AUTONOMOUS AERIAL MANIPULATION USING A QUADROTOR

by

Vaibhav Ghadiok

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

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Wei Ren Major Professor	Dr. Todd K. Moon 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

Autonomous Aerial Manipulation Using a Quadrotor

by

Vaibhav Ghadiok, Master of Science

Utah State University, 2011

Major Professor: Dr. Wei Ren
Department: Electrical and Computer Engineering

This paper presents an implementation of autonomous indoor aerial gripping using a low-cost, custom-built quadrotor. Such research extends the typical functionality of micro air vehicles (MAV) from passive observation and sensing to dynamic interaction with the environment. To achieve this, three major challenges are overcome: precise positioning, sensing and manipulation of the object, and stabilization in the presence of disturbance due to interaction with the object. Navigation in both indoor and outdoor unstructured, Global Positioning System-denied (GPS-denied) environments is achieved using a visual Simultaneous Localization and Mapping (SLAM) algorithm that relies on an onboard monocular camera. A secondary camera, capable of detecting infrared light sources, is used to estimate the 3D location of the object, while an underactuated and passively compliant manipulator is designed for effective gripping under uncertainty. The system utilizes nested Proportional-Integral-Derivative (PID) controllers for attitude stabilization, vision-based navigation, and gripping. The quadrotor is therefore able to autonomously navigate, locate, and grasp an object, using only onboard sensors.

(127 pages)

To my parents

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) have found potential applications in both military and civilian domains. Military applications include border patrolling, mine detection, reconnaissance, etc., while civilian applications are in disaster management, bridge inspection, search and rescue, etc. With advances in sensor and battery technology over the last few years, UAVs have become more accessible for use in a variety of applications. This thesis presents a specific type of a Vertical Take-off and Landing (VTOL) aircraft called the quadrotor.

The quadrotor is an underactuated mechanical system that has six degrees of freedom (DOF) but only four control inputs namely roll, pitch, yaw, and thrust and is a dynamically unstable system. However, this gives it the ability to maneuver in tight indoor spaces. The concept of a quadrotor is not new, with the first reference dating back to the Breguet-Richet Quadrotor helicopter Gyroplane No. 1 built in 1907 [1]. The first known hover is said to have occurred in October, 1922 [2].

This thesis presents the design and development of a Micro Aerial Vehicle (MAV) capable of autonomous navigation in Global Positioning System-denied (GPS-denied) environments and aerial manipulation.

1.1 Motivation

Imagine a scenario, such as the Fukushima Daiichi nuclear power station disaster, where access to humans is denied. This presents an ideal situation, where robots could be used to help in controlling the situation and indeed robots were deployed. A Global Hawk UAV was deployed within the first couple of days to provide aerial imagery of the nuclear power station [3]. Global Hawk is an autonomous, jet-powered UAV with a sensor package

that includes synthetic-aperture radar, electro-optical and infrared sensors. It is shown in Fig. 1.1. While a lot of sensory data in the form of both optical and infrared images was collected, the need of the hour was indoor surveillance to get a better idea of the situation inside. Some of the pictures returned by the Global Hawk are shown in Fig. 1.2.

Ground Robots are one option to use in such a scenario, however, the terrain was rough and had gaps and obstacles. The ceiling had collapsed in a few buildings and the ground was littered with scattered debris and mangled steel beams. This is a very challenging environments for any kind of ground robot whether using wheels, legs, tracks, or a combination of the former actuation capabilities. Ground robots such as the iRobot's 510 Packbot and 710 Warrior were deployed in buildings where the terrain was considered negotiable [4]. Figure 1.3(a) shows one such iRobot 510 Packbot in operation inside one of the reactor buildings.

Another robot used was the T-Hawk MAV from Honeywell shown in Fig. 1.3(b), that was flown in close proximity to the reactor buildings to provide a better picture of the damage [5]. Some of the pictures taken by the MAV are shown in Fig. 1.4.

These robots needed to be remote controlled and relied on either wireless communication in the case of the MAV or a tether in the case of ground robots. The thick walls and metal used in the reactor buildings precluded the use of wireless transmission when operating inside the reactors. Therefore, the ground robots used an optical fiber tether to communicate with the base station at the cost of limited range and mobility. This also

Fig. 1.1: Global Hawk.

(a) (b)

Fig. 1.2: Photographs of the Fukushima nuclear disaster taken from a Global Hawk.

restricted the use of the T-Hawk to only outside the reactor buildings.

What was required?

While these robots mention above performed appreciably, the need was for:

1. Indoor surveillance,

2. Autonomous operation inside buildings,

3. Ability to open and close valves and activate pumps to get the reactor under control,

(a) An iRobot 510 Packbot inside a power station building.

(b) T-Hawk MAV from Honeywell.

Fig. 1.3: Robots deployed at the Fukushima Daiichi nuclear power station.

4. Ability to deploy sensors to measure radiation inside the buildings.

This thesis tries to address some of these issues to varying degrees of extent.

1.2 Problem Statement

The question this thesis tries to answer is, "Is it possible to obtain autonomy in GPS-denied environments and demonstrate manipulation capability on a small, light-weight MAV?" The problem is made even more challenge by having the restriction of using only low-cost sensors.

1.3 A Brief Discussion on Quadrotors

Quadrotors are preferred due to their simple mechanical structure as opposed to traditional helicopters that need complex mechanical control linkages for rotor actuation. Each of the four arms of the quadrotor is mounted with a motor coupled to a fixed-pitch propellor. The use of four motors ensures that individual motors are smaller than the equivalent main rotor on a helicopter, relative to the airframe size. All this leads to robustness and modularity resulting in low maintenance costs. It also gives rise to a simpler dynamic model enabling the precise control of quadrotors. Moreover, the quadrotor can be enclosed within a protective shroud for safety. This is especially important if quadrotors are to be flown indoors. However, some of the disadvantages of quadrotors are their high energy consumption

(a) (b)

Fig. 1.4: Photographs of the Fukushima nuclear disaster taken from separate vantage points using the T-Hawk.

and poor survivability if one of the arms or propellors is damaged.

1.4 Commercially Available Quadrotors

Typically quadrotors available commercially are equipped with an attitude stabilization system coupled with GPS. Some examples of GPS equipped commercially available quadrotors are the Ascending Technologies Pelican, Microdrones MD-200, and the Mikrokopter [6–8]. The first two systems are an order of magnitude more expensive than the system presented in this thesis. While the third system is relatively cheaper than the first two, it has no indoor navigation capabilities. There are other quadrotors such as the Quanser Q-ball that are again more than an order of magnitude more expensive [9]. The AR Drone can maintain a vision-based hover, however, it has no navigation capabilities and payload capacity [10].

1.5 Limitations of Off-the-Shelf Quadrotors

A major problem in stabilizing a quadrotor is translational drift. While a three-axis inertial measurement unit (IMU) can stabilize the craft so that it stays level while in fight, outside forces may exert a horizontal velocity upon the aircraft causing it to translate without changing pitch, roll, or yaw. Horizontal velocity is not detected by the IMU, and so the craft may be perfectly level and still manage to coast across the room and crash into a wall. Translational drift can also be due to sensor resolution or sensor noise. One problem faced commonly is that slight tilts of the quadrotor one way or another may be imperceptible to the IMU. Therefore, the quad can appear to be in translational drift, which is really a slight angle tilt caused internally and not externally.

Indoor stabilization presents an additional challenge by limiting the kind of sensors that can be used. In an outdoor environment, one has the advantage of using magnetometer, thermopile, and GPS sensors for additional sensory perception. However, these cannot be employed in an indoor environment. Traditionally, quadrotors flying indoors have relied on indoor localization systems such as as the Vicon Motion Capture System [11]. These can provide sub-mm accuracy in stationary conditions but are extremely expensive costing about

$150,000 [11]. Lately, research has focused on quadrotors with monocular or stereoscopic vision to solve the localization problem.

1.6 Proposed Solution

This thesis proposes a custom-built quadrotor measuring 50 cm in diameter and weighing 1.4 Kg., equipped with low-cost altitude and vision sensors and capable of autonomous navigation in indoor and outdoor unstructured environments. The cost-sensing equation is shown in Fig. 1.5. It performs Simultaneous Localization and Mapping (SLAM) in a GPS-denied environment using techniques that can be implemented on-board the quadrotor with available technology. It can hover precisely and has strong disturbance rejection capabilities. Both these qualities are essential for a quadrotor operating inside a collapsed structure. The quadrotor can gracefully transition from outdoor to indoor environments as may be required in the scenario above.

Although SLAM has been successfully applied in a number of ground and underwater vehicles, applying it to UAVs is challenging. While ground vehicles can get fairly accurate odometry through optical encoders, no such sensing modality exists in MAVs. They are also restricted to use light-weight and low-power sensors. Moreover, the quadrotor in particular is an dynamically unstable MAV that has six degrees of freedom (DOF) and cannot stop and revaluate its pose. This requires the use of 3D SLAM. The constraint of using relative lower quality Microelectromechanical systems-grade (MEMS-grade) sensors makes the problem even more challenging. Data obtained from such sensors is noisy and needs to be filtered.

Fig. 1.5: Total cost for all the sensors.

Camera is the navigation sensor of choice due to it being light-weight, low-cost, low-power, dual-use (surveillance and navigation) sensor, and has the capability to provide rich information. However, cameras have limited field-of-views (FOV), cannot operate in the dark and require high computational requirements for extracting useful information from the images.

The quadrotor developed in this thesis is shown maintaining a precise hover in an indoor unstructured environment in Fig. 1.6.

1.7 Outline of Chapters

Chapter 2 describes the platforms leading up to the current system and discusses the drawbacks of each approach.

Chapter 3 introduces the dynamic model of a quadrotor and provides a detailed overview of the airframe, mechanical setup, sensors, computation hardware, and the communication system. It also discusses the integration of these components in the final system.

With the hardware ready, the first requirement is the stabilization of attitude. Chapter 4 discusses filtering, attitude estimation, and control. It presents results of attitude and altitude stabilized flights.

Chapter 5 presents experimental results for autonomous hover and path following in indoor GPS-denied environments. The chapter introduces the challenges of indoor navigation and presents possible approaches to solve the problem. It presents a solution in the form of Simultaneous Localization and Mapping using a camera and discusses how it is adapted to the case of the quadrotor. This chapter introduces a novel nonlinear controller to perform robust navigation. It presents results of the quadrotor navigating both indoors and outdoors and also demonstrates the disturbance rejection characteristics of the quadrotor.

The capabilities of the system are expanded with the addition of an IR camera and a gripper to the quadrotor to enable autonomous aerial gripping while hovering. The addition of a guidance loop, along with experimental results for aerial manipulation are presented in Chapter 6.

Finally, Chapter 7 presents concluding remarks and a discussion on future work.

Fig. 1.6: Quadrotor presented in this thesis demonstrating autonomous flight in an indoor unstructured environment.

Chapter 2

Initial Approaches

This chapter gives a brief overview of attempts at building an autonomous quadrotor before arriving at the current approach. The drawbacks of each approach are discussed and a brief sketch of all the intermediate platforms is given. This chapter aims to succinctly describe all the challenges faced and lessons learned from the initial approaches to achieving autonomous indoor flight.

2.1 Draganflyer

The first attempt at achieving autonomous flight involved automating a commercially available radio-controlled quadrotor called the Draganflyer [12]. This was a platform already available in the lab and was therefore the platform of choice. A schematic diagram of the intended setup, shown in Fig. 2.1, was given to be implemented. As shown in Fig. 2.1, sensor data was acquired using Analog-to-digital Converters (ADC) on the robostix. This data was polled by the gumstix over I^2C and sent to a host computer over Wi-Fi. The host computer was responsible for running estimation and control algorithms and interfaced with the trainer port on the Radio Controller using a Universal Serial Bus (USB) to Pulse Position Modulation (PPM) Converter. The input to the controller is Pulse Position Modulated signals for controlling thrust, roll, pitch, and yaw. However, there were numerous drawbacks with this approach and a stable flight was not achieved.

2.1.1 Drawbacks

1. Wi-Fi - The delay over the Wi-fi (IEEE 802.11 a/b/g) standard as opposed to Ethernet (IEEE 802.3) is much higher, highly variable, and uncharacterized. In our case, the delays ranged from 6 ms to 16 ms.

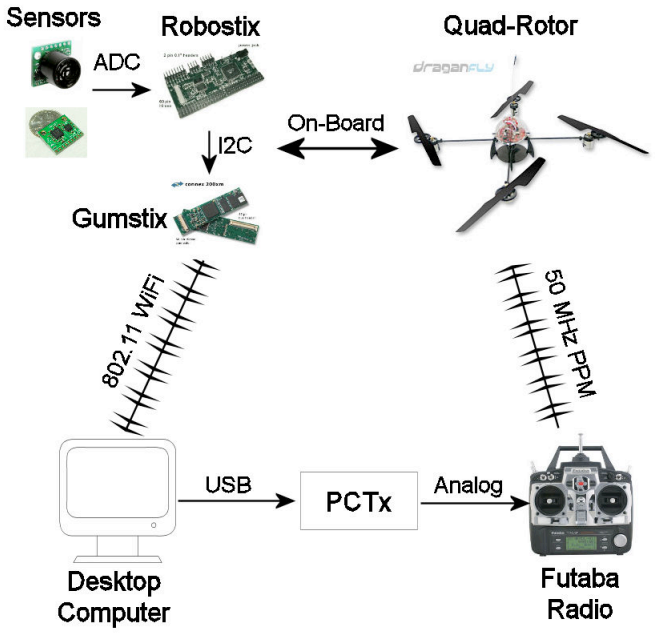


Fig. 2.1: Communication paths.

2. Transmission Control Protocol (TCP) - Unsuitable for real-time operation due to many overheads. User Datagram Protocol (UDP) is a more suitable protocol, but does not solve all the problems.

3. PCTx - The Endurance R/C PCTx module used to interface the host computer with the radio controller has unknown delays. It restricted the control update rate to no higher than 25 Hz. This was a severe impediment, especially given the fact that the Radio Controller (Futaba T7CAP) was capable of transmitting at a peak of 55.5 Hz, which is more than double the rate the PCTx would allow us to achieve.

4. Coarse-grained Control - The PCTx setup only allows sending of values ranging from 100-200 thereby providing coarse-grained control. For a highly-dynamic system such as this, the design of control laws becomes harder given this constraint.

5. Scheduling Delays - The host machine was running Windows and was therefore susceptible to scheduling delays by the Operating System. The gumstix is also a multi-tasking Single Board Computer (SBC) running Linux which could introduce its own scheduling delays, however, these are minor.

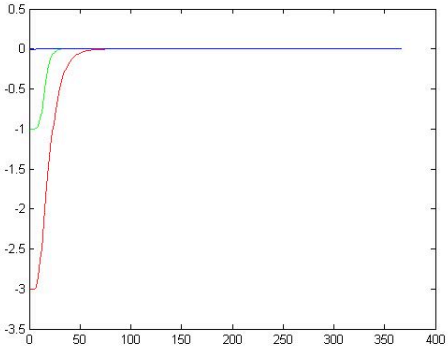
For implementing a closed-loop control for a system as dynamic as a quadrotor, delays can be catastrophic and can lead to system instability. As can be seen in Fig. 2.1, the entire control loop passes through four communication channels and three computational devices out of which two are multi-tasking devices. A point to note here is that the enemy is latency and not data throughput rates, though throughput rate of I^2C did became a concern later on. In essence, what we discovered was that closed-loop attitude control over Wi-Fi is an implausible idea.

2.1.2 Simulation with Delay in Control Loop

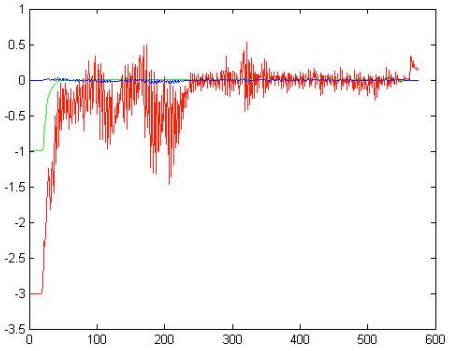
In order to validate this hypothesis, the mathematical model of the quadrotor described in Chapter 3 was simulated using Matlab/ Simulink. No sensor or actuator noise was assumed and Proportional-Derivative (PD) controllers were chosen for attitude stabilization.

Figure 2.2(a) shows the convergence of roll, pitch, and yaw angles to zero starting from arbitrary positions. Figures 2.2(b) and 2.2(c) show even in the case of such an ideal system, a delay of 17 ms causes jitters and increasing the delay to 22 ms leads to divergence. In conclusion, the most obvious next step to eliminate a majority of the delays was to implement onboard control.

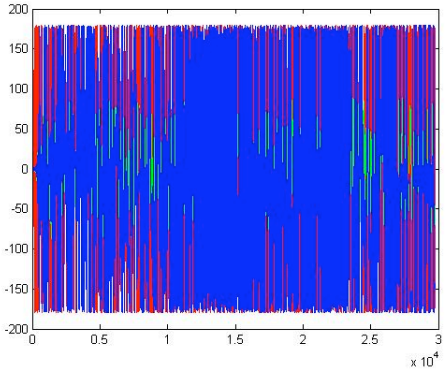
In order to have 3D visualization, the Matlab program was interfaced with an open-source flight simulator called Flightgear [13]. The Aerospace Blockset in Matlab provides an interface block to send the full state vector of the quadrotor to the flight simulator. The x and y co-ordinates of the quadrotor needed to be converted to latitude and longitude in order to achieve this. Since, a model of a quadrotor was not available, an inbuilt model of a helicopter (Eurocopter Bo105) is used instead.



(a) No delay.



(b) 17 ms delay.



(c) 22 ms delay.

Fig. 2.2: Convergence of roll, pitch, and yaw angles with different amounts of delay.

2.2 Testing

Initially a testing rig was designed to test the quadrotor stabilization. Figure 2.3 shows the picture of an initial test rig. The test rig was designed so as to decouple the axes to enable the control of only one axis at a given time. One critical flaw in this design was the assumption that the quadrotor rotated around the center of the frame. In fact, the quadrotor rotates about the actuation plane and not the center of the frame. The actuation plane is the rotor plane in the case of a quadrotor.

The testing rig was discarded in favor of testing in more realistic conditions by suspending the quadrotor from the end of a horizontal pole, with both roll and pitch axes being controlled. However, with this setup, the quadrotor was not kept absolutely flat; the restraining cables would often induce vibrations by constraining the movement of the

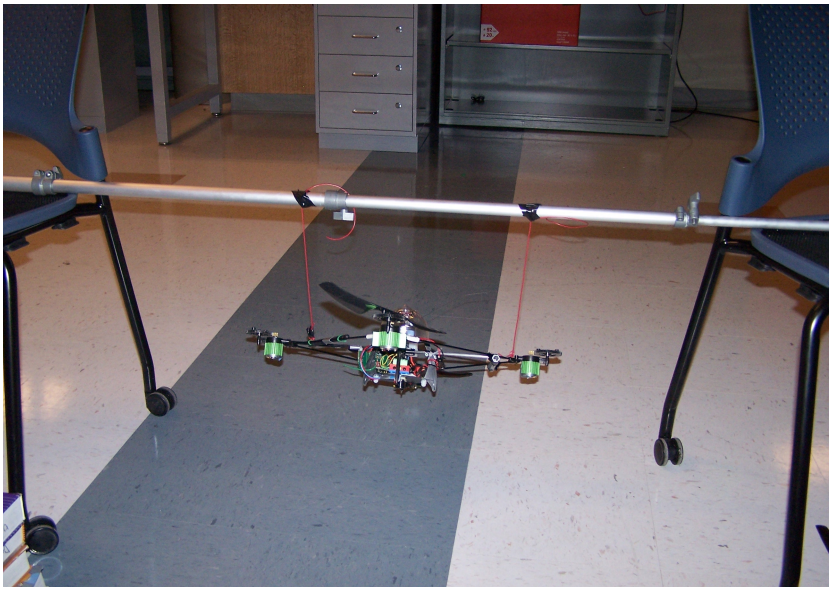


Fig. 2.3: An initial test rig for the quadrotor.

quadrotor. With either of these setups, testing at high throttles was not possible. Finally, a protective shrouding was built for the quadrotor to enable free flight tests.

2.3 Onboard Control

The simplest way to implement onboard control was to send Pulse Width Modulation (PWM) commands directly to motor controllers onboard the quadrotor. However, after careful evaluation, it was realized that this was not possible without modifying the onboard circuitry. Instead, it was decided to feed a signal exactly like the one transmitted by the Radio Controller to the quadrotor. A schematic of the onboard control is shown in Fig. 2.4.

The waveform generated by the radio controller was observed on an oscilloscope and replicated on a PWM pin on the Robostix. The on-board receiver is bypassed and this signal is directly fed to the demodulator.

The waveform generated by the Robostix is shown in Fig. 2.5. It is a Pulse Position Modulated (PPM) signal with a duration of 18 ms resulting in an update rate of 55.5 Hz. It can send six channels: roll, pitch, throttle, yaw, thermopile, and the last channel is unused. Each channel remains low for a fixed duration of 0.5 ms and high for anywhere between

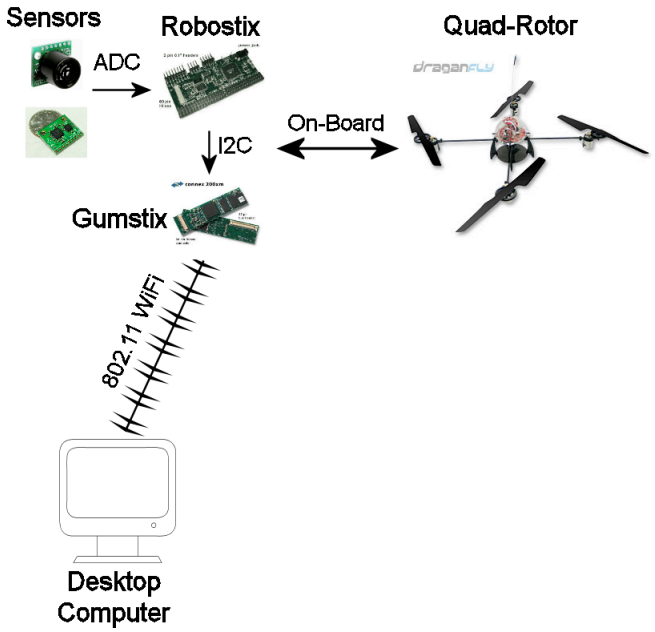


Fig. 2.4: Schematic of onboard control.

