COGNITIVE FORMATION FLIGHT IN MULTI-UNMANNED AERIAL VEHICLE-BASED PERSONAL REMOTE SENSING SYSTEMS

by

Long Di

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. YangQuan Chen Major Professor	Dr. Doran Baker Committee Member
Dr. Donald Cripps Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2011

Abstract

Cognitive Formation Flight in Multi-Unmanned Aerial Vehicle-Based Personal Remote Sensing Systems

by

Long Di, Master of Science

Utah State University, 2011

Major Professor: Dr. YangQuan Chen
Department: Electrical and Computer Engineering

This work introduces a design and implementation of using multiple unmanned aerial vehicles (UAVs) to achieve cooperative formation flight based on the personal remote sensing platforms developed by the author and the colleagues in the Center for Self-Organizing and Intelligent Systems (CSOIS). The main research objective is to simulate the multiple UAV system, design a multi-agent controller to achieve simulated formation flight with formation reconfiguration and real-time controller tuning functions, implement the control system on actual UAV platforms and demonstrate the control strategy and various formation scenarios in practical flight tests. Research combines analysis on flight control stabilities, development of a low-cost UAV testbed, mission planning and trajectory tracking, multiple sensor fusion research for UAV attitude estimations, low-cost inertial measurement unit (IMU) evaluation studies, AggieAir remote sensing platform and fail-safe feature development, altitude controller design for vertical take-off and landing (VTOL) aircraft, and calibration and implementation of an air pressure sensor for wind profiling purposes on the developed multi-UAV platform. Definitions of the research topics and the plans are also addressed.

(157 pages)

To my father Xiaohong Di, mother Zexia Du, and my lovely big families who always support me and provide me the most fabulous life experience.

Acknowledgments

I would like to thank Dr. YangQuan Chen for providing me the opportunity to join the CSOIS OSAM UAV team during my junior year as an undergraduate research assistant, supporting me to work on different UAV projects and giving me full authority and trust to lead certain projects, encouraging me to always target research excellence and pursue outstanding research accomplishments. Whenever I had questions regarding my work or needed new research ideas, he always directed me with great answers and suggestions. Without his continuous support and instructions, my current achievement would have been impossible.

I would like to thank Dr. Haiyang Chao, who was my mentor during my undergraduate studies and early graduate research, for his motivation and guidance. He was always open to any discussions with me, and he has positively impacted me in different prospectives. His support during the 2009 SUAS competition provided me the initial confidence in doing UAV research. Without his advising and cooperation, I would have not been able to finish so many flight tests and make current achievements.

I want to express my sincere gratitude to my colleagues of the OSAM UAV team: Calvin Coopman for his assistance during the beginning of the low-cost IMU development and joyful discussions of all sorts of topics, and it was also quite a memorable experience to work with him during the 2010 SUAS competition; Austin Jensen for his suggestions and help during the Paparazzi project development; Jinlu Han for his assistance during the multi-UAV formation flight experiments; Yaojin Xue for his discussions and collaboration regarding cooperative control research; Tobias Fromm for his support on the sensor fusion studies and my first journal publication; Yiding Han for his help during several software developments and cowork for the 2009 SUAS competition; Dr. Ying Luo for his guidance regarding several control techniques; Dr. Yongcan Cao for his discussions on multi-agent control; Aaron Quitberg, Hu Sheng, Aaron Dennis, Jonathan Nielsen, Brandon Stark, and Daniel Morgan for their efforts and collaborative work on the UAV research.

I would like to thank my committee members, Dr. Baker and Dr. Cripps, for reading and revising my master's thesis, and I also appreciate their help on my personal statement and recommendation letters during my PhD applications.

I want to thank the Utah Water Research Lab and Dr. Mac McKee for the funding support; without this support, this research would have been impossible to accomplish. I would also like to thank the USTAR TCG grant for supporting the multi-UAV development.

Many appreciations to my parents for their decision to send me to Utah State University after my high school, and for their constant concern, faith, love, cultivation, toleration, and understanding so I can become a great person.

Last, but not least, I want to appreciate my girl friend (May) Wei Zou's support and love, so I can confront all the difficulties in both life and studies, and finally reach a stage of temporary success.

Long Di

Contents

List of Tables

List of Figures

Acronyms

ADC	analog-to-digital converter
AGL	above ground level
AMSL	above mean sea level
AP	autopilot
ARX	autoRegresive model with external
AUVSI	Association for Unmanned Vehicle Systems International
CG	central gravity
COTS	commercial off-the-shelf
CPS	cyber-physical systems
CSOIS	Center for Self-Organizing and Intelligent Systems
DCM	direction cosine matrix
DOF	degree of freedom
EPP	expanded polypropylene
FOC	fractional order control
FOPTD	first order plus time delay
FOV	field of view
FSM	finite state machine
GCS	ground control station
GPS	global position system
GUI	graphic user interface
IMU	inertial measurement unit
INS	inertial navigation system
IR	infrared
MEMS	microelectromechanical systems
MZNPI	modified Ziegler-Nichols PI
OSAM-UAV	open source autonomous multiple unmanned aerial vehicle

P2P	point to point
P2MP	point to multi-point
PPM	pulse-position modulation
RC	radio control
RGB	red-green-blue
RPC	remote procedure call
SISO	single-input and single-output
SPI	serial peripheral interface bus
SUAS	student unmanned aerial system
TCAS	traffic collision avoidance system
TWOG	tiny without GPS
UART	universal asynchronous receiver/transmitter
UAS	unmanned aircraft system
UAV	unmanned aerial vehicle
USB	universal serial bus
UTM	universal transverse mercator
UWRL	Utah Water Research Lab
VTOL	vertical take off and landing

Chapter 1

Introduction

1.1 Overview

Low-cost small and miniature unmanned aerial vehicles (UAVs) have attracted broad interest for their different uses in many areas [1–4]. UAVs have high potential to replace manned aircrafts in various military, civilian, and agricultural applications [5,6]. In military missions, UAVs can carry a variety of payloads, such as cameras, radars, and even weapons. UAVs can be used for reconnaissance in hostile environments and surveillance with long endurance without the need for an onboard human. Civilian applications include monitoring natural resources and management of the impacts of disasters. Besides their broad practical usages, small and micro UAVs are also valuable platforms for scientific research given their abilities. In recent years, with the development of compact onboard autopilot system, micro attitude estimation sensors, low-cost GPS and wireless communication devices, UAVs are able to perform autonomous flight and some basic trajectory tracking under low-level control algorithms [5]. These equipments guarantee and expand the capability of UAVs to accomplish different missions.

Cooperative control of multi-agent systems has attracted a lot of attention from researchers and developers [7–9]. In nature, multi-agent systems such as a school of fish, a flock of birds, a herd of goats, and even a groups of humans are very common. If the internal connections among all the agents can be established through some general protocol, all the agents can be driven to perform a particular function cooperatively. Cooperative control can reduce the demand of capabilities of one agent. On the other hand, it is usually operated in a distributed manner which can increase the redundancy and hence enhance the robustness of the whole system. It has been used to resolve problems which are difficult or impossible for an individual agent to solve, and they are widely applied in different areas,

such as network, industrial manufacturing, transportation, mobile technology, and security systems. Research focused on this area will help improve the efficiency, reduce the cost, increase the stability, and even maneuverability of the whole system.

Formation control is one approach to realize cooperative coordination [10]. Multi-UAV formation flight combines the research of both UAV and coordination, so it has gained significant attention from both unmanned system and control communities. Cooperative coordination is defined as requiring that a group of unmanned aerial vehicles to follow a predefined trajectory for flight missions while using their on-board sensors to acquire useful information while maintaining a specified formation pattern. The flight path can be a set of waypoints or a predefined fly zone with boundaries. Because formation flights of a UAV fleet can significantly increase the global security and universal efficiency of the entire system, it can benefit most of the applications which are handled by a single UAV. Therefore, multiple UAV formation control is the focus of this master thesis.

Cognition is defined as pertaining to the mental process of perception, memory, judgment, and reasoning [11]. Related to multi-UAV formation flight, it means every UAV agent is able to communicate with each other, exchange the flight data for rapid formation transition and response, improve its own performance by analyzing how others behave, determine the best flight path by optimizing the internal relative positions of all the UAVs. The details of the cognitive formation flight are presented in Chapter 2.

CSOIS AggieAir [12] personal remote sensing system that has been developed since 2008 and it has become a fully autonomous, low-cost, easy to utilize, and free of runway platform. It is able to carry different types of electronic devices, such as digital cameras, thermal cameras, fish tracking units, air pressure sensors, for image acquisition and wind profiling applications. My research regarding the AggieAir platform is primarily on the autopilot system, airframe design, sensor implementation and calibration, mission planning and flight data analysis, providing the foundations for the multi-UAV development.

The multiple UAV project [13] is based on open source Paparazzi software [14] and UAVs engineered by the CSOIS UAV team members. The Paparazzi autopilot was intro-

duced in summer 2007 and many improvements have been made on the original architecture since summer 2008 regarding airframes, navigation units, image systems, ground station software, etc. The current UAVs produced by CSOIS are capable of full autonomy. Numerous flight experience has been conducted based on the existing platforms resulting in the formulation a standardized flight test protocol to guarantee the success of each flight test. Therefore, the multi-UAV project brings new and exciting research challenges on the current system so we need to develop an appropriate testbed, improve the communication, implement the formation controller, perform-real time controller tuning, and resolve other issues.

1.2 Motivation

Considering if there is a wide piece of land and there are a variety of plants growing there, we want to monitor the growing condition of a certain plant but we have only one UAV carrying cameras. It will take more than ten flights to cover the whole area due to the endurance capability of one UAV and the size of the area. Then we can collect all the images of this land and make an explicit analysis of how that type of plants are growing based on the image data. There will be given certain number of images containing similar information because the flight path of each flight will sometimes overlap, which is not efficient for the whole process. If the single UAV system or the payload fails during the mission, it will be difficult to recover needed image data with a missing flight. As the number of flights increase, the chance of vehicle failure will correspondingly increase. Therefore, it is important to reduce the amount of flights and try to acquire the most information of interest within as fewer flights as possible.

If we can engineer a robust multiple agent control structure based upon the current AggieAir platform, these problems will be minimized. The mission time can be reduced significantly by flying more than two UAVs simultaneously, and the number of UAVs can also be adjusted dynamically based on the size of area to be mapped. During the flight mission, the formation shape can also be modified, such as a flying string, a flying triangle, or even flying a square traverse depending upon which types of images the users need. The

efficiency can also be improved since multiple UAVs can be regarded as one large ensemble so the flight path can be better optimized and overlapping of aerial images can be reduced. The overall system reliability can be enhanced since the flying agents can share information and monitor the status of one another, if the navigation system of one agent fails, it is still able to obtain the navigation information from other agents so the formation can still be maintained and image collecting task will not be interrupted. If one agent's camera system fails, the other UAVs can adjust the formation shape to cover the missing areas and still finish the task with minimum loss. This would avoid the previous flight.

There are several potential advantages of utilizing a multi-UAV system for the AggieAir applications and there are many practical issues preventing us from achieving an intelligent, stable, and robust UAV-based multi-agent formation control scenario. This thesis will address the UAV and control problems and present the results showing that such a scenario is close to be realized. The low-cost UAV testbed is the basis for research on UAV formation flight control. Besides, formation controller design and implementation, communication and formation reconfiguration issues, real-time controller tuning, is the focus of the research. Additionally, UAV platform development, attitude estimation, data fusion of multiple sensors, flight controller designs are also emphasized based on the current autopilot architecture.

1.3 Contribution and Organization

The major contributions documented in this thesis include the following perspectives:

(1) Low-cost UAV testbed development for cooperative UAV flight control research;

(2) Routinized formation flight of multiple miniature UAVs;

(3) Sensor fusion studies of several low-cost attitude estimation sensors;

(4) Boomtail conventional fixed-wing UAV platform development;

(5) Visual attitude estimation for miniature UAV;

(6) Consensus-based UAV formation control;

(7) AggieAir UAS platform development;

(8) Different flight controller design and validations.

This thesis is divided into four major chapters. Chapter 2 presents the concepts of multiple UAV-based personal remote sensing and cognitive formation flight. Several UAV platform developments involving the author's contributions are introduced in Chapter 3, which also includes the hardware and software architecture, major components, flight test protocols, and experimental results for each platform. Chapter 4 presents the studies on the attitude estimation for UAV navigation, which contain a two-stage calibration method using infrared sensors and a data fusion system for low-cost UAV attitude estimation using multiple inexpensive sensors. Afterwards, cooperative control of multiple UAVs is presented in Chapter 5, and it includes leader-follower experimental formation flight studies, which consists of the control structure, formation flight interface, controller tuning procedures, communication improvement, and a set of flight test results and performance analysis. Chapter 6 presents the studies on flight control system, which contains the speed control of fixed-wing UAVs and altitude control of a VTOL UAV. Chapter 7 gives conclusions which relate to the objectives and suggestions about follow-on research are drawn.

Chapter 2

Cognitive Personal Remote Sensing

2.1 Personal Remote Sensing Using UAVs

Personal remote sensing has become a popular application topic during recent years [15]. It basically means techniques based on instruments used in people's daily life are employed in the acquisition and measurement of spatially and temporarily organized data and information, so that these instruments can contain the property within the sensed scene which correspond to features, objects, and materials [15]. Some specific technologies have been adapted to improve personal remote sensing, such as electromagnetic radiation, acoustic energy sensed by lasers, radio frequency receivers, sound detectors, and so on. By applying more than one technique in the remote sensing process, the system accuracy can be increased, and the personal experience can also be enhanced, such as using multiple sensors to detect human body movements during exercise and indicate the best position and amplitude of each motion. Using UAV as a personal remote sensing platform, with multiple sensors and various payloads it can carry, people are able to obtain valuable information such as the growing condition of the plants, water contents in the river, construction condition of the highway, even air pollution and wind profiles. An example personal remote sensing application is illustrated in Figure 2.1. This payload integrates a regular red-green-blue (RGB) camera and a thermal camera so multi-spectral images can be obtained. After the imaging system is installed on one of our UAV remote sensing platforms, people can easily utilize the images to monitor their field, or perform search and rescue when a disaster happens.

Using UAV-based remote sensing platforms, solutions for various realistic problems can be achieved. Because of the advantages of a cooperative UAV system regarding operation range, safety, efficiency, and many other perspectives over isolated UAV systems, it is nec-

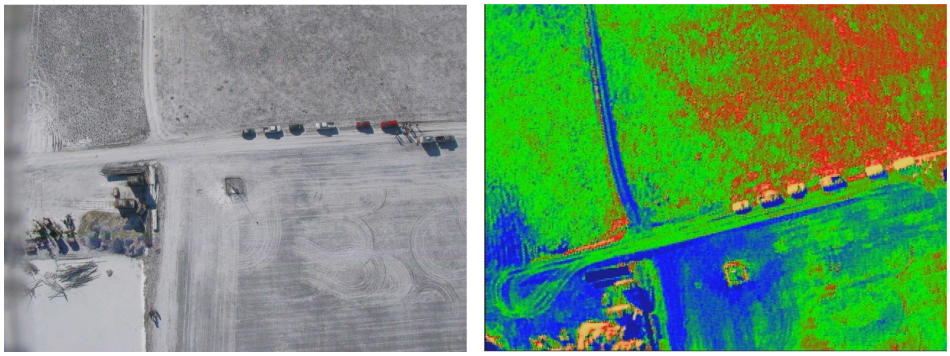


Fig. 2.1: RGB and thermal aerial images (Taken: 02/08/2011 Cache Junction, UT).

essary to explore the potential of multi-UAV-based personal remote sensing research. The following scenarios explicitly describe the ideas of using multi-UAV for various practical applications.

The first scenario is agricultural monitoring and irrigation management [16] (Figure 2.2). Given a piece of land, relying on a single UAV, the mission time and cost of the field survey can be enormous. If the single UAV suffers a system failure, the whole mission will be compromised. However, depending upon multiple UAVs, these problems are not that threatening any more. The user can send out a group of UAVs carrying cameras or other devices, and their virtual center can track the trajectory while all the UAVs are located with an equivalent distance between each other using a pre-defined flight plan. When all the UAVs are moving around the virtual center, their coverage can guarantee most areas of interest are captured into the images. Then the ground station can record the growing conditions of the all the plants and arrange irrigation based upon the aerial images. Even if any UAVs malfunction during the mission, other UAVs can compensate the loss and complete the mission, significantly improving the reliability and minimize the cost and time factors.

The second scenario is natural disaster amelioration (Figure 2.3 and Figure 2.4). Forest fire monitoring [17] and tornado surveillance [18] are two examples of using UAVs for in disaster management. When a forest fire occurs, depending on the areas it invades, numbers

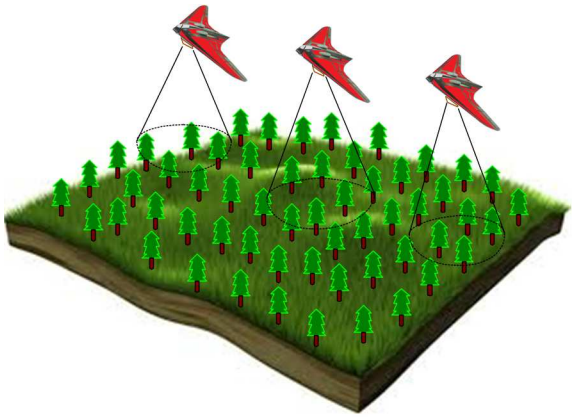


Fig. 2.2: Agriculture and irrigation monitoring.

of UAVs can be dispatched and guided in appropriate formations to fly over the endangered areas. Based on the image or video capturing devices and wireless transmission equipments on those UAVs, staff from the fire control department can acquire a complete view of the fire and send out fire fighters and manned aircraft to combat the wildfire effectively. When a tornado occurs, it is important to ascertain the movement of the tornado and predict which direction it will move. A group of UAVs flying a square traverse can be sent out with pressure sensors, and they can formulate a wind profile and deduce where the tornado is headed. The onboard video device can also report the information to the disaster control staff about the damage condition. The most important advantage of using a team of UAVs is the safety concern since both the forest fire and the tornado can also cause hazards to inspecting vehicles such as manned aircraft.

The third application of UAV formation flight is for the security purpose such as patrolling and surveillance around an important area or building [19] (Figure 2.5). If there is an area containing valuables or a building full of national secrets, in order protect them from thieves of terrorists, intensive security system with long hour monitoring needs to be established. If all the patrolling only relies on humans, we need to hire a lot of security guards and they have to rotate to continue the protections, which is very costly and there can be negligence. If there involves any violence, the consequence can be even more severe.

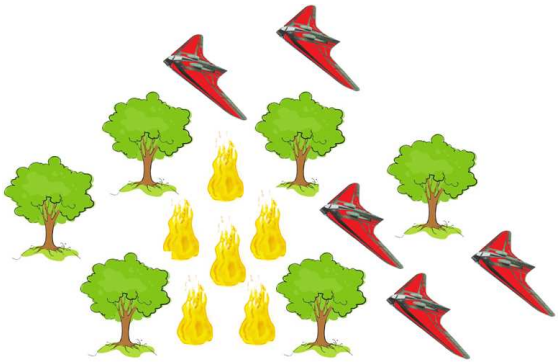


Fig. 2.3: Forest fire monitoring.

Fig. 2.4: Tornado surveillance.

However, with a fleet of UAVs, these problems can be resolved. Small or micro UAVs typically use electrical power, therefore, when they are cruising, there is little noise and they can stay in the air and remain patrolling for a much longer time than humans. Assigning different formation schemes and using video thermal cameras and target detecting software, they can cover an entire area no matter day or night without any blind spots, and perform image scanning on any suspicious objects. Once they find any suspects, they can lock their positions and directly report to the cognizant ground monitor station. Then the ground station can dispatch appropriate counter measures.

2.2 Cognitive Formation Flight

Since cognition means the capability of acquiring knowledge from external resource,

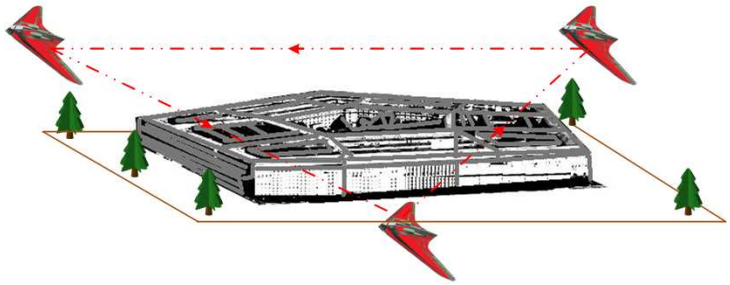


Fig. 2.5: Priority area patrol.

when applying to the multi-UAV formation flight, it means every agent in the cooperative UAV system is fully cognizant about any other agents and they are able to exchange the entire flight information including the navigation data and performance, actuator and sensor health, payload status, etc. Based on the feedback information, the multi-UAV system is able to perform self-diagnosis, self-compensation, and self-improvement. In this way, the formation flight can accomplish more missions with better safety and more resilience. An example of cognitive multi-UAV system framework is shown in Figure 2.6. In this framework, the cognitive architecture is based on the internal network among all the UAVs and an external datalink between the synergistic airborne system and a GCS. The cognitive multi-UAV system is able to understand mission objectives and always put safety as the first priority. The internal network is the first stage of the cognitive framework. Without any human intervention, all the agents perform autonomously and every UAV with its payload plays the roles of both sensors and actuators under this framework. Through the internal datalink, the flight information of any UAV is distributed to the other UAVs and each UAV can be guided based on the sensor feedback from the others. The external datalink is the second stage because the GCS involves the human operation and it has less authorization than the automation stage. The flight status from all the UAVs is reported back to the GCS and most of the time the GCS is used to monitor the airborne system and issue new flight mission commands through the external datalink. The human operation will take action

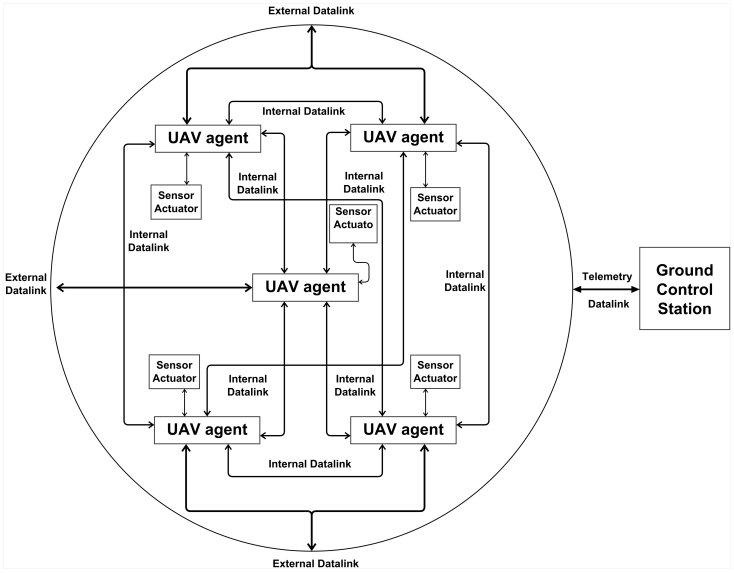


Fig. 2.6: Cognitive multi-UAV control framework.

only when the entire automation network fails considering some unanticipated situation occurrence.

The cognitive multi-UAV system can be modeled into three layers. The first layer is the trajectory tracking module, the second layer is the sensor network and the third layer is the formation reconfiguration module. Assuming there are n UAVs with i=1,2,3,$\cdots$,n, the estimation of the virtual center c that tracks the desired trajectory from the ith agent UAV. This is described as:

$$\dot{\hat{c}}_i \quad = \quad -\sum_{j=i}^{n} a_{ij}(\hat{c}_i - \hat{c}_j), \tag{2.1}$$

where $\hat{c}_i$ is the estimated center from the ith UAV, and

$$\begin{cases} a_{ij} > 0 & \text{if UAV j can receive info. from UAV i,} \\ a_{ij} = 0 & \text{if UAV j and i can not communicate.} \end{cases}$$

Then the desired position is calculated for each UAV using the following function:

$$\hat{s}_i = h(d, \hat{f}, \hat{c}_i), \tag{2.2}$$

where $\hat{s}_i \in \mathbb{R}^3$ and represents the desired 3D position for the ith UAV, d is the payload information provided by the sensor network, and $\hat{f}$ is the desired formation scheme generated by the formation reconfiguration module.

Afterwards, the local control input p_i can be generated and applied to an assumed UAV model with simple dynamics using the following equations:

$$p_i = g(b, \hat{s}_i), \tag{2.3}$$

$$\dot{s}_i = f(p_i, s_i), \tag{2.4}$$

where b is the sensor feedback provided by the sensor network and other complicated UAV models can be also utilized with specific control techniques such as PID or backstepping.

In the trajectory tracking control case (Figure 2.7), the virtual center c is formulated based on the $\hat{c}_i$ from all the UAV agents, then c will follow the desired flight path towards the final destination. The points of interest are located on the red line, and in the meantime, the desired formation scheme based on the feedback $\hat{f}$ is generated so the ith UAV moves to specified local position to follow the scheme and cover all the points dynamically. In the self-compensation case (Figure 2.8), while several UAVs are staying at the altitude h_1 and covering an area of interest, the payload feedback d detects some of the image information that is missing. Then the sensor network tells other agents one of the UAVs is malfunctioning, and a modified $\hat{f}$ is formulated so the remaining UAVs can cover the same area. While the rest of UAVs move following the new formation scheme in the horizontal dimension, the desired altitude also shifts from h_1 to h_2. When the environmental situation changes, for instance if the wind speed or direction varies, the data from all the air pressure sensors can be collected and included into b. A small wind profile then can be established for analysis. The cooperative UAV system can also use b to determine which direction

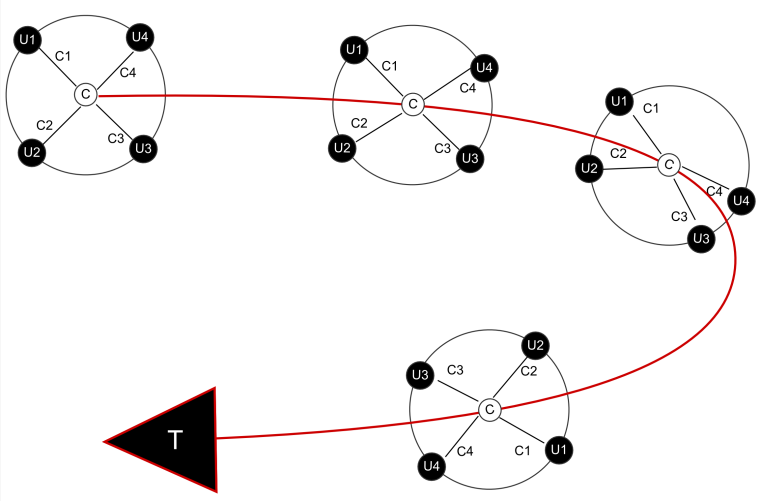


Fig. 2.7: Cognitive formation flight-trajectory tracking.

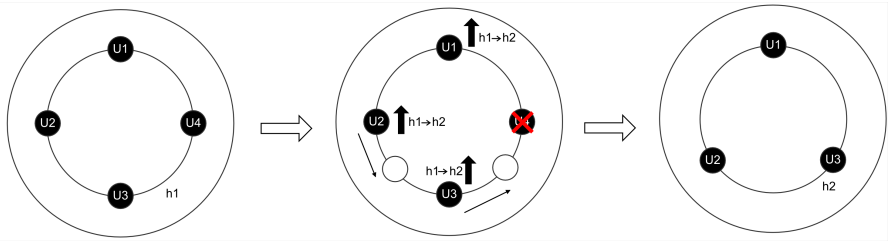


Fig. 2.8: Cognitive formation flight-self compensation.

has the lowest wind resistance and guide the UAV fleet to follow the most efficient path towards the destination. Besides those cognitive formation flight scenarios explained above, there are some other scenarios regarding efficiency, cost and robustness advantages of the cognitive formation flight. In other words, the cognition abilities make the coordination of the cooperative UAV system more secure, resilient and economical. The cognitive process for multi-UAV formation flight is shown in Figure 2.9.

2.3 Chapter Summary

This chapter introduces the concept of cognitive personal remote sensing, which includes personal remote sensing using UAVs and cognitive formation flight. In the first section, the notion of personal remote sensing is briefly explained, and then three practical scenarios related to multi-UAVs are described and illustrated. In the second section, the concept and a basic frame work of the cognitive formation flight are presented. Then two cases are presented to explain how the cognitive abilities can help improve the stability and performance of the cooperative UAV system.

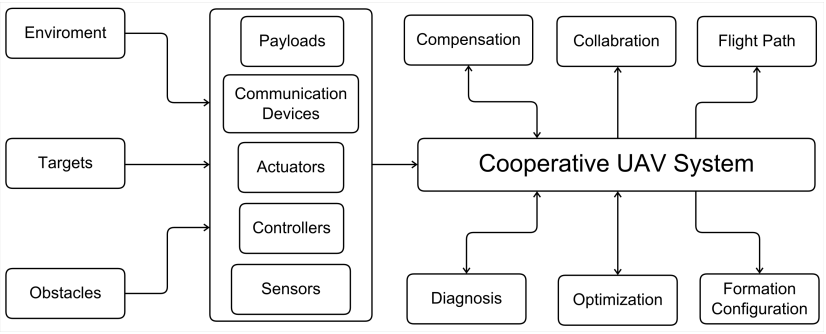


Fig. 2.9: Cognitive formation flight process.

Chapter 3

Autonomous Flight of a Single UAV

3.1 Introduction

Radio control (RC) aircraft have been favorite toys of aviation hobbyists for years because they are relatively inexpensive to obtain, straightforward to assemble and joyful to control. RC aircrafts not only bring the hobbyist similar experience in flying airplanes, but also can be extended in many applications, such as stunt flight show, flying targets for military shooting training, aerodynamic research, etc. With an experienced RC pilot, they can be deployed for reconnaissance and surveillance purposes. Although RC aircraft have potential in many areas, their reliability and other performance aspects are limited if humans are always in the control loop. When the RC aircraft is far away from the pilot, it is difficult for the pilot to identify the instantaneous attitudes and altitude. Therefore, the aircraft has to always stay within a certain range where the pilot's line of sight can reach. When the RC aircraft is flying, typically there is no feedback to the pilot, such as when the fuel will be drained, how well the actuators perform, which are all based upon the pilot's accumulated experience. A critical drawback of RC aircraft is their fail-safe features. If a component malfunctions and jeopardizes the safety of the aircraft, only the pilot can save the situation and avoid damage to the plane. If the aircraft crashes in an open area, it can cause more challenges in retrieval if there is no GPS position feedback available.

For the purposes of resolving the drawbacks and extending the usage of RC aircraft, to convert them into unmanned aerial vehicles (UAVs) by installing navigation and communication units is a reasonable approach. However, most autonomous navigation units available in the current market are not really applicable to inexpensive RC platforms because of the higher costs [20]. Therefore, designing and integrating an autonomous system

on an RC aircraft that can both improve its autonomy and maintain the overall cost as low as possible becomes the preferred solution. Researchers have made efforts towards this direction [21–23]. The system integrations generally involve both aerospace and electrical expertise, and if customers purchase off-the-shelf autopilot and avionics systems, they are usually not only expensive but also not open-source. This means it is usually not possible for the researchers to test their own algorithms or implement new functions. If users decide not to purchase off-the-shelf RC aircraft but to build their own RC testbed, the whole design process will involve considerations in aerodynamics and stability. Prior equipping the vehicle with all the avionics and payloads, the validation has to be performed for stable RC flights.

In order to achieve the goal stated above, this chapter gives our systematic approaches on developing low-cost UAV systems, focusing on the RC platform selection, system integration, surveys on additional alternative hardware and flight performance analysis. This chapter specifically presents the author's work on the developments of several major UAV platforms in CSOIS for different purposes. It first briefly introduces the AggieAir UAS platform, and then details the history, functionality, design, configuration, performance of the low-cost UAV testbed and Boomtail fixed-wing platform. Afterwards, it introduces the hardware and software architectures with major components on the low-cost UAV testbed, such as autopilot, IMU, on-board computer, etc. A flight test protocol for single UAV flight is given.

3.2 Platform Overview

In order to carry heavier payloads, such as digital cameras, thermal infrared cameras, the development of 72-inch AggieAir UAS [16] platform started in the summer of 2008, and the author built the first two experimental aircrafts carrying infrared sensors for autonomous navigation. Then a GX2+Gumstix configuration was implemented [24] and the navigation performance of the AggieAir UAS got significantly improved. After that the author helped to build the first AggieAir UAS for UWRL and assisted with finalizing all the parts and writing the construction manual. The AggieAir UAS is now a mature platform and has

been widely used in many civilian applications [25].

In order to manage the multi-UAV formation flight research, the UAV with delta wing configuration and a wingspan of 72-inch (AggieAir platform) was originally considered because it is equipped with a commercial IMU for attitude estimations of high fidelity. However, since most 72-inch UAVs have been used for georeferencing purposes [16], they also carry other electronic payloads. For the realistic multi-UAV formation flights, there involves real-time controller tuning and the current multiple UAV control scheme has been mostly tested in simulations without sufficient practical experience. Besides those, the size and total value of each 72-inch UAV system also introduce fly zone and damage control issues, so it involves risk and uncertainty to make demonstration flights with these platforms.

Based on the reasons described above, the similar design but smaller 48-inch UAV has replaced the 72-inch UAV as the new experimental platform and there are several advantages of utilizing the 48-inch UAV.

(1) Lower cost. The 48-inch UAV originally uses inexpensive infrared sensors as the navigation unit, and it does not need to carry other equipment related to flight missions or research, which significantly reduces the total cost of building a new system and the loss if the UAV crashes.

