DEVELOPMENT OF A UAV TESTBED

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ZEYNEP ÇAKIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

MAY 2011

Approval of the thesis:

DEVELOPMENT OF A UAV TESTBED

submitted by **ZEYNEP ÇAKIR** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp _____
Head of Department, **Aerospace Engineering**

Asst. Prof. Dr. İlkay Yavrucuk _____
Supervisor, **Aerospace Engineering Department, METU**

Prof. Dr. Ozan Tekinalp _____
Co-supervisor, **Aerospace Engineering Department, METU**

Examining Committee Members:

Asst. Prof. Dr. Ali Türker Kutay
Aerospace Engineering Dept., METU _____

Asst. Prof. Dr. İlkay Yavrucuk
Aerospace Engineering Dept., METU _____

Prof. Dr. Ozan Tekinalp
Aerospace Engineering Dept., METU _____

Prof. Dr. Aydan M. Erkmen
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ZEYNEP ÇAKIR

Signature :

ABSTRACT

DEVELOPMENT OF A UAV TESTBED

Çakır, Zeynep
M.S., Department of Aerospace Engineering
Supervisor : Asst. Prof. Dr. İlkay Yavrucuk
Co-Supervisor : Prof. Dr. Ozan Tekinalp
May 2011, 76 pages

The development and testing for a UAV testbed to be used in academic research and undergraduate education is proposed in this thesis. Analysis on commercial off-the-shelf UAV systems and autopilots lead to the development of a custom, open-architecture and modular UAV testbed. The main focus is to support research in UAV control field and education of the undergraduate students. The integration and use of commercial-off-the-shelf avionics and air vehicle are described in detail. System performance is examined both in flight and on the ground. Results of the system tests show that the developed system is a functional UAV testbed to be used in research of different flight control algorithms.

Keywords: Unmanned Aerial Vehicle, Autopilot, Flight Control System, Avionics System

ÖZ

İHA TEST PLATFORMU GELİŞTİRİLMESİ

Çakır, Zeynep
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. İlkay Yavrucuk
Ortak Tez Yöneticisi : Prof. Dr. Ozan Tekinalp
Mayıs 2011, 76 sayfa

Bu tezde akademik araştırmalarda ve lisans eğitiminde kullanılmak üzere bir İHA test platformu geliştirilmesi önerilmiştir. Ticari kullanıma hazır İHA ve otopilot sistemleri üzerine yapılan yerel ve uluslararası piyasa araştırmaları özgün, açık mimari ve modüler bir İHA test platformu geliştirilmesine sevk etmiştir. Geliştirilen sistemin odak noktası İHA kontrolü üzerine eğitim ve AR-GE araştırmalarına destek olmaktır. Ticari kullanıma hazır aviyonik sistem ve test platformu olarak kullanılan hava aracı detaylı olarak anlatılmıştır. Sistem performansı yer testleri ve uçuşlu testlerle değerlendirilmiştir. Sistem test sonuçları, geliştirilen sistemin, farklı uçuş kontrol algoritmalarının test edilmesinde kullanılabilecek fonksiyonel bir İHA test platformu olduğunu göstermiştir.

Anahtar Kelimeler: İnsansız Hava Aracı, Otopilot, Uçuş Kontrol Sistemi, Aviyonik Sistem

To Our Wonderful Planet

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisors Asst. Prof. Dr. İlkay Yavrucuk and Prof. Dr. Ozan Tekinalp for their invaluable support in this thesis study. I would also like to thank Prof. Dr. Aydan M. Erkmen for her precious guidance and motivation which helped me elaborate on this work.

I wish to state my special thanks to Batu Demir whom I worked with throughout my thesis study. He was always by my side in good times and in bad times with his optimism, knowledge and endless support. I would have never completed this thesis if it weren't for him.

I would like to thank Alper Erdener who shared his valuable knowledge and experience with me and showed me a starting point for this study. He came with an answer whenever I had a question. I want to emphasize my special thanks to Aykut Kutlu and Onur Tarakçıoğlu for their guidance in Kalman filter design and avionics, for their useful discussions and motivation throughout the study. I also want to thank to my dear friend and colleague Sinem Işık for being part of this study and sharing her precious knowledge with me. I would like to express my thanks to my dear friends in Simulation, Control and Avionics Laboratory and Aerospace Engineering Department; Onur Tarımcı, Deniz Yılmaz, Tuba Elmas, Yunus Emre Arslantaş, Eser Kubalı, Mehmet Can Kılıç, Sharmila Kayashta, Mehmet Katırcıoğlu, Berk Korkut, Gökhan Ahmet, Sinan Körpe, Eda Doğan and Tuğba Ünlü for their support and positive energy. I should not forget Sinan Ekinci and Volkan Kargın from whom I had invaluable advices.

I would like to express my special thanks to one of the best RC pilots Dr. Saffet Gökuç for his invaluable help, optimism and guidance during the flight tests. Also, thanks to Murat Usta and his team in 5^{th} Main Maintenance Center for repairing our UAV testbed after a severe crash.

Last, but not least, I would like to express my deepest thanks to my parents and to my love and best friend Duysal İmsel for their endless support, trust, understanding and patience.

This thesis is financially supported by 107M346 project of the Scientific and Technological Research Council of Turkey (TÜBİTAK).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Unmanned Aerial Vehicles (UAV) as an emerging technology are expected to claim their place in most of the missions once accomplished by manned aerial vehicles. They are used for autonomous surveillance, search and rescue, cargo transportation as well as military missions. In the past, UAVs were only remotely piloted aircraft. Recently, autonomous control is increasingly employed to UAVs due to the rapid developments in sensor, navigation and computer technologies. The degree of autonomy is one of the design considerations for a UAV. Autonomy requires work on sensor fusion, communications, path planning, trajectory generation, trajectory regulation, cooperative tactics, task allocation and scheduling. Some of these functions are accomplished in the low level autopilot.

The UAV test bed considered in this thesis is an autonomous, unmanned aerial platform which is used to integrate and test various parts of an automatic flight control system and associated flight control software. Thus, the UAV test bed is equipped with the necessary avionics to perform autonomous flight.

Commonly a flight control algorithm design starts with a simulation model of the aircraft. The flight controller is then designed based on this simulation model. The controller at this stage usually does not perform as expected in real flight and requires further enhancement to the flight control software. An aerial test platform is therefore necessary to complete the final stage -flight testing- in the flight controller design process.

Several UAV systems are available in the market today. These UAV systems could be used as a UAV test bed. However, most of these systems do not allow the end-user to load and test different flight control algorithms on the UAV autopilot. Moreover, they are very expensive systems and these systems are usually far more capable than needed for a UAV test bed. Therefore, most of these systems are not suitable to be used as a UAV test bed. An alternative

approach is to integrate an RC model airplane with a commercial autopilot. There is a wide range of commercial autopilots in the world market. Some of these are open-source and some are closed-source. These autopilot systems are equipped with all the necessary sensors and RF communication hardware. Closed-source autopilots -just like UAV systems- do not allow the user to test different flight control algorithms on the autopilot. Open-source autopilots are good candidates; however, some of them are not sold in Turkey and some of them are not easily programmable. Therefore, the use of commercial autopilots in the UAV test bed is not desirable.

Considering the above discussion about the UAV test bed configurations, the final and best configuration is then decided to be the integration of commercial-off-the-shelf (COTS) avionics system (i.e. computers, sensors and other electronic equipment) with an RC model airplane. This configuration has many advantages. It is cheaper because the autopilot components are COTS products. Also, COTS products do not impose any dependency to a specific producer or a country. The system has increased modularity due to the use of standard data communication interfaces (e.g. RS-232). The modularity makes it possible to update the system with the latest technology products when necessary. Considering the above benefits, the UAV test bed is decided to consist of a modular COTS avionics system and an RC model airplane.

This thesis describes the efforts undertaken in the Department of Aerospace Engineering of the Middle East Technical University for the development of a UAV test bed to meet the need for an autonomous aerial platform to be used in the flight testing of different flight control algorithms. From educational point of view, the UAV test bed will allow students to test their novel flight control algorithms in real flight and compare with simulation results. It will make the students part of design projects from initial concept to final flight testing phase. The autopilot used in the test bed will be easily programmable by the students using a high-level programming language such as Matlab.

1.1 Literature Survey

There has been various efforts to build UAV testbeds both in the industry and academia. A design of an autopilot capable of flying small unmanned aerial vehicles is presented in [1]. The autopilot consists of a central processing unit (CPU), inertial sensors, GPS receiver, commu-

nication hardware and bypass circuit. The autopilot is capable of automatic takeoff-landing and waypoint navigation using a basic PID controller. The autopilot board is small and light weight. However, the CPU core module- RCM3100 of Rabbit Semiconductor, Inc.- has two main drawbacks: 29 MHz clock speed and 8-bit operation. Although it can do sensor processing and run simple control algorithms, the computing power is insufficient for complex operations.

[2] discusses development of a low-cost UAV test-bed for educational purposes. The test-bed consists of an RC model airplane, autopilot and ground station. The autopilot is equipped with a micro-controller, inertial and pressure sensors, actuators and communication devices. The micro-controller is 29 MHz Rabbit 3000. Both of the works mentioned previously use the same micro-controller. It is easier for the undergraduate students to write flight control algorithms using a high level language, e.g. Matlab/Simulink. The mentioned processor needs low-level language programming.

In a similar work ,[3], open architecture hardware (flight computer, sensors, modems, etc) is integrated with a so-called 'rapid prototyping software suite' to realize autonomous flight of an RC airplane called 'FROG'. In this work, the concept of 'rapid prototyping' is applied to the discipline of flight control. The goal of this work is to accelerate the flight control projects using 'rapid prototyping system'. The open architecture hardware consists of an inertial measurement unit, a GPS receiver, RF modems and a ground control computer. Different from the previous works, flight control is performed on the ground control computer on which Matrix_x software tools are running. [4] takes the work of [3] further by using a newer rapid prototyping software, xPC Target from the Mathworks, Inc. Different from [3], in this work, flight control algorithms run on the airborne PC/104 computer.

Development of a low-cost and modular avionics system is presented in [5]. A new design methodology is formulated to develop the low-cost modular avionics system. The methodology focuses on the extensive use of commercial-off-the-shelf (COTS) technology. The designed avionics system consists of a flight control computer, various sensors (AHRS, DGPS, sonar), wireless ethernet, mux circuit and DC/DC converter. The avionics system is then implemented on a small UAV helicopter airframe.

1.2 Contributions of the Thesis

The focus of this thesis is to develop an unmanned aerial test bed to be used in flight testing of several flight control algorithms written by graduate or undergraduate students. The UAV test bed is developed to aid research and development projects in aerial control engineering field. There are several related works in the literature. Although being similar to some of the past studies in the literature, this thesis has some improvements relating to fields mentioned in the following paragraphs.

Flight control computer used in the thesis work is a high-performance and high-speed computer compared to low-speed micro-controllers ([1] and [2]). The high computing speed allows to run complex control algorithms or do data processing without any latency.

Matlab/Simulink is a very common simulation and programming environment preferred by control engineers and students. It makes programming easier by making use of high-level language programming. The flight control computer used in the thesis allows the user to download flight control software directly from Matlab/Simulink environment. Therefore it makes programming easier compared to the micro-controllers which are programmed using low-level languages such as C/C++ and Assembly.

The developed avionics system consists of COTS products rather than custom parts. COTS products help keep the system price to a minimum and relieves dependency on a particular manufacturer. Also, the system is designed to be modular. Modularity enables the user to upgrade system components with the latest technology counterparts.

1.3 Thesis Structure

Chapter 1 gives an overview about the UAV testbed. Definition of the UAV testbed, purpose of the thesis, necessity to develop a UAV testbed, literature survey and contributions of the thesis are stated under subtitles of this chapter. Chapter 2 gives a detailed description of the UAV testbed. Section 2.1.1 describes the onboard avionics system and ground control station components. Properties of the aircraft used as the UAV testbed are given in Section 2.2. Software tools used in the development of the UAV testbed and the structure of real-time controller are described in Chapter 3. System test results and discussions are given in Chapter 4. The noise sensitivity of the attitude sensor due to mechanical vibrations caused by the engine

is analyzed in Chapter 5. The limitations and constraints of the developed system are listed in Chapter 6. Summary of the study, advantages of the developed UAV testbed over existing systems, general remarks and suggestions for future work are stated in Chapter 7 which concludes the thesis. Appendix A describes the structure of the Kalman filter used in state estimation. Hardware interface models for attitude heading reference system (AHRS), GPS receiver, servo driver board and analog inputs are explained in detail in Appendix B. Appendix C introduces the graphical user interface used to communicate with the UAV testbed. Data logging method and recorded states are given in Appendix D.

CHAPTER 2

UAV TESTBED INTEGRATION AND HARDWARE

Three approaches were considered in the selection of the most appropriate UAV testbed configuration. First approach was to buy a complete UAV system. Several UAV systems were available in the local market today. Some of the operational mini UAV systems manufactured by national companies are Globiha of Global Teknik, Bayraktar of Baykar Makina, Efe of Vestel and Güventürk of Middle East Technical University. These UAV systems were the first candidates as a UAV testbed. However, these systems either used closed sourced off-the shelf autopilots, or would use their own with closed source control systems and did not allow the end-user to load and test different flight control algorithms on the UAV autopilot. On top of that, these systems are expensive and the capabilities of these systems such as endurance, range and operational altitude are beyond the requirements of a university UAV test bed. Therefore, these systems were not suitable to be used as a UAV test bed.

The second approach was to buy a radio-controlled (RC) model airplane and combine it with a commercial autopilot. There is a wide range of commercial autopilots in the world market. Results of a market survey on commercial autopilots is given in (Table 2.1). Some of the autopilots, listed in Table 2.1, are open source and some are closed source. These autopilot systems are equipped with necessary sensors and RF communication hardware. Some of the closed-source autopilots are capable of waypoint navigation, auto-takeoff/landing and can support unconventional control surface configurations (e.g. elevon). Closed source autopilots let the user to adjust the controller gains based on the aerial platform. However, they do not allow the user to load and test different flight control algorithms on the autopilot. Therefore, they are not preferred in the development of the UAV testbed. SPB400 & MNAV is an open-source autopilot mostly preferred in UAV applications; however, the processor unit, SPB400, is an obsolete product of Crossbow company, it is not produced anymore. Piccolo LT of Cloud

Cap Technology is another open-source autopilot, but it is currently not exported to Turkey. PhoenixAX of O-navi,on the other hand, seems to be a better option, but still it has some drawbacks such as low processing speed and low-level language programming. Therefore, the open-source autopilot candidates listed in Table 2.1 are not used in the UAV testbed.

Table 2.1: Market Survey on Commercial Autopilots (as of April, 2008)

Product Name	**Processor**	**Memory**	**Sensors**	**Open/Closed Source**
SPB400 & MNAV (Crossbow)	400 MHz (Intel PXA255)	64Mb (sd) 32Mb (flash)	3-axis accelerometers 3-axis angular rate sensors 3-axis magnetometers altitude sensor airspeed sensor GPS	open source
MP2028xp (MicroPilot)	n/a	n/a	y-accelerometer roll, pitch, yaw gyros altitude sensor airspeed sensor GPS	closed source
Kestrel (Procerus)	29 MHz (Rabbit RCM 3100)	512K	3-axis accelerometers 3-axis angular rate sensors 3-axis magnetometers altitude sensor airspeed sensor GPS	closed source
Piccolo LT (Cloud Cap Technology)	40 MHz (Motorola MPC555)	448K (flash) 26K (sram)	3-axis accelerometers 3-axis angular rate sensors static pressure sensor dynamic pressure sensor GPS	open source
UNAV 3500 (UNAV, LCC)	16-bit	256K (flash)	inertial measurement unit barometer airspeed sensor GPS	closed source
PhoenixAX (O-Navi)	32 MHz (Motorola MMC 2114)	32K (ram) 256K (flash)	3-axis accelerometers 3-axis angular rate sensors static pressure sensor dynamic pressure sensor GPS	open source

Considering the above discussion on alternative UAV test bed configurations, the most appropriate configuration is decided to be composed of an RC model airplane and COTS avionics system; processor, sensors, actuator drivers and miscellaneous electronic equipment. The

components of the selected configuration are explained in detail in the following sections.

2.1 Avionic System Hardware

2.1.1 Onboard Avionics

Airborne components of the RPS and the communication interfaces between the components is shown in Figure 2.1.

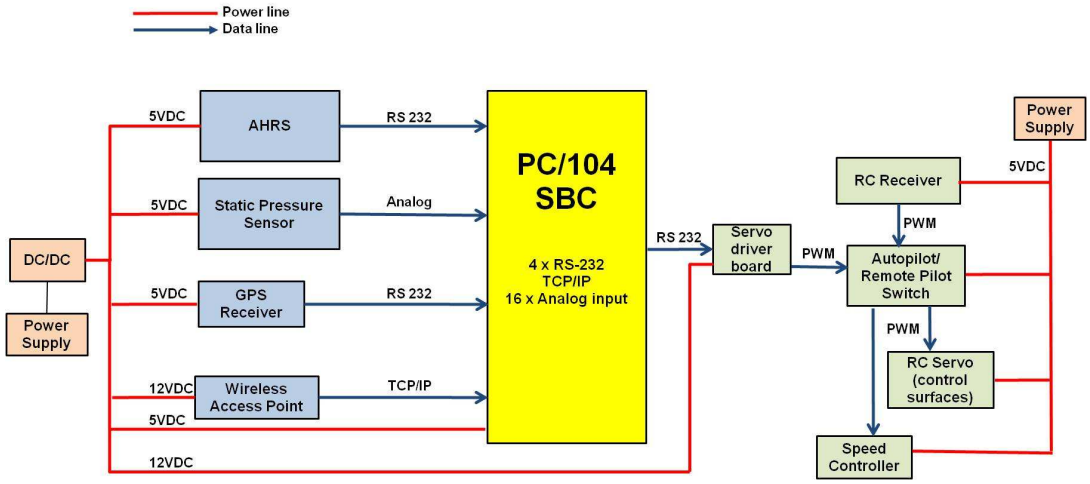


Figure 2.1: Hardware Interface Schematic of the Prototype Autopilot

2.1.1.1 Flight Control Computer

The flight control computer is the main computing power of the autopilot. It is required to have at least a processor and sufficient memory. Two types of processor options are considered to be used in the UAV test bed: micro-controllers and single board computers. Micro-controllers, such as PIC, are light and cheap; however, they generally have low processing speed. Single board computers (SBC), on the other hand, are generally faster than micro-controllers. There are several types of SBCs having different form factors and data buses. Some of the most common SBCs are PC/104, PC/104+, CompactPCI and VME boards. The SBCs are named after their data bus types and form factors. VME and CompactPCI data buses are designed to utilize a backplane configuration for data transmission to peripheral devices [5].

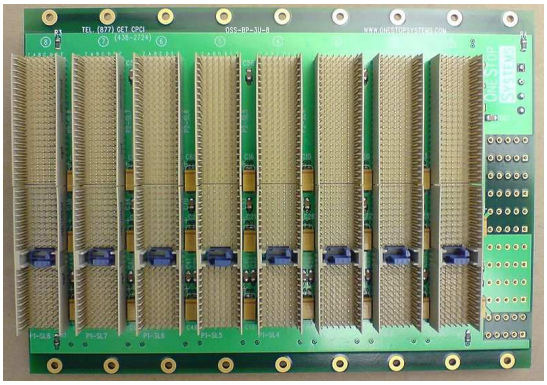


Figure 2.2: CompactPCI Backplane

The PC/104 bus standard is based on the industry ISA standard. PC/104+ adds support for the PCI bus, in addition to the ISA bus. Both cards are designed so that peripheral cards are stacked on top of each other.

The Form factor basically specifies the physical dimensions of the single board computer. Specifically, standard form factors ensure that parts are interchangeable across computing vendors and generations of technology [6]. PC/104 has a relatively small form factor compared to PC/104+, CompactPCI and VME. External PC/104 add-on cards can provide additional peripheral support. PC/104 standard is a reduced form factor implementation of the regular PC bus (IEEE P996) for embedded applications.

There are a couple of reasons explaining why PC/104 SBC is preferred to be used as the flight control computer in this work. First, PC/104 uses standard PC architecture. By standardizing hardware and software around the broadly supported PC architecture, development costs, risks, and time can be substantially reduced. Another important advantage of using the PC architecture is that its hardware and software standardized components are widely available. These components are also significantly more economical than traditional non-PC bus architectures such as STD, VME, and Multi-bus. This means lower product costs [7]. Second, PC/104 offers full architecture, hardware, and software compatibility with the PC bus, but in ultra-compact (3.6” x 3.8”) stackable modules due to its small form factor. Third, unique self-stacking bus eliminates the cost and bulk of backplanes and card cages. Lastly, it has low power consumption (1-2 Watt) compared to other bus types. In short, PC/104 makes it possible to embed standard PC technology in limited space applications providing a wide range of multi-vendor support. Therefore, PC/104 SBC is selected as the flight control computer regarding the above benefits.

Athena II SBC of Diamond Systems Corporation [19] (Figure 2.4) is selected to be the flight control computer of the UAV test bed due to its low power consumption, on-board data acquisition circuit and Matlab/Simulink support. The SBC has 800MHz processor with 256MB on-board RAM and a 16-bit low-noise integrated data acquisition circuit, 4 serial ports (RS-232/485), 4 USB ports (1.1), and 10/100 Mbps ethernet (Figure 2.3).

The built-in PC/104 expansion bus enables the SBC to work with other data acquisition, I/O or communication boards. Stack-through PC/104 connector on the board helps stack other boards of the same form factor on top of the SBC (Figure 2.4).