0.5 - 1.5 ms depending on the actuation value commanded. For the roll, pitch, and yaw axes a high signal for a duration of 1 ms signifies no control output and a high signal for a duration greater or less than 1 ms signifies a control signal of magnitude ||high_duration - 1 ms|| in one direction or the other. The thermopile sensor aids in stabilization outdoors and is used to measure small differences in temperature above and below the quadrotor body to determine orientation. This is turned off with a 0.5 ms duration signal and turned on by keeping the signal high for 1.5 ms.

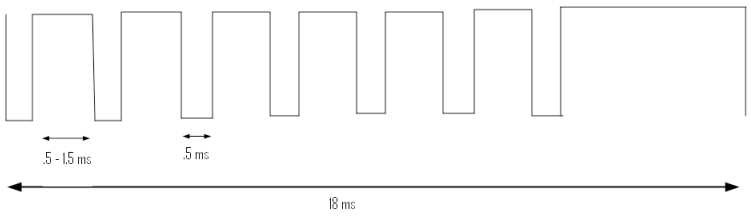


Fig. 2.5: Waveform generated by the Robostix for actuating the quadrotor.

The attitude estimation and control was implemented on the Gumstix onboard the quadrotor and the frequency of the attitude control loop was 55 Hz. Moreover, this setup significantly reduced delays at every stage in the system. There was one additional benefit of finer-grained control with this implementation. Actuation could be commanded on a range of 1000 steps (0.5 ms - 1.5 ms) as opposed to the 100 steps that were available earlier (100 - 200). With these changes, the quadrotor could be stabilized on a fingertip, but the goal of free-flight had still not been achieved.

2.4 Onboard Control Using a Custom-Designed Motor Controller Board

One of the drawbacks of the system from the beginning was the fairly low-bandwidth attitude loop. The only way to rectify this problem was to build custom motor controller that could be commanded using the PWM pins on the Robostix. This would replace the onboard circuitry on the quadrotor used to actuate the motors. A power MOSFET along with a flywheel diode was used to construct a simple motor controller operating at a frequency of 16 kHz for smooth operation of the brushed motors. This circuit is shown in Fig. 2.6. This circuit was replicated to control all four motors.

This lead to an increase of four times in the bandwidth of the control loop making the quadrotor more responsive and led to somewhat better free flights but not by much. The platform with the motor controller board and a landing gear is shown in Fig. 2.7.

With added components, multiple boards and landing gear to protect the quadrotor

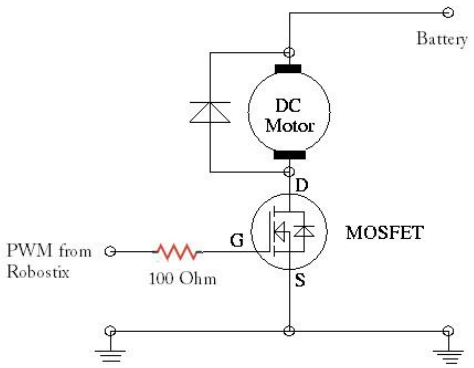


Fig. 2.6: Circuit diagram for the motor controller for a single motor. Four such circuits were assembled on a board.

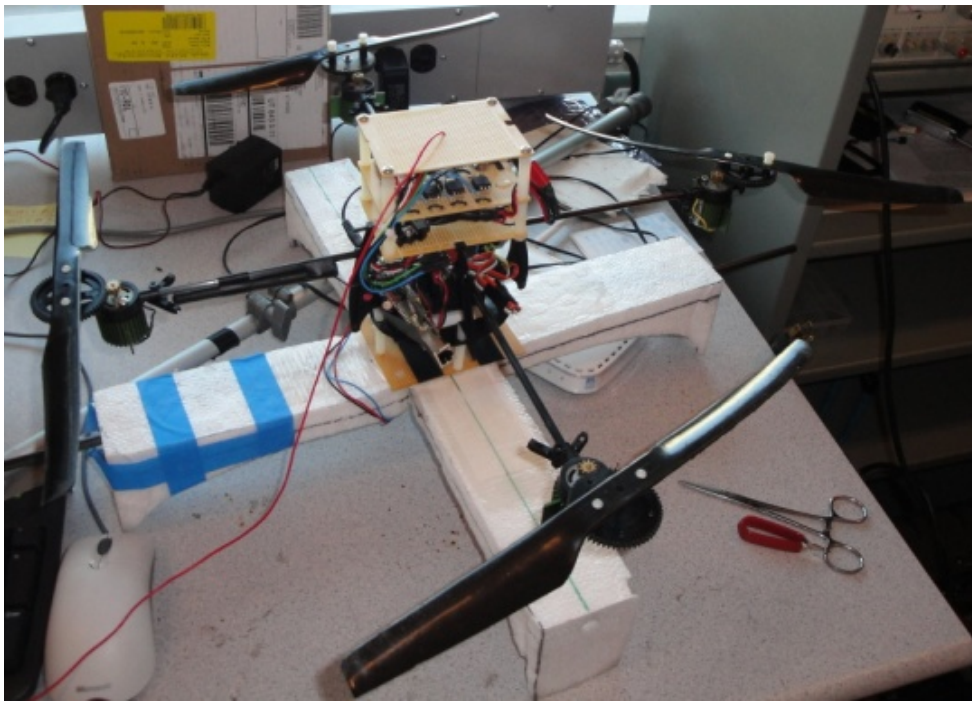


Fig. 2.7: Draganflyer retrofitted with a custom motor controller board and a landing gear.

from crashes, the thrust to weight ratio started to approach unity making the Draganflyer an unviable platform. The total weight of the Draganflyer plaform was up to 800 g while the thrust produced by the motors was only 950 g. The propellors produced almost 25% less thrust at the given altitude of 4500 feet as compared to sea level and this further compounded the problem. Moreover, the brushed DC motors used are inefficient. All this led to the abandoning of the Draganflyer platform and a decision to use Brushless DC motors to provide superior thrust to weight ratio was made.

2.5 Custom Frame

The custom frame was constructed using Aluminum to keep the weight low. Aluminum sheets were cut into squares to support the core structure by sandwiching square aluminum tubes used as arms. The plates and tubes were hand-drilled using a drill press to allow fastening them using screws. A picture of the frame can be seen in Fig. 2.8.

This frame coupled with the brushless motors was able to achieve free-flight in an area of 50 m^2 for up to 20 seconds, but not without pitfalls. It was fraught with stabilization

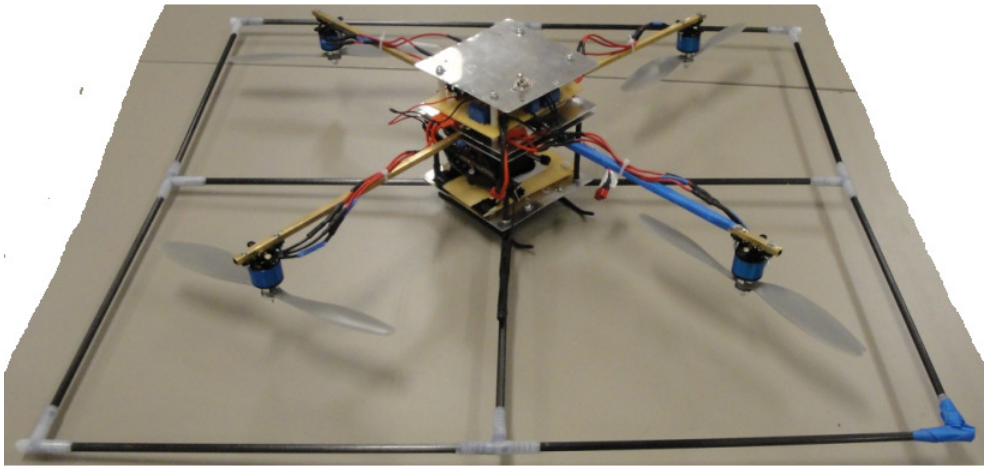


Fig. 2.8: Custom-built frame.

problems which led to large roll or pitch angles often going uncorrected. Another issue was the uncontrollable yaw that led to a constant spinning of the platform. This might have been the reason for quadrotor remaining airborne for a reasonable period of time.

The two main drawbacks of a hand-made frame were:

1. Inaccuracy in the construction of the frame has a huge impact on stability;
2. A poorly constructed frame lead to excessive noise coupling into the MEMS-based sensors, making it hard to filter out the noise.

One of the primary lessons learned from this experience was the need for a well-built and precise frame.

Chapter 3

Hardware and Software Architecture

This chapter introduces the dynamic model of the quadrotor and gives a detailed overview of the airframe, mechanical setup, sensors, computation hardware, and the communication system. It also discusses the integration of these components in the final system.

3.1 Quadrotor Dynamic Model

A quadrotor helicopter is a rotorcraft with two pairs of counter-rotating rotors of a fixed-pitch located at the four ends of the aircraft, as shown in Fig. 3.1. The quadrotor is maneuvered by varying the rotational speed of the rotors in order to manipulate the thrust. Pitch and roll angles, defined as the front/back and left/right angles, are controlled using moments generated by a differential thrust between rotors on opposite sides of the vehicle. The yaw rotation is controlled using the difference in reaction torques between the pitch and roll rotor pairs, as each pair is rotating in opposite directions and thus generating a torque opposite to each one's direction of rotation due to air friction. Vertical motion is controlled by adjusting the total thrust of all rotors together, and lateral acceleration is achieved through a pitch and/or roll of the aircraft.

With four actuators and six degrees of freedom (roll, pitch, yaw, x, y, z), the quadrotor is an underactuated system. Quadrotor helicopters, like traditional helicopters, are dynamically unstable. Unlike some fixed wing aircraft, left without active control the quadrotor will diverge into instability.

The model has been derived using Newtonian mechanics under the following assumptions:

1. The effects of the body moments on the translational dynamics are neglected,

2. The center of mass and the body fixed frame origin coincide,

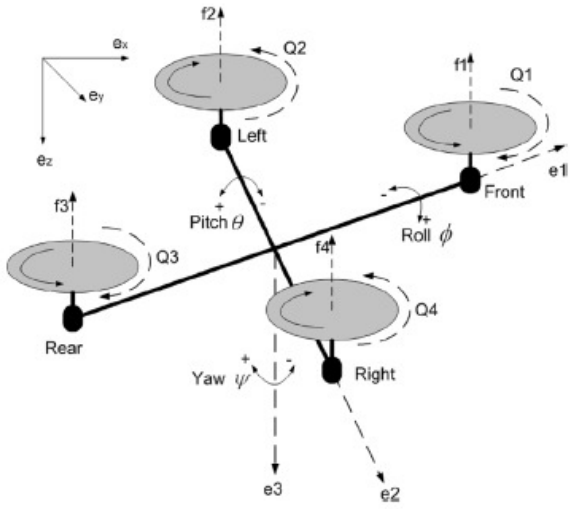


Fig. 3.1: Model of a quadrotor.

3. The ground effect is neglected,

4. Blade flapping is unmodeled,

5. Friction is only considered in the yaw motion,

6. The frame structure is rigid,

7. The helicopter structure is symmetric (diagonal inertia matrix - no axis cross-coupling),

8. Thrust and drag are proportional to the square of the propellers speed.

Aerodynamic effects, such as blade flapping, rotor body dynamics, rotor flapping due to yaw, and variable inflow velocities as a result of craft pitch and roll, are ignored in the model presented here but have been modeled in other quadrotor systems [14–17].

The dynamic model, with the assumptions above and ignoring aerodynamic effects, is essentially a rigid-body model with just abstract force and torque actuators and no aerodynamics. The model here has adjustments to the well-known rigid-body model of the inclusion of an additional gyroscopic term caused by the rotation of the airframe due to the counter-rotating rotors, as well as four additional equations describing the dynamics of

the four rotors [18]. Let $\mathcal{I} = e_x, e_y, e_z$ denote the inertial frame, and $\mathcal{B} = e_1, e_2, e_3$ is the aircraft body frame as shown in Fig. 3.1. Then the dynamic model is given by

$$\dot{\xi} = \upsilon, \tag{3.1}$$

$$\dot{\upsilon} = ge_z - \frac{1}{m}TRe_z, \tag{3.2}$$

where the vector $\xi = [x\ y\ z]^T$ represents the position of the origin of the body-fixed frame, $\mathcal{B}$, with respect to the inertial frame, $\mathcal{I}$, the vector $\upsilon = [\upsilon_x\ \upsilon_y\ \upsilon_z]^T$ represents the linear velocity of the origin of $\mathcal{B}$, expressed in the inertial frame, and $e_z = [0\ 0\ 1]^T$ is the unit vector in the inertial frame, $\mathcal{I}$; g is the acceleration from gravity ($9.81m/s^2$) and m is the mass of the vehicle. The orientation of the vehicle frame is given by the orthogonal rotation DCM matrix, $R \in SO(3)$, and depends on the three Euler angles, ϕ, θ, and ψ of roll, pitch, and yaw. T is the thrust generated by the four rotors in free air and given by

$$T = b\sum_{i=1}^{4}\omega_i^2, \tag{3.3}$$

$$Q_i = k(\omega_i^2), \tag{3.4}$$

where Q_i is the reaction torque generated in free air by the rotor due to drag. k and b are two proportionality constant parameters that depend on aerodynamic effects, including the density of the air, and the size, shape, and pitch angle of the rotor blades. k is on the order of 1.1×10^{-6} and b is around 2.9×10^{-5}.

The next equations give the rest of the model

$$\dot{R} = R \cdot sk(\Omega), \tag{3.5}$$

$$I_f\dot{\Omega} = -\Omega \times I_f\Omega - G_a + \tau_a, \tag{3.6}$$

$$I_r\dot{\omega}_i = \tau_i - Q_i,\ i \in \{1, 2, 3, 4\}, \tag{3.7}$$

where Ω is the rotational velocity of the vehicle in the body frame, $\mathcal{B}$. $sk(X)$ denotes the creation of a skew-symmetric matrix generated using the vector inside the parenthesis, such

that $sk(X)Y = X \times Y$ for any vector $X \in \mathbb{R}^3$ with $\times$ denoting the vector cross product. I_f is the inertia matrix of the airframe with respect to the body frame, $\mathcal{B}$, measured in $kg \cdot m^2$, where the center of mass coincides with the origin of the frame. I_r signifies the moment of inertia of the rotor blades, and is roughly $I_r = 3.4 \times 10^{-5}\ kg \cdot m^2$, while ω_i is the speed of the rotors $1, 2, 3, 4$. G_a is the gyroscopic torque due to the combination of the rotation of the airframe and the four rotors and is given by

$$G_a = \sum_{i=1}^{4} I_r(\Omega \times e_z)(-1)^{i+1}\omega_i. \tag{3.8}$$

τ_a is the airframe torque generated by the rotor given by

$$\begin{aligned} \tau_a &= (\tau_i^1, \tau_i^2, \tau_i^3)^T \\ \tau_a^1 &= d \cdot b(\omega_2^2 - \omega_4^2) \\ \tau_a^2 &= d \cdot b(\omega_1^2 - \omega_3^2) \\ \tau_a^3 &= k(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \end{aligned} \tag{3.9}$$

where d is the distance from the rotors to the center of the aircraft. τ_i is contrasted from τ_a, and is the four control inputs to the system in the form of a motor torques.

3.2 Mechanical Design of the Platform

The aim of this work is to build a robust, reliable, and low-cost experimental platform that is easily extensible and capable of carrying various payloads. After considering resources available to manufacture a frame, it was decided to purchase an off-the-shelf frame. The Mikrokopter MK50 frame best met the requirements in terms of size, weight, and cost [8]. However, owing to a lead time of three months which would have delayed the entire project, a decision was made to build a quadrotor frame. This is shown in Fig. 3.2. While some respectable performance was achieved, the platform was always spinning at a rate of 1-2 Hz owing to the imprecision in manufacturing the frame leading to a net reactive force.

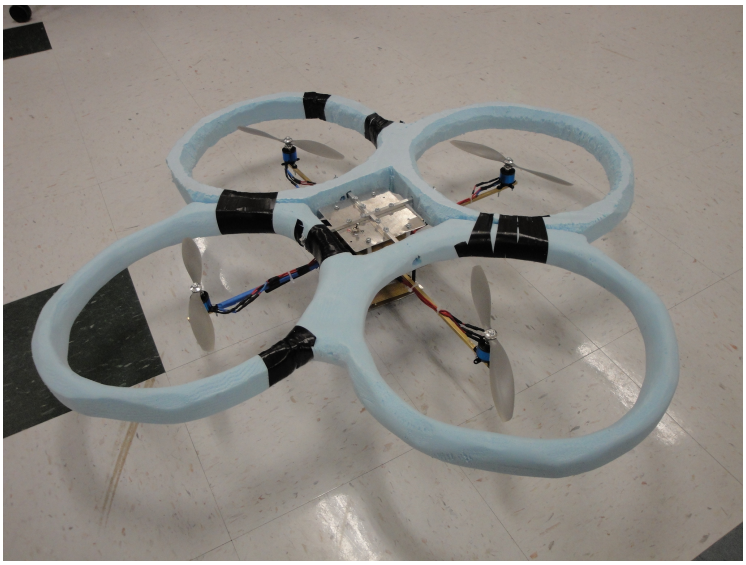


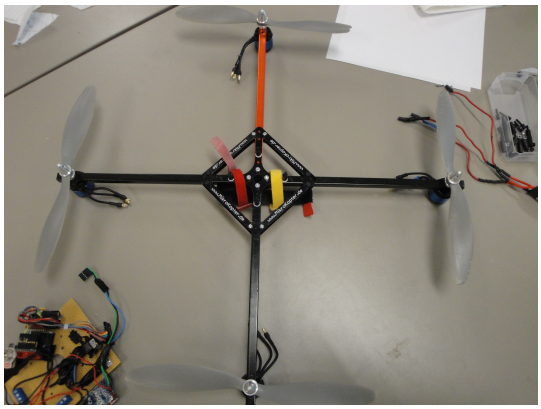
Fig. 3.2: Custom handmade platform with a protective shroud.

3.2.1 Frame

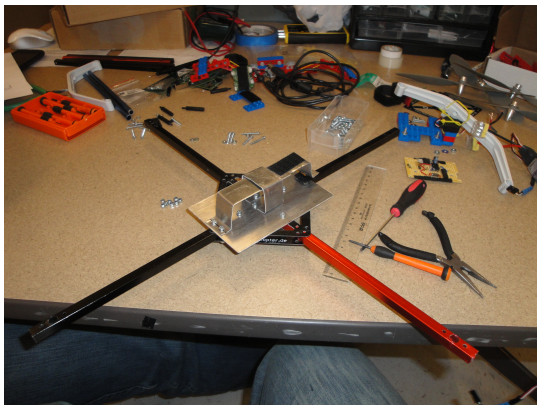
The key lesson learned from the handmade frame was the need for a precisely manufactured frame. And finally, the Mikrokopter MK50 frame shown in Fig. 3.3(a) was purchased. The dimensions of the Frame are 50 x 50 x 1.2 cm and weighs 120 g. A metal plate was bolted to the frame to prevent the roll rods from sagging. Additionally, a small battery holder was made using an aluminum sheet to provide a safe enclosure for the battery and to enable convenient removal and placement of the battery. The battery holder and the reinforcement plate can be seen in Fig. 3.3(b).

3.2.2 Landing Gear

The requirements laid down for a good landing gear were for it to be well-balanced (to keep the quadrotor flat during calibration), rigid, robust, and tall. A tall landing gear was preferred so as to compensate for the six inch deadzone of the sonar and to enable the placement of a camera, battery, and a gripper. Initially, the landing gear from Mikrokopter was used but this was found to be flimsy and very flexible leading to a bouncy landing



(a) MK50 frame with motors and propellors mounted.



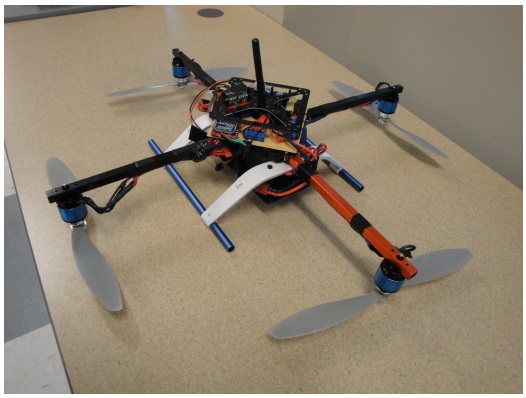
(b) Battery holder and reinforcement plate.

Fig. 3.3: MK50 frame.

causing damage to the quadrotor [8]. Currently a landing gear from a T-Rex 600 Radio-Control (RC) Helicopter is used [19]. It is only 7 cm tall but satisfies all other requirements. There was no reasonably priced (<$100) landing gear available off-the-shelf that satisfied all the requirements.

3.2.3 Center of Gravity

A low center of gravity (CG) is desired in the case of a quadrotor. Although increasing the CG distance from the rotor plane does not affect the natural frequency, it does increase the damping leading to a less than desirable performance in flight. Bristeau et al. discuss the placement of the CG and state that a CG below the rotor plane benefits forward flight stability while a CG above it helps in wind gust rejection [14]. Pounds et al. analyzed the effect of the distance of the CG from the rotor plane on the value of the Bode integral [20]. They showed that the Bode integral is minimized when the CG is coincident with the rotor plane but rises sharply as the CG is moved away from the rotor plane. Placing it a small distance from the center helps minimizing the effect of small errors on the stability. It was these insights that led to the placement of the CG at a distance of 2.3 cm below the rotor plane. The CG was primarily adjusted by changing the position of the motors and is found to be (1.28, -3.97, -22.53) mm. Some of the tested configurations can be seen in Fig. 3.4.



(a)

(b)

Fig. 3.4: Quadrotor configurations.

3.2.4 The Inertia Matrix

In controlling a physical system, it is always desirable to have at least a crude model of the plant dynamics and even more so in the case where the system is highly unstable and nonlinear such as in the case of a quadrotor. However, It is difficult to perform a system identification on an unstable system. Knowing the mathematical model of the quadrotor one can calculate the inertia matrix and be able to get a model that describes the rigid body dynamics. This does not account for any aerodynamic effects. Masses and distances of components right down to a screw were measured and used to calculate an inertia matrix. In calculating the inertia matrix, components are considered as point masses and some of the larger components were split into subcomponents. Another reason to split some components was to avoid the problem of symmetry interfering with the Inertia matrix.

$$I = \begin{bmatrix} 0.0116 & -0.0000 & -0.0000 \\ -0.0000 & 0.0121 & -0.0002 \\ -0.0000 & -0.0002 & 0.0219 \end{bmatrix} \text{Kg-m}^2 \tag{3.10}$$

3.3 Propulsion

This section presents the actuation sub-system of the quadrotor. It discusses the criteria for selecting the components used, and also elaborates on the effect of each of these components to the performance of the quadrotor.