(2) Stability. The 48-inch UAV was the first platform built at CSOIS by the author and it has been tested many times. One 48-inch UAV was used to participate in the 2008 AUVSI student UAV competition and won the second place award, which proves the reliability of the 48-inch UAV platform.

(3) Size. In a constricted air space, it allows more UAVs to fly at the same time, which can help test the capacity of the ground communication device or reduce the risk of collisions.

(4) Maneuverability. The lighten weight of the 48-inch UAV greatly increases its maneuverability and benefits the formation flight.

There have been several 48-inch UAV testbeds in the current flight service. Detailed descriptions and flight performance evaluations are provided in a subsequent subsection.

Although AggieAir platform is able to fulfil many application requirements, some inherent characteristics due to its configuration have limited its potential in many other applications. The first limit is payload capacity and facility. Because of its delta wing design, most of the payloads have to be located in the center of the airframe to maintain the best balance. However, the thickness of the fuselage is restricted so that it can only handle certain types of cameras, and the weight can not exceed a specific limit as well. The UAV configuration does not have a tail, and there are no rudder or elevator. Its roll and pitch angle controls are achieved through the elevons, which is a combination of elevator and ailerons. Compared with traditional RC aircrafts, this flight vehicle configuration introduces less drag and consequently has higher fuel efficiency [26]. However, it is inherently more difficult to control and less manoeuvrable because of the tailless configuration. Additionally, because it is built from a COTS airframe, the whole aircraft can not be detached, which causes difficulties in transportation.

Based on those factors introduced above, the Boomtail in-house airframe design was brought up in 2009 and the author led the efforts to make it achieve full autonomy from an immature stage. The UAV was introduced in the AUVSI SUAS 2010 competition and several flight demos with different payload systems were performed after that. The success in those flights validated its advantages over the COTS design. Some of the significant performance improvements include:

(1) Higher payload capacity,

(2) Increased stability,

(3) Improved handling and maneuverability,

(4) Transportability.

Details regarding the Boomtail platform are shown in a subsection.

3.2.1 AggieAir UAS Platform

The AggieAir UAS platform is based upon a COTS airframe and underwent several rounds of modifications to improve the design. The current platform is made of EPP foam and covered with two layers of tapes. Three carbon spars are disposed as a triangular shape and embedded into the foam to enhance the rigidity. In order to distinguish the surface and bottom, they are covered with different colors of tape. It usually takes up to 40 hours to manually build one AggieAir airframe.

This aircraft is powered by electrical batteries so it does not cause any pollution. Some detail specifications are shown in Table 3.1 [27]. A bungee is used to launch the aircraft and it lands using its belly so there is no need for a dedicated runway. Its foam design absorbs the collision forces and properly protects the onboard electronics. The sample layouts of the airframe and its main bay with two cameras and the navigation unit are shown in Figure 3.1 and 3.2, respectively.

Table 3.1: AggieAir UAS specifications.

Weight	about 7.3 lbs
Wingspan	72-inch
Endurance Capability	about 1 hour
Cruise Speed	15 m/s
Operational Range	up to 5 miles
Operating Battery Voltage	10.5V-12.5V
Control Inputs	elevon & throttle

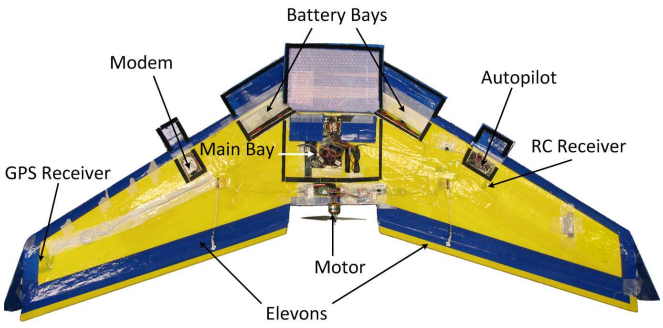


Fig. 3.1: AggieAir airframe layout.

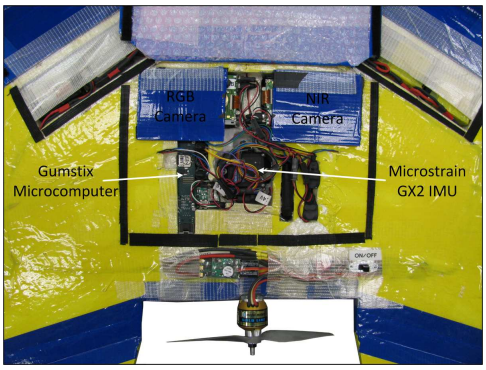


Fig. 3.2: AggieAir main bay.

3.2.2 Low-cost Miniature UAV Testbed

Compared with traditional RC aircraft, the RC flying wing has a simple configuration, introduces less drag and consequently has higher fuel efficiency. Therefore, it was chosen as the UAV development platform. One of the leading RC flying wing manufacturers in the market is Zagi, which produces different configurations of airframes and all required radio control required accessories [28]. However, due to cost, it was decided to design our own UAV platform based on a raw airframe from Unicorn Wings without any accessories.

The airframe is also made of EPP (Expanded Polypropylene) foam, and strapping tape is used to cover the entire surface so that the main body is well protected. When the left and right EPP wings are glued together, the wingspan is about 48 inches (122 cm). Once everything is ready to be installed, the procedure in a formulated detailed construction manual is followed to install in the airplane with all the accessories including electrical motor and servos, autopilot, RC receiver, wireless communication unit, and navigation unit. A finished 48-inch flying-wing UAV that is ready for autonomous navigation is shown in Figure 3.3 and the specification of the 48-inch UAV is listed in the Table 3.2.

Compared with many other UAV platforms, the 48-inch flying-wing UAV testbed has the following positive specialities.

(1) Light weight. The airframe is constructed of foam and tape, so the total weight of the main body is less than 3.5 lb (1.59 kg), which leaves extra capacity for payload

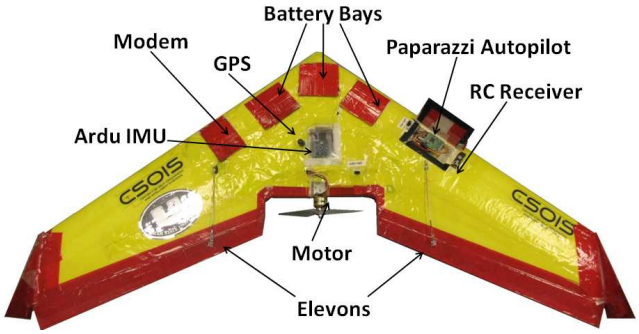


Fig. 3.3: 48-inch UAV layout.

Table 3.2: 48-inch UAV testbed specification.

Weight	3.3lb or 1.5kg
Endurance Capability	45 minutes
Cruise Speed	15 m/s
Control Inputs	Elevons & throttle
Operational Range	5 miles
Operating Battery Voltage	10.5V-12.5V
Operating Temperature	14 -104°K

weight given its current motor lift.

(2) Runway free. The UAV uses a bungee to take off and features belly landing; therefore it does not need a special runway to operate.

(3) Durability. The UAV has flown for numerous hours under different weather conditions with no modification. The material is resilient to temperature changes and because all the cables are embedded in the foam, they are protected from wear.

(4) Safety. The main body of the UAV is soft so it can absorb most impact and protect the on-board avionics that are secured inside the wing. Because of its small size and weight, and since it uses electrical power, there is minimized risk of injuring people or damaging properties.

(5) Flexibility. The foam structure makes it easy to cut and create spaces for supplementary batteries and new payloads. All the avionics can be moved around to achieve the

best aerodynamic balance for the airframe.

(6) Open-source solution. The autopilot and navigation units are all based upon open-source software and hardware. With the support from the community, people share ideas and resolve each others' questions, which is convenient and supportive for our project development. Other researchers have easy access to the resources of USU UAV team and can thereby improve their own designs.

(7) Low cost. Based on the open-source software and hardware, significant reduction of investment is achieved. All the on-board components are selected taking into account price and performance to achieve the desired low-cost scenario.

In order to achieve reasonable navigation performance, attitude estimation with high fidelity is indispensable. Accurate orientation measurements are crucial for the flight controller to stabilize the entire UAV system and to ensure smooth autonomous flight performance. Inertial measurement units (IMUs) are popular on UAVs and they play an important role in attitude estimation because of their high accuracy. GPS is another important navigation sensor because it can measure position, altitude, velocity, and course angles of the UAV. By combining both IMU and GPS, most essential data for UAV autonomous navigation is provided. However, most commercial IMUs are expensive due to the high quality hardware and sophisticated algorithms. For the purpose of balancing performance and cost, the team decided to explore low-priced sensor solutions.

With the development of low-cost inertial and navigation sensors, there have been several inexpensive IMUs and GPS available in the current market. Relying on less sophisticated algorithms, they work similarly to the expensive commercial sensing systems while the price is less than 200 US dollars [29]. Even though their accuracy is less than that of commercial systems, they are sufficient for low-cost UAV development. One of the low-cost navigation units combines Ardu IMU and uBlox GPS. Ardu IMU was originally introduced by DIYDRONE at a cost of 100 US dollars [29]. It consists of a 3-axis accelerometer which is used to measure linear accelerations and a 3-axis gyroscope that is used to measure the angular velocities. The processor is Arduino-compatible that runs the filtering and parsing

code. Figure 3.4 shows a sample Ardu IMU. In order to estimate the orientation angles, a direction cosine matrix (DCM) complementary filter is implemented [30] and it can output attitude estimates with a frequency of around 50 Hz. The uBlox GPS receiver is a popular solution for navigation because it is inexpensive and powerful. It can update up to 4 Hz and it has been integrated into the Ardu IMU.

Many flight tests have been performed and here shows a series of flight test results collected in the Cache Junction research farm belonging to Utah State University. Both IMU and GPS sensor data were saved through Paparazzi's logging function. From Figure 3.5(a) to Figure 3.6, we show the roll angle tracking errors, pitch angle tracking errors, altitude tracking, course angle tracking, and flight path, respectively. The results are all highlighted for the autonomous flight mode so that they can show the comprehensive performance of this system. From Figure 3.5(a) and 3.5(b), it can be observed that the roll channel tracks pretty well so that the UAV has consistent circling performance. The pitch channel is not as well behaved as the roll channel due to the flying-wing design, but most of the time the UAV has sufficient ascending and descending performance. Shown in Figure 3.5(c), the altitude is maintained close to the reference with a small oscillation due to wind disturbance. Figure 3.5(d) shows the actual course tracking given the reference course from GPS and they are close to each other. The last figure shows the smooth autonomous flight path, which includes standby circling, line tracking and circling, and autonomous landing. The autonomous landing is achieved through several functions in the flight plan. Basically the UAV first circles down to an altitude of 50m based on the GPS estimation, then it flies towards a touchdown waypoint at the ground altitude with attitude controls and zero throttle. We have also successfully tested autonomous takeoff and it is achieved using a similar concept as the landing. We first find the exact GPS coordinate where the bungee is located, and then we extend the bungee to launch the UAV. When the UAV passes the bungee waypoint, its throttle will be turned and climb to a certain altitude. During this process, its attitude control is also activated so it can confront small cross wind.

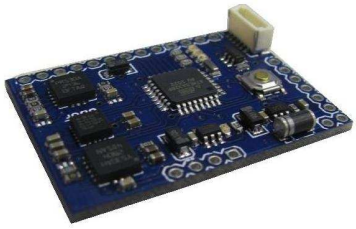
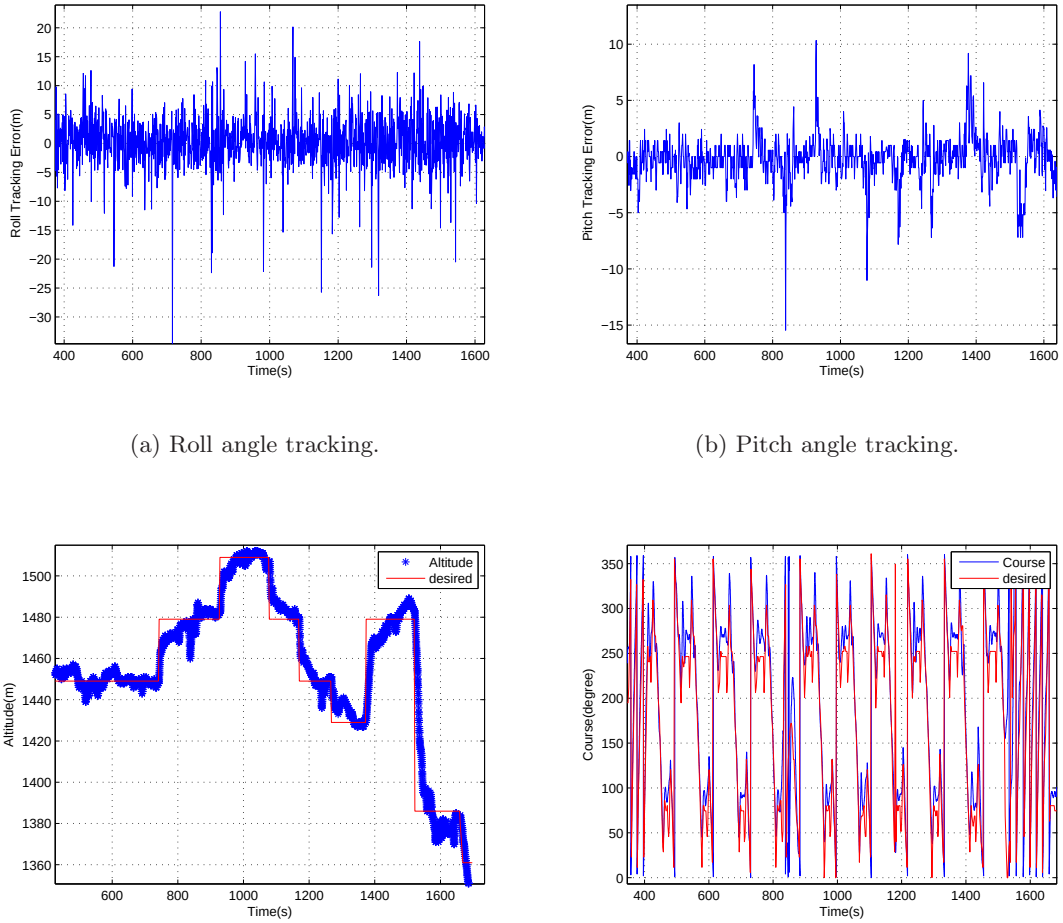


Fig. 3.4: Ardu IMU.

3.2.3 Boomtail Fixed-wing UAV Platform Development

This airframe was originally designed for the 2009 AUVSI SUAS competition by a group of students from USU majoring in aerospace engineering but it did not provide acceptable performance to be used at that time. As a matter of fact, it had never flown under autonomous mode beyond one minute. Then the airframe design has undergone another year of development by some other aerospace engineering students. The author took over this project when the second round of design began and led most of the efforts until satisfactory autonomous flying capability was achieved and we used it for the 2010 AUVSI SUAS competition with excellent performance.

Compared with the AggieAir platform and the low-cost UAV testbed, the Boomtail employs a conventional tail plane design. The tail design solves engineering challenges including balance, backward center of gravity, and insufficient stability. With a separate fuselage and wing areas, there is more flexibility in manipulating the location for different parts. Based on the detachable tail and wings, a modular airframe design was realized, which resolves the problem of transportation since everything can fit into a standard suitcase. The total width of Boomtail is almost the same as that of the AggieAir platform while its weight is almost twice heavier. Most of the additional weight is from the fuselage, because in order to carry more payloads, it was designed to be much thicker than the previous design and several aluminum tubes are enclosed in the airframe to realize the modular design while the rigidity can still be maintained. Besides, its foam body is wrapped with fiberglass and poly cover for increased strength. Other features of Boomtail design are the avionics and payload modular designs. On the AggieAir platform, because there is insufficient special



(a) Roll angle tracking.

(b) Pitch angle tracking.

(c) Altitude tracking.

(d) Course angle tracking.

Fig. 3.5: Testbed flight performance.

space for all the parts, most of the parts have to be distributed in different locations as shown in Figure 3.1, and in order to retain correct balance, those locations have to be carefully estimated. However, where to install all the electronic parts for Boomtail is no longer an issue because all the parts can be managed in the fuselage. With two plastic plates, the avionics and payloads can be easily arranged so that they can handily fit into the fuselage in an organized manner. The sample airframe layout and the modular designs of the avionics and payloads are shown in Figures 3.7 and 3.8, respectively.

Regarding flying performance, the most significant improvements are stability under

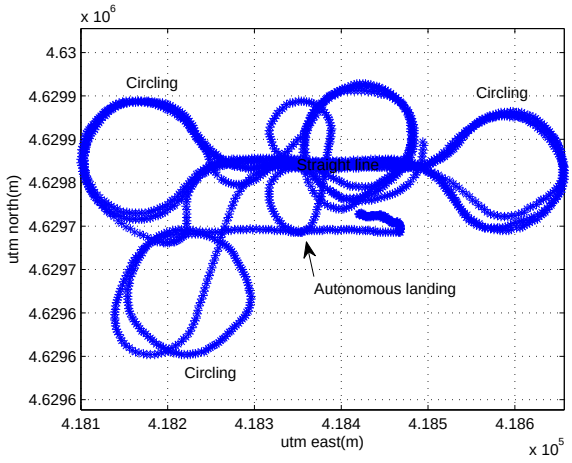


Fig. 3.6: Flight path.

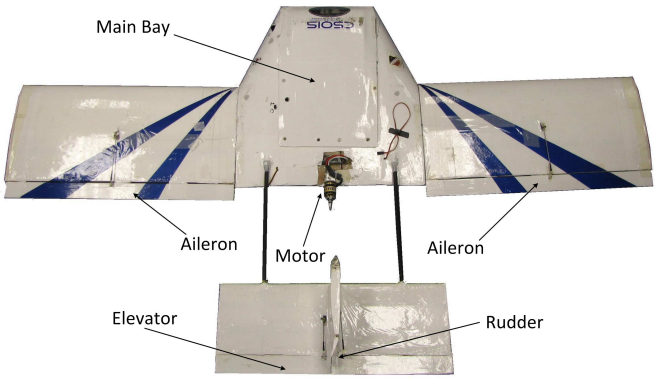
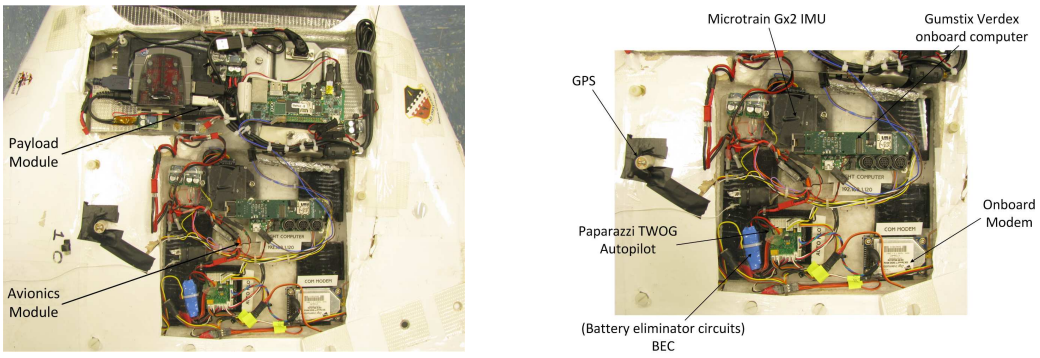


Fig. 3.7: Boomtail airframe layout.

wind disturbance and turning ability with the addition of yaw control. Due to the conventional airframe design and more powerful motor, Boomtail can perform under stronger wind conditions than the AggieAir platform and correspondingly, is more resilient to the disturbances while circling. However, due to its weight, it is fairly difficult to use the bungee to launch the airplane. Therefore, an advanced launcher design is being engineered. Some preliminary flight test results of Boomtail are shown from Figure 3.9(a) to 3.9(d), respectively.



(a) Boomtail main bay.

(b) Boomtail avionics module.

Fig. 3.8: Boomtail main bay and module.

3.3 Hardware Architecture

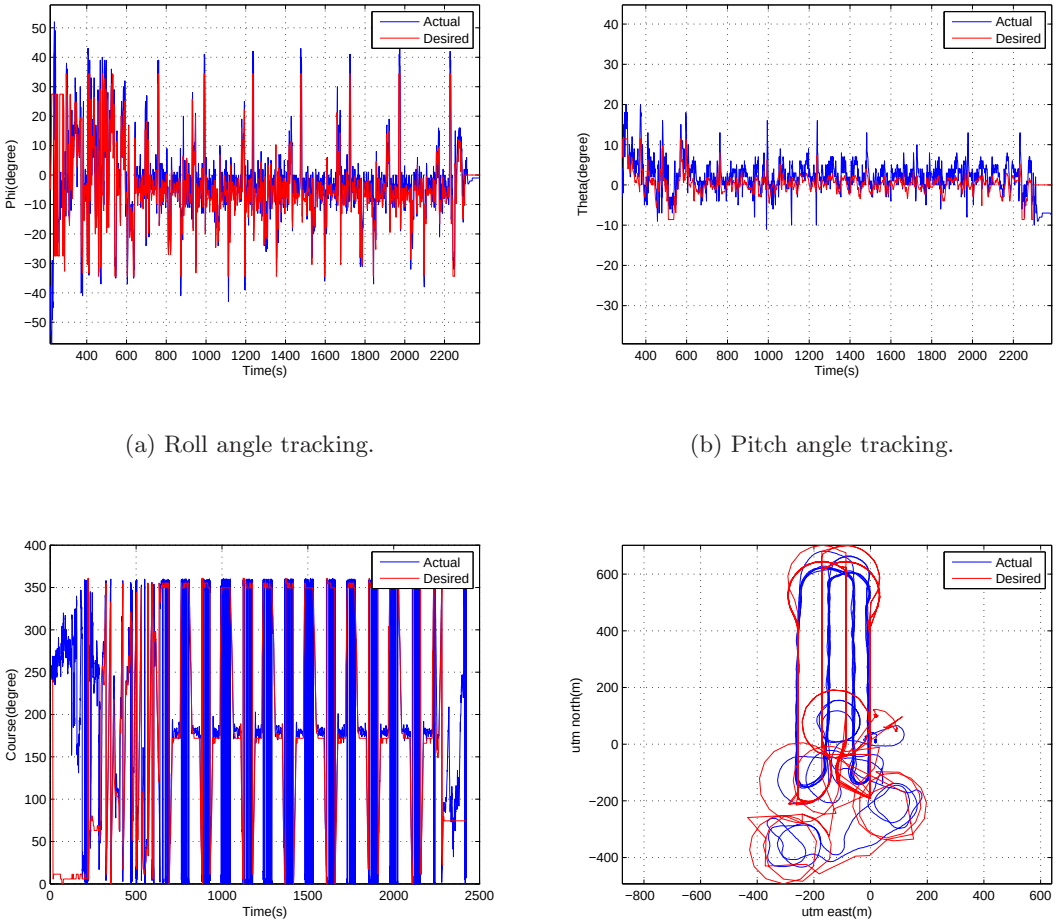
This section focuses on a low-cost UAV testbed and explain the hardware architecture in detail regarding design issues and the implementation process. The hardware includes the major components of the system block diagram shown in Figure 3.10. The current communication units include one 72MHz RC receiver for the safety link and a 900MHz wireless modem for the datalink. The modem is able to handle up to 40 miles [31] and we usually limit the flight area within one mile due to legal reasons. If the datalink has been lost for 30 seconds, the UAV will return to the base station and circle around it. Then the safety pilot can take over the control. A differential air pressure sensor that can measure airspeed and provide feedback for closed-loop speed control has also been designed and implemented for the system using the ADC port.

(1) RC airframe

(2) Autopilot

(3) Navigation units

(4) Communication units



(a) Roll angle tracking.

(b) Pitch angle tracking.

(c) Course angle tracking.

(d) Flight trajectory tracking.

Fig. 3.9: Boomtail flight performance.

3.3.1 Airframe

As introduced before, the airframe is a 48-inch wingspan delta wing made out of foam. Due to its small size, the balancing and drag reduction are critically important for flight performance. After constructing almost all the delta wing airframes for CSOIS and gaining the most comprehensive experience during the author's undergraduate studies, the author is able to make the most delicate flying wing airframe right now. As shown in Figure 3.11, most surface areas are smooth except for the GPS antenna due to the height of the receiver. The covers can firmly fasten all the components without increasing extra drag.

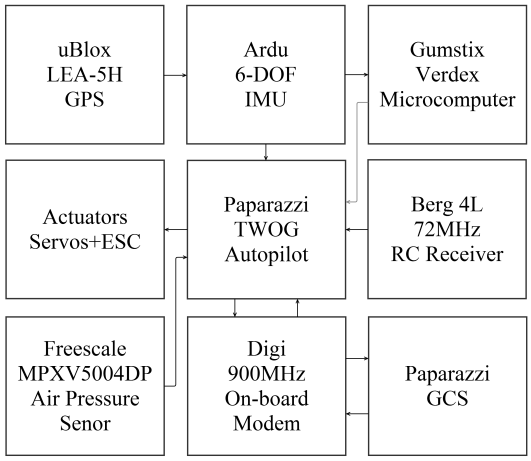


Fig. 3.10: System block diagram.

Fig. 3.11: Smooth airframe surface.

The balancing of the airframe is accomplished using a open source software called CG calculator [32] to find the central gravity line. First the following parameters were collected by measuring the airframe as illustrated in Figure 3.12.

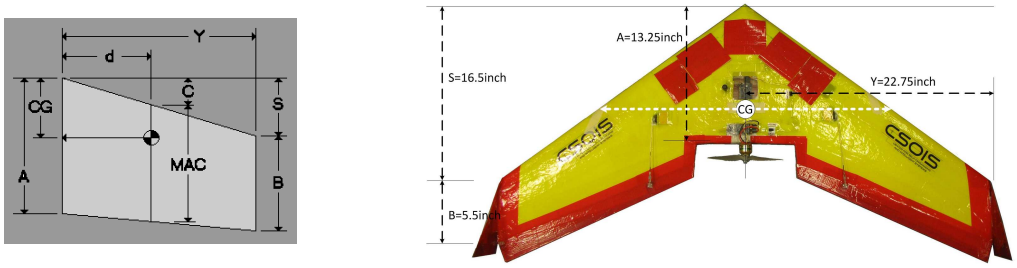
(1) Root Chord (A)

(2) CG Graphic Enter Tip Chord (B)

(3) Sweep Distance (S)

(4) Half Span (Y)

(5) %MAC Balance Point



(a) Calculator with all the parameters.

(b) 48-inch UAV illustration.

Fig. 3.12: Central gravity calculation.

The following equations were used to find the CG line.

$$C = (S(A+2B))/(3(A+B)), \tag{3.1}$$

$$MAC = A-(2(A-B)(0.5A+B)/(3(A+B))), \tag{3.2}$$

$$d = (2Y(0.5A+B))/(3(A+B)), \tag{3.3}$$

$$CG = \%MAC \times (MAC)+C. \tag{3.4}$$

where C is the Sweep Distance at MAC, MAC means Means Aerodynamic Chord, d means MAC Distance from Root.

After calculating the central gravity point and balancing all of the accessories, the near-optimal position for each component could be found. Figure 3.13 shows the placement of all the components on the airframe.

After all electronics are installed in the airframe, it is ready for a set of in-door tests, such as actuators check, RC range check, datalink test, etc. Then it is taken to the field for RC tunings and autonomous tunings. Finally, the UAV is able to perform stable autonomous navigation and fulfil different mission performance requirements.

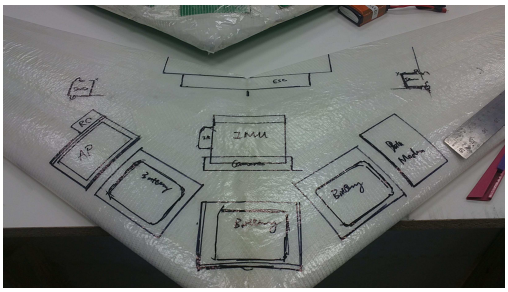


Fig. 3.13: Component placement.

3.3.2 Autopilot

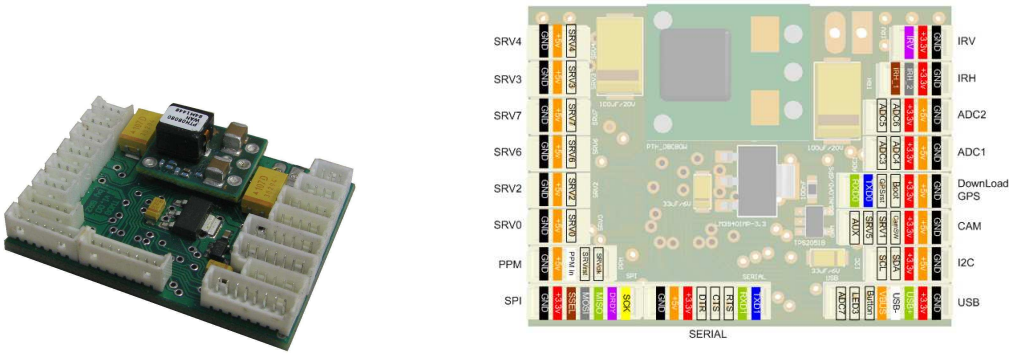
The autopilot is the brain of a UAV. It plays an essential role in autonomous navigation by collecting and processing sensor data then generating commands to the actuators for correct guidance of the flight. In order to choose a suitable autopilot that satisfies requirements, several available commercial-off-the-shelf (COTS) autopilots [33–36] were surveyed and compared in Table 3.3.

Table 3.3 illustrates that Procerus Kestrel and Micropilot MP2028 are both small, lightweight, and powerful autopilot choices. However, both are expensive closed-source choices. Closed-source means that the users are only able to manipulate the standard functions as the internal software is inaccessible. The users are prevented from implementing new flight control algorithms and integrating other hardware. The Paparazzi TWOG is an open-source autopilot including complete software support. A sample TWOG is shown in Figure 3.14(a). It is made up of an ARM7 micro-controller running at 60 MHz and it executes all the control loops for the autopilot. TWOG has eight analog input channels, one SPI bus, one I^2C bus, one UART with 3.3 V to 5 V, one client USB port, one switching power supply providing 6.1 V to 18 V voltage and weighs 8 grams [14]. The pinout of the board is shown in Figure 3.14(b).

The open-source settings make it a flexible, effective and inexpensive solution for low-cost UAV development [14]. Options are also available for other hardware to be integrated into the system so that more functions can be activated. The same autopilot has been used on other platforms and hours of successful autonomous flights have validated its robustness

Table 3.3: Autopilot comparisons.

Autopilot	Micropilot MP2028	Cloud Cap Piccolo SL	Procerus Kestrel V2.4	Paparazzi TWOG
Cost(k USD)	5	N/A	5	0.125
Size(cm)	10x4x1.5	13x5.9x1.9	5.1x3.5x1.2	4x3x0.95
Weight(g)	28	110	17	8
CPU	3MIPS	40MHz	29MHz	32-Bit ARM7
Vin(volts)	4.2-26	5-30	-0.3-16.5	6.1-18
Power	140mA (6.5V)	4w	500mA (3.3 or 5V)	N/A
Memory	N/A	448KB	512KB	32KB



(a) TWOG appearance.

(b) Board pinout.

Fig. 3.14: Paparazzi TWOG autopilot.

[37]. Moreover, an advanced flight controller has been designed and implemented based on the same software and hardware [27]. The accomplished test results demonstrate the capability and potential of this autopilot.

3.3.3 Navigation Units

As introduced in the previous section, the low-cost Ardu IMU is utilized as the navigation unit. The Ardu IMU is designed to accept GPS information for yaw drift correction and direct parsing all the navigation data to the TWOG autopilot through its UART port, Figure 3.15 illustrates all the ports available on the Ardu IMU. In order to quantify the

performance of Ardu IMU, a logging system based on Gumstix Verdex microcomputer has been designed so that the entire inertial sensor and GPS data can be saved into a SD card for further data analysis. Gumstix microcomputer consists of a Verdex Pro for processing and memory, a Netpro-vx for ethernet connection, and a Console-vx for serial, USB, and I^2C connections (Figure 3.16). A linux operating system is running in the processor with a speed of 600 MHz. All three components when assembled together have a total weight of only 36 g [38].

Gumstix Verdex is also able to directly parse the navigation data to the autopilot while that is an optional setting. The two configurations for Ardu IMU parsing to TWOG autopilot are shown in Figure 3.17. In order to utilize the sensor data from Ardu IMU, both airborne code and IMU parsing code follow the same Ugear format, which was created for the AggieAir platform [12]. Using this protocol, there is no need to modify the Paparazzi airborne code, so any IMU that needs to communicate with the autopilot just converts its sensor outputs into a standard Ugear format, and then the UAV can perform autonomous navigation based on that IMU. Following this protocol, the software is fully compatible with other commercial IMUs owned by CSOIS [39].

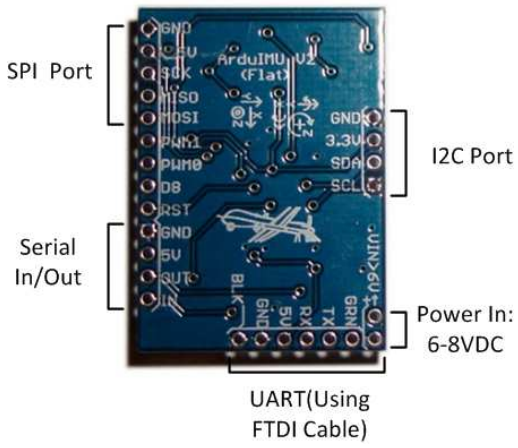
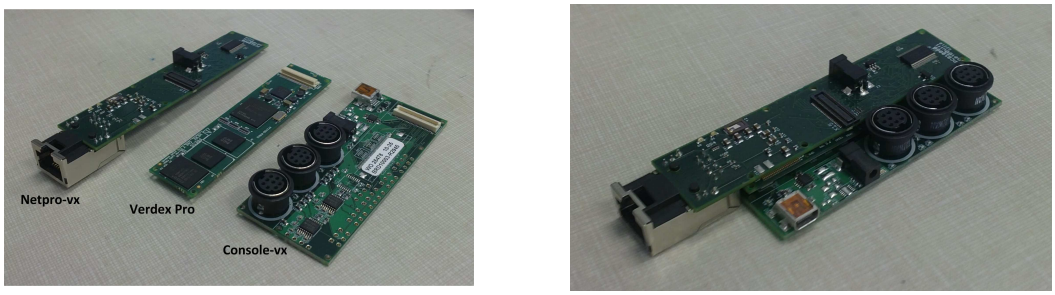


Fig. 3.15: Ports available on Ardu IMU.



