Association for Unmanned Vehicle Systems International

Unmanned Aerial System Competition

2007-2008 Design Report

Flagship Envy

University of California, Los Angeles

Abstract

A team of undergraduate students at the University of California, Los Angeles, has developed an autonomous airplane for entry into the 2007-2008 Unmanned Aerial System competition hosted by Autonomous Unmanned Vehicle Systems International. The contest requires that teams develop and demonstrate an aerial vehicle capable of autonomous flight and visual acquisition of specified ground targets. In meeting the competition objectives, the UCLA team designed the majority of the systems required: the aerial platform, video capture system, autopilot, as well as a ground station. This allowed team members to gain valuable multidisciplinary experience with the problems encountered in designing, integrating, and operating the various systems of an autonomous unmanned aerial vehicle.

An airplane was designed and manufactured to satisfy design guidelines on static and dynamic stability; concurrently, an autopilot system was developed. Control systems were designed using a model of the aircraft. A variety of sensors were also acquired to provide data that the autopilot needs to maintain flight stability and respond to navigation commands. A ground station was then created to allow an operator to view the state of and to direct the aircraft. Extensive testing was required to verify the expected performance of the autopilot and to prepare the fully integrated aerial system for completing the competition missions.

Table of Contents

1.0 Executive & Management Summary 3
  1.1 Project Philosophy 3
  1.2 Development Summary 4
  1.3 Team Architecture 4
2.0 System Overview 5
  2.1 Design Requirements 5
    2.1.1 Mission Requirements 5
    2.1.2 Payload Requirements 6
    2.1.3 Aircraft & Autopilot Specifications 6
  2.2 General Architecture 7
  2.3 Systems Engineering 8
3.0 Aerial Vehicle 8
  3.1 Design Objectives & Parameters 8
  3.2 Aircraft Development 9
    3.2.1 Analysis 9
    3.2.2 Design 10
    3.2.3 Construction 10
4.0 Autonomous Control and Navigation 11
  4.1 Control System Development 11
  4.2 Sensors and Flight Computer 12
    4.2.1 Sensor Package 13
    4.2.2 Sensor Data Fusion 14
    4.2.3 Electronics 14
    4.2.4 System Architecture 15
  4.3 Communication and Operator Input 16
    4.3.1 Command & Data Link 16
    4.3.2 Failsafe & Safety Pilot 17
    4.3.3 Ground Station Software 18
5.0 Video System 18
  5.1 Video Camera 18
  5.2 Wireless Link 19
6.0 Summary & Conclusions 19
7.0 References 20

List of Figures

Figure 2.1: Diagram of general system architecture 7
Figure 2.2: Example diagram of systems interdependency.. 8
Figure 3.2: CAD of the Flagship Envy before field modifications 10
Figure 3.1: The AVL model used for stability analysis 10
Figure 4.1: General control system block diagram in Simulink 12
Figure 4.2: The ADXL330 on the top and SM5812 on the bottom 14
Figure 4.3: Complete components connectivity diagram.. 16
Figure 4.4: Digi International XBee Pro with U.FL connector 17
Figure 4.5: A preliminary iteration of the ground control station... 18

List of Tables

Table 1.1: UCLA AUS team roster and tasks 5
Table 3.1: Various important plane parameters 9
Table 4.1: List of plane state sensors.. 13
Table 4.2: Plane states and the sensors from which they are derived.................................... 13

1.0 EXECUTIVE & MANAGEMENT SUMMARY

This report summarizes the development of UCLA's entry into the 2007-2008 Unmanned Aerial System (UAS) Competition hosted by the Association for Unmanned Vehicle Systems International (AUVSI). The UAS contest demands that student teams develop and demonstrate an aircraft capable of autonomous flight and visual acquisition of ground targets. UCLA's team for the UAS competition set out to develop all the systems relevant to achieving the primary mission of the contest. This includes the design of the aerial platform, as well as the systems for control, navigation, and operation. A project of such a wide scope requires significant planning and organization, as well as significant commitment from the members of the team.

1.1 Project Philosophy

The UCLA team for the UAS contest was formed to foster student interest in the technologies and methods involved in developing autonomous unmanned aerial vehicles (UAVs). In that light, it was decided that team members would derive the most benefit from the project by designing and implementing as many facets of the vehicle as possible. In particular, the team leadership decided to develop both an autopilot, including the sensor package and control system, and an aerial vehicle to work with it.

The team is comprised mostly of aerospace engineers, none of whom have had prior experience working with electronics or designing a complex control system. However, a familiarity with disciplines traditionally outside of the aerospace engineer's domain of study is becoming increasingly valuable as vehicle systems become more complex. For example, a working knowledge of the space requirements for electronics can greatly affect the design and layout of an aircraft fuselage.

UCLA's aerospace curriculum, in general, does not provide opportunities to accumulate such experience. Student projects, such as the UAS team, are typically left to provide it. The team believes that this competition, with the rules as provided this year, offers a unique opportunity to develop that multidisciplinary experience.

1.2 Development Summary

The UCLA team for the UAS contest was quickly formed in late December, shortly before team registrations were due. This forced the team into a rushed development schedule. Despite these setbacks in starting time, the team was able to complete the development of an aerial platform and make significant progress toward development and deployment of an autopilot.