3.3.1 Motors

A table detailing motor characteristics was found on the RCGroups forums [21]. A decision to use the motor with the least vibrations was taken. Some of the key decision criteria were thrust to weight ratio, a low power to thrust ratio and low vibrations. Ultimately, the KDA 20-22L motor was chosen and this reasonably satisfied all the criteria. The motor weighs 56.5 g and delivers a maximum thrust of around 900g with the below-mentioned propellor.

3.3.2 Propellors

The key design aspect to be kept in mind before choosing a propellor is that the thrust produced by a propellor is proportional to the fourth power of its radius and square of its angular velocity. Consequently the input power reduces with a larger propellor. While experiments demonstrated that a 11-inch propellor with a 4.7 pitch angle would be the best match to the chosen motor, the size of our frame prevented the use of this size and instead it was decided to use 10 in propellors at the cost of a 20% reduction in the amount of thrust generated at hover condition. As discussed in section 3.1, the quadrotor utilizes a set of counter rotating propellors in order to cancel the moments generated by one set of propellors by the other. This restricts the choice in propellors to the ones that are available in a pusher-tractor configuration. Ultimately, the decision was to use the 1047 propellors from APC in a pusher-tractor configuration [22]. Table 3.1 gives the thrust generated by the propellors at various speeds and the thrust-RPM curve is found to be fairly linear. The thrust generated by the propellors is less than expected due to the propellors being tested at an altitude of 4500 feet. The operational range of propellors is between 2000 and 9000 RPM.

A subtle point overlooked initially is the mounting of the propellors to the motors. Experiments were conducted with four different configurations - adaptor supplied with the motor, prop saver, screw-based prop adaptor, and the collet-style prop adaptor. The first method required the reversal and shortening of the motor shafts. This is a time-consuming procedure, and moreover there were concerns over play between the adaptor

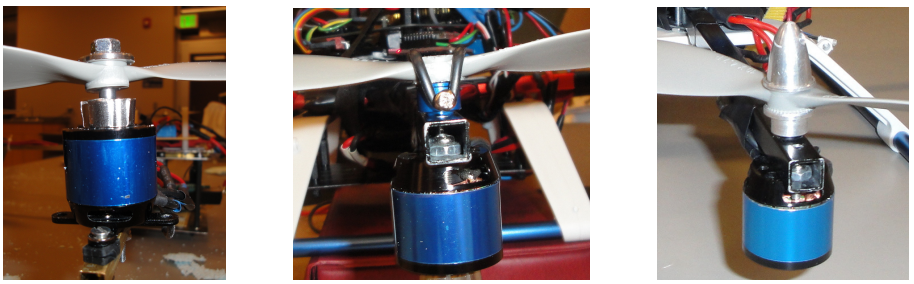
Table 3.1: Thrust generated by propellors at different RPM.

RPM	Thrust (in g)
1980	44
2700	88
3300	137
4080	220
4830	320
5180	374
5550	436

and the propellor. The second arrangement was found to have the worst performance as propellors would detach themselves from the adaptor and fly off during testing. No flight tests were conducted with this configuration. The collet-style and the screw-based adaptor are very similar, however, the former was found to have a more secure fit to the motor and also provided a convenient way to replace propellors. It allows the placement of the motors underneath the rods in order to adjust our center of gravity. This is elaborated in section 3.2.3 and has an impact on the attitude stabilization performance of a quadrotor. Some of the mounting techniques are shown in Fig. 3.5.

3.3.3 Motor Response and System Identification

Conventional helicopters can change the thrust produced by the propellor by varying the pitch angle of the propellor and this is a very effective way of varying the thrust. However, quadrotors owing to mechanical simplicity use fixed pitch propellors, and therefore the only way to vary the thrust is by changing the speed of the propellors. This immediately



(a) (b) (c)

Fig. 3.5: Propellor mounts.

brings up the question of the motor response time. It is ideal to use the a motor/propellor combination with the least inertia in order to reduce to motor response time and maximize actuator bandwidth. This is essential given the fact that this is the slowest part of the system and the bottleneck in the system. Analogous to having a blazing fast processor but a very slow hard disk ensuring that the processor is starved for data. As with any physical system the motor response time is fairly large vis a vis the control frequency.

Typical values range from 50 ms to 100 ms depending on the size of the motor and the propellor used. A simple setup using a photodiode and an infrared (IR) Light-emitting diode (LED) was employed to get an estimate of the motor response to a step input. Data was logged using a National Instruments Data Acquisition Card and analyzed in Labview [23]. The motor constant was found to be 80 ms. This can be attributed to the fast update rate of the ESCs despite the rather heavy motors employed. This is discussed in the following section.

3.3.4 Electronic Speed Controllers

Motor control is typically modeled as a single pole system and most motor controllers employ proportional feedback shown in (3.11) in order to achieved the desired speed. Most modern ESCs sense the back emf in order to determine the speed of the motor.

$$V_{motor} = K(\omega_{des} - \omega), \tag{3.11}$$

where ω is the angular speed, ω_{des} is the desire angular speed, K is the motor constant, and V_{motor} is the voltage to the motor. There are two considerations in selecting an Electronic Speed Controller (ESC): firstly, it should meet the power requirement, and secondly, it should have the highest update rate possible. While it is easy to get an ESC that meets the first requirement, the issue of update rate is a little more subtle. While some hobby-grade ESCs accept up to 400 Hz input update rate, most operate at a 50 Hz update rate. Moreover, it was found that all the hobby-grade ESCs low-pass filter their output in order to give a smooth throttle response and conserve power. This is ideal in the case of a fixed-

wing aircraft, however, requirements for attitude stabilization of a quadrotor dictate fast thrust dynamics requiring a high update rate.

This problem has already been addressed by some hobbyists. It was found that the firmware code has been rewritten to provide high update rates and remove the LP Filter. The gumstix was used to reflash the Atmega8 microcontoller inside the ESCs with this new code. This is shown in Fig. 3.6. A few parameters in the code needed to be tuned and a 400 Khz update rate was achieved. An immediate difference was seen in the performance in the quadrotor and also while testing the motor response vigorously shaking the IMU.

A decision to use hobby grade ESCs was made and the Turnigy Pro Plush 30 A ESCs which could accept 400 Khz input are used. Even though 18 A ESCs would be sufficient, ESCs that could handle higher current were chosen to enable easy upgrades to the system. While systems with updates rates as high as 1 Khz have been designed, 400-500 Hz is an adequate control update rate given the limitations on the sensor bandwidth and the very low actuator bandwidth and this fact is confirmed by the flight performance [24]. However, a key limitation is the very limited resolution of 8 bits at the output of the ESC. This prevents the change of speed of the motors in fine increments leading to less efficient control. The resolution at the input is 1 μs which corresponds to a change in the speed of the motor by 10 RPM which corresponds to a thrust of about 2.2 g and therefore, is not seen as an issue.

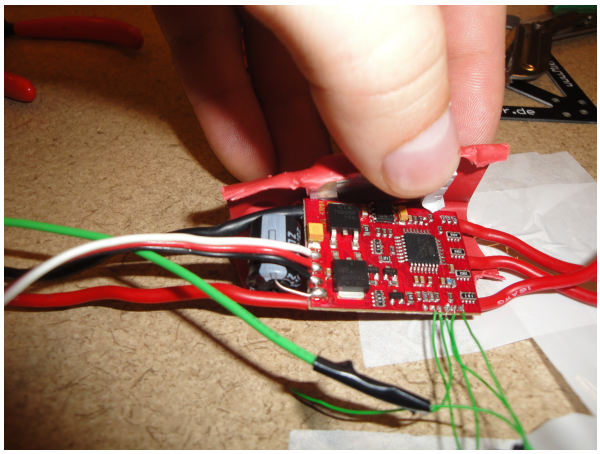


Fig. 3.6: ESC programming setup.

3.4 Computation Hardware

Figure 3.7 shows the the computation hardware used, and the communication paths between them.

3.4.1 Gumstix

The Gumstix Verdex Pro is a small Linux based Single Board Computer (SBC) measuring 80mm x 20mm x 6.3mm and weighting a mere 8g. It runs on a Marvell PXA270 processor based on the ARM XScale architecture with a clock speed 600 MHz and is equipped with 64 MB DDR RAM and 32 MB NAND flash memory. It runs an embedded Linux Distribution Ångström. The Gumstix has expansion connectors on each sides for attaching to expansion boards. It is used in conjunction with the netpro-VX expansion board which enables both wired and wireless network access (via a wi-fi module).

The Gumstix was chosen due to it being already available in the lab. It is the main processor used for all tasks on the quadrotor such as attitude stabilization and navigation. However, one of the key drawbacks is the lack of a floating point unit (FPU), requiring the use of the "soft-fpu" option, when compiling.

3.4.2 Robostix

The Robostix board consists of an Atmel ATMega128 microcontroller with the I/O pins exposed and it communicates with the Gumstix over I^2C and is sold as an expansion board to the Gumstix [25,26]. Two Robostixs are used in the current configuration with the primary being used to generate the PWM signals being fed to the ESCs and to continuously sample the sonar. An interrupt driven structure is employed for the robostix program wherein it receives commands to modify the PWM signals or return the sonar value over the I^2C line as an interrupt. The packet structure used, to modify PWM values or return ADC values, ensures minimal latency. A second Robostix is needed to actuate the gripper due to insufficient (exposed) 16-bit PWM pins on the primary Robostix.

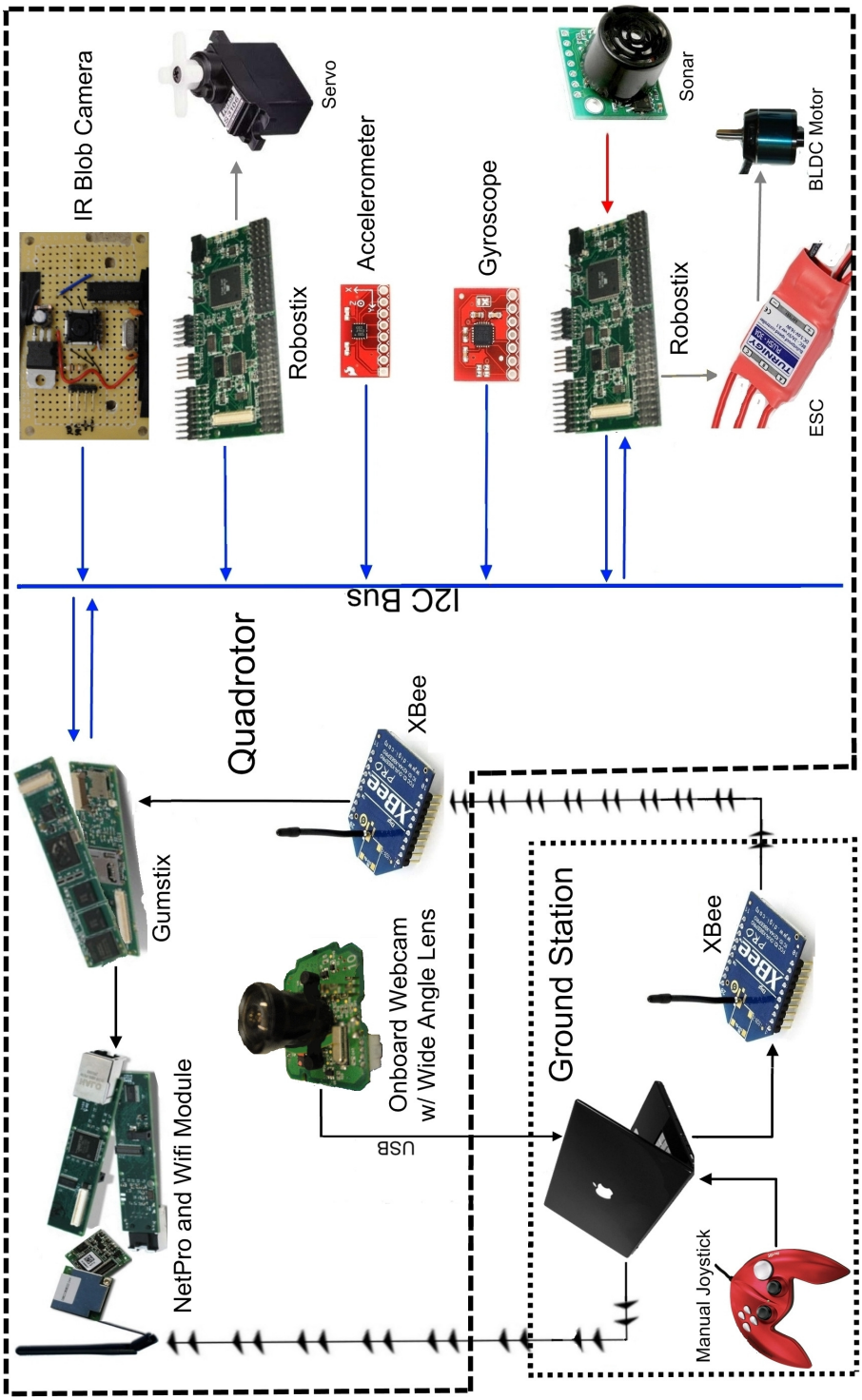


Fig. 3.7: Communication diagram of quadrotor system.

3.5 Sensing

Figure 3.7 shows the the sensors used, and the communication paths between the sensors and the computation hardware.

3.5.1 Inertial Measurement Unit

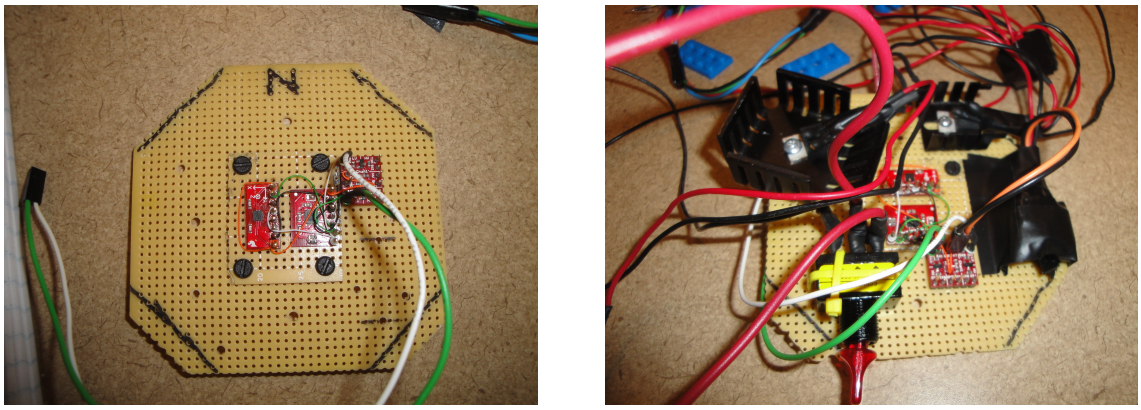
A low-cost and light-weight IMU is developed using an accelerometer and a gyroscope mounted on a small board. The IMU measures less than 2.5 x 2.5 cm and weighs less than 6g. This forms one of the most critical components of the entire system. The IMU is shown in Fig. 3.8(a). It is shown along with the power distribution board in Fig. 3.8(b).

Gyroscope

A MEMS-based gyroscope from Invensense ITG-3200 is used to obtain angular rates [27]. It has a full scale range of ± 2000 deg/s at a resolution of 14.375 LSBs/s. Three integrated 16-bit ADCs provide simultaneous sampling of gyros and these are fed to digital FIR filters. The filtered data is sent over I^2C.

Accelerometer

A Bosch BMA-180 3-axis accelerometer that reports accelerations up to 16g in the inertial frame is used [28]. The analog signals are sampled using a 14-bit ADC and are then fed to user-selectable digital FIR filters. A subtle issue in the use of accelerometers concerns its placement. The accelerometers need to be placed in the center as this prevents the sensor from reporting centripetal acceleration (even in the absence of linear acceleration) when the quadrotor is yawing. Empirically, it is observed that accelerations beyond ± 2g are never reported in stable flight conditions. Therefore, the accelerometer is set to have a resolution of 0.25 mg giving a full scale range of ± 2g. Moreover, the accelerometer is configured to work in the ultra low-noise mode.



(a) (b)

Fig. 3.8: Sensors and the power distribution board.

3.5.2 Sonar

There is a dearth of low-cost altitude sensors that can provide high resolution measurements at a sufficient bandwidth. After much deliberation the Maxbotix LV EZ-2 sonar was selected [29]. It has resolution of 1 in (2.54 cm) and can report distance at a rate of 20 Hz. It has a medium cone angle which enables it to be invariant to the attitude of the quadrotor to a certain degree and at the same time being able to detect relatively small obstacles.

3.5.3 Camera

A single Logitech Quickcam Pro 5000 webcam is used as the primary sensor for navigation [30]. It was chosen for its ability to capture video at a resolution of 640 x 480 at 30 frames per second over a Universal Serial Bus (USB) connection. Moreover, it uses a standard M12 lens mount that allows it to be equipped with a different lens. A 2.1 mm wide-angle lens was employed to maximize the area seen by the camera.

3.5.4 IR Camera

Nintendo Wii remote controler or the Wiimotes are equipped with a special camera capable of tracking four infrared light sources at a rate of up to 200 Hz and reporting their x, y position and size as projected on to the camera image over fast mode I^2C [31]. This

camera was removed from the Nintendo Wiimote and fitted onto a custom made board that supplies it with a clock, power source, and I^2C lines to communicate with the Gumstix. It enable the quadrotor to track moving sources. This is further discussed in Chapter 6.

3.6 Communication

3.6.1 I^2C

The I^2C communication protocol forms the backbone for the entire system as shown in Fig. 3.7. It connects the main onboard processor Gumstix to the all the sensors and microcontrollers at a rate of 400 kHz. One challenge faced while using this protocol was the need for bi-directional level shifters to enable conversion from 3.3 V to 5V and vice-versa to interface with devices operating at 3.3 V with those operating at 5 V. These allow either end of the voltage line (3.3V or 5V) to pull the line low for acknowledging. This is achieved by using a separate N-channel enhancement MOS-FET on both the serial data (SDA) and serial clock (SCL) lines with their gates tied to the 3.3 V source. A board with the necessary MOS-FETs integrated from Sparkfun is used.

3.6.2 Zigbee

There is a need for a simple wireless transceiver with minimal latency that can transmit data from a ground station to the quadrotor so as to be able to maximize control loop bandwidth. The Zigbee standard was chosen to be used as it was found to have minimal latency. An XBee module from Digi Inc. is used [32]. Data can be transmitted/received over a serial line making it very convenient to interface with the gumstix.

3.6.3 Wi-Fi

This is primarily used to remote login into the gumstix in order to initiate the program in the quadrotor as well as change configuration parameters. It also enables the use of the host computer to be used for emergency functions such as landing.

3.7 Ground Station

3.7.1 Manual Control

A manual controller is provided for a human operator to take over in the event of a malfunction or system failure. An off-the-shelf game controller with dual joysticks is interfaced to the ground station and manual commands are sent over Zigbee. A human operator is allowed to take over at any point of time. It has especially been found useful in testing new control algorithms or in tuning a controller.

3.7.2 Host Computer Setup

The host computer serves three purposes: cross-compilation of code, run heavy duty computation such as the Simultaneous Localization and mapping (SLAM) algorithm, and finally as an interface to the manual controller.

The Gumstix is based on an ARM processor which necessitates the cross-compilation of code using a host computer. A utility known as BitBake is used to cross-compile code and forma package to be deployed to the target (Gumstix). It uses BitBake recipes that tell it how to build a particular package. It includes all the package dependencies, configuration, compilation, build, install, and remove instructions.

3.8 Power Routing and Consumption

3.8.1 Battery

Lithium Polymer batteries have a high energy density and a high discharge rate. Three factors that govern the choice of batteries are voltage, capacity, and the discharge rate. Tests were conducted to measure the current draw of the motor at hover speed and was found to be 3A. A 3-cell 11.1 V 2600 mAh Lithium Polymer Battery with a discharge rate of 20C (52 A) manufactured by ThunderpowerRC is used [33]. The battery measures 102 x 31 x 25 and weighs 185 g and is placed at the center of the frame.

3.8.2 Power Routing

The battery is used to supply power to the motors as well as all the onboard electronics. Power is directly fed to the ESCs after which it is down converted to 5V using two LDO Regulators and further down to 3.3 V for supplying to accelerometers, gyroscopes, and the IR camera. Converting the voltage down to 5V for the electronics leads to a huge power dissipation (11.1 - 5) x 0.62A = 3.78W. This necessitates the use of large heat sinks in order to avoid resetting of the gumstix. The reason for using linear regulators is the unavailability of buck converters in easily usable packages such as TO-220, although high frequency noise might need to be taken care of.

3.8.3 Power Consumption and Running Time

The quadrotor achieves a flight time of 8 minutes using a 2600 mAh battery. About 90% of the power is consumed by the motors. The flight time can be easily increased by using a higher capacity battery.

3.9 Summary

This chapter presented the entire hardware and software architecture of the quadrotor system. It discussed the criteria for selection, and integration of all the components of the quadrotor. A picture showing the assembled quadrotor, hovering in an unstructured environment is shown in Fig. 3.9.