(a) Gumstix components.

(b) Integrated Gumstix Verdex.

Fig. 3.16: Gumstix Verdex microcomputer.

In order to evaluate the performance of the Ardu IMU, it was compared with Microstrain GX2, a commercial IMU priced at 1700 USD. The dynamic accuracy specifies at $\pm 2^o$ [40]. After several flight tests using the Gumstix logging function, some preliminary flight test comparisons are documented in Figure 3.18 to 3.20, respectively. The raw sensor data of both IMUs was compared and it can be observed that the low-cost sensors of Ardu IMU, especially the accelerators get extremely noisy under vibrating conditions, while the gyros track closely to the GX2's, and consequently, the roll and pitch angle estimations get distorted adversely. It can be seen that Ardu is quite sensitive to vibrations, and apparently the motor on the 72-inch IMU experimental testbed which is used to carry up to four different IMUs for comparisons caused so much noise to the Ardu IMU that it could not function correctly. While working on the software development and modification, it was also noticed that the Arduino software timer was not completely punctual although most of the time it stayed at 50 Hz, which led to occasional incorrect sensor outputs. Some thorough ground testings have also been finished and it has been found that the estimation accuracy of the Ardu IMU is close to that of the GX2. In order to resolve the motor noise issue, a specially designed mounting which not only tightly holds the IMU but also absorbs the vibration is installed on the 48-inch UAVs so that the Ardu IMU can send steady navigation sensor data to the autopilot. After detail studies on the characteristics of the Ardu IMU and successfully integrating it with the Paparazzi TWOG autopilot, extensive flight tests

(a) Ardu IMU parsing through Gumstix.

(b) Ardu IMU direct parsing.

Fig. 3.17: Two configurations of Ardu IMU.

were conducted, and test results have verified the success of the new configuration.

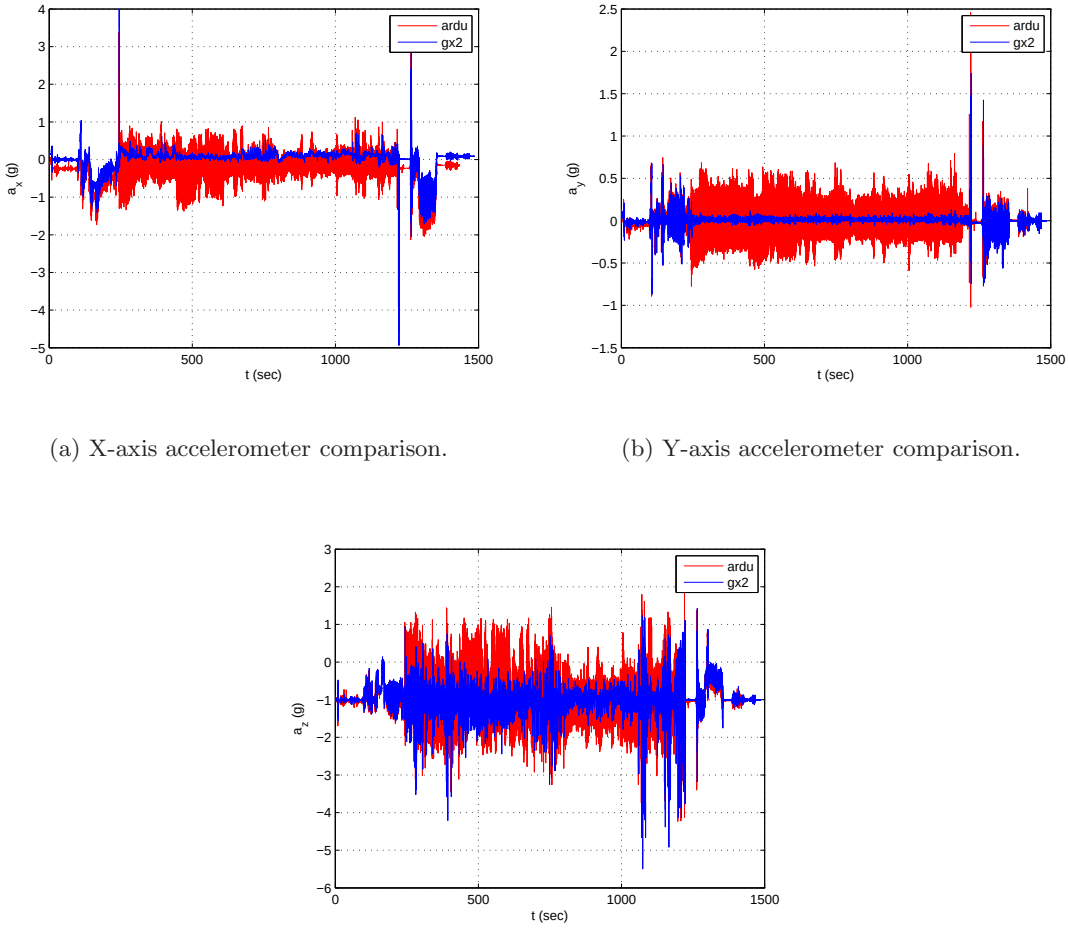
3.3.4 Communication Units

The communication subsystems include a RC receiver and a data modem. The current RC receiver is from Castle Creation with 72 MHz frequency, and we used to use the Electron 6 RC receiver before, but it has been discontinued so a Castle Berg 4 has been chosen as the replacement since its specification is similar to that of the Electron 6.

In order to generate the PPM signal and send it to TWOG autopilot, the original Castle Berg 4 RC receiver has to be modified (See Figure 3.21), so it enables the switchings among manual mode, semi-autonomous control (Auto1), and fully autonomous control (Auto2) through the transmitter. The modification procedure is documented as follows.

(1) Remove the original shrink wrap. Be careful not to damage any components on the receiver.

(2) Desolder the headers. They will not be used with the TWOG autopilot as the servos are connected directly to the AP. Cut the header off and remove the left-over pins one by one with a regular iron. There is a piece of shielding material that is connected



(a) X-axis accelerometer comparison.

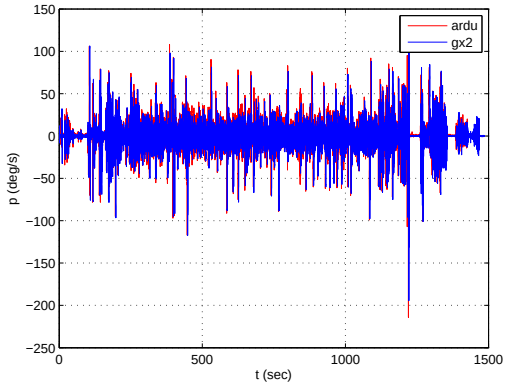
(b) Y-axis accelerometer comparison.

(c) Z-axis accelerometer comparison.

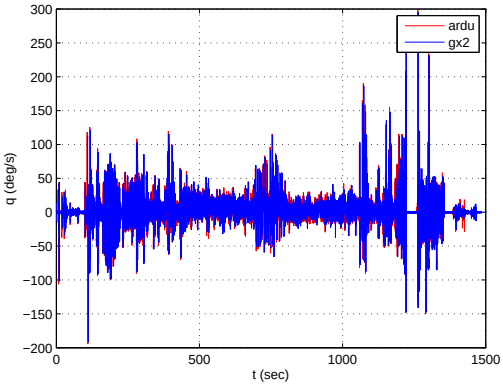
Fig. 3.18: Raw sensor (accelerometer) comparisons.

to one of the ground pins of the header. It needs to be removed carefully from the header and re-soldered to the Gnd pad.

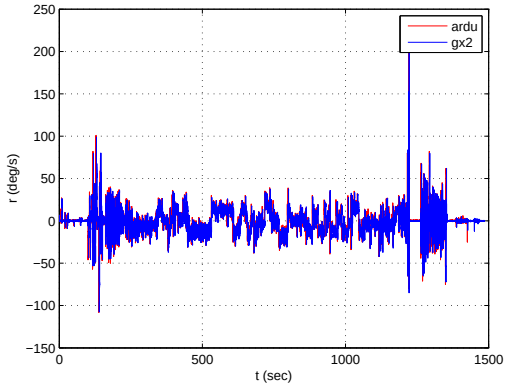
(3) Solder three wires to the receiver, which are for the ground, +5 VDC and PPM. To locate the PPM signal, first find the PIC micro controller close to the location of the headers. The PPM signal is from the pin closest to the corner of the receiver. Solder a wire directly to the pin. For the power connections, employ the pads that were used for the header. The most outside pin is for the Gnd and the second pin is for +5 VDC. After soldering the wires on the pad straight down, then loop the three wires



(a) X-axis gyro comparison.



(b) Y-axis gyro comparison.



(c) Z-axis gyro comparison.

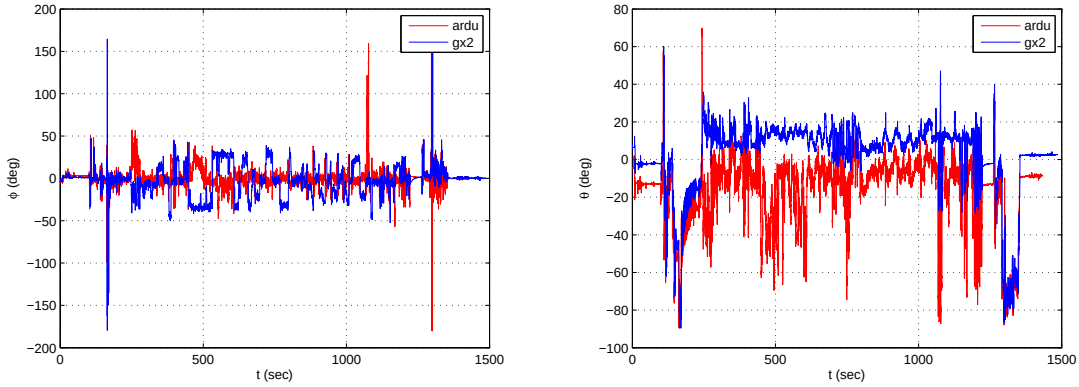
Fig. 3.19: Raw sensor (gyro) comparisons.

with 360 degrees and glue them to the PCB.

(4) Remove the crystal connector and solder the crystal directly to the PCB for more reliability.

(5) Use hot glue to add more strain to the antenna so it can be adhesive to the receiver.

(6) Cover the entire receiver again with large shrink wrap.



(a) Roll angle comparison.

(b) Pitch angle comparison.

Fig. 3.20: Attitude angle comparisons.

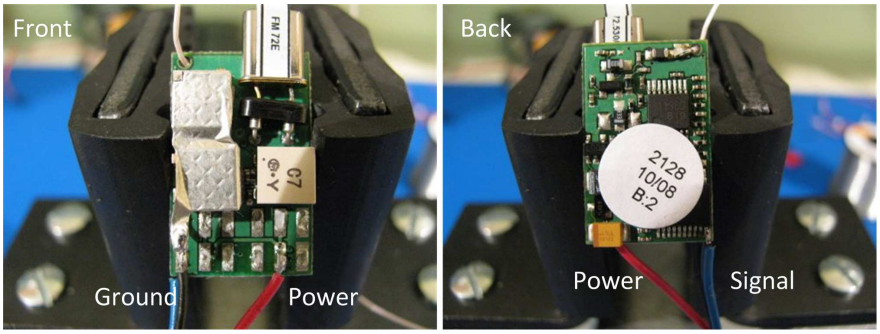


Fig. 3.21: RC receiver modification.

Whereas the RC control link is not required for autonomous flights, the link help during the tuning of a new aircraft using the semi-autonomous mode and it plays a critical role for the control safety and reliability.

The onboard and ground modems make it possible for bidirectionnal wireless communication which supports both telemetry and telecontrol. Because of the datalink, flight status is available in real time so full control of the navigation and tuning of one or several aircraft are possible from the ground control station.

The current modem systems are from Digi, the ground modem is a XTend RF Modem with 900-MHz frequency and outdoor line-of-sight range up to 40 miles (with high gain

antenna). The onboard modem is a XTend OEM RF Modules with also 900-MHz frequency and fully compatible with the ground modem. The pinouts and connections to the TWOG autopilot are shown in Figure 3.22 and Table 3.4.

In order to enhance the fail-safe features of current platform, based on the communication hardware, a datalink based manual control through a joystick which is also called the teleoperation addresses any problem requiring the need for an observer. Currently, manual operation of the UAV is achieved by the human pilot watching the UAV from the ground and using the RC transmitter to send commands to the UAV. The receiver on the UAV receives commands from the transmitter and moves the control surfaces accordingly. During autonomous operation, the human pilot acts as the observer standing-by in case another aircraft flies into the airspace. If this happened, the pilot could take manual control to rectify the problem. The difficulties with this is that from a single ground vantage point, the altitude and position of aircraft in the air can be very different than what they appear depending on the size of the aircraft and the interpretation by the observer. Avoiding a mid-air collision could be more effective if the pilot could monitor from the vantage point of the cockpit. One way to do this is to have a forward looking camera on the UAV which streams real-time video to the ground station (Figures 3.23 and 3.24). In addition, commands from the RC transmitter would be sent through the datalink for longer range control. This would not only give a better vantage point for collision avoidance, but it would also give the pilot a better idea of the location and flight of the UAV. The experimental setup has been accomplished and demonstrated on the ground. This configuration needs to be tested in actual flights to prove its effectiveness.

Table 3.4: Modem connections with TWOG AP.

9XTend Header	Name	TWOG Serial Header	Notes
1	GND	1(GND)	Ground
2	VCC	2(5V)	5V power
5	RX	8(TX)	3-5V input
6	TX	7(RX)	5V output
7	Shutdown	2	5V power

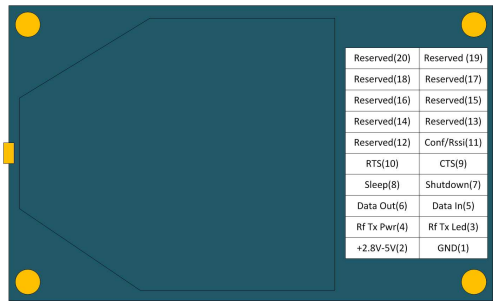


Fig. 3.22: Xtend modem pinouts.

3.4 Software Architecture

The software architecture for AggieAir and Boomtail platforms is documented in Dr. Haiyang Chao's PhD dissertation [24] since both platforms share the same hardware. This section focuses on the software architecture of the low-cost UAV testbed. Compared with the other platforms, the testbed's structure is simpler because the Gumstix Verdex micro-computer now is an optional component only when the logging function for explicit data analysis is required. For the system implementation, most processing and data parsing of the navigation function is directly accomplished through the micro-processor on Ardu IMU. A flow chart illustrating the main software architecture is shown in Figure 3.25.

3.4.1 Ardu IMU/GPS

The Ardu IMU runs a software called Arduino, which is an open-source electronic prototyping platform. The Arduino programming language and the Arduino development environment are fully compatible with Ardu IMU and it comes with basic test firmware from the factory. In order to get full functionality, the latest AHRS firmware needs to be loaded following the procedures [29].

(1) Download the latest code from the Ardu-IMU repository, which is usually called as ArduIMU(Version number).zip.

(2) Download the latest Arduino (Needs 0019 or higher, recommend 0021) and install it on the computer.

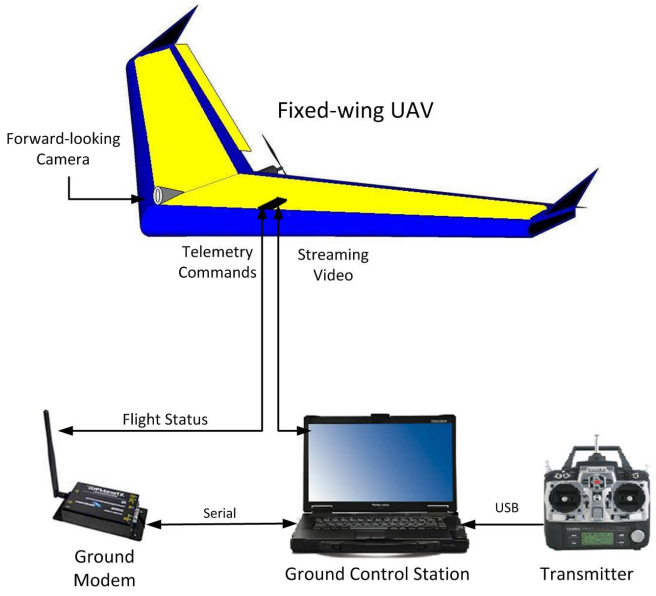


Fig. 3.23: Teleoperation diagram.

Fig. 3.24: Forward-looking camera.

(3) Launch Arduino and load the ArduIMU code from the folder by choosing the arduimu.pde file. The associated files will be also loaded at the same time.

(4) Connect a FTDI cable to the Ardu IMU and use it to load the code into the board by first verifying and building the code, then choosing the right port and uploading it.

(5) In order to check the sensor outputs, use the serial monitor function on Arduino and choose the baudrate specified in the code.

The Ublox GPS receiver can provide numerous information for navigation. It comes with an operating software called U-Center, with which one can modify certain parameters

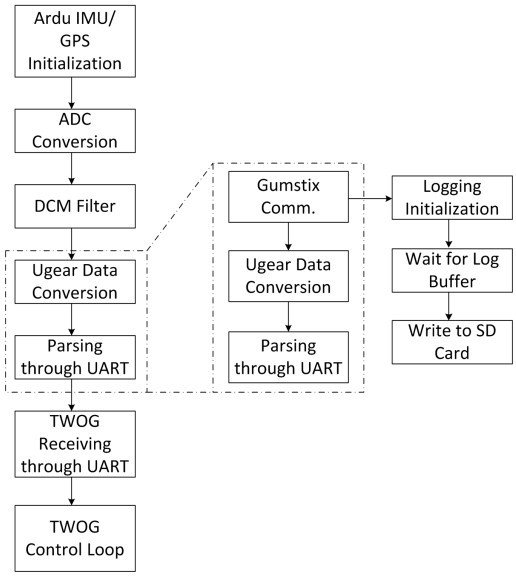


Fig. 3.25: Main software architecture.

and adjust the function of the GPS receiver. For Paparazzi application, the following parameters are indispensable:

(1) NAV-POSLLH (Geodetic Position Solution): iTOW, lon, lat, alt, alt_MSL;

(2) NAV-STATUS (Receiver Navigation Status): gpsFix;

(3) NAV-SOL (Navigation Solution Information): pAcc, pDop, numSV;

(4) NAV-VELNED (Velocity Solution in NED): velD, speed_3d, ground speed, ground course, sacc.

The parsing code of some required messages is not included in the Ardu IMU software, but it is straightforward to add a new message class and enable additional messages. Tables 3.5 and 3.6 describe the message structure and payload content with NAV-VELEND as an example [41].

3.4.2 Paparazzi

Paparazzi open-source software is powerful and flexible. It consists of both the airborne

Table 3.5: Sample GPS message structure (NAV-VELNED).

Header	ID	Length(Bytes)	Payload	Checksum
0×B5 0×62	0×01 0×12	36	See table below	CK_A CK_B

Table 3.6: Sample payload content structure (NAV-VELNED).

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS Millisecond Time of Week
4	I4	-	velN	cm/s	NED north velocity
8	I4	-	velE	cm/s	NED east velocity
12	I4	-	velD	cm/s	NED down velocity
16	U4	-	Speed	cm/s	Speed(3-D)
20	U4	-	gSpeed	cm/s	Ground Speed(2-D)
24	I4	1e-5	heading	deg	Heading 2-D
28	U4	-	sAcc	cm/s	Speed Accuracy Estimate
32	U4	1e-5	cAcc	cm/s	Course/Heading Accuracy Estimate

software and ground system software. The airborne software includes the following features [14]:

(1) PPM signal decoding for RC receiver;

(2) Servos and motor control through PPM signal;

(3) Manual control through radio frequency;

(4) Manual control with augmented stability (AUTO1);

(5) 3D Autonomous navigation (AUTO2), including:

   (1) Autonomous takeoff and landing,

   (2) Way-point navigation,

   (3) Standby circle navigation,

   (4) Altitude holding,

   (5) Complex flight plan,

   (6) Multi-UAV formation flight capability;

(6) Datalink to the ground station;

(7) Telecontrol from the ground station (flight plan switching, navigation control, waypoint modifications, control parameter tuning).

The airborne code is written in ANSI C code and compiled with the GNU C compiler. All the configuration code is translated from XML files and is segregated into fly-by-wire (manual control) and autopilot (stabilization and navigation) two processes.

The ground system software includes the following functions:

(1) Compiling tools so the airborne code can be generated from the assigned configuration;

(2) A GUI to monitor, control, and interact with different UAVs during flight;

(3) A flight simulator with the same environment as actual flights to facilitate the development of flight plans.

Although the Paparazzi autopilot can function independently from the ground control station (GCS), which is shown in Figure 3.26, the received sensor data can be interpreted by the autopilot and transmitted to the GCS through a wireless communication device. The Paparazzi GCS software is part of the Paparazzi open-source project and is one of the most powerful tools available. Its concise GUI offers an incredible simplicity to monitor and control the UAV. When new commands are sent, it will display them in the console box and generate voice messages to notify the operator. It also shows the UAV status in great detail, such as the ground speed, battery voltage, throttle percentage, flight mode, communication quality, and so on. The 2D map window of the GCS is another important segment. It displays the predefined waypoints in the flight plan and highlights the path of the UAV. The flight plan can be modified to meet new mission requirements and the UAV can automatically terminate its mission if an unforeseen problem occurs. The Paparazzi GCS is granted with a high level of control authority to ensure its effectiveness. By collaborating with the on-board autopilot, the GCS can guide the UAV to accomplish diverse flight tasks.

The Paparazzi autopilot integrates two sections for the high level flight controls, which are the navigation loop and altitude loop [14]. The navigation loop consists of the roll

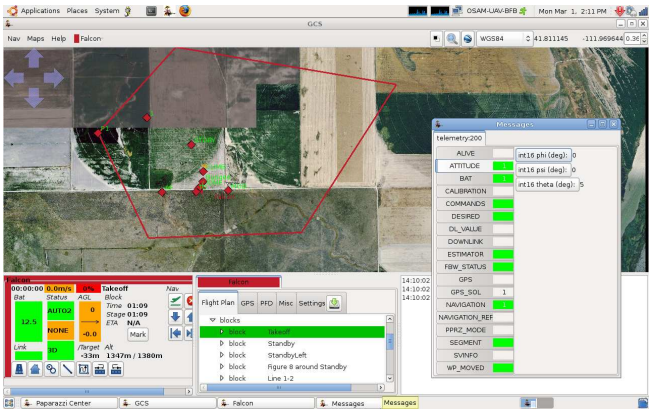


Fig. 3.26: Paparazzi GCS.

channel control and the altitude loop consists of the pitch channel control and the throttle control. It uses simple PID controllers for the low-level controls, and specifically, a proportional controller is designed for the roll channel while a proportional and differential controller is adopted for the pitch channel. The current speed loop is using an open-loop controller and a closed-loop design is undergoing. In order to tune a newly built aircraft for autonomous navigation, we follow a simple basic procedure.

(1) Launch the UAV manually using our bungee. During the RC control, the safety pilot needs to trim the elevons so the UAV can fly steadily under manual mode.

(2) Switch the control mode to semi-autonomous (Auto1), so the control is under augmented stability. With Auto1, the safety pilot can set the throttle to the cruising value defined in the airframe file and check if the UAV is able to stay at the same altitude.

(3) Depending on the performance under Auto1, the GCS operator needs to tune the roll and pitch loop neutrals so the UAV can fly straight and flat when the safety pilot has no control on the UAV except the throttle.

(4) After the semi-autonomous tuning, switch the control mode to fully autonomous (Auto2). We can first increase the roll P gain until oscillation on the roll channel happens, and then reduce it so we can find the closest gain value for the best roll

control performance. Afterwards, we can follow the same method to tune the pitch channel.

(5) When the UAV is flying in Auto2, we have designed several flight routines such as circling left and right, flying straight to refine the roll and pitch neutral value. Meanwhile, we also slightly tune the course control P gain so we can obtain the near optimal heading performance.

Once this standard procedure is finished, we change all the relevant parameters in the airframe file and load the modified files into the autopilot. Then UAV is ready to perform satisfactory autonomous flight.

3.5 Flight Test Protocols

For the purpose of achieving successful flight tests, a series of flight test protocols need to be established and strictly followed. The protocols consist of three major steps: (1) flight preparation, (2) field operation, and (3) post flight analysis. The flight preparation should be finished several hours before the flight, especially because of the necessary battery charging time. Several components need to be inspected prior to every flight such as the actuator function, datalink quality, balancing, and no damage on the airframe. The following items are enumerated in the checklist and need to be carried to the field:

(1) Complete aircraft with all removable parts;

(2) RC transmitter and RC channel tags;

(3) Ground modem;

(4) GCS laptop, flashing USB cable;

(5) For repeated flights: sufficient fully-charged batteries, battery charger, and power supply;

(6) GPS locator;

(7) Bungee kit;

(8) Tool container;

(9) Aircraft backup box;

(10) First aid kit.

While in the field, the pre-flight checklist (see Appendix A) and GCS operation procedures (see Appendix B) must be rigorously followed. The procedures are based on numerous hours of field test experience to guarantee the success and safety. When a flight test is completed, some new results and feedback need to be documented. Therefore, it is important to perform a post-flight analysis to make a informative conclusion. There are several ways to acquire the post flight data:

(1) Save a copy of the log files (.log and .data from var/logs);

(2) Save the IMU and GPS logs from the SD card on Gumstix;

(3) Transfer the image files to a laptop with flash memory backup from the camera SD card.

All the telemetry data received during a flight are stored for subsequent analysis. The update rate of certain messages can be adjusted for more explicit analysis, especially those related to the navigation and attitude estimation, such as GPS altitude and course, roll, pitch, and yaw data from IMU.

3.6 Chapter Summary

This chapter introduced several major UAV platforms developed in CSOIS for different application and research purposes, and those platforms involve most of the author's contributions on the system design, implementation, and testing. The focus is on the development of the low-cost UAV testbed for cooperative flight control research, including the airframe design, the major hardware components and the software architectures. Detail flight test data for each platform is provided to demonstrate their performance. A general flight test protocol with a standard pre-flight checklist and GCS operation manual are also shown.

Chapter 4

Attitude Estimation for Miniature UAVs

4.1 Thermal Calibration for Attitude Measurement Using IR Sensors

4.1.1 Introduction

IR sensors have numerous applications including military, scientific, medical, and security scenarios because they can detect and measure infrared radiation. IR sensors are also popular navigation instruments because they are relatively low-cost and easy to implement. They can provide acceptable measurements of distance and attitude on various platforms, including ground robots, UAVs, etc. However, the accuracy of IR sensors heavily relies on environmental factors especially on the temperature [42], which requires calibrations to be made. In account readings can directly lead to instability of the system and can adversely affect the system performance. Therefore, sophisticated methods need to be adopted and implemented on the calibration of IR sensors, so they can provide better attitude estimation performance in low-cost orientated projects [43].

Small and micro UAVs have drawn wide interest based upon their extensive applications in engineering, agriculture, and military fields [5]. Since the civilian UAV developments put high emphasis on low-cost speciality [12], cheap IR sensors become a good engineering choice for attitude estimation.

UAVs can provide the desired flight performance at low altitudes because the human pilot is replaced by an autopilot for autonomous navigation. Nevertheless, success depends upon whether the UAV has an accurate and reliable navigation system. The UAV performance is also highly constrained by the weight and power consumption. In order to complete complicated flight missions and reduce the expense, it is important to adopt an

elaborate IR sensor calibration methods for attitude estimations of the low-cost UAVs. Other researchers have used IR sensors for distance measurement in mobile robots [44] or to ascertain the absolute attitude of small UAVs [45].

This section introduces a two-stage IR sensor calibration method for attitude estimation of the low cost UAVs so that researchers can improve the navigation accuracy of their autopilot system. The major contributions of this section are: (1) to provide an experience-based IR sensor calibration method for low-cost UAV autonomous navigation, and (2) to use the IMU to help calibrate IR sensors on a UAV during actual flight.

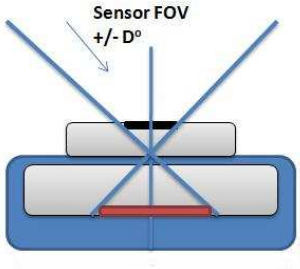
This section is organized as follows: The principle of infrared sensors and the IR sensors employed are introduced in subsection 4.1.2. The two-stage calibration method is extensively introduced and explained in subsection 4.1.3. Subsection 4.1.4 shows the flight and simulation results.

4.1.2 Description of Infrared Sensors

Infrared radiation is an electromagnetic wave with an optical frequency lower than the range of human eye's responsivity [46]. It can be generated without suffering electromagnetic interference, so it is popular for controls and communications.

Because every object emits infrared energy due to its temperature, IR can be used as the electromagnetic transmission to carry information from the operating equipment to the central system [42] and vice versa. Typical IR sensors consist of two to four pixels of equal areas made by photosensitive material. The sensor responds to the infrared radiation from objects according to their temperature and emissivity. The radiation incident upon front of the sensor is converted into signal current. The signal current is converted to signal voltage in the internal sensor circuit [42,45]. The Infrared measurements get the difference between each pair of the sensor pixels and determine the attitudes of the object. Figure 4.1 shows a example IR sensor and also the field of view (FOV) of a typical IR sensor.

There are many manufacturers of IR sensors and each model of IR sensor has specific applications. The IR sensors employed on one of the UAV projects is made by Melexis. These low-cost IR sensors have been implemented on UAVs for attitude determination [45].



(a) IR sensor field of view.

(b) IR sensor appearance.

Fig. 4.1: Sample IR sensor.

The IR sensors compared with other IMUs originally are dedicated to be part of the FMA Direct Co-Pilot Flight stabilization system [47] so they are capable of providing useful roll and pitch data once they are correctly calibrated. They have +/- 45 degree of FOV with a 90 degree full angle [47]. There are two channels (IR1 and IR2) in each sensor and both channels have two sensor faces so they can detect the difference in infrared signature from the earth surface and the carbon dioxide in the atmosphere [47]. Figures 4.2 and 4.3 show a typical installation of IR sensors on a fixed wing UAV.

To implement the IR sensor calibration, five tuning parameters are defined as follows [14]:

(1) Lateral correction (k_{lat}),

(2) Longitudinal correction (k_{lon}),

(3) Vertical correction (k_v),

(4) Roll neutral (n_ϕ),

(5) Pitch neutral (n_θ).

The lateral and longitudinal corrections are the coefficient gains that correct the IR sensor outputs and are proportional to the original value in the horizontal axis. The lateral

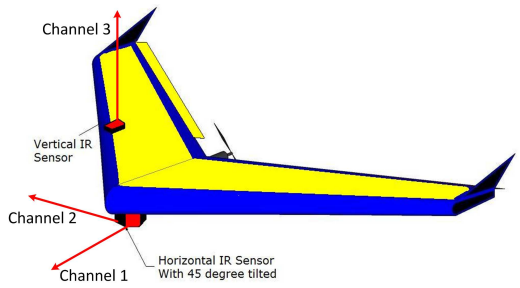


Fig. 4.2: IR sensors on UAV (3D illustration).

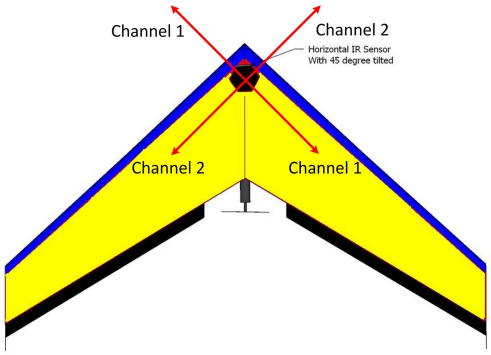


Fig. 4.3: IR sensors on UAV (horizontal plane illustration).

correction is to adjust the raw roll data while the longitudinal correction adjusts the raw pitch data. The vertical correction is the coefficient gain that is proportional to the original value in the vertical axis. The roll neutral and pitch neutral are the angles when the UAV is in a position where the bottom is parallel with the ground so the UAV can fly straight at a constant altitude. In order to measure the attitudes of a UAV, there should be at least two IR sensors [14], one of which is for the movements on the horizontal axis and the other one is for the movements on the vertical axis. For the horizontal IR sensor, its two IR channels are defined as channel 1 and channel 2. For the vertical one, its IR channel is defined as channel top.

The raw data of the roll loop (r), pitch loop (p), and top (t) are specified as the following equations [14]:

$$r = k_{lat} \times [-(\frac{s_1}{n_1} - n_{a1}) + (\frac{s_2}{n_2} - n_{a2})], \tag{4.1}$$

$$p = k_{lon} \times [(\frac{s_1}{n_1} - n_{a1}) + (\frac{s_2}{n_2} - n_{a2})], \tag{4.2}$$

$$t = k_v \times (\frac{s_t}{n_t} - n_{at}), \tag{4.3}$$

where s_1 is the sum of every value per sample and n_1 is the number of samples for channel 1, n_{a1} is the adc neutral for channel 1, s_2 is the sum of every value per sample, and n_2 is the number of samples for channel 2, n_{a2} is the adc neutral for channel 2, s_t is the sum of every value per sample, and n_t is the number of samples for channel top, n_{at} is the adc neutral for channel top. The values for n_{a1}, n_{a2}, and n_{at} are 512 by default.

