measurement, and the relatively high frequency of the phugoid mode make it possible to use n_z SAS feedback to suppress gust disturbances in the phugoid frequency range. In this case $Z_{\delta_e} - x_a M_{\delta_e} > 0$, and with this position of accelerometer the $n_z(s)/\delta_e(s)$ transfer function has a non-minimum phase zero very close to the origin, which puts a limitation on the controller bandwidth. In addition the short period poles increase in frequency and decrease in damping as n_z feedback gain is increased. To avoid adverse effect of positive n_z feedback on the short period mode, a lag compensator was added in feedback path. The lag compensator will also increase the stiffness with respect to forward and vertical velocity gusts at low frequency. Figure 3-12 shows the root locus of $n_z(s)/\delta_e(s)$ transfer function augmented with the lag compensator. The figure illustrates the stabilizing effect on phugoid poles and the adverse effect on short period poles.

Finally the pitch CAS loop was closed with two cascaded proportional plus integral compensators in the forward path to provide needed steady state performance. Figure 3-13 shows the improvement in stability and steady state tracking of the pitch rate command.

The pitch rate pulse response is a bit sluggish, but well damped (Figure 3-14). The vertical gust disturbances at high frequencies are well supressed by the pitch rate feedback, and the low frequency component is attenuated by n_z feedback (Figure 3-15).

The diagram of pitch rate CAS is given in Figure 3-16. It should be noted that for practical implementation the steady component $g_0 \cos\theta \cos\phi$ should be subtracted

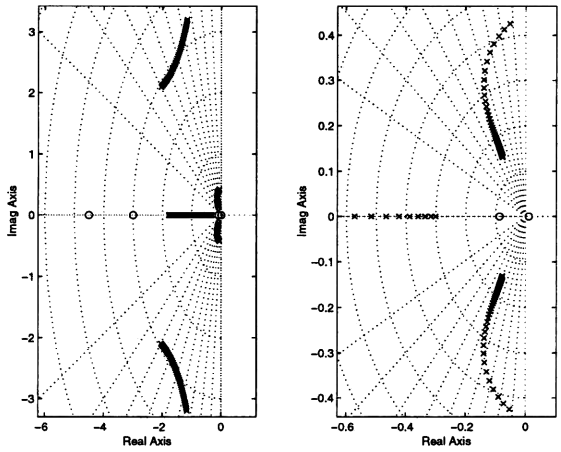

Figure 3-12: Root locus for $n_z/\delta_e(s)$ transfer function with lag, $K_{nz} = 0.015$

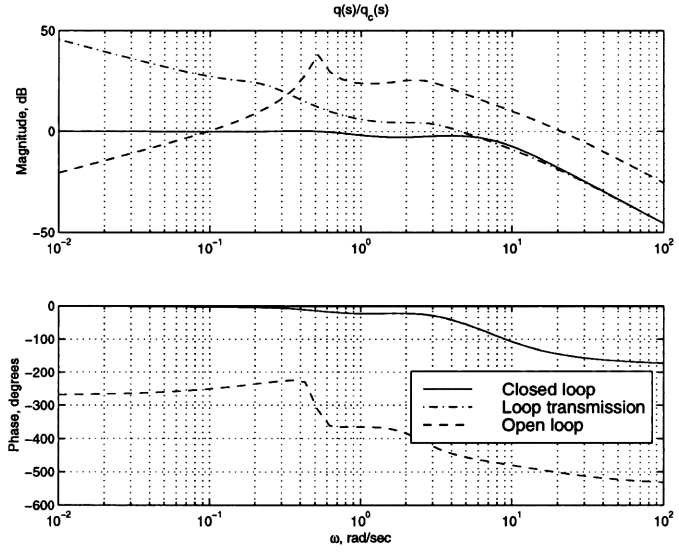

Figure 3-13: Open loop, closed loop and loop transmission for $q(s)/q_{cmd}(s)$ transfer function

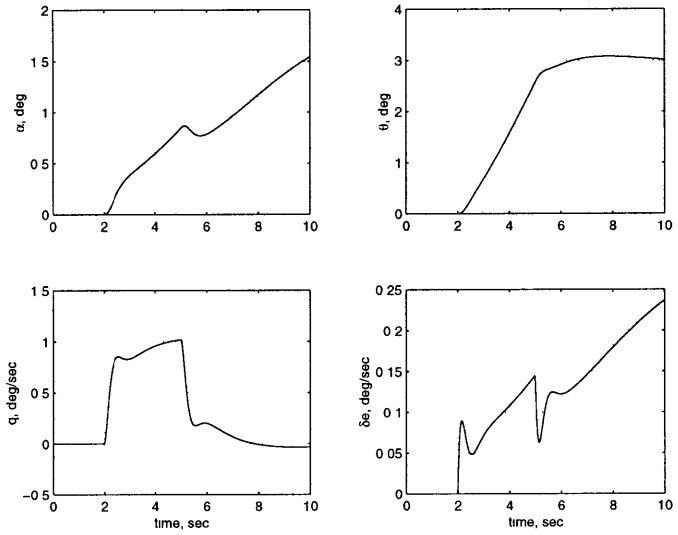

Figure 3-14: Pitch rate pulse response with CAS-on

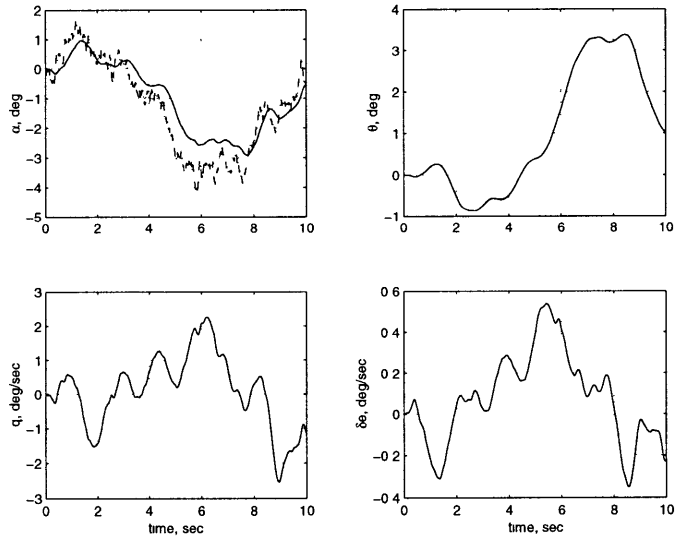

Figure 3-15: Vertical gust response with CAS-on

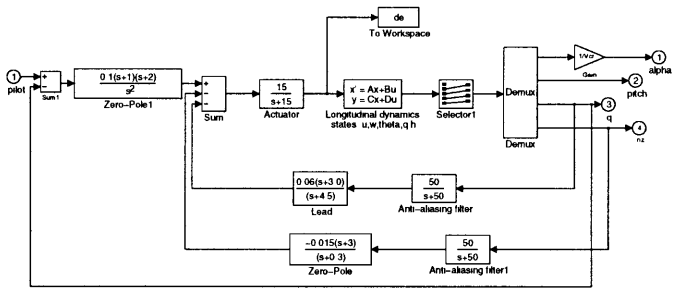

Figure 3-16: Pitch CAS

from the accelerometer measurement to assure a linear response. It requires attitude information. Such information can be provided by integrating kinematic equations for pitch and roll angles, and correcting these estimates with the low-pass filtered accelerometer measurements. The design of the attitude estimation filter is given in section 3.7.