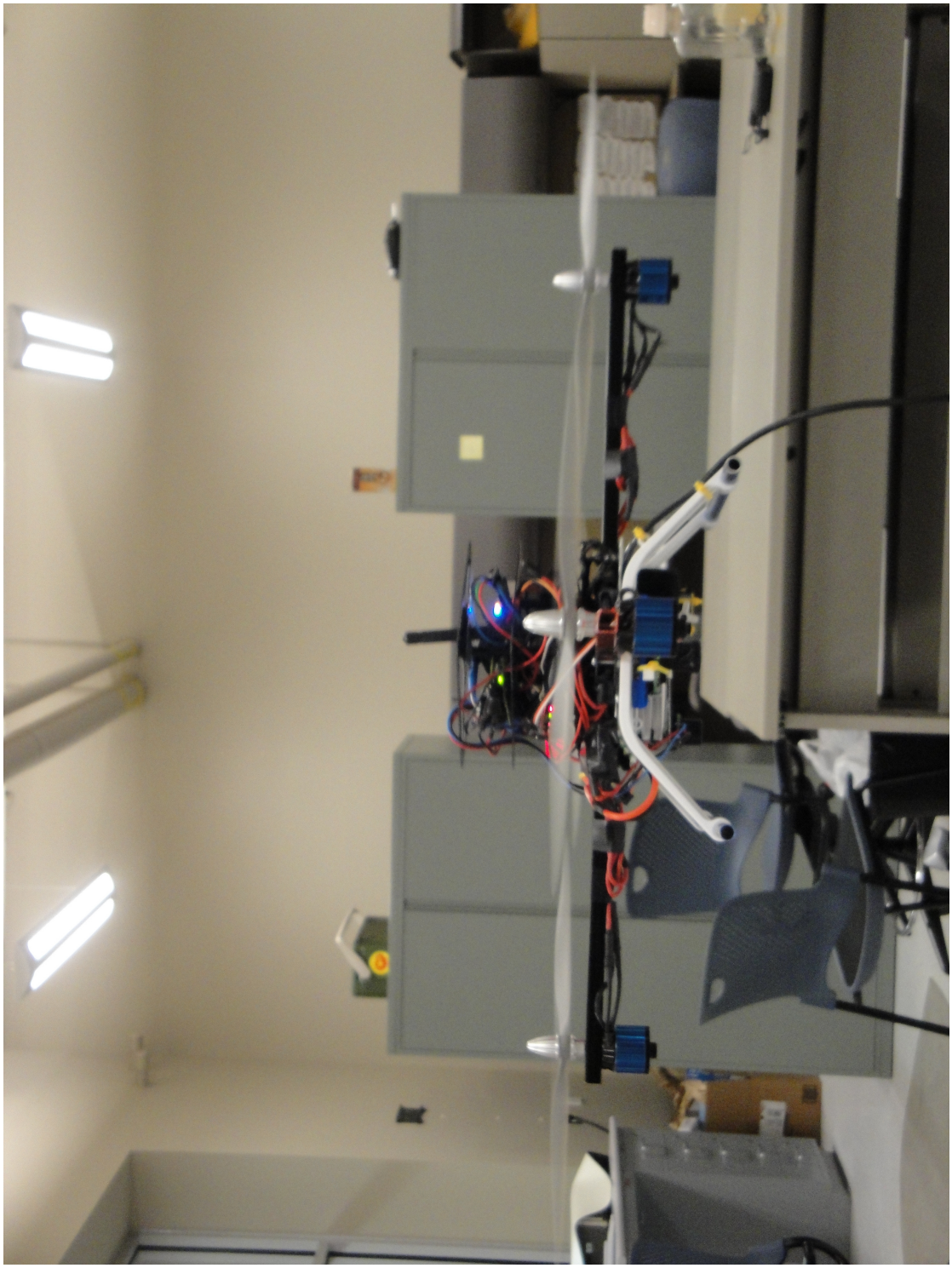


Fig. 3.9: Quadrotor hovering.

Chapter 4

Attitude Estimation and Altitude Control

A fundamental problem in MAVs is the estimation of attitude and its control. The attitude control loop is responsible for controlling a highly unstable and nonlinear system. This is a high-bandwidth loop that forms the inner-most loop in the control system for the quadrotor, and is commanded by navigation and guidance loop. With the proliferation of MEMS technology, gyroscopes and accelerometers are available at increasingly lower cost and small form factor and have found their way into many MAV systems in the last decade. The aim of this work is use ultra low-cost IMU to stabilize the quadrotor. While there is no standard definition of‘ “low-cost,” the IMU developed is more than an order of magnitude cheaper than similar systems in the literature.

4.1 Related Work

The attitude control problem has been addressed by several researchers and a wide variety of solutions have been proposed. Wen and Kreutz-Delgado have a brief survey in their paper and the interested reader is encouraged to look this up [34]. The dynamic model of a quadrotor has an additional gyroscopic term caused by the rotors and in the absence of this term quadrotor control reduces to the well known rigid body problem in control. Some of the earliest work in the field concerned the attitude stabilization problem.

4.1.1 Attitude Control Using Onboard Sensors

One of the first quadrotor projects was ETH Zurich's OS4. A number of control schemes such as backstepping, LQR, and Sliding-Mode control were implemented and evaluated both experimentally on a test bench and in simulation [17, 35–38]. The performance of sliding mode was found to be less than satisfactory due to its switching nature leading to high-

frequency, low-amplitude vibrations causing the sensors to drift, while LQR was found to be inadequate due to unmodeled effects. Tayebi and McGilvray present model-based PD^2 and model-independent PD controllers based on quaternions and prove the former to provide global exponential stabilization [18]. They also present experimental results using a tethered quadrotor attached to a stationary ball joint base. Hoffmann et al. present a LQR based attitude stabilization and Guenard et al. present a nonlinear control law applied to some of the first quadrotors demonstrating quasi-stationary flight [39, 40]. Position control was achieved with the aid of a human operator. Gurdan et al. presented a light-weight quadrotor with a control loop running at 1 khz achieving very good attitude stabilization performance. Position control was achieved using a Vicon motion capture system [11, 24].

4.1.2 Attitude Control Using 3D Tracking System

Castillo et al. present one of the first free flying quadrotors with control of all degrees of freedom, although they use an external power supply and an external electromagnetic-based position/orientation sensor [41]. They applied a nested saturation controller which they showed to perform better than LQR [42]. Valenti et al. presented one of the first systems demonstrating full control of a quadrotor over a relatively larger area with both attitude and position control relying on the Vicon motion capture system [43]. The Vicon motion capture system, depending on the number of cameras used, is able to provide accuracy ranging from sub-mm to 50 μm at a rate of 300 Hz. However, these restrict the flight envelope of the quadrotor and preclude the addressing of real-world sensor issues.

Altitude control, despite its many challenges, has typically been solved using a PID loop. However, other solutions such as Integral Sliding Mode and Reinforcement Learning have been applied [44].

4.2 Pre-processing and Filtering IMU data

As discussed in the previous chapter, the two sensors that sit at the heart of the quadrotor are the gyroscope and the accelerometers. Raw data returned from the sensors is typically noisy and needs to be filtered. Ideally, gyroscopes need to be band-pass filtered

to remove the very low-frequency (< 1 Hz) bias component and the high frequency noise caused by the spinning of the motors. But in practice, it is hard to design digital filters of reasonable length that can cutoff very low frequencies, and therefore the raw data from the gyroscopes is low-pass filtered and the bias is estimated in the attitude extraction filter with the help of accelerometers.

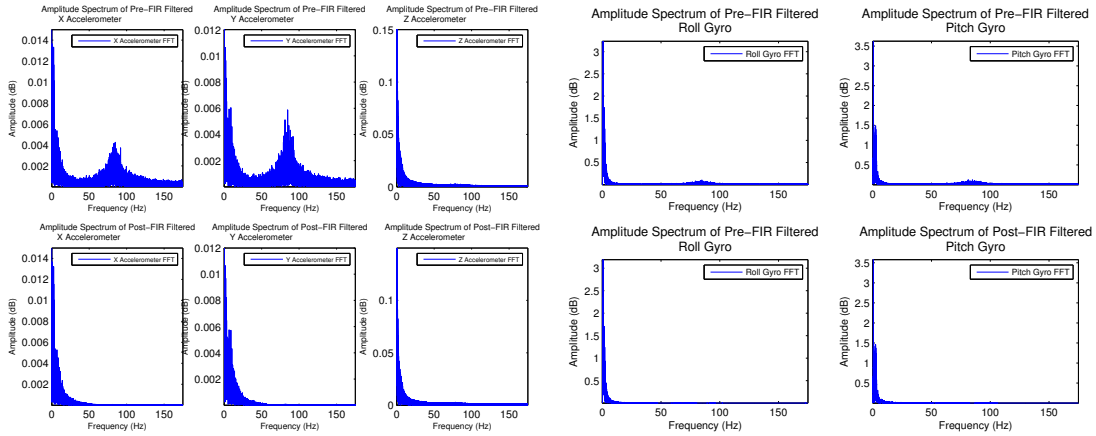
4.2.1 Current Sensors

Even though the gyroscopes and the accelerometers have inbuilt user-selectable Finite Impulse Response filters (FIR), it was found necessary to filter some more after analyzing the plots of the Fast Fourier Transform of the data. As can be seen in Figs. 4.1(a) and 4.1(b), the maximum noise is at around 80 Hz. This was suspected to be noise from the motors and was confirmed after a system identification showed that the motor was indeed spinning at around 80 Hz (4800 RPM).

The next step presents the choice of either using an Finite Impulse Response or an Infinite Impulse Response Filter (IIR). While a FIR filter has the desirable properties of being stable and having a linear phase, it takes relatively more computation for a given frequency response as compared to an IIR filter. However, the coefficients for an odd-order FIR filter are symmetric and this can lead to efficient computation. The FIR and IIR filters seemed to have similar performance and this was confirmed through flight tests.

Initially, window-based design methods were employed to design FIR filters. These typically provided a low stop-band attenuation and a very high transition bandwidth. Shifting to the weighted least squares design technique gave filters with a performance much closer to IIR Filters. Care has to be taken to minimize the order of the filter as it directly relates to the delay in the data.

The inbuilt low-pass FIR filters of the gyroscope are configured to to have a cutoff 20 Hz, however as discussed earlier, additional filtering is required. An 8th-order filter with a cutoff frequency of 16 Hz is used. Similarly, the internal filters in the accelerometer are set to a cutoff frequency of 10 Hz and an additional 8th order filter with a cutoff frequency of 5 Hz is employed. The magnitude and phase response of the filter is shown in Fig.



(a) Accelerometers. (b) Gyroscopes.

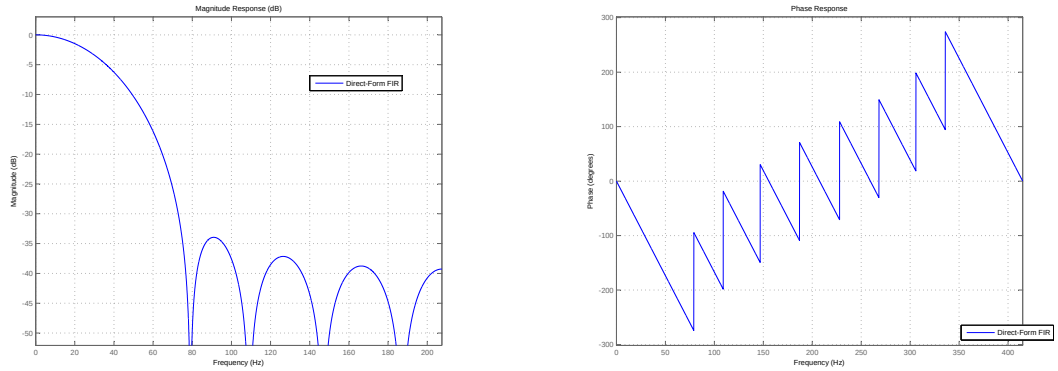
Fig. 4.1: FFT magnitude for unfiltered and filtered accelerometers and gyroscopes data.

4.2. Furthermore, the accelerometer is set to operate in an "ultra-low noise mode" that provides additional accuracy and precision at the cost of using more power. The range of accelerations to be reported can be set using the internal registers on the accelerometers and is chosen to be $\pm 2g$ after examining flight data. The resolution of the gyroscopes and accelerometers is 0.696 deg/s and 0.25 mg, respectively.

Figure 4.3 shows the unfiltered and filtered angular rates and accelerations. The raw values reported by the gyroscopes (after onboard filtering) are ± 8 deg/s and after FIR filtering on the gumstix are ± 1deg/s. The accelerations reported by the accelerometer on the x, y, and z axes are ± 0.3 m/s^2, ± 0.5 m/s^2, and ± 0.2 m/s^2. The roll axis is found to be more sensitive to noise and this could be attributed to the fact that landing gear provides support to the pitch axis while the roll rods lack any such structural support.

4.2.2 Original Sensors

The fist sensor used was a Sparkfun 5-DOF IMU consisting of an Invensense IDG-500 2-axis Gyroscope and an Analog Devices ADXL-335 3-axis accelerometer on one board [45–47]. The sensor employed returns both angular rates and accelerations as analog signals. These are sampled using a 10-bit ADC onboard the Robostix and this data is sent to the Gumstix



(a) Magnitude.

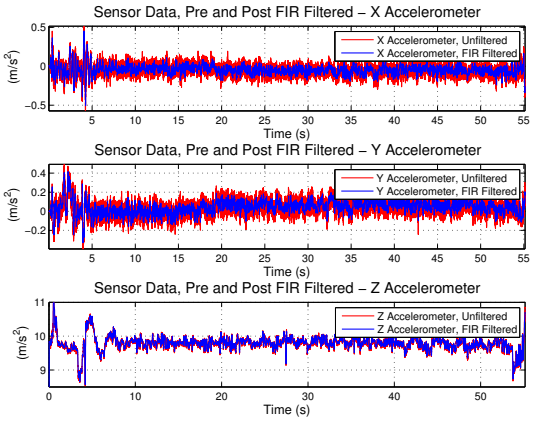
(b) Phase.

Fig. 4.2: Magnitude and phase response for the FIR filter designed to filter accelerometer data.

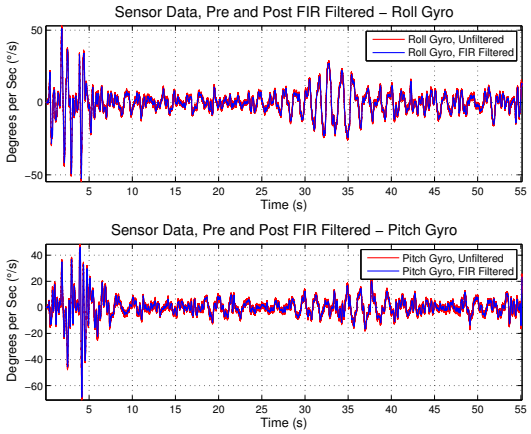
for further processing. This precluded the use of FIR filters of long length due to the lack of a FPU on the Gumstix. The sensor employed seemed particularly susceptible to noise and a range of measures were taken to get "clean" data out of the sensor. The frequency spectrum of the data from the sensors is shown in Figs. 4.4 and 4.5, and the poor quality of the data can be inferred when compared to Figs. 4.1(a) and 4.1(b).

The gyroscope output has low-pass RC filters (750 Ω and 0.1 μF) with a cutoff of 2.2 kHz even though the gyroscope chip has internal low-pass filters with a cutoff of 140 Hz. These can be seen as anti-aliasing filters. One of the first attempts at obtaining better data was the addition of a 2.2 μF capacitor in parallel to give a cutoff frequency of 92 Hz. Although this did lead to relatively better data quality, it was not good enough for sustained flights. Another reason considered for the presence of high frequency noise was electro-magnetic interference. A small metal enclosure was designed to counter this, but no change in the frequency spectrum was noticed.

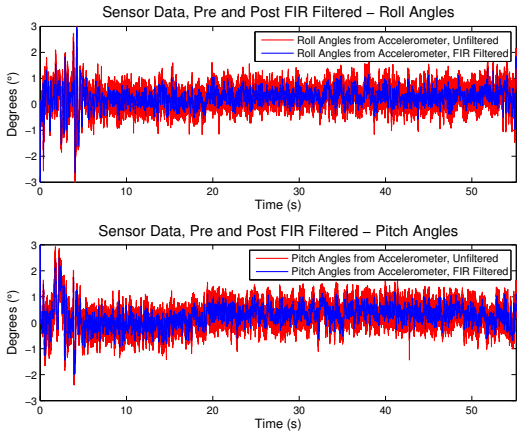
After reflashing the ESCs, it was observed that both the proportional and derivative gains on the attitude controller needed to be reduced. Despite this sustained flights were still not possible. An effort to filter more aggressively using longer FIR filters led to rates of 0-6 deg/s being reported with the quadrotor on the ground. However, the length of the FIR filters used led to a delay and this actually led to a drop in performance.



(a) Filtered vs unfiltered acceleration data.



(b) Filtered vs unfiltered gyroscope data.



(c) Filtered vs unfiltered angles extracted from accelerations.

Fig. 4.3: Filtered vs unfiltered IMU data.

In investigating the source of the noise, various measures to mechanically dampen the noise on the IMU were also taken. The IMU board was mounted using different materials such as foam, double-sided tape, grommets, springs, etc., in a variety of configurations. The various mounting procedures were compared based on the RMS value of the data after running the motors for a fixed period of time. Another measure taken to dampen the noise was to use grommets for mounting the motors. However, this was not found to make a difference and was ultimately discarded.

Tests were also conducted by flicking various components such as the battery, PWM wires, roll/pitch rods, antenna, etc., and observing the effect on the IMU data. A key

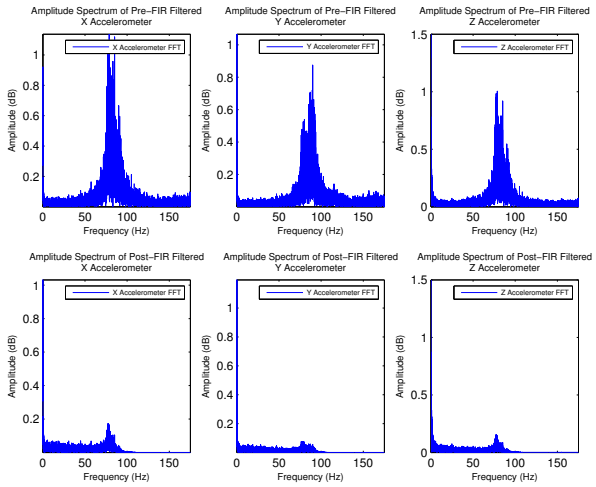


Fig. 4.4: Frequency spectrum of filtered vs unfiltered accelerations reported by the original sensor.

lesson learned was to strap everything down more securely to reduce electrical noise from connectors. This led to marginally better data from the sensors and allowed higher gains on the attitude controller of the quadrotor.

Angles reported by the attitude extraction filter seemed accurate with motors both on and off, with the quadrotor on the ground. This was also verified with a cheap off-the-shelf digital inclinometer and the values reported by the IMU and inclinometer seemed to agree within a degree. However, not much should be read into this as the accuracy of the inclinometer may itself be suspect. However, this was a good sanity test. In the absence of any external motion system, it is difficult to verify the sensor data while in flight. Large rate values were reported in flight even though there was nothing to suggest this behavior. The quadrotor was flown higher than 1 foot off the ground to prevent ground effects from interfering with the flight. The accelerations, angular rates, and angles extracted from the accelerations during a typical flight are shown in Fig. 4.6. It is worth noticing that the raw data was extremely noisy as compared to the filtered data. This is especially true for the accelerometers, where there is an order of magnitude difference between the raw and filtered data.

Another hypothesis for the poor performance with these sensors is the low-sampling resolution, as the gyroscopes have been used on other projects such the the PIXHAWK

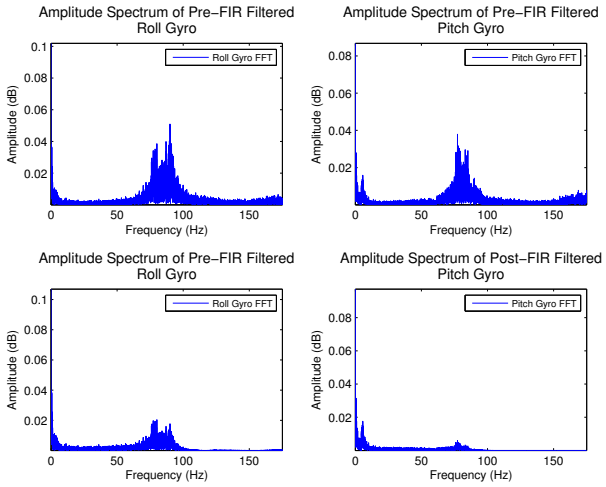


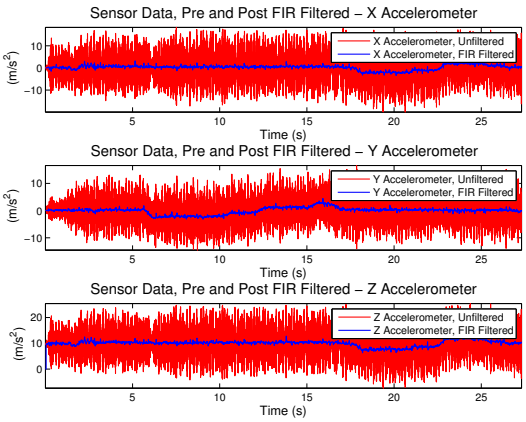
Fig. 4.5: Frequency spectrum of filtered vs unfiltered angular rates reported by the original sensor.

quadrotor from ETH Zurich [48]. Sampling using a 10-bit ADC coupled with the low-resolution sensors led to high quantization and a resolution of 2.5 deg/s for the gyros and 0.16 g for the accelerometers. Converting the accelerometer estimates to angles gives a resolution of 0.95 degrees which is very coarse.

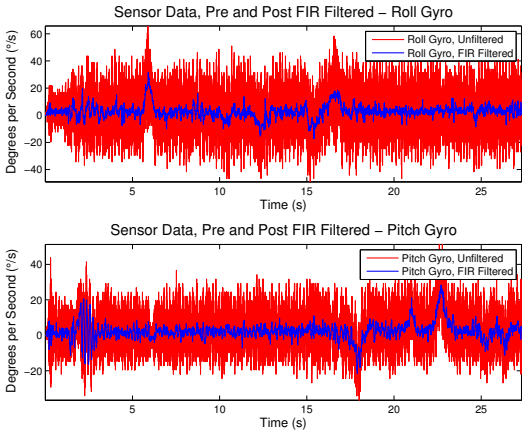
The sensor seemed to be reporting wrong values especially in flight conditions, although there is no way to verify this hypothesis. One proof of this was when the old sensor was replaced with newer accelerometers and gyroscopes, extremely stable flight with the gyroscope reporting rates of ± 2 deg/s in flight were achieved. In conclusion, the sensor might have had a defect which led to all the problems.

4.2.3 Calibration of Sensors

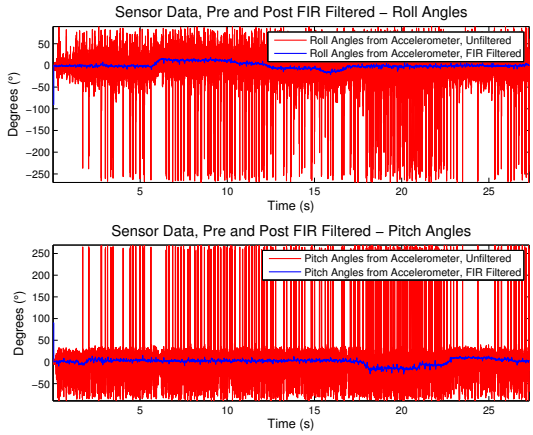
An automatic calibration procedure is initiated at the beginning of each flight to determine the zero values for both the accelerometers and gyroscopes. The raw values are averaged over a short period of time to obtain these zero values. This is implemented on the Gumstix. Tests were conducted to find the appropriate duration over which to average, and it was seen that a period of about 1 s is sufficient.