Once those three types of raw data are collected, the estimated roll (ϕ) and pitch (θ) can be found by the following equations [14]:

$$\phi = \tan^{-1}(\frac{r}{t}) - n_\phi, \tag{4.4}$$

$$\theta = \tan^{-1}(\frac{p}{t}) - n_\theta, \tag{4.5}$$

where n_ϕ is the roll neutral and n_θ is the pitch neutral as indicated previously. Figure 4.4 illustrates how to compute the pitch and roll based on the IR sensor outputs. The estimated pitch value is bounded ($-90^o < \theta < 90^o$) because in reality the UAV is not able to fly upside down.

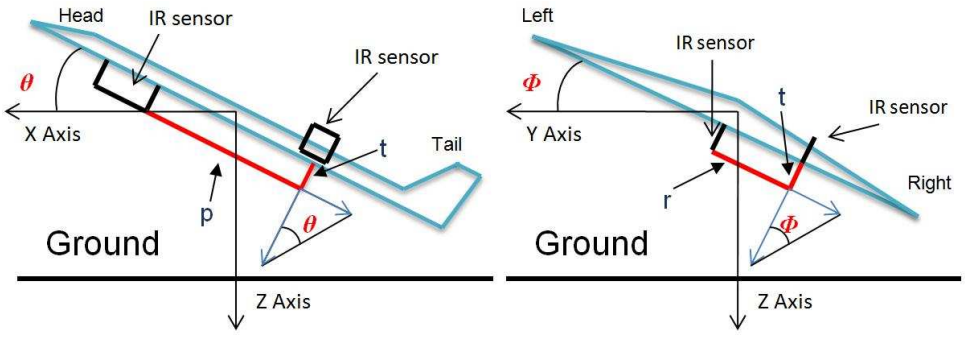


Fig. 4.4: Pitch and roll calculations based on IR sensor readings.

4.1.3 Two-stage IR Sensor Calibration Method

Based upon the previous section, the implementation of IR sensors on UAVs is straightforward. IR sensors have several advantages over other methods for attitude estimation. However, they heavily rely on the infrared radiation of different locations and various weather conditions. Thus, it is necessary to adopt a formalized IR sensor calibration method in order to achieve reasonable accuracy in meeting the low-cost development objective.

The calibration method proposed in this paper is based on two stages. The first stage is based on experience, which means the procedure is established from multiple times of successful ground and actual flight tunings. The second stage is based on the comparison of IMU and IR sensor data collected from flight tests. By utilizing the relatively more accurate IMU data, the corresponding IR sensor parameters, such as the correction gains and neutral values can be compared and adjusted so that the accuracy can be improved to be consistent with the IMU data.

Experience-based Tuning Procedures

The IR sensor made by FMA Direct is applicable to the experimental part of this research program. The Paparazzi Twog, which means Tiny without GPS [14], is used as the autopilot. One IR sensor is installed under the nose of the UAV with a 45 degree tilt to the central line. It is used to measure the movements in the horizontal axis. The sensor is installed vertically to measure the movements in the vertical axis. Both IR sensors are

directly connected to the autopilot and both channels from the horizontal IR sensor are used while only one channel from the vertical IR sensor is employed.

The basic steps of calibrating IR sensor performance are summarized as follows and the specific procedures are then documented:

Ground Testing

(1) Find the correct adc neutral value for all the sensor channels,

(2) Find the closest roll and pitch neutral values,

(3) Adjust the correction gains to match the real and defined angles;

Flight Tuning

(1) Fly the UAV and check if oscillations occur,

(2) Adjust the roll and pitch P gains to reduce the oscillations,

(3) Verify the roll and pitch neutrals from the ground testing,

(4) Change the neutrals until the UAV flies straight and flat.

Before new IR sensors are installed on the UAV, the calibration needs to be completed. First, put them into some enclosed form, and then change adc neutrals to zeros. After connecting the IR sensors with the autopilot board and flashing it, launch the Paparazzi ground control station (GCS) [14]. The actual neutral values for roll and pitch readings can be found on the GCS. Then assign these values back to adc neutrals for the three channels. The desired values are all supposed to be 512 without regard to the outside temperature and weather condition. The IR sensors used on the experimental UAV have all the adc neutral values set at 512.

After setting the adc neutrals, the roll and pitch neutrals need to be changed to zeros. Then the UAV is taken to an open area with no obstacles within a distance of 1000 ft. Next, raise the UAV into a flat position, where the bottom line should be as parallel as possible with the ground surface. There should be no blocks to the sensing ports of the vertical

and horizontal IR sensors. The actual roll and pitch readings are the roll neutral and pitch neutral values. In order to verify the neutral values, it is best to place the UAV at a certain angle. Then check whether the angle displayed matches the measured angle. If not, the lateral, longitudinal, and vertical corrections need to be adjusted so the actual angle matches the measured angle. Once the attitude window on the GCS displays consistent movements of the UAV, it is ready for the autonomous flight.

Although IR sensors can provide attitude estimations after the ground calibrations, uncertainties may still be present such as openness at the ground test area and the ground temperature. These factors can adversely affect the accuracy of the IR sensor readings, for example on very cold day [42].

Therefore, it is necessary to execute the autonomous flight tuning. After the UAV climbs to a safe altitude, the control mode can be switched to Auto1, which is the semi-autonomous mode [14]. Then check whether oscillations occur on the roll loop and reduce the roll P gain which is the dynamic tuning parameter on the GCS setting window. Once any oscillations are squelched, the roll P gain can again be increased until the UAV reaches the boundary point where oscillations could happen again. For the pitch loop, the pitch P gain is adjusted following the same procedures.

Then it is possible to verify the roll and pitch neutral values obtained from the ground test. The safety pilot needs to stay with Auto1 mode and to set the throttle to be the nominal value, namely 70%. This is the throttle percentage for operating the UAV at cruise speed [14]. Then the safety pilot is relieved of the RC transmitter control and check whether the UAV flies flat and straight. If UAV does fly this way, the flight tuning is completed. Otherwise, roll and pitch neutrals have to be changed again until the UAV flies as desired.

IMU-based Calibration

The basis of this calibration method is to install both IR sensors and IMU on the same UAV with the IMU as the primary attitude estimation sensor. The UAV continues to log the data from IR sensors. A comparison can be made and analyzed on the roll and

pitch loops between the IR sensors and IMU. Based upon the analysis and the formulas documented in previous sections, a new group of parameters related to the IR sensors can be generated, which should improve the attitude estimation accuracy of the IR sensors.

An AggieAir 72-inch UAV is used as the experimental platform, and it was built by the author from the delta wing RC airframe called Unicorn. Two FMA Direct IR sensors were installed as well as an IMU (Xsens Mti-g). Xsens Mti-g IMU has an integrated GPS and contains accelerometers, gyroscopes, magnetometers so that it can provide angle readings (ϕ, θ) up to 120 Hz with a dynamic accuracy of 1^o [48]. The layout of the 72-inch UAV Falcon is shown in Figure 4.5.

Based upon the actual flight IMU data, the referenced roll (ϕ) and pitch (θ) are given in degrees. From the IR sensor logging, the data of roll loop (r), pitch loop (p), and top (t) can be found as well. The roll neutral and pitch neutral for the IR sensors can be found after ground calibration. Based upon equations (4.4) and (4.5), the roll (ϕ_{ir}) and pitch (θ_{ir}) from the IR sensors can be found. Then use the following formulas are used based on the mean value theorem [49], to find the actual roll neutral (n_ϕ) and pitch neutral (n_θ), and the roll and pitch correction gains (k_{lat} and k_{lon}):

$$n_\phi = \frac{\sum_{k=j_1}^{j_N} \phi_{ir}(k)}{j_N - j_1} - \frac{\sum_{k=i_1}^{i_N} \phi(k)}{i_N - i_1}, \tag{4.6}$$

$$n_\theta = \frac{\sum_{k=j_1}^{j_N} \theta_{ir}(k)}{j_N - j_1} - \frac{\sum_{k=i_1}^{i_N} \theta(k)}{i_N - i_1}, \tag{4.7}$$

$$k_{lat} = (\frac{\sum_{k=j_1}^{j_N} |\phi_{ir}(k) - n_\phi|}{j_N - j_1})/(\frac{\sum_{k=i_1}^{i_N} |\phi(k)|}{i_N - i_1}), \tag{4.8}$$

$$k_{lon} = (\frac{\sum_{k=j_1}^{j_N} |\theta_{ir}(k) - n_\theta|}{j_N - j_1})/(\frac{\sum_{k=i_1}^{i_N} |\theta(k)|}{i_N - i_1}). \tag{4.9}$$

The parameters j_1 and j_N are the first and last index numbers for the IR sensor data and i_1 and i_N are the first and last index numbers for IMU data.

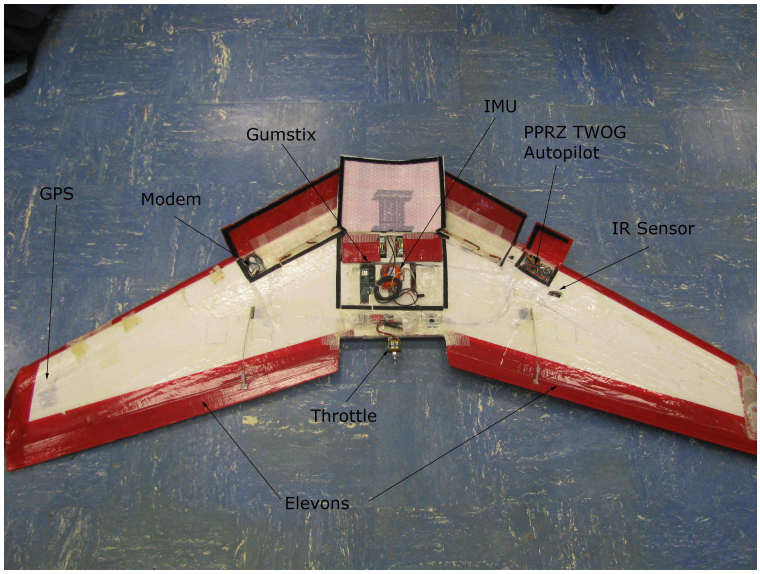


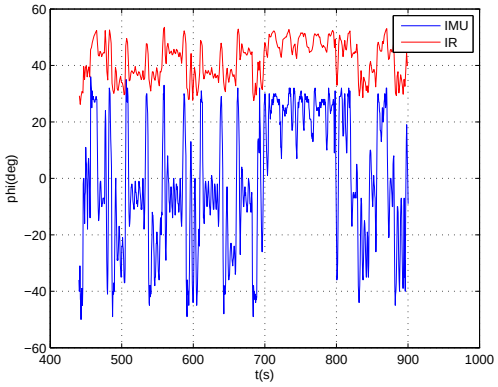
Fig. 4.5: AggieAir experimental UAV.

4.1.4 Flight and Simulation Results

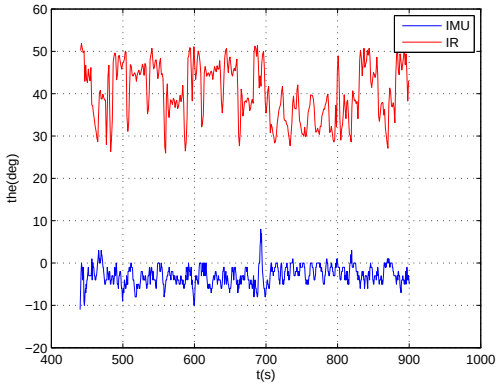
The flight test was made in July, 2008, at the Cache Junction research farm of Utah State University. Both the IMU and IR sensor data were collected for comparisons on the roll and pitch loops. From the plot of the roll loop shown in Figure 4.6(a), both curves have similar shapes except the IMU data has a proportionally greater range and the IR sensor data are about 40 degrees higher. Therefore, the lateral correction gain (k_{lat}) needs to be increased and roll neutral (n_{ϕ}) needs to be reduced. From the plot of the pitch loop, also shown in Figure 4.6(b), both curves also have similar shapes except the IR sensor data have a proportionally bigger range and it is about 40 degree higher, so it appears that the longitudinal correction gain (k_{lon}) needs to be reduced and also the pitch neutral (n_{θ}) needs to be reduced.

Based upon the equations from (4.6) to (4.9), the IR sensor data can be calibrated to be more inclined with the IMU data. Figure 4.7 shows the comparisons of IR sensor and IMU data after the calibration.

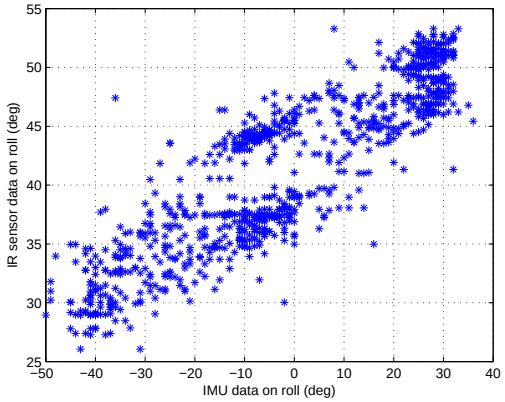
From the plots after calibrations, the roll data of the IR sensors is in better agreement



(a) IMU versus IR on roll loop.



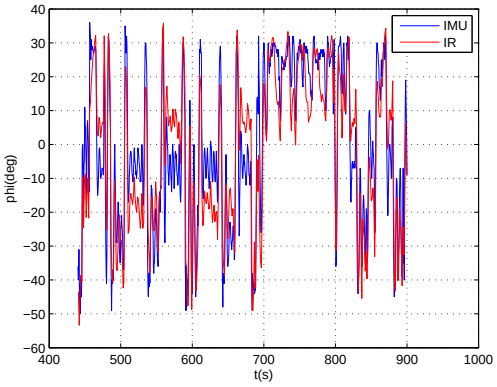
(b) IMU versus IR on pitch loop.



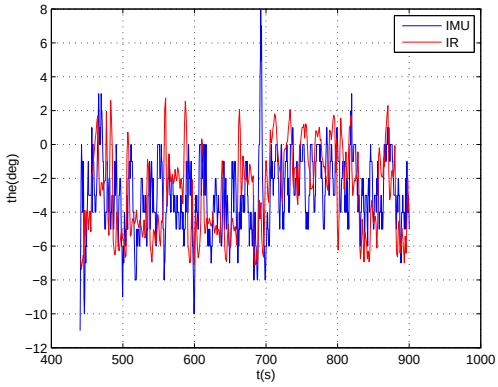
(c) Roll loop linearity comparison.

Fig. 4.6: IMU and IR comparisons before calibration.

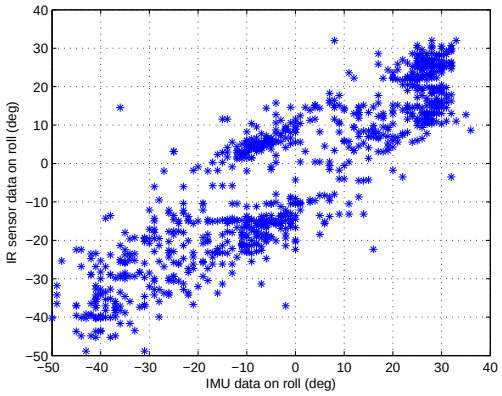
with the IMU roll data. However, there are still some dynamic discrepancies due to different sampling times and the IR sensors themselves are not perfectly calibrated. For the same reasons, the pitch data from the IR sensors is slightly different from the IMU pitch data but is in better agreement. To better analyze the sensor difference, a portion of the roll data between the IMU and IR sensors is selected and plotted in Figures 4.6(c) and 4.7(c), from which the relation between the the two sensors is seen to be nonlinear. Analysis of the nonlinearity of sensor distortion is the next step in the UAV control system research [50].



(a) IMU versus IR on roll loop.



(b) IMU versus IR on pitch loop.



(c) Roll loop linearity comparison.

Fig. 4.7: IMU and IR comparisons after calibration.

4.2 Data Fusion of Multiple Attitude Estimation Sensors

4.2.1 Introduction

Tremendous progress has been achieved in the development of miniature UAVs. Improvements include the avionics, airframes, and payloads, especially regarding size, material, and power consumption [5, 12]. Miniature UAVs have attracted wide interest because of their numerous applications in civilian, agricultural, and military areas. Since nonmilitary UAV developments put more emphasis on the low-cost feature [12], navigation systems

based on low-cost sensors become an appropriate solution to satisfy this attribute.

Low-cost sensors can provide measurements with relatively lower accuracy compared to expensive industrial or commercial sensing systems. This is usually because they are restricted by employing less accurate hardware and less sophisticated software. Due to the constraints on precision and accuracy for low-cost sensors, they cannot be fully relied upon to play an important role in applications such as attitude estimation of aircraft. But if several low-cost sensors, each with different characteristics, are used, it is possible to achieve improved accuracy by using designed fusion algorithms. In this case, each of the low-cost sensors can be used to measure the attitude of an UAV under certain limitations and deficiencies. If each sensor is independently used to estimate the attitudes of an UAV, they can fulfill the purpose. Nevertheless, some sensors might not respond fast enough, and could cause adverse data delay. Some sensors might be affected by the environmental factors and provide inconsistent measurements. Also, some sensors might give inaccurate readings if sensor errors accumulate with time. These behaviors could cause instability in the system performance and perhaps even lead to system failure. Therefore, a data fusion system which can accept all the sensor readings, compensate the flaws of every sensor, and make the necessary optimal corrections for attitude measurements would greatly enhance performance. The objective is to improve the flight performance of miniature UAV for nonmilitary low-cost applications. This section reports the author's synergistic efforts in this regard.

Attitude estimations are essential to achieve autonomous navigation for UAVs. In particular, for missions using UAVs carrying remote sensing payloads for geo-referencing purposes, accurate attitude estimation is critical. Due to weight and size limitations of miniature UAVs, it is easy for them to be affected by external disturbances, such as wind. Therefore, precise orientation data play crucial role for the controller to stabilize the system so that the autonomous flight is more safe and smooth. In order to keep the overall system cost as low as possible while accomplishing stable and accurate flight missions, to combine the available low-cost sensors by using a smart data fusion system can provide a preferred

engineering solution.

This section presents a low-cost data fusion system based upon infrared (IR) sensors, inertial sensors, and vision sensors [51]. By integrating the data from all three sensors and feeding it into the proposed weighting filter-based fusion system, it can be made a near optimal attitude estimation based on all the data comparisons. The major principal contribution of this section is to provide a practical low-cost solution for accurate attitude estimation of miniature UAVs using different inexpensive sensors.

This section is organized as follows: Subsection 4.2.2 introduces the basics of UAV attitude angles and compares three low-cost attitude estimation sensors. Then subsection 4.2.3 explains in detail the algorithms of these sensors for attitude estimation. In subsection 4.2.4, a weighting filter-based sensor fusion system is presented. Subsection 4.2.5 describes the whole system implementation and gives preliminary test results.

4.2.2 Basics of UAV Attitude Estimation

In order to achieve the desired navigation performance, attitude estimation is requisite. Attitude estimation with high fidelity will significantly improve both system stability and robustness. The most important parameters in UAV system states are the attitude angles. These are also called the Euler angles when they are considered as angular rotations regarding the body-fixed frame.

(1) Roll(ϕ): Rotation angle around the X axis of its body frame.

(2) Pitch(θ): Rotation angle around the Y axis of its body frame.

(3) Yaw(ψ): Rotation angle around the Z axis of its body frame.

The body frame is fixed to the UAV airframe, and the Euler angles rotate with respect to specified axes in the coordinate system so that the attitude of the UAV can be measured. In the present airframe design, the UAV system does not possess all the conventional control surfaces like aileron, rudder, and elevators. Two elevons are incorporated, namely, a combination of the aileron and elevator, so there is no dedicated control on yaw angle. Therefore,

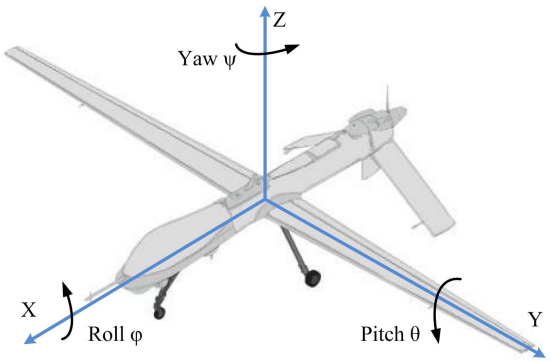


Fig. 4.8: Body frame and attitude angles.

only the roll and pitch angles will be considered. The details regarding the airframe are described subsequently.

Several types of commercial sensors have been used for UAV navigation purposes, such as inertial sensors, thermal sensors, and vision sensors. Although they are based on different working principles with their own advantages and disadvantages, each model can provide attitude measurements and assist UAV autonomous navigation.

Thermal sensors possess simple configuration with small costs [52]. The most common thermal sensor is the IR sensor. It measures the infrared radiation emitted from objects having different temperature and compares the differences from sensing surfaces. Based upon the changes of response of its internal photosensitive material, the output current and voltage of the sensor circuit also change. Through a set of algorithms used to calculate those measurements, the UAV attitude angles can be estimated.

Since the IR sensors generate analog signals, the sensor data can be sampled quickly to achieve rapid estimation updates using sophisticated algorithms. However, because IR sensors heavily rely on temperature factors which can change randomly in an outdoor area, the accuracy is questionable if sometimes the weather changes abruptly. However, if the IR sensor is calibrated properly, its performance is still valid.

Inertial sensors include gyros, accelerators, and magnetometers. Generally they are coupled to work together and comprise an inertial measurement unit (IMU). IMUs can

provide accurate attitude measurements if special configurations and filtering algorithms are used. Most IMUs available in the market are expensive and therefore, typical use is for military UAVs, commercial aircraft, and space shuttles. Consequently high quality IMUs are not usually applicable to civilian low-cost orientated projects.

With the development of relatively inexpensive inertial sensors, use for some less sophisticated algorithms requires reduced computational power. Several low-cost IMUs containing three axes gyros, accelerometers, and even magnetometers have become available [53]. These IMUs work similarly to the commercial ones, but are available at prices below one to two hundred US dollars [39]. Although the accuracy is not comparable to the industrial and commercial IMUs, they are suitable for the use of navigation missions in hobbyist and low-cost UAV projects.

For IMUs, especially the low-cost ones, the gyros can have large drift with time increase because the gyro bias is integrated over time. The attitude estimation can be inaccurate if the errors are accumulated to some extent, and there are no other ways to rectify the sensor data. Once the IMU function fails, the UAV system will be jeopardized and possibly leading to a crash.

On the other hand, vision sensors for UAV navigation have become an alternative choice. Video cameras are one of the most common vision sensors although there are other dedicated optical sensors, such as the Centeye visual microsensor [54]. Video cameras are being improved in quality and becomes more available at lower cost (less than 100 USD for high resolution). Based on advanced algorithms, video cameras can provide reasonably accurate attitude estimation. Therefore, video sensors can be used for UAV autonomous navigation or to verify the readings from other sensors.

Another advantage of a vision-based navigation system is that it does not need GPS [28], which enables the system to be operated indoor or in areas which have difficulties receiving GPS signals. A drawback of using vision sensors for attitude estimation is that they require high computational power [55]. For a low-cost UAV project, this adds an additional expense for microcomputers and other hardware. In order to keep the overall

cost as low as possible, the computational power has to be sacrificed. This means the sensor update rates cannot be very high. However, since a video camera is able to provide reliable attitude estimation, it can be combined with IR sensors and IMUs to rectify their readings and improve the navigation performance of the overall system.

After the brief analysis of the three primary low-cost sensors, it can be concluded that IR sensor and IMU are both able to perform reasonable attitude estimation, and with GPS, they can individually serve as navigation unit for UAV autonomous flight. The video camera is also able to provide attitude measurement with medium fidelity. However, limited by the cost restriction, its sensor update rates cannot ideally support UAV navigation. IR sensors and IMUs have obvious drawbacks, such as sensor drifting and effects of environmental factors. However, based on the smart data fusion system proposed in this paper, the adverse features of each sensor can be overcome to achieve the most accurate measurements from their combinations. A comparison of the sensors introduced above is summarized in Table 4.1. In order to clarify the definition of low cost, Table 4.2 shows a classification of IMUs in terms of price [39]. The low-cost standard defined in this paper is attributed to the Hobbyist Grade and it is less than 200 USD.

4.2.3 Sensor Altitude Estimation Algorithms

IR Sensor

The details of implementing IR sensors for attitude estimations were presented in the previous section. In order to compare with other navigation sensors, major information of IR sensors for UAV attitude estimation is included.

The raw data of the roll loop (r), pitch loop (p), and top (t) are defined by the following equations [14]:

$$r = k_{la} \times (s_1 + s_2), \tag{4.10}$$

$$p = k_{ln} \times (s_2 - s_1), \tag{4.11}$$

$$t = k_v \times s_3, \tag{4.12}$$

Table 4.1: Sensor general comparisons.

IMU	IR sensor	Video camera
Low cost	Low cost	Low cost
Fast sensor updates	Fast sensor updates	Slow sensor updates
Most accurate	Least accurate	Medium accuracy
Sensor drift	Rely on temperature	Require computation power

Table 4.2: IMU categories.

IMU Type	Cost(USD)	Example
Navigation Grade	>50k	Honeywell HG9848
Tactical Grade	10-20k	Honeywell HG1900
Industrial Grade	0.5-3k	Microstrain Gx2
Hobbyist Grade	<500	Ardu IMU

where s_1 is the value calculated from the channel 1 reading, s_2 is the value calculated from channel 2, and s_3 is the value calculated from channel 3.

Once the roll loop, pitch loop, and top raw data are obtained, the roll and pitch angles can be calculated using the following equations [14]:

$$\phi = \tan^{-1}(\frac{r}{t}), \tag{4.13}$$

$$\theta = \tan^{-1}(\frac{p}{t}). \tag{4.14}$$

Because the installations might have misalignments, which cause discrepancies on the final roll and pitch calculations, it maybe necessary to add neutral correction parameters to compensate for these differences. However, in this case, it is assumed that the IR sensors were perfectly installed so that all the neutral values can be considered as zero.

IMU

A detail study of representative state estimation algorithms surveyed several IMU software sensor fusion algorithms [39], including General Extended Kalman Filter, Quaternion-Based Extended Kalman Filter, Euler Angle-Based Extended Kalman Filter, AggieEKF,

a GPS-Aided Extended Kalman Filter, and Complementary Filters. Since most of these algorithms, especially the various Kalman Filters, are developed for inertial sensors of high fidelity, they may require excessive computational power. For low-cost inertial sensors, in order to maintain the balance between accuracy and power consumption, the complementary filter has been found to be a suitable solution [56].

Whereas most IMUs incorporate accelerometers and gyroscopes, some also include magnetometers, GPS and pressure sensors to achieve accurate estimations. Accelerometers are used to measure linear accelerations and they can directly be used to measure attitude of an object because they can measure the gravity if there is no mechanical acceleration. Gyroscopes are used to measure the angular velocity around certain axes in the UAV body frame and after integration to estimate the angles. Magnetometers can provide measurements of the heading angles and they are also able to compensate gyro bias. GPS can provide information about absolute position, altitude, velocity, and course angle of an object.

The IMU used in this project has integrated three-axis accelerometers and three-axis gyroscopes, and there is also a port for an external GPS receiver. In order to calculate the attitude angles, a direction cosine matrix (DCM) complementary filter was designed and implemented [30].

The direction, velocity and acceleration vectors can be transformed to be a 3×3 matrix between the UAV body frame and earth reference frame. The rotation vectors can be multiplied by a direction cosine matrix as follows [30]:

$$Q = \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix}, \tag{4.15}$$

$$R = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix}. \tag{4.16}$$

The random vector Q represents velocity, direction, acceleration, and R is the rotation matrix.

$$Q_E = RQ_U \tag{4.17}$$

The vector Q_E is the earth reference frame and Q_U is a vector for the UAV body frame. In order to interpret the relationship between the rotation matrix and attitude angles, the rotation matrices with Euler angles ($\phi\ \theta\ \psi$) around X, Y, and Z axis have to be found,

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \tag{4.18}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{4.19}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.20}$$

The rotation matrix R can be obtained through the multiplication of the three matrices as follows:

$$R = R_x(\phi)R_y(\theta)R_z(\psi). \tag{4.21}$$

After R is found, certain components in this Euler angles based rotation matrix can be selected to calculate the UAV attitude angles (roll, pitch, and yaw). The calculations are

expressed by the following equations:

$$\phi = \tan^{-1}(\frac{R_{32}}{R_{33}}) = \tan^{-1}(\frac{\sin\phi\cos\theta}{\cos\phi\cos\theta}) = \tan^{-1}(\tan\phi), \quad (4.22)$$

$$\theta = \tan^{-1}(R_{31}) = \sin^{-1}(\sin\theta), \quad (4.23)$$

$$\psi = \tan^{-1}(\frac{R_{21}}{R_{11}}) = \tan^{-1}(\frac{\cos\theta\sin\psi}{\cos\theta\cos\psi}) = \tan^{-1}(\tan\psi). \quad (4.24)$$

Video Camera

The idea of determining aircraft attitude angles through a camera is not new. Several research groups already investigated various methods, some use image processing [57, 58]; others have also tried machine learning techniques [59–61].

In terms of robustness against varying environmental conditions (*e.g.* due to clouds, fog, lighting, and uneven horizons), machine learning based approaches appear to be superior to those only relying on image processing.

The Visual Attitude Estimation approach here [62] therefore uses a Decision Tree to classify sky and ground and thus detect the horizon in images taken by a forward-looking camera mounted on the UAV. In order to build this Decision Tree, it first has to be trained by manually classifying several training images. The classifier can be made more robust against weather and lighting conditions by using a greater diversity of training images. Researches [59, 60] show that machine learning techniques can reach an accuracy of well above 90% by sufficient classifier training.

After classifying sky and ground and having performed image preprocessing, all required horizon lines are found by running the Hough Transform edge detection algorithm on the emerging binary image. In order to find the correct horizon line from all candidates, the correctness of each candidate is evaluated on the classified image, taking the best fitting one as the resulting horizon line [60].

Finally, the aircraft's attitude can be calculated depending on the horizon line and camera parameters [57]. As an aircraft moving parallel with earth surface, only turning in left or right directions, does not observe a change of horizon, it is not possible to ascertain

the yaw (left and right turning) angle. The results computed by this framework are the UAV's current roll and pitch angles.

Compared with existing methods, the major advantage of the approach introduced above is that this working system is achievable even under many restrictions. There are not many known projects dealing with such a low-cost, but effective, method for vision-based attitude estimations with applications to UAVs.

The aircraft's roll angle (output in radians) can be calculated as follows [57]:

$$\phi \quad = \quad \tan^{-1}(m_{line}), \tag{4.25}$$

where m_{line} is the slope of the horizon line on the image plane which can easily be calculated from the horizon line's representation:

$$m_{line} \quad = \quad \frac{h_{left} - h_{right}}{\omega_{img}}. \tag{4.26}$$

This approach takes h_{left} and h_{right} as the y values for the horizon line as the left and right image borders and ω_{img} as the image's width.

The pitch angle can also be determined, although the method is not as intuitive:

$$\theta \quad = \quad \tan^{-1}(\frac{u \sin\phi + v \cos\phi}{f}). \tag{4.27}$$

While f denotes the focal length, u and v are the metric positions of an arbitrary pixel $P(x/y)$ on the horizon line as it is depicted on the image plane (the 2D representation of the camera image located inside the camera).

Values for u and v have to be calculated from the image's pixel coordinates. As the reference point, the middle of the horizon line with $x = (\frac{\omega_{img}}{2})$ and $y = h_{line} - \frac{h_{img}}{2}$ is used with h_{line} denoting the value of the middle of the horizon line, and h_{img} denoting the

image's height.

$$u = \omega_p \cdot x = \omega_p \cdot \frac{\omega_{img}}{2}, \tag{4.28}$$

$$v = h_p \cdot y = h_p \cdot (h_{line} - \frac{h_{img}}{2}). \tag{4.29}$$

where h_p and ω_p are pixel height and width in mm, respectively, and h_p and ω_p represent the actual height and width of a pixel on the camera's image sensor.

Hence, Equation (4.27) can be modified as follows:

$$\theta = \tan^{-1}(\frac{\omega_p \cdot \frac{\omega_{img}}{2} \sin\phi + h_p \cdot (h_{line} - \frac{h_{img}}{2}) \cos\phi}{f}). \tag{4.30}$$

A set of sample test images with classifications is shown in Figure 4.9.

4.2.4 Low-cost Data Fusion System

Once the attitude estimation algorithms of all three sensors are well established and continuous input data are obtained from these sensors, a weighting filter-based data fusion system can be designed to achieve near-optimal attitude estimations. Motivated by other researchers, making the sensors parley with each other becomes a rational approach [63, 64]. This means comparing all the sensor outputs and drawing a consensus about which combination will generate a better estimation. Instead of categorizing each sensor output by its frequency range, the weighting emphasizes on the sensor output differences. The basic purpose of this filter is to use the camera and IMU estimations while the UAV is flying in small movements, such as flying straight and flat, with no rapid attitude angle changes. This ensures that the slow update rate of the camera will not affect the accuracy. When the UAV performs rapid turning, descending or ascending, the filter adopts the IMU and IR sensor estimations for fast sensor updates and required accuracy.