Lateral CAS

The lateral-directional dynamics does not change much with a change of cg location, and roll angles are considered to be small. Thus only one linearized model, corresponding to a steady level flight, should be considered for the lateral CAS design. The linear plant dynamics has lightly damped dutch roll mode, neutrally stable spiral, and fast roll mode.

The stability augmentation problem will be considered first. Looking at the phasor diagrams for the dutch roll mode in Figure 3-17, it is clear that this mode consists primarily of roll rate and roll angle response, with considerably smaller sideslip and yaw rate responses.

This is due to the vehicle's large yawing inertia, and the small area and short moment arm of the vertical tail. Still, there is a sufficient amount of yaw rate to use the standard yaw rate to rudder feedback for damping the dutch roll mode. The root locus of $r(s)/\delta_r(s)$ transfer function is shown in Figure 3-18.

The proportional controller in the feedback path will suffice, and the gain was chosen such that the closed loop dutch roll poles have frequency of $\omega_{n_d} = 3.0$ and damping ratio of $\zeta = 0.7$. A washout circuit with a time constant of two seconds was

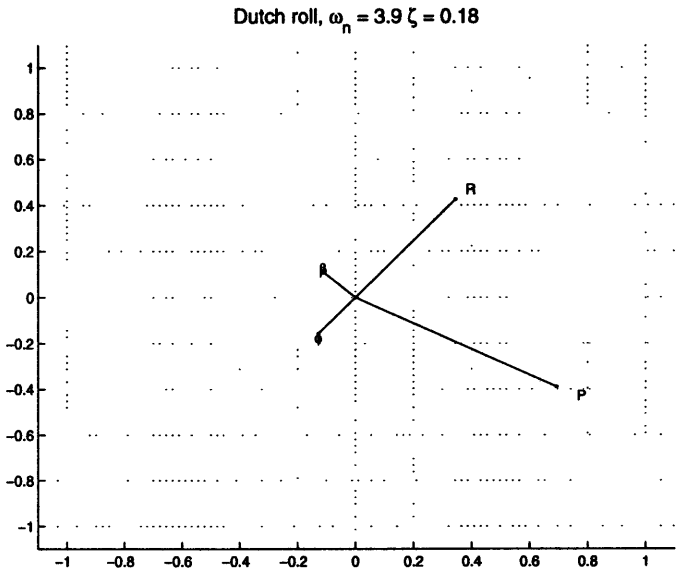

Figure 3-17: Dutch roll phasor diagram for the FTV

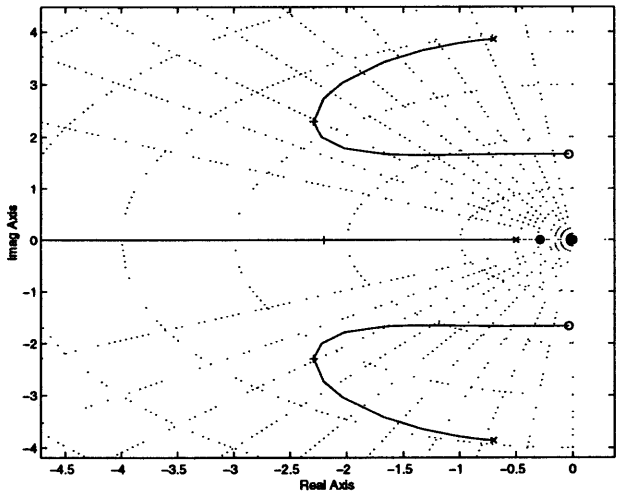

Figure 3-18: Root locus for $r(s)/\delta_r(s)$ transfer function, $K_r = 1.3$

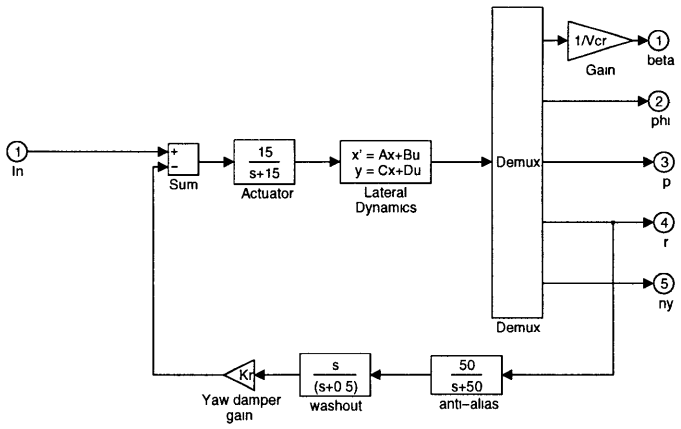

Figure 3-19: Lateral SAS

added in the feedback path to prevent the controller from resisting steady-state pilot commands. The lateral SAS is shown in Figure 3-19. The gain K_r will be tuned in flight to compensate for errors in stability derivatives estimates.

The yaw rate feedback also damps out high-frequency response to lateral gust inputs. The lateral acceleration feedback can be used to further stiffen the system with respect to gusts. According to the gust model the gust shaping filter rolls off fast enough so that with the yaw damper in the system there is mostly low-frequency gust response.

Since the vehicle does not have ailerons it enters a turn by exciting the dutch roll with a differential pulse on the stabilizers. This makes the vehicle sideslip, roll and yaw. Consider a positive turn, right wing down. To enter such a turn, a pulse command is given to differential stabilizers. Left actuator angle of attack (AoA) is increased, and right actuator AoA is decreased by the same amount to keep the lateral and longitudinal dynamics decoupled. This creates a positive rolling moment and negative sideslip due to anhedral on the tails. The dihedral on the wings makes the vehicle turn to the right. There is an adverse yaw effect due to increased induced drag on the left tail. The roll rate CAS is not useful in this case because there are no ailerons. The stabilizer pulse response is shown in Figure 3-20.