(a) Accelerations reported by the original sensor.



(b) Angular rates reported by the original sensor.



(c) Angles extracted using accelerations from the original sensor.

Fig. 4.6: Filtered vs unfiltered IMU data for the original sensor.

4.3 Extraction of Angles from Accelerations

Estimated roll and pitch angles are extracted from accelerations using

$$\phi = \arctan\left(\frac{\ddot{z}}{\ddot{y}}\right) + \frac{\pi}{2}, \tag{4.1}$$

$$\theta = -\arctan\left(\frac{\ddot{z}}{\ddot{x}}\right) - \frac{\pi}{2}, \tag{4.2}$$

where ϕ is the roll angle, θ is the pitch angle, $\ddot{x}$ is the acceleration on the x-axis (in m/s^2), $\ddot{y}$ is the acceleration on the y-axis (in m/s^2), and $\ddot{z}$ is the acceleration on the z-axis (in m/s^2) [1].

4.4 Attitude Estimation

Attitude estimation deals with the problem of extracting the attitude by fusing the data obtained from two sensors: accelerometers and gyroscopes with complementary characteristics. This problem is challenging as angular velocities cannot simply be integrated to obtain angles. Moreover, a quadrotor's accelerated frame of reference presents additional challenges as compared to fixed wing MAVs. The aim of this section is to present the attitude estimation algorithm used and discuss the challenges faced in obtaining a reasonable attitude from low-cost sensors.

Traditionally, high quality and expensive gyroscopes such as fiber optic gyroscopes have been used for stability and control of UAVs. While MEMS technology has dropped the cost of sensors, these low-cost systems suffer from other problems. The gyroscopes provide relatively higher quality data compared to the accelerometers, but suffer from drift. Drift is essentially a time-varying bias on the reported angular velocities which, if left uncorrected, could lead to catastrophic failures. While the accelerometer do not suffer from drift, they are susceptible to high-frequency noise. The most useful data returned by the accelerometers is in the lower-frequency range, while gyroscopes have their most useful data in the relatively higher-frequency range. This property is often exploited in the design of

[1] In practice, the atan2 function is used and care needs to be taken to handle angles exceeding π and below $-\pi$.

attitude estimation filters.

4.4.1 Attitude Estimation Filters

On surveying the literature, three major attitude estimation filters were found:

1. Linear Complementary Filter;
2. Kalman Filter (including variants such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), etc.);
3. Nonlinear Complementary Filter.

The Linear Complementary Filter uses a fixed gain to obtain a convex combination of the data from the gyroscope and the accelerometer to obtain an attitude estimate. It was discarded as it is only valid for small angle changes and makes fairly stringent assumptions. Kalman Filter is the most widely used solution to get reliable angle estimates in the presence of noise [49]. The rate data from the gyroscopes is integrated using the Simpson's rule to obtain angles and is used in the prediction step of the KF and the accelerometer data is used in the measurement update step.

4.4.2 Nonlinear Complementary Filter

The nonlinear complementary filter has been developed in the past few years and provides a superior method to fuse the accelerometer and gyro measurements, exploiting the known properties of the underlying system [50]. The specific form of the nonlinear complementary filter that is used for our implementation is designated as the passive nonlinear complementary filter and is shown in Fig. 4.7 [50]. The rotation of the quadrotor can be described using a rotation matrix and all possible orientations of the quadrotor live in the space of 3x3 orthogonal matrices known as the Special Orthogonal Group SO(3). This filter is specifically designed on the SO(3) group to exploit the known rotational kinematics of

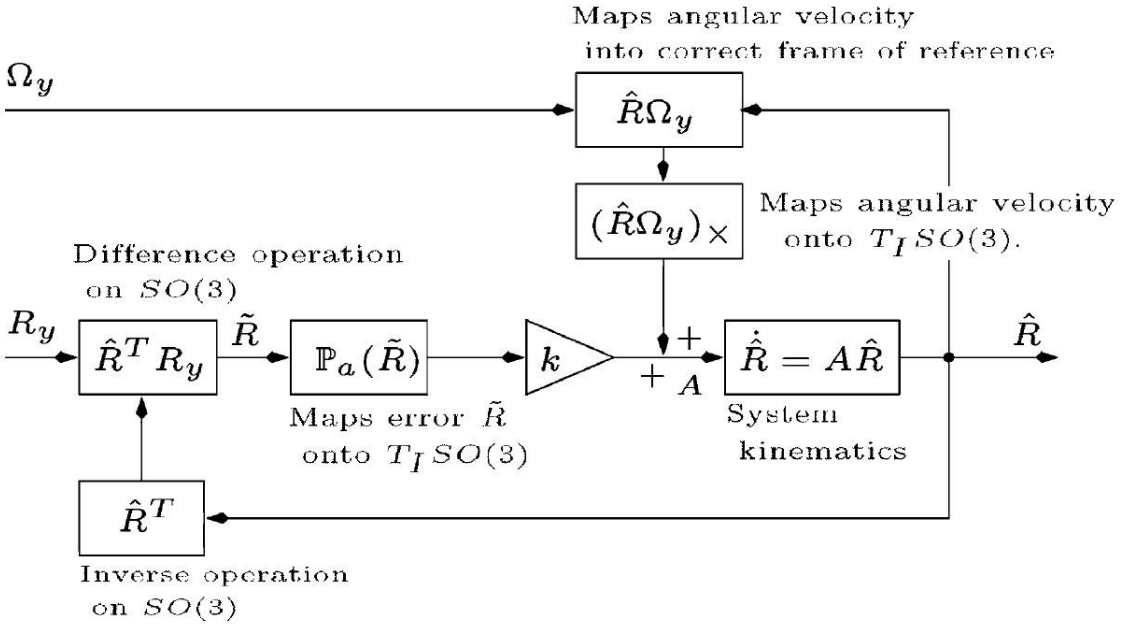


Fig. 4.7: Block diagram of the passive nonlinear complementary filter.

the quadrotor. The proposed observer/filter is posed as a kinematic system in (4.3) to give an estimate, $\hat{R}$, of the attitude of the system, using

$$\dot{\hat{R}} = (\hat{R}\Omega + k_p \hat{R}\omega)_X \hat{R}, \tag{4.3}$$

where $\hat{R}$ is the estimated attitude, Ω is the angular velocity given by the gyros, ω is the correction/innovation term and is a function of the error $\tilde{R}$, given by $\hat{R}^T R_y$, and k_p is a non-zero positive gain. $()_X$ denotes the creation of Skew-symmetric matrix from the generating vector. The pre-multiplication of Ω by the rotation matrix R is to ensure that the velocity is in the correct frame of reference. This is necessary since measured angular velocity lies in the body-fixed frame, while the filter requires the two measurements in the same frame. If the measured angular velocity was obtained directly in the inertial frame, for example, such as would be supplied by an external vision system, then this transformation would be unnecessary.

The aim is to design an observer that drives the error $\tilde{R}$ to I, where $\tilde{R}$ is the error

between the attitude estimated by the filter $\hat{R}$ and the estimated attitude R_y using the accelerometers given by (4.1) and (4.2). The innovation term, ω, can be seen as a nonlinear approximation of the error between R and $\hat{R}$. Through Lyapunov analysis, this is found to be the mapping of the error onto the tangent space of SO(3), which is the space of skew-symmetric matrices. It is given by $vex(\pi_a(\tilde{R}))$, where $\pi_a(\tilde{R}) = \frac{1}{2}(\tilde{R} - \tilde{R}')$ and vex returns the generating vector of a given skew-symmetric matrix.

From the block diagram of the nonlinear filter shown in Fig. 4.7, one can notice the structural similarity with the linear/classical complementary filter, and hence the name. In the block diagram setup of the filter, the $\hat{R}^T$ operation is an inverse operation on SO(3) and is equivalent to a "-" operation for a linear complementary filter. The $\hat{R}^T R$ operation is equivalent to generating the error term $y - \hat{x}$. The two operations $\pi_a(\tilde{R})$ and $(R\Omega)_X$ are maps from the error space and velocity space into the tangent space of SO(3). The two skew-symmetric matrices thus generated are added using a positive gain k_p, giving

$$B = k_p \cdot \pi_a(\tilde{R}) + (\hat{R}\Omega)_X. \tag{4.4}$$

Finally, taking the exponential map brings us back to the Lie Group SO(3). However, given the constraints of embedded processors, it is desirable to find a computationally cheap solution to this matrix exponential. Fortunately, the matrix exponential of skew-symmetric matrices has a closed form solution using the Rodrigue's Equation,

$$A = I_3 + B\frac{sin(|vex(B)|\,dt)}{|vex(B)|\,dt} + B^2\frac{1 - cos(|vex(B)|\,dt)}{|vex(B)|\,dt}. \tag{4.5}$$

The final simplified form of the filter along with the bias estimator is shown in Fig. 4.8. It is given by

$$\dot{\hat{R}} = \hat{R} \cdot (\Omega_y - \hat{b} + \omega)_X, \tag{4.6}$$

$$\dot{\hat{b}} = -k_b \cdot vex(\pi_a(\tilde{R})), \qquad k_b > 0, \tag{4.7}$$

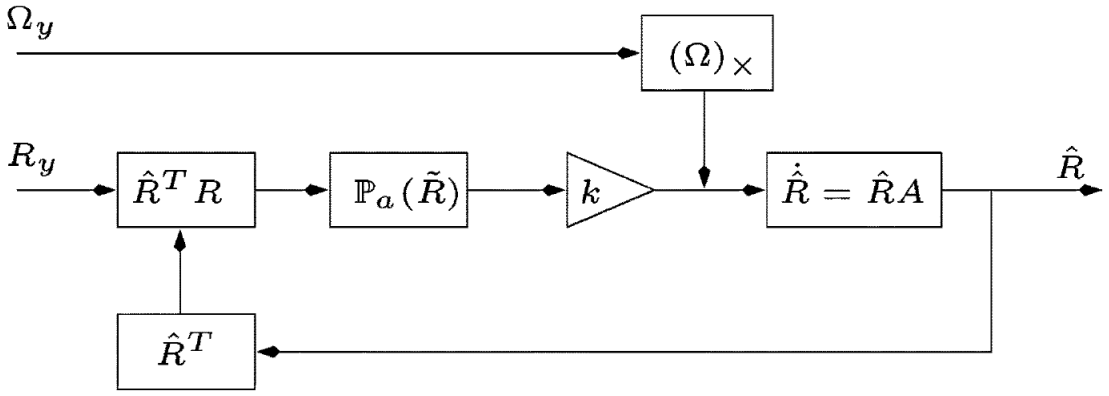


Fig. 4.8: Block diagram of the simplified passive nonlinear complementary filter.

where $\hat{b}$ is the estimated bias and k_b is a bias gain which needs to be greater than 0. The passive filter can be written as (4.6) with no pre-multiplication of Ω by R. The minimum value for the pre-set gain, k_p, of this filter is determined using a Lyapunov Argument, although the calculated minimum value for stability is much lower than the practical value used. Bias is estimated using (4.7). In actual flights, gyroscope drift has been found to be on the order of 0.2 deg/s per minute. Figure 4.9 shows the angular rates before and after compensating for drift and estimated bias.

Yaw angle estimation does not benefit from fusion as there is no sensor to give the absolute heading. Yaw angle is estimated by integrating the rate data. As mentioned earlier, indoor environments preclude the use of sensors such as magnetometers that could give absolute heading. However, absolute heading is obtained using a camera in this work and it discussed further in Chapter 5.

4.4.3 Performance and Motivation

The nonlinear passive complementary filter is found to perform significantly better than the Kalman filter or accelerometer only angles, as can be seen in Figs. 4.10 and 4.11 from data taken during a typical hover. KFs are proven to be optimal estimators for linear systems with Gaussian noise, however, most real systems are nonlinear and have non-Gaussian noise, and this might be the key reason why the NCF outperforms the KF. An

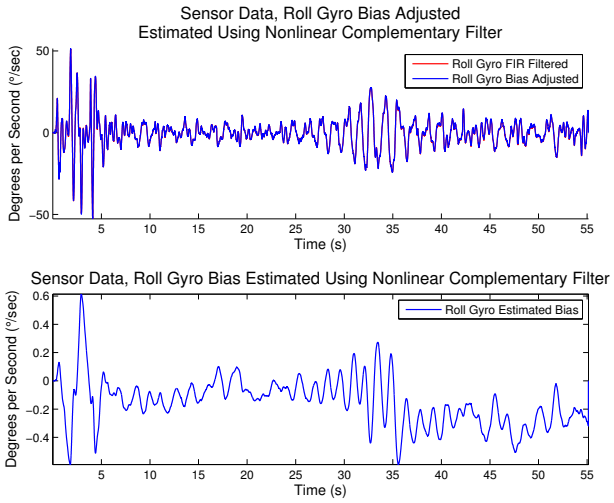


Fig. 4.9: Gyroscope bias estimated by the filter, and bias adjusted gyroscope compared to unbiased data.

advantage of the Kalman Filter is the adaptive calculation of the Kalman gain as compared to the NCF where the gain is decided beforehand. However, one does need to provide the noise variance for both the process and update steps in a KF. No such parameters need to be tuned in a NCF.

Stable flight was not achieved using the Kalman filter. Moreover, this particular nonlinear complementary filter implementation has a second feedback loop that makes use of the filtered attitude $\hat{R}$ in the angular velocity term as opposed to R_y which is derived from the noisy accelerometer data. This has the advantage of avoiding corrupting $\hat{R}$ with the noise and errors in the reconstructed pose from the accelerometers. Using R_y leads to the direct nonlinear complementary filter, which has stronger stability properties. None the less, the passive complementary filter was found to have superior performance.

4.4.4 Challenges

The challenges in implementing the attitude estimation algorithms on a gumstix were the lack of a FPU and limited memory. The limited on board memory prohibited us from cross-compiling the standard matrix libraries to the gumstix. Given the lack of an FPU, routines for matrix computations needed to be hand coded. In general, care needed to be

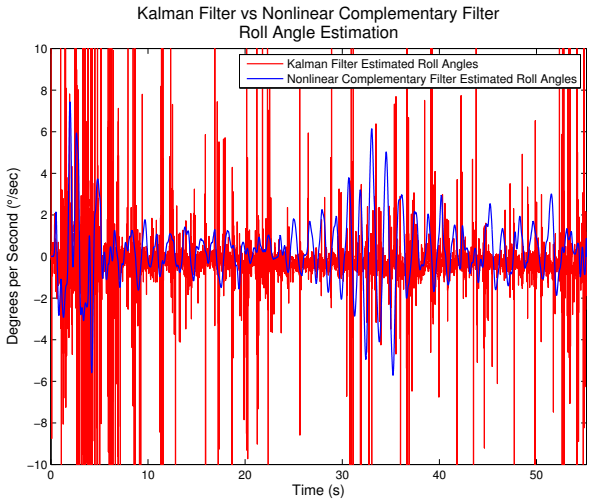


Fig. 4.10: Roll angles estimated using Kalman filter vs nonlinear complementary filter for a typical hover flight.

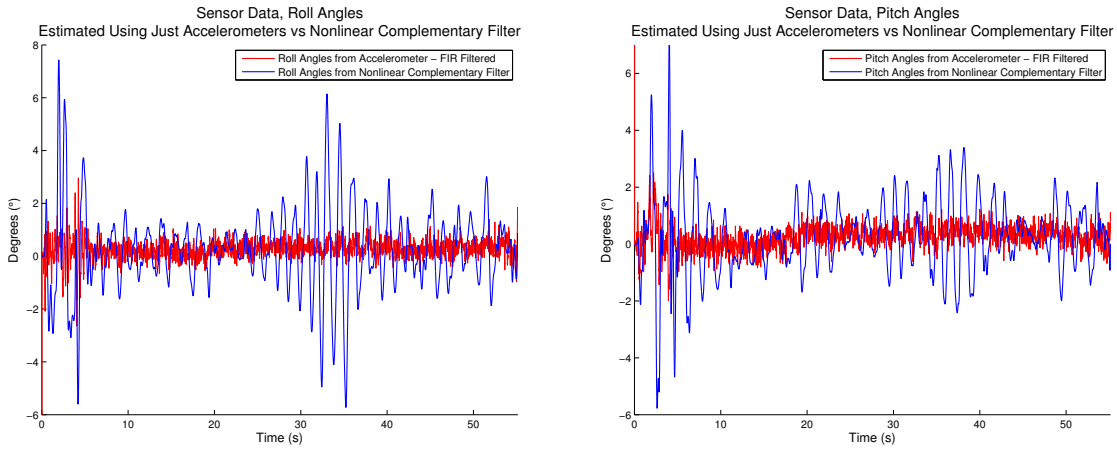
taken to reduce floating-point operations as they directly impact the update rate in our current configuration.

4.5 Attitude Stabilization

The attitude controller utilizes three independent PD controllers for roll, pitch, and yaw stabilization. PD control makes no assumptions and is robust to unmodeled effects. The quadrotor is an under-damped system, and therefore a proportional controller would be insufficient.

4.5.1 Model-Independent Control

The quadrotor is a highly unstable and nonlinear system that necessitates the need for a high bandwidth controller. The attitude stabilization controller runs at a frequency of 400 Hz on the Gumstix. While some competing systems run at up to 1 kHz, given the bandwidth of the sensors and especially the much slower motor response time, it was determined that no additional advantage might have been gained with a higher update rate.



(a) Roll axis.

(b) Pitch axis.

Fig. 4.11: Angles estimated using nonlinear complementary filter vs angles estimated using only accelerometers.

The attitude controller is shown in Fig. 4.12 and is governed by

$$u_\phi = k_{p,\phi} \cdot (\phi^{des} - \phi) + k_{d,\phi} \cdot (\dot{\phi}^{des} - \dot{\phi}), \tag{4.8}$$

$$u_\theta = k_{p,\theta} \cdot (\theta^{des} - \theta) + k_{d,\theta} \cdot (\dot{\theta}^{des} - \dot{\theta}), \tag{4.9}$$

$$u_\psi = k_{p,\psi} \cdot (\psi^{des} - \psi) + k_{d,\psi} \cdot (\dot{\psi}^{des} - \dot{\psi}), \tag{4.10}$$

where the superscript, des, indicates the desired angle or angular rate. $k_{p,\phi}, k_{p,\theta}, k_{p,\psi}$ are the proportional control gains; and $k_{d,\phi}, k_{d,\theta}, k_{d,\psi}$ are the derivative control gains.

An integrator is specifically not used as it was found to increase in one direction (not necessarily the same direction each flight) making the quadrotor unstable. To verify this, the quadrotor was commanded to hover using data from the camera. The quadrotor stayed in place during this time, which should have kept the net angle at zero, however, the integrator continued to build in one direction. This can be attributed to sensor noise.

A signum function is sometimes added to provide more robustness to PID control. Its use is probably inspired from the Sliding Mode Control literature. However, in the case of the quadrotor, results using PD control without adding the signum function were found to be better as compared to with using it. Adding the signum function caused a more

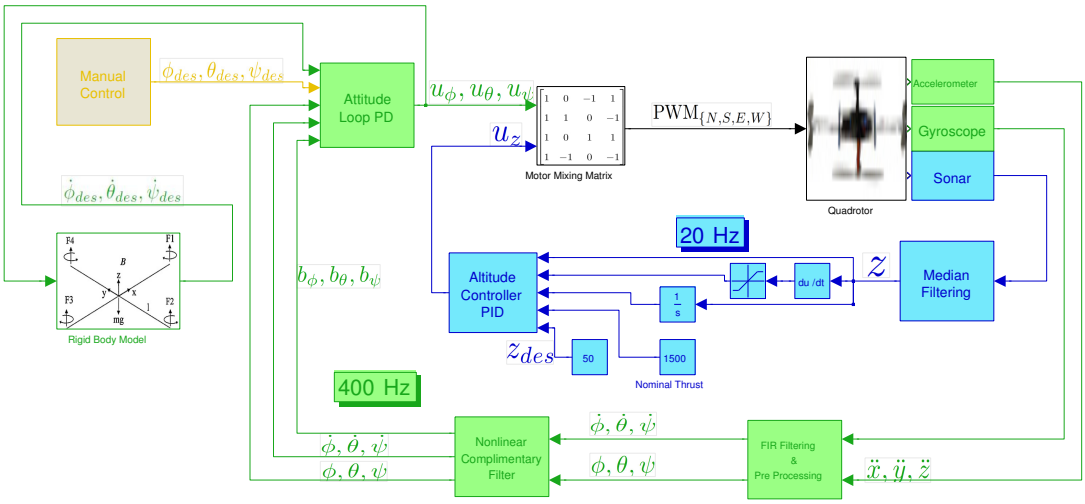


Fig. 4.12: Control system block diagram of attitude and altitude controller. The green blocks and wires indicate the attitude control system; the blue blocks and wires are for the sonar altitude controller. A backup manual controller for safety is shown in a tan color.

oscillatory signal.

The output obtained from the controller is added to the nominal value of control. As discussed in the next section, the nominal value of control is 1500. This value is called the trim and can be adjusted to better reflect the actual nominal values. The trims need to be set experimentally.

4.5.2 Results

Figures 4.13 and 4.14 show the roll and pitch angles and angular velocities. The angles are within ±2 deg and angular velocities within ±20 deg/s. The roll angles are a little worse due to structural issues. Figure 4.15 shows the gain-adjusted contributions of the proportional and derivative part of the PD controller along with the PD controller output in terms of PWM. Figure 4.16 shows the yaw angle and angular rate and these can be seen to be within ±2.5 deg and ±5 deg/s, respectively.