Let the output of the IMU be modeled as $S_1 = \binom{\phi_1}{\theta_1}$, the output of the IR sensors as $S_2 = \binom{\phi_2}{\theta_2}$, the output of the camera as $S_3 = \binom{\phi_3}{\theta_3}$. In the first stage of the filter, since the updating rates of IMU and IR sensors are much closer to each other, S_1 and S_2 are

Fig. 4.9: Test images.

compared first, and the IMU outputs are used as the current reference:

As introduced before, IR sensors need calibration before they can measure attitude angles correctly. Therefore, before the S_d goes below 5^o for either roll or pitch, the tunings of the IR sensors need to be continued. However, the IR sensor calibration is not necessary at the beginning of every flight test. When a new system is established, the IMU can be used to tune the IR sensors, and they can then provide similar performance. Another condition for recalibration of IR sensors is when the weather changes significantly, such as from summer to winter. The IR sensors can be sampled as fast as the low-cost IMU so there is no conflict with synchronizing both data.

After IR sensor output S_2 is close to S_1, the system can move to the second stage of the filter, which compares S_1 and S_3. As explained previously, the camera update rate is much slower than the IMU update rate, so it is necessary to follow the updating frequency of the camera output first and then check whether the angles reporting from the IMU and the camera are close to each other. Then S_1, S_2, and S_3 can be sent to the third stage of the filter.

Because of the dynamics of the current airframe configuration and the physical mounting of the navigation unit, such as alignment between the IMU and camera, and also based on actual flight experience, the weightings for roll channel and pitch channel are different. Therefore, two different rules are applied in the next stage to establish how to generate the

near-optimal roll and pitch angles.

In the third stage of the filter, $S_1(S_{\phi 1}, S_{\theta 1})$, $S_2(S_{\phi 2}, S_{\theta 2})$, and $S_3(S_{\phi 3}, S_{\theta 3})$ are taken and then used in the following procedures to classify inputs to obtain the near-optimal final output $S_o(S_\phi, S_\theta)$:

$$S_\phi = \alpha_1 S_{\phi 1} + \beta_1 S_{\phi 2} + (1 - \alpha_1 - \beta_1) S_{\phi 3}, \tag{4.31}$$

$$S_\theta = \alpha_2 S_{\theta 1} + \beta_2 S_{\theta 2} + (1 - \alpha_2 - \beta_2) S_{\theta 3}. \tag{4.32}$$

For the *roll channel*:

(1) Find $S_{\phi 1}$ and compare with $S_{\phi 3}$,

$$S_{c\phi} = k_\phi |S_{\phi 1} - S_{\phi 3}|, \tag{4.33}$$

where $S_{c\phi}$ is the weighting factor for the roll channel, k_ϕ is the roll tuning gain, and $k_\phi \in (0,2)$is defined such that this criterion can be applied for the roll angle estimation under different circumstances. In the following procedures, k_ϕ is chosen as unity.

(2) If $S_{c\phi}$ keeps fluctuating and above 10^o, it means the UAV is under rapid roll movements, so $\alpha_1 = \frac{3}{4}$, $\beta_1 = \frac{1}{4}$, and $S_\phi = \frac{3}{4} S_{\phi 1} + \frac{1}{4} S_{\phi 2}$.

(3) If $S_{c\phi}$ is close to steady state and below 5^o, it means the UAV is under small roll movements, so $\alpha_1 = \frac{3}{4}$, $\beta_1 = 0$, and $S_\phi = \frac{3}{4} S_{\phi 1} + \frac{1}{4} S_{\phi 3}$.

(4) In the third circumstance, when $5^o < S_{c\phi} < 10^o$, $\alpha_1 = \frac{1}{2}$, $\beta_1 = \frac{1}{4}$, $S_\phi = \frac{1}{2} S_{\phi 1} + \frac{1}{4}(S_{\phi 2} + S_{\phi 3})$.

For the *pitch channel*:

(1) Find $S_{\theta 1}$ and compare with $S_{\theta 3}$,

$$S_{c\theta} = k_\theta |S_{\theta 1} - S_{\theta 3}|, \tag{4.34}$$

where $S_{c\theta}$ is the weighting factor for the pitch channel, k_θ is the pitch tuning gain, and $k_\theta \in (0,2)$ is defined such that this criterion can be applied for the pitch angle estimation under different circumstances. In the following procedures, k_θ is chosen as 1.

(2) If $S_{c\theta}$ keeps fluctuating and above 15^o, it means the UAV is under rapid pitch movements, so $\alpha_2=\frac{3}{4}$, $\beta_2=\frac{1}{4}$, and $S_\theta=\frac{3}{4}S_{\theta 1}+\frac{1}{4}S_{\theta 2}$.

(3) If $S_{c\theta}$ is close to steady state and below 10^o, it means the UAV is under small pitch movements, so $\alpha_2=\frac{3}{4}$, $\beta_2=0$, and $S_\theta=\frac{3}{4}S_{\theta 1}+\frac{1}{4}S_{\theta 3}$.

(4) In the third circumstance, when $10^o<S_{c\theta}<15^o$, $\alpha_2=\frac{1}{2}$, $\beta_2=\frac{1}{4}$, $S_\theta=\frac{1}{2}S_{\theta 1}+\frac{1}{4}(S_{\theta 2}+S_{\theta 3})$.

The block diagram of the fusion system is shown in Figure 4.10.

When there is a wind gust, definitely only the IMU and IR sensors can reflect the sudden disturbance, which is why both IMU and IR estimations are combined while the UAV is under large fluctuations. When the IMU begins reporting erroneous data under large movements, the IR sensors can compensate that behavior. Under small movements, both IR sensor and video camera can compensate the errors. Because of the flying-wing airframe configuration, most of the time its roll performance is better than its pitch performance, which is why the weighting range for the roll channel is smaller than the range of the pitch channel. The selection of 5 degrees as the current weighting factor is conservative because the miniature UAV is not tested under inclement weather conditions, defined as when the wind speed prediction is above 10 miles/h (4.47 m/s = 8.689 knots). This limit is predicted on the size and weight of the UAV plus flight safety restrictions. However, this weighting factor is revisable and can be adjusted using the tuning gains to accommodate different models of aircraft.

4.2.5 System Implementation and Test Results

The complete system is based on the open-source Paparazzi autopilot and a UAV designed by the authors. The UAV platform with a single navigation unit (IMU or IR sensors)

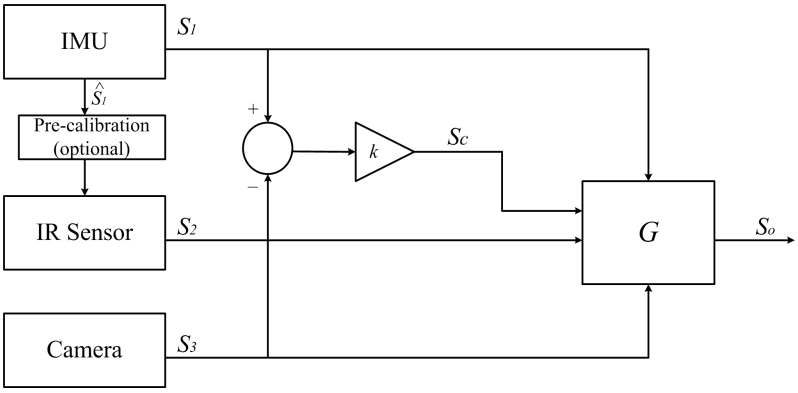


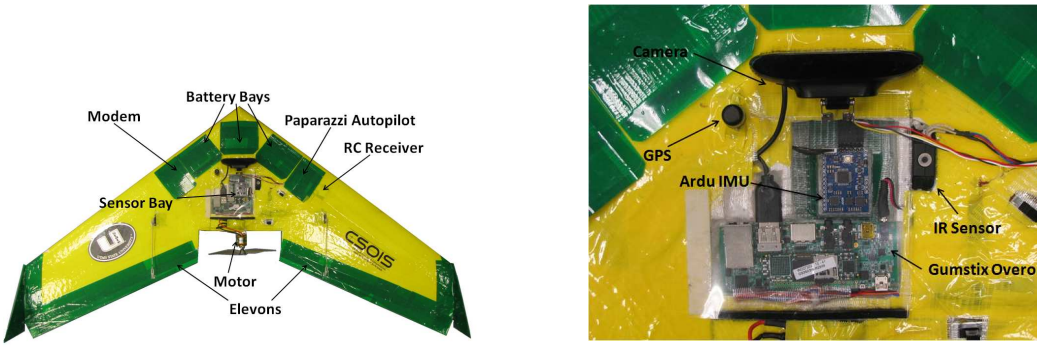
Fig. 4.10: Data fusion system block diagram.

has been tested multiple times and achieved reasonable autonomous flight performance. The camera unit has also been tested on the ground and obtained satisfactory classification accuracy. The integration of all sensors has been completed and all real-time sensor data have been through several ground tests. The original and filtered data with a commercial IMU were also compared worth an estimated five times the value of current low-cost sensor system. This comparison shows the effectiveness of the weighting filter. Finally the fusion system was implemented on the autopilot and verified in actual flights.

The UAV platform described is a totally self-made delta flying wing originally designed for RC purpose. After equipping it with an autopilot, a navigation unit and communication devices, the UAV is able to perform stable autonomous flights. The UAV layout and sensor bay of the 48-inch UAV is shown in Figure 4.11(a) and Figure 4.11(b), respectively.

The IR sensor was made by FMA Direct [47]. It was originally part of a Co-Pilot Flight Stabilization System for guiding trainees to practice flying RC planes. It has a field of view of $\pm 45^\circ$ and two sensor channels [52]. As illustrated previously, there are two IR sensors installed on the UAV, and the outputs of IR sensors are directed to the autopilot through two analog-to-digital converter (ADC) ports.

The implemented IMU is called Ardu and was originally designed by DIYDRONE [29], a website for fans of autonomous systems. The IMU costs only 100 US dollars, which is equivalent to the cost of IR sensors. A 3-axis accelerometer and a 3-axis gyroscope are



(a) 48-inch UAV carrying three sensors.

(b) Sensor bay layout.

Fig. 4.11: Sensor fusion platform.

integrated in the IMU and it runs a DCM filter to calculate the attitudes. A uBlox GPS receiver is directly connected to the Ardu IMU, and the DCM filter takes GPS data to correct the yaw drift. Then Ardu IMU sends both attitude angles and GPS data to the autopilot through a UART port.

The video camera used is a Logitech HD Pro Webcam C910 [65]. It can provide full HD 1080p video recording with a resolution of 10 megapixels. The camera is connected to a Gumstix Overo single-board computer through a USB connection which processes every image taken from the webcam. The computer calculates the attitude angles based on the installed algorithm software. The Overo sends the information to the autopilot through an I^2C bus.

Regarding the visual attitude estimation part for autonomous navigation, the framework is still under development. The horizon classification framework consists of three independent parts, where Java is used as one of them for software engineering reasons, such as easy reusability of existing tools which provide more consistent procedures when adding new features. The other parts are implemented in C++, so the data are transferred via a remote procedure called (RPC) framework. Therefore, the image data are transmitted among the three parts and needs to be converted before and after the transmission. This makes the framework unnecessarily slow. These performance issues on the current implementa-

tion prevent productive autonomous flights. Apart from this, the visual attitude estimation framework was working during the ground test and satisfactory results were achieved [62]. The UAVs will be flown as soon as the software exits beta state. The current hardware block diagram is shown in Figure 4.12, and Table 4.3 compares the three sensors in several different attributes.

Some preliminary ground test results for roll and pitch angle comparisons are shown in Figures 4.13 and 4.14, respectively. The sensor data were collected with the help of Paparazzi ground station logging function. Due to weather conditions and human operation interference, the IR sensors were not absolutely consistent, but most of the time they track the Ardu IMU closely. Because of hardware limitations, the current microcomputer cannot provide sufficient computational power, and the I2C bus causes transmission delay. Therefore, the original camera estimation has some lagging issues, but unlike the IR sensors it does not critically rely on environmental factors and thus can provide more consistent measurements. A Mircrostrain GX2 IMU is used as a reference comparison. It costs about $1695 and gives 0.5g accuracy under static test conditions [40]. Because the IMU has been implemented into the current Paparazzi autopilot system, numerous flight experiences have been obtained and it has demonstrated its reliability and accuracy.

As shown in Figures 4.13(c) and 4.13(d), the attitude estimation differences in the roll channel from different sensors are within 5 degrees most of the time, but sometimes the IR sensors provide noisy readings due to inconsistency. The filter is able to effectively adopt the near-optimal combinations among all the sensors and give much smoother estimations. Most of the time the accuracy of the filtered data is better than any other three sensors and its curve is closer to the reference reading from Microstrain GX2 IMU, which shows the effectiveness of the proposed weighting filter.

As shown in Figure 4.14, the attitude estimation in the pitch channel presents similar behavior as in the roll channel. However, because of disturbances from human operation (raising and lowering the UAV head) the IR sensors can provide noisy readings. The camera also has a boundary limit of measurement because the pitch up angles cannot be too

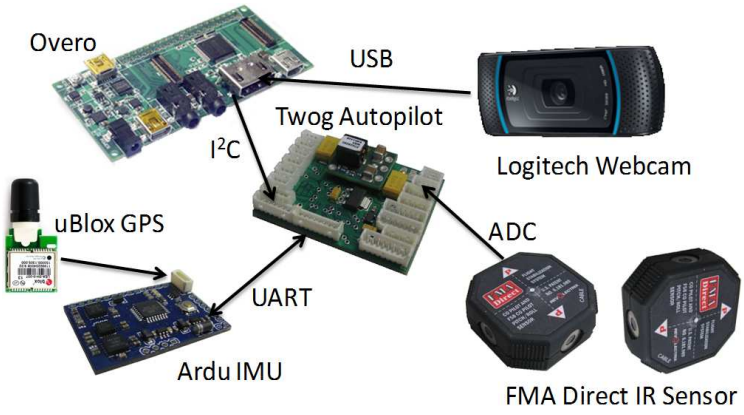


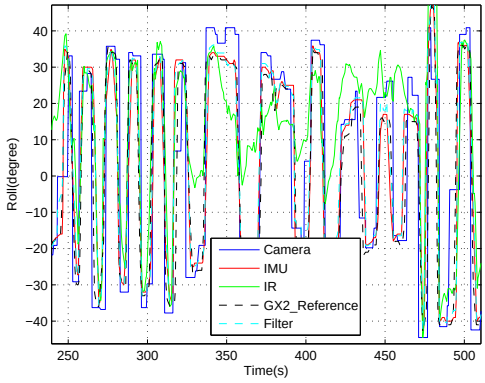
Fig. 4.12: Hardware block diagram.

Table 4.3: Sensor comparisons.

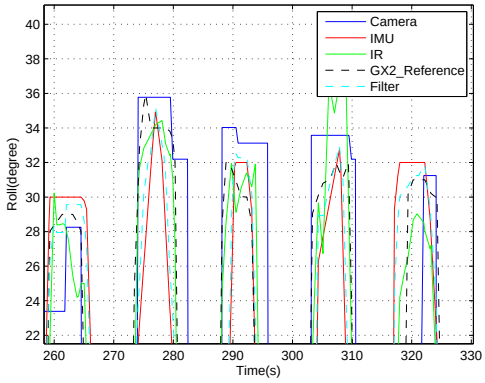
Sensor Type	Weight (oz)	Power Consumption	Updating Frequency	Price($)
Ardu IMU	0.2	<100mW	50Hz	100
FMA Direct IR Sensor ×2	0.78	<60mW	60Hz	80
Logitech Webcam	13.6	<1W	<1Hz	100

high. Otherwise the ground disappears from the camera image, and this completely disables the horizon classification. The filter is still able to effectively adopt the near-optimal combination of all three sensors, avoid the adverse effect from each sensor and give smooth estimations. The filtered data are closer to the reference reading most of the time.

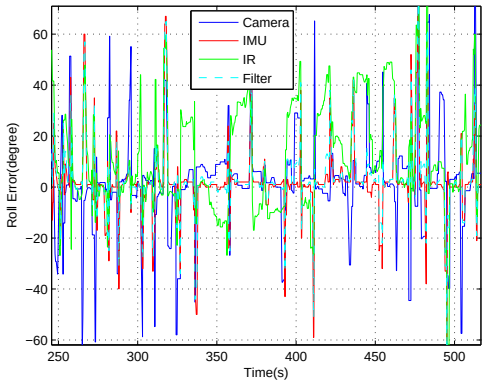
Some preliminary autonomous flight test results are shown in Figures 4.15 and 4.16. Due to the reason explained before, the vision-based attitude estimation was not involved in the flight test. Both IMU and IR sensor data are sampled at the same frequency, and a reference attitude generated by the autopilot software was used to evaluate the filtered data compared with the original sensor data. Since the filtered angle data is based on the IMU and IR sensor, the distributions of the weighting factors are slightly different from the ground test. The original weighting for the camera is now equally assigned to the other two sensors since the relative accuracy is between the IMU and IR sensors. The roll angle



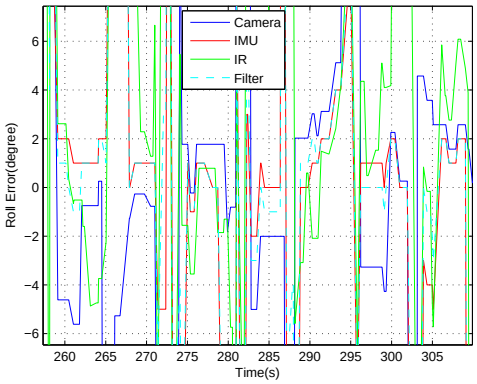
(a) Roll angle comparisons.



(b) Zoom-in roll angle comparisons.



(c) Roll angle error comparisons.

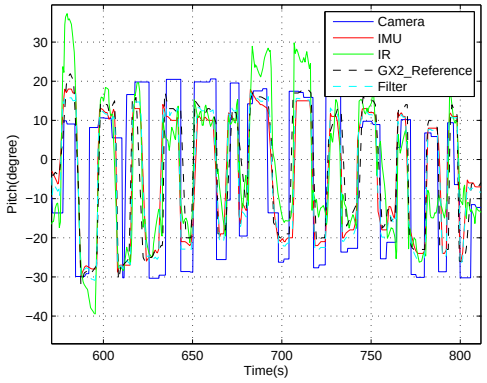


(d) Zoom-in roll angle error comparisons.

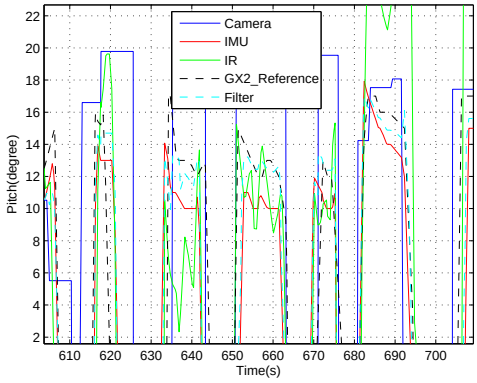
Fig. 4.13: Roll channel ground comparisons.

estimations from the two sensors are close, and they both track the reference angle within an acceptable range. The IMU estimation is more accurate than that from the IR sensors, especially for larger angles (when ϕ is bigger than 15^o). However, most of the time the IMU generates more overshoot with respect to the reference angle. Using the proposed weighting filter, the IR sensor can compensate for this behavior regarding its smaller angle estimation. At smaller angles ($\phi \leq 5^o$), the IMU estimation is still slightly more accurate than the IR sensor estimation, so all the weightings are attributed to the IMU.

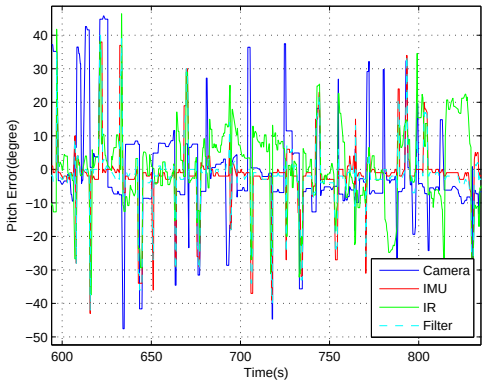
The filtered data of the roll angle estimation error have shown advantages over the



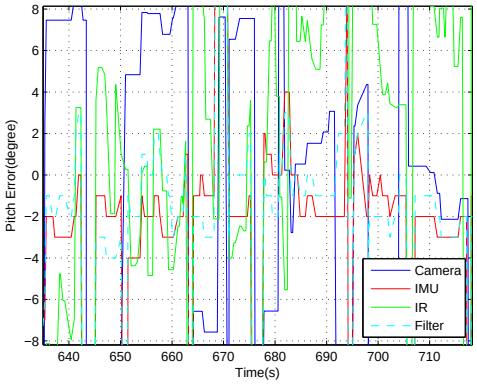
(a) Pitch angle comparisons.



(b) Zoom-in pitch angle comparisons.



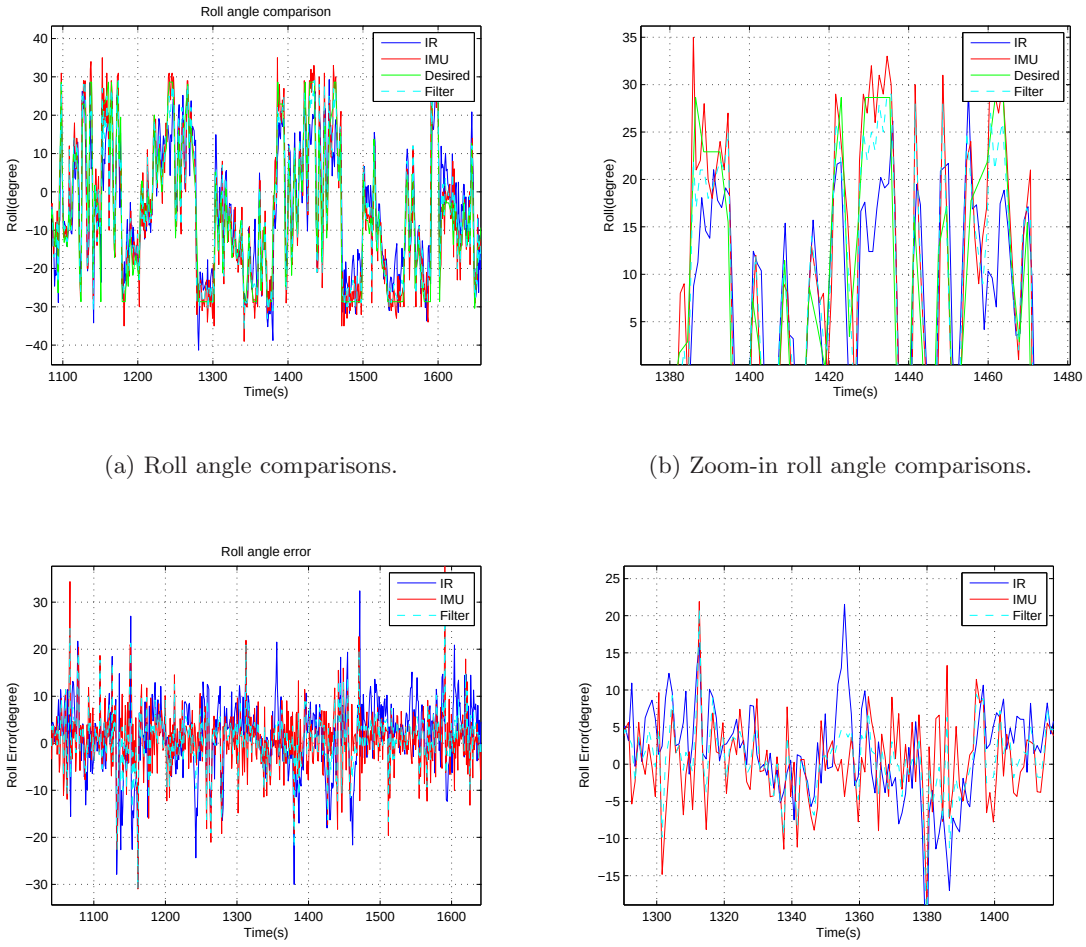
(c) Pitch angle error comparisons.



(d) Zoom-in pitch angle error comparisons.

Fig. 4.14: Pitch channel ground comparisons.

individual sensor data compared with the reference attitude angle. The results are explicitly shown in Figures 4.15(c) and 4.15(d), respectively. In Figure 4.16, the pitch angle estimations are similar to the roll angle estimations. Both the IMU and IR sensor provide similar measurements even for bigger angles (when θ is bigger than 15^o). Sometimes the IR sensor has more overshoot than the IMU and vice versa. However, most of the time they can both track the reference angle closely. At smaller angle estimations ($\theta \leq 10^o$), the IR sensor is nosier than the IMU, so the IMU requires heavier weighting, which also includes the weighting factor from the camera. At larger angles ($\theta \geq 15^o$), the measurements from



(a) Roll angle comparisons.

(b) Zoom-in roll angle comparisons.

(c) Roll angle error comparisons.

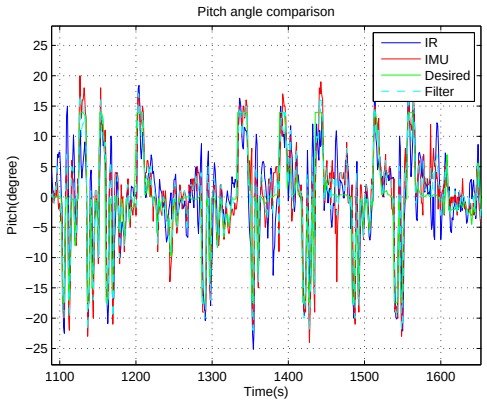
(d) Zoom-in roll angle error comparisons.

Fig. 4.15: Roll channel flight comparisons.

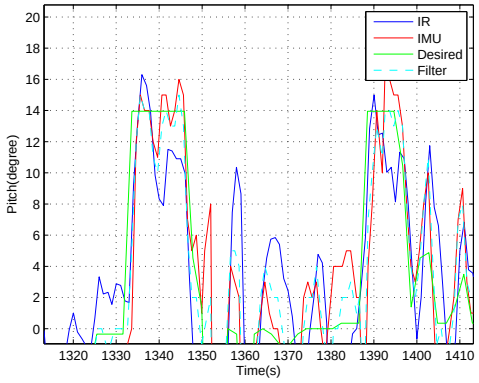
IR sensor and IMUs are combined to provide the most complementary estimation, which is closer to the reference angles. Some detailed comparisons of the filtered data error versus the original sensor errors are shown in Figures 4.16(c) and 4.16(d), respectively, which verifies the advantage of the proposed weighting filter.

4.3 Chapter Summary

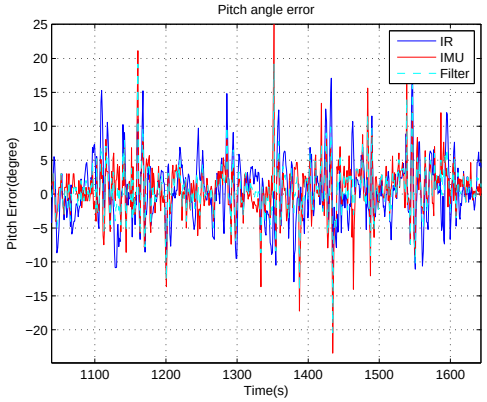
In the first part of this chapter, a two-stage IR sensor calibration method for the attitude estimation of a low-cost UAV was proposed and demonstrated based upon experienced



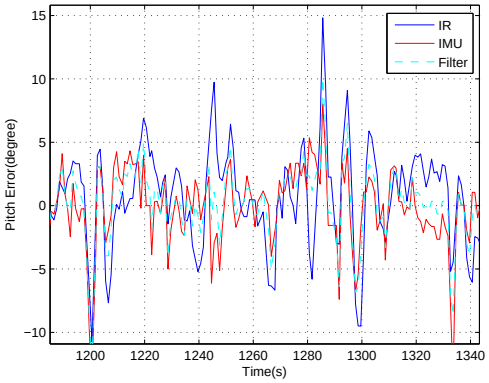
(a) Pitch angle comparisons.



(b) Zoom-in pitch angle comparisons.



(c) Pitch angle error comparisons.



(d) Zoom-in pitch angle error comparisons.

Fig. 4.16: Pitch channel flight comparisons.

ground plus flight tuning procedures, and comparisons between IR sensor and IMU data collected from a actual flight test.

In the second part of this chapter, a practical low-cost data fusion system based on inexpensive IR sensors, IMU and video camera is presented. The basic engineering approach for the fusion system was to integrate all three sensors' data and then delivering the result to an autopilot which runs the weighting filter based algorithms. The filter generates combined near-optimal attitude estimations based upon the sensor inputs according to appropriate types of UAV movements. The characteristics of each low-cost sensors used were compared

and analyzed to show the advantages and drawbacks of each sensor. The attitude estimation algorithms applied on all three sensors were studied individually. Some preliminary ground test results and comparisons with a commercial IMU are provided to show the effectiveness of the proposed weighting filter. In additions, some flight test results are presented to verify the filter performance.

Future work includes reimplementation of the software for the vision attitude estimation so the camera's updating frequency is sufficient for independent autonomous flight and synchronization with the other two measurement resources. The plan is to implement the proposed fusion system with all three low-cost sensors for UAV autonomous navigation and to show the advantages of the sensor fusion system in practical applications.

Chapter 5

Cooperative Multiple UAV Formation Flight

5.1 Introduction

Research on unmanned aerial vehicles (UAVs) has become important because of their potential advantages in various important applications including reconnaissance, surveillance, combat, etc. Multi-UAV formation flight is gaining numerous attentions because it involves multiple agent control, cooperative path planning, obstacle avoidance, communication topology plus time delay, UAV platform design and implementations. In addition to the level of complexity for multi-UAV formation flight, the level of beneficial applications is also increasing. Compared with a single or multiple isolated UAV system, a cooperative UAV system provides efficiency in terms of mission time and cost, increased resilience and safety against a single point of failure. Every agent is able to distribute the information to the other agents, and the probability of mission success despite the risk of any agent malfunctioning during the flight is enhanced.

Many researchers have performed formation flights based upon various UAV platforms and different approaches. The DragonFly UAV program at Stanford University is one of the earliest groups performing in multiple UAV research, and one of their papers presented automated multiple UAV flight with a game theory approach to improve UAV guidance and solve air traffic problems [66]. Other researchers introduced a UAV testbed and demonstrated a two UAV formation flight with autonomous rendezvous using timing control [67], presented a quadrotor-based testbed and multiple rotary craft flight with a finite-state machine (FSM) encoding a hybrid system approach [68], and performed formation flights using two unmanned helicopters in real-time with other simulated helicopters to study mesh stability [69]. Researchers from Brigham Young University described its multiple UAV testbed

development [21] and showed a formation flight with three UAVs under consensus seeking strategy [7]. Researchers from National University of Singapore presented a control system design based on the leader-follower pattern for formation fights using unmanned helicopters [70]. Compared with other researchers' approaches, the present approach is based upon a centralized configuration using several low-cost fixed-wing UAVs. One UAV acts as the leader and the others follow its trajectory so specific formation flight can be achieved. This approach is also under the development towards decentralized coordination.

With the development of low-cost avionics, airframe materials, and electrical power systems, UAVs have become affordable for civilian applications [12]. There has been an increasing demand of using UAVs in disaster control, search and rescue, water management [6], agricultural monitoring, and mineral exploration, etc. Based on the advantage of a cooperative UAV system, multi-UAV formation flight can be used to enhance the capability of the AggieAir platform [12] so a group of the AggieAir personal remote sensing systems can better fulfill civilian application oriented scenarios, such as wind profile measurement [71] and to maximally improve overall stability and efficiency.

There are several approaches to design a distributed controller for multi-UAV formation flight. The most common one is a leader-follower strategy [72]. In this approach, there is usually one leader with multiple followers; however sometimes there can be multiple agents serving as leaders. While the leader flies according to the preprogrammed flight path, the followers attempt to track the reference waypoints relative to that of the leader's to maintain the formation scheme. The advantage of this strategy is that its algorithms and control architecture are simpler to implement than by other approaches. Reliance on the communication between the ground station and UAVs is imperative. This means that transmission delays or interruptions can lead to control system malfunctions, and leader UAV failure can disrupt the whole system. The virtual structure method is another approach for multi-UAV formation flight [72, 73]. It is different from the leader-follower strategy because there is no hierarchy among all the agents and the whole group is treated as a rigid formation. In this approach, once a desired structure of the whole group is defined,

all the agents will follow the command generated from the virtual structure for specified motion and orientation so that the rigid formation can be maintained. The advantage of this approach is that it is simple to define the group behavior given a desired structure [73]. The drawback is that the formation shapes cannot be modified with the time varying and one agent failure can jeopardize the whole group. Other approaches, such as behavior-based formation control [74], have also been studied, nevertheless, for practical multi-UAV formation flight, leader-follower strategy is judged to be the engineering method against which other approaches are compared.

5.2 Leader-follower Formation Flight

5.2.1 Control Structure

The current multi-UAV formation control structure is based on Paparazzi software. Paparazzi is an open-source autopilot system designed by Pascal Brisset et al. from ENAC in France [14]. Its airborne code and ground station software are fully accessible and there are various hardware options developed by the community. After years of evolution, it has become a mature platform for both applications and research. Based on the existing platform, many new functions and features have been designed and implemented by the USU team so the overall system is flexible and robust [12]. Besides its sophisticated flight control system for a single UAV, the Paparazzi software has some basic functions to accommodate multiple UAV formation flight [13]. These include setting different flight plans for the leader and followers so their roles are distinguished and they can always stay in individual flight mode until the formation commands are sent. There is also a traffic collision avoidance system (TCAS) built in. When the UAVs are too close to one another, TCAS will be activated and some of them will go up while the others will go down to avoid collisions, which significantly increases the safety during formation flights.

For the purpose of manipulating several agents and driving them to achieve some desired mission in a certain formation, a distributed architecture is required. The distributed architecture of Paparazzi is described as follows and it is illustrated in Figure 5.1:

(1) Simultaneous operation of multiple agents is permitted;

(2) Airborne network: It is possible for all agents to share information through an internal network while they can directly communicate with ground control station (GCS);

(3) Ground network: The configuration and status of the agents can be shared through messages between GCS and other wireless devices;

(4) GCS can monitor the status of the agents and control their behavior through datalink.