A steady turn requires steady pitch rate. Both can be calculated from a steady

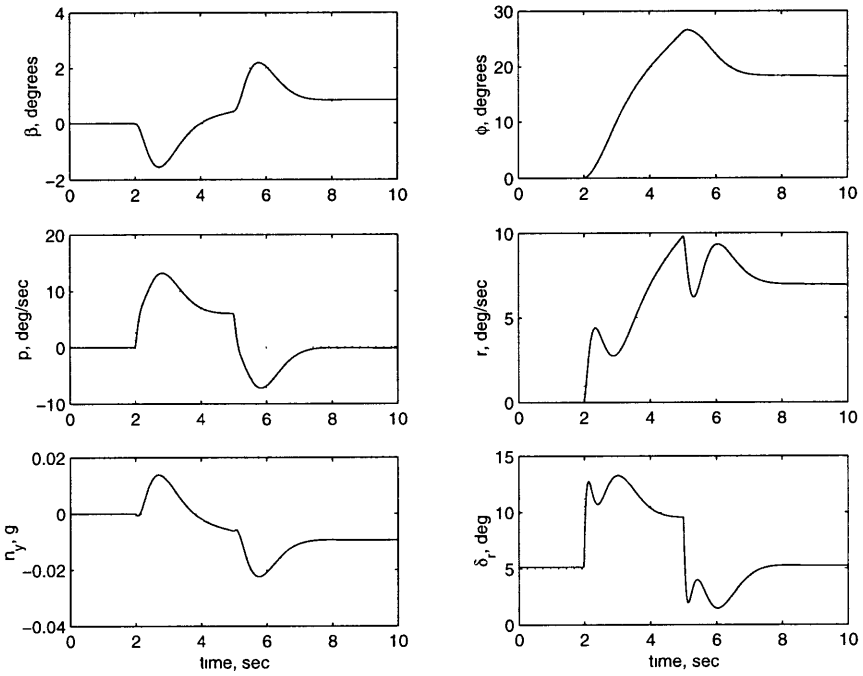

Figure 3-20: Differential stabilizer pulse response

state load factor, or roll angle ([28, vol. 1, page 226]):

$$\begin{aligned}
&\dot{\psi} = \tfrac{g\tan\phi}{V_T} && \text{turn rate}\\
&n = L/W = 1/\cos\phi && \text{load factor}\\
&Q = \tfrac{g\tan\phi\sin\phi}{V_T} = \tfrac{g}{V_T}\left(n - \tfrac{1}{n}\right) && \text{pitch rate}\\
&R = \tfrac{g\sin\phi}{V_T} = \tfrac{g}{nV_T}\sqrt{n^2-1} && \text{yaw rate}
\end{aligned} \tag{3.37}$$

The roll angle is available from the attitude determination system (Section 3.7). This estimate should be low pass filtered, and the corresponding pitch rate fed as an input into the pitch rate command channel. The low pass filter time constant should be chosen in such a way that the roll angle (ϕ) crossfeed does not affect stability, and at the same time the turn entry is fast enough. This constant can also be tuned in flight.

3.7 Navigation and Attitude Determination Systems

3.7.1 Gravity Aiding with Strapdown Sensors

Gravity aiding is an attitude determination method that employs angular rate and translational acceleration information from a strapdown IMU. Most small UAVs are equipped with strapdown, as opposed to gimballed, IMUs. The gravity aiding approach was used for the prototype vehicle. When the IMU is used in a stand alone mode the gravity aiding is the only way to provide attitude information needed for autopilots. Other applications may include a backup vertical determination system, which is run in parallel with a master system, and provides accurate attitude information for flight stabilization in case of failures of other sensors. For example an unmanned helicopter considered in Chapter 2 has an extended Kalman filter (EKF) as a master filter, which uses GPS, IMU, compass and sonar measurements. In case of loss of GPS track, the attitude estimate will diverge unless frequency characteristics of sensor models are taken into account in EKF design. The system may temporarily switch to a gravity aiding estimator, engage pitch attitude hold and bank angle hold autopilots, and maintain steady level or hovering flight until GPS is reacquired.

The approach uses the same idea as a vertical gyro in a regular aircraft, in which a restoring torque, proportional to accelerometer measurements, is used to correct the low frequency gyro drifts. In case of a strapdown unit, the role of the vertical gyro is played by a set of three rate gyros with orthogonal axes. The angular rate measurements are used in the attitude kinematic equations to solve for Euler angles. This provides high-bandwidth attitude information. The gyros have drifts, and the low pass filtered accelerometer measurements are used to correct for them. The low pass filter time constant should be big enough to filter out the vehicle dynamics from accelerometer measurements.

Frequency and time domain design approaches for the gravity aiding attitude reference system are considered in [4]. The time domain design is based on an EKF.

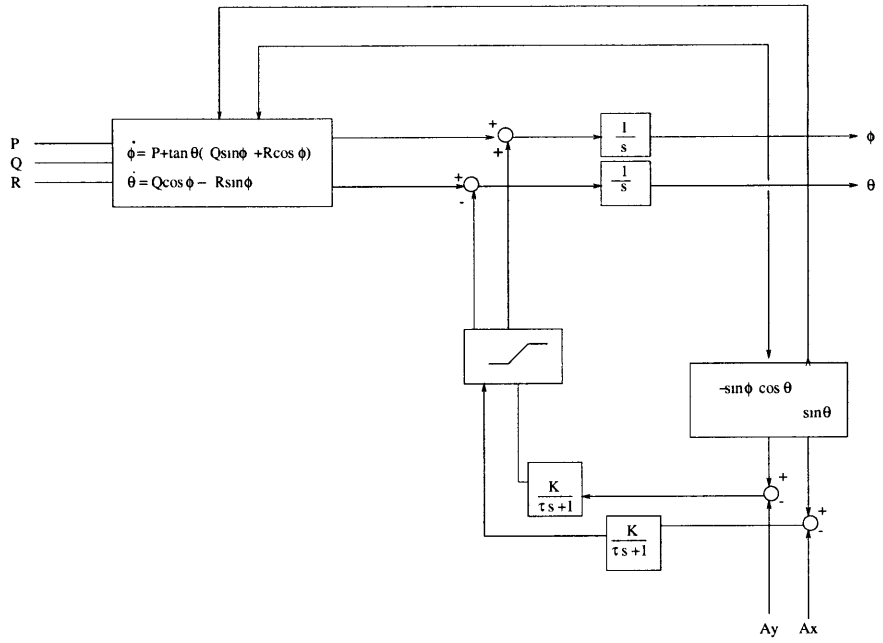

Figure 3-21: Attitude determination system with gravity aiding

It involves modeling the error components for the gyros and accelerometers, and results in higher complexity and greater flexibility due to a number of parameters available for tuning. It is shown in [4] that in this case a low-order frequency domain design is easy to tune and provides adequate results. The frequency domain design method was chosen for the application to FTV. The block-diagram of the system is given in Figure 3-21. Note positive feedback of the roll angle correction. This is due to the fact that positive roll angle induces negative side acceleration: $a_y \approx -g\phi$. This sign was flipped in the paper, and the sign bugs are the hardest to catch!

Assuming that the roll and pitch angles are small, the kinematic equations can be linearized to allow a linear analysis of the filter. The linearized filter equations are decoupled for roll and pitch channels. The channels are the same, thus the low pass filters $F_\phi(s)$ and $F_\theta(s)$ can be chosen to be the same as well.

$$F_\phi(s) = F_\theta(s) = \frac{K}{\tau s + 1} \tag{3.38}$$

To choose the parameters K and τ [4] considers important error sources. The biases and scale factors are significant for both sensors. Since accelerations are used

only to correct for a very slowly varying gyro bias, the vehicle dynamics in the accelerometer measurements is also an error source for this application. The white noise was also added to both sets of sensors. Detailed account of error modeling for the IMU is given in Section 3.5.7. From the linearized model analysis, the suggested values for the low pass filter parameters are:

$$\tau = \frac{1}{K} \tag{3.39}$$

$$K = \frac{\epsilon_0}{\delta\theta_\omega} \tag{3.40}$$

where ϵ_0 is assumed gyro bias value, and $\delta\theta_\omega$ is the specified target value for the steady state pitch angle error due to ϵ_0. There is no steady state error due to initial error. The values τ and K were optimized with respect to steady state performance. This design also guarantees -40 dB/decade slope for the transfer function from acceleration to pitch estimate error, thus rejecting high frequency accelerometer errors. Effectively, there is a usual trade off between high frequency noise rejection and bias rejection.