4.5.3 Model-Based Control

The dynamic model of the quadrotor is used for calculating the desired angular rates

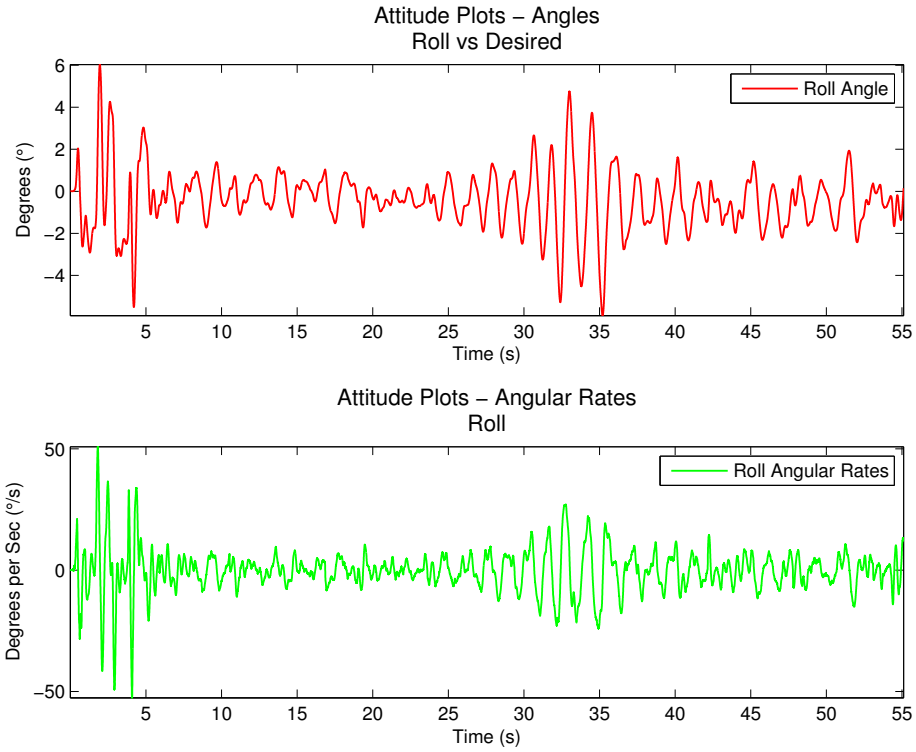


Fig. 4.13: Roll angles and angular rates.

$\dot{\phi}^{des}, \dot{\theta}^{des}, \dot{\psi}^{des}$. Using the approximation that the rotation matrix in (3.5) is identity, and linearizing (3.6) about the hover point with small angle approximations, we get

$$\dot{\phi}^{des} = \frac{4k_F L\omega_h}{I_{xx}}(u_\phi k_{\text{rpm}} + c_{rpm}), \tag{4.11}$$

$$\dot{\theta}^{des} = \frac{4k_F L\omega_h}{I_{yy}}(u_\theta k_{\text{rpm}} + c_{rpm}), \tag{4.12}$$

$$\dot{\psi}^{des} = \frac{8k_M\omega_h}{I_{zz}}(u_\psi k_{\text{rpm}} + c_{rpm}), \tag{4.13}$$

where $L = 23.2$ cm is the distance from the axis of rotation of the rotors to the center of the quadrotor. k_{rpm} converts the u_ϕ, u_θ, u_ψ PD commands to RPMs, as they are in terms of PWM values, and is determined to be equal to 10 when around the hover thrust region. An offset, c_{rpm}, is needed to match the nominal RPM with the nominal PWM. The values $u_\phi,\ u_\theta,\ u_\psi$, are determined using (4.8), (4.9), and (4.10), and the terms, $I_{xx},\ I_{yy},\ I_{zz}$, are the diagonal elements of the inertia matrix. The inertia matrix was calculated using direct

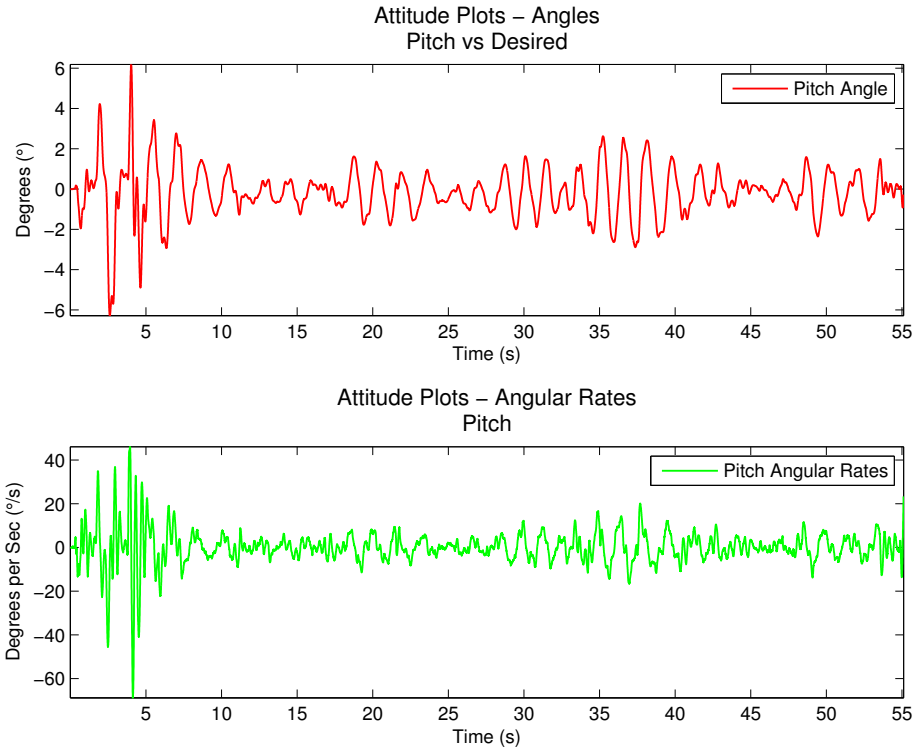


Fig. 4.14: Pitch angles and angular rates.

measurements of the distances and masses of the quadrotor, being as accurate as possible by calculating large components as containing subsets of smaller components. The determined inertia matrix in $kg - m$ is

$$\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} = \begin{bmatrix} 0.02509 & 0.00016 & -0.00276 \\ 0.00016 & 0.02610 & 0.00070 \\ -0.00276 & 0.00070 & 0.02262 \end{bmatrix}. \tag{4.14}$$

Figure 4.17 shows the desired and achieved roll and pitch angular velocities.

4.5.4 Disturbance Rejection Characteristics

The controller was tested for its disturbance rejection capability and it performed admirably. It was pushed down, pulled hard below the motors, whacked with a stick, and in all cases it was able to recover from these perturbations in a very short amount of time.

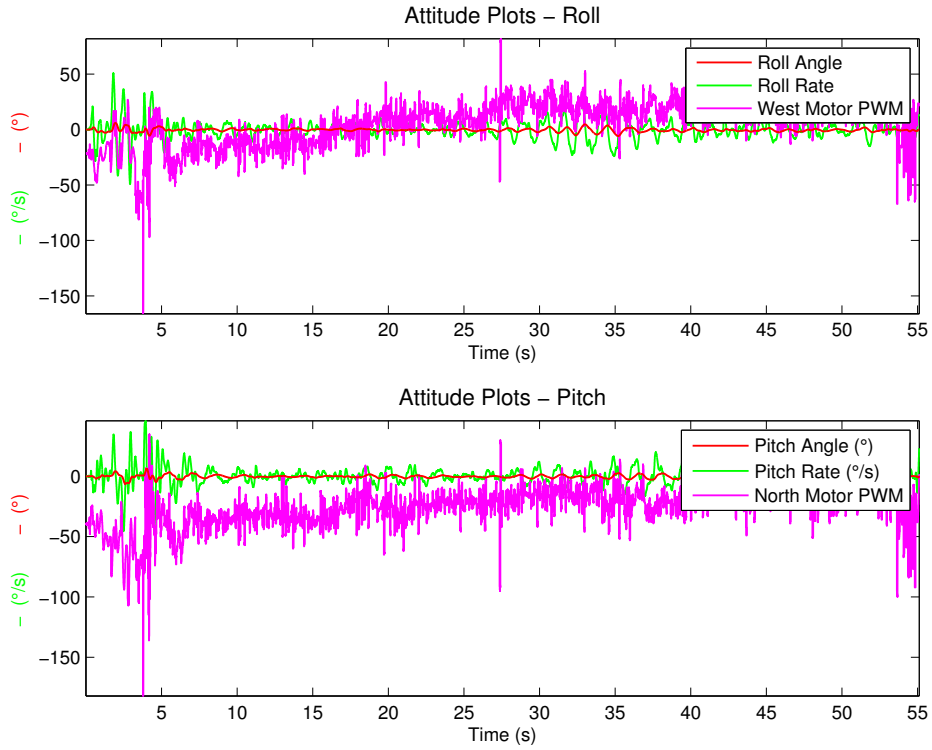


Fig. 4.15: Roll and pitch and motor commands.

An observation made while tuning the gains was that better disturbance rejection capability is achieved using a relatively softer set of gains as compared to the gains needed for a precise hover.

4.5.5 Gain Tuning

It is hard to do system identification of an unstable system, therefore, most of the tuning of gains is empirical. Usually the derivatives are tuned first, with the proportional gains set to zero. This gives the quadrotor the ability to stabilize at an angle and with no absolute information about its angle, it may tend to stabilize at a non-zero angle. This behavior is corrected using manual control. The proportional gains are tuned next giving the quadrotor the ability to stabilize at a desired angle. Later on, methods such as relay tuning, Ziegler Nichols were also considered, but found unsuitable to be used with a highly unstable system [51].

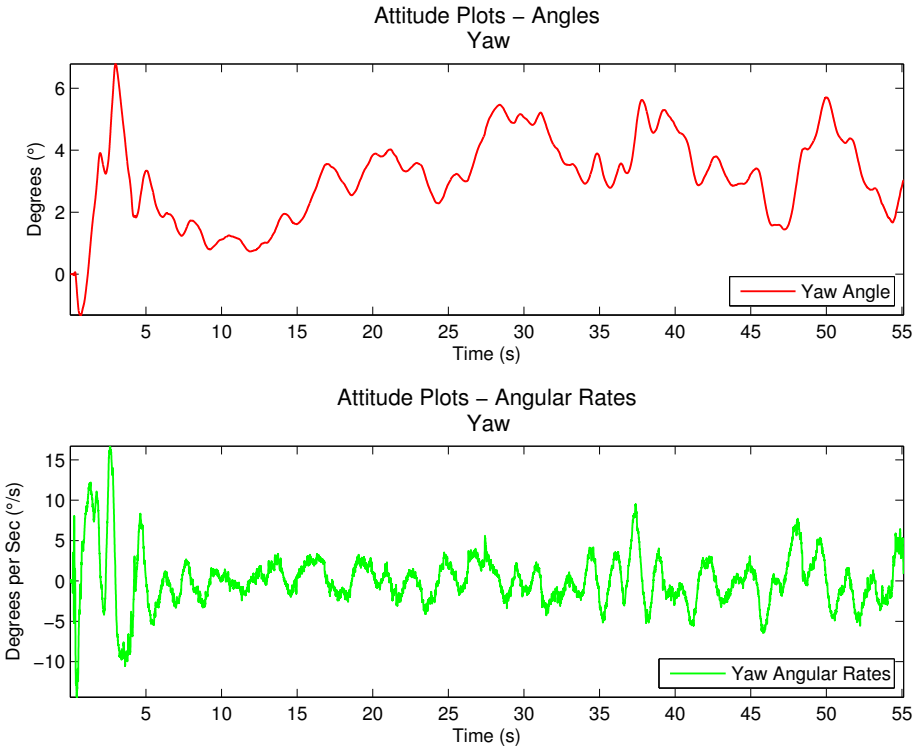
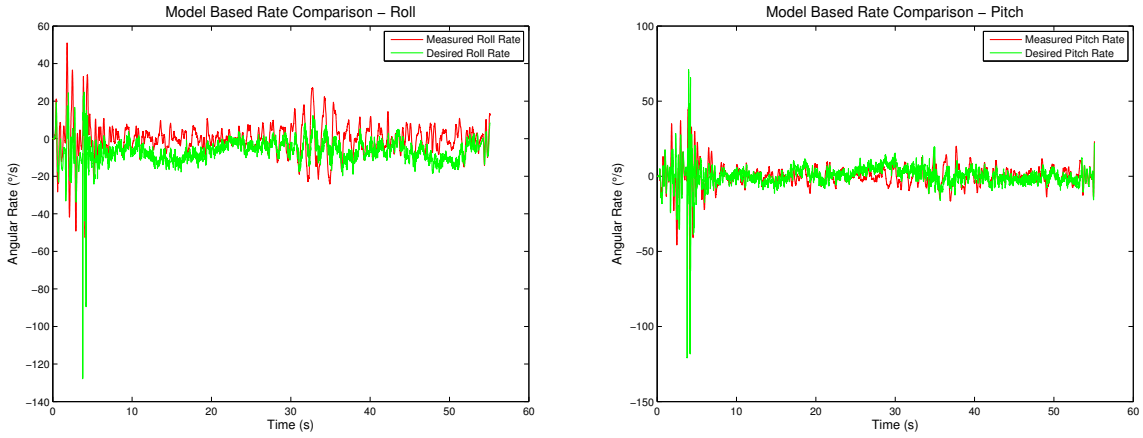


Fig. 4.16: Yaw angles and angular rates.



(a) Roll axis.

(b) Pitch axis.

Fig. 4.17: Desired angular rates vs achieved angular rates.

4.5.6 The Role of Trims

Trims play an absolutely essentially role in achieving good attitude stabilization. It was discovered that the the pusher blades generated more thrust than the tractor blades and this lead to a slight yaw in the quadrotor. This is compensated using trims. Despite being a symmetric platform, the placement of computing hardware, sensors or other payload necessitates the need for trimming. Another situation where trimming is essential is with worn out propellors. Worn-out propellors usually display a huge drop in the amount of thrust they can generate.

4.6 Altitude Control

Altitude control is a difficult problem due to nonlinear effects such as the interaction of the downwash from the four rotors, airflow disruption due to ground effect, battery discharge dynamics, total thrust variation, and blade flapping [44, 52]. This is further complicated by the fact that sonars typically suffer from non-gaussian noise, false echos, and dropouts. A Maxbotix sonar is used to obtain altitude data, however, the high quantization (1 inch, 2.54 cm) and low-bandwidth (20 Hz) of the sensor limit the ability to get high quality data. A Median Filter of window size 3 is used to prevent outliers from corrupting the data. This leads to a delay of one sample time or 50 ms.

4.6.1 Height Controller

A PID controller (4.15) with a feedforward term in the form of nominal thrust is implemented to stabilize the height. PID was considered ideal especially due to its ability to cope with unmodeled effects.

$$u_{alt} = k_{p,alt} \cdot (z^{des} - z) + k_{i,alt} \cdot \int_0^t (z^{des} - z)dt + k_{d,alt} \cdot (\dot{z}^{des} - \dot{z}) + u_{nom}, \quad (4.15)$$

where z^{des} is the desired height, $k_{p,alt}, k_{i,alt}, k_{d,alt}$ are the PID gains, and z indicates the measured height from the ground by projecting the sonar reading onto the inertial z axis,

using

$$z = (\cos\phi \cdot \cos\theta) \cdot z_{\text{sonar}}. \tag{4.16}$$

The quadrotor operates in a hover condition most of the time, however, the sonar would misestimate the height when it is banked at sharp angles. Therefore, the height returned from the sonar z_{sonar} is projected onto the inertial z-axis. In (4.15), the velocity, $\dot{z}$, is obtained by finite differentiation of the input sonar measurements.

One of the ideas tested to compensate for the delay in the altitude data was to introduce a predictive term such as the z acceleration, but, this was found to be very noisy. Integration was used as a means to smooth out this signal and obtain a velocity estimate, however, this had a tendency to accumulate in one direction and was unusable.

One of the major issues with using the sonar has been that it sometime sees the camera wire, and therefore misreports the height from the ground. This problem could be solved by fusing data from the sonar and the camera.

4.6.2 Results

The top graph in Fig. 4.18 shows the position error in z. The quadrotor is able to maintain a stable altitude within a range of ± 6 cm and instantaneous velocities are bounded within ± 50 cm/s (as seen in the middle graph). The error in altitude is greatly dependent on the low-resolution and low-bandwidth of the sonar. The bottom graph shows the accumulation of integral error to compensate for the drop in battery voltage. The graph in Fig. 4.19 shows the gain-adjusted contribution of each term in the PID. The thrust commands (output of the PID) to the motor are shown in magenta.

4.7 Actuation

The controller outputs, $u_\phi, u_\theta, u_\psi,$ and u_{alt}, are inputs to the ESCs in terms of PWM values; and they are converted to individual motor commands using

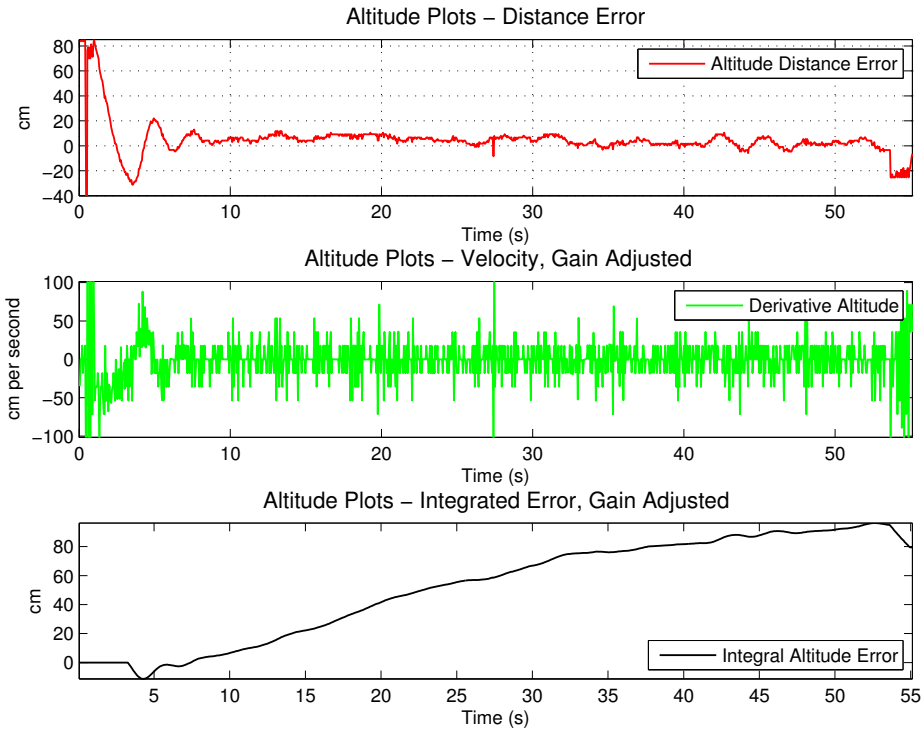


Fig. 4.18: Altitude data showing position error, velocity, and integral error.

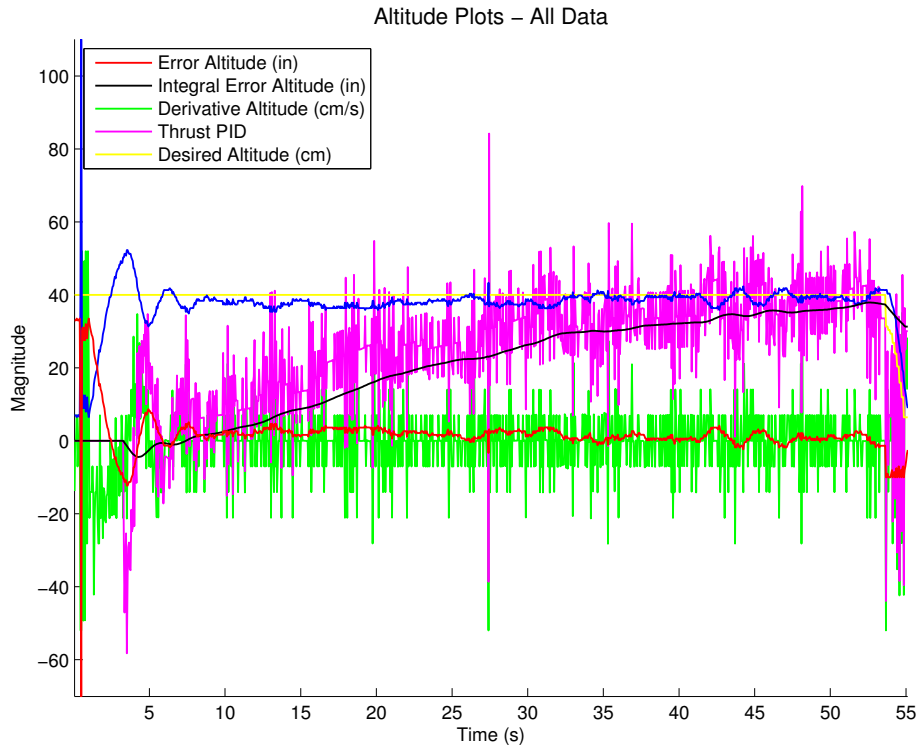


Fig. 4.19: Altitude data showing gain-adjusted contribution of each term of PID.

$$\begin{bmatrix} u^1_{\text{PWM}} \\ u^2_{\text{PWM}} \\ u^3_{\text{PWM}} \\ u^4_{\text{PWM}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_{alt} \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix}, \tag{4.17}$$

where u^i_{PWM} denotes the output PWM value to motor i. The u^i_{PWM} outputs to each motor are related to the airframe torque of the dynamic model (3.6) with

$$\omega_i = k_{rpm} \cdot u^i_{\text{PWM}} + c_{rpm}, \; i \in 1,2,3,4, \tag{4.18}$$

$$\tau_a = (\tau^1_a, \tau^2_a, \tau^3_a)^T, \tag{4.19}$$

$$\tau^1_a = L \cdot b(\omega^2_2 - \omega^2_4), \tag{4.20}$$

$$\tau^2_a = L \cdot b(\omega^2_1 - \omega^2_3), \tag{4.21}$$

$$\tau^3_a = k(\omega^2_1 + \omega^2_2 + \omega^2_3 + \omega^2_4). \tag{4.22}$$

This attitude and altitude controller achieve successful stabilization of the roll, pitch, yaw, and height of the quadrotor, allowing the addition of a vision navigation loop for position control of the vehicle. Attitude and altitude system gains used to achieve stabilization are shown in Table 4.1.

4.8 Automatic Landing

An automated landing procedure is used to ensure a smooth landing. It is also used as a safety measure if communication is lost either over Zigbee or Wi-Fi. The height controller is employed in order to aid automatic landing. The desired height is reduced after fixed intervals of time and finally an open-loop landing sequence is invoked at a height of 10 inches. A final landing sequence is needed due to the inability of the sonar to resolve distances smaller than 6 inches.

Table 4.1: Attitude system gains.