An airplane was designed to be capable of remote as well as autonomous flight. Static and dynamic stability drove the design, leading to a fairly conventional airplane configuration. This more conservative aircraft helps reduce the burden on the autopilot, allowing the design of the control system to focus on autonomous waypoint navigation. The sensor package for the autopilot went through several proof of concepts as each component was integrated into the overall system. Ground station software was also developed to meet the needs of the competition and to allow for ease of use.

1.3 Team Architecture

UCLA's 2007-2008 team for the UAS competition is composed of ten students, all undergraduates, and most of them, aerospace engineering students. A core group of students formed the leadership of the team. All of them are veteran members of UCLA's teams for UAV Design/Build/Fly competition, hosted by the American Institute of Aeronautics and Astronautics. This experience was invaluable in the design of the airplane. Responsibilities were divided between team members to allow for parallel development of the major components. The team list and division of labor is shown in the table below.

Name	Year	Major	Task
Viet Nguyen	4	Aerospace Engineering	Project lead, electronics, software
Gerard Toribio	4	Aerospace Engineering	Controls
Jerry Huang	4	Aerospace Engineering	Aircraft design
Gaurav Bansal	2	Aerospace Engineering	Propulsion
Scott Larson	2	Aerospace Engineering	Manufacturing
Eric Huang	2	Aerospace Engineering	Manufacturing
Jeffrey Duh	3	Aerospace Engineering	Camera and video
Charles Jaikumar	3	Aerospace Engineering	Aerodynamics
Clarence Gan	3	Aerospace Engineering	Aerodynamics
Song Zheng	1	Electrical Engineering	Radio and data

Table 1.1: UCLA AUS team roster and tasks

2.0 SYSTEM OVERVIEW

2.1 Design Requirements

2.1.1 Mission Requirements

The UAS contest requires that an aircraft autonomously navigate a series of waypoints defined by GPS coordinates and altitudes. The list of waypoints will be provided at the competition and may be modified by contest judges during the mission attempt. After navigating those waypoints, the aircraft must enter a search pattern. In both cases, the team must spot and provide locations for ground targets. Mission performance will be scored primarily on the accuracy of the information provided about the ground targets and whether the vehicle was able to navigate the competition course.

The mission profile is framed in the context of an aircraft providing support to a United States Marine Corps unit in the field. The vehicle is to spot hostile targets in the field and provide accurate locations so that an air strike can be carried out with minimal collateral damage. Therefore, accuracy of the targets' identity and location is paramount.

2.1.2 Payload Requirements

The payload that the aircraft must carry is composed of two parts: the electronics necessary to run the autopilot and the components of a video capture and transmission system. The autopilot components are chiefly comprised of the various sensors, processing units, and other supporting circuits. The configuration will be driven by component layout and the data requirements of the control system.

The video system consists of cameras that must be capable of taking images at angles of up to 60 degrees in all directions from the vertical below the aircraft. It must also be able to transmit captured images to a ground station for viewing by the operators and contest judges. This suggests that a camera be mounted on a gimbaled platform to allow for those viewing angles and that cameras be linked to a radio system for wireless transmission of the data. It must also allow operators to spot targets at altitudes up to about 500 ft MSL.

Targets will be of various shapes, sizes, and colors. Target size will vary from about 2 to 8 feet wide with target thickness between 6 and 18 inches. The targets may also be mounted as high as 6 feet above the ground. An alphanumeric will be painted on the target in one color against a background of a different color. The color used will be one of seven: red, orange, yellow, green, blue, black and white.

2.1.3 Aircraft & Autopilot Specifications

The aircraft, of course, must be capable of carrying the payload. In addition, operation of the aircraft should allow for smooth image capture to allow for more accurate spotting of targets. Design of the aircraft should also allow for easy application of the autopilot; this suggests that a fair level of stability should be inherent to the aircraft. In addition, the aircraft should be designed with endurance in mind; the maximum time allowed for completion of the mission is 40 minutes. Preparation for the worst case scenario should take that into account. Safety requirements also limit the weight of the aircraft to 55 pounds.

The autopilot must be capable of accepting updated waypoint commands, both changes in heading and altitude, from the ground station. It must also be capable of accepting commands to change airspeed. Autonomous takeoff and landing are optional components of the contest; transition to manual control for those portions of the mission is allowed. Operation of the aircraft must allow for a safety pilot to override the autopilot and take control of the vehicle. In addition, the aircraft must be equipped with a contest-mandated failsafe configuration that brings the aircraft down quickly in the event of loss of control. The ground station must also be capable of displaying the current location of the aircraft relative to the designated no-fly zones.

2.2 General Architecture

When the problem was approached we kept simplicity in mind. There are a multitude of configurations that can achieve the same goal. We chose to have the on-board computer perform the necessary sensor conditioning as well as the flight control to literal actuation of control surfaces. No extra processing was left on-board for image processing or search pattern generation. All results of sensor conditioning and flight control are sent to the ground for data logging and time-stamping. Video is transmitted in parallel along a completely independent system to reduce any cross system dependency that may propagate failure.