For the Crossbow IMU used in FTV the short term (100 sec) bias stability was estimated at 0.01 deg/sec. The accelerometer bias is on the order of 5 mg, which results in 0.3 degrees attitude error. Thus, designing for a steady state error due to gyro bias $\delta\theta_\omega = 0.3$ deg the low pass filter cutoff frequency is 0.03 rad/sec, which is two decades below the aircraft short period frequency. This should give a sufficient attenuation of vehicle dynamics. In case the aircraft undergoes a steady turn or acquires significantly non-zero steady pitch attitude, the algorithm will lead to steady state estimation error. To account for it, the amount of accelerometer feedback should be limited to 1-2 degrees. This is analogous to the erection cutoff mechanism in a vertical gyro, where there is a limit on the restoring torque.

The aircraft response to pulse commands in both channels was simulated, and Figure 3-22 shows that the estimator tracks high frequency dynamics due to pulse inputs, and the accelerometers correct long term drifts due to gyro biases. This is conspicuously seen in the pitch response in this case.

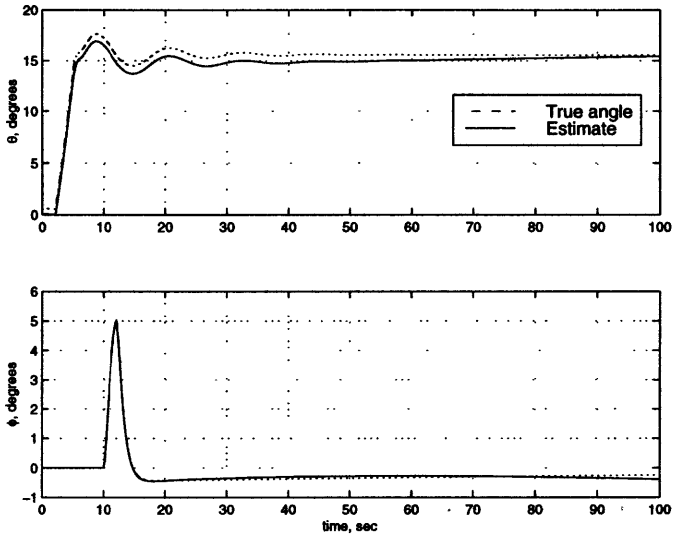

Figure 3-22: Gravity aiding algorithm performance

3.7.2 Velocity Matching

GPS information can also be used for attitude determination. The idea of the velocity matching algorithm is to compare INS computed velocity with the velocity measured by GPS, and extract attitude information from the comparison. The method can be employed in the operational WASP system for an in-flight attitude and heading angles determination.

The velocity matching method uses linearized error equations, thus initial rough estimate of vehicle's attitude is needed. To provide this estimate, GPS-only algorithm for attitude determination is used. Assume that the aircraft angle of attack and angle of sideslip are small, then the heading and pitch angle can be determined from GPS velocity information, which gives information on flight path angle. Single antenna GPS receiver was used for roll angle determination in [21]. The algorithm is based on the derivation of inertial acceleration of the vehicle from the GPS velocity information by using the third-order Kalman filters. This GPS-only alignment technique will provide up to several degrees accuracy in all axes in less than 5 sec. Then GPS velocity matching can be used to assure fine alignment.

Design guidelines for velocity matching method are given in [34]. Non-linear INS equations are initialized with GPS-derived position, velocity and attitude. Velocity

and attitude satisfy:

$$\begin{aligned} \dot{v_e^n} &= C_b^n f^b - [2\omega_{ie}^n + \omega_{en}^n] \times v_e^n + g_l^n \\ \dot{C_b^n} &= C_b^n \Omega_{nb}^b \end{aligned} \tag{3.41}$$

where

- v_e^n – vehicle velocity with respect to Earth in NED frame
- C_b^n – transformation matrix from body frame to NED frame
- f^b – accelerometer measurements
- ω_{ie}^n – Earth angular rate in NED frame
- ω_{en}^n – NED frame angular rate relative to Earth-fixed axis in NED frame

At a nominal WASP velocity of 50 m/s, $\| [2\omega_{ie}^n + \omega_{en}^n] \times v_e^n \| < 0.3$ mg. This is below accelerometer bias, thus these terms can be omitted. Then the linearized INS error equations are used to design a Kalman filter. It is important to model the gyro biases as states of Kalman filter. This "tells" the filter that the low frequency information from the gyros is biased, and the filter should assign more weight to the GPS and accelerometer measurements at low frequencies. In this sense the KF is similar to the kind of complementary filter approach employed in design of the gravity aiding algorithm of Section 3.7.1. Dynamic driving noise due to accelerometer and gyro high-frequency errors should be modeled as well. This noise is relatively small compared to the high frequency noise of GPS, but the introduction of the dynamic driving noise eliminates possible singularity in the covariance matrix. For applications where pre-calibration of the accelerometers is impossible, such as aerial deployment after gun launch, the accelerometer bias states should be included in the filter to achieve in-flight calibration. The biases are modeled as first order Markov processes with a large time constant. If the time constant for the bias term is much larger than the mission time, they can be modeled as random constants.

The filter design is given below. Let C_b^n be a transformation matrix from local NED frame into body frame, so that for any vector ξ: $\xi^n = C_b^n \xi^b$. Define $\hat{C}_b^n = [I - \Psi] C_b^n$ - current estimate of transformation matrix. Ψ is a small angle rotation

matrix:

$$\Psi = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix}$$

where $\delta\alpha$, $\delta\beta$, $\delta\gamma$ attitude errors. The linearized error equations for attitude states are:

$$\dot{\psi} = -\omega_{in}^{n} \times \psi - C_b^n \omega_b - C_b^n \xi_g \tag{3.42}$$

where

- ψ – attitude error
- ω_{in}^{n} – NED frame rate
- ω_b – gyro biases
- ξ_g – gyro noise

Because the vehicle is moving slowly, $\| \omega_{in}^{n} \| \approx 14$ deg/hr, which is below the MMISA sensor drifts, angular rate of the NED frame can be neglected. The accelerometers and GPS will be the primary sensors providing the low-frequency attitude information. Thus the attitude error equation can be reduced to:

$$\dot{\psi} = -C_b^n \omega_b \tag{3.43}$$

The error equations for velocity states are:

$$\delta\dot{v} = f^n \times \psi + C_b^n a_b + C_b^n \xi_a \tag{3.44}$$

where

- f^n – accelerometer measurements resolved in NED frame
- a_b – accelerometer biases
- δv – north and east velocity errors in NED frame
- ξ_a – accelerometer noise

The equations for bias states are

$$\dot{a_b} = 0 \quad \dot{\omega_b} = 0 \tag{3.45}$$

Thus, Kalman filter states are:

$$\delta x = \begin{bmatrix} \delta\alpha & \delta\beta & \delta\gamma & \delta v_e & \delta v_n & \omega_{bx} & \omega_{by} & \omega_{bz} & a_{bx} & a_{by} & a_{bz} \end{bmatrix}$$

The dynamic equations are 3.43, 3.44 and 3.45, which should be represented in a matrix form for KF solution. A diagonal 11×11 covariance matrix, for dynamic driving noise, should be built using specifications for gyro noise and accelerometer noise.