In order to accomplish the basic formation flight, an initializing function first needs to be added into the flight plans. This function takes the unique aircraft ID and its x, y, and z coordinates whenever a new aircraft is added into the formation scheme. Then it defines which ID will be the leader UAV. Afterwards, it offers an option to convert the coordinates of the followers to be either based upon global coordinates or relative to the leader UAV. Usually, it is more meaningful to use the relative coordinate system so the followers' orientation will always trace the leader's to give the similar headings. In the leader's flight plan, there can be several flight paths including waypoint navigation or standby cycling with a pre-function to activate the formation flight mode. Once the leader UAV starts formation flight, the followers can decide to either activate the formation mode by switching the flight plan block or remain in individual mode. When both are in the formation mode, the follower will track the virtual waypoints generated according to the leader's position and the relative coordinates defined in the initializing function. Speed control is achieved by adjusting the throttle setpoint based on the difference between the followers' actual coordinates and the desired coordinates. When there is no need to continue formation flight, the leader and followers can simply switch their flight plan blocks and return to individual mode.

The control structure for the current formation controller is centralized [75]. This means that the global information is available on the GCS, but mission planning and UAV team coordination rely upon a single network. This can lead to issues such as constricted communication range, system delay, and intermission. During operation, the flight status of each UAV is transmitted back to the GCS. Then the formation control module processes

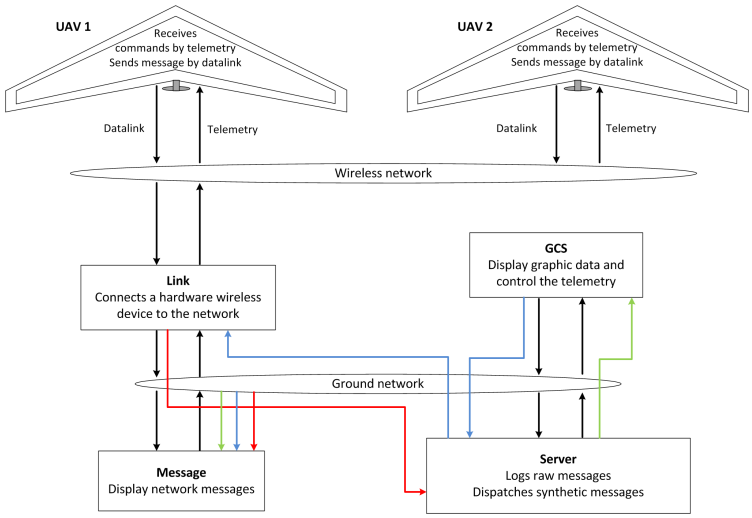


Fig. 5.1: Paparazzi distributed architecture.

the information from each agent and compares them with the predefined settings in the initial configuration. Afterwards, the module generates control inputs for the followers so they can track the leader's trajectory and maintain the formation configuration. When all the UAVs fly normally and the controller works properly, the GCS is used to monitor the UAVs, tune the controller, change the flight path, and broadcast the formation commands back to the UAVs. However, if an emergency occurs, the GCS is able to terminate the formation flight and control each agent separately so the hazards will not jeopardize the whole system. The current formation control structure is shown in Figure 5.2.

5.2.2 Controller Tuning

After the development of a stable low-cost UAV testbed was completed, formation controller tuning was the next most critical part prior to experimental flights. In order to achieve reliable performance, the structure of the multi-agent formation controller was analyzed and key parameters regarding different control states were introduced [14] as follows:

(1) Position proportional gain (k_p),

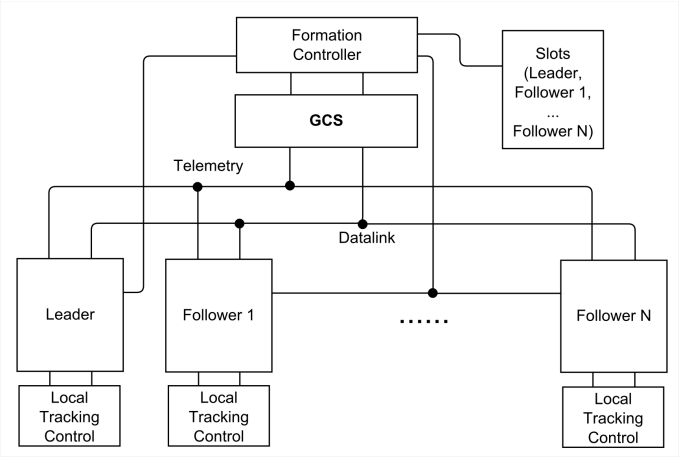


Fig. 5.2: Centralized configuration.

(2) Speed proportional gain (k_s),

(3) Course proportional gain (k_c),

(4) Altitude proportional gain (k_a).

The course proportional gain (k_c) is an optional parameter because the dedicated heading control is not meaningful for the entire formation controller. The other three parameters are used to tune the corresponding control loops and they can also be adjusted in the GCS, which enable the real-time controller tunings.

In this chapter one leader UAV and one follower UAV are selected as an example to explain the control algorithms. For additional followers, similar formulas are used for calculation of the control inputs. In the coordinate frame of the controller, the position of the leader is considered as (0, 0, 0) in longitude (X axis), latitude (Y axis), and altitude (Z axis). The followers' positions are set relative to the position of the leader. The leader's position and states are defined as:

(1) Leader longitude (x_l),

(2) Leader latitude (y_l),

(3) Leader altitude (z_l),

(4) Leader course (χ_l),

(5) Leader airspeed (v_l).

Similar to the leader parameters, the follower's instantaneous position and states are defined as:

(1) Follower longitude (x_f),

(2) Follower latitude (y_f),

(3) Follower altitude (z_f),

(4) Leader course (χ_f),

(5) Leader airspeed (v_f).

The leader-follower control architecture with the signal flow is shown in Figure 5.3. The flight plan module generates the desired path [x_{dl}, y_{dl}, z_{dl}] for the leader UAV, and the leader's tracking controller creates the reference speed and course angle [V_{dl}, χ_{dl}] for the autopilot. The autopilot generates corresponding attitude and throttle commands to the actuators. After the leader's position is updated, the navigation unit records the new coordinates [x_l, y_l, z_l] and the communication unit sends the information to the follower UAV. When the formation planning module receives the position of the leader UAV, according to [x_s, y_s, z_s], which is the relative position of the follower to the leader derived from the initial formation scheme, the module generates the desired position [x_l-x_s, y_l-y_s, z_l-z_s] for the formation controller. Then the formation controller creates a desired position [x_{df}, y_{df}, z_{df}] for the follower UAV to track. Following a data flow diagram similar to the leader's, the follower's actual coordinates [x_f, y_f, z_f] are sent back to the formation controller and a new referenced position is generated for the follower UAV.

In the case of choosing the relative coordinates to the leader's, three parameters need to be described:

$$fm_e = x'_f \times \sin\chi_l - y'_f \times \cos\chi_l, \tag{5.1}$$

$$fm_n = x'_f \times \cos\chi_l + y'_f \times \sin\chi_l, \tag{5.2}$$

$$fm_a = z'_f, \tag{5.3}$$

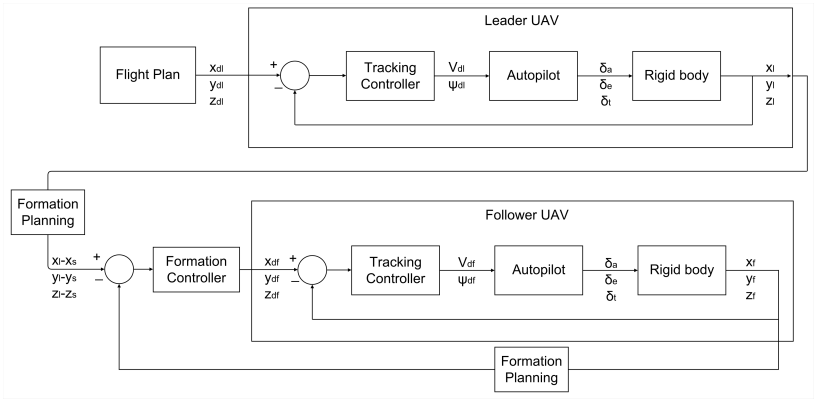


Fig. 5.3: Leader-follower formation controller architecture.

where x'_f, y'_f, are z'_f are the follower's desired local coordinates defined in the initializing functions. Transforming from the inertial frame to the UAV frame, the fm_e is used to calculate the initial separation distance between leader and follower regarding the longitudinal coordinate. The fm_n is used to calculate the initial distance regarding the latitudinal coordinate.

In order to calculate the control requisites, four control parameters related to the position, altitude, and speed control loops are defined:

$$f_e = (x_f + v_f \times \sin\chi_f \times t_s - x'_l) - (fm_e - x'_f), \tag{5.4}$$

$$f_n = (y_f + v_f \times \cos\chi_f \times t_s - y'_l) - (fm_n - y'_f), \tag{5.5}$$

$$f_a = (z_f - z'_l) - (fm_a - z'_f), \tag{5.6}$$

$$f_s = v_f, \tag{5.7}$$

where x'_l, y'_l, are z'_l are the leader's desired relative coordinates which are usually set at 0s, t_s is the sample time. f_e and f_n are the error values for the longitudinal and latitudinal coordinates. The calculations are based upon the sum of the follower's current position and its instantaneous velocity at specified sample time minus the reference position. The f_a is the error value for the altitude which is calculated based upon the follower's current altitude minus the reference altitude. The speed control loop is based on the follower's velocity.

Regarding the altitude control loop, altitudes of the leader and follower are found, then the control effort is initialized based upon difference altitude and is updated using the following equations:

$$z'_c = z_l - z'_l, \tag{5.8}$$

$$z_c = z'_c + z'_f + k_a \times f_a. \tag{5.9}$$

In order to achieve a steady altitude tracking, k_a needs to be carefully selected. The default setting is 0.03 and it is in the range between 0.01 and 0.1. When the follower is unable to track the leader's altitude but is always below the reference altitude, k_a must be increased beyond 0.03 and vice versa.

The position control requires calculation of the differences of longitudinal and latitudinal positions between the leader and the follower. Then the desired longitudinal and latitudinal positions for the follower are generated using the following equations:

$$d_x = x_l + V_{NS} \times fm_c \times \sin\chi_l + x'_f - x'_l, \tag{5.10}$$

$$d_y = y_l + V_{NS} \times fm_c \times \cos\chi_l + y'_f - y'_l, \tag{5.11}$$

where V_{NS} is the nominal airspeed and fm_c is an offset time. Both constants are defined in the airframe configuration file. By multiplication the distance between the desired waypoint and the follower is generated. Then d_x and d_y become the position control inputs for the follower UAV's navigation function.

Regarding the speed control loop, the control requisite for the follower is initialized as the constant defined in the airframe configuration file. This is then updated according to following equation:

$$s_c = k_p \times (f_n \times \cos\chi_l + f_e \times \sin\chi_l) + k_s \times f_s, \tag{5.12}$$

where f_s is the current velocity of the follower. The proportional position gain k_p has a

default setting of 0.01 and it is in the range between 0.01 and 0.1. The proportional speed gain k_s has a default setting of 0.2 and it is in the range between 0.1 and 1. When the follower always exceeds the desired distance and is too close to the leader, the value of k_s must be reduced. The overshoot situation has to be checked and rectified, and then the k_p is adjusted to achieve smoother position tracking. If the follower is beyond the desired distance, k_s is increased and then k_p is adjusted until the follower can closely track the leader's position within the reference distance.

5.2.3 Experiments

For the purpose of achieving stable multi-UAV formation flights, a special flight plan has been designed. For the follower UAV, since it needs to follow the trajectory of the leader UAV, its distance and altitude relative to the leader's during the formation mode need to be monitored and actions need to be taken when they are too close. Other than that, users just need to make sure it stays in a different fly zone and different altitudes before the formation mode gets activated. For the leader UAV, a series of consecutive waypoints were created, several of which remain in one direct line so that the formation controller can be tuned while both UAVs are commanded to fly straight. Also the leader's altitude is set to be 50 m higher than that of the follower for the first trial because it is always easier for the UAV to descend than to ascend. Once their altitudes stay steady, the altitude difference can be gradually reduced and finally they can fly in the same altitude.

Numerous simulation tests have been conducted under the Paparazzi ground station software. Detail formation flight software setup under Paparazzi architecture is documented in Appendix C. The simulator is a trustworthy tool to test new flight plans and verify the formation flight when any parameters is modified because it uses the same map and waypoints. The flight plan and procedures presented above have shown their effectiveness and reliability. Since the actual flight is similar to the simulation test, it is important to practice monitoring and controlling more than one UAV at the same time. Figure 5.4 shows the Paparazzi GCS simulation with multi-UAV interface.

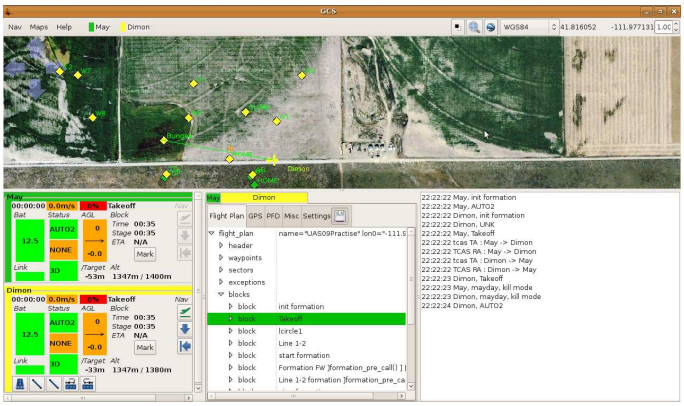


Fig. 5.4: Simulated multi-UAV formation flight.

To date, the following formation scenarios have been tested in the simulations[1]:

(1) Two UAVs: leader-follower formation;

(2) Three UAVs: leader-followers triangle and string formation;

(3) Four UAVs: square formation;

(4) Four UAVs: tetrahedron formation (One leader above with three followers beneath);

(5) Five UAVs: string and pyramid formation.

In all the formation scenarios, there is always one leader, which means the number of followers can be dynamically increased or decreased accordingly. The UAVs have behaved stably under different simulation scenarios because there are no practical hardware constraints or various environmental restrictions. Under near optimal conditions, the actual flight performance should be close to the simulated results. A formation reconfiguration setting has also been designed so that the formation configuration can be modified with increased flexibility. This function includes three parameters, x_s, y_s, z_s. Instead of defining them as constants in the airframe configuration file as before, now they can be adjusted within certain ranges. Therefore, different formation scenarios can be readily achieved during flight tests by modifying the parameters based upon the actual mission requirements.

[1]OSAM Youtube Channel: http://www.youtube.com/user/USUOSAM

Different formation scenarios play critical roles in actual flight missions. The string formation flight can reduce the time to cover a certain area when the UAVs are searching and acquiring images above this area. Also, the covered area can be expanded using the same amount of time to improve the efficiency. The team of UAVs can also fly at different altitudes so that images, for instance for ground surface surveillance can be obtained at different spatial resolution. When four UAVs are flying as a square formation, the distance between each other can be adjusted so their cameras can cover a large area with the least overlapping.

The current fleet for formation flight is shown in Figure 5.5. All the 48-inch UAVs have been fully tested and able to individually perform satisfactory autonomous flights.

The flight test and tuning protocol is explained in the following procedures.

(1) Step 1: Launch the leader and follower UAVs sequentially. Let the two UAVs circle in their own fly zones and maintain different altitudes.

(2) Step 2: Let the leader UAV as the first enter formation mode and start flying towards the waypoint routine, and then switch to follower UAV and keep the original altitude until the follower starts tracking the leader's flight path.

(3) Step 3: When both UAVs are in the formation mode and flying straight, tune the proportional position and speed gains in the formation controller so that their latitudinal and longitudinal differences converge to the desired values. Then adjust the proportional altitude gain so their altitude difference stay close to the desired one.

(4) Step 4: When both UAVs are turning, the leader UAV might drop altitude so their distance might get too close; therefore, make certain that the traffic collision avoidance function is activated and reserve additional distance for security reasons.

(5) Step 5: If both UAVs maintain good formation shape in the straight line, controller tuning can be discontinued and different formation scenarios can be managed. Otherwise, if any UAV behaves abnormally, the communication quality need to be inspected. The wind condition also plays a critical role in the tuning process.

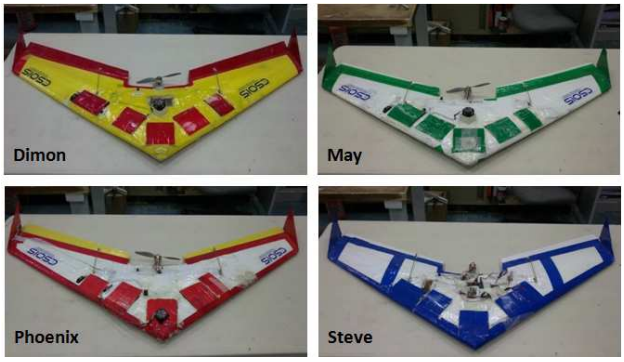


Fig. 5.5: Current formation flight fleet.

(6) Step 6: When formation flight is completed, the follower should be removed from the formation mode to fly towards its proper fly zone, then switch to circle and be ready to land. After the follower UAV has landed, the leader can follow the same procedure to finish the flight.

The original communication topology was point-to-point (P2P) (Figure 5.6(a)), which is the simplest topology and it builds a steady link between two endpoint. In present applications, the consequence is that the ground modem can communicate with each RF module on the UAV only one at a time. For the formation flights, two or more UAVs need to send messages to the ground modem and receive new commands from it. The P2P setting narrows the bandwidth and restricts the number of messages that can be transmitted simultaneously to the ground station. Because of this restriction, significant issues in the present experiments were found to be communication delay and intermissions with data loss. This adversely affect the performance of the formation controller since the centralized configuration requires sufficient message feedback so it can issue punctually updated commands to the follower.

In order to achieve consistent formation flight performance, the communication problem has to be addressed and solved. Fortunately, the modem bundle adopted provides the options for point-to-multipoint topology and even mesh networking. For the current control structure, the point-to-multipoint (P2MP) setting appear to positively resolve the delay and

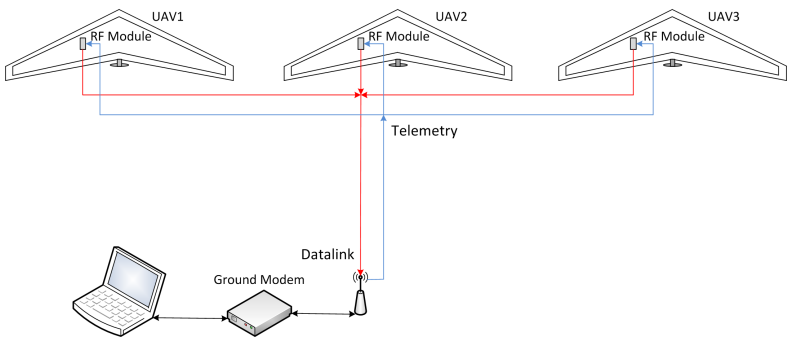
intermission issues. The comparisons between P2P and P2MP are illustrated in Figure 5.6.

To set up the P2MP for the ground modem, the TX/RX mode can be directly changed through the dip-switch settings on the case. For the RF module, they need to be reprogrammed using the software called X-CTU which is associated with the hardware. Table 5.1 and Figure 5.7 illustrate how to program the RF module from the P2P to a P2MP.

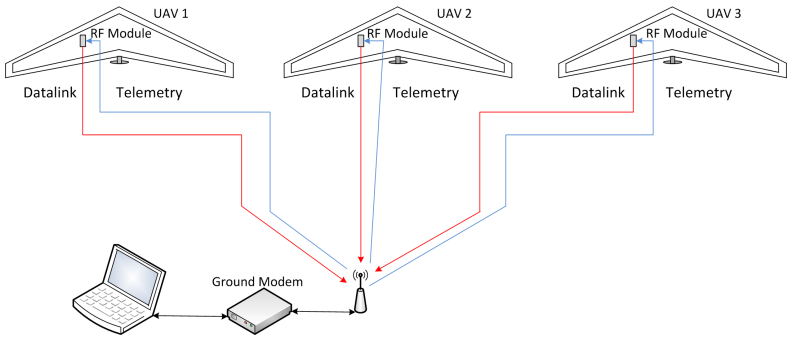
Ground logging tests between P2P and P2MP using two UAVs have been performed. While the ground modem and two RF modules for the airborne modems were all in the lab and within a short distance, the telemetries were working without significant delays under both topologies and there was no obvious intermissions on the datalink. However, during the same logging time, there were more data received using P2MP. While the two RF modules were located outside the lab and the ground modem remained in the lab for acquiring the GPS logging, it was nearly impossible to issue new commands or change parameters through telemetry under P2P. Nevertheless, under P2MP, the telemetry still worked stably without significant delays or interruptions. The P2MP setting has also been used to test the communications of three UAVs during ground, and flight tests and the results are satisfactory. Tables 5.2 and 5.3 compare the logging results between P2P and P2MP under different test conditions.

Table 5.1: Modem power-up options.

Condition	Behavior	Commands Sent
If SW5 & SW6 are off	Multipoint Base	ATMY 0 (Source Address) ATDT FFFF (Destination Address) ATMT 3 (Multi-Transmit option)
If SW5 is OFF & SW6 is ON	Multipoint Remote	ATAM (Auto-set MY, MY=unique) ATDT 0 (Destination Address) ATMT 0 (Multi-Transmit option) ATRR A (Retries)
If SW5 is ON & SW6 is OFF	Point-to-point	ATAM (Auto-set MY, MY=unique) ATDT FFFF (Destination Address) ATMT 3 (Multi-Transmit option)



(a) Point-to-point topology.



(b) Point-to-multipoint topology.

Fig. 5.6: Communication topology comparisons.

5.2.4 Formation Flight Results

Successful formation flights were made weekly beginning March 2011. The test flights were conducted with two and three UAVs flying simultaneously. Figure 5.8 illustrates the equipments at a recent flight tests when three UAVs participated.

Figures 5.9 to 5.14 show the latest results when two 48-inch UAVs flew simultaneously and performed a leader-follower formation flight under different scenarios. There are two formation modes designed into the software. The default mode is `NAV_MODE_COURSE` (*COURSE*), which uses the position of the guiding carrot, the position and the speed vector of the UAV to compute a navigation course angle. The other mode is `NAV_MODE_ROLL`

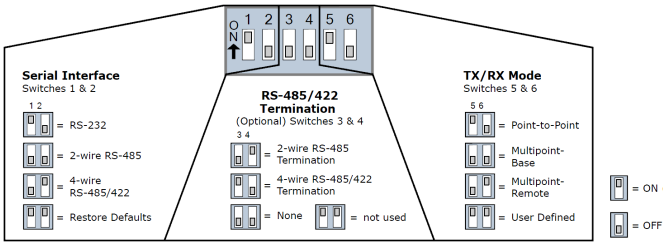


Fig. 5.7: DIP switch settings.

Table 5.2: Data logging comparisons of two UAVs.

Test	Duration	Data Size
(1)P2MP (Indoor)	25min	2.5MB
(2)P2MP (Indoor)	25.5min	2.8MB
(3)P2P (Indoor)	25min	1.7MB
(4)P2P (Indoor)	25min	1.7MB
(5)P2MP (Outdoor)	25min	2.9MB
(6)P2P (Outdoor)	25min	2.1MB
(7)P2P (Flight)	26min	2.48MB
(8)P2MP (Flight)	24min	3.86MB

($ROLL$), which computes the course angle directly from the the guiding carrot and its speed vector. From the comparisons of using both modes for formation flights, it can be concluded that the $COURSE$ mode provides more accurate heading-angle tracking as it updates faster. However, for the follower UAV, the software adds additional complications because the heading of the leader updates more frequently. The $ROLL$ mode provides more time for the follower to respond since the leader's heading changes more slowly. The actual formation flight perform well regarding flying trajectories. On the other hand, for single UAV autonomous flight, the $COURSE$ mode should be adopted.

Figure 5.9 compares the performance under the square shape path using the two formation modes. In the square shape formation flights, the leader was maintained at 180 m (1530 m AMSL) while the desired altitude altitude for the follower was 60 m lower in altitude. A four-waypoint route was designed with equivalent distance between each vehicle, so the leader was able to stay at similar altitudes during each waypoint transition. The

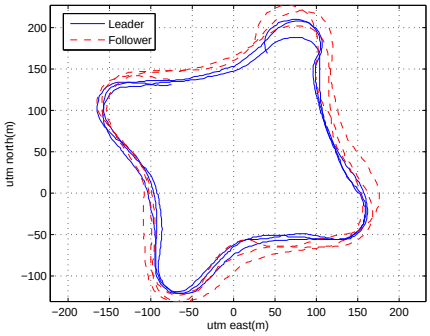
Table 5.3: Data logging comparisons of three UAVs.

Test	Duration	Data Size
(1)P2MP (Ground Modem Indoor)	26.5min	3.2MB
(2)P2MP (Ground Modem Outdoor)	25min	4.2MB
(8)P2MP (Flight)	46min	8.2MB

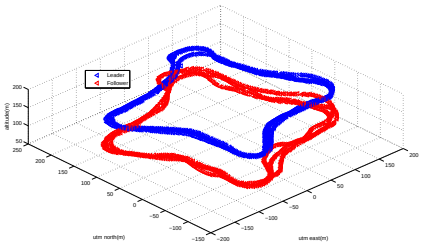
Fig. 5.8: Three low-cost UAV testbeds during flight test.

follower was able to track the leader's trajectory under both modes while the *COURSE* mode enabled the leader to track the path more accurately. The trajectory was close to a square shape. Relatively, the tracking performance of the follower was less accurate than under *ROLL* mode because the complexity introduced by the leader. For *ROLL* mode, the trajectory was more like a cross since the leader's tracking was updated relatively slowly so the follower was able to smoothly trace it.

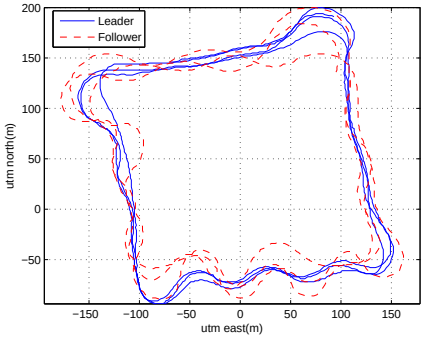
Figures 5.10 and 5.11 show the circle formation and waypoint tracking formation under `NAV_MODE_ROLL`, respectively. In Figures 5.10(a) and 5.11(a), the blue line shows the leader's trajectory and the red dashed line shows the follower's trajectory. The leader UAV took off first and flew towards its circling waypoint and then stayed in standby circling with an altitude of 180 m (1530 m AMSL). Then the follower UAV took off and reached the altitude of 120 m (1470 m AMSL) by circling in a separate flight zone. After both UAVs steadily circled for several turns, the formation reconfiguration module was used to set the follower's formation altitude to be 60 m below that of the leader. Then the leader was switched to



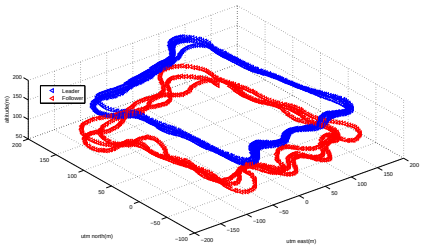
(a) 2D flight path ($ROLL$).



(b) 3D flight path ($ROLL$).



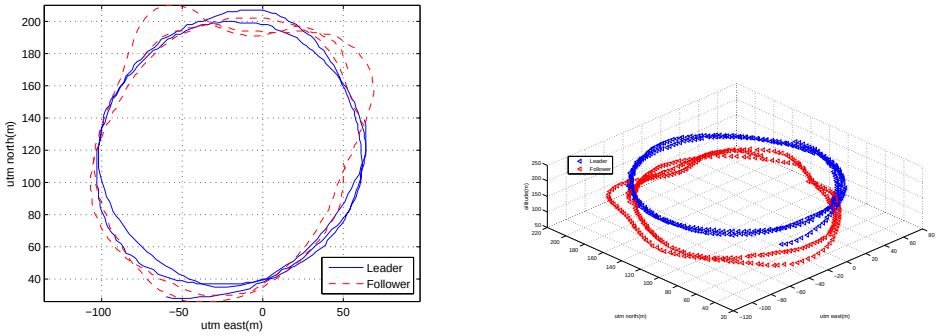
(c) 2D flight path ($COURSE$).



(d) 3D flight path ($COURSE$).

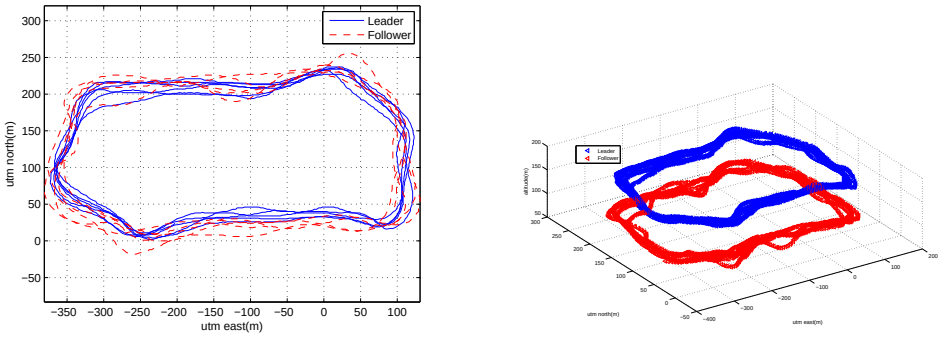
Fig. 5.9: Square shape formation comparisons.

formation mode and continued circling and the follower started approaching the leader when its formation mode was also activated. The follower was able to closely track the leader's trajectory and altitude, and the results were stable under three scenarios. The first scenario was a square shape path which has been described in the previous paragraphs. The second scenario was the circling formation at 180 m (1530 m AMSL) for the leader while the follower was also circling about 60 m below. After that, a 9-waypoint flight path at 180 m (1530 m AMSL) was tested for several rounds, and the follower was able to track the trajectory. The results were satisfactory after refined tunings on the formation controller were made. The TCAS function in each case was able to guide both UAVs and diverge their headings if they were too close. Sometimes, this caused the UAV altitudes to fluctuate. The



(a) 2D flight path. (b) 3D flight path.

Fig. 5.10: Standby circling formation flight trajectories.



(a) 2D flight path. (b) 3D flight path.

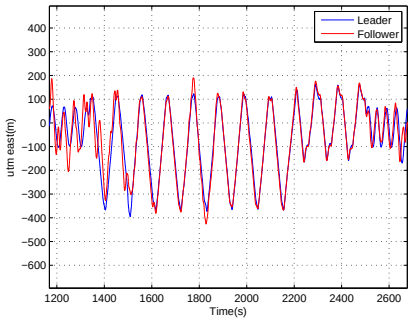
Fig. 5.11: Way-point tracking formation flight trajectories.

settings of the TCAS function were modified to reduce the sensitivity. After they flew for about 40 minutes, the safety pilot manually landed both UAVs.

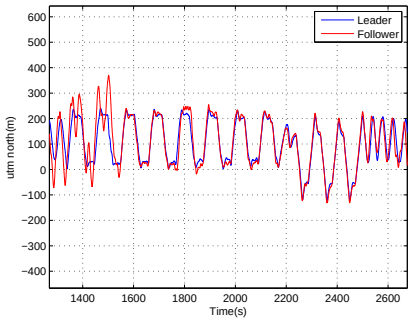
From Figures 5.12 to 5.14, a set of explicit altitude tracking and 2D position tracking data are illustrated. From these plots, it is observed that the leader is able to track its referenced altitude with an offset range between ±10 m (Figure 5.13(a)). Because the follower's referenced altitude is based upon that of the leader's, the UAVs have to be separated with a specified altitude difference so the collision avoidance function sensitivity can be reduced. The best altitude tracking error achieved so far is about +20 m (Figure

5.13(b)). The biggest challenge to the 3D tracking was due to the altitude error. Because the altitude control loop of a single UAV is achieved through both pitch and throttle control loops, and by default, the UAV first adjust its throttle to reach the desired altitude. The altitude error results in the follower staying at a high throttle most of time with a fast airspeed. When the follower reaches the desired altitude, it might also pass the horizontal position the leader located, and have to circle back and try to repeat the trace. This adversely affects the 3D position tracking. After the follower's pitch neutral values were increased and the maximum throttle percentage was reduced, an improved balance between the altitude and speed control loop was acquired. Since the altitude tracking error was reduced, the requirement that the follower kept a high throttle in order to track the desired altitude was removed. Meanwhile, a 20 m distance was added between the leader and the follower. This provided a larger buffer zone for the follower which helped to avoid an overshoot situation. By these means, a satisfactory 3D trajectory tracking performance was achieved with small altitude, x axis, y axis tracking errors as demonstrated in Figures 5.13(b), 5.14(a), and 5.14(b), respectively.

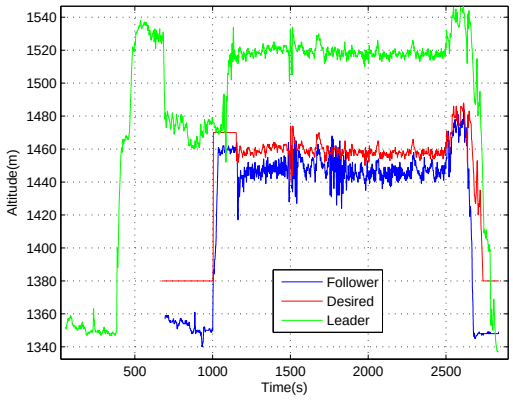
Figure 5.15 shows the first successful trial of flying three UAVs at the same time. The test was used to validate the new P2MP communication topology. This assured that the hardware could support actual 3-UAV formation flights. In the test, three UAVs were launched consecutively, and those UAVs circled at different altitudes (100 m, 120 m, 150 m) over three different areas. Then one remained circling at a standby waypoint while the other two continued regular leader-follower formation flight. The entire flight lasted about 46 minutes involving many flight plan switchings and flight parameter adjustments. The only problem encountered was the altitudes of the follower and non-formation UAV were too close. This often activated the TCAS function, so consistent altitude tracking from the follower UAV could not be obtained. Other than that occurrence, the communication was stable with insignificant delays or unnoticeable data loss on the datalink. The test verified that the design approach can be adopted for three UAV triangular formation flights.