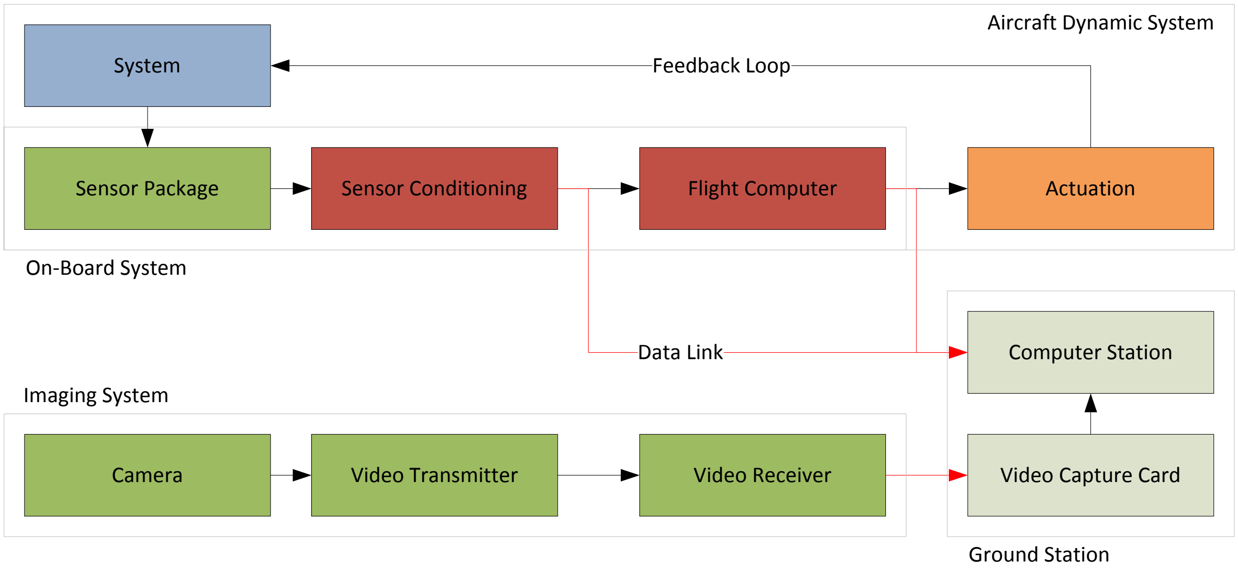


Figure 2.1: Diagram of general system architecture

2.3 Systems Engineering

The project is very much a systems engineering problem that requires careful management of subsystems. After an analysis of the mission requirements the project was divided into four main pillars of development: aircraft, controls, electronics, and software. Of particular difficulty to work with is the inter-dependency between systems. Veteran members of the team were tasked with specific pillars, and frequent meetings provided the means to resolve dependency problem.

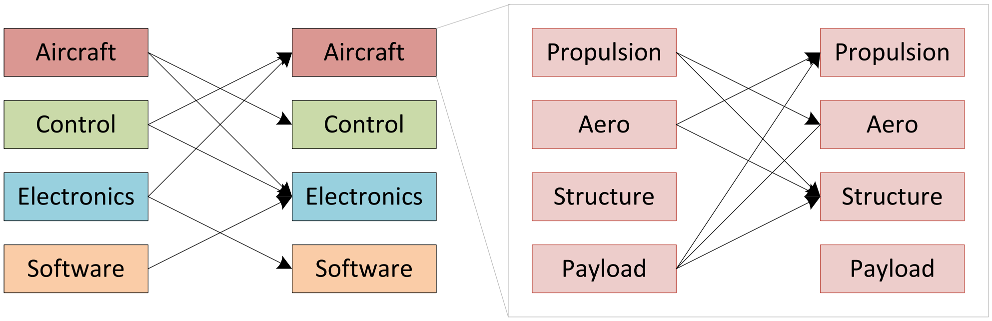


Figure 2.2: Example diagram of systems interdependency

3.0 AERIAL VEHICLE

3.1 Design Objectives & Parameters

The team designed an aerial vehicle to work with the autopilot and to carry the video system. The primary driving factors in the design were: a planned flight time of 40 minutes as well as the static and dynamic stability of the vehicle.

Research into performance of teams participating in the UAS contest in previous years suggested that the UCLA team could expect to have a flight time close to the maximum limit. This demands a propulsion system that can provide sufficient power for 40 minutes. An electric system powered by lithium polymer batteries is one option the team considered. However, it was decided to utilize a gas motor instead. The electric system would generate tremendous amounts of heat, which could cause some solder connections to fail after extended use, and raises concerns of electromagnetic interference from the rapidly changing magnetic field inside the motor. A gas motor, while potentially more hazardous due to the use of volatile fuels that require careful storage, could provide high performance for a longer period.

3.2 Aircraft Development

The competition aircraft is indigenously designed and built by the student club. This was not decided based on technical merits. Rather, it allowed for a medium to let upper-classmen apply more of their classroom knowledge and for lower-classmen to gain valuable hands-on experience.