The measurement equations are

$$z = \begin{pmatrix} V_e^{INS} - V_e^{GPS} \\ V_n^{INS} - V_n^{GPS} \end{pmatrix} = \begin{pmatrix} \delta v_e \\ \delta v_n \end{pmatrix} + \begin{pmatrix} \eta_e \\ \eta_n \end{pmatrix}$$

where η is the GPS velocity measurement noise. The noise covariance is $R = \sigma_\eta^2 I_{2\times 2}$. For P(Y) code, $\sigma_\eta = 0.01$ m/s. An initial error covariance matrix is determined by ambiguity in initial estimates of corresponding states. Then a continuous time KF model is discretized and split into measurement update and time propagation equations. The simulation shows that the pitch and roll angle estimate convergence is dependent primarily on the accelerometer biases. To estimate the yaw angle, the aircraft has to make a lateral-directional maneuver to excite error dynamics, so that it becomes observable in the inertial velocity measurements. To make this point clear, consider scalar form of 3.44.

$$\begin{aligned} \delta\dot{v}_N &= -f_D\delta\beta + f_E\delta\gamma + a_{b_N} \\ \delta\dot{v}_E &= f_D\delta\alpha - f_N\delta\gamma + a_{b_E} \end{aligned} \tag{3.46}$$

If the aircraft is flying straight and level, the apparent acceleration vector is aligned with down axis in NED frame: $f_E = 0$ and $f_N = 0$, hence $\delta\gamma$ becomes unobservable

in the velocity measurement. To make heading error observable, a lateral-directional maneuver is needed. In the example, given in [34], a .1 g weave maneuver with 10 deg/hr gyro bias and 2 mg accelerometer bias results in subdegree level accuracy in all three axis. The gust disturbances may be sufficient to excite error dynamics, then no maneuver is needed to guarantee convergence.

3.7.3 Navigation System for Operational Vehicle

A standard EKF for navigation is suggested, using the IMU as a master sensor and GPS position and velocity information for measurement updates. Depending on the quality of micromechanical inertial sensors, various error sources should be modeled. As it currently exists, scale factor errors and short term biases should be modeled for both gyros and accelerometers. A standard tightly coupled INS/GPS system was used as a single source of navigation and attitude information in several guided munitions programs in Draper Laboratory. The unmanned helicopters developed at Draper Laboratory and MIT also use EKF, using additional sensors. The detailed design of an EKF for the unmanned helicopter navigation and attitude determination is considered in Section 2.3.

Chapter 4

Conclusion

In the process of development of the avionics systems for two small unmanned aircraft, many lessons in system integration, modeling, simulation and algorithms were learned. These will be used in future designs of unmanned aircraft.

In hardware, the improvements in performance and cost can be achieved by foreseeing the common problems for small UAVs in the preliminary design phase, and by using the experience gained to mitigate the adverse effects. These include sensor degradation due to vibration, electromagnetic (EM) noise, effects of EM interference on ranges of remote control and communication links.

Fast computers and data acquisition systems allow better modelling of errors in estimation filters. This places less stringent requirements on sensor performance. A set of attitude determination and navigation algorithms have been tested in simulation and in the field, and important sensor errors were identified or predicted.

A model for a low speed gun-launched surveillance aircraft was developed and analyzed. A control augmentation system for a prototype vehicle was designed and tested with the pilot-in-the-loop simulation. A classical design approach was taken to develop an intuitive feeling for a physically sound choice of the control variables and feedbacks. The problem of gust disturbance rejection for a low speed vehicle can be effectively addressed by using the modern control methods, along with the knowledge gained from the model analysis and classical design. A reasonable choice would be to use a design approach that allows frequency weighting of desired outputs, and simul-

taneously incorporates ambiguity in the estimates of aircraft parameters, especially the lateral-directional control and damping derivatives. New software packages for real μ synthesis [22], [12] allow to account for available information on gust frequency content and uncertainty in aerodynamic derivatives. This tool also allows one to obtain boundaries for performance, which will provide limitations on product use. This determines whether the UAV can be used in a certain application. The algorithm performance is degraded with the increasing number of uncertain parameters, thus the careful analysis of the model helps identify the most important ones.

The field of small UAVs is growing rapidly, and the work on these two projects during the past two years has been challenging. A great number of improvements can be made in these systems, and new designs will contribute to the exciting area of unmanned aircraft.

Appendix A

Nonlinear Model in Simulink

A.1 Block Hierarchy

The model is built using a hierarchy of Subsystem blocks. At the lowest level there are two S-functions 'aero.m' and 'eqmot10.m' which contain computation of aerodynamic forces and moments, and aircraft 6-DOF equations of motion correspondingly, and an engine model Subsystem block. At the higher level of hierarchy the actuator block provides inputs to the aircraft block, and sensor block takes inputs from the a/c block. There are two sets of these blocks. One is a fully linear (except non-hard actuator deflection limiters) and analog, which is used for the linearization. The other includes the sensor and actuator non-linearities, sensor noise and quantization. At the highest level are the trim inputs, pilot inputs and control augmentation system. All three levels in this hierarchy are presented in Figures A-1 - A-3.

The simulation is initialized from Matlab workspace. The simulation parameters are passed to Simulink blocks and S-functions automatically. The inputs to the simulation are defined in workspace. The simulation is run using 'sim' command with a vector of inputs and a time vector.