Gain Term	Notation	Value
Roll Proportional	$k_{p,\phi}$	4.0
Pitch Proportional	$k_{p,\theta}$	4.0
Yaw Proportional	$k_{p,\psi}$	8.5
Roll Derivative	$\dot{k}_{p,\phi}$	0.65
Pitch Derivative	$\dot{k}_{p,\theta}$	1.0
Yaw Derivative	$\dot{k}_{p,\psi}$	3.5
Altitude Proportional	$k_{p,alt}$	2.9
Altitude Integral	$k_{i,alt}$	0.025
Altitude Derivative	$k_{d,alt}$	1.0
Altitude Nominal	u_{nom}	1505
Nonlinear Complementary Filter Proportional	k_{NLF}	1.0
Nonlinear Complementary Filter Bias	k_b	0.3

4.9 Manual Control

A gamepad attached to the host computer is used as a manual controller. This is an important safety measure especially when testing new control algorithms or sensing modalities on the quadrotor as this gives the ability for a human to override the automatic control and invoke the automatic landing routine to land the quadrotor safely or in extreme circumstances use an emergency stop button. The controller can also be used to pilot the quadrotor by controlling the roll, pitch, yaw, and thrust.

Two different methods to accomplish this are tested.

1. Direct-actuation control - The manual commands $u_{\text{man},*}$ are directly control the motors to allow for more reactive control, transforming (4.17) to

$$\begin{bmatrix} u^1_{\text{PWM}} \\ u^2_{\text{PWM}} \\ u^3_{\text{PWM}} \\ u^4_{\text{PWM}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \cdot \left(\begin{bmatrix} u_{alt} \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} + \begin{bmatrix} u_{\text{man,thrust}} \\ u_{\text{man},\phi} \\ u_{\text{man},\theta} \\ u_{\text{man},\psi} \end{bmatrix} \right). \tag{4.23}$$

The problem with this was the conflict between the attitude and manual controllers, with one trying to counteract the other. This manifested in somewhat jerky behavior

of the quadrotor with it being unable to maintain an accurate hover.

2. Desired angle control - $u_{\text{man},\phi}$, $u_{\text{man},\theta}$, and $u_{\text{man},\psi}$ become the the desired angles ϕ_{des}, θ_{des}, and ψ_{des} and are directly fed to (4.8), (4.9), and (4.10), respectively. Thrust is still directly added to the motors at the actuation stage.

4.10 Latency in the System

As discussed in Chapter 2, latencies can often lead to instability, and therefore every effort was made to minimize them. Figure 4.20 shows the latencies and delays within the system.

Some sources of latencies are enumerated below.

1. ADC sampling - Communication between the Gumstix and Robostix is interrupt based, with the ADC on the Robostix sampling continuously and only providing the data when the Gumstix asks for it. This is in contrast to the previous implementation where the Robostix would sample the ADC and send the data as part of the request.

2. Single instruction multiple reads/writes - Multiple reads or writes are implemented in a single command much in the spirit of Single instruction, multiple data (SIMD). Despite this, some basic profiling using the processor's built-in timer showed that almost 40% of the total loop time is required just to get sensor data.

3. I^2C - The I^2C protocol is set to run at 400 Kbps, which is the fastest the devices used will allow. Rough net throughput of data for the protocol is 27.5 Kbytes/sec plus an estimated overhead of 0.3 ms.

4. Data logging delays - Printing data onscreen decreases the loop update frequency by more than 10%. Printing to a file onboard the Gumstix is less damaging, but still leads to large spikes in loop time periodically while the file buffer is loaded into the file. These spikes are typically 90 ms in duration as shown in Fig. 4.21. Even with such periodic, excessive delays, the attitude system is still able to maintain level flight in the presence of such latencies.

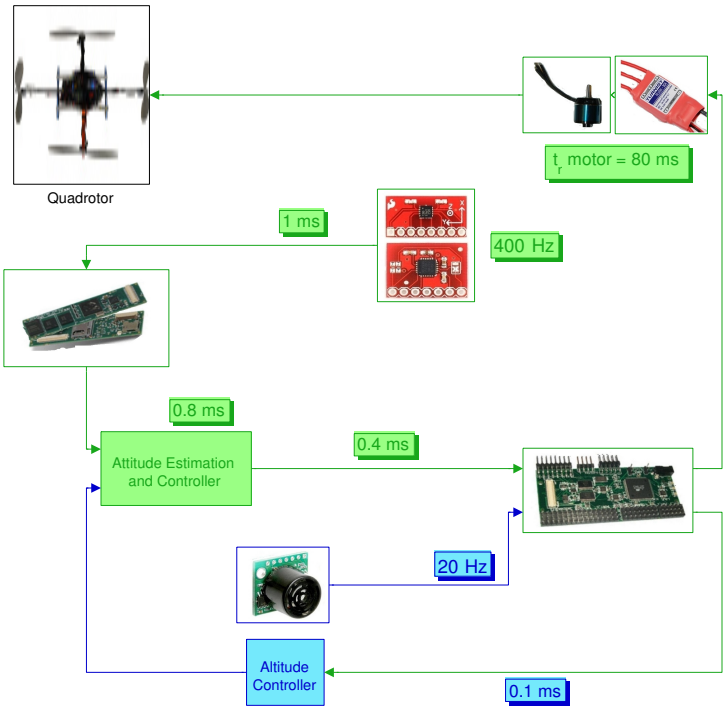


Fig. 4.20: Sources of latencies in attitude control.

4.11 Limitations

The nonlinear filter is implemented using Euler Angles rather than quaternions, therefore precluding the quadrotor from performing acrobatic maneuvers requiring angles greater than 180° such as flips.

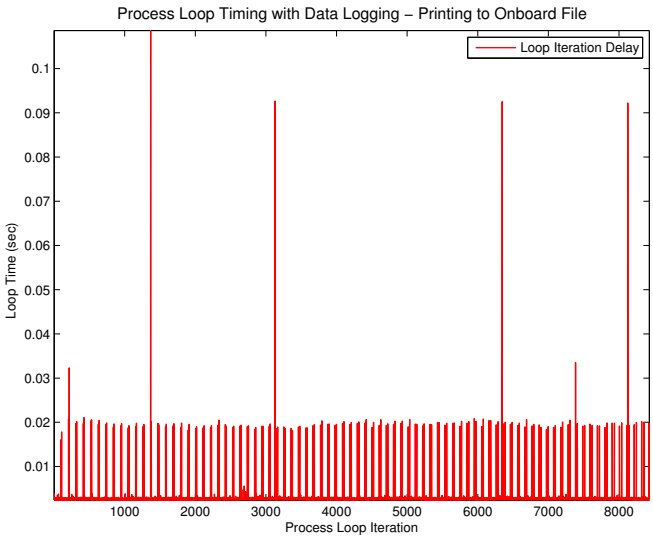


Fig. 4.21: Process delays with data logging. Note the spikes at around 90 ms, plus the more frequent 20 ms loop latencies. However, since the vast majority of loops are at the floor level of around 2.4 ms, the average loop frequency is 400 Hz.

Chapter 5

Vision-Based Hover and Navigation

The previous chapter focused on controlling the quadrotor's nonlinear and unstable dynamics to maintain a stable attitude and altitude. This chapter focuses on the problem of controlling the position and heading of the quadrotor and endowing it with the capability to navigate indoors. In fact to even maintain a stable hover, it is important to control the position. This is due to the fact that any small deviation in roll or pitch from zero would cause the quadrotor to drift. These small deviations are imperceptible to MEMS-grade sensors, necessitating the need for exteroceptive sensor that senses in the world frame to report the x, y, and z position of the quadrotor. For quadrotors operating outdoors, a GPS receiver coupled with a magnetometer perform appreciably, however, both these modalities are absent indoors, making the problem of indoor navigation challenging.

The solutions proposed in this chapter are applicable more broadly to any area without GPS coverage such as "urban canyons," tunnels or another planet. Results from a vision-based outdoor flight are presented at the end of the chapter to support this.

5.1 Why Is Indoor Navigation for MAVs Hard?

Ground robots can be equipped with optical encoders that provide high frequency relative measurements and these coupled with exteroceptive sensors such as laser range finders, cameras, etc., can be used to obtain fairly accurate estimates of the position of the vehicle. Additionally, there are no constraints on the weight or power consumption of the sensor. The situation changes drastically for a MAV and more so for quadrotors. At best, one can obtain indirect odometry, and dead-reckoning is not an option given the noisy data returned by the accelerometers.

Quadrotors operating outdoors use GPS and are able to obtain fairly reliable position

estimates, whereas, indoor navigation precludes the use of GPS for position or magnetometers for yaw. Moreover, quadrotors navigating outdoors can tolerate errors on the order of a meter, however, indoor navigation requires higher accuracy. Also, position estimation needs to be done in real-time as a quadrotor is in constant motion and cannot stop to revaluate its position. While smoothness assumptions can be made for motion models for fixed wing MAVs, no such assumptions can be made in the case of a quadrotor.

5.1.1 Possible Approaches

A variety of sensors have been employed to endow MAVs with navigation capabilities. There are various possible approaches to this problem.

1. SONAR - These typically provide short-range, low-resolution data and are power hungry.

2. Laser Range Finders - These are incredibly accurate ranging sensors that can report bearing and range to obstacles in a 180° to 270° arc. However, these are expensive costing upwards of $2000, heavy and can have catastrophic failure modes when used in environments with glass or mirrors. Even if a LRF is off by a huge margin, it would report the incorrect range with a high certainty.

3. Beacons - Triangulation using artificially placed beacons is an option, however, this would require an infrastructure change which is not always possible. Most buildings are today equipped with Wi-FI and multiple routers could be used to triangulate the position. However, this is a hard estimation problem given the fact that the noise is non-gaussian. Current estimation schemes are only able to estimate with an accuracy of 10 cm, which is insufficient for an indoor flying platform.

4. Camera - It is a low-cost, light-weight, and low-power sensor that provides very rich data. Moreover, it can be used for providing both navigation capabilities as well as relaying a video feed to a human operator. On the flip side, extracting information

from the rich data provided is a challenging problem. Also, cameras can only provide bearing information and cannot operate in dark or textureless environments.

After carefully evaluating all the possibles approaches, a camera was the sensor of choice.

5.1.2 Overview of the Proposed Approach

This chapter proposes the use of a technique known as Simultaneous Localization and Mapping (SLAM) in order to estimate the 6-DOF pose of the vehicle. A navigation loop employing PID is designed to utilize the position data to close the loop. A cascaded controller approach is employed with the high frequency attitude loop (400 Hz) being the inner loop and a lower frequency navigation loop running at 30 Hz being the outer loop. The position reported by the SLAM algorithm are used to close the navigation loop. The navigation loop sends desired accelerations to the attitude loop, which are then transformed to desired angles with the knowledge of the model of the quadrotor. Moreover, the yaw reported by the camera is more accurate than that achieved by integrating the yaw angular rate. The yaw reported by the gyro and camera are fused to get a better estimate of the heading. Experimental results of the quadrotor performing trajectory tracking both indoors and outdoors are presented.

5.2 Related Work

This section discusses the related work, categorized based on the methodology used.

5.2.1 Vision-Based Hovering Using a Known Target or Artificial Markers

One of the earliest works used an external camera and restricted the quadrotor to vertical and yaw motion [53]. This work was further developed to use a dual-camera system with one camera on the quadrotor and another on the ground tracking colored blobs [54]. Position and yaw control was achieved using a tethered quadrotor [55]. Tournier et al. used an onboard camera looking down on an optical grating with a special pattern called the Moiré pattern to extract 6-DOF information. This required the target pattern to be backlit

and the position and yaw was used for control [56]. Romero et al. presented results on position and yaw control of a quadrotor by observing individually identifiable markers on the ground with known geometry [57]. Guenard et al. demonstrated a tight hover of 10 cm around the desired position using a camera looking at 4 black marks in a square configuration on the ground [58]. Bourquardez et al. also present results on position control using image-based visual servo control [59]. Results using an external camera tracking a blob on the quadrotor are presented by Hoffmann et al. [60]. A recent paper used the ARtoolkitplus to provide localization information to the quadrotor [61, 62]. Most approaches described in this paragraph employ visual servoing and have a limited flight envelope.

5.2.2 Optical Flow

Many different methods for computing optic flow exist and are discussed in a paper by Beauchemin and Barron [63]. Optical flow techniques on their own are typically only able to generate a fairly inaccurate depth map and need to be augmented with other techniques to provide navigation. Therefore, they can only be used to provide a stable hover. One of the first successful position hover using optical flow was achieved by Fowers et al., Harris feature detection using template matching running in real-time in hardware on an on a Field-programmable gate array (FPGA) onboard the quadrotor [64]. However, image matching or correlation methods lack sub-pixel precision, have quadratic computational complexity and are inaccurate in presence of image deformation due to rotation. Positional control was also attempted using optical flow by Romero et al. in their 2006 paper, however, the authors remark that unstable behavior was noticed due to the change in attitude of the downward looking camera mounted on the quadrotor. This motivated the development of a new platform in an eight-rotor configuration that decoupled the translational and rotational dynamics leading to more stability [65]. The pyramidal implementation of the KLT optical flow tracker available in OpenCV is used and a hover of about ± 0.5m is demonstrated [66, 67].

An on-board implementation of optical flow using a field of view (FOV) camera with a small catadioptric mirror to provide 360° view is used to follow a corridor path with artificial

textures [68]. Combined block-matching and differential algorithm optic flow-based vision system is developed for autonomous 3D localization and control of small aerial vehicles [69]. A comparison of nonlinear controllers applied to a quadrotor utilizing the pyramidal KLT tracker in Open CV is given [70]. A commercial available quadrotor, AR Drone also employs optical flow to maintain a stable hover, however, the specifics of the implementation are not known [10].

5.2.3 Navigation Using Laser Range Finders

Successful instances of indoor navigation have been demonstrated using a Laser Range finder mounted on a quadrotor. Grzonka et al. present a quadrotor capable of estimating its relative position using a scan matching technique and mapping using a technique presented by Hahnel et al. [71,72]. It localizes using Monte Carlo Localization in indoor environments using a pre-generated map acquired with a ground robot [73]. Some results for SLAM with the quadrotor under manual control have also been presented. A similar approach was taken with an of-the-shelf quadrotor at MIT [74,75]. Scan matching is used to provide estimates of the uncertainties for the motion model and Gmapping is used to perform SLAM [76,77]. Some preliminary results for pose estimation using Visual Odometry (VO) are also presented and authors report higher oscillation as compared to LASERs.

5.2.4 Navigation Using Vision-Based Techniques

A downward facing low quality monocular camera was successfully used for localization of a blimp, using offboard processing and some assumptions about the scale of the feature scene [78]. Using monocular vision SLAM for low resolution 3D mapping offboard, a quadrotor was able to perform drift free hover and execute a path through obstacles within a controlled environment, using the Vicon system to simulate IMU measurements [79]. Monocular vision-based SLAM using corner-like features and straight architectural lines is implemented on a small helicopter [80]. Using a commercially available quadrotor, a similar approach to SLAM as the current thesis is used [81]. VO using a stereo camera and the pyramidal Kanade-Lucas-Tomasi (KLT) feature tracker in Open CV is used for navigation [82].

An algorithm combining data-driven image classification with optical flow techniques is used for localizing a self-stabilizing co-axial helicopter in a pre-mapped environment [83].

5.3 Simultaneous Localization and Mapping Using a Camera

Simultaneous Localization and Mapping is the problem of estimating the pose of the robot and a map of its surroundings. Initially, the primary sensor of choice for mapping was a Laser Range Finder that is able to give the distance and bearing to obstacles in the environment very accurately. More recently, there has been a focus on using a camera and this makes the problem much more challenging.

Ego estimation using a camera is usually done either by tracking distinctive features in the images or using dense motion algorithms such as optical flow, which track image intensities in order to give a motion flow field. The approach of feature tracking was chosen for this work owing to computational requirements and the difficulty of extracting environment geometry from optical flow.

The problem of feature-based visual SLAM is typically solved using two different approaches. First is a Kalman filter (KF) based approach, in which the state vector consists of the 6-DOF pose of the vehicle (also, velocity and acceleration depending on the model used), and is augmented to include distinctive landmarks in the environment as they are observed [84]. The drawback of this approach is the continuously expanding state vector, as even the most efficient matrix inversion algorithms require on the order of $O(n^{2.376})$ time to invert a matrix. To address this issue, a small number of high quality features that are invariant to changes in scale, viewpoint, rotation, and illumination to an extent, such as the Scale Invariant Feature Transform (SIFT) are employed, however, this leads to a very sparse map [85]. Moreover, the linearization of the motion and measurement model takes a toll on the accuracy.

Unscented KF based approaches have been proposed to rectify this, but still need to deal with matrix inversion. A related approach to this is the FastSLAM algorithm that maintains multiple hypotheses of the vehicle pose and map and is computationally more efficient [86].

Recently, studies have concluded that maintaining a high number of relatively low quality features and using Structure from Motion (SfM) techniques leads to more accurate maps [87]. This was the motivation of choosing the second approach, SfM, over Kalman Filter-based techniques.

5.4 PTAM

The outlined approach uses the visual SLAM algorithm developed by Klein and Murray [88]. Feature tracking is employed to get visual odometry and then Bundle Adjustment is applied to take care of drift and to refine estimates [89]. The approach uses a large number of but relatively lower quality Features from Accelerated Segment Test (FAST) corner features [90]. This SLAM algorithm is highly capable of tracking and mapping from a single camera, and the motion model employed copes well with sudden accelerations that occur on the quadrotor. The basic approach of the algorithm is the splitting of the tracking and mapping tasks into two separate threads.

The tracking thread runs continuously and performs feature detection and matching based upon each successive frame, in order to compute an estimate of the current camera pose. It does this through a robust mechanism of extracting and matching features at multiple scales. New points are initialized using epipolar search and for added accuracy refinement, Bundle Adjustment is run on the last few frames. Tracking is done at 30 Hz.

The mapping thread runs a subset of all the frames, called key frames, through Bundle Adjustment to generate a more accurate map. These key frames are distinctive frames that are selected on heuristic criteria such as minimum euclidean distance between key frames, passage of minimum frames, etc. Bundle adjustment solves a nonlinear least squares optimization problem where the objective function is the reprojection error. The reprojection error is the difference between where a feature is observed and where it is expected to be observed, projected on the camera frame. The most popular algorithm to solve this problem is the Levenberg-Marquardt Algorithm, which takes into account not only the gradient but also the curvature. The intuition behind the algorithm is opposite to that of gradient descent, in the sense that larger steps are taken when the gradient is small. More effi-

cient algorithms, such as Sparse Bundle Adjustment, exploit the sparsity of the Hessian Matrix [91].

5.4.1 Implementation Details

Due to the lack of depth information from the monocular camera, there is the problem of the unobservability of the map scale. For rapid-prototyping simplicity, the map scale is initially estimated by hand using the 5-point algorithm to densely initialize the map from a stereo pair. This map initialization serves a dual purpose of also allowing the quadrotor to have a small pre-mapped section where it takes off, as the quadrotor takes off blindly until the navigation system can track within the known area using a recovery procedure, before moving to and mapping new locations [92].

5.4.2 Computation Time

The PTAM algorithm is run on a 2.0 GHz, dual-core computer running Ubuntu. The tracking thread uses 60% of one core, while the mapping thread spikes utilizes 80% of the second core, when global Bundle Adjustment is being done.

5.5 Image Acquisition

5.5.1 Camera Modification

An off-the-shelf webcam Logitech QuickCam Pro 5000 is retrofitted with a 2.1 mm wide-angle lens with a FOV of 81°. This allows the observation of more features and maximizes the persistence of features which allows the SLAM algorithm to deal with jerks.

5.5.2 Camera Calibration

Before using a camera to estimate pose or position, the camera needs to be calibrated. Calibration involves the process of estimating a matrix that gives the relation between the position of a feature in the world and its projected image on the camera. This matrix

includes the focal length, skew and the principal point of the camera and is found to be

$$\begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & \nu_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5547 & 0.7381 & 0.5536 \\ 0 & 0.5534 & 0.9713 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.1}$$

5.6 Navigation Controller

The navigation and attitude (and altitude) controllers are cascaded with the attitude (and altitude) controller forming the inner loop. The ground station receives images from the onboard camera at a rate of about 30 Hz, and then sends the positioning information to the navigation controller after a processing and communication delay of approximately 50 ms. Running the navigation loop at 30 Hz also ensures it being spectrally separated from the attitude loop.

5.6.1 Direct Actuation-Based Control

Initially, the navigation controller was implemented as a PID controller with its outputs $u_{nav,x}$ and $u_{nav,y}$ directly controlling the motors to allow for more reactive control, transforming (4.17) to

$$\begin{bmatrix} u^1_{\mathrm{PWM}} \\ u^2_{\mathrm{PWM}} \\ u^3_{\mathrm{PWM}} \\ u^4_{\mathrm{PWM}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \left(\begin{bmatrix} u_{alt} \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} + \begin{bmatrix} 0 \\ u_{nav,x} \\ u_{nav,y} \\ 0 \end{bmatrix} \right). \tag{5.2}$$

The problem with this was the conflict between the attitude and navigation controllers, with one trying to counteract the other. This manifested in somewhat jerky behavior of the quadrotor with it being unable to maintain an accurate hover.

5.6.2 Model-Based Control

The navigation controller was later changed to a PID controller with the desired linear accelerations as outputs

$$\ddot{x}^{des} = k^x_{p,nav} \cdot (x^{des} - x) + k^x_{d,nav} \cdot (\dot{x}^{des} - \dot{x}) + k^x_{i,nav} \cdot \left(\int_0^t (x^{des} - x) dt \right), \quad (5.3)$$

$$\ddot{y}^{des} = k^y_{p,nav} \cdot (y^{des} - y) + k^y_{d,nav} \cdot (\dot{y}^{des} - \dot{y}) + k^y_{i,nav} \cdot \left(\int_0^t (y^{des} - y) dt \right), \quad (5.4)$$

where the superscript, des, refers to the desired position and velocity, and $k^*_{p,nav}, k^*_{i,nav}$, and $k^*_{d,nav}$ are the proportional, integral, and derivative gains. The integrator can effectively account for small offsets (unlike in the attitude controller) due to in-flight effects such as actuator friction buildup in the bearing and efficiency changes due to heating. As expected, an integrator will always be trying to catch up, essentially adding a delayed proportional term to the system, which can easily lead to unstable results if the integrator term is too large. In this case, utilizing a small integrator in the navigation control noticeably improved the position error. The PID controllers (5.3) and (5.4) were modified to include a translational acceleration term with the idea of adding a feed-forward term to anticipate translational movement of the quadrotor. However, in practice this led to poor performance, and is most likely due to noisy data from the accelerometers.

Finite differentiation of the position is used to obtain an estimate of the velocity, as the accelerometer data is too noisy to give a reliable estimate when integrated. Figure 5.1 shows the divergence of integrated accelerations despite the quadrotor maintaining a stable hover. Navigation gains used to achieve accurate hover and path following are shown in Table 5.1. These have been found empirically.