3.2.1 Analysis

Preliminary analysis is performed using models and equations available in popular aircraft design and analysis books such as *Aircraft Design, a Conceptual Approach* by Daniel Raymer and *Aerodynamics, Aeronautics and Flight Mechanics* by Barnes McCormick. Prior experiences acquired from past AIAA Design/Build/Fly competitions were also taken into consideration. These preliminary analyses showed that the aircraft, from structural and flight dynamics points of view, needs to be reasonably large and sturdy to handle the amount of mechanical vibrations and the anticipated payload weight.

Detailed aerodynamics and stability analysis is performed using AVL, a vortex lattice code written by MIT professor Mark Drela. The software allowed for accurate calculations of many parameters, including static stability analysis, coefficient of lift, stability and control derivatives and root-locus plots of the fundamental dynamic modes. It also allowed for fast iterations between design changes for optimization purposes.

Parameter	Value	Parameter	Value
Wing span	7.64 ft (91.68 in)	Empty weight	~13 lbs
Wing root chord	1 ft (12 in)	Takeoff weight	~17 lbs
Length	4.3 ft (51.60 in)	Power plant	Zenoah G26ei
Height	~2 ft	Fuel capacity	50 fl. Oz

Table 3.1: Various important plane parameters

3.2.2 Design

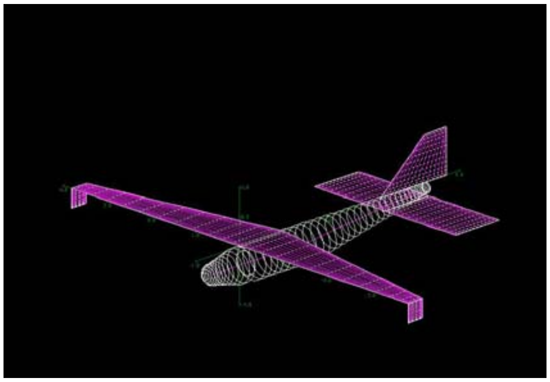


Figure 3.1: The AVL model used for stability analysis

The final vehicle is simple and straightforward. The wings are optimized for low speed flight; they feature a high aspect ratio of approximately 11 and a taper ratio of 0.35. Ailerons are located on the outboard sections. The empennage is given a long moment arm and is of conventional layout. The large horizontal tail is rectangular and is designed to be all moving. The vertical tail has a highly swept leading edge and features a relatively large rudder.

Several less common aerodynamic features were implemented for stability purposes. In particular, the trailing edge forward sweep seen on the wings as well as the downward winglets were implemented to correct for a diverging Dutch-roll mode at low flight speeds. The horizontal tail, believed to be excessively large by some, was necessary to correct for diverging phugoid mode at certain center-of-gravity locations encountered at partial fuel.

After aerodynamic design was complete, structural design commenced in parallel with the actual plane CAD. SolidWorks 2007-2008 Student Edition was used to CAD the plane.

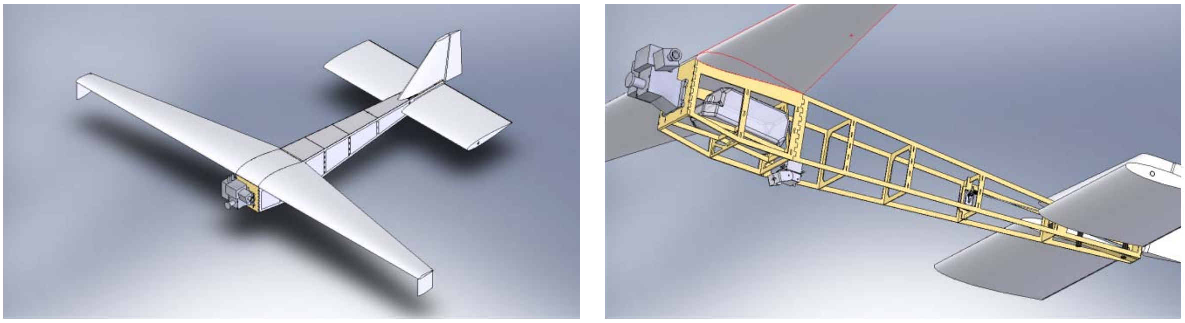


Figure 3.2: CAD of the Flagship Envy before field modifications

3.2.3 Construction

Construction featured a mixture of wood and composites. The fuselage featured plywood for the frame and balsa skin. All lifting surfaces featured a foam core and fiberglass coating. The spars utilized carbon fiber tubes. Several modifications were made during the manufacturing phase after consulting with our test pilot Rip Rippey. A last minute change in spare placement resulted in the wings having a slight anhedral, making the winglets optional for stability purposes. The tail was also changed from all moving to a conventional elevator design as the entire surface was too large for servos to handle.

4.0 AUTONOMOUS CONTROL AND NAVIGATION

As the focus of the AUVSI competition is to produce an aircraft capable of autonomous flight, considerable attention was paid to the development of the sensors and control system comprising the autopilot. In light of the philosophy of the UCLA team's ground-up design approach, it was also decided to develop the control system rather than purchase a commercially available, ready-to-use system.

4.1 Control System Development

The aircraft was designed to be fairly stable, statically and dynamically, to lighten the burden on the autopilot. Thus, the primary focus of control design was to enable the airplane to accept and follow navigation commands: chiefly, to meet altitude, heading, or speed commands. The control system must also maintain flight stability in achieving those commands.