(a) X-axis comparisons.



(b) Y-axis comparisons.



(c) Altitude comparisons.

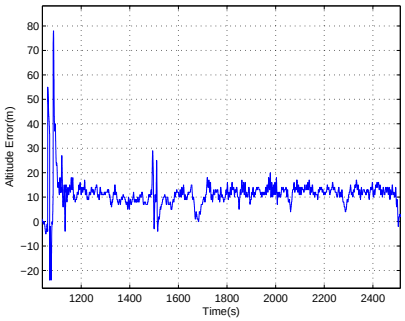
Fig. 5.12: Leader and follower 3D position comparisons.

Practical lessons learned from the tests are summarized as follows to achieve successful multi-UAV formation flights.

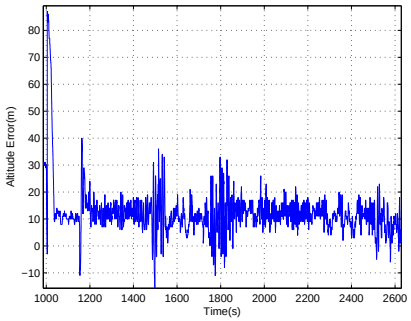
(1) The single UAV needs to be well tuned and capable of performing fully autonomous flights.

(2) Because centralized structure requires large bandwidth in the wireless communication units, it is necessary to eliminate irrelevant messages regarding UAV navigation and fail-safe measures.

(3) System delays can be expected which randomly cause the formation controller to

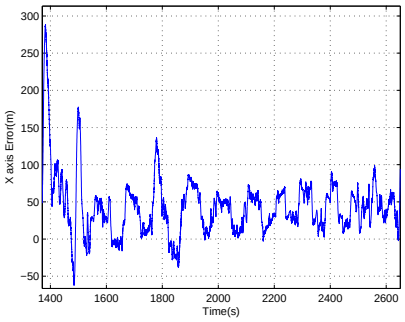


(a) Leader altitude tracking error.

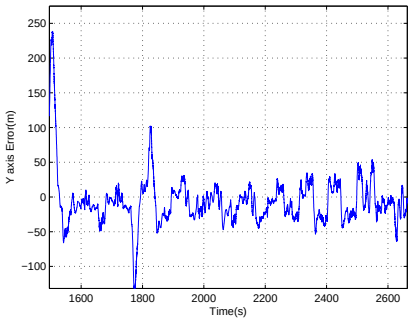


(b) Follower altitude tracking error.

Fig. 5.13: Leader and follower altitude tracking.



(a) Follower X-axis tracking error.



(b) Follower Y-axis tracking error.

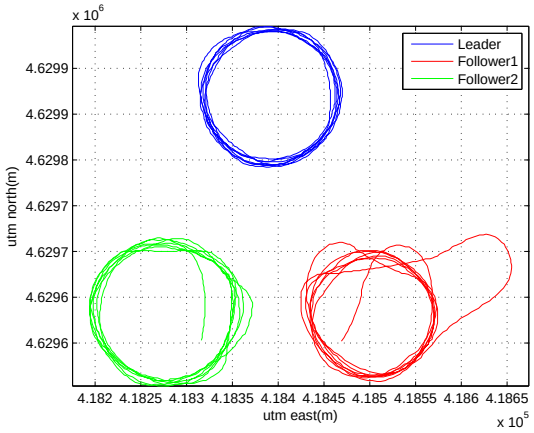
Fig. 5.14: Follower 2D position tracking errors.

malfunction. Therefore, it is important to reserve sufficient vertical and horizontal distances between the leader and the follower.

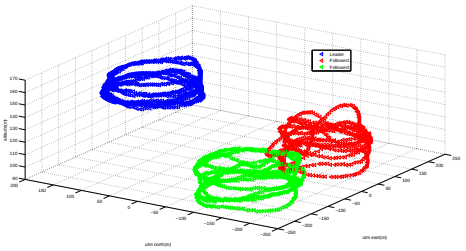
(4) While the control algorithm was still in the development stage, it is preferred that each UAV has its individual safety pilot. When an emergency occurs, the safety pilot has sufficient time to rectify the situation.

5.3 Chapter Summary

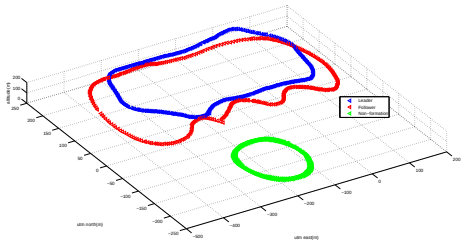
In this chapter, the system development is documented based upon the control principle and the adopted structure. Afterwards a tuning procedure based on the centralized



(a) 2D flight path.



(b) 3D flight path1.



(c) 3D flight path2.

Fig. 5.15: Circling flight test of three UAVs.

formation controller is presented including the detailed algorithms. The experiments include simulation studies of different formation flight scenarios and the advantages of each, a flight test and tuning protocol. Several lessons are drawn based on the actual experience and comprehensive routinized formation flight results. Explicit performance analysis are also presented.

Future work includes (1) implementation of a cognitive decentralized control structure on the current system, (2) design of a mesh network topology to improve the communication performance, and (3) exploitation of the heterogeneous formation flight such as combining a vertical take-off and landing (VTOL) UAV and the fixed-wing UAV.

Chapter 6

Flight Controller Designs

6.1 Introduction

Flight control is the most essential function in achieving complex flight missions for UAVs. In these studies, research was conducted into various types of flight controls. These included fixed-wing UAV roll channel control [27], VTOL UAV altitude control [76], and preliminary work on speed control. This chapter briefly introduces the precursory research into the speed loop control with focus upon the VTOL altitude control. The principal control loops on Paparazzi are responsible for the roll, pitch, course, and altitude control and for generating correct control of the actuators. The main control structure is shown in Figure 6.1.

6.2 Fixed-wing UAV Airspeed Control

The speed control loop is shown in Figure 6.2. The pitch and throttle are controlled separately and are not coupled in the control loops. Airspeed is controlled by two cascaded PI loops. The first one is used to indicate the ground speed and the second one is to indicate the airspeed. This can be used to ensure that when the ground speed decreases below a fixed value, the airspeed can be increased for compensation so that the UAV can maintain a valid GPS heading.

In order to achieve closed-loop speed control and measure wind profiles for multiple UAVs, air pressure sensors have been integrated into the current UAV platform. Two types of pressure sensors were employed, the first is absolute pressure sensor, which measures input pressure in relation to a zero pressure. The absolute pressure sensor is used to measure altitude. The other one is differential pressure sensor, which is designed to accept

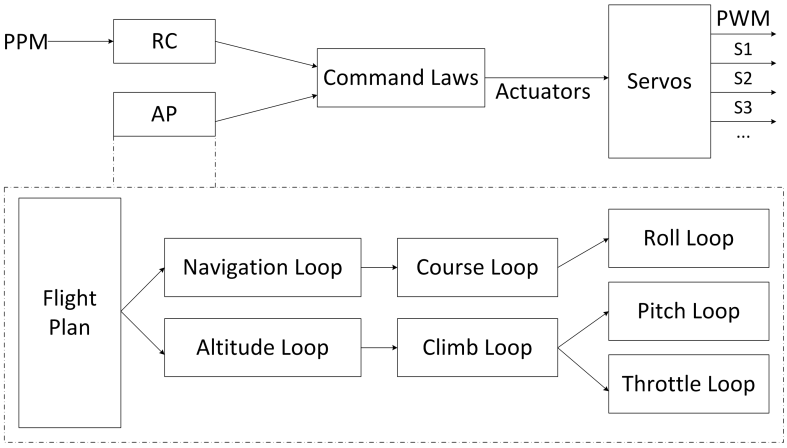


Fig. 6.1: Paparazzi main control structure.

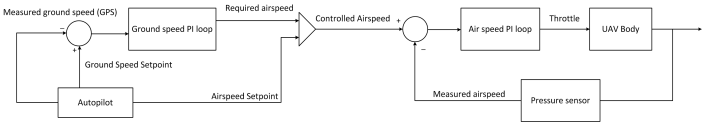


Fig. 6.2: Speed control loop with pressure sensor feedback.

simultaneously two independent pressure sources. The output is proportional to the difference between the two sources. The differential pressure sensor is used to measure indicated airspeed.

The air speed calculation is based on the air pressure and can be calculated through the following equation:

$$v = \sqrt{\frac{2}{\rho} \times q}, \tag{6.1}$$

where q is the dynamic pressure and ρ is the air density.

The current system has three differential pressure sensors available; these are the Freescale MPXV7002DP, Freescale MPXV5004DP, and Eagle Tree Airspeed Microsensor V3. Comparisons of the three sensors are summarized in Table 6.1, and preliminary test results are shown in Figure 6.3.

6.3 VTOL UAV Altitude Control

6.3.1 Introduction

In recent years, vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV) has attracted the interest of researchers. VTOL UAVs have extensive applications because they are able to be operated from airfields in a diverse array, independent of launching and landing spaces, and are able to hover at specified altitudes. Typical applications for VTOL UAVs are search and rescue, acquisition of aero-images, transportation. Although VTOL unmanned aircraft has relative simplicities in the structure, VTOL UAV usually is an unstable system with nonlinear dynamic behavior. Therefore, it includes more complexity on the controller design. Due to current and potential applications, the altitude stabilization of VTOL unmanned aircraft has become an important research topic. In order to achieve stable and fast response on the altitude control, an advanced controller needs to be developed with complex control strategy. Some control strategies like PID are presented for a linear model.

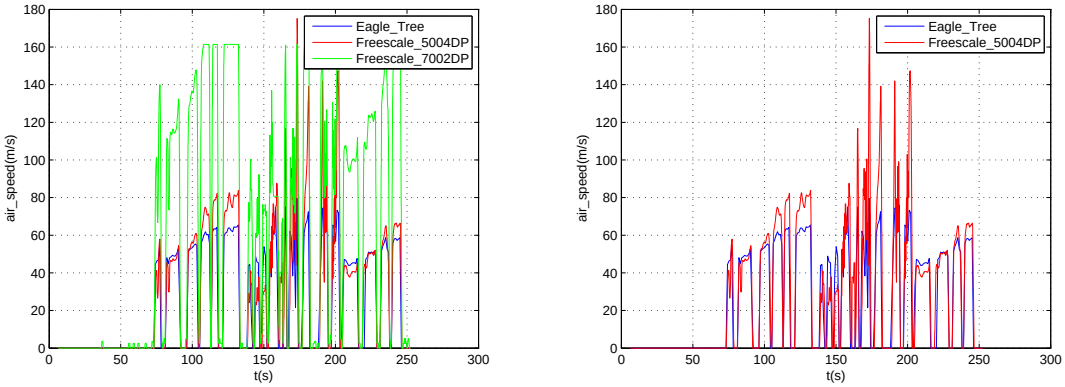
This chapter is organized as follows: In subsection 6.3.2, the flight control basics of VTOL UAV are presented. In subsection 6.3.3, system identifications are drawn for the decoupled altitude flight control of VTOL UAV. In subsection 6.3.4, the designed controllers are presented and the simulation results are illustrated regarding the designed modified Ziegler-Nichols PI and designed IOPID controllers.

6.3.2 VTOL UAV Flight Control Basics

A simple quadrotor VTOL UAV is shown in Figure 6.4 and the dynamics of VTOL UAV flight control system can be modeled as given below [77]:

Table 6.1: Pressure sensor comparisons.

Sensor Type	MPXV7002DP	MPXV5004DP	Eagle Tree
Pressure Range	-2 to 2 kPa	0 to 3.92kPa	—
Offset Value	771	56	1532
Airspeed Range	—	—	3kph-563kph
Accuracy	±2.5%	±1.5%	1kph
Sensitivity	1 V/kPa	1 V/kPa	—
Input Voltage	4.75-5.25V	4.75-5.25V	3-16V



(a) Three pressure sensor comparisons.

(b) Two pressure sensor comparisons.

Fig. 6.3: Pressure sensor comparison.

(1) Position: longitude (x), latitude (y), altitude (z);

(2) Attitude: roll(ϕ), pitch(θ), yaw(ψ);

(3) Velocity in the body axes: v_x, v_y, v_z;

(4) Angular rate: roll rate (P), pitch rate (Q), yaw rate (R);

(5) Acceleration: a_x, a_y, a_z;

(6) Moment of inertia: I_x, I_y, I_z;

(7) Drag forces in the body axes: F_{dx}, F_{dy}, F_{dz}.

Fig. 6.4: Quadrotor VTOL UAV.

Most of the VTOL aircraft, such as helicopters, quadrotors, and airships, have one control input, namely, the throttle. In this paper, study is focused upon an unmanned quadrotor, where throttle is the only flight control input considered.

Several nonlinear equations can be used to model the six degrees of freedom in VTOL UAV dynamics. However, it is difficult to analyze the nonlinear model for the altitude control. Therefore, a trimming point is considered so that the altitude control loop can be decoupled as a single-input and single-output (SISO) system. This enables the linear system theory to be applied to the analysis and controller designs.

The decoupled altitude loop control problem is treated in this chapter after simplifying the altitude loop control as a SISO (throttle-thrust) case. Cascaded controllers were designed to provide a stable and robust altitude control performance.

6.3.3 System Identification for the Altitude Control of the VTOL UAV

Closed-loop Altitude Control System Identification

In order to design an altitude controller for quadrotors, it is necessary to find an accurate model representing the altitude control loop. A traditional method to identify control loop on the altitude is to use an open-loop analysis. However, several constraints restrict the application of this method, including small references, difficulties of stabilizing the quadrotor under open loop, etc. As a consequence a close-loop system identification

method was used since it is able to ensure the altitude stability of the system. However, PID rough tuning has to be added in order to meet the circumstances discussed previously. Then the system identification procedure is able to identify the model under nearly stable state conditions to establish an acceptable accurate model. The diagram of the close-loop system identification is shown in Figure 6.5.

An ARX model is used because the first order ARX model is able to provide a relatively simple controller design. The altitude ARX model is defined as:

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}, \tag{6.2}$$

where $Y(z)$ is the actual altitude output and $U(z)$ is the reference altitude input.

Additionally, in order to design the proposed modified Ziegler-Nichols PI controller, a first-order plus time delay (FOPTD) model is identified based upon the ARX model. It is defined as:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Ke^{-Ls}}{\tau s + 1}. \tag{6.3}$$

System Identification Results

OS4 Simulation Platform is a Matlab Simulink model developed by Samir Bouabdallah from Ecole Polytechnique Fédérale de Lausanne (EPFL) [78,79] for the system modeling and control design on an actual quadrotor platform. Based upon the actual system dynamics, the simulation system consists of dynamics of the actuator, aerodynamics of the quadrotor, a controller to avoid obstacles, and a planner to define the waypoints.

The control input of the OS4 model is the throttle, with white noise and delays. The outputs comprise [78]:

System States:

(1) Position: x, y, z;

(2) Ground speed: v_x, v_y, v_z;

(3) Attitude: ϕ, θ, ψ;

(4) Angular rate: p, q, r;

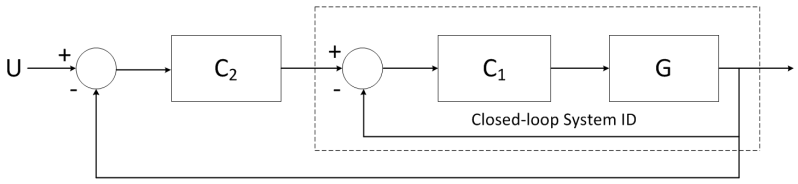


Fig. 6.5: Closed-loop system identification procedure.

Sensors Measurements:

(1) IMU: ϕ, θ, ψ;

(2) Range Sensor: z;

(3) Position Sensor: x, y, ψ.

The relationship between the input vector U and every motor speed is shown below:

$$U = [U_1\ U_2\ U_3\ U_4]^T, \tag{6.4}$$

$$\begin{cases} U_1 = T({n_1}^2 + {n_2}^2 + {n_3}^2 + {n_4}^2), \\ U_2 = T(-{n_1}^2 + {n_3}^2), \\ U_3 = T({n_2}^2 - {n_4}^2), \\ U_4 = D(-{n_1}^2 + {n_2}^2 - {n_3}^2 + {n_4}^2), \end{cases} \tag{6.5}$$

where T is the thrust factor, D is the drag factor, and n_1, n_2, n_3, n_4 are the rotor speeds for each motor.

The sampling period in the simulation system is 0.01 s. The block diagram is shown in Figure 6.6.

A PID controller was added in the original altitude control block and the parameters were roughly tuned to achieve a marginally stable closed-loop flight control. A square wave at 0.05 Hz in frequency and a magnitude between 0 and 1 was chosen to be the reference input. This provides the system model sufficient time to track, and it fully identifies the model for increasing and decreasing the altitude. Because it is difficult to select a dependable

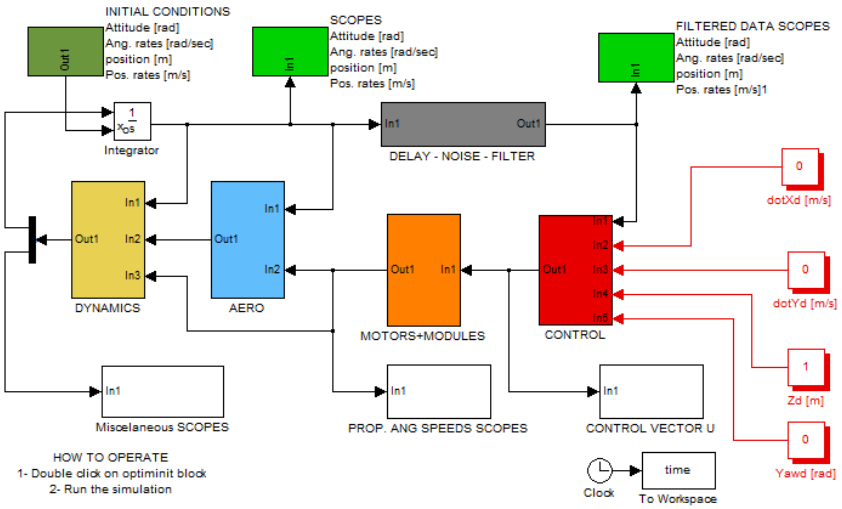


Fig. 6.6: OS4 quadrotor simulation platform.

frequency range, the system identification was based upon the time domain. A $5th$ order ARX model is applied for the Steiglitz-Mcbride iteration method:

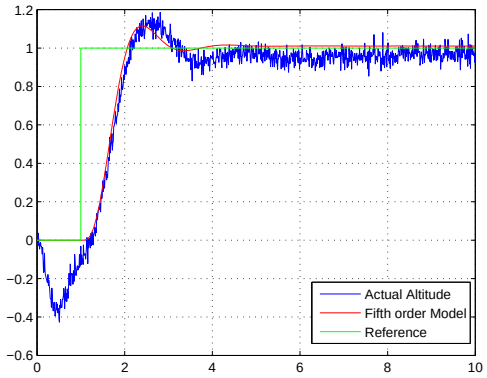
$$G_1(s) \quad = \quad \frac{B_1(s)}{A_1(s)}, \tag{6.6}$$

where $B_1(s) = 6.337 * 10^{-7}s^5 - 0.0004355s^4 + 0.1146s^3 - 12.14s^2 + 80.98s + 4.627 * 10^4$ and $A_1(s) = s^5 + 75.79s^4 + 1523s^3 + 8578s^2 + 2.774 * 10^4 s + 4.578 * 10^4$.

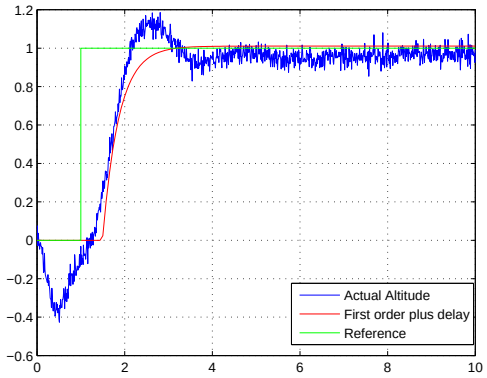
Then, the first order plus time delay (FOPTD) model is derived from the $5th$ order ARX model. The two models are shown in Equations (6.6) and (6.7):

$$G_2(s) \quad = \quad \frac{1.0108e^{-0.4938s}}{0.3741s + 1}. \tag{6.7}$$

Figure 6.7 shows the step responses of the two identified models on the altitude control compared with the reference and the simulated altitude from the OS4 Simulation Platform. It is observed that both outputs of the $5th$ order ARX model and FOPTD model are able to track the outputs of the OS4 simulator.



(a) Fifth order ARX model.



(b) First order plus delay model.

Fig. 6.7: System identification of altitude control loop.

6.4 Integer Order Controllers Design for VTOL Altitude Control

The integer-order controller design is given for the VTOL altitude control system, according to the identified model of subsection 6.3.3. The modified Ziegler-Nichols PI, integer-order PID controllers are designed, respectively. These two designed controllers were each implemented and compared on the OS4 Simulation Platform.

The identified first-order plus time delay model for the VTOL altitude control system discussed in this paper has the following transfer function:

$$P(s) = \frac{K}{Ts+1}e^{-Ls}. \tag{6.8}$$

This identified model was used for the controllers design given in the following section.

6.4.1 Modified Ziegler-Nichols PI Controller Design

The integer-order modified Ziegler-Nichols PI controller has the following transfer function:

$$C_1(s) \quad = \quad K_{p1}(1+\frac{1}{T_{i1}s}). \tag{6.9}$$

One of the most useful PID controller tuning rules, namely the modified Ziegler-Nichols PI (MZNPI) tuning rule [80], divides the tuning problem into several cases based upon different system dynamics 6.8,

(1) $L < 0.1T$ (Lag dominated)

$$K_{p1} = 0.3T/(KL), \quad T_{i1} = 8L;$$

(2) $0.1T < L < 2T$ (Balanced)

$$K_{p1} = 0.3T/(KL), \quad T_{i1} = 0.8T;$$

(3) $L > 2T$ (Delay dominated)

$$K_{p1} = 0.15/K, \quad T_{i1} = 0.4L.$$

Based on the identified first-order plus time delay model (6.8), the MZNPI controller was designed as follows:

$$C_1(s) = 0.2248(1 + \frac{3.3414}{s}). \tag{6.10}$$

The Bode plot of the open-loop system with $C_1(s)$ and $P(s)$ was presented in Figure 6.8, which can be compared with the Bode plots of other designed controllers.

6.4.2 IOPID Controller Design

The IOPID controller is designed according to the previous tuning strategy. Three design specifications are applied.

Controllers Design Scheme

In this section, the generalized controller is notated as $C(s)$, which represents the following IOPID controller:

$$C_2(s) = K_{p2} + \frac{1}{T_{i2}s} + T_{d2}s, \tag{6.11}$$

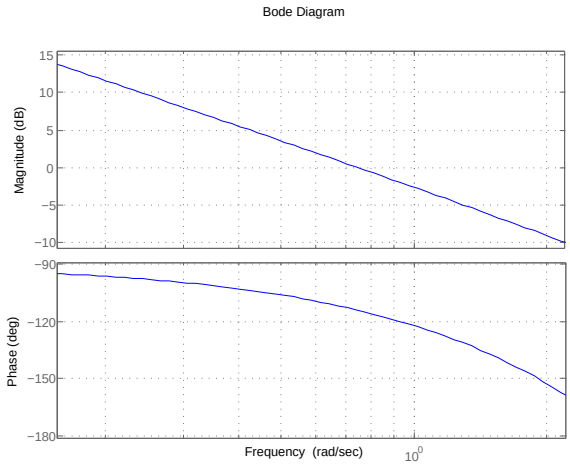


Fig. 6.8: The Bode plot of the open-loop system with the designed MZNPI controller.

Given two specifications, namely gain crossover frequency ω_c and phase margin ϕ_m, one can obtain,

$$\mathrm{C(j\omega)P(j\omega)}|_{\omega=\omega_c} = \mathrm{e}^{-(\pi-\phi_m)}.$$

Thus, two relationships about the gain and phase of open-loop system can be derived as follows:

Phase margin relationship

$$\mathrm{Arg[G(j\omega_c)]} = \mathrm{Arg[C(j\omega_c)P(j\omega_c)]} = -\pi + \phi_m; \tag{6.12}$$

Open-loop system gain relationship

$$|\mathrm{G(j\omega_c)}| = |\mathrm{C(j\omega_c)P(j\omega_c)}| = 1; \tag{6.13}$$

In order to obtain the robustness property regarding system gain variations, the flat phase specification [81] is applied for the third specification of the controllers design,

$$\frac{d(\mathrm{Arg(G(j\omega))})}{d\omega}|_{\omega=\omega_c} = 0. \tag{6.14}$$

The robustness regarding the plant gain variations demands that the phase derivative with respect to the frequency be zero, i.e., the phase Bode plot is flat around the gain crossover frequency.

The IOPID controller has three parameters. Three equations concerning the three parameters of the controllers can be built according to Equations (6.12), (6.13), and (6.14) for the three specifications. In theory, the three parameters of the IOPID controller can be obtained jointly. However, it is difficult to find the analytical solution as these three equations are complicated. Fortunately, a graphical method [27] can be used as a relatively simple means to find values for the three parameters of the controller. Therefore, the IOPID controller was designed following this scheme.

Given the gain crossover frequency $\omega_c = 0.7rad/s$, and the phase margin $\phi_m = 75°$,

the IOPID controller was designed following the design scheme, as presented in Equation (6.15).

$$C_2(s) = 0.3410(1 + \frac{1}{0.351s} + 1.7752s), \tag{6.15}$$

The Bode plots of the open-loop system with the designed controller and the identified model is shown in Figure 6.9. It can be seen that, each Bode plot illustrates the flat phase feature around the designed gain crossover frequency. At the same time, the phase margin requirement is satisfied in each of the three Bode plots.

6.4.3 Simulation Illustration

In this section, the designed MZNPI, IOPID controllers are implemented in the OS4 VTOL Simulation Platform for the altitude control, as shown in Figure 6.6. Figure 6.10 shows the step responses of the designed MZNPI controller with system gain variations of $\pm 20\%$. This is a baseline for the comparisons with the other three designed controllers.

From Figure 6.10, the system does not appear robust to gain variations, namely, the overshoot of the step response changes significantly as the system gain varies $\pm 20\%$. Meanwhile, the response overshoot is more than 20% of the step at the normal system gain. Using the designed IOPID controller, Figure 6.11 shows that the control robustness relative to system gain changes. The overshoots in the step responses almost maintain the same value with $\pm 20\%$ system gain variations. However, the overshoot magnitudes with the designed IOPID are comparable with that using the MZNPI.

6.5 Chapter Summary

This chapter introduces several flight controller designs with focus upon the VTOL UAV altitude control. This part of research also involves advanced roll channel controller design, but it is not included in this thesis. This chapter also includes the studies on speed control using pressure sensor feedback. The controller design scheme is shown with two decoupled PD controllers, and a comparison is made with several pressure sensors.

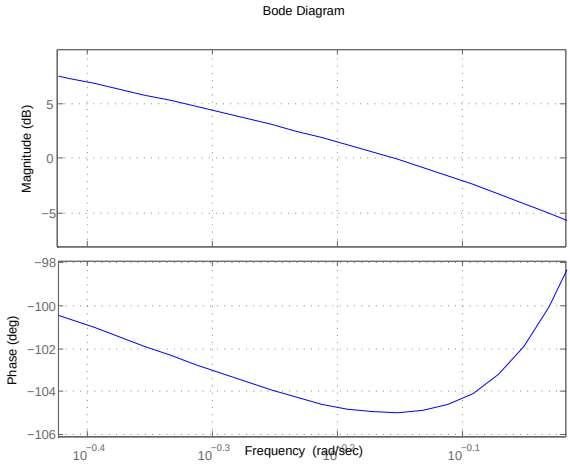


Fig. 6.9: The Bode plot of the open-loop system with the designed IOPID controller.

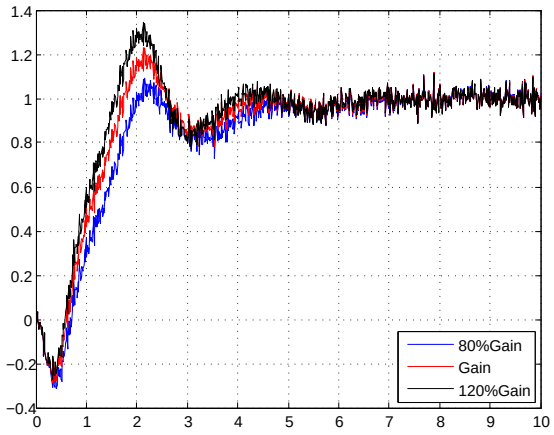


Fig. 6.10: Step responses using the MZNPI controller with plant gain variations.

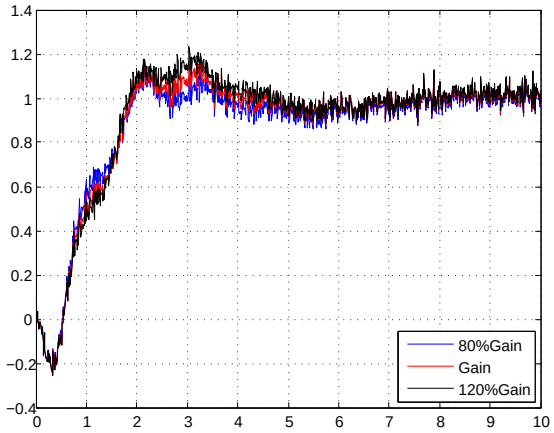


Fig. 6.11: Step responses using the design IOPID controller with plant gain variations.

Chapter 7

Conclusion and Future Suggestions

7.1 Summary

This thesis first presents the concept of cognitive personal remote sensing which includes using multiple UAVs for remote sensing purposes plus an intelligent architecture for UAV cognitive formation flight. Then it describes several UAV platforms involving the author's contributions and it focuses on the development of a low-cost UAV testbed for the research of multi-agent cooperative control as well as other usages. The airframe design, major hardware components, and software structure of the single UAV system are given in detail and the manifests for the general flight test protocols are given for each flight test. Afterwards, the research regarding attitude estimations for miniature UAVs is presented. It consists of a two-stage calibration method of using infrared sensors for UAV attitude measurement and a data fusion system using multiple attitude estimation sensors for enhanced UAV autonomous navigation performance. After that, the core of the master thesis regarding cooperative multi-UAV formation flight is presented including literature reviews of formation flight control strategies, the current control structure, and the distributed architecture. Then it describes a controller tuning procedures based the formation control architecture, experimental configuration, and flight test results. Several different flight controller designs which involve the author's participation and contributions are shown which provide explicit materials for future flight controller designs.

7.2 Future Work

There are several ideas for the future work. The first one is an intelligent multi-agent formation control architecture, which includes cognitive feature, inter-vehicle communication, heterogeneous capability. Inter-vehicle communication is the base of the new forma-

tion control structure (decentralized coordination and sensor network), and based on that, the cognitive features incorporate self-diagnosis, self-compensation, and self-optimization among all the UAV agents and their payload information. The heterogeneous capability combines both fixed-wing UAVs and VTOLs for synergistic collaborations. The second one is continuing estimation algorithm research for low-cost navigation sensors so those cheap gyroscopes, accelerators, GPS, magnetometers, and even pressure sensors can be combined to generate the near optimal attitude estimations for the single UAV development. The third one is advanced flight control designs for close-loop airspeed control using pressure sensor feedback and attitude control (pitch and yaw channels) using fractional order control techniques.

References

[1] A. Jensen, M. Baumann, and Y. Chen, "Low-cost multispectral aerial imaging using autonomous runway-free small flying wing vehicles," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium 2008 (IGARSS 2008)*, vol. 5, pp. V–506, 7-11 July 2008.

[2] J. S. Jang and D. Liccardo, "Small UAV automation using MEMS," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22(5), p. 30, May 2007.

[3] T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin, "Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of UAV," in *Proceedings of 2007 IEEE International Conference on Robotics and Automation*, vol. 5, 10-14 Apr. 2007.

[4] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," *IEEE Robotics and Automation Magazine*, vol. 16(2), p. 35, June 2009.

[5] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. Mclain, and M. Goodrich, "Autonomous vehicle technologies for small fixed wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 5, pp. 92–108, 2005.

[6] H. Chao, M. Baumann, A. Jensen, Y. Chen, Y. Cao, W. Ren, and M. McKee, "Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control," in *Proceedings of the IFAC World Congress*, 2008.

[7] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," *Proceedings of the IEEE*, vol. 94(7), pp. 1306–1324, July 2006.

[8] J. Lawton and R. W. Beard, "Synchronized multiple spacecraft rotations," *Automatica*, vol. 38, no. 8, pp. 1359–1364, 2000.

[9] P. K. C. Wang and F. Hadaegh, "Coordination and control of multiple microspacecraft moving in formation," *The Journal of the Astronautical Sciences*, vol. 44, no. 3, pp. 315–355, 1996.

[10] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. Hedrick, "An overview of emerging results in cooperative UAV control," in *Proceedings of 43rd IEEE Conference on Decision and Control (CDC 2004)*, vol. 1, p. 602, Dec. 2004.

[11] "Definition of cognition," http://www.answers.com/topic/cognition.