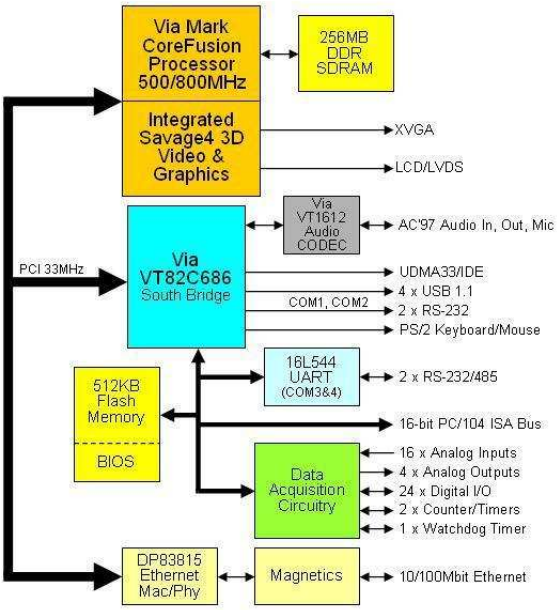


Figure 2.3: Athena II System Block Diagram

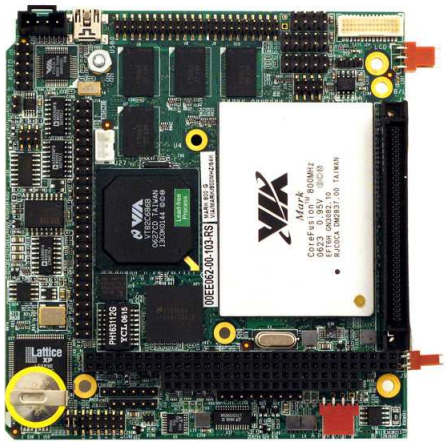


Figure 2.4: Athena II SBC

An integrated data acquisition circuit has 16 single-ended/8 differential analog inputs with 16-bit resolution and 4 analog outputs with 12-bit resolution. Maximum A/D sampling rate is 100 KHz. Athena II data acquisition block diagram is seen in Figure 2.5.

A solid state IDE flashdisk module installed on the computer is used for flight data storage. Data logging method and the list of recorded states are given in Appendix D. xPC Target is booted from the flash disk module.

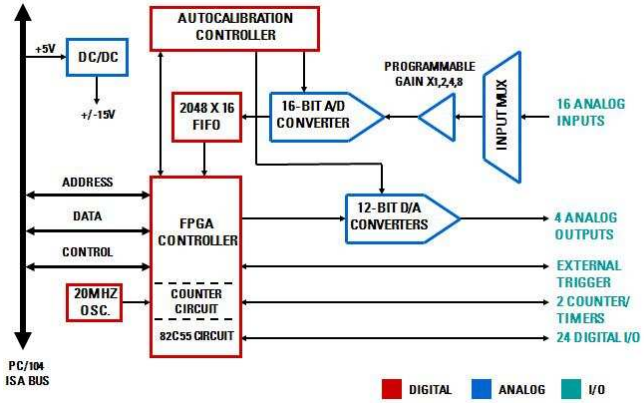


Figure 2.5: Athena II Data Acquisition Block Diagram

2.1.1.2 Attitude Heading Reference System

The 3DM-GX1® AHRS of MicroStrain, Inc. is used to do the inertial measurements [20]. The sensor combines three angular rate gyros with three orthogonal accelerometers, three orthogonal magnetometers, multiplexer, 16-bit A/D converter and embedded micro-controller to measure the attitude, heading and acceleration of the airplane in static and dynamic environments. The benefits of this system can be listed as follows:

- Small, lightweight and low power
- Specifically designed and manufactured for unmanned vehicles, robotics and navigation applications
- On-board processing of accelerometer, gyro and magnetometer outputs using the embedded micro-controller relieve the central processor (flight computer) from the burden of orientation calculations
- Accelerometer and gyro biases are fully compensated over wide temperature range (-20, 60°C)

It provides orientation in matrix, quaternion and Euler formats with a sampling rate of 100 Hz. The digital serial output also provides temperature compensated and calibrated data from all nine orthogonal sensors at update rates of 350 Hz. Roll, pitch yaw (heading) angles and angular rates and three axes body accelerations are the measured outputs of the AHRS.

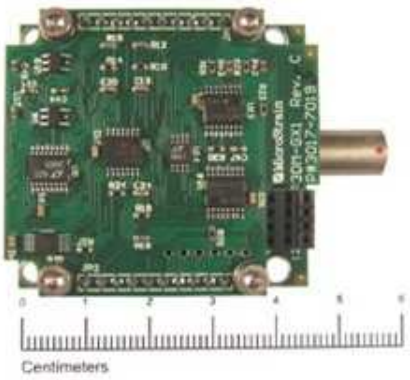


Figure 2.6: AHRS without Cover

Figure 2.7: AHRS with Cover

Table 2.2: AHRS Specifications

Specifications	
Orientation range Roll, pitch, yaw	±90, 180, 180°
Orientation resolution	<0.1°
Accelerometer range	±5g
Accelerometer bias stability	±5 mg
Accelerometer nonlinearity	0.2%
Accelerometer resolution	0.2 mg
Gyro range	±300°/sec
Gyro nonlinearity	0.2%
Gyro bias stability	0.7°/sec
Gyro resolution	0.01°/s
Magnetometer range	±1.2 Gauss
Magnetometer nonlinearity	0.4%
Magnetometer bias stability	0.010 Gauss
Magnetometer resolution	0.2 mGauss
A/D converter resolution	16 bits
Update rate	100Hz
Digital outputs	Serial RS-232 and RS-485
Operating temperature	-40°to 70
Weight	74.6 gr
Shock limit	500 G's

2.1.1.3 Global Positioning System Receiver

The EVK-5 GPS receiver of u-blox AG is used to measure the position of the airplane with respect to earth's center (Figure 2.8). The GPS receiver is able to monitor 50 satellites simultaneously. GPS data is updated once every second. Horizontal position accuracy is less than 2.5 m. Velocity and heading accuracies are less than 0.1 m/s and 0.5°, respectively. GPS data is sent using NMEA protocol over RS-232 serial data link. NMEA messages sent by the GPS receiver are based on NMEA 0183 Version 2.3 [22] . Figure 2.9 shows the structure of an NMEA protocol message. GPS performance is summarized in Table 2.3. GPS receiver outputs longitude, latitude and altitude measurements.

Figure 2.8: EVK-5 GPS Receiver

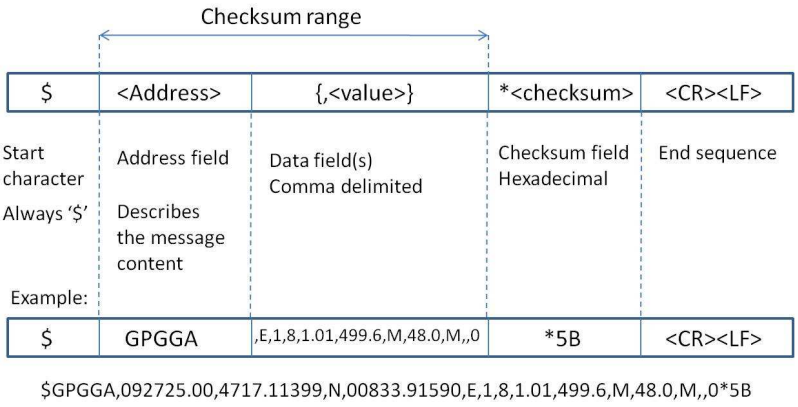


Figure 2.9: NMEA Protocol Message

2.1.1.4 Barometer

MPXA6115A pressure sensor of Freescale Semiconductor, Inc. is used to measure the pressure altitude (Figure 2.10). The sensor measures absolute pressure and outputs voltage. Mea-

Table 2.3: GPS Performance

Parameter	**Specification**	
Receiver type	50 channels GPS L1 frequency, C/A code	
Time-to-first-fix	Cold start Warm start Hot start	29s 29s <1s
Horizontal positioning accuracy	<2.5m	
Velocity accuracy	<0.1m/s	
Heading accuracy	0.5°	
Dynamics	≤ 4g	
Operational limits	Altitude Velocity	50000m 500m/s

surement range of the pressure sensor is between 15 kPa and 115 kPa. The relation between the output voltage with the static atmospheric pressure is seen in Figure 2.11. Typical, minimum and maximum output curves are shown for operation over 0 to 85°temperature range. The output saturates outside the rated pressure range. Pressure error band is ± 1.5 kPa.

Figure 2.10: Static Pressure Sensor

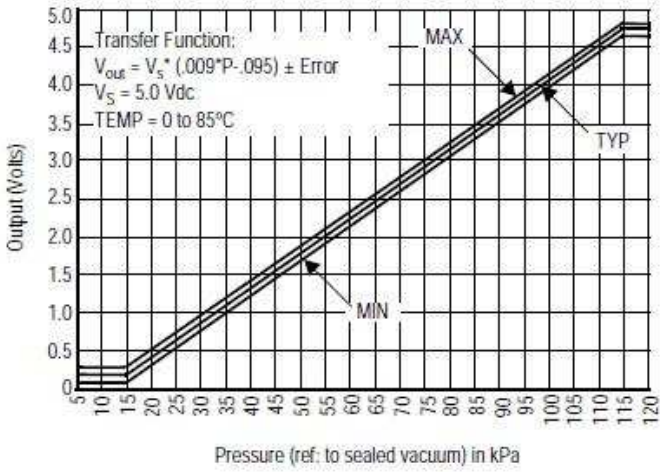


Figure 2.11: Output Voltage vs Absolute Pressure[23]

The transfer function relating pressure measurement to output voltage is given in Eqn.2.1.

$$V_{out} = V_S \times (0.009 \times P - 0.095) \pm (P_{err} \times \text{Temp. Factor} \times 0.009 \times V_S) \qquad (2.1)$$

'P_{err}' denotes the pressure error in 'kPa' (Figure 2.12) and 'Temp. Factor' denotes the temperature error factor (Figure 2.13).

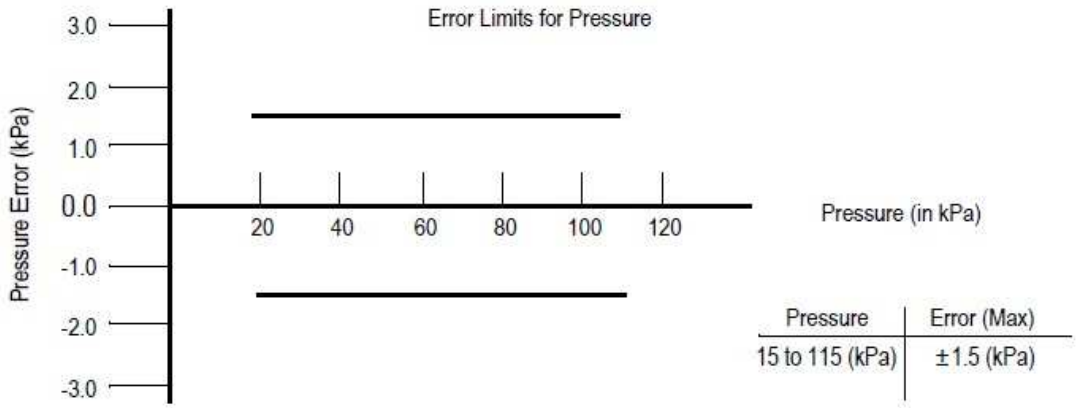


Figure 2.12: Pressure Error Band[23]

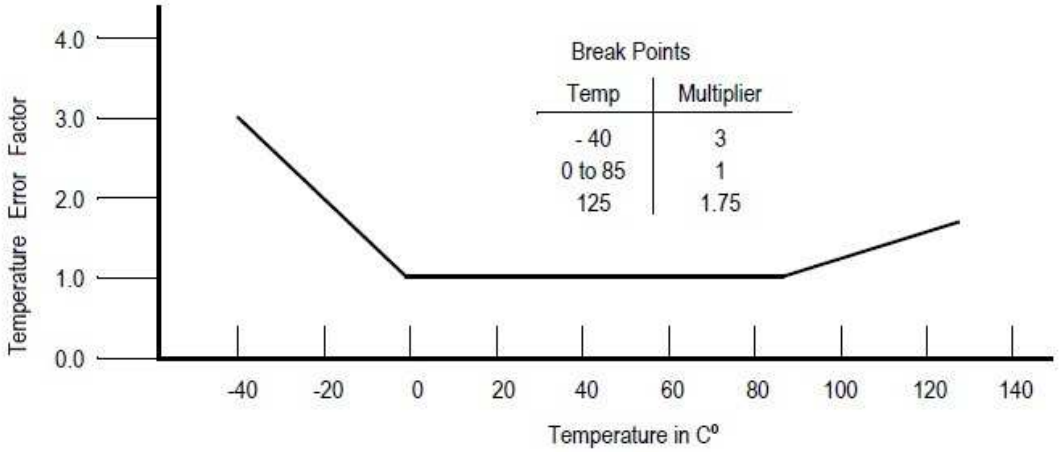


Figure 2.13: Temperature Error Band[23]

Pressure output of the sensor is converted to altitude using International Standard Atmosphere model [24].

In order to obtain correct measurements from the barometer, it has to be calibrated before flight. This calibration is automatically done before each flight using GPS altitude measurement. After the GPS position fix is completed and the position measurements are settled, the barometer is calibrated with respect to the GPS altitude measurement before take-off.

2.1.1.5 Actuator Servos

Analog RC servo motors are used to actuate the control surfaces. A typical servo motor block diagram is shown in Figure 2.16. Control pulse is the pulse-width modulation (PWM) signal

which drives the servo motor. Figure 2.17 shows a PWM signal. PWM signals are characterised by period (T) and duty cycle. Duty cycle of a PWM signal determines how much the servo arm and thus the control surface deflects. Servo motors used in the thesis work have 20 ms period with a duty cycle range of 4-11%. 4% duty cycle (1050 μs pulse length) corresponds to 0° and 11% duty cycle (1950 μs pulse length) corresponds to 90° servo arm deflection.

Position sensor in Figure 2.16 is generally a potentiometer which outputs voltage that is related to the absolute angle of the output shaft. It measures the servo arm deflection and feedbacks it to the error amplifier which minimizes the difference between the commanded and the actual servo deflection. Error amplifier produces positive or negative voltage which is input to the motor. If it is positive, the motor turns in one direction, if negative the other. Control pulse is generated by either the RC receiver (Section 2.1.1.11) or the servo driver board (Section 2.1.1.6).

Figure 2.14: Standard Analog RC Servo

Figure 2.15: RC Servo (inside)

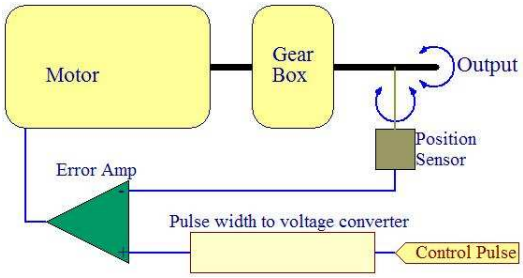


Figure 2.16: Servo Motor Block Diagram [26]

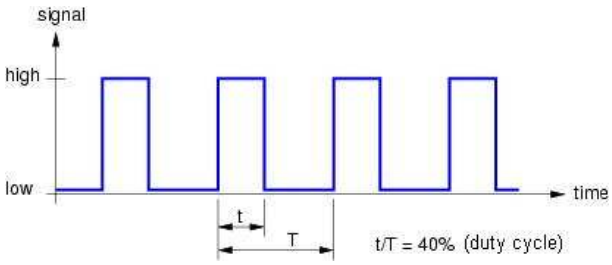


Figure 2.17: PWM Signal

2.1.1.6 Servo Driver Board

The servo driver board is used to drive the actuator servo motors. Servo motors are driven using a Pulse-width-modulation (PWM) signal as mentioned in Section 2.1.1.5. The board basically consists of a micro-controller (PIC 18F4520) which generates PWM signals and a

MAX232 integrated circuit for serial communication between the board and the flight control computer. Control surface deflection angles generated by the flight control algorithm is sent to the micro-controller on the board and generated PWM signals are sent to the servo motors. Up to four servo motors can be driven by the servo driver board. Figure 2.18 shows the servo driver board installed on the airplane.

2.1.1.7 Autopilot/Remote Pilot Switch

Autopilot/remote pilot switch (AP/RP Switch) is a small multiplexer board -RxMux of Acroname Robotics (Figure 2.19). The board has sixteen inputs; eight channels reserved for receiver PWM input (input A in Figure 2.20) and eight channels reserved for autopilot PWM input (input B in Figure 2.20) to servo motors. One channel is reserved for trigger input, therefore the board can pass maximum seven servo inputs. Working principle of the board is simply passing one of the two input sets according to the value of the trigger input. Trigger input is sent from one of the channels of the remote control so that in case of an emergency (e.g. a failure in the autopilot), the pilot has control over the airplane.

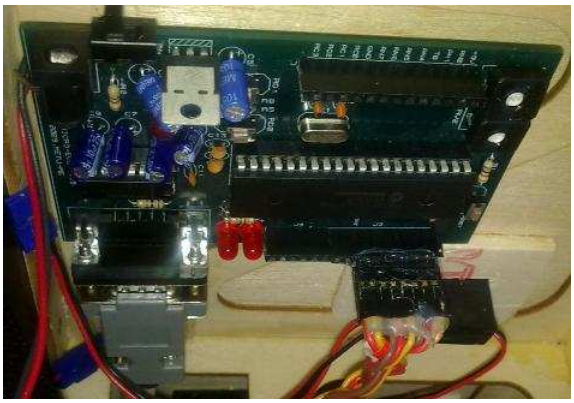


Figure 2.18: Servo Driver Board Mounted Inside the Airplane

Figure 2.19: RxMux

2.1.1.8 Ground Communication Hardware

AWK-3121 wireless access point (WAP) of Moxa, Inc. is used to establish ground communication. Flight control computer and ground control computer communicates over wireless TCP/IP network connection using two wireless access points (Figure 2.21). Operating frequency of WAP is 2.4 GHz. Communication range is between 500m to 10km. Transmit

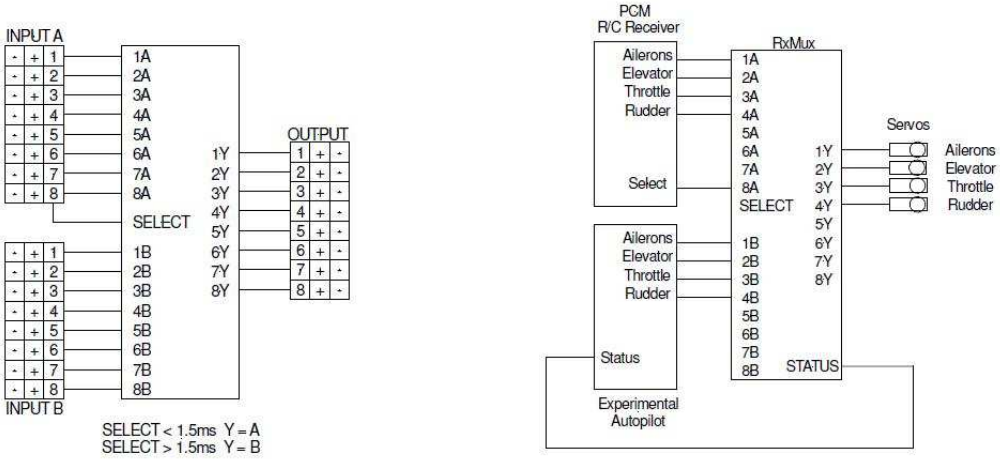


Figure 2.20: Autopilot/Remote Pilot Switch Schematic [28]

power of the WAP is decreased as the range is increased. Since the WAP on the airplane is mobile, transmit power plays a more important role. In order not to lose too much transmit power, the range is set to 2km which is a sufficient range to perform tests on the prototype autopilot.

Figure 2.21: Air-ground Communication Using Wireless Access Points

2.1.1.9 Potentiometers

Potentiometers measure the deflection angle of control surfaces (Figure 2.22). Bourns® 6639 precision potentiometers are used on the UAV testbed. Potentiometers output voltage which are proportional to the control surface deflection angles. They are intended to be used in the system identification or parameter estimation processes.

Figure 2.22: Potentiometer

2.1.1.10 DC/DC Converter

Sensors on the airplane require different supply voltages (Figure 2.1). In order to get rid of unnecessary battery weight, a DC/DC converter is used to power the electronics on the airplane.

A DC/DC converter is basically a power regulator. It regulates battery voltage. He104+DX DC/DC converter of Tri-M Engineering is used in the thesis work (Figure 2.23). It is a PC/104 DC/DC converter requiring an input battery voltage of 6-40VDC. It is stacked with the flight control computer (Figure 2.24) and supplies power for all of the sensors installed on the airplane as well as the flight control computer itself. It supplies +3.3, +5 and ±12VDC voltage output.