Control design was accomplished with proportion-integration-derivative (PID) controllers implemented using multiple loop closure. Simple PID loops are used with major flight parameters, such as bank angle and altitude, to achieve and maintain stable flight. In designing the control loops, the team referenced a set of Massachusetts Institute of Technology course notes available for download on the Internet (MIT OCW 16.333).

The control system was designed in MATLAB Simulink; modeling of the aircraft dynamics was achieved using the AeroSim Blockset (Unmanned Dynamics AeroSim Blockset), a third-party block library enabling six degree-of-freedom simulation of airplanes. The AeroSim airplane model is configured for an individual aircraft using its basic geometrical and mass properties, its stability derivatives, as well as the characteristics of the propeller and engine.

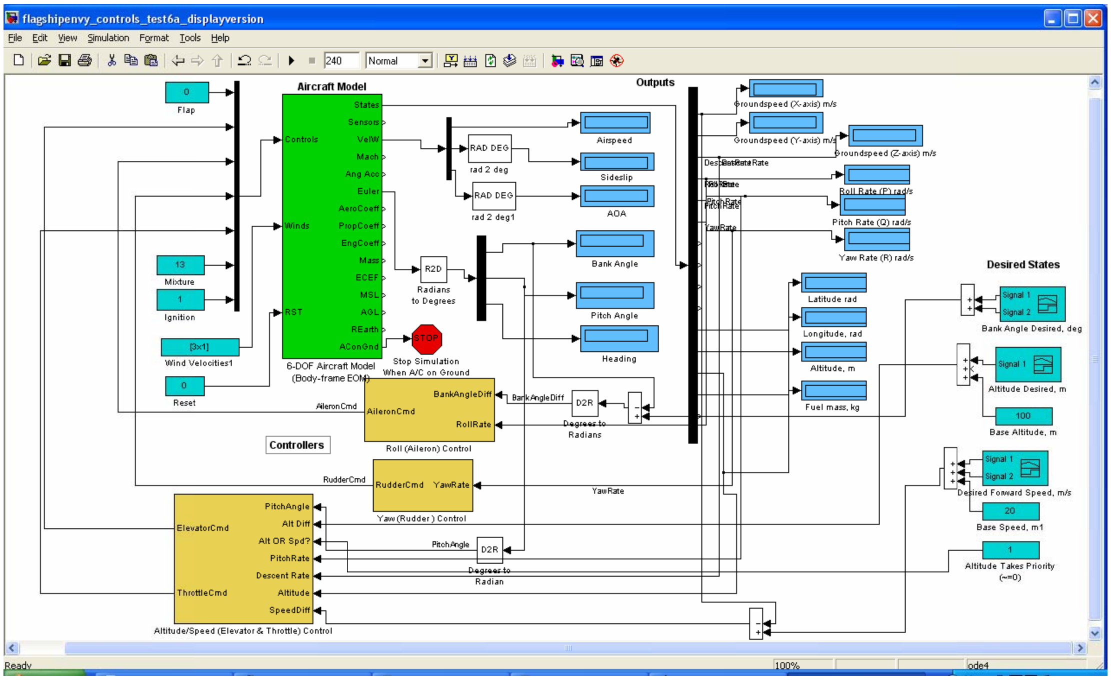


Figure 4.1: General control system block diagram in Simulink

At the most basic level, the control system is composed of three autopilot holds: heading, altitude, and speed. PID gains for those autopilots were set and modified to obtain stable and desirable flight responses to various inputs; saturations were also set for the throttle and control deflections to provide more realistic limitations on the aircraft's response to commands. The main drivers for setting the gains were to track complex commands fairly well and to avoid needing excessively large actuator responses, such as elevator deflections of 60 degrees. Control logic was also developed to mediate potentially conflicting commands, such as throttle and elevator commands to hold both speed and altitude. Sample responses are included in the appendix. Extensive testing, both in the lab and in the field, will be required, in addition to these software-only simulations, to properly tune the gains for the aircraft and the hardware and sensors specific to it.

4.2 Sensors and Flight Computer

The data required by the autopilot must be provided by a set of sensors onboard the airplane. A comprehensive sensor package including sensors such as gyros, accelerometers, and a magnetometer will provide the control system with the data required to navigate and stabilize the aircraft. The aircraft is also equipped with a GPS receiver to provide the navigation system with precise location data. Sensors were chosen with careful regard to the autopilot's needs and the requirements for the competition.

4.2.1 Sensor Package

In order for the auto-pilot to perform properly it must know its own system state at all times. The system state includes position (local to ENU and global to GPS), velocity (ground and air speed), and orientation. A variety of sensors is necessary to develop all of the information needed to determine all states. These sensors and their respective uses are shown in Table 2 below.