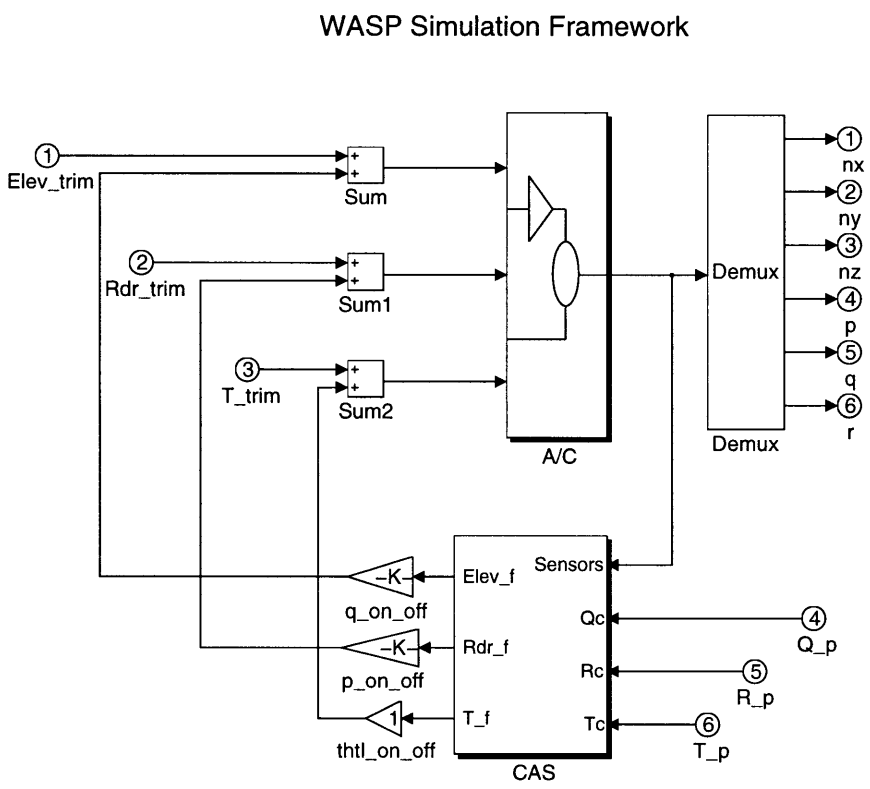

Figure A-1: Simulation. Level 1: I/O

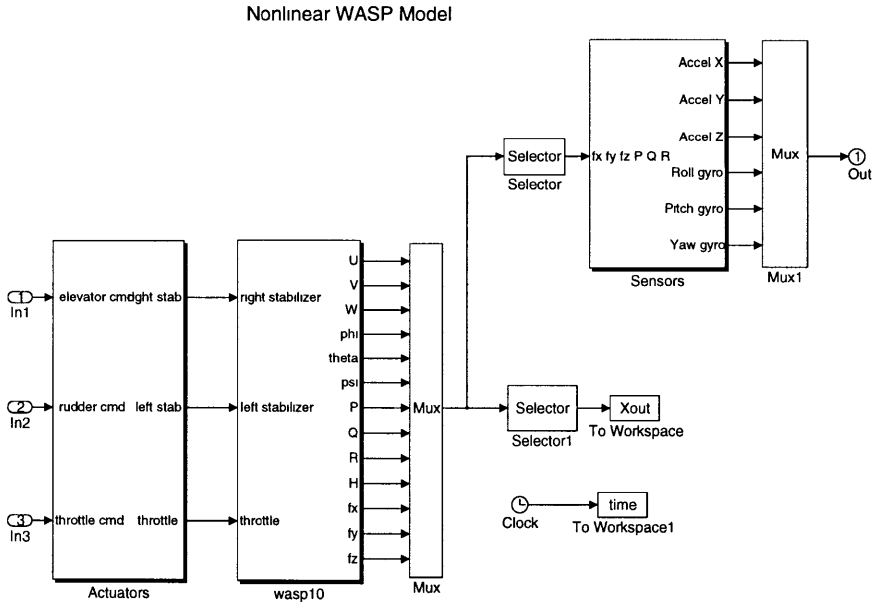

Figure A-2: Simulation. Level 2: sensors and actuators

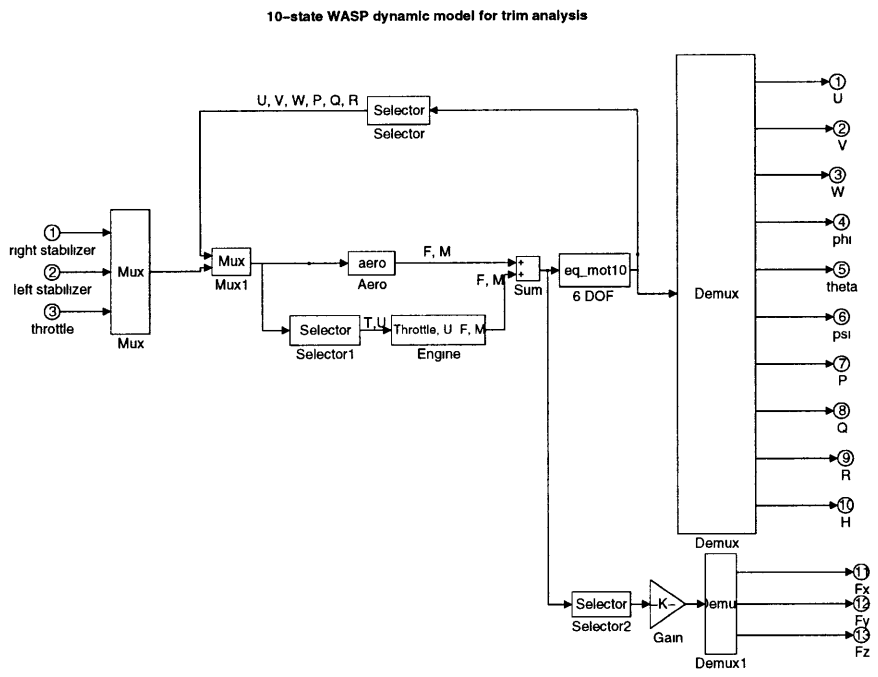

Figure A-3: Simulation. Level 3: equations of motion

A.2 C-MEX vs M files

The user-defined S-function blocks can be written as either M-files or so called C-MEX files. The C-MEX files are just the C language source code files, written according to certain simple rules. Using the M-file S-functions is slow compared to the C-MEX S-functions. Even for a relatively low-order simulation like WASP's, it is worth using the C-MEX functions. If one is going to use the Simulink simulation with animation output or with human operator external input, then C-MEX implementation is a must.

A.3 Animated Pilot-in-the-Loop Simulation

In the development of the prototype vehicle two versions of animated simulation were used extensively for control system design and for pilot training. For the design of control system, the Simulink based simulation was converted to C-MEX files, and connected to simplified animation program and to the RC receiver interface. The animation was based on OpenGL graphics package, it featured the rigid body vehicle

dynamics, terrain with a grid and the vehicle shadow when altitude is low. This is a minimum requirement for a realistic pilot perception. The animation program takes vehicle attitude and position as inputs. It can be also used for post flight analysis of the flight using recorded data. The receiver interface, described in Section 2.2 communicated with the host computer via serial interface to provide pilot commands. This setup allows rapid prototyping of control systems, using standard Simulink libraries.

The animation of the WASP vehicle was done in Draper Laboratory, the interfaces between Simulink and the animation and the receiver interface were provided by Emilio Frazzoli, a graduate student at MIT.

For pilot training a Silicon Graphics workstation-based simulation with realistic terrain and detailed graphics of the vehicle was used. This tool is written in C-language, and uses SGI Graphic Libraries for animation. The same receiver interface was used for pilot command inputs.