Figure 2.23: He104+DX DC/DC Converter

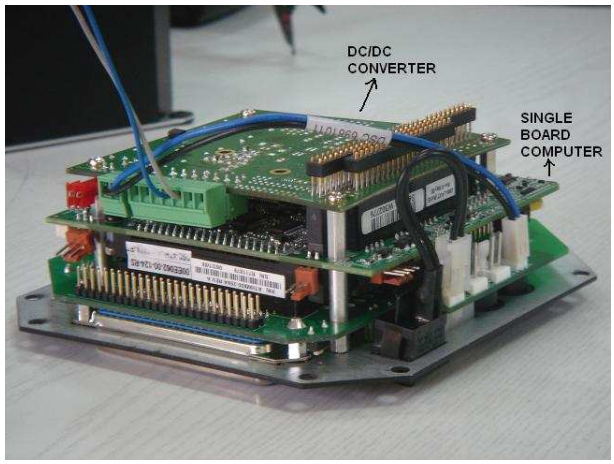


Figure 2.24: SBC-DC/DC Converter Stack

2.1.1.11 Radio Control (RC)

A radio control is used to control the airplane from ground using radio signals. It is composed of a receiver and a transmitter (Figure 2.25). The receiver is placed on the airplane and the transmitter is used by the pilot to control the airplane. The control signals sent from the

transmitter are delivered to the actuator servos by the receiver. Operating frequency of the system is 2.4 GHz with frequency hopping at every 2ms. Hopping frequency helps avoid interference with other wireless devices such as wireless access points.

Figure 2.25: Components of the Radio Control System

2.1.2 Ground Control Station

The ground control station consists of a ground control computer, an RC transmitter (Section 2.1.1.11) and a wireless access point (Section 2.1.1.8). Ground control computer is a standard laptop computer which runs Matlab, Simulink and xPC Target. It communicates with the flight control computer thru WAP and a graphical user interface called 'xPC Target Explorer' (see Appendix C). Instantaneous flight data is monitored on the ground control computer over this telemetry link. Flight data can be either downloaded to the ground control computer or stored on the flight control computer flash disk. Ground control station architecture is seen in Figure 2.26.

2.2 Aircraft

A fixed-wing RC aircraft is selected as the test bed owing to its higher stability and lower vibration characteristics compared to a rotary-wing RC aircraft. The airplane selection is based on stability and payload requirements. The avionics weight budget is given in Table 2.4.

Total avionics payload is approximately 3 kg. There were few RC model airplanes in the

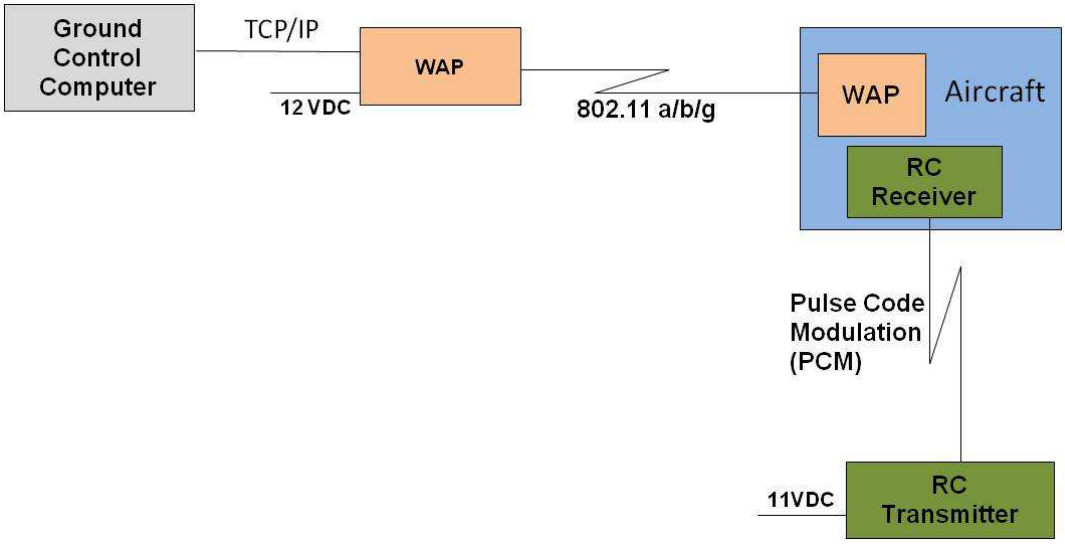


Figure 2.26: Ground Control Station Architecture

Table 2.4: Avionics Weight Budget

Component	**Weight (g)**
Flight control computer	150
Computer enclosure	693
AHRS	74
GPS Receiver	181
Barometer & AP/RP switch & servo driver board	200
Wireless access point	850
DC/DC converter	183
Power supply	600
Total	2931

local market capable of carrying this much payload. The SIG Rascal 110 is a commonly used, large, stable and easy-to-use RC airplane available in the model market. Therefore, it is selected as the UAV testbed (Figure 2.27). The airplane specifications are listed in Table 2.5. Besides its high payload capacity, Rascal 110 is a stable airplane which makes it suitable to be an autopilot testbed.

Placement of avionic system components, engine and batteries on the test bed is seen in Figure 2.28.

Although the selected airplane is originally designed to fly with a fuel engine, a brushless DC motor is used to propel the airplane. The primary reason for selecting a DC motor is

Figure 2.27: SIG Rascal 110 Testbed

Table 2.5: UAV Testbed Specifications

SIG RASCAL 110 Specifications	
Wing span	2794 mm
Wing area	0.982 m^2
Length	1924 mm
Empty weight (battery, servos, engine installed)	7 kg-f
Gross weight	10 kg-f

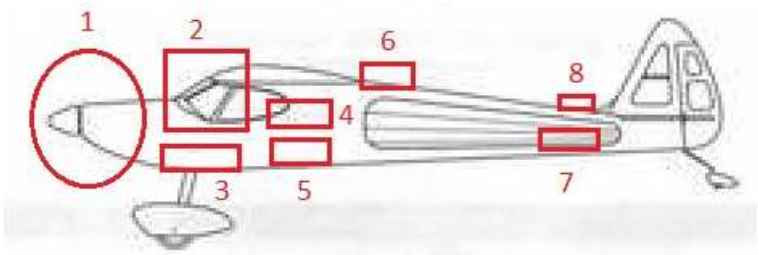
Table 2.6: DC motor and Propeller Specifications

Propulsion Specifications	
Continuous current	40 A
Maximum current	90 A
Shaft diameter of the motor	10 mm
Motor diameter	63 mm
Motor height	73 mm
Propeller length	22 in
Propeller twist	10 deg

that the fuel engines generate excessive vibrations on the airplane which affects the sensor measurements. A less significant consideration is the pollution inside the airplane and on the avionic equipment caused by the fuel and oil particles spread across the airplane after each flight. The disadvantage of DC motors is the flight duration, i.e. endurance. Fuel engines operate longer than the DC motors with equal fuel (battery) weight.

DC motor installed on the testbed is a powerful brushless DC motor. The properties of the DC motor are listed in Table 2.6. It is powered by a 10-cell Lithium-Polymer (Li-Po) battery pack with 37 VDC and 3700 milliampere-hour (mAh) power capacity. An experiment is performed to measure the thrust capacity, RPM and current driven by the DC motor with the selected propeller-battery pair. Especially thrust capacity is an important parameter which helps determine the maximum payload the airplane can carry. The experiment results yield that the selected motor-propeller-battery set can produce up to 70 N of thrust. The thrust-to-weight ratio of the airplane turns out to be '0.7'. Experiment results are given in Figure 2.30.

Three independent battery packs are used to power the electronic equipment on the airplane

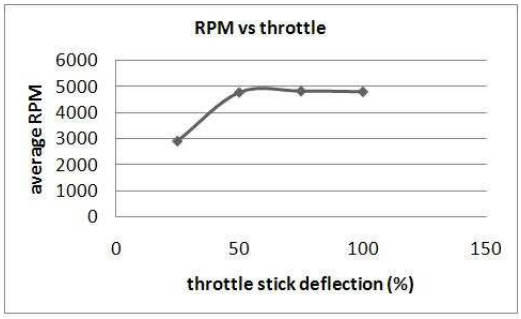


1- engine, ESC, engine battery
2- flight computer
3- avionics battery
4- RC receiver, actuator battery and servo driver board
5- autopilot/remote pilot switch and barometer
6- AHRS
7- GPS receiver
8- GPS antenna

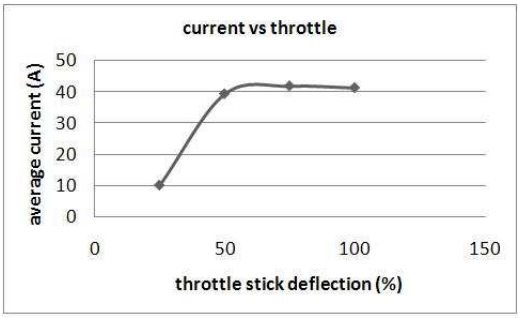
Figure 2.28: Payload Placement

(Figure 2.31). The avionics group consist of flight control computer, AHRS, GPS receiver, barometer, servo driver board and wireless access point. An 18.5 VDC, 5000 mAh 5-cell Li-Po battery pack is used to power the avionics group. The power is distributed over DC/DC converter. Propulsion group consists of electronic speed controller (ESC) and DC motor. A 37VDC, 3700 mAh 10-cell Li-Po battery pack is used to power the propulsion group. Actuator group consists of RC receiver, servo motors and autopilot/remote pilot (AP/RP) switch. A 7.2 VDC 5-cell Ni-Mh battery pack is used to power the actuator group. This kind of power distribution acts as a fail-safe mechanism. For example, even if the engine battery goes off for some reason or the autopilot inputs saturate, it will be possible to control the airplane since the servo motors and AP/RP switch are powered by a separate battery pack.

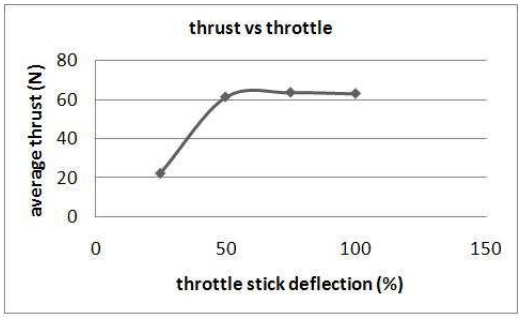
Figure 2.29: Motor Test Setup



(a) Variation of RPM with hrottle



(b) Variation of current with throttle



(c) Variation of thrust with throttle

Figure 2.30: Experiment Results of the DC Motor

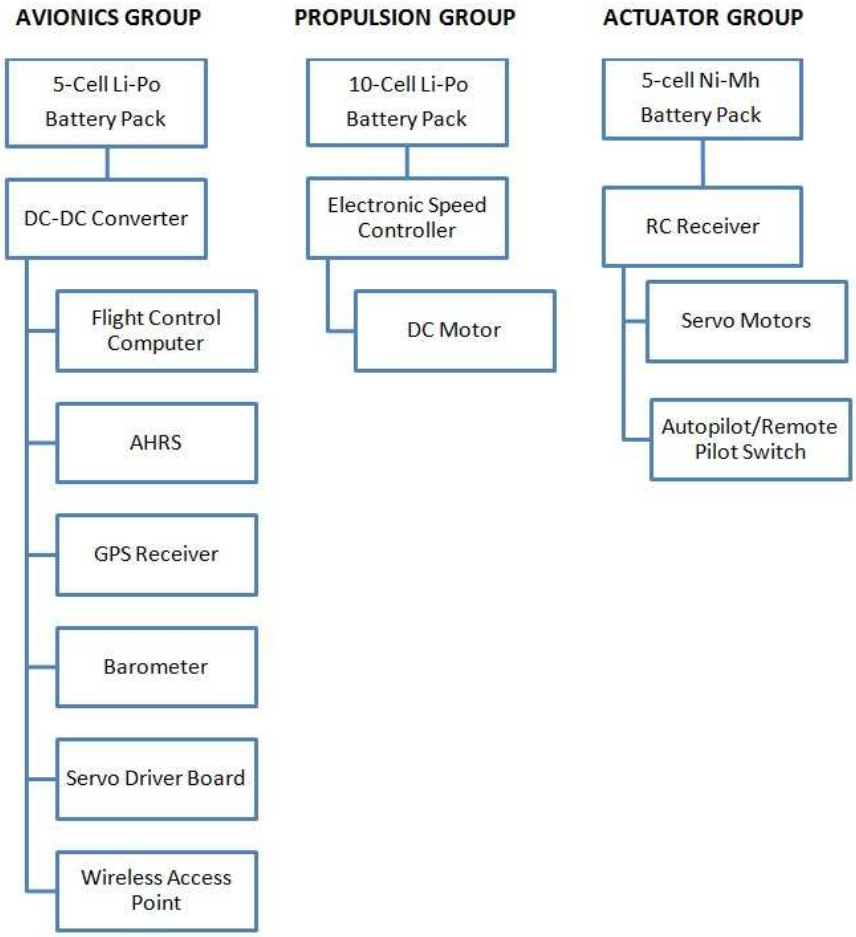


Figure 2.31: Power Tree

CHAPTER 3

SOFTWARE DEVELOPMENT

3.1 Software Tools

Commercially available software tools used in the development of the UAV test bed consist of The MathWorks,Inc software suite; Matlab®, Simulink®, Real-Time Workshop® and xPC Target™.

Controller algorithm and sensor/actuator communication interface models are developed and simulated in Simulink®. Within Simulink®, models are deployed to flexible controller hardware and run in real time. Source codes of the controller algorithm are generated automatically with Real-Time Workshop® reducing the time and effort needed for low-level language programming. Therefore, the designer can focus more on testing rather than programming. Real-time operation on target-computer hardware is performed using xPC Target™ [8]. Detailed information about the software tools is given in the following paragraphs.

MATLAB

MATLAB (stands for *matrix laboratory*) is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping

- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows the user to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran [16].

Simulink®

Simulink® is software for modeling, simulating, and analyzing dynamic systems. Simulink enables the user to pose a question about a system, model it, and see what happens. With Simulink, models can easily be built from scratch, or existing models can be modified to meet the user's needs. Simulink supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate -having different parts that are sampled or updated at different rates [16].

Real-Time Workshop®

Real-Time Workshop® is an extension of capabilities of Simulink® and MATLAB® that automatically generates, packages, and compiles source code from Simulink models to create real-time software applications on a variety of systems.

The process of generating source code from Simulink models using Real-Time Workshop is shown in Figure 3.1.

xPC Target™

xPC Target is a solution for prototyping, testing, and deploying real-time systems using standard PC hardware. It is an environment that uses a target PC, separate from a host PC, for running real-time applications.

In this environment desktop computer is used as a host PC with MATLAB, Simulink, and Stateflow® (optional), to create a model using Simulink blocks and Stateflow charts. After creating the model, the simulations can be run in nonreal time.

xPC Target lets the user add I/O blocks to their model and then use the host PC with Real-Time Workshop® and a C/C++ compiler to create executable code. The executable code is downloaded from the host PC to the target PC running the xPC Target real-time kernel. After

downloading the executable code, target application can be run and tested in real time.

- Hardware requirements

  The xPC Target software requires a host PC and a target PC. A host PC can be either a desktop PC or a notebook PC. A Target PC can also be a desktop PC or an industrial PC (e.g. CompactPCI, PC/104, PC/104+ or any other single board computer (SBC)). For I/O operations, the target PC must have I/O boards supported by xPC Target.

- Software requirements

  The xPC Target software requires a C/C++ compiler in addition to MATLAB, Simulink, and Real-Time Workshop.

xPC Target supports two connection and communication protocols between the host PC and the target PC: serial and network (Figure 3.2).

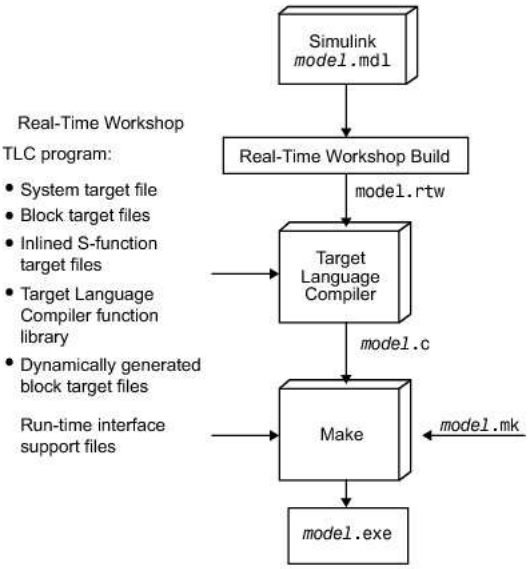


Figure 3.1: Source Code Generation Using Real-Time Workshop [16]

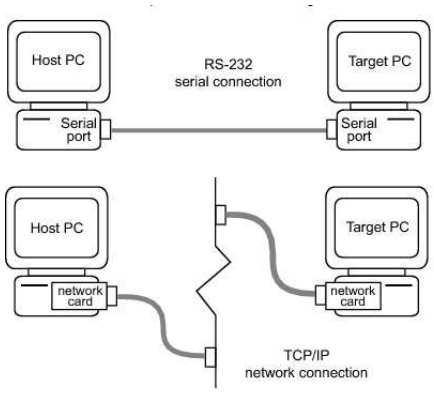


Figure 3.2: Host-Target Connection [16]

3.2 Testbed Flight Controller

Development of a flight controller for the UAV test bed starts with generating the mathematical model of the airplane. The mathematical model and design of the pitch-hold controller is part of the thesis work of Sinem Işık [13]. Mathematical model consisted of 6 degree-of-freedom equations of motion, aerodynamic database, thrust database, environment model

(atmosphere and gravity) and actuator models. Aerodynamic database of Rascal 110 is generated using a computer program called Digital Datcom [15]. Thrust database is generated as a result of a series of thrust experiments (Section 2.2). International Standard Atmosphere model [24] and WGS84 gravity model [25] are used to model the environment. The developed non-linear mathematical model is trimmed and linearized around 20 m/s wings-level flight at 1000 m. The properties of the longitudinal and lateral modes are given in Tables 3.1, 3.2 and 3.3. It is seen that all the modes are on the left-hand side of the complex plane which indicates the stability of the given flight condition.

Table 3.1: Longitudinal Mode Characteristics of Rascal [13]

Mode	**Roots**	**Natural Frequency ω_n [rad/s]**	**Damping Ratio ζ**
Short Period	-7.7276 ± 2.3562i	8.08 (= 1.286 Hz)	0.957
Phugoid	-0.0414 ± 0.3114i	0.314 (= 0.05 Hz)	0.132

Table 3.2: Dutch Roll Mode Characteristics of Rascal [13]

Mode	**Roots**	**Natural Frequency ω_n [rad/s]**	**Damping Ratio ζ**
Dutch Roll	-0.3437 ± 2.7040i	2.73 (= 0.4345 Hz)	0.126

Table 3.3: Roll and Spiral Mode Characteristics of Rascal [13]

Mode	**Roots**	**Time Constant [s]**	**Time to Half Amplitude [s]**
Roll	-8.5525	0.1169	0.081
Spiral	-0.0896	11.1607	7.7344

After verifying the simulation model with a similar but a larger scale airplane (Cessna 172), the controller model is added to the simulation model. An inner loop controller is designed first. The reason is that the inner loop involves fast states (i.e. angular rates and attitude) and without controlling and stabilizing the fast states, it would not be possible or very difficult to control the outer loop dynamics (i.e. altitude, speed). Therefore, a pitch attitude controller is developed to hold the pitch angle. Although the airplane is inherently quite stable, a pitch stability augmentation is placed in the most inner loop. Pitch angle is controlled in the outer loop. Pitch rate and pitch angle measurements are fed back from the AHRS (Section 2.1.1.2). Outer loop command passes through a 2^{nd} order command filter to input the commanded angle gradually and smoothly. Pitch -hold controller block diagram is given in Figure 3.3.

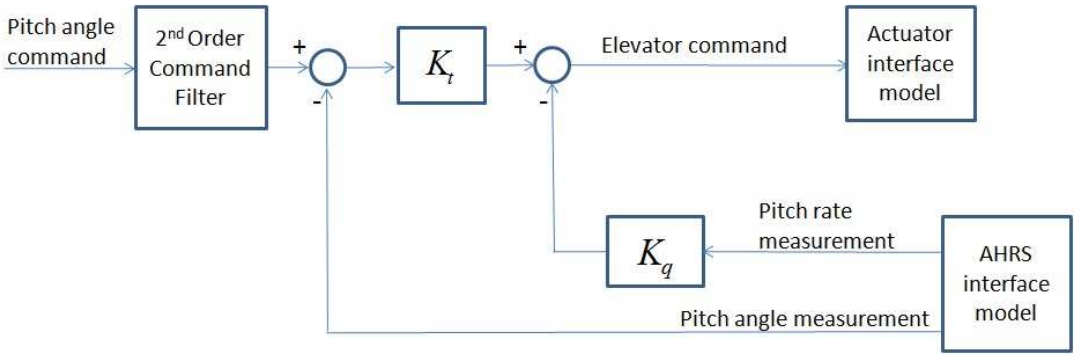


Figure 3.3: Pitch-hold Controller Block Diagram

After the controller algorithm is developed, several simulations are performed to tune the controller parameters until the controller performance satisfies the design requirements. Sensors are modeled by Gaussian noise specifying a mean value and standard deviation for each sensor. Kalman filter design also proceeded in parallel (see Appendix A). The controller algorithm is then ready to be tested in real-time.

Hardware interface models for AHRS, GPS receiver, barometer and servo driver board are written in Simulink (see Appendix B) and tested on desktop. Then, the controller model is integrated with the hardware interface models. Executable code of the final model is generated using Real-Time Workshop and downloaded to target PC to run in real-time (Figure 3.4). The simulation time step is set as 0.01s (100 Hz). Since the sensors may send measurement update at different frequencies, multiple sample times are used in the real-time simulation model. Sampling rates of the components in the simulation model are given in Figure 3.5 and the constraints on the sampling rate are discussed in Chapter 6.