Sensor	Description	Purpose	Quantity
ADXRS300	Single-axis rate gyro	Orientation, inertial measurement (rotational rates)	3
ADXL330	Triple-axis accelerometer	Translational, inertial measurement (accelerations)	1
MicroMag3	Triple-axis magnetometer	Orientation, relative to local magnetic field	1
MPX4115A	Pressure transducer	Airspeed, altitude, in conjunction with pitot-static tube	2
EM-406A	GPS module	Translational, absolute position	1

Table 4.1: List of plane state sensors

The combination of sensors provides at least one data source for each pertinent plane state while also overlapping in responsibility and providing redundant data for more accurate data. Table 4.2 lists the plane states and their respective handlers with inertial measurement unit (IMU) representing the combination of three single-axis rate gyros and one triple-axis accelerometer.

State	Sensors	Redundancy
Orientation	Inertial measurement unit, triple-axis magnetometer	3
Heading	Magnetometer, GPS, inertial measurement unit	3
Altitude	Pitot-static system, inertial measurement unit, GPS	2
Ground speed	GPS, inertial measurement unit	2
Air speed	Pitot-static system	1

Table 4.2: Plane states and the sensors from which they are derived

4.2.2 Sensor Data Fusion


Figure 4.2: The ADXL330 at the top and MPX4115A at the bottom

When dealing with redundant data the problem of how to combine that data arises. The most popular method is the use of a Kalman filter. However, Kalman filters are both computationally expensive (particularly for 8-bit microcontrollers that must emulate floating point precision) and difficult to create and tune. In light of these difficulties, it was decided that a simplified process similar to a Kalman filter would be implemented.

Kalman filters rely on the Kalman gain to develop the optimal weighted average combination of redundant data and system state estimate. Utilizing the same idea, our data fusion routine also uses a weighted average to combine redundant data with a system state estimate. The system state estimate is developed using a simple time-step integration equivalent to a system difference matrix. What differs between our approach and a Kalman filter is how we develop the gains. Instead of computing a Kalman gain which requires intensive matrix inversion, particularly when the system can be as large as 17 states, our data fusion routine will crudely estimate the error of each state and each redundant measurement. It will then use the fraction of the data error out of the total measurement error as the gain with which to weight average the new estimate.

Our data fusion method is less accurate and involves some more tuning, but the implementation is both simpler and hardware friendly. The system is still capable of updating at 10 Hz. In addition to the actual data fusion process software filters are run on incoming data to emulate filters such as low-pass filters to reduce noise. To test the process the recursive algorithm is written and tested on a dataset within MATLAB first before implementation into microcontroller code.

4.2.3 Electronics

All processing is done on multiple chained Atmel ATMega168 8-bit microcontrollers. The reason for this is they are extremely easy to program for using the Arduino tool chain. That tool chain utilizes the avr-gcc compiler allowing us to develop code in the relatively simple and extremely robust C++ language. In our setup the ATMega168 provides roughly 14 kB for program space, runs at 16 MHz, and has thirteen general digital I/O pins and six 10-bit resolution analog to digital converters (ADC) (Arduino). Six of these microcontrollers are used, two for control and four for data processing. Of the two chained in control one is dedicated to sensor fusion and the other to the auto-pilot. The four used for data processing act as data buffers for a manual clear-to-send (CTS) multiplexing (MUX) scheme due to a single universal asynchronous receiver/transmitter (UART) limitation on the ATMega168.

Each sensor will be “black boxed” by soldering them onto their own perf-boards and connecting the boards together into a main board. This allows for expansion in the future should the need to upgrade the sensors come up. For example, the inertial measurement unit will contain its own CTS handling microcontroller interface. A new board that mimics the same interface can be created and integrated with the main board easily should it be necessary.

All of the electronics as well as the servos are powered by a single ThunderPower three-cell 4450 mAh capacity lithium polymer battery from the High Performance product line. The voltage rating is high enough to power both the camera and video transmitters which require 12 volts to operate. A battery elimination circuit (BEC) is used to provide 6 volts to the servos and electronics. A majority of the electronics run at 5 volts which is provided by the BEC in conjunction with a voltage regulator. Some electronics require 3.3 volts, also to be provided by a voltage regulator. The lithium polymer battery is charged with a lithium polymer specific charger per safety precautions. It is never left unattended unless in safe storage disconnected from all equipment.

4.2.4 System Architecture

Components alone do not comprise an autonomous system alone. The architecture of which they are integrated allows each component to play its part in providing the overall autonomous flight functionality. Communication between systems must be handled carefully to ensure timing and avoid data collision. Since the primary flight computers utilize a microcontroller with only one available UART for serial communication, a manual CTS system has to be implemented in multiple system interfaces using general digital I/O pins and MUX integrated circuits (IC). Figure 4 shows a complete components connectivity and data flow.