A.4 Suggested Framework for Hardware-in-the-Loop Simulation

Adding hardware-in-the-loop capability to the simulation greatly reduces time for debugging the real time code. Using the onboard CPU in the simulation with exactly the same interfaces and timing can be easily done, if all the onboard I/O is easy to emulate. A possible simulation framework for the FTV is given in Figure A-4.

This framework can be extended for any other sensors having serial outputs. In case the sensor I/O is analog, the emulation of the interface and timing may become quite complicated (it is routinely done in the industry for large scale products). For rapid prototyping with COTS elements the use of this architecture will allow designers to concentrate on the research, and spend less time on device drivers and debugging the real-time software.

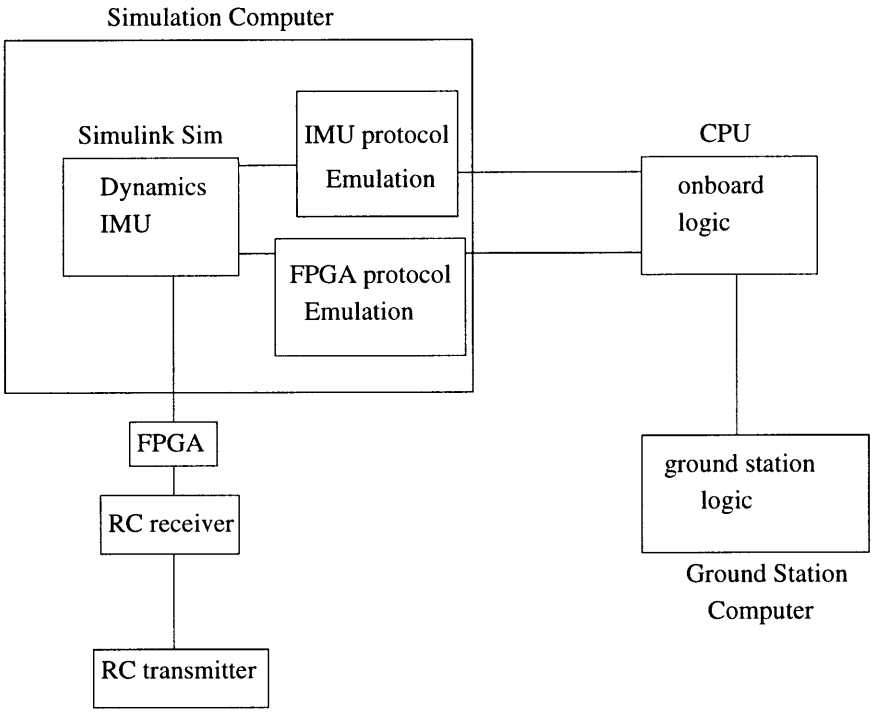

Figure A-4: Hardware-in-the-loop simulation

Appendix B

Additional Data for Navigation Filter

B.1 Linear Propagation Matrix

Jacobian matrix from equation 2.8 can be represented in the following block way:

$$A = \begin{bmatrix} 0 & A_{pv} & A_{pq} & 0 & 0 \\ 0 & A_{vv} & A_{vq} & A_{vg} & A_{v\gamma} \\ 0 & 0 & A_{qq} & 0 & A_{q\gamma} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.1}$$

where zeros stand for the zero matrices of corresponding dimensions. Computing partial derivatives of right hand side of equation 2.8, we shall arrive at:

$$A_{pv} = C \tag{B.2}$$

$$A_{vv} = -\Omega_3 + \Gamma_3 \tag{B.3}$$

$$A_{qq} = -\frac{1}{2}\left(\Omega_4 - \Gamma_4\right) \tag{B.4}$$

$$A_{pq} = \begin{bmatrix} q_0u - q_3v + q_2w & q_1u + q_2v + q_3w & -q_2u + q_1v + q_0w & -q_3u - q_0v + q_1w \\ q_3u + q_0v + q_1w & q_2u - q_1v + q_0w & q_1u + q_2v + q_3w & q_0u - q_3v + q_2w \\ -q_2u - q_1v + q_0w & q_3u - q_0v - q_1w & -q_0u + q_3v - q_2w & q_1u + q_2v + q_3w \end{bmatrix} \tag{B.5}$$

$$A_{vq} = 2g \begin{bmatrix} -q_2 & q_3 & -q_0 & q_1 \\ -q_1 & -q_0 & q_3 & q_2 \\ q_0 & -q_1 & -q_2 & q_3 \end{bmatrix} \tag{B.6}$$

$$A_{q\gamma} = \begin{bmatrix} q_0 & q_3 & -q_2 \\ -q_3 & q_0 & -q_1 \\ q_2 & q_1 & q_0 \end{bmatrix} \tag{B.7}$$

$$A_{vg} = \begin{bmatrix} 2(q_1q_3 - q_0q_2) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{B.8}$$

$$A_{v\gamma} = -V_3 = \begin{bmatrix} 0 & w & -v \\ -w & 0 & u \\ v & -u & 0 \end{bmatrix} \tag{B.9}$$

B.2 Measurement Matrices

A block matrix for the GPS velocity measurement $H_{vq} = A_{pq}$.

Let us derive the measurement equation for the compass. After compensation for the magnetic variation, the compass gives information about the true heading to the north. The compass measurements are internally compensated for the pitch and roll angles of up to 20 degrees, the measurements are not used if the tilt is higher. For $\theta < 20$ degrees it is safe to use the following formula:

$$\psi = \arctan\left(\frac{C_{21}}{C_{11}}\right) \tag{B.10}$$

Thus, the linearized measurement matrix:

$$H_{\psi q} = \nabla_q \arctan\left(\frac{C_{21}}{C_{11}}\right) = \frac{\nabla_q \frac{C_{21}}{C_{11}}}{1 + \left(\frac{C_{21}}{C_{11}}\right)^2} = \frac{C_{11}\nabla_q C_{21} - C_{21}\nabla_q C_{11}}{C_{11}^2 + C_{21}^2} \tag{B.11}$$

And, finally, evaluating partial derivative epressions:

$$H_{\psi q} = \frac{2}{C_{11}^2 + C_{21}^2}\left[\begin{array}{cccc} C_{11}q_3 - C_{21}q_0 & C_{11}q_2 - C_{21}q_1 & C_{11}q_1 + C_{21}q_2 & C_{11}q_0 + C_{21}q_3 \end{array}\right] \tag{B.12}$$

B.3 Conversion of Measurements to Center of Gravity

B.3.1 Accelerometer Measurements

Let us define $\overline{r}_{imu}$ the vector in body axes connecting the IMU center and the vehicle cg location. The accelerometer measurements are corrected using the following formula:

$$\overline{a}_{cg} = \overline{a}_{IMU} - \overline{\omega} \times \overline{\omega} \times \overline{r}_{imu} - \dot{\overline{\omega}} \times \overline{r}_{imu} \tag{B.13}$$

Angular acceleration vector should be estimated by a bandlimited differentiation of the measured angular rates. The transfer function is $Ts/(Ts+1)$. The roll off frequency $1/T$ should be chosen above the rigid body vehicle dynamics and well below the Nyquist frequency.