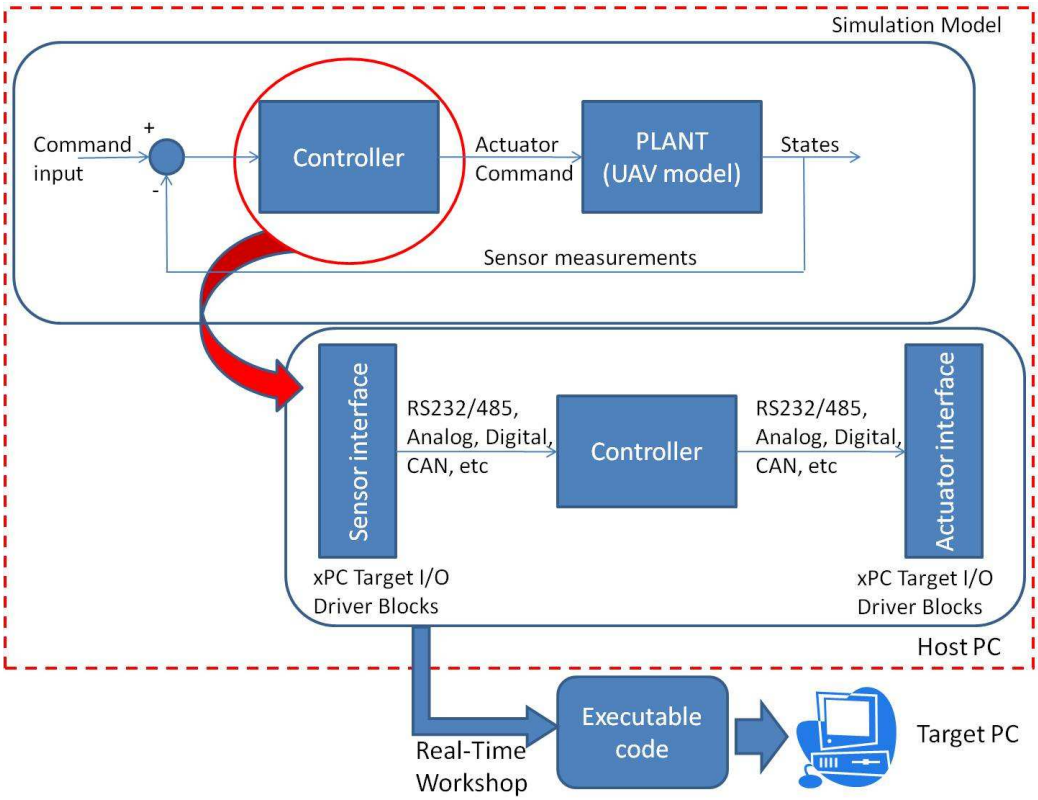


Figure 3.4: Sensor/Actuator Interaction with the Controller

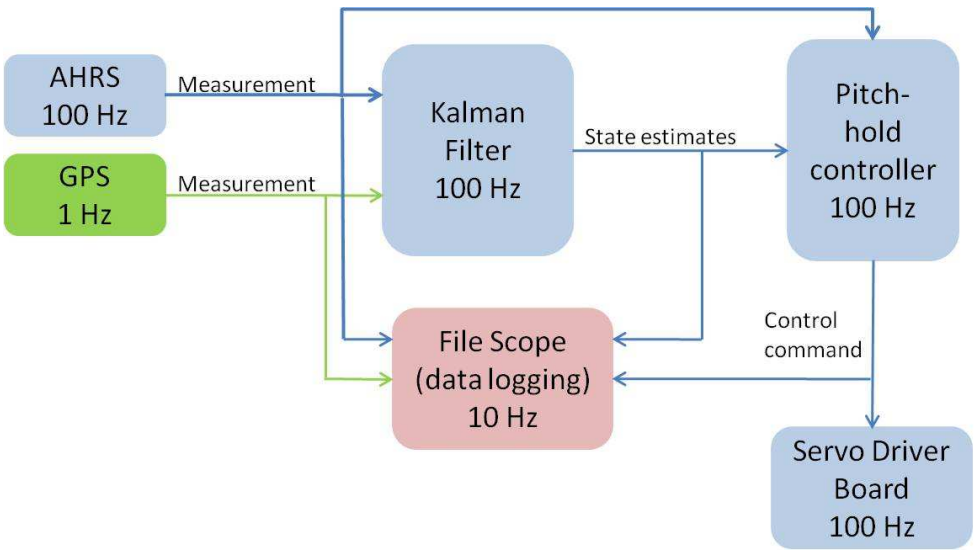


Figure 3.5: Sampling Rate Schematic

CHAPTER 4

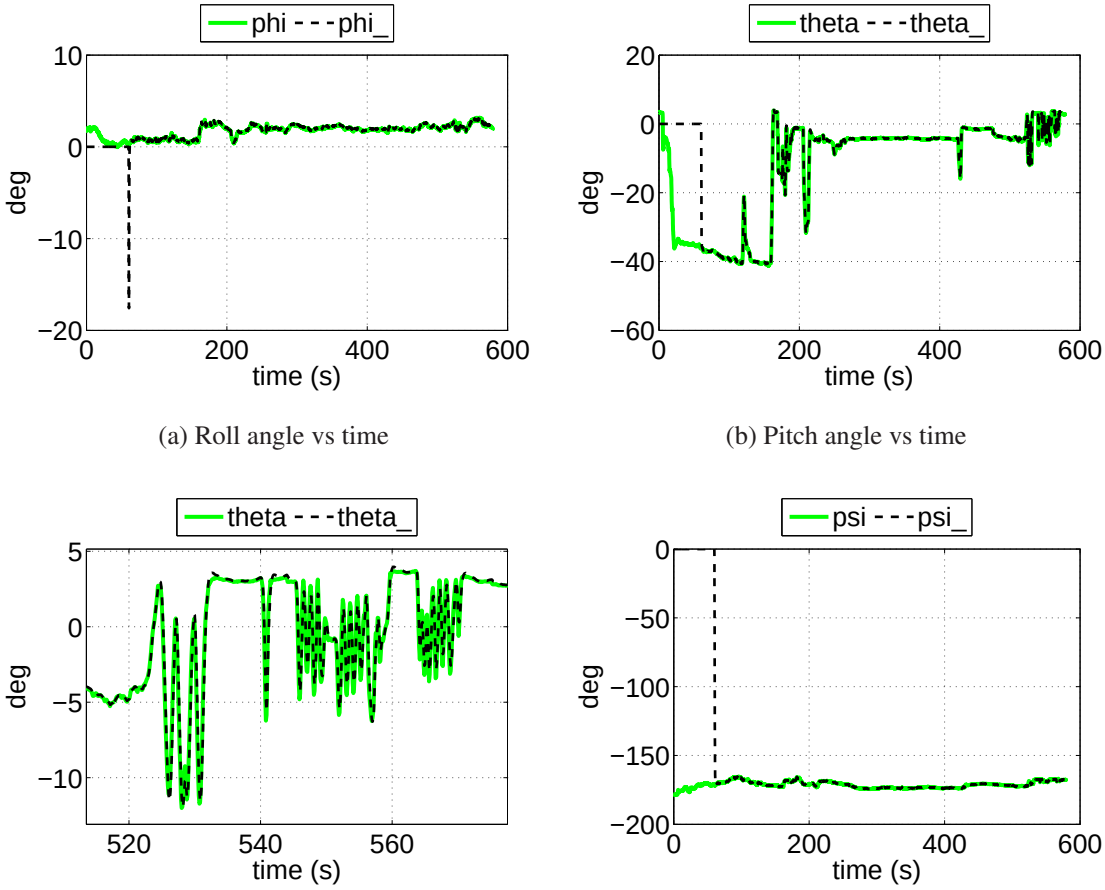
SYSTEM TESTS

After the avionics system is integrated with the testbed aircraft and the controller software is completed, the UAV system is required to be tested to evaluate system performance and detect possible errors. There are certain sources of error. The avionics integration phase is one of them. Although the system components work well individually, they may not operate as expectedly when they come together. Furthermore, although the integration is completed successfully and the system operates flawlessly on the ground, there is still a strong probable possibility that the test bed will come up with new system faults in the aerial tests. Both cases are well-experienced in this work.

4.1 Ground Testing

A series of ground tests are performed to check possible faults related to system integration, sensor functionality and measurements. Sensor measurements are recorded to the flight control computer during the tests. Pitch-hold controller functionality is also tested by changing the pitch angle of the test bed on the ground. A set of ground test results are given in the following graphs.

Variation of Euler angles with time is given in Figure 4.1. Solid data lines are the roll, pitch and yaw angles of the test bed measured by the AHRS. Dashed data lines are Kalman filter estimates of the Euler angles. Kalman filter estimates are seen to be zero for the first 60 seconds. The reason is that Kalman filter algorithms start running after all the sensor measurements are settled. This is a precaution taken especially against GPS measurements which may require some time to fix onto the satellites and provide accurate measurements. The tail of the airplane is moved up and down and sensor measurements and pitch attitude controller



(a) Roll angle vs time

(b) Pitch angle vs time

(c) Pitch angle vs time (zoomed in)

(d) Yaw angle vs time

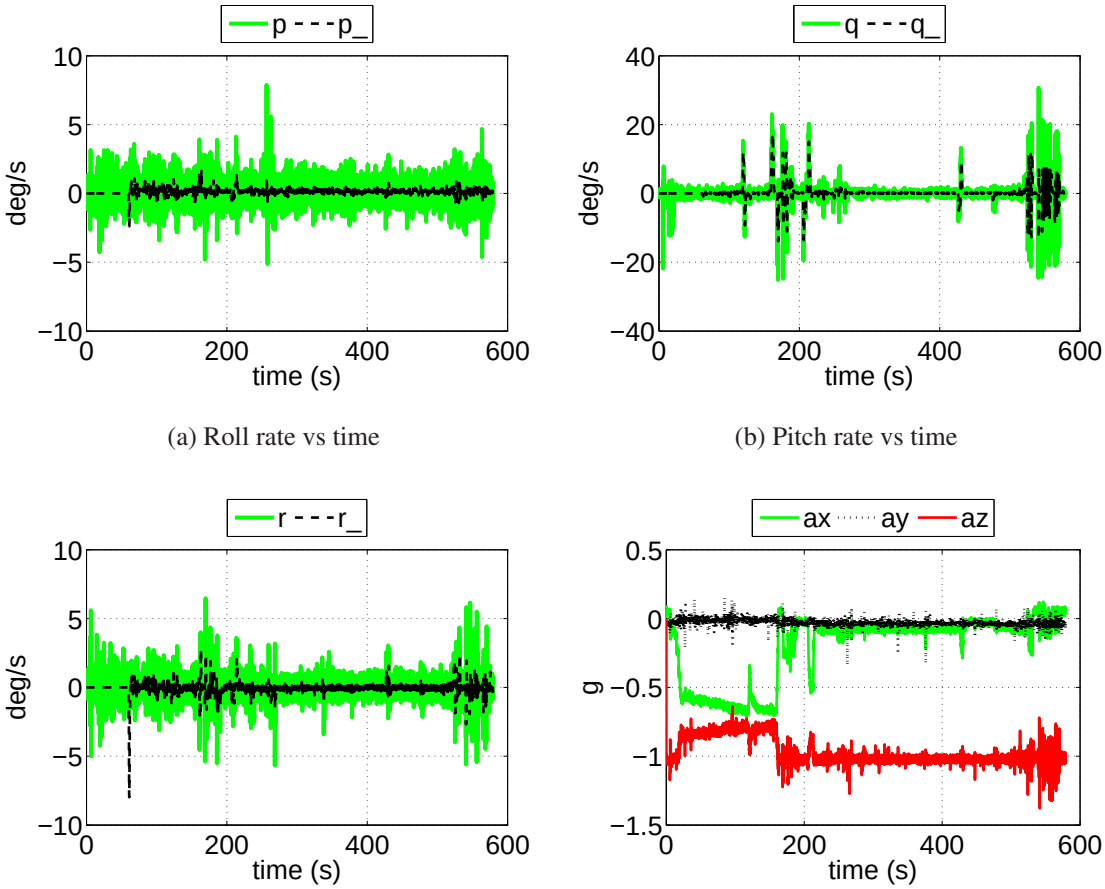
Figure 4.1: Variation of Euler Angles with Time (solid line actual, dashed line Kalman filtered)

output (elevator deflection command) are recorded on the flight control computer. Pitch angle variation due to tail movements are apparent in the pitch angle plot (Figure 4.1b and 4.1c). Roll angle and yaw angle are approximately constant during the test.

Figure 4.2 shows the variation of body angular rates and translational acceleration over time. Solid lines in Figure 4.2a, 4.2b and 4.2c are AHRS measurements of roll, pitch and yaw rates and dashed lines are Kalman filter estimates.

GPS on/off status signal is plotted in Figure 4.3a. GPS on/off status signal is '1' ('on') if all of the following conditions are true ('1'):

1. Position fix status. Binary quantity. '1' if GPS receiver has locked on to the satellites
2. Number of satellites viewed by the GPS receiver. Binary quantity. '1' if GPS receiver



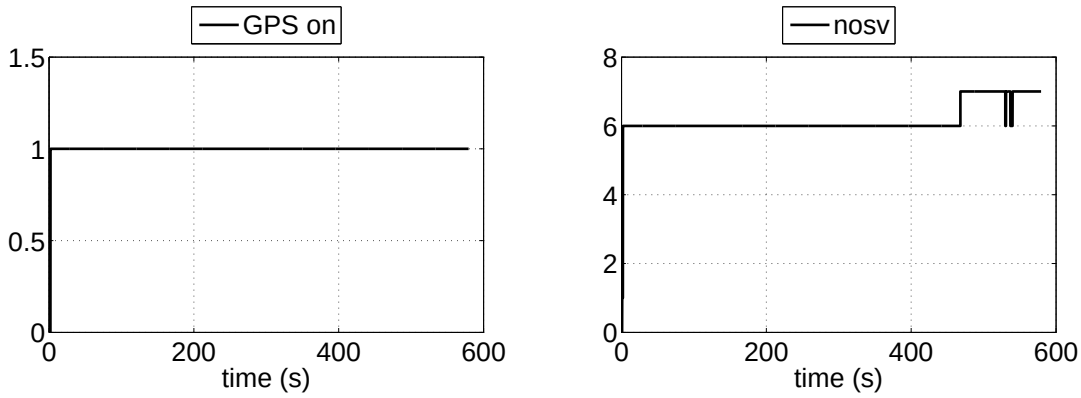
(a) Roll rate vs time

(b) Pitch rate vs time

(c) Yaw rate vs time

(d) Acceleration vs time

Figure 4.2: Variation of Body Rates and Translational Acceleration with Time (solid line actual, dashed line Kalman filtered)



(a) GPS on/off signal

(b) Number of GPS satellites viewed

Figure 4.3: Satellite Monitoring and GPS Status Signals

sees at least 5 satellites

3. Checksum. Binary quantity. '1' if the data transmitted thru RS-232 serial data link is not corrupted or lost

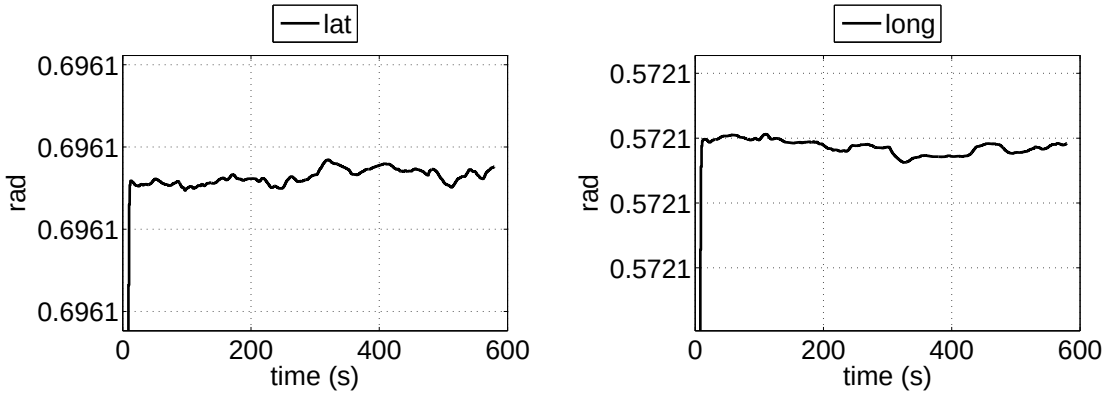
4. Maximum speed. Binary quantity. '1' if the maximum speed calculated based on GPS measurements is not higher than maximum speed of the aircraft (25 m/s)

5. Data cable disconnection. Binary quantity. '1' if two consecutive measurements of latitude, longitude and altitude are not the same.

Figure 4.3b shows the number of satellites viewed by the GPS receiver.

Latitude and longitude of the airplane measured by the GPS receiver are plotted in Figure 4.4. GPS altitude measurement and barometric altitude measurement are given in Figure 4.5a and 4.5b, respectively. Barometer altitude measurement is calibrated with the GPS altitude measurement. Calibration takes place after the GPS measurements are settled i.e. 60s after the flight control algorithm starts running [Figure 4.5b].

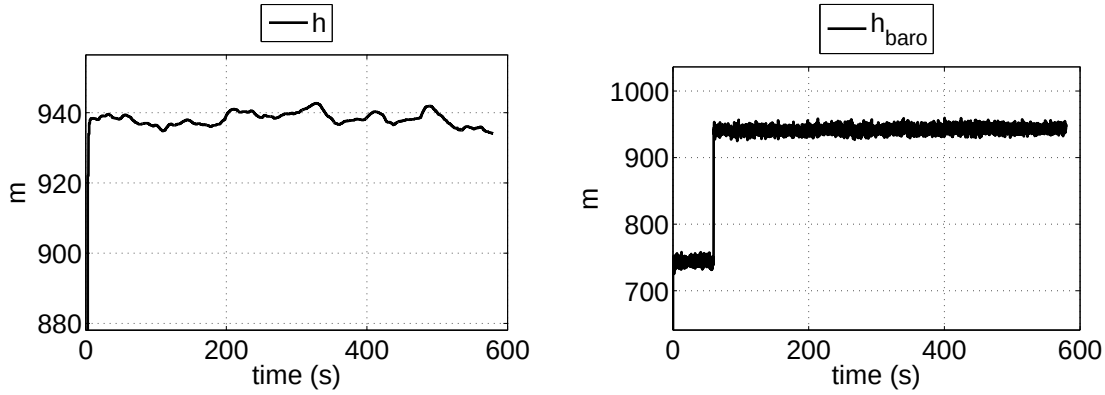
The elevator deflection command generated by the pitch-hold controller is given in Figure 4.6. Pitch angle command to the controller is zero. Therefore, the elevator is deflected to zero the non-zero pitch angle of the aircraft generated by up and down movement of the tail.



(a) Latitude vs time

(b) Longitude vs time

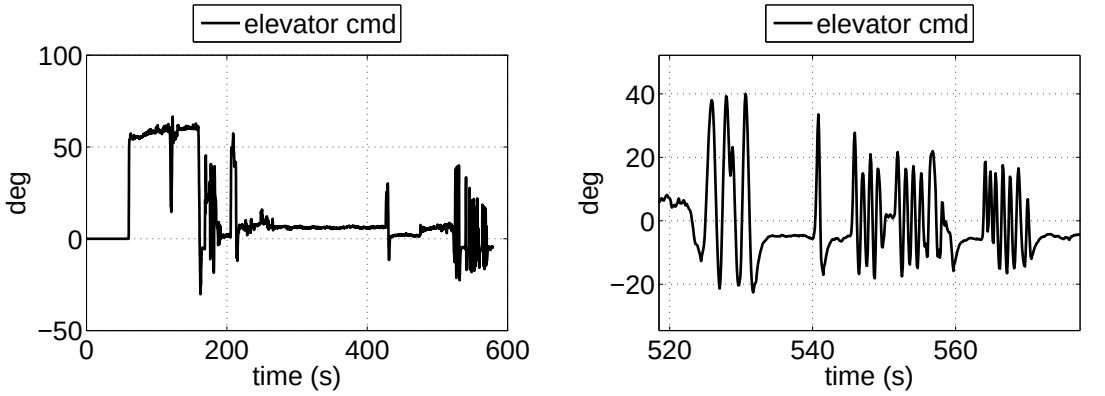
Figure 4.4: Variation of GPS Position Measurements with Time



(a) GPS altitude vs time

(b) Barometric altitude vs time

Figure 4.5: Variation of Altitude with Time



(a) Elevator deflection vs time

(b) Elevator deflection vs time (zoomed in)

Figure 4.6: Variation of Elevator Deflection with Time

4.2 Flight Testing

After checking the items on the ground, they are checked in the air. Even if the system works as expected on the ground, actual flight presents very different circumstances, such as vibration, high acceleration especially during take-off. Flight tests are performed in four phases. The first phase was the flight of the aircraft as an RC Model. The airplane was flown with radio control without any avionics payload and trimmed in the air. In the second phase the payload capacity of the airplane was tested. The airplane was flown with a dummy payload of 2.5 kg. In the third phase sensor measurements were recorded on the flight control computer. Those flights were performed to test sensor functionality under vibration and electro-magnetic interference. The last phase consisted of partial autonomous flights in which pitch-hold controller performance was tested. The following set of flight test results belong to phase three in which AHRS, GPS receiver and barometer measurements are recorded.