[12] H. Chao, A. M. Jensen, Y. Han, and Y. Q. Chen, *Geoscience and Remote Sensing*, ch. AggieAir: Towards low-cost cooperative multispectral remote sensing using small unmanned aircraft systems, pp. 467–490. IN-TECH, 2009.

[13] P. Brisset and G. Hattenberger, "Multi-UAV control with the Paparazzi system," in *Proceedings of the First Conference on Humans Operating Unmanned Systems (HUMOUS'08)*, 2008.

[14] P. Brisset, A. Drouin, M. Gorraz, P. S. Huard, and J. Tyler, "The Paparazzi solution," in *Proceedings of MAV*, 2006.

[15] C. Coopman, "Architecture, inertial navigation, and payload designs for low-cost unmanned aerial vehicle-based personal remote sensing," Master's thesis, Utah State University, 2010.

[16] A. Jensen, Y. Chen, M. McKee, T. Hardy, and S. Barfuss, "AggieAir - a low-cost autonomous multispectral remote sensing platform: new developments and applications," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium 2009 (IGARSS 2009)*, vol. 4, pp. IV–995, 12-17 July 2009.

[17] D. Casbeer, R. Beard, T. McLain, S.-M. Li, and R. Mehra, "Forest fire monitoring with multiple small UAVs," in *Proceedings of 2005 American Control Conference*, p. 3530, 8-10 June 2005.

[18] H. Chao and Y. Chen, "Surface wind profile measurement using multiple small unmanned aerial vehicles," in *Proceedings of 2010 American Control Conference*, pp. 4133–4138, June 30-July 2 2010.

[19] J. Dufrene, W.R., "Mobile military security with concentration on unmanned aerial vehicles," in *Proceedings of the 24th Digital Avionics Systems Conference (DASC 2005)*, vol. 2, p. 8, 30 Oct.-3 Nov. 2005.

[20] H. Chao, Y. Cao, and Y. Q.Chen, "Autopilots for small fixed wing small unmanned air vehicles: a survey," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*, pp. 3144–3149, Aug. 2007.

[21] T. W. McLain and R. W. Beard, "Unmanned air vehicle testbed for cooperative control experiments," in *Proceedings of the 2004 American Control Conference*, June 30-July 2 2004.

[22] R. Clothier, A. Harrison, D. Dusha, I. McManus, D. Greer, and R. Walker, "Development of a low-cost UAV system for civilian airspace integration trials," in *Proceedings of AIAC-11 Eleventh Australian International Aerospace Congress*, Mar. 13-17 2005.

[23] D. Jung, E. J. Levy, D. Zhou, R. Fink, J. Moshe, A. Earl, and P. Tsiotras, "Design and development of a low-cost test-bed for undergraduate education in UAVs," in *Proceedings of 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Dec. 12-15 2005.

[24] H. Chao, *Cooperative Remote Sensing and Actuation using Networked Unmanned Vehicles*. Ph.D. dissertation, Utah State University, 2010.

[25] "AggieAir flying circus,"
http://aggieair.usu.edu/aggie.html.

[26] L. Di and Y. Chen, "Autonomous flying under 500 USD based on RC aircraft," in *Proceedings of 2011 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 2011, accepted.

[27] H. Chao, Y. Luo, L. Di, and Y. Chen, "Roll-channel fractional order controller design for a small fixed-wing unmanned aerial vehicle," *Control Engineering Practice*, vol. 18(7), pp. 761–772, 2010.

[28] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: a survey," *International Journal of Control, Automation, and Systems*, vol. 8(1), pp. 36–44, 2010.

[29] "DIYDRONE ArduIMU,"
http://diydrones.com/profiles/blogs/arduimu-now-available.

[30] W. Premerlani and P. Bizard, "Direction cosine matrix IMU: theory,"
http://gentlenav.googlecode.com/files/DCMDraft2.pdf.

[31] "Digi Xtend manual,"
http://www.digi.com/pdf/ds_xtend.pdf.

[32] "Center of gravity calculator,"
http://www.scaleaero.com/CG_Calculator.htm.

[33] "MicroPilot MP2028 autopilot,"
http://www.micropilot.com/products-mp2028g.htm.

[34] "Cloud Cap Technology Piccolo SL autopilot,"
http://www.cloudcaptech.com/piccolo_sl.shtm.

[35] "Procerus Technology Kestrel V2.4 autopilot,"
http://www.procerusuav.com/productsKestrelAutopilot.php.

[36] "Paparazzi TWOG autopilot,"
http://paparazzi.enac.fr/wiki/Twog_v1.

[37] P. Brisset, "The Paparazzi UAV system," Mar. 2010,
http://inc.eng.kmutt.ac.th/pornpoj/Quadrotor/PascalBrisset_InfoIndustrielle_Paparazzi.pdf.

[38] "Gumstix computer-on module,"
http://www.gumstix.com/store/catalog/index.php?cPath=27.

[39] H. Chao, C. Coopmans, L. Di, and Y. Chen, "Comparative evaluation of low-cost IMUs for unmanned autonomous systems," in *Proceedings of IEEE 2010 International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 211–216, 2010.

[40] "Microstrain GX2,"
http://www.microstrain.com/3dm-gx2.aspx.

[41] "Ublox GPS receiver manual,"
http://www.u-blox.com/en/lea-5h.html.

[42] Y. Shan, J. E. Speich, and K. K. Leang, "Low-cost IR reflective sensors for submicrolevel position measurement and control," *IEEE/ASME Transactions on Mechatronics*, vol. 13, pp. 700–709, 2008.

[43] C. L. Wyatt, *Radiometric Calibration: Theory and Methods*. New York: Academic Press, 1978.

[44] G. Benet, F. Blanes, J. Simo, and P. Perez, "Using infrared sensors for distance measurement in mobile robots," *Robotics and Autonomous Systems*, vol. 40, pp. 255–266, 2002.

[45] G. Egan and B. Taylor, "The use of infrared sensors for absolute attitude determination of unmanned aerial vehicles," Technical Report MECSE-22-2006, Monash University, Australia, 2006.

[46] "The electromagnetic spectrum,"
http://science.hq.nasa.gov/kids/imagers/ems/infrared.html.

[47] "FMA Direct Co-Pilot,"
http://www.fmadirect.com/detail.htm?item=1489&section=20.

[48] "Xsens MTi-G IMU,"
http://www.xsens.com/images/stories/products/PDF_Brochures.

[49] "Mean value theorem,"
http://mathworld.wolfram.com/Mean-ValueTheorem.html.

[50] Y. Tarte, "Detection, identification, and compensation of nonlinearities and an experimental verification platform forest nonlinear controllers," Master's thesis, Utah State University, 2006.

[51] L. Di, T. Fromm, and Y. Chen, "A data fusion system for attitude estimation of low-cost miniature UAVs," in *Proceedings of the 2011 International Conference on Unmanned Aircraft Systems*, May 2011.

[52] L. Di, H. Chao, and Y. Chen, "A two-stage calibration method for low-cost UAV attitude estimation using infrared sensor," in *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pp. 137–142, 2010.

[53] J. Munoz and B. Premerlani, "AuduIMU open source project,"
http://diydrones.com/profiles/blogs/arduimu-v2-demo-video.

[54] "Centeye—visual microsensor technology,"
http://www.centeye.com/.

[55] S. M. Oh, S. Tariq, B. Walker, and F. Dellaert, "Map-based priors for localization," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2179–2184, 2004.

[56] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," in *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, pp. 340–345, 2008.

[57] D. Dusha, W. Boles, and R. Walker, "Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow," in *Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pp. 485–492, 2007.

[58] S. Ettinger, M. Nechyba, P. Ifju, and M. Waszak, "Towards flight autonomy: vision-based horizon detection for micro air vehicles," in *Proceedings of the 2002 Florida Conference on Recent Advances in Robotics*, 2002.

[59] S. Fefilatyev, V. Smarodzinava, L. Hall, and D. Goldgof, "Horizon detection using machine learning techniques," in *Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA)*, pp. 17–21, 2006.

[60] T. McGee, R. Sengupta, and K. Hedrick, "Obstacle detection for small autonomous aircraft using sky segmentation," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4679–4684, 2006.

[61] S. Todorovic and M. Nechyba, "Sky/ground modeling for autonomous MAV flight," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1422–1427, 2003.

[62] T. Fromm, L. Di, H. Voos, and Y. Chen, "Visual attitude estimation for low-cost personal remote sensing systems," in *Proceedings of 2011 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 2011, accepted.

[63] P. F. Swaszek and P. Willett, "Parley as an approach to distributed detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 447–457, Jan. 1995.

[64] A. J. Fleming, A. G. Wills, and S. O. R. Moheimani, "Sensor fusion for improved control of piezoelectric tube scanners," *IEEE Transactions on Control System Technology*, vol. 15(6), pp. 1265–6536, Nov. 2008.

[65] "Logitech HD Pro webcam C910," http://www.logitech.com/en-us/webcam-communications/webcams/devices/6816.

[66] R. Teo, J. S. Jang, and C. J. Tomlin, "Automated multiple UAV flight - the Stanford DragonFly UAV program," in *Proceedings of 43rd IEEE Conference on Decision and Control*, Dec. 14-17 2004.

[67] J. How, E. King, and Y. Kuwata, "Flight demonstrations of cooperative control for UAV teams," in *Proceedings of AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, Sep. 20-23 2004.

[68] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robotics and Automation Magazine*, vol. 17(3), pp. 56–65, Sep. 2010.

[69] E. Shaw, H. Chung, J. Hedrick, and S. Sastry, *Cooperative Systems: Control and Optimization*, vol. 588, ch. Unmanned helicopter formation flight experiment for the study of mesh stability. Springer, 2007.

[70] B. Yun, B. M. Chen, K. Y. Lum, and T. H. Lee, "A leader-follower formation flight control scheme for UAV helicopters," in *Proceedings of the IEEE International Conference on Automation and Logistics*, Sep. 2008.

[71] H. Chao and Y. Chen, "Surface wind profile measurement using multiple small unmanned aerial vehicles," in *Proceedings of the 2010 American Control Conference*, pp. 4133–4138, June 30-July 2, 2010.

[72] Y. Chen and Z. Wang, "Formation control: a review and a new consideration," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3664–3669, Aug. 2005.

[73] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27(1), pp. 73–82, 2004.

[74] F. Giulietti, L. Pollini, and M. Innocenti, "Formation flight control: a behavioral approach," in *Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference*, 2001.

[75] G. Hattenberger, R. Alami, and S. Lacroix, "Planning and control for unmanned air vehicle formation flight," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[76] Y. Luo, L. Di, J. Han, H. Chao, and Y. Chen, "VTOL UAV altitude flight control using fractional order controllers," in *Proceedings of 4th IFAC Workshop on Fractional Differentiation and Its Application (FDA 2010)*, 18-20 Oct. 2010.

[77] N. Yoshitani, S. ichi Hashimoto, T. Kimura, K. Motohashi, and S. Ueno, "Flight control simulators for unmanned fixed-wing and VTOL aircraft," in *Proceedings of ICROS-SICE International Joint Conference 2009*, Aug. 18-21 2009.

[78] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2247–2252, Apr. 2005.

[79] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 153–158, Oct.-Nov. 2007.

[80] D. Y. Xue, Y. Q. Chen, and D. Atherton, *Linear Feedback Control: Analysis and Design with MATLAB*. Philadelphia: SIAM Press, 2007.

[81] H. S. Li, Y. Luo, and Y. Q. Chen, "A fractional order proportional and derivative (FOPD) motion controller: tuning rule and experiments," *IEEE Transactions on Control Systems Technology, Digital Object Identifier: 10.1109/TCST.2009.2019120*, vol. 18, no. 2, pp. 516–520, 2010.

Appendices

Appendix A

Multi-UAV Flight Test Preflight Checklist

(1) Upload the new leader and follower codes for all the UAVs.

(2) Actuator Check

  (2.1) Motor mount screws tight.

  (2.2) Servo linkage tight.

  (2.3) Elevons move correctly.

  (2.4) Throttle turns on.

  (2.5) Transmitter can change the flight mode (Auto 1/2,manual).

(3) Navigation Sensor Check

  (3.1) Correct orientation (GCS PFD).

  (3.2) GPS Fix.

(4) Tape Access Patches.

(5) Initialize the formation scheme using the reconfiguration module (Start with the follower 30m below leader).

(6) Walk 25 big steps into the wind after the bungee is fully extended.

(7) Turn on the motor switch right before the launch.

Appendix B

Paparazzi GCS Operation Manual

B.1 Introduction

The objective of this manual is to introduce the GCS operator as to how to manipulate the Paparazzi GCS. However, this manual does not concentrate on the details of Paparazzi GCS software. Instead, it focuses on the manipulation part so that the new GCS operator can avoid making mistakes while in the learning curve. All the manipulations are based on AggieAir platform developed in CSOIS together with UWRL.

Safety of both humans and property is the highest priority for UAV flight tests. The GCS operator needs to cooperate with the safety pilot to achieve safety in conducting the entire flight test. The tasks for the GCS operator during an UAV autonomous flight include: (1) UAV Health Monitoring. The GCS operator needs to monitor the UAV health continuously including sensors, actuators, and communication modules. Then the operator needs to communicate with the safety pilot in case of any emergent situations. (2) UAV Command. The GCS operator needs to command the UAV to different mission blocks based upon the safety and the flight plan.

B.2 UAV Health Monitor

Basics

The GCS operator has to know the UAV autonomous mode, which includes:

(1) Manual mode: the safety pilot controls the UAV through RC transmitter;

(2) Auto 1 mode: a half autonomous mode for tuning, the UAV flies straight by default, but the safety pilot still can control it with RC transmitter;

(3) Auto 2 mode: a fully autonomous mode, the UAV files following the flight plan.

Actuator Health

The actuators of AggieAir include elevon and throttle. To tell if the actuators are working properly, the following are monitored.

(1) Battery voltage. The battery portion shows the voltage for the whole system. The GCS operator needs to direct the UAV to land if the voltage stays below 10.5.

(2) Throttle percentage. The throttle portion fives a percentage representing at how much throttle the UAV is running. It has to be close to 90% for take-off. Specially, the red color means that kill-throttle mode is on, or the propeller is not turning. Check the FAQ part for detailed explanation on kill throttle part.

(3) Altitude/desired altitude. The GCS operator needs to observe the altitude/desired altitude all the time. If a well-tuned UAV cannot track the desired altitude ($\pm$20 m) in the middle of the flight with no other indications, the propeller is probably damaged.

(4) Ground speed. If a well-tuned UAV flies extremely slowly in the middle of the flight (less than 10 m/s in a light wind) with no other indications, the propeller probably is defective.

Sensor Health

The health of the sensor indicates whether the UAV could fly autonomously (Figure B.1), which is critical to the whole system. The GCS operator needs to make sure that the sensors are working correctly before the take-off and during the flight.

(1) GPS health. The GPS information includes longitude, latitude, altitude, velocity, plane updated frequency and accuracy of each. GPS position data (lon, lat, alt) are transmitted to the ground at about 4 Hz. Other GPS parameters need to be given attention.

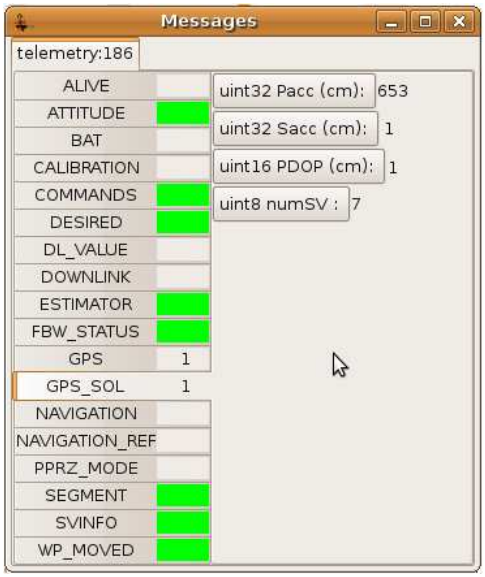


Fig. B.1: Paparazzi GPS health monitor.

(1.1) Position accuracy (Pacc). A valid GPS lock and ¡ 10 m ground Pacc should be checked before take-off. The GCS operator also needs to check the Pacc value periodically in the middle of the flight to make sure it is within limits. The Pacc can be checked in two ways: GCS and message tab, as shown below.

(1.2) Updating frequency. The GPS position data are used to update the GCS map. There is an indication something wrong if the GPS position of the UAV suddenly ceases updating during the flight. It could be gumstix not giving the right GPS data, a GPS hardware problem, or a modem communication problem.

(2) IMU health. The IMU information includes the attitude information of the UAV, namely, roll, pitch, and yaw angles. The reading could be shown either in the PFD section or in the message part. The IMU needs to be checked before the take-off following right, left, up, and down, respectively (Figure B.2).

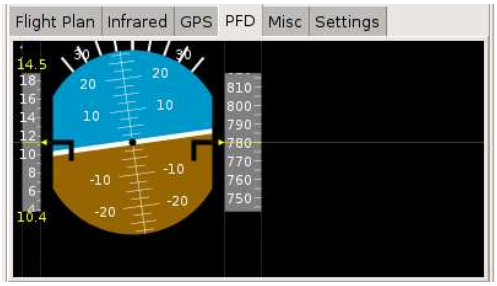


Fig. B.2: Paparazzi IMU health monitor.

Payload Health

The GhostFoto image system requires the monitoring of its status periodically during the flight. Its status can be checked through the message tab the GPS_SOL section.

(1) Sacc represents the status of the left camera.

(2) PDOP represents the status of the right camera.

The status codes are:

(1) 1 $\cdots$ Camera initialized,

(2) 2 $\cdots$ Lens extended,

(3) 4s/5s $\cdots$ Picture taking.

B.3 GCS Commanding

The GCS operator needs to command the UAV as to which navigation block to be covered based upon the collected UAV health information. Inappropriate commands could lead the UAV to crash. Most exceptions or dangerous maneuvers occurs in the take-off or landing parts of the flight. The GCS operator needs to decide when to direct the UAV to landing in the middle of a flight.

(1) Takeoff. Bungee waypoint is crucially related to the autonomous takeoff. If the flight plan is well made, the only thing needs to be edited in the field is the Bungee waypoint.

The GCS operator needs to ensure that the bungee point is correct on the GCS map and that the UAV is in Takeoff block before launching.

(2) Landing. The GCS operator needs to select the AF and TD point so that the landing field is soft and far away from any obstacles. The distance between the AF and TD should be above 150 m and up to 200 m. The GCS operator needs to make sure that the TD point is close to the Ground Station and that the landing trace is into the wind.

(3) Navigation block transition. The GCS operator can double click the block intended for making the transition..

A screen shot picture of the GCS, indicating where the information location is shown in Figure B.3.

B.4 Emergency Response

This section outlines how the GCS operator should perform in case of middle of some emergency circumstances that could result in a crash the UAV.

(1) Take-off failure. Depending on whether it is autonomous takeoff or manual takeoff, there are different approaches to rectify the problems. For autonomous takeoff, if the throttle fails to turn on, the safety pilot needs to switch back to manual mode. Then the GCS operator manually turn off the kill-throttle switch. The UAV can then be switched back to Auto 2. For manual takeoff, it depends on the skills and experience of the safety pilot. Wind also plays an important role during takeoff, so the plane always needs to be launched into the wind.

(2) Landing failure. Either because the UAV couldn't get down or glide too far. If this occurs in the final approach of the landing, only the safety pilot can rectify the situation by switching to manual because the throttle will be killed using the new landing routine code. A most important thing is that the plane needs to be landed into the wind.

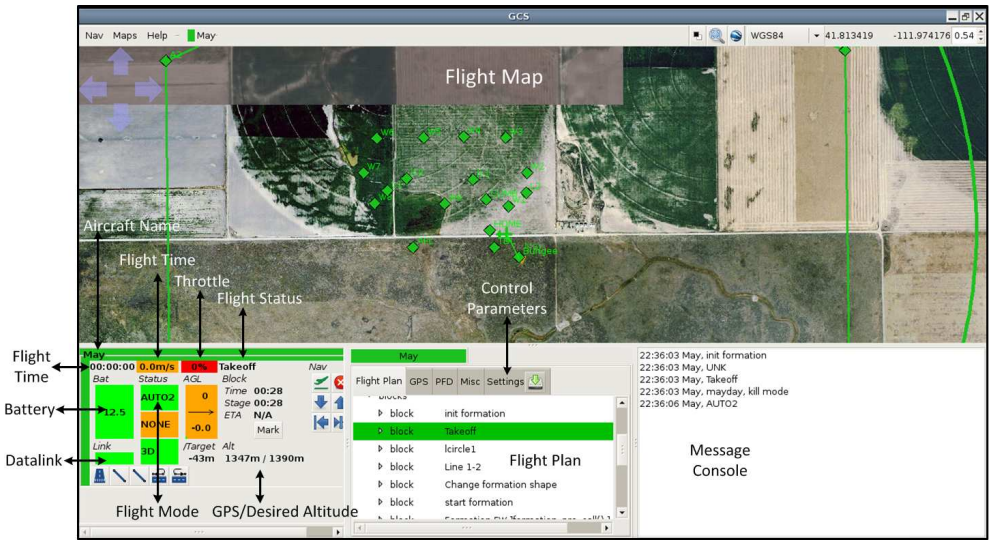


Fig. B.3: Paparazzi GCS interface.

(3) Communication problem. If datalink gets defective (hard to commit changes of the waypoints) or losses (color changed from green to red, increasing number appears), the ground modem antenna should be adjusted and the safety pilot should be informed to manually take over the plane.

(4) GPS problem. Sometimes during flight, the GPS signal might get defective, and the operator will observe long delay of the plane's moving trace. At this moment, the operator needs to try to bring the plane back to the Standby point and check both GPS accuracy and the updating frequency. If either information looks strange, the safety pilot needs to be informed to take over the plane.

B.5 FAQ

(1) What are the three most frequent and important places to monitor?

(1.1) Altitude (Real altitude/Desired altitude). Make sure the actual altitude is safe and valid at every stage. Autonomous landing altitude should be about 15 to 18 m.

(1.2) Flight map. Make sure the distance between the plane and the Ground Station is excessive. Then bring the plane back to Standby if the plane flies out of the boundary and out of the sight of the safety pilot.

(1.3) Battery voltage. This directly relates to autonomous takeoff and how much flight time remains. When the battery voltage is below 11.3 V after the previous flight, batteries have to be recharged or replaced. If the battery voltage drops to 10.5 V during flight, the plane has to land.

(2) What does the kill throttle mean and how can it be avoided?

(2.1) Before taking off, the throttle of the plane is always off, so the operator needs to watch the throttle percentage window when launching the plane. If the throttle window shows red color instead of brown, this means kill throttle is still on. Therefore, the operator needs to immediately tell the safety pilot to manually take off the plane, change the flight plan block to Standby, and manually turn the kill throttle off through Figure B.4.

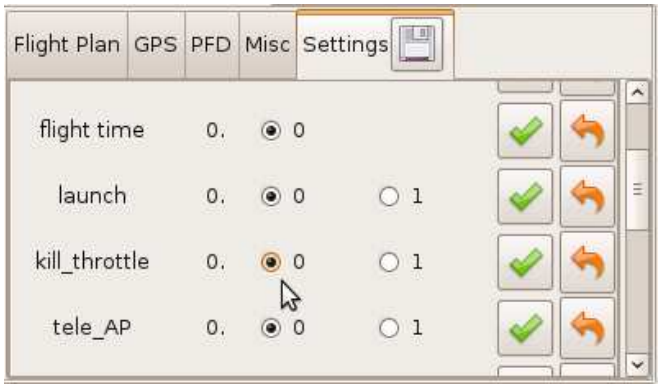


Fig. B.4: Kill throttle switch.

Appendix C

Formation Flight Software Setup

Airframe File

The following formation parameters (Figure C.1) need to be specified for each UAV that will participate in the formation flight. Each UAV's parameters can be different.

For software in the simulation setup, the user needs to add traffic_info.c, tcas.c, formation.c if the users want to activate these functions (Figure C.2(a)).

For actual flight, the user needs to add the section (Figure C.2(b)) in the airframe configuration file to activate the formation and traffic collision avoidance functions.

Flight Plan File

The user needs to add the formation.h file in the flight plan file in order to run the formation functions (Figure C.3).

It is necessary to initialize the formation, and the users needs to define how many UAVs are in the configuration (Figure C.4(a)), define their ID, X (longitude), Y (latitude), and Z (altitude) distances regarding the leader. The user also needs to specify the formation mode, where 1 is relative and 0 is global, and it means follower will fly behind leader at 10 m if the X is set as 10 and mode is 1. Otherwise, there will be a change depending on the heading of the leader when mode is set as 0.

When all the UAVs are ready for formation flight, user activates this block (Figure C.4(a)), which is in the leader's flight plan. The UAVs will start doing formation flight. If the UAVs need to transform their formation schemes, they can use this block (Figure C.4(b)) to convert.

Whenever switching to a new block of formation flight plan, add the post_call and pre_call functions (Figure C.5) to make sure the UAVs stay in the formation mode.

When formation flight is completed and trying to get the UAV out of the formation mode, use the stop_formation function (Figure C.5).

Setting File

By adding the following blocks (Figure C.6) into the setting file, the GCS operator is able to tune the formation control parameters in real time, the operator can also adjust the range of these parameters, it must be remembered whenever that something here is changed, the Paparazzi must be rebuilt for activation.

Control Panel File

In order to simulate multiple UAV formation flight, the user needs to construct as many UAVs as needed. An example is building two UAVs and with UAV assigned different names to enter into the current airframe configuration list (Figure C.7).

If the formation controller code adding new functions need to be modified, the user needs to be familiar with Paparazzi software. Based upon all the procedures enumerated above, a new user should be able to start the formation simulation study and even actual flight tests. After all these steps have been performed, the user should see similar figures as follows (Figure C.8) in the Paparazzi GCS.

```
<!--Formation flight portion1-->
  <section name="FORMATION" prefix="FORM_">
    <define name="CARROT" value="3." unit="s"/> <!-- carrot distance for followers -->
    <define name="POS_PGAIN" value="0.02"/> <!-- coef on position error -->
    <define name="SPEED_PGAIN" value="0.2"/> <!-- coef on speed error -->
    <define name="ALTITUDE_PGAIN" value="0.03"/> <!-- coef on altitude error -->
    <define name="PROX" value="60." unit="m"/> <!-- proximity distance -->
    <define name="MODE" value="1"/> <!-- mode 0 = global, 1 = local -->
```

Fig. C.1: Initial formation parameters.

```
# Config for SITL simulation/////////////////////For formation
include $(PAPARAZZI_SRC)/conf/autopilot/sitl.makefile
sim.CFLAGS += -DCONFIG=\"tiny.h\" -DAGR_CLIMB -DLOITER_TRIM -DTRAFFIC_INFO -DALT_KALMAN -DTCAS -DFORMATION
sim.srcs += nav_survey_rectangle.c nav_line.c traffic_info.c tcas.c OSAMNav.c formation.c
```

(a) Simulation formation setup.

```
# formation////////////////////////
ap.CFLAGS += -DFORMATION -DTRAFFIC_INFO -DTCAS
ap.srcs += traffic_info.c formation.c snav.c tcas.c
```

(b) Experimental formation setup.

Fig. C.2: Airframe file formation setup.

```
#include "nav_line.h"
#include "datalink.h"
#include "traffic_info.h"
#include "OSAMNav.h"
#include "formation.h"
#include "airframe.h"
```

Fig. C.3: Header files for formation flight plans.

```
    <block name="init formation">
<!--Edit by Dee for the real time tuning on formation shape-->

      <call fun="add_slot(2, 0, 0, 0)"/>
      <call fun="add_slot(188, form_s_e_1, form_s_n_1, form_s_a_1)"/>
      <set value="1" var="form_mode"/>
      <set value="2" var="leader_id"/>
      <set value="NAV_MODE_ROLL" var="nav_mode"/>
    </block>
```

(a) Formation flight initialization function.

```
<!--Edit by Dee for the real time tuning on formation shape-->
   <block name="Change formation shape">
      <call fun="add_slot(2, 0, 0, 0)"/>
      <call fun="add_slot(188, form_s_e_1, form_s_n_1, form_s_a_1)"/>
      <set value="1" var="form_mode"/>
      <set value="2" var="leader_id"/>
      <set value="NAV_MODE_ROLL" var="nav_mode"/>
    </block>
<!---->

    <block name="Follow_formation">
      <call fun="start_formation()"/>
      <call fun="formation_flight()"/>
    </block>
```

(b) Change formation scheme and start formation flight.

Fig. C.4: Formation initialization and reconfiguration.

```
<block name="start formation">
  <call fun="start_formation()"/>
  <deroute block="Formation FW"/>
</block>

<block name="Formation FW" post_call="formation_flight()" pre_call="formation_pre_call() ">
  <go approaching_time="1" wp="W1"/>
  <go approaching_time="1" from="W1" hmode="route" wp="W2"/>
  <go approaching_time="1" from="W2" hmode="route" wp="W3"/>
  <go approaching_time="1" from="W3" hmode="route" wp="W4"/>
  <go approaching_time="1" from="W4" hmode="route" wp="W5"/>
  <go approaching_time="1" from="W5" hmode="route" wp="W6"/>
  <go approaching_time="1" from="W6" hmode="route" wp="W7"/>
  <go approaching_time="1" from="W7" hmode="route" wp="W8"/>
  <go approaching_time="1" from="W8" hmode="route" wp="W9"/>
  <deroute block="Formation FW"/>
</block>
<block name="Circle for search" post_call="formation_flight()" pre_call="formation_pre_call()">
  <circle alt="1500" radius="75" wp="suvery entry"/>
</block>
<block name="Search area" post_call="formation_flight()" pre_call="formation_pre_call()">
  <survey_rectangle grid="150" wp1="S1" wp2="S3"/>
</block>
<block name="stop formation">
  <call fun="stop_formation()"/>
</block>
```

Fig. C.5: Formation flight plan blocks.

```
<dl_settings NAME="formation">
<dl_setting MAX="24" MIN="0" STEP="1" VAR="leader_id" module="formation"/>
<dl_setting MAX="100" MIN="-200" STEP="10" VAR="form_s_e_l"/>
<dl_setting MAX="100" MIN="-100" STEP="10" VAR="form_s_n_l"/>
<dl_setting MAX="100" MIN="-50" STEP="10" VAR="form_s_a_l"/>

<dl_setting MAX="0.1" MIN="0.01" STEP="0.01" VAR="coef_form_pos" shortname="coef_form_pos_error" module="formation"/>
<dl_setting MAX="1" MIN="0.1" STEP="0.1" VAR="coef_form_speed"  shortname="coef_form_speed_error" module="formation"/>
<dl_setting MAX="0.5" MIN="0.01" STEP="0.02" VAR="coef_form_alt"  shortname="coef_form_alt_error" module="formation"/>
<dl_setting MAX="100" MIN="40" STEP="10" VAR="form_prox"  shortname="prox distance" module="formation"/>


</dl_settings>
```

Fig. C.6: Setting file for formation reconfiguration and tuning.

```
<session name="MultiUAV OSAM 3">
  <program name="Simulator">
    <arg flag="-a" constant="Phoenix"/>
    <arg flag="-boot"/>
    <arg flag="-norc"/>
    <arg flag="-noground"/>
  </program>
  <program name="Simulator">
    <arg flag="-a" constant="May"/>
    <arg flag="-boot"/>
    <arg flag="-norc"/>
    <arg flag="-noground"/>
  </program>
   <program name="Simulator">
    <arg flag="-a" constant="Dimon"/>
    <arg flag="-boot"/>
    <arg flag="-norc"/>
    <arg flag="-noground"/>
  </program>

  <program name="Server">
    <arg flag="-n"/>
  </program>
  <program name="Messages"/>
  <program name="GCS">
    <arg flag="-track_size" constant="10"/>
  </program>
</session>
```

Fig. C.7: Simulation setup for three UAV formation flight.

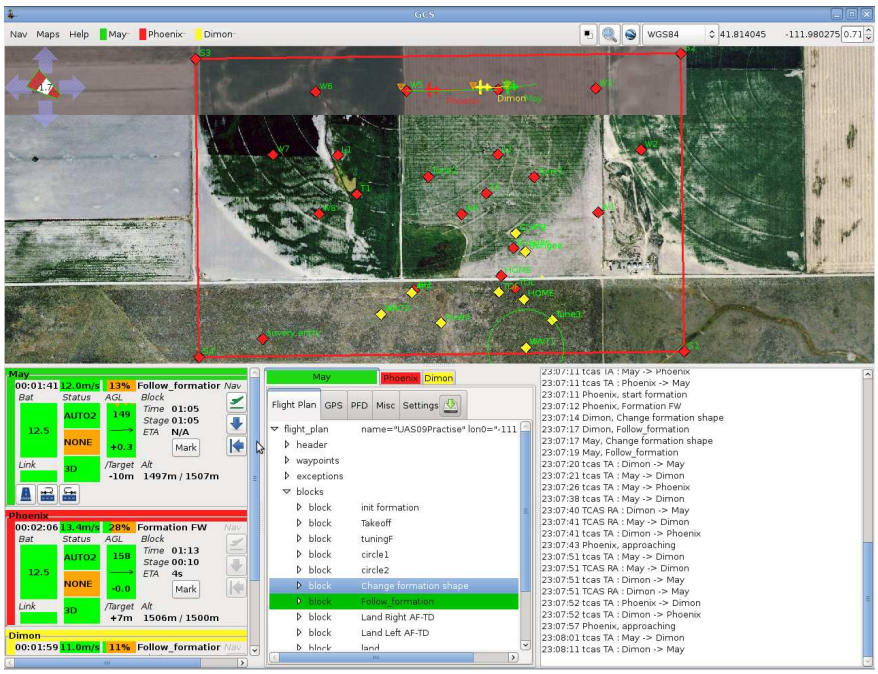


Fig. C.8: Simulation with three UAV formation flight.