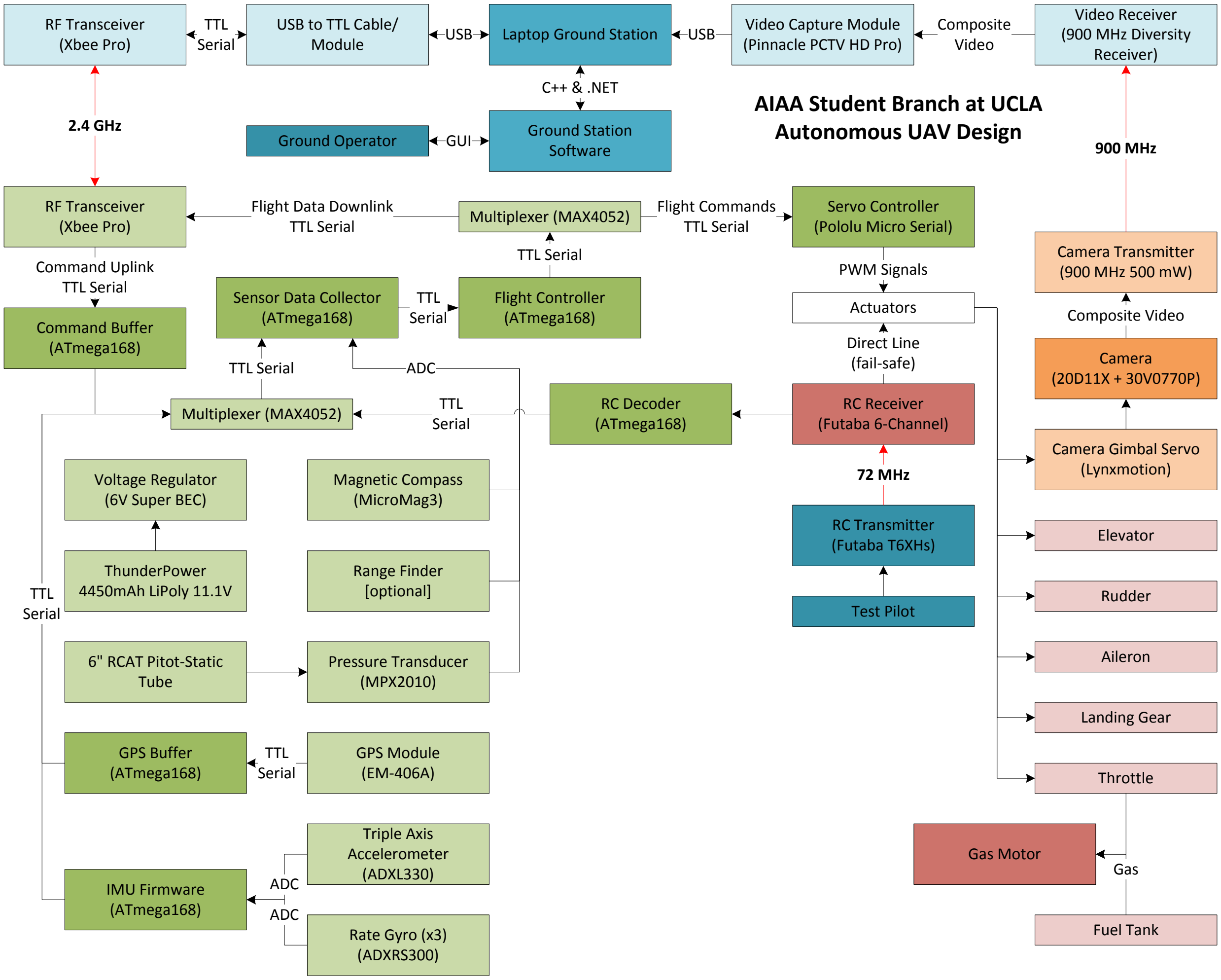


Figure 4.3: Complete components connectivity diagram

4.3 Communication and Operator Input

In normal operation, the aircraft accepts and follows navigation commands, based on GPS coordinates, from the ground station. The aircraft will fly from one GPS waypoint to another or, if no new navigation commands are available, maintain a holding pattern. In general, the aircraft will fly the shortest distance to the next waypoint. An aircraft operator, working at the ground station, will also be able to upload new controller gains and GPS waypoint commands to the system as required.

4.3.1 Command & Data Link

For data transfer between the plane and ground we utilize the XBee Pro transceiver from Digi International in conjunction with a 2.4 GHz 5 dBi duck antenna. The setup provides a range of roughly over one mile. XBee Pro modules allow for transparent operation in which it replaces the “wire” between two communicating UARTs at TTL level serial and automatically transfers the data (XBee Pro Datasheet). All administration of headers, packets, and checksums are taken care of internally. These modules operate on the 802.11 band at 2.4 GHz.

The interface between the air and ground is based on simple ASCII based commands following the format “$[command name],[parameter 1],[parameter 2],…” which are similar to NMEA GPS data strings. Each side will have its own command dictionary to provide parsing rules. Commands sent from the ground and data coming back from the air will share the same format. Although much of the data from the air will follow a similar parsing scheme, the parameters will be converted to binary format instead of ASCII representation to allow for higher data rates.