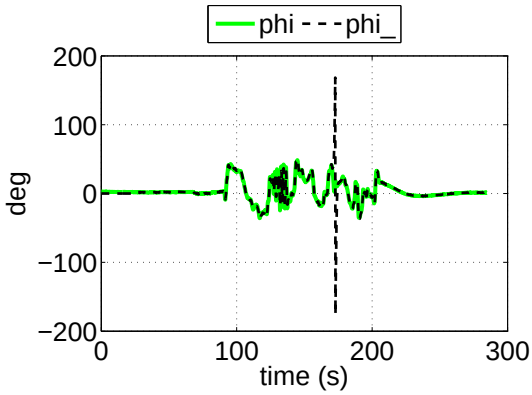
4.2.1 Piloted Flight

In this flight, the aircraft was flown under the control of the pilot with avionics onboard. It took-off, climbed to cruise altitude (50m above ground), performed altitude, pitch and roll oscillations and landed. The variation of Euler angles during the flight is given in Figure 4.7. Solid data lines are AHRS measurements of roll, pitch and yaw angles; dashed data lines are Kalman filter estimates as before.

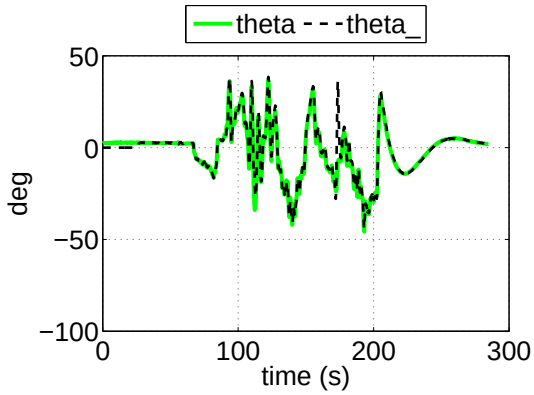
Figure 4.8 shows the variation of body angular rates and translational acceleration over time. Solid lines in Figure 4.8a, 4.8b and 4.8c are AHRS measurements of roll, pitch and yaw rates and dashed lines are Kalman filter estimates. The effect of engine-caused vibration is very clear especially in the acceleration plot (Figure 4.8d). The accelerometer noise amplitude reaches ± 1g which necessitates and justifies the presence of Kalman filter.

Latitude and longitude of the airplane measured by the GPS receiver are plotted in Figure 4.9. It is seen that GPS receiver does not send any measurements for a certain period of time. The source of this problem is found to be GPS data cable disconnection which occurs under high load factors encountered during take-off.

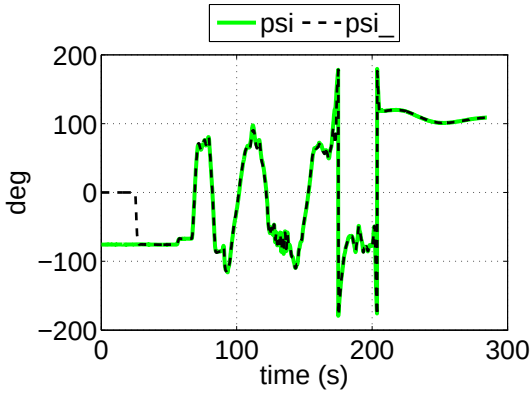
The variation of altitude with time is seen Figure 4.10. Barometric altitude is calibrated with



(a) Roll angle vs time

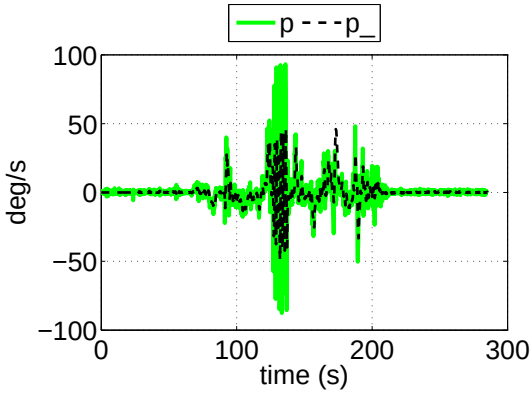


(b) Pitch angle vs time

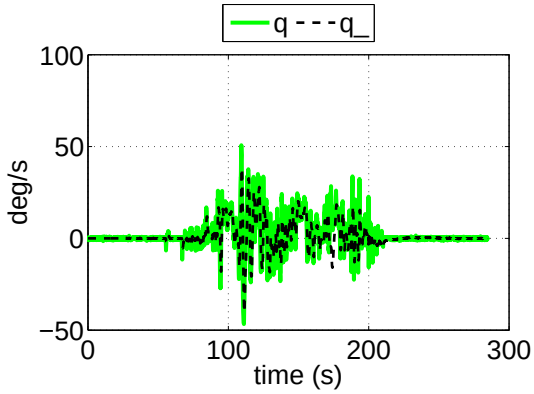


(c) Yaw angle vs time

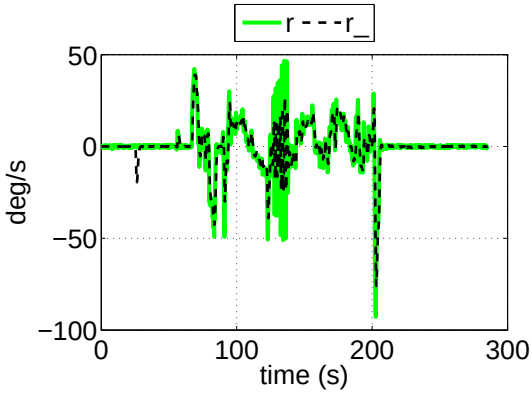
Figure 4.7: Variation of Euler Angles with Time



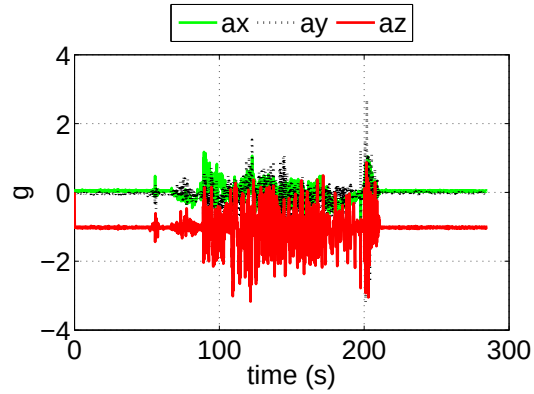
(a) Roll rate vs time



(b) Pitch rate vs time

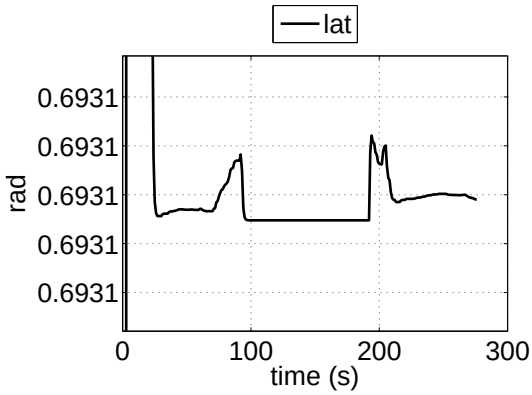


(c) Yaw rate vs time

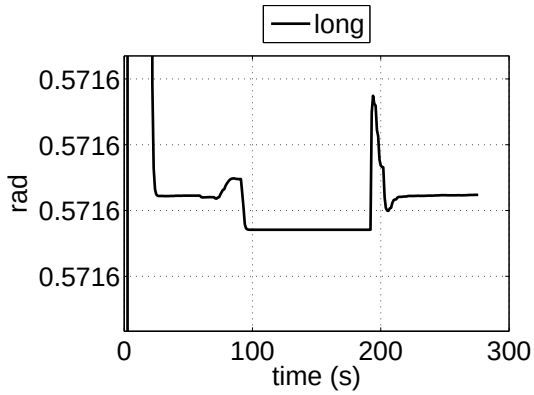


(d) Acceleration vs time

Figure 4.8: Variation of Body Rates and Translational Acceleration with Time

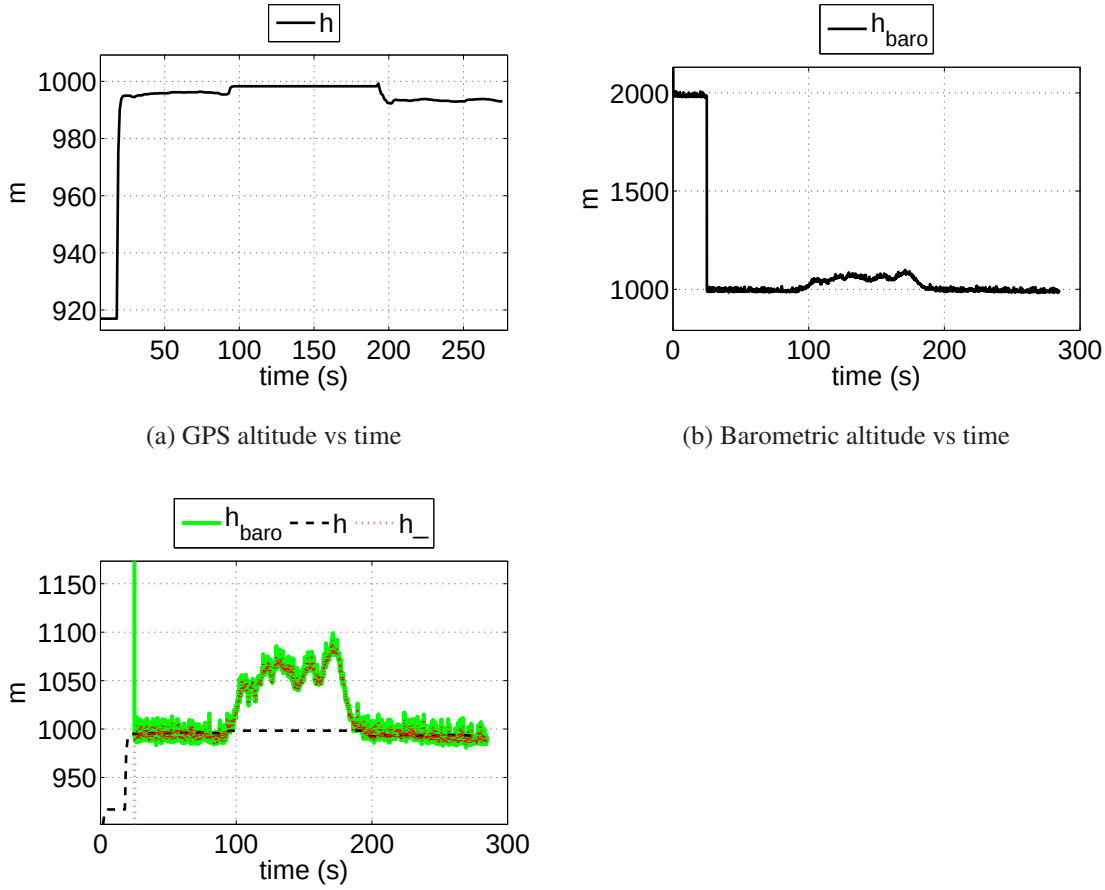


(a) Latitude vs time



(b) Longitude vs time

Figure 4.9: Variation of GPS Position Measurements with Time



(a) GPS altitude vs time

(b) Barometric altitude vs time

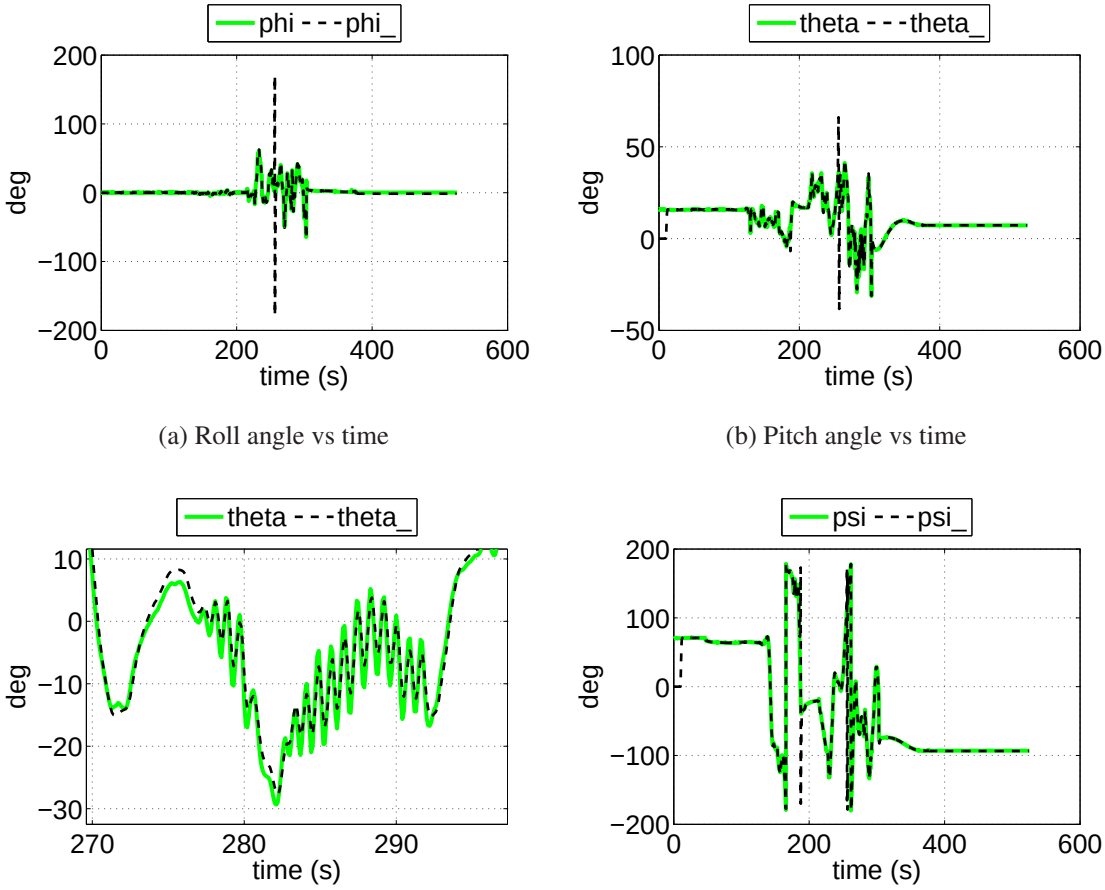
(c) Altitude comparison

Figure 4.10: Variation of Altitude with Time

GPS altitude measurement at the beginning of the flight. Barometric altitude, GPS altitude measurement and Kalman filter altitude estimate are plotted comparatively in Figure 4.10c. Barometric altitude is plotted with solid data line, GPS altitude measurement is plotted with dashed data line and Kalman filter estimate is plotted with dotted data line. It is seen that Kalman filter estimates altitude based on barometer measurements even if the GPS measurements are gone.

4.2.2 Flight with Flight Controller

In this flight, the pilot took the airplane to a safe altitude and switched the autopilot on. Pitch-hold controller tried to zero the pitch angle of the airplane. Variation of Euler angles and body angular rates are given in Figure 4.11 and Figure 4.12, respectively. Solid data lines are



(a) Roll angle vs time

(b) Pitch angle vs time

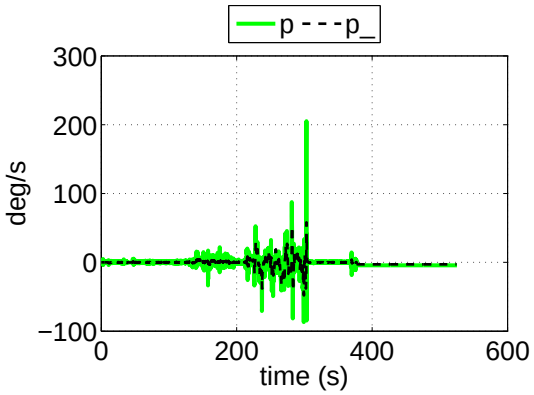
(c) Pitch angle vs time (zoomed in)

(d) Yaw angle vs time

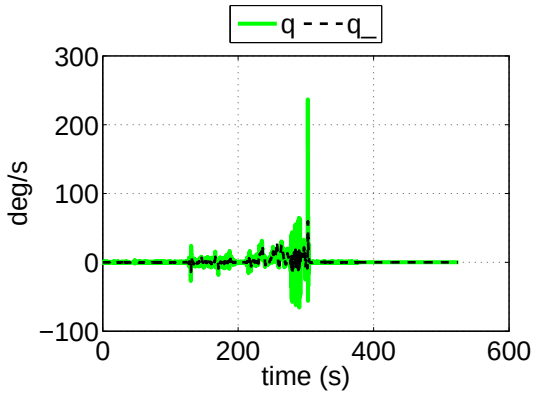
Figure 4.11: Variation of Euler Angles with Time

AHRS measurements of roll, pitch, yaw angles and rates; dashed data lines are Kalman filter estimates.

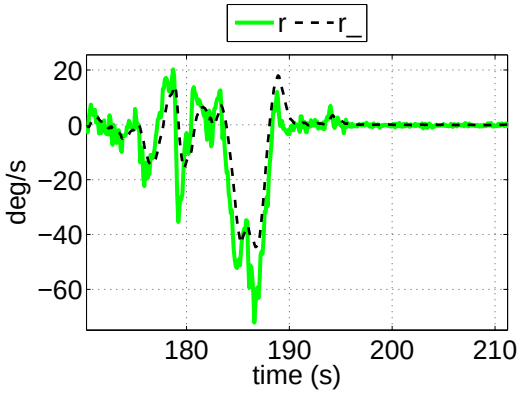
The controller starts operating at t=276s. At around t=280s, the pilot releases aileron control mistakenly and the airplane starts rolling while the pitch autopilot is in control. Later, the aileron control is taken over by the pilot and the controller stabilizes after t=283s. Figure 4.11a shows that the roll angle of the test bed is non-zero while the pitch-hold controller is in action (Figure 4.11c). This helps understand the aggressive movement of the controller output which can be explained as follows: In presence of roll angle, the pitching moment generated by the elevator deflection does not directly affect the pitch angle of the airplane. The pitch movement becomes smaller as the magnitude of the roll angle increases. Since the controller is unaware of the non-zero roll angle, it is signaled to the controller as if the elevator deflection can not generate enough pitching moment and thus it increases or decreases



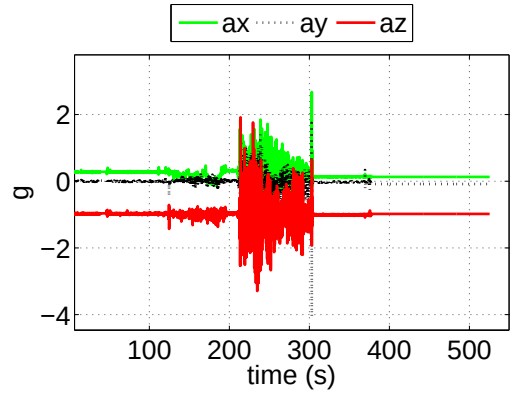
(a) Roll rate vs time



(b) Pitch rate vs time

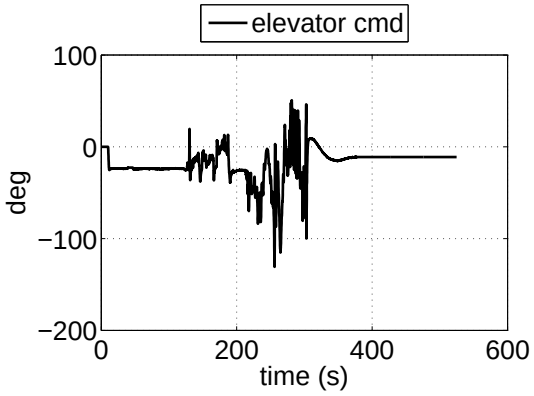


(c) Yaw rate vs time

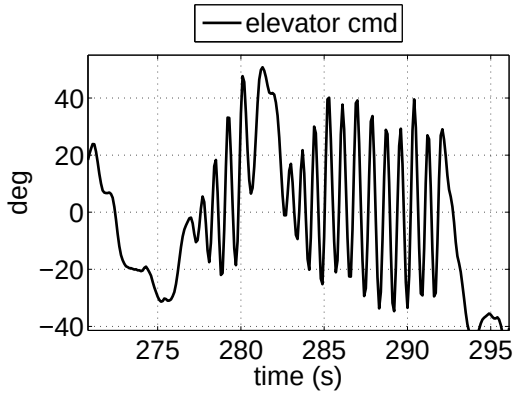


(d) Acceleration vs time

Figure 4.12: Variation of Body Rates and Translational Acceleration with Time



(a) Elevator deflection vs time



(b) Elevator deflection vs time (zoomed in)

Figure 4.13: Variation of Elevator Deflection with Time

the elevator deflection further. Therefore, it can be concluded that roll and pitch-hold control should be applied simultaneously to realize wings-level flight.

CHAPTER 5

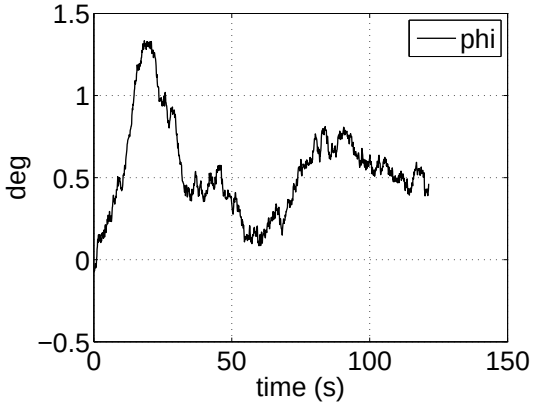
SENSITIVITY ANALYSIS

System test results presented in Chapter 4 show that AHRS measurements are affected the most from engine caused vibrations (Figure 4.8). In order to analyze to what extent the AHRS measurements are affected by the external mechanical vibrations due to engine, a sensitivity analysis is conducted.