B.3.2 GPS measurements

Let us define the GPS antenna offset by a_{GPS}. The GPS position measurements are corrected:

$$Z_p = p_{GPS} - Ca_{GPS} \tag{B.14}$$

The velocity measurements correction:

$$Z_v = v_{GPS} - \overline{\omega} \times Ca_{GPS} \tag{B.15}$$

B.3.3 Altitude measurements

Let us define a vector connecting cg location and sonar antenna location: $[x_s \ y_s \ z_s]$. The coordinates are defined in body axes. Then the raw measurement h_s should be corrected:

$$Z_h = h_s - (C_{31}x_s + C_{32}y_s + C_{33}z_s) \tag{B.16}$$

B.4 Field Heading

For the case of a helicopter with a small range, using the NED frame may not be convenient. Instead, a suitable local frame can be defined by rotation of the NED frame about Down axis by a positive ψ_f - the field heading angle. Then the calibration procedure is necessary to determine ψ_f. DGPS with long enough baseline is used for this purpose. This setup is also used to find the magnetic variation: the local angle ψ_{var} between true north and magnetic north. In this case, the baseline vector should be extracted from the GPS data according to:

$$\begin{aligned} Z_x &= (\varphi - \varphi_b)R\cos\psi_f + (\lambda - \lambda_b)R\sin\psi_f\cos\varphi_b \\ Z_y &= -(\varphi - \varphi_b)R\sin\psi_f + (\lambda - \lambda_b)R\cos\psi_f\cos\varphi_b \\ Z_z &= h_b - h \end{aligned} \tag{B.17}$$

where φ, λ, h - latitude, longitude and altitude of the rover receiver, φ_b, λ_b, h_b - latitude, longitude and altitude of 'the base receiver, located at the origin of the local frame, R - Earth radius. The values of the position and velocity covariances are then corrected to account for ψ_f rotation.

The raw compass measurement is corrected in the following way:

$$Z_\psi = \psi - \psi_f - \psi_{var} \tag{B.18}$$

Bibliography

[1] Inertial Sensor Technology Trends. Guidance Technology Center, Draper Laboratory, Cambridge, MA, January 1997.

[2] John D. Anderson. *Fundamentals of Aerodynamics.* McGraw-Hill, Inc., 1991.

[3] Michael Athans. *6.245 - Multivariable Control Systems.* Lecture Notes, MIT, 1997.

[4] I.Y. Bar-Itzhack and I. Ziv. Frequency and Time Domain Designs of a Strapdown Vertical Determination System. *Guidance and Control*, 1986.

[5] Arthur E. Bryson. *Control of Spacecraft and Aircraft.* Princeton University Press, 1993.

[6] Thierry Casiez. Compact, High-g, High Efficiency Folding Wing for a Cannon-Launched Reconnaissance Vehicle. Master's thesis, MIT, 1998.

[7] Paul DeBitetto. Red Team Review of the Draper Small Autonomous Aerial Vehicle. Technical report, Draper Laboratory, Cambridge, MA, May 1996.

[8] John J. Deyst. *16.840 - Real-Time Aerospace Systems.* Lecture Notes, MIT, 1997.

[9] Peter Donaldson. *Unmanned Vehicles Handbook.* The Shephard Press, 1993-97.

[10] John R. Dowdle and Karl W. Flueckiger. An Integrated GPS/Micro-Mechanical IMU for 5" Navy Projectiles . *Guidance and Control*, 1996.

[11] Marc Drela. *16.110 - Flight Vehicle Aerodynamics.* Lecture Notes, MIT, 1997.

[12] P. M. Young et. al. Computing Bounds for the Mixed-μ Problem. *Intern. Journal of Robust and Nonlinear Control*, 5:573–590, 1995.

[13] Michael C. Francis. The UTA: Redefining The Air Combat System . *Aerospace America*, May 1998.

[14] A. Gelb. *Applied Optimal Estimation.* The Analytical Sciences Corp., 1974.

[15] L.A. Gould, W.R. Markey, J.K. Roberge, and D.L. Trumper. Control Systems Theory. unpublished, 1994.

[16] D. Gustafson and D. Lucia. Autonomous Local Vertical Determination For Guided Artillery Shells. *Presented at the ION 52nd Annual Meeting*, 1996.

[17] Jean-Marc Hauss. Design of an Unmanned Aerial Vehicle. Master's thesis, MIT, 1998.

[18] Paul Horowitz and Winfield Hill. *The Art of Electronics.* Cambridge University Press, 1989.

[19] Eric N. Johnson, Paul A. Debitetto, Christian A. Trott, and Michael C. Bosse. The Draper Small Autonomous Aerial Vehicle Technical Description. Technical report, Draper Laboratory, Cambridge, MA, 1996.

[20] Sebastien Katch. Concept Development, Mechanical Design, Manufacturing and Experimental Testing for a Cannon Launch Reconnaissance Vehicle. Master's thesis, MIT, 1998.

[21] Richard P. Kornfeld, R. John Hansman, and John J. Deyst. Single Antenna GPS Based Aircraft Attitude Determination. *Institute of Navigation*, 1998.

[22] L. Lublin. *The use of additional actuators to enhance performance robustness.* PhD thesis, MIT, 1996.

[23] Duane McRuer, Irving Ashkenas, and Dunstan Graham. *Aircraft Dynamics and Automatic Control.* Princeton university Press, 1972.

[24] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-time Signal Processing.* Prentice Hall, 1989.

[25] James Paduano. *16.334 - Advanced Flight Control Systems.* Lecture Notes, MIT, 1998.

[26] Richard E. Phillips and George T. Schmidt. GPS/INS Integration. *AGARD MSP Lecture Series LS-207*, 1996.

[27] *QNX Operating System. System Architecture.*

[28] Jan Roskam. *Airplane Flight Dynamics and Automatic Flight Controls.* DAR-corporation, 1995.

[29] Christopher P. Sanders, Paul A. DeBitetto, and Eric Feron. Hierarchical Control of Small Autonomous Helicopters. *Submitted to Journal of Guidance and Control*, 1998.

[30] George T. Schmidt. High Integrity Global Navigation Systems. Guidance Technology Center, Draper Laboratory, Cambridge, MA, January 1997.

[31] George M. Siouris. *Aerospace Avionics Systems: A Modern Synthesis.* Academic Press, Inc., 1993.

[32] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation.* John Wiley and Sons, Inc., 1992.

[33] Survey of GPS Receivers. *GPS World*, 1997.

[34] David H. Titterton and John L. Weston. *Strapdown Inertial Navigation Technology.* Lavenham Press, Lavenham, United Kingdom, 1997.

[35] Christian A. Trott. Electronics Design For An Autonomous Helicopter. Master's thesis, MIT, 1998.

[36] David L. Trumper. *2.171 - Digital Control Systems Analysis and Design.* Lecture Notes, MIT, 1997.

[37] David L. Trumper. *2.737 - Mechatronics.* Lecture Notes, MIT, 1997.

[38] P. M. Young. *Robustness and Parametric and Dynamic Uncertainty.* PhD thesis, Stanford University, 1993.