Figure 4.4: Digi International XBee Pro with U.FL connector

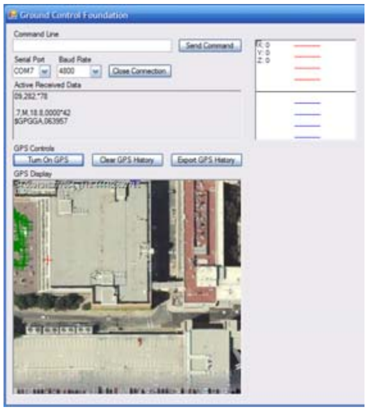
4.3.2 Failsafe & Safety Pilot

The ability to switch between manual remote control to auto-pilot control is essential, regardless of the ability to autonomously take off and land. Our implementation involves two layers. The first is a remote controlled single pole single throw relay. However, pending testing results, if electromagnetic interference or transient signals cause the relay to falsely activate the relay a mechanical system involving a servo and switch will be implemented. When the auto-pilot is activated a voltage is sent to a multiplexer which reroutes the source of RC signals to the servos. For the second control switch layer the control can be changed from the ground station software or on-board computer. This also involves generating a voltage with the on-board microcontroller to activate the multiplexer. The multiplexer uses a resistor to pull the addressing signal towards manual control in case of failure. Should the ability to switch control ever be lost a mechanism to mechanically remove the multiplexer from the circuit and switch to complete remote control can be implemented.

4.3.3 Ground Station Software

All communication between the ground station and aircraft except for video is handled by the ground station software. The program is written from the ground up in managed C++ using the .NET Framework. Communication to hardware is achieved with a USB to TTL level serial converter that utilizes the FT232RL chip and corresponding Windows XP driver to allow direct communication with the Serial Port object in .NET.

A simple Windows Forms based graphical user interface is created that shows the flight area, virtual flight instruments, command line and graphical interface to the plane, and various other data instruments. Efforts to include the video feed directly into the program are being attempted currently with hopes of having a single, unified, in-house program.



Serial Port: COM7
Baud Rate: 4800
X: 0
Y: 0
Z: 0

Active Received Data

```
09,282,*78

,7,M,18.8,0000*42
$GPGGA,063957
```

Figure 4.5: A preliminary iteration of the ground control station

By developing the ground station software completely from scratch, the possibility for expansion is limitless. As future upgrades are made to the platform and system the software can always be augmented to link with those changes.

5.0 VIDEO SYSTEM

5.1 Video Camera

After analysis of the necessary on-screen pixel resolution for target identification based on worst-case scenarios our camera was chosen. The worst case scenario chosen was the 200 ft altitude and 250 ft off-center target to be identified. We chose a camera after identifying a screen size of at least 32 pixels for strong identification of target. The camera and lens was selected based on the necessary focal length and camera resolution. Our camera is the Blue Sky Series 1/3" CCD Color Board Camera (20D118) from Videology Incorporated. This camera provides simple composite video output, automatic iris control, and a CS mount for lenses (20D11X Datasheet). The CS mount was of particular important as to ensure robustness for any possible lens changes that would be needed.

Lens selection was based on the same optical analysis. A varifocal lens was desirable to account for the two situations intended for the camera’s use: target search and target identification. Target search demanded a large field of view to first spot targets. When the target is spotted the lens would zoom in to identify the target characteristics. For our purposes we selected the 30V0770P, also from Videology Incorporated. This lens has a focal length ranging from 7.0 mm to 70.0 mm. On a 1/3” CCD those focal lengths correspond to a field of view of 5.2° and 50.7° respectively (C and CS Mount Camera Lenses).

5.2 Wireless Link

In our simple systems approach we wanted a reliable commercial video transmitter and receiver system that would provide an independent and parallel data link between the camera and the ground. We selected the 500 mW 900 MHz video transmitter and diversity receiver, both capable of 4 channels (910 MHz, 980 MHz, 1010 MHz, and 1040 MHz), from RangeVideo (RangeVideo 900 MHz Transmitter). Both units are simple systems capable of taking on more robust antennae if necessary. The transmitter takes video input from a video composite line; it can also take line-level audio.

6.0 SUMMARY & CONCLUSIONS

The UCLA team created for the UAS competition a new airframe, a low-cost autopilot system, as well as supporting software and hardware. These systems were developed to meet the mission and safety criteria. In doing so, team members built up valuable systems engineering experience, having been exposed to aspects of aircraft design and operation that usually are left untouched by the UCLA aerospace engineering curriculum. These systems, since they were custom-built, were also designed to be amenable to future upgrades. In particular, they were assembled using commonly available commercial components. This allows improvements in coming years to be made more readily. These improvements include more refined control systems and sensor data fusion methods, stabilized camera platforms, as well as an aircraft design that is more durable and can be more readily manufactured.

7.0 REFERENCES

20D11X Datasheet. <http://videologyinc.com/media/products/data%20sheet/20D10X/PDS-20D11X.pdf>.

Arduino. <http://www.arduino.cc/>.

AUVSI. "Competition Rules." AUS Student Competition (2008).

C and CS Mount Camera Lenses. <http://videologyinc.com/lenses/camera-c-cs-mount-lenses.htm>.

MIT OCW 16.333. <http://ocw.mit.edu/OcwWeb/Aeronautics-and-Astronautics/16-333Fall-2004/CourseHome/index.htm>.

RangeVideo 900 MHz Transmitter. <http://www.rangevideo.com/index.php?main_page=product_info&cPath=35_21&products_id=24&zenid=3b31ff0977bdabd0711895c6a2b4ba07>.

Unmanned Dynamics AeroSim Blockset. <http://u-dynamics.com/aerosim/default.htm>.

XBee Pro Datasheet. <http://www.digi.com/hottag.jsp?ht=/pdf/ds_xbeemultipointmodules.pdf>.