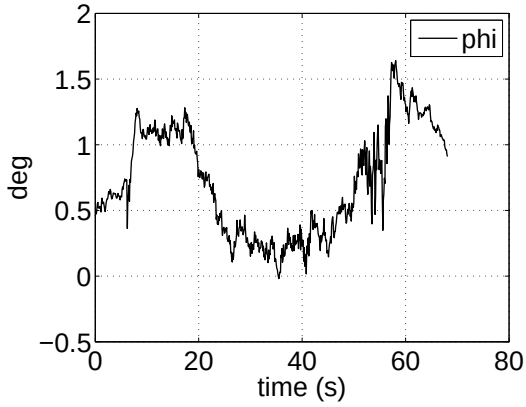
Two ground tests are performed to analyze the noise sensitivity of accelerometers and rate gyros to mechanical vibrations. In both of the tests, the testbed is kept stationary on the ground. In the first test, engine is switched off and AHRS measurements are recorded. In the second test, the engine is switched on, throttle is increased to maximum (100%), then decreased to zero and AHRS measurements are recorded. Test results are plotted side-by-side for each measurement.

The amplitude of the noise on Euler angle measurements due to external vibrations is on the order of 0.1° which corresponds to 0.0014V (Figure 5.1b, 5.2b and 5.3b). This measurement error is not very significant for the control of the aircraft since the attitude error tolerances are on the order of 1°. The noise appearing in the Euler angle plots is purely due to external mechanical vibrations. The artifacts on the measurements due to transient magnetic interference is filtered out using a complementary filtering algorithm that runs in the microcontroller of the AHRS. This process is called gyro-stabilization.

Since the airplane is held stationary on the ground during these tests, an increasing reaction force is generated by the ground as the engine rpm is increased. This reaction force from ground is mostly effective in longitudinal plane in which pitch angle measurements are taken. This force is secondarily effective on roll angle measurements and the least effective in yaw angle measurements. This explains the relatively large variation in the amplitude of the pitch

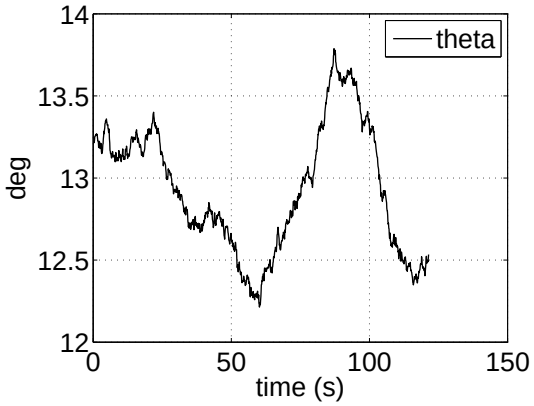


(a) Roll angle vs time (engine-off)

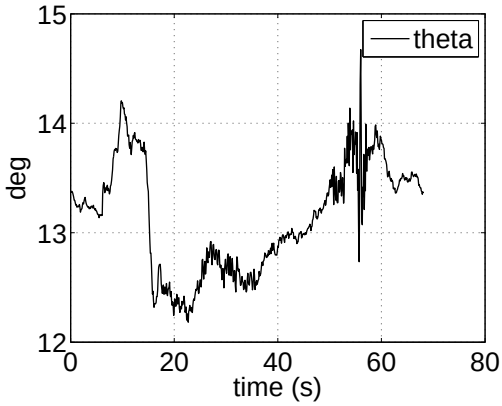


(b) Roll angle vs time (engine-on)

Figure 5.1: Variation of Roll Angle with Time

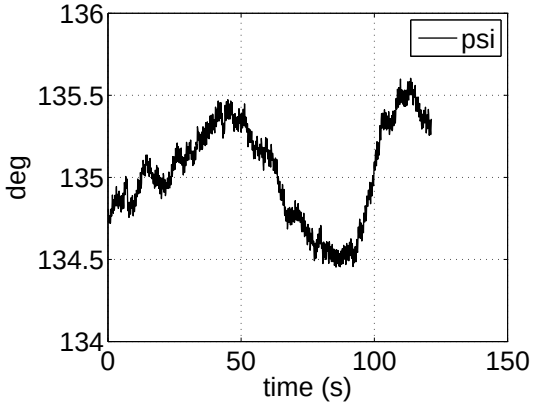


(a) Pitch angle vs time (engine-off)

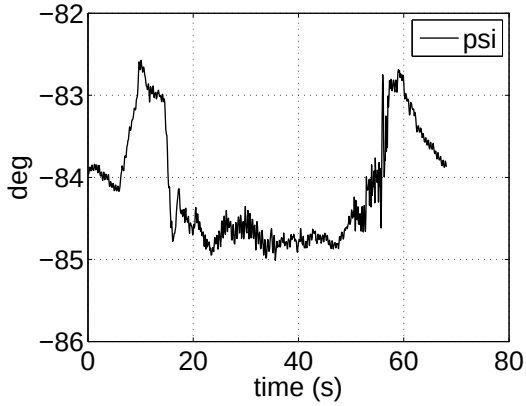


(b) Pitch angle vs time (engine-on)

Figure 5.2: Variation of Pitch Angle with Time

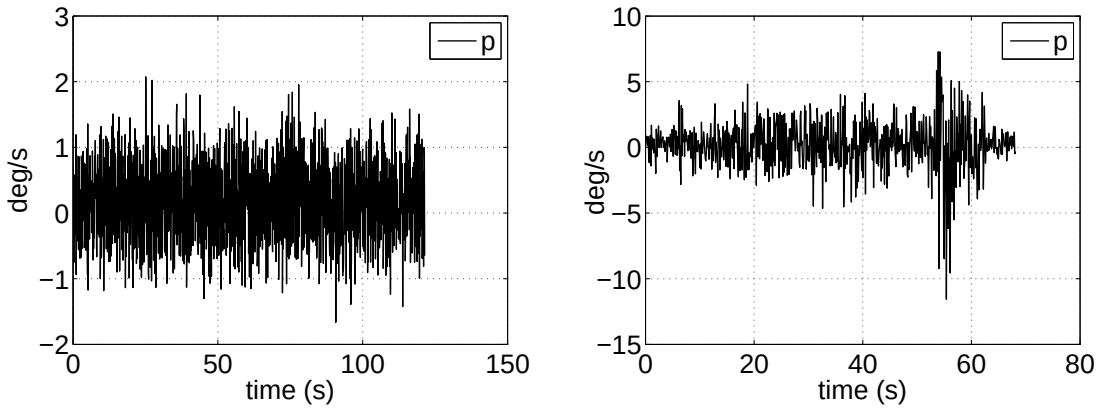


(a) Yaw angle vs time (engine-off)



(b) Yaw angle vs time (engine-on)

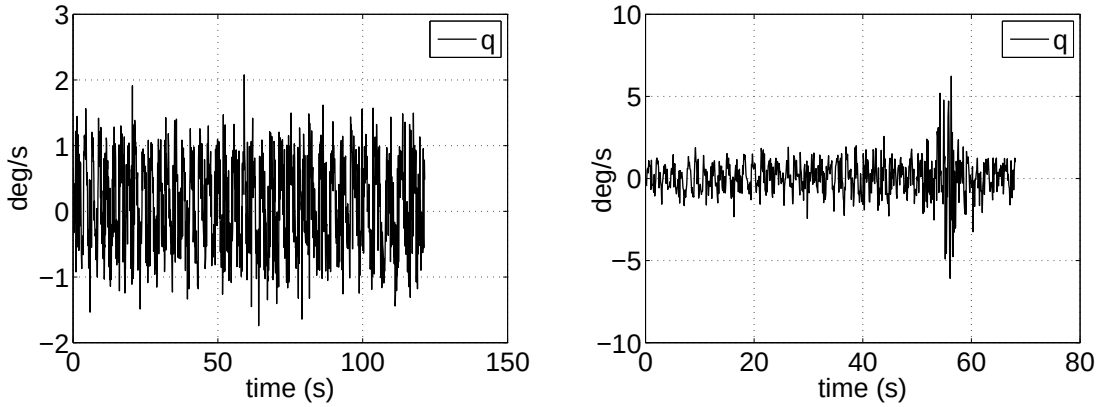
Figure 5.3: Variation of Yaw Angle with Time



(a) Roll rate vs time (engine-off)

(b) Roll rate vs time (engine-on)

Figure 5.4: Variation of Roll Rate with Time



(a) Pitch rate vs time (engine-off)

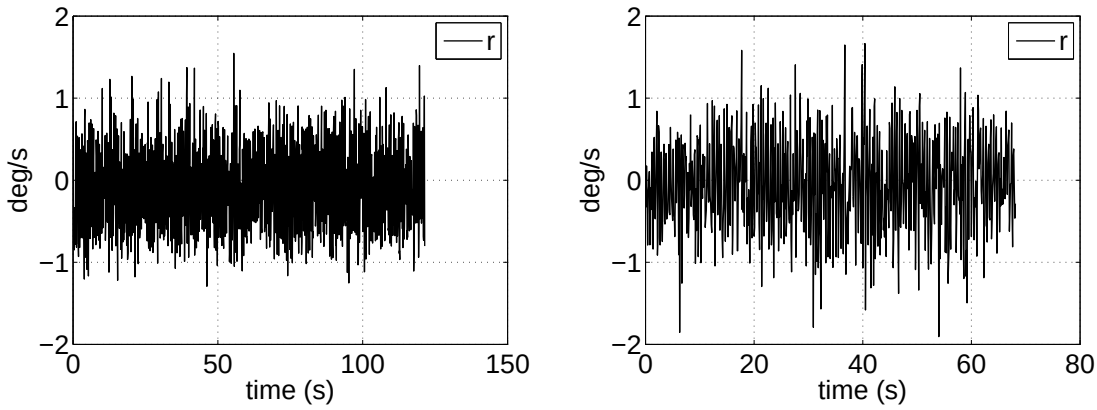
(b) Pitch rate vs time (engine-on)

Figure 5.5: Variation of Pitch Rate with Time

angle measurement noise which occurs at around time=55s where the throttle input reaches 100% (Figure 5.2).

The amplitude of the noise on roll rate and pitch rate measurements due to external vibrations is on the order of 1°/s (Figure 5.4b and 5.5b). At full throttle input, the amplitude of the noise in roll rate measurement reach 10°/s which corresponds to 0.0833V. It is seen that yaw rate measurement is the most ignorant to external vibrations (Figure 5.6b), whereas the roll rate measurement is the mostly affected (Figure 5.4b).

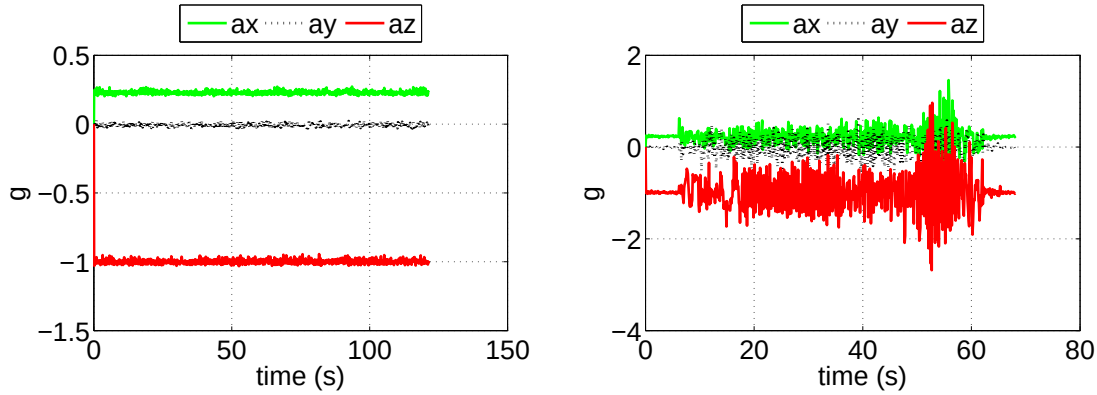
The amplitude of the noise on acceleration measurements due to external vibrations is on the order of 0.1g (Figure 5.7). At full throttle input, the amplitude of the noise reach 1g which corresponds to 0.5V. Since position states are obtained by taking double integral of the



(a) Yaw rate vs time (engine-off)

(b) Yaw rate vs time (engine-on)

Figure 5.6: Variation of Yaw Rate with Time



(a) Acceleration vs time (engine-off)

(b) Acceleration vs time (engine-on)

Figure 5.7: Variation of Translational Acceleration with Time

accelerometer measurements, the measurement noise will be amplified in the position calculation unless a filtering algorithm is applied. In the thesis work, extended Kalman filter algorithm is used to reduce the amplitude of the noise (see Appendix A).

The rate and accelerometer measurements obtained from AHRS are not gyro-stabilized measurements and thus the measurement noise might result from electromagnetic interference as well as external mechanical vibrations.

CHAPTER 6

LIMITATIONS AND CONSTRAINTS

The users of the developed UAV testbed should consider the following limitations and constraints of the testbed while designing their controller.

1. Sampling Rate and Simulation Time Step Limitation

   - AHRS supports data output rates of 100, 111, 125 and 143Hz for processed outputs such as Euler angles, quaternions, orientation matrix and gyro-stabilized accelerometer, rate and magnetometer vectors. Sampling rate of uncompensated, raw sensor measurements can be set as 166, 200 or 333Hz. Sampling rates are adjusted by setting certain EEPROM addresses to appropriate values[21]. The baud rate setting must be 38400 or 115200; 19200 does not support all these rates.
   - GPS data update rate is 1Hz.
   - A/D converter on Athena II single-board computer has a maximum sampling rate of 100KHz.
   - Servo driver board can operate at most 600Hz.

   In the simulation, sample times of the interface models are obtained by taking the inverse of the sampling rates, i.e. 1/sampling rate or 1/frequency.

   Recommended sample time settings in the simulation model are as follows; AHRS interface- 0.01s (100Hz), GPS interface- 1s (1Hz), analog input interface- 0.01s (100Hz), servo driver board interface- 0.01s (100Hz). The simulation sample time is set as 0.01s (100Hz), but it is recommended to be taken as small as possible. As a rule of thumb, the controller sampling rate should be at least 10 times faster than the fastest mode of the airplane[14]. For longitudinal dynamics, the fastest mode is the short period mode

and the natural frequency of the short period mode is 1.3Hz (Table 3.1). Therefore, the controller sampling rate should be at least 10Hz; in other words, the sample time should be at most 0.1s.

2. Processor Speed Limitation

   The processor speed of the Athena II single-board computer is 800MHz. Controller algorithms that require faster processors can not be tested on the UAV testbed.

3. Baud Rate Limitation

   Baud rate limitation for RS-232 serial connections are as follows:

   - AHRS baud rate: 19200 / 38400 (default) / 115200
   - GPS receiver baud rate: 4800 / 9600 / 19200 / 38400 / 57600 / 115200
   - Servo driver board: 19200

4. Latency Constraint

   Latency is another constraint of the developed system. It is related to analog measurements. Single-board computer initiates a request for data and Analog-to-Digital converter on the SBC starts sending the requested data. Time that passes between this request and the beginning of actual data transfer is defined as latency. Latency increases as the number of analog inputs increases (see Appendix B). For example, latency becomes 56 μs for 5 analog inputs and it is added to the sampling time that can be achieved without I/O hardware. The suggestion for this constraint is to use the possible least number of analog input channels or to do the analog conversion on a separate board and send the digitized data through RS-232 serial port.

5. Operating Temperature Limits

   - Flight computer (Athena II SBC) temperature limits: -40, +85°C
   - AHRS temperature limits: -40, +70°C (with enclosure), -40, +85°C (without enclosure)
   - GPS receiver temperature limits: -40, +85°C
   - DC/DC converter temperature limits: -40, +85°C
   - Wireless access point (WAP) temperature limits: 0, +60°C
   - Potentiometer temperature limits: -40, +125°C

- Barometer temperature limits: -40, +125°C
- Autopilot/Remote Pilot Switch temperature limits: -25, +100°C

6. Actuator Limits

   Actuator limits are used to saturate the unrealizable autopilot commands. Elevator, aileron and rudder deflection limits should be set as [-20°, +20°] which are the physical deflection limits of the control surfaces and the throttle limits should be set as [0%, 100%].

7. Analog Input Resolution

   A/D converter on the Athena II SBC is a 16-bit converter. Therefore, the resolution for 0-10V input range becomes 10V/65535 = 153μV. The resolution changes for different input ranges.

CHAPTER 7

CONCLUSION

The development and testing of a UAV test bed is described in this thesis. A UAV test bed is an unmanned aerial platform developed to test different flight control software. It is equipped with an avionics system composed of commercial-off-the-shelf sensors and boards implemented on an RC model airplane. The advantages of the designed system over existing systems are listed as follows:

- Easier coding due to the use of high-level language in programming. Matlab/Simulink and xPC Target let the users, especially students, model, simulate and test novel flight control algorithms without getting lost in lines of codes. It accelerates the design and testing phase of flight controllers
- Component and interface (e.g. RS-232) level modularity makes it possible to upgrade avionics system parts to the state-of-the-art technology
- Making use of COTS products remove dependency on custom manufacturers and increases the number of alternatives which also decreases the overall system price

In the thesis study, avionics system integration is successfully completed. State estimation is accomplished by Kalman filter and pitch-hold controller is tested as a sample application. Aircraft state estimates are obtained from Kalman filter based on AHRS, GPS receiver and barometer measurements. Servo commands generated by the controller are transmitted to the elevator and throttle servos over servo driver board. However, there were also problems which turned out to be invaluable experience for future studies. It is seen that system weight -due to avionics payload- was too much which makes the airplane fly at its limits. This was especially

critical during take-off which caused a severe crash. To improve fail-safety, power sources of the engine, avionics and actuators are separated. Also, the engine battery is placed away from the sensors to avoid electro-magnetic interference.

The following items are general remarks based on experience and will help future users of the UAV test bed:

- Reduce the total weight of the testbed as much as possible. Every single gram is a burden to the airplane and reduce the effectiveness of flight tests
- Separate the power sources of the engine, avionics and actuators to improve safety. In that way, the control of the airplane will still be possible even if the engine or the avionics battery run out
- Place the engine away from the sensors and flight computer to avoid electro-magnetic interference
- Ensure air flow through the airplane during flight. Increased temperature may damage the avionics and batteries
- LiPo batteries should be handled carefully. Excessive discharging (below 3.0V/cell) or overcharging (above 4.2V/cell) LiPo batteries may cause permanent cell damage, fire and explosions [27].
- If possible, perform analog/digital (A/D) conversion on a separate board with a separate processor. The main processor may not perform real-time operation due to the latency caused by A/D conversion (see Section B.4)
- Try to avoid flight tests with a single-channel controller. Interference of pilot-controlled states or wind may lead to misleading conclusions
- Pass sensor measurements through filters
- Place the payload (avionics and batteries) in such a way that the original center of gravity location of the airplane is not altered more than a few centimeters

7.1 Suggestions for Future Work

Shortcomings of the system identified from the test results provide a basis for future improvements. Suggested future improvements and studies are listed as follows:

- In order to avoid excessive weight, the casing of the flight computer should be removed and some other electro-magnetic protection can be employed. Wireless access point is also one of the heavy components. It can be replaced by a lighter device
- A/D conversion should be performed on a separate board with a separate processor so that the main processor can be devoted completely to real-time operation
- A tracking antenna may be used on the ground to track the UAV test bed. Commercial WAP antenna does not perform well on mobile objects
- Kalman filter covariance matrices may be further tuned to improve filter performance
- System identification or parameter estimation processes may be performed to improve the fidelity of the current mathematical model of the airplane

REFERENCES

[1] Christiansen, R.S., "Design of an Autopilot for Small Unmanned Aerial Vehicles ", MSc Thesis, Brigham Young University, 2004

[2] Jung, D., Levy, E.J., Zhou, D., Fink, R., Moshe, J., Earl, A., and Tsiotras, P., "Design and Development of a Low-Cost Test-Bed for Undergraduate Education in UAVs"

[3] Komlosy III, John A., "Applications of Rapid Prototyping to the Design and Testing of UAV Flight Control Systems", MSc Thesis, Naval Postgraduate School, 1998

[4] Lim, Bock-Aeng, "Design and Rapid Prototyping of Flight Control and Navigation System for an Unmanned Aerial Vehicle", MSc Thesis, Naval Postgraduate School, 2002

[5] Khan, A.D., "The Design and Development of a Modular Avionics System", MSc Thesis, Georgia Institute of Technology, 2001

[6] Computer form factor - Wikipedia, the free encyclopedia. Retrieved 14 April 2011 <http://en.wikipedia.org/wiki/Computer_form_factor>

[7] PC/104 Consortium - Specifications. Retrieved 14 April 2011 <http://www.pc104.org/specifications.php>

[8] Rapid Prototyping for Embeded Control Systems, Mathworks. Retrieved 21 December 2010 <http://www.mathworks.com/rapid-prototyping/embedded-control-systems.html>

[9] xPC Target Interactive Hardware Selection Guide. Retrieved 28 April 2011 <http://www.mathworks.com/support/product/XP/productnews/interactive_guide/wiz_script.cgi>

[10] Erkkinen, T., "Model Style Guidelines for Flight Code Generation", American Institute of Aeronatics and Astronautics Conference Proceedings, 2005

[11] Lizarraga, M.I., Dobrokhodov, V., Elkaim, G.H., Curry, R., and Kaminer, I., "Simulink Based Hardware-in-the-Loop Simulator for Rapid Prototyping of UAV Control Algorithms", American Institute of Aeronatics and Astronautics Conference Proceedings, Seattle, Washington, 2009

[12] Cakir, Z., Demir, B., Tekinalp, O., and Yavrucuk, I., "Flight Control System Design and Integration for a Small UAV Testbed", 5th Ankara International Aerospace Conference Proceedings, Ankara, 2009

[13] Işık, Sinem, "Flight Control System Design for an Over Actuated UAV Against Actuator Failures", MSc Thesis, Middle East Technical University, 2010

[14] Franklin, G. F., Powell, J. D., Workman, M. L., "Digital Control of Dynamic Systems", Addison-Wesley , Third edition, 1999

[15] ”The USAF Stability and Control Digital Datcom”. AFFDL-TR-79-3032

[16] Real-Time Workshop® Version 7.0 Getting Started Guide

[17] Support - Supported / Compatible Compilers - Release 2007b. Retrieved 25 April 2011 <http://www.mathworks.com/support/compilers/release2007b/>

[18] PC/104 consortium. Retrieved 22 December 2010 <http://www.pc104.org/>

[19] Small Form-factor Embedded Computing Solutions: SBCs, COMs, FeaturePak, PC/104, SUMIT, data acquisition, and I/O. Retrieved 22 December 2010 <http://www.diamondsystems.com>

[20] 3DM-GX1®:: MicroStrain®; AHRS Orientation Sensor. Retrieved 23 December 2010 <http://www.microstrain.com/3dm-gx1.aspx>

[21] 3DM-GX1® Fastest Data Output Rates. Retrieved 10 May 2011 <http://www.microstrain.com/tech_notes/TechNote_3DM-GX1_Fastest_Rates.pdf>

[22] National Marine Electronics Association - NMEA. Retrieved 25 December 2010 <http://www.nmea.org/ >

[23] MPXA6115A Technical Data, Revision 3, Freescale Semiconductor,Inc., 2007

[24] International Organization for Standardization, Standard Atmosphere, ISO 2533:1975, 1975

[25] World Geodetic System. Retrieved 22 April 2011 <https://www1.nga.mil/ProductsServices/GeodesyGeophysics/WorldGeodeticSystem/Pages/default.aspx >

[26] How R.C. Servo Motors Work. Retrieved 30 December 2010 <http://www.digitalnemesis.com/info/docs/rcservo/>

[27] Lipo Battery Charging & Safety Guide. Retrieved 12 May 2011 <http://www.rchelisite.com/lipo_battery_charging_and_safety_guide.php>

[28] RXMUX Functional Specification, Revision 1.3, Reactive Technologies

[29] Ogata, Katsuhiko, Modern Control Engineering, Englewood Cliffs, N.J.: Prentice Hall, 1990

[30] Kutlu, Aykut, ”Design of Kalman Filter Based Attitude Determination Algorithms for a LEO Satellite and for a Satellite Attitude Control Test Setup”, MSc Thesis, Middle East Technical University, 2008

[31] Zarchan, P., and Musoff, H., ”Fundamentals of Kalman Filtering & a Practical Approach”, American Institute of Aeronatics and Astronautics Conference Proceedings, Virginia, USA, 2000

[32] Kalman, R.E., ”A New Approach to Linear Filtering and Prediction ”, Journal of Basic Engineering, Vol.82, No.1, March 1960, 35-46

[33] Welch, G., and Bishop, G., ”An Introduction to the Kalman Filter”, University of North Carolina at Chapel Hill, Chapel Hill, NC, 2006

[34] Titterton, D.H., and Weston, J.L., Strapdown Inertial Navigation Technology, Reston, VA : American Institute of Aeronautics and Astronautics, 2004

[35] Çiçek, S., "CCS C ile PIC Programlama", İstanbul, Altaş Yayıncılık, 2007

[36] Bayezit, I., Ates, S., Aksungur, M., and Inalhan, G., "The Autonomy of Fixed-wing Aerial Vehicles and Experimental Design Steps on Implementing Autonomous Navigation, Landing and Take off Operations for the Trainer60 Model Aircraft", 5th Ankara International Aerospace Conference Proceedings, Ankara, 2009

[37] Kingston, D., Beard, R., McLain, T., Larsen, M., and Ren, W., "Autonomous Vehicle Technologies for Small Fixed Wing UAVs", American Institute of Aeronatics and Astronautics Conference Proceedings, 2003

[38] Blauwe, H.D., Bayraktar, S., Feron, E., and Lokumcu, F., "Flight Modeling and Experimental Autonomous Hover Control of a Fixed Wing Mini-UAV at High Angle of Attack", American Institute of Aeronautics and Astronautics Guidance, Navigation and Control Conference Proceedings, Hilton Head, South Carolina, 2007

[39] Ng, T.,L., "Miniature Real-Time Embedded Systems for Control Applications", MSc Thesis, Polytechnic University, 2005

[40] Tekinalp, O., Yavrucuk, İ., Çakır, Z., Demir, B., Elmas, T. Ç., and Işık, S., "Artık Eyleyicili Otonom Hava-Uzay Araçlarının Kontrolü", Project Final Report submitted to The Scientific and Technological Research Council of Turkey, Project no: 107M346, 2010

APPENDIX A

Kalman Filter

Sensors mentioned in Section 2.1.1 measure acceleration, angular rates and the position of the airplane with respect to the inertial reference frame. However, the position, velocity or attitude (orientation) of the airplane are controlled with respect to a local frame. Therefore, navigation equations are solved to compute the position, velocity and orientation of the airplane with respect to local North-East-Down (NED) frame (Figure A.1)- also known as 'navigation frame'.

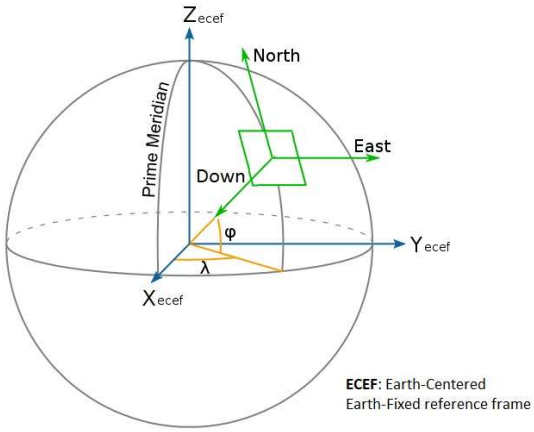


Figure A.1: North-East-Down Reference Frame

This process is called *Navigation Frame Mechanization*. In navigation frame mechanization, position, velocity and orientation of the airplane are calculated with respect to local NED frame using measurements of inertial sensors such as gyroscopes and accelerometers. The sensor suite composed of gyroscopes and accelerometers is called an *Inertial Measurement Unit (IMU)*. AHRS covers IMU and on top of that it calculates the orientation of the airplane using its embedded micro-controller.

Navigation frame mechanization incorporates accelerometer measurements with local grav-

ity correction and Earth's coriolis acceleration to obtain the acceleration with respect to the navigation frame (NED). Orientation of the airplane is obtained from the navigation frame to body frame transformation matrix using gyroscope measurements (angular rates). Navigation frame mechanization steps are listed below:

1. Calculation of the transformation matrix

   The transformation matrix from body to navigation frame (C_b^n) is calculated using quaternion measurements from AHRS (Eqn. A.1).

$$C_b^n = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{A.1}$$

   where

   $q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}$ is the quaternion vector

2. Calculation of the specific force

   Specific force is composed of 3-axis accelerometer measurements. Accelerometers measure acceleration along body-axes. Specific force along body-axes, f^b, are converted to navigation frame using the transformation matrix from body-to- navigation frame (A.1).

$$f^n = \begin{bmatrix} f_{North} & f_{East} & f_{Down} \end{bmatrix} = C_b^n f^b \tag{A.2}$$

3. Calculation of the time rate of change of transformation matrix

$$\dot{C}_b^n = C_b^n \tilde{\omega}_{nb}^b \tag{A.3}$$

   ω_{nb}^b is the vector representing the angular velocity of the aircraft with respect to navigation frame. This vector is composed of body rates measured by the rate gyros of AHRS (ω_{ib}^b), Earth's rotational velocity (ω_{ie}^n) and transport rate (ω_{en}^n). Transport rate is the angular velocity of the navigation frame with respect to Earth frame. Therefore, ω_{nb}^b is expressed as;

$$\omega_{nb}^b = \omega_{ib}^b - C_n^b(\omega_{ie}^n + \omega_{en}^n)$$

   where

   $\omega_{ib}^b = \begin{bmatrix} p & q & r \end{bmatrix}$; gyroscope measurements written in aircraft body axes

$\omega_{ie}^{n} = \begin{bmatrix} \Omega cos(L) & 0 & -\Omega sin(L) \end{bmatrix}$; Earth's rotational velocity written in navigation frame where $\Omega \approx 15°$/h, Earth's rotational speed; L= latitude

$\omega_{en}^{n} = \begin{bmatrix} \frac{V_E}{R_E} & \frac{-V_N}{R_E} & \frac{-V_E tan(L)}{R_E} \end{bmatrix}$

where

$V_e^n = \begin{bmatrix} V_N & V_E & V_D \end{bmatrix}$; velocity of the aircraft with respect to Earth frame expressed in navigation frame

$R_E \approx 6378$ km; Earth's radius

$\tilde{\omega}_{nb}^{b}$ in Eqn.A.3 is the skew-symmetric form of $\omega_{nb}^{b} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}$.

$$\tilde{\omega}_{nb}^{b} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

4. Calculation of the time rate of change of aircraft velocity with respect to Earth frame

$$\dot{V}_e^n = f^n - (2\omega_{ie}^n + \omega_{en}^n) \mathrm{x} V_e^n + g_l^n \tag{A.4}$$

where

g_l^n is the local gravity vector expressed in navigation frame

Velocity and position of the airplane with respect to the navigation frame are obtained by taking the integral of the acceleration (Eqn. A.4) and angular position of the airplane (quaternions, and thus Euler angles) is obtained by taking the integral of Eqn. A.3. However, navigation frame mechanization is not a reliable way to calculate position and velocity states of the airplane. The problem with the navigation frame mechanization is related to the integration of acceleration. Since there are sensor measurements in the mechanization process, the measurement errors such as noise and drift will also be involved in the process. Taking the integral of the acceleration amplifies these errors as well. Therefore, aiding sensor data (GPS and barometer) is used to correct inertial measurement errors and calculate the position and velocity of the airplane accurately.

Sensor measurements obtained from AHRS, GPS receiver and barometer are combined using Extended Kalman Filter (EKF) ([30], [34]). Separate filters are used for attiude estimation and navigation estimation (Figure A.2). Attitude filter estimates orientation of the airplane

based on the orientation and angular rate measurements from AHRS. Body angular rates, ω_{ib}^{b}, and quaternions are the states estimated by the attitude Kalman filter.

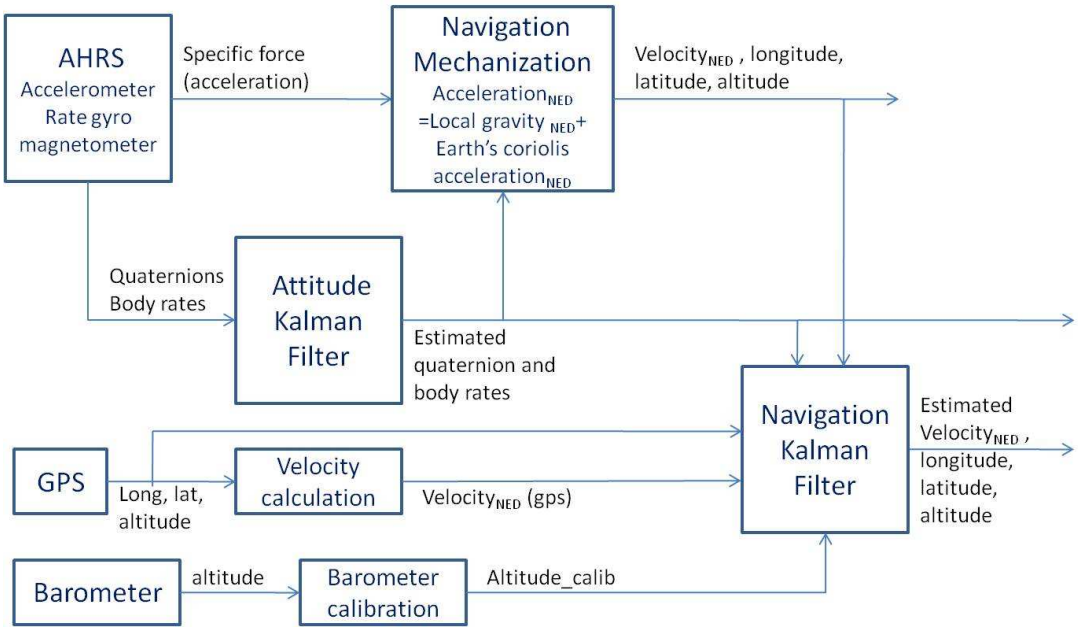


Figure A.2: State Estimation Schematic

Navigation filter, on the other hand, estimates position (longitude, latitude and altitude) and velocity (V_e^n) of the airplane based on GPS and barometer measurements as well as navigation frame mechanization outputs, i.e. NED velocity and orientation matrix.

Performance of Kalman filter depends on the tuning of process and measurement noise covariance matrices as well as appropriate initial selection of error covariance matrix. In order to avoid divergence of the filter, a correct initial state estimate should also be input to the filter. Filter tuning mainly has two purposes: converge to the closest estimation in the shortest time. Initial error covariance matrix (P_0) determines the initial convergence rate. This matrix does not effect the long-term performance of the filter. Larger inputs to the P_0 matrix increases convergence rate. In this case, the filter will prefer the information from the first measurements over the information already in the model. If the initial state values are known exactly P_0 will be a zero matrix. Measurement noise covariance matrix (R) is usually measured prior to operation of the filter. Measurement error covariance is measured based on the measurement error characteristics of the sensors. The measurements are trusted more and more rather than filter estimates as R approaches to zero. The determination of process noise

covariance Q is rather difficult, since it is not possible to directly observe the process being estimated. Therefore, whether or not these matrices are generated on a rational basis, the best filter performance is obtained by tuning the filter parameters Q and R.

Final tuned covariance matrices of attitude Kalman filter are as follows;

Initial error covariance matrix; $P_0 = 10^{-6} I_{7x7}$

Process noise covariance matrix; $Q = 10^{-9} I_{7x7}$

Measurement noise covariance matrix;

$$R = \begin{bmatrix} 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-6} \end{bmatrix}$$

Estimated states of attitude Kalman filter are;

$$x_{att} = \begin{bmatrix} p & q & r & q_1 & q_2 & q_3 & q_4 \end{bmatrix}$$

Final tuned covariance matrices of navigation Kalman filter are as follows;

Initial error covariance matrix;

$$P_0 = \begin{bmatrix} 10^{6} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{6} \end{bmatrix}$$

Process noise covariance matrix;

$$Q = \begin{bmatrix} 10^{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{3} \end{bmatrix}$$

Measurement noise covariance matrix;

$$R = diag(2^2, 2^2, 2^2, 8.73e^{-6}, 8.73e^{-6}, 10^2, 10^2, 10^2, 10^2, 8.73e^{-7}, 8.73e^{-7}, 5^2, 30^2)$$

The first six entities in the diagonal measurement noise covariance matrix are the squares of standard deviations of AHRS measurements and calculations for 3-axis velocity, latitude, longitude and altitude. Following six entities are the squares of standard deviations of GPS measurements of the same states. The last entity is the square of standard deviation of barometer altitude measurement.

Estimated states of navigation Kalman filter are;

$$x_{nav} = \begin{bmatrix} V_N & V_E & V_D & \text{latitude} & \text{longitude} & \text{altitude} \end{bmatrix}$$

APPENDIX B

Hardware Interfaces

xPC Target library has a wide range of commercial-off-the-shelf I/O driver support. AHRS, GPS receiver and servo driver board communicates with the flight control computer thru RS-232 serial connection. The Simulink model of serial communication is given in Figure B.1. 'Baseboard Serial F' block in xPC Target library is used for serial communication. RS-232 communicaton settings are adjusted from the block parameters dialog box.

B.1 AHRS Communication Interface

AHRS sends measurement data over RS-232 serial data link. It is connected to port COM3 of the flight control computer. Port settings are as follows;

- IRQ : 9
- Port address: 0x3e8
- Baud rate : 38400
- Parity : None
- Data bits : 8
- Stop bits : 1
- Hardware FIFO size : 16 deep
- Receive FIFO interrupt level : 1
- Auto RTS/CTS : on

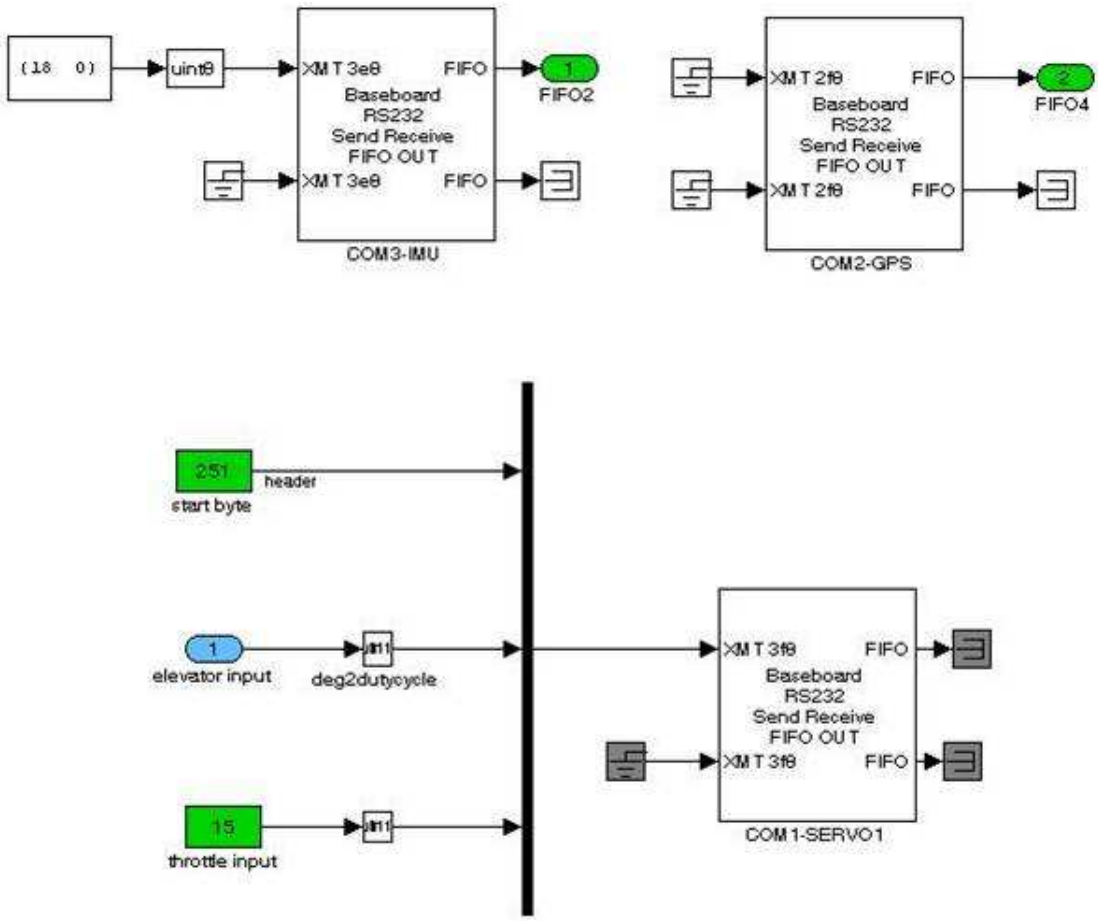


Figure B.1: Simulink Model for RS-232 Serial Communication

- Transmit and receive software FIFO size : 4096
- Transmit FIFO data type : 8 bit unsigned integer (uint) null terminated

AHRS sends data based on request. In this work, gyro-stabilized quaternions and instantaneous magnetic field, acceleration and angular rate vectors are returned after the hexadecimal command 0x12 sent to the AHRS. The received data is decomposed to obtain quaternion, accelerometer, angular rate and magnetic field measurements.

B.2 GPS Communication Interface

GPS receiver sends measurement data over RS-232 serial data link. It is connected to port COM2 of the flight control computer. Port settings are as follows;

- IRQ : 3
- Port address: 0x2f8
- Baud rate : 9600
- Parity : None
- Data bits : 8
- Stop bits : 1
- Hardware FIFO size : 16 deep
- Receive FIFO interrupt level : 1
- Auto RTS/CTS : on
- Transmit and receive software FIFO size : 4096
- Transmit FIFO data type : 8 bit unsigned integer (uint) null terminated

GPS receiver sends data using NMEA protocol. Data is sent continuously without request. Each data set with a specific header is called an 'NMEA sentence'. In this work, $GPGGA sentence is read. The sentence content is seen in Figure B.2.

GPS data given in Figure B.2 is received in an array. ASCII code of each character is contained in the array. Latitude (field 2 in Figure B.2), longitude (field 4), position fix (field 6), satellites used (field 7), mean-sea level altitude (field 9), geoid separation (field 11) and checksum (field 15) are the data used in the thesis work. A simple Matlab script is written to parse this data.

GPS altitude with respect to datum (reference ellipsoid defined by WGS84[25]) is calculated as follows:

$$h = H - N$$

where;
h : GPS altitude with respect to datum
H : altitude with respect to the geoid (gravitational equipotential surface= mean-sea-level)
N : geoid separation

Field No.	Example	Format	Name	Unit	Description
0	$GPGGA	string	$GPGGA	-	Message ID, GGA protocol header
1	092725.00	hhmmss.sss	hhmmss.ss	-	UTC Time, Current time
2	4717.11399	ddmm.mmmm	Latitude	-	Latitude, Degrees + minutes, see Format description
3	N	character	N	-	N/S Indicator, N=north or S=south
4	00833.91590	dddmm.mmmm	Longitude	-	Longitude, Degrees + minutes, see Format description
5	E	character	E	-	E/W indicator, E=east or W=west
6	1	digit	FS	-	Position Fix Status Indicator, See Table below and Position Fix Flags description
7	8	numeric	NoSV	-	Satellites Used, Range 0 to 12
8	1.01	numeric	HDOP	-	HDOP, Horizontal Dilution of Precision
9	499.6	numeric	msl	m	MSL Altitude
10	M	character	uMsl	-	Units, Meters (fixed field)
11	48.0	numeric	Altref	m	Geoid Separation
12	M	character	uSep	-	Units, Meters (fixed field)
13	-	numeric	DiffAge	s	Age of Differential Corrections, Blank (Null) fields when DGPS is not used
14	0	numeric	DiffStation	-	Diff. Reference Station ID
15	*5B	hexadecimal	cs	-	Checksum
16	-	character	<CR><LF>	-	Carriage Return and Line Feed

Figure B.2: $GPGGA Sentence Data Content

B.3 Servo Driver Board Interface

Servo driver board receives servo commands from the flight control computer over RS-232 serial data link. It is connected to port COM1 of the flight control computer. Port settings are as follows;

- IRQ : 4
- Port address: 0x3f8
- Baud rate : 19200
- Parity : None
- Data bits : 8
- Stop bits : 1
- Hardware FIFO size : 16 deep
- Receive FIFO interrupt level : 1
- Auto RTS/CTS : on

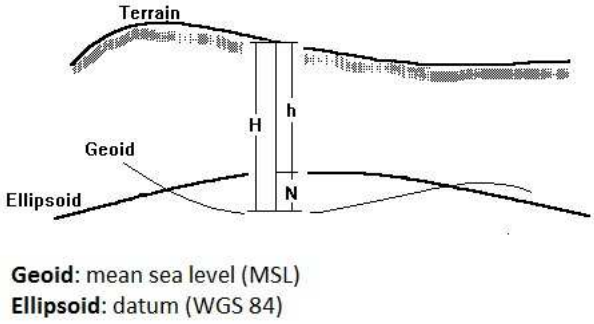


Figure B.3: Geoid Separation Illustration

- Transmit and receive software FIFO size : 4096
- Transmit FIFO data type : 8 bit unsigned integer (uint) null terminated

The interface is written to drive two servo motors at the same time. The data is sent to the servo driver board with a specific header. Elevator command (in 'deg') and throttle command (in %) is converted to duty cycle and sent to the servo driver board following the header. Based on the duty cycle commands, two PWM signals are generated by the servo driver board and sent to elevator and throttle servos.

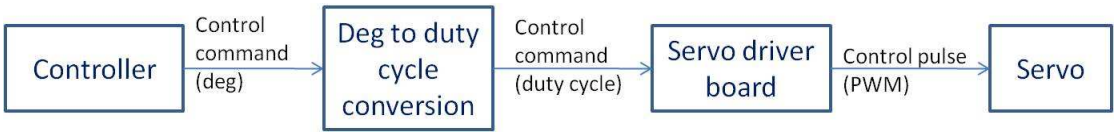


Figure B.4: Servo Command Generation

B.4 Analog Interface

Pressure sensor and potentiometers use analog input port of the flight control computer. Analog input driver specifically written for Athena II SBC ('Athena/Prometheus Analog Input') in xPC Target library is used for I/O hardware communication. Analog input settings are adjusted from the block parameters dialog box (Figure B.5).

Analog input parameters are defined as follows:

- First channel: Denotes the number of the first channel in a set of contiguous channels.

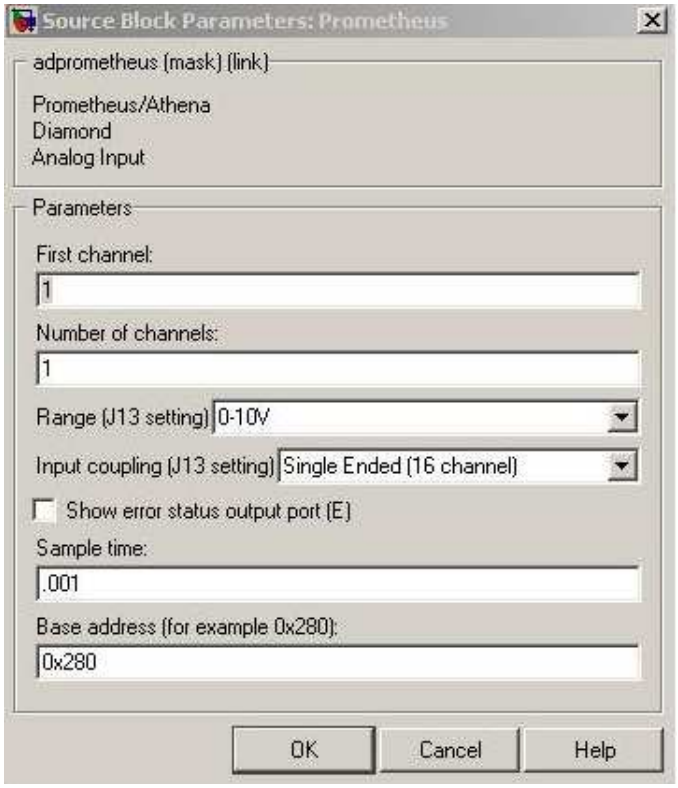


Figure B.5: Analog Input Block Settings

- Number of channels: Denotes the number of input channels to be used. For example, in order to read analog inputs from channels 4,5 and 6, 'first channel' is set to '4' and 'number of channels' is set to '3'.

- Range: Denotes input voltage range. For single-ended inputs, 0-10V, 0-5V and 0-2.5V options are available. For differential inputs, ±10V, ±5V, ±2.5V and ±1.25V options are available. The input range applies to all channels.

- Input coupling: Two options are available; single-ended and differential. A single-ended input uses two wires: input and ground. The measured input voltage is the difference between these two wires. A differential input uses three wires: input(+), input(−) and ground. The measured input voltage is the difference between the (+) and (−) inputs. The advantage of differential input mode is the higher noise immunity. Since the noise affects (+) and (−) input wires equally, the noise is cancelled out in the measurement. The disadvantage of differential input, on the other hand, is that only half as many input channels are available compared to single-ended input since two input pins are required to produce a single differential output.

- Base address: Hexadecimal base address of the data acquisition board.

In the thesis work, the 0-10V input range is used which corresponds to the range of 0-65535 for a 16-bit unsigned integer. Therefore, the smallest input voltage change that can be detected (resolution) turns out to be 10V/65535 = 153μV.

An important property of A/D conversion which affects the real-time operation is the latency. Latency is defined as the time between initiating a request for data and the beginning of the actual data transfer. Several I/O board specifications are given in [9] for Diamond System products. All of the boards on the list have an A/D latency approximation of 6+10N (μs), where N denotes the number of analog input channels. Although the exact latency varies depending on the target PC motherboard and processor, xPC Target kernel version, I/O board hardware revision, and BIOS settings, this approximation gives a rough idea. If one analog input channel is used, the latency can be approximated as 16 μs. If five channels are used (1 for the barometer, 4 for the potentiometers), the latency becomes 56 μs. The importance of this parameter is due to its role on the real-time performance of the target computer (i.e. flight control computer). The latency is added to the sampling time that can be achieved without I/O hardware. If the target application requires more CPU time than the sampling time of the model, 'CPU overload' error occurs and the current run is stopped. This problem is experienced in several flight tests. Solution suggestion is using a separate or low latency I/O board.

APPENDIX C

Graphical User Interface: xPC Target Explorer

xPC Target Explorer is a graphical user interface for xPC Target. It runs on the host computer. Host PC and target PC configuration and communication settings are adjusted through xPC Target Explorer. Figure C.1 shows the snapshot of xPC Target Explorer window.

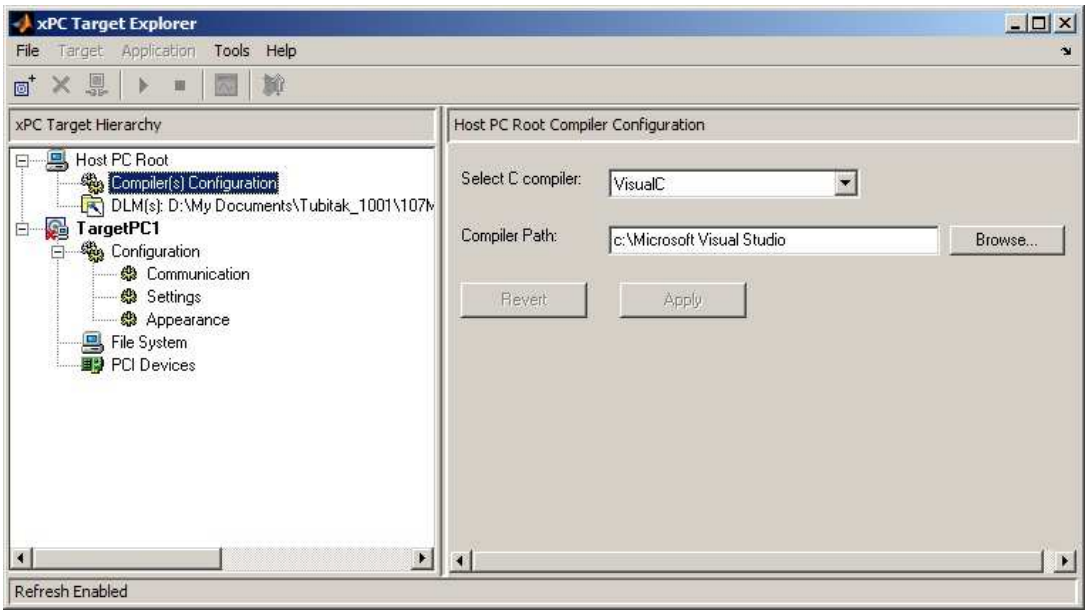


Figure C.1: xPC Target Explorer Window (Ver3.3)

On the left hand pane, host PC and target PC nodes are present. Configuration settings are adjusted through the windows under these main nodes.

- Host PC Configuration

  Compiler and compiler path selection are performed under host PC node. Microsoft® Visual C++® 2005 compiler is used in the study. Other supported compilers for Matlab Release 2007b are given in [17]. Compiler selection is performed on Matlab command

window using command 'mex -setup'.

- Target PC Configuration

  Target PC - host PC communication settings are given below;

  - Communication protocol: TCP/IP. This communication protocol is preferred to RS-232 due to higher rate and longer-distance data transmission properties of the TCP/IP communication protocol.
  - Target PC IP address: 192.168.127.251
  - TCP/IP target port: 22222
  - LAN subnet mask address: 255.255.255.0
  - TCP/IP gateway address: 192.168.127.254
  - TCP/IP target driver: NS83815. Denotes the ethernet chip family. NS83815 is the target driver for Argon 10/100MB National Semiconductor DP83815 PCI Fast Ethernet Adapter.
  - TCP/IP target bus: PCI

  Host PC TCP/IP settings are also adjusted to establish a TCP/IP network between the two computers. Host PC IP address is set as 192.168.127.5, subnet mask and gateway addresses are set as 255.255.255.0 and 192.168.127.254, respectively.

APPENDIX D

Data Logging

The sensor measurements and Kalman filter estimates are recorded on the PC/104 SBC flash disk during flight. The block used for data logging is xPC target 'File Scope'. In the sample model given in Figure D.1, Euler angle measurements are read from the AHRS and logged to the file system of the target PC using File Scope.

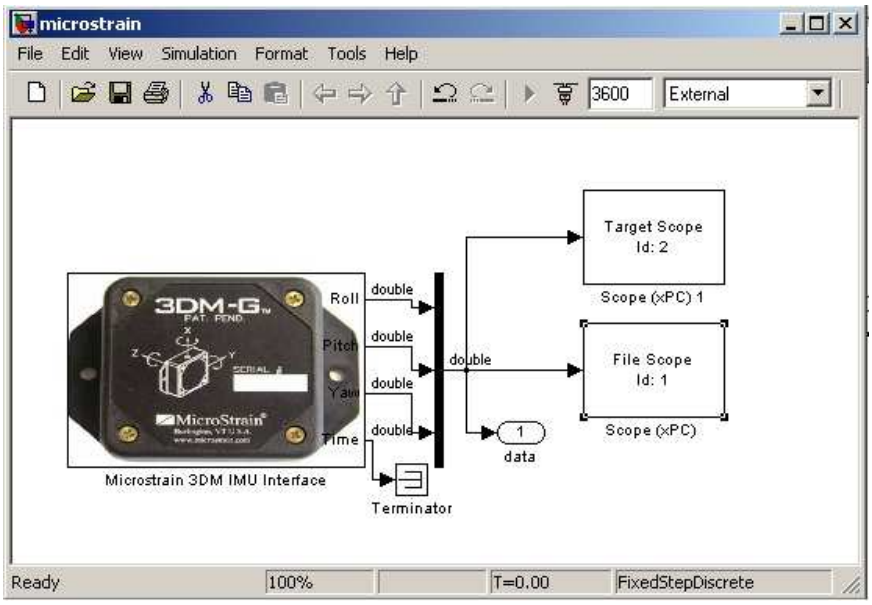


Figure D.1: Data Logging Using File Scope

The most significant parameter of the File Scope is the 'number of samples' (Figure D.2). Data is collected at every simulation second. For example, in the above model, the simulation time step is 0.01 s (100 Hz). The simulation is run for an hour (3600s). Therefore a total of 360001 samples are collected. The measurement data is written to a file named 'data.dat' (Figure D.3). After the simulation is over, the data.dat file is uploaded to the host PC. Target file system is seen through xPC Target Explorer window (Figure D.3).

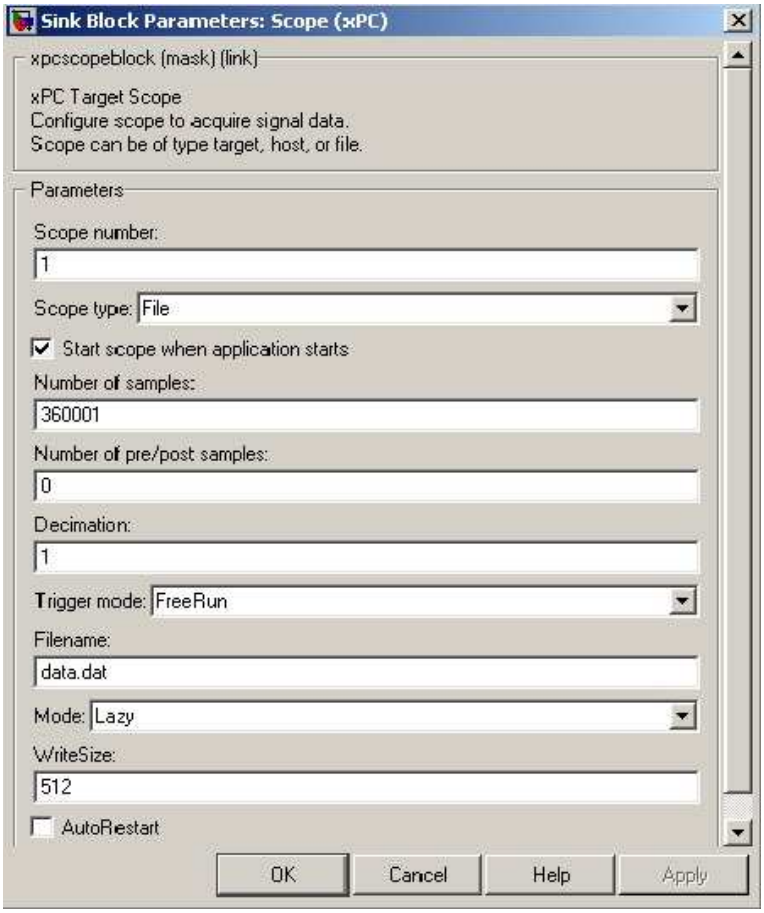


Figure D.2: File Scope Parameters

Data.dat file contains the data logged during 60 min flight and data5.dat file contains the data logged during 5 min flight. It is seen that the file size of data.dat is twelve times the size of data5.dat.

In this example only three flight parameters (Euler angles) are recorded. For an hour long flight, 11520.5 KB = 11.25 MB memory space is used. Therefore, for a 15 min flight, a quarter of this much space (2880.125KB=2.813MB) is going to be used. Similarly, we can deduce that 12 flight parameters (four times the currently recorded parameters) can be stored in 11.25 MB memory space for a 15 min flight.

The parameters to be recorded during flight are seen in Table D.1. Approximately (11.25/12)x33 $\approx$ 31 MB memory space is needed to store 33 states during a 15 min flight (provided that the sampling rate is 100 Hz). If the sampling rate is doubled, twice the memory space is needed to record the parameters of a 15 min flight. Similarly, if the flight time is doubled, the memory requirement is going to be doubled as well.

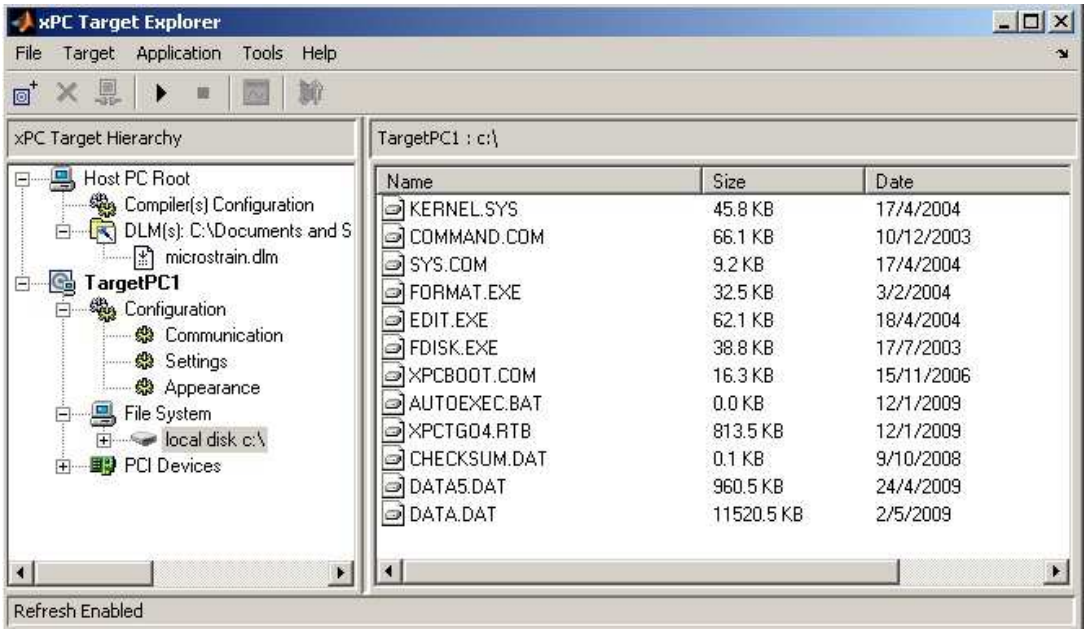


Figure D.3: Target File System on xPC Target Explorer

Table D.1: Recorded States

#	Symbol	Name & Description	Data Type
AHRS Measurements			
1	ϕ	Roll angle	double
2	θ	Pitch angle	double
3	ψ	Yaw angle	double
4	p	Roll rate	double
5	q	Pitch rate	double
6	r	Yaw rate	double
7	a_x	Body x-acceleration	double
8	a_y	Body y-acceleration	double
9	a_z	Body z-acceleration	double
GPS Measurements			
10	GPS_on	GPS on/off signal 1: GPS on 0: GPS off	binary

Continued on Next Page...

Table D.1 – Continued

#	Symbol	Name & Description	Data Type
11	Checksum	GPS checksum flag 1: no data loss 0: data loss in RS-232 data link	binary
12	lat	Latitude	double
13	long	Longitude	double
14	h	Altitude	double
15	V_N	North velocity	double
16	V_E	East velocity	double
17	V_D	Down velocity	double
Barometer Measurements			
18	h_{baro}	Calibrated pressure altitude	double
19	h_{baro_raw}	Raw barometer altitude reading	double
20	Baro_on	Barometer on/off signal 1: Barometer on 0: Barometer off	binary
Kalman filter estimates			
21	$\phi_{\_}$	Estimated roll angle	double
22	$\theta_{\_}$	Estimated pitch angle	double
23	$\psi_{\_}$	Estimated yaw angle	double
24	$p_{\_}$	Estimated roll rate	double
25	$q_{\_}$	Estimated pitch rate	double
26	$r_{\_}$	Estimated yaw rate	double
27	$lat_{\_}$	Latitude	double
28	$long_{\_}$	Longitude	double
29	$h_{\_}$	Altitude	double
30	$V_{N\_}$	North velocity	double
31	$V_{E\_}$	East velocity	double
32	$V_{D\_}$	Down velocity	double
Controls			

Continued on Next Page...

Table D.1 – Continued

#	Symbol	Name & Description	Data Type
33	δ_e	Elevator deflection angle	double