The outputs of the PID controllers (5.3) and (5.4) give desired accelerations in the inertial frame and the desired angles are calculated after accounting for yaw, using

$$\begin{bmatrix} \phi^{des} \\ \theta^{des} \end{bmatrix} = \frac{1}{g} \begin{bmatrix} -\sin\psi & -\cos\psi \\ \cos\psi & -\sin\psi \end{bmatrix} \begin{bmatrix} \ddot{x}^{des} \\ \ddot{y}^{des} \end{bmatrix}, \quad (5.5)$$

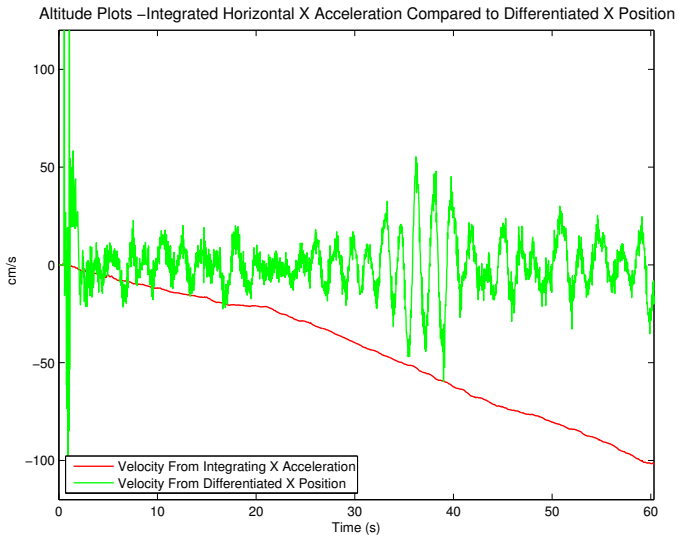


Fig. 5.1: Comparison of calculation of translational velocity during a hover.

with g is gravitational acceleration.

This solves a subtle problem in the case the quadrotor is yawed. Assume the quadrotor is yawed 10°, and is commanded to move 50 cm to the left, the quadrotor would roll to its left and hit a point behind the desired position and would then try to correct. It would move on a diagonal rather than a straight line to reach a point that is directly to is left. This problem is solved using (5.5).

5.6.3 Yaw Control

Integration of the yaw angular rate is found to give insufficiently accurate estimates of the yaw angle over time and this can be attributed to drift due to integration and partly due to gyro drift, despite the use of a bias estimator in the attitude estimation filter. Therefore, the yaw angle, ψ, is determined from fusing an appropriately delayed value of the integrated angular yaw rate with the yaw determined from the SLAM algorithm, using a Kalman filter [49]. The process update step uses the integrated angular rate while the measurement step uses the yaw estimated by the SLAM algorithm, which operates at a frequency that is an order of magnitude lower. The integrated angular yaw rate needs to be delayed as the yaw obtained through the vision algorithm is delayed by 50 ms due to

Table 5.1: Navigation system gains.

Gain Term	Notation	Value
X Proportional	$k^x_{p,nav}$	1.15
Y Proportional	$k^y_{p,nav}$	1.20
X Integral	$k^x_{i,nav}$	0.002
Y Integral	$k^y_{i,nav}$	0.002
X Derivative	$k^x_{p,nav}$	0.70
Y Derivative	$k^y_{p,nav}$	0.95
Navigation Altitude Proportional	$k^{nav}_{p,alt}$	3.1
Navigation Altitude Integral	$k^{nav}_{i,alt}$	0.023
Navigation Altitude Derivative	$k^{nav}_{d,alt}$	1.2
Yaw KF Process Noise Variance	Q^ψ	0.1
Yaw KF Observation Noise Variance	R^ψ	0.08

communication and computation overhead. The yaw estimated using the Kalman filter in (5.6) is used in the attitude estimator as well. This improved the yaw regulation to $\pm 2^\circ$.

$$\hat{\psi}_{k|k-1} = \psi^{\text{IMU}}_{\text{delayed},k-1|k-1} , \tag{5.6}$$

$$P^\psi_{k|k-1} = P^\psi_{k-1|k-1} + Q^\psi_k , \tag{5.7}$$

$$\tilde{\psi}_k = \psi^{nav}_k - \hat{\psi}_{k|k-1} , \tag{5.8}$$

$$S^\psi_k = P^\psi_{k|k-1} + R^\psi_k , \tag{5.9}$$

$$K^\psi_k = \frac{P^\psi_{k|k-1}}{S^\psi_k} , \tag{5.10}$$

$$\hat{\psi}_{k|k} = \hat{\psi}_{k|k-1} + K^\psi_k \tilde{\psi}_k , \tag{5.11}$$

$$P^\psi_{k|k} = (1 - K^\psi_k) P^\psi_{k|k-1} , \tag{5.12}$$

where the noise variance for the camera is set as $R^\psi = 0.08$ and the angles integrated from the gyro is $Q^\psi = 0.1$

Yaw estimated by the Kalman filter for a hover flight is shown in Fig. 5.2. There is a slight discrepancy between the initial yaw from the camera and the yaw from the integrating the yaw angular rate. The yaw from the attitude gyro is initialized based on the take-off

position of the quadrotor, while the yaw from the camera is initialized based on the initial map generated to aid take-off.

5.6.4 Need for a Better Controller

Stability is achieved with a linear PID controller when the quadrotor is near the desired hover point. However, there are two cases in which the navigation system will find itself far from the desired location: when a disturbance causes the quadrotor to be moved far from the steady state desired location, or when a desired position is given that is far away. These situations may lead to the quadrotor displaying oscillatory or unstable behavior, in part due to the linearization performed in the control of the quadrotor.

To address the problem of disturbance, the gains might be reduced to obtain a slightly looser controller, however, this is not a completely satisfactory solution as a larger disturbance may still drive the quadrotor to instability, and performance is reduced for hover region flight. If the quadrotor is commanded to a far away position such as in the case of landing, interpolated waypoints between the current and desired location could be generated to allow the quadrotor to reach its destination, however, this again is a less than satisfactory solution.

5.6.5 Nonlinear Controller

A modified controller design is implemented using the sigmoid function, which is an "S" shaped curve with a saturation. A symmetrical function is desirable so that no switching action needs to be done based on the desired direction, thus the upper and lower asymptotes of the function will be equal and opposite. And with the output required to be zero when the input is zero, the function takes the simplified form

$$Y(t) = \frac{A}{1 + e^{-Bt}} - A/2, \tag{5.13}$$

where A sets the upper and lower asymptotes and B sets the growth rate.

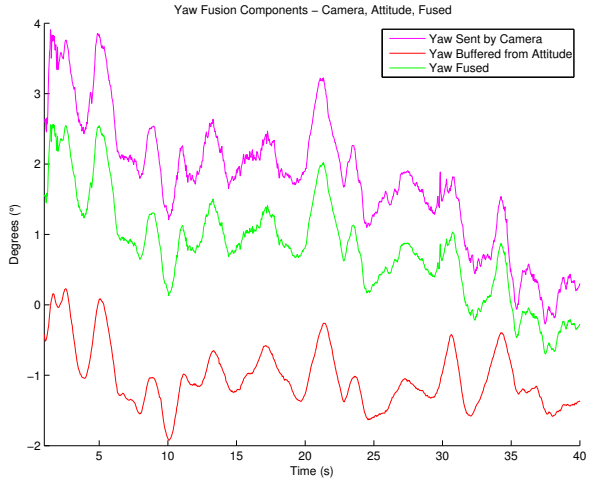


Fig. 5.2: Fusion of yaw using a Kalman filter.

Navigation for both hover and path following is regulated based on a controller using PID inputs, with the commands derived using the sigmoid function. The controller is

$$
\begin{aligned}
S^{*}_{kp} &= \frac{A^{nav,*}_{kp}}{1+e^{-B^{nav,*}_{kp}(*^{ref}_{pos}-*_{pos})}} - \frac{A^{nav,*}_{kp}}{2}, \\
u_{nav,x} &= S^{x}_{kp} + ki^{nav,x}_{sig}\int_0^t S^{x}_{kp}dt + \frac{A^{nav,x}_{kd}}{1+e^{-B^{nav,x}_{kd}(\dot{x}^{ref}_{pos}-\dot{x}_{pos})}} - \frac{A^{nav,x}_{kd}}{2}, \qquad (5.14) \\
u_{nav,y} &= S^{y}_{kp} + ki^{nav,y}_{sig}\int_0^t S^{y}_{kp}dt + \frac{A^{nav,y}_{kd}}{1+e^{-B^{nav,y}_{kd}(\dot{y}^{ref}_{pos}-\dot{y}_{pos})}} - \frac{A^{nav,y}_{kd}}{2}, \qquad (5.15)
\end{aligned}
$$

where $*$ is either x or y depending on the context, and $u_{nav,x}, u_{nav,y}$ are desired accelerations which are used to determine the desired angles according to (5.5). The gain constants used for the nonlinear navigation are indicated in Table 5.2. The final controller is shown in Fig. 5.3.

5.6.6 Noisy Measurement Benefits of Controller

An auxiliary and unexpected benefit of the nonlinear controller is its ability to limit the effects of measurement noise, which is a very important advantage when using consumer-grade components. With the linear controller, large changes in velocity in the z direction were often observed. This was mostly due to the hanging camera wire entering the measurement cone of the sonar causing a relatively smaller altitude reading leading to a huge change

Table 5.2: Nonlinear navigation controller gains.

Gain Term	Notation	Value
Navigation x Proportional Asymptote Bound	$A_{kp}^{nav,x}$	100
Navigation x Proportional Growth Rate	$B_{kp}^{nav,x}$	0.04
Navigation y Proportional Asymptote Bound	$A_{kp}^{nav,y}$	100
Navigation y Proportional Growth Rate	$B_{kp}^{nav,y}$	0.04
Navigation x Derivative Asymptote Bound	$A_{kd}^{nav,x}$	90
Navigation x Derivative Growth Rate	$B_{kd}^{nav,x}$	0.042
Navigation y Derivative Asymptote Bound	$A_{kd}^{nav,y}$	90
Navigation y Derivative Growth Rate	$B_{kd}^{nav,y}$	0.052
Navigation x Integral Gain	$ki_{sig}^{nav,x}$	0.004
Navigation y Integral Gain	$ki_{sig}^{nav,y}$	0.005

in velocity. This happened despite the use of a median filter to remove outliers in the sonar readings. This same issue was also seen in the reporting of translational velocities. As can be seen in Fig. 5.4, the desired acceleration being output by the linear navigation controller near the 130 *sec* mark is above 3000 m/s gain, which is due to a large change in velocity reported by the SLAM algorithm. This is mostly caused either due to measurement noise or finite differentiation effects because of a small dt. The nonlinear controller gracefully handles this scenario as can be seen in the lower graph. The relatively much slower dynamics of the quadrotors as compared to the frequency of measurement data makes it impossible for the quadrotor to have changed its position by a huge amount in one measurement interval of 20-33 ms. This reduction in large changes due to measurement noise provided a large improvement in the equality of the flight of the quadrotor, as well as provided greater stability since the quadrotor would be prevented from receiving a command that would cause it to reach the unstable flight region.

5.7 Vision-Based Altitude Control

Height is also reported by PTAM and and the PID controller (5.16) is used to achieve visual height control. This proves to be important for tasks such as gripping and is further

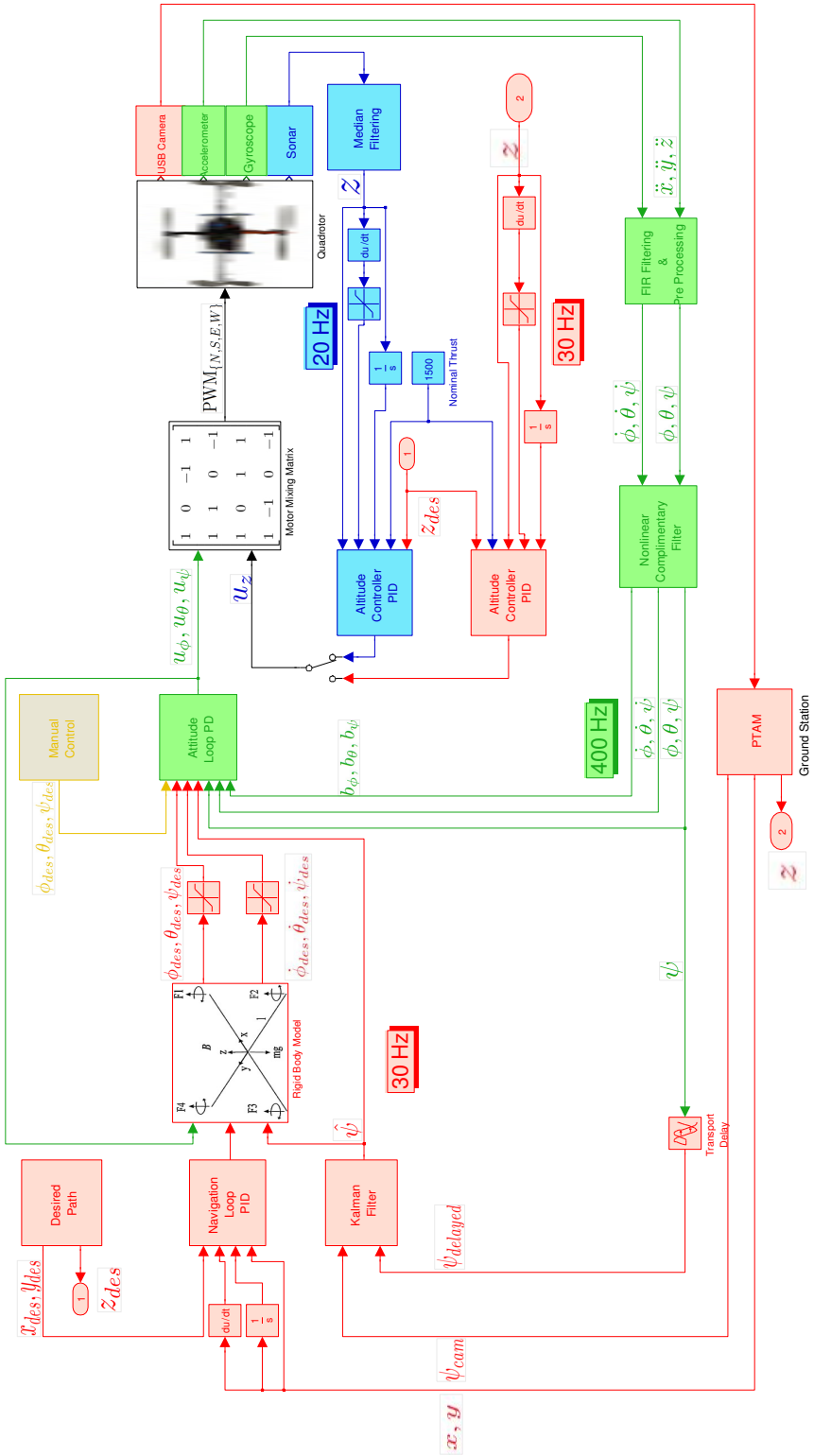


Fig. 5.3: Nested control loops for navigation.

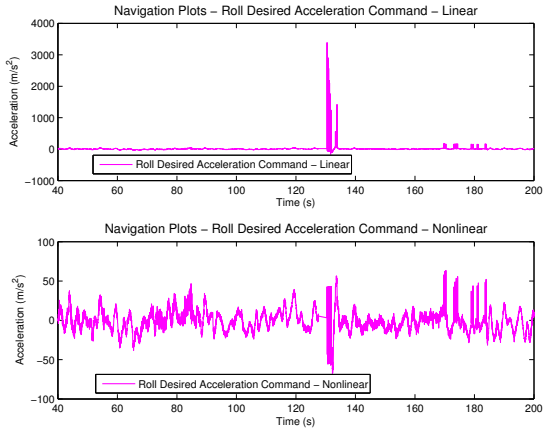


Fig. 5.4: Desired accelerations for the linear vs nonlinear controllers - x axis.

discussed in Chapter 6.

$$u_{alt,nav} = k^{nav}_{p,alt}(z^{des} - z) + k^{nav}_{i,alt}\int_0^t (z^{des} - z)dt + k^{nav}_{d,alt}(\dot{z}^{des} - \dot{z}) + u_{nom}, \tag{5.16}$$

where $u_{alt,nav}$ is used as the input to all four motors, the same way u_{alt} is used for the sonar based height controller.

For the nonlinear controller, the P and D commands are derived using the sigmoid function. The integral term is kept using the linear control, since its purpose is to counteract the decaying battery power, which can be handled using a linear approximation. The controller is then

$$\begin{aligned} u_{alt,nav} \quad = \quad & \frac{A^{alt}_{kp}}{1 + e^{-B^{alt}_{kp}(z^{des} - z)}} - \frac{A^{alt}_{kp}}{2} + k_{i,alt}\int_0^t (z^{des} - z)dt \\ & + \frac{A^{alt}_{kd}}{1 + e^{-B^{alt}_{kd}(\dot{z}^{des} - \dot{z})}} - \frac{A^{alt}_{kd}}{2}, \end{aligned} \tag{5.17}$$

where $u_{alt,nav}$ is again the input to all four motors. The gain constants used for the nonlinear altitude are indicated in Table 5.3.

5.8 Latency: Analysis and Effects

System latencies can contribute to instability and poor performance. A full analysis of

Table 5.3: Nonlinear altitude controller gains.

Gain Term	Notation	Value
Altitude Proportional Asymptote Bound	A_{kp}^{alt}	60
Altitude Proportional Growth Rate	B_{kp}^{alt}	0.2
Altitude Derivative Asymptote Bound	A_{kd}^{alt}	40
Altitude Derivative Growth Rate	B_{kd}^{alt}	0.071

the latencies in the system was done in order to minimize them. A diagram showing time taken for every step in the system is shown in Fig. 5.5. An inadvertent bug in the code for packet parsing caused 33% of the packets to be dropped leading to additional delay. While the quadrotor kept stable, the additional delays made the quadrotor unable to maintain a tight hover. This was an empirical confirmation of the fact that delays in the system must be minimized.

5.9 Results

This section gives results achieved for a hover, disturbance rejection, path tracking, and outdoor flight.

5.9.1 Vision-Based Hover

The navigation controller can operate in one of two modes: Hover or Navigation. In the hover mode, the desired velocity is set to zero, to ensure a tight hover. An example of the roll and pitch measurement results from a typical hover are shown in Figs. 5.6(a), 5.6(b), 5.6(c), 5.6(d), respectively, including the output commands in terms of desired angles per Equation (5.5).

The performance of the navigation system during a hover is indicated in Figs. 5.7(a), 5.7(b), showing the x and y measured positions versus the desired position as separate variables over time, as well as together on a coordinate plot. The position error on a hover for the x axis is ± 13 cm, and ± 11 cm for the y axis. The y axis has a tighter error due to the differences between the roll and pitch axes, as described in Chapter 4.

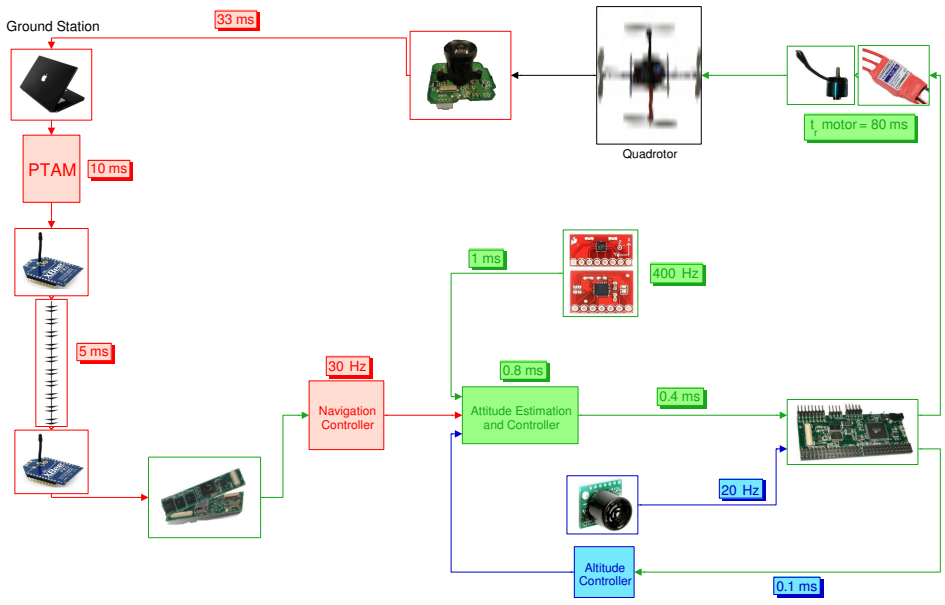
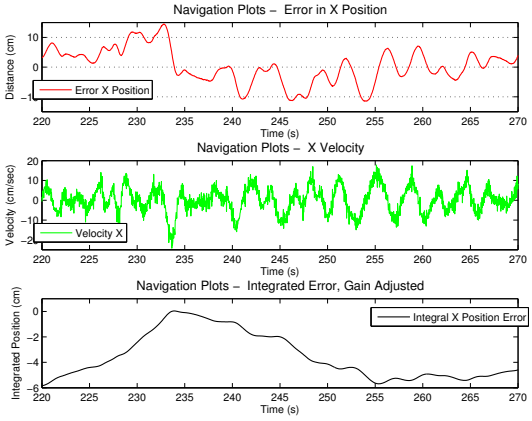


Fig. 5.5: Navigation system latency diagram by component and communication.

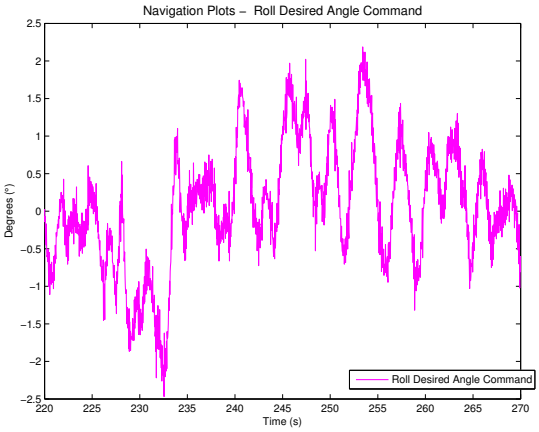
5.9.2 Disturbance Rejection While Hovering

A key requirement in any MAV is robustness. In order to test the the robustness of our system, the arms of the quadrotor were hit using a stick and the hanging USB cable pulled. This is shown in Figs. 5.8(a) and 5.8(b), respectively. The quadrotor is shown to maintain a stable hover despite severe disturbances in Fig. 5.8(c). The individual x and y positions of the quadrotor during a disturbance are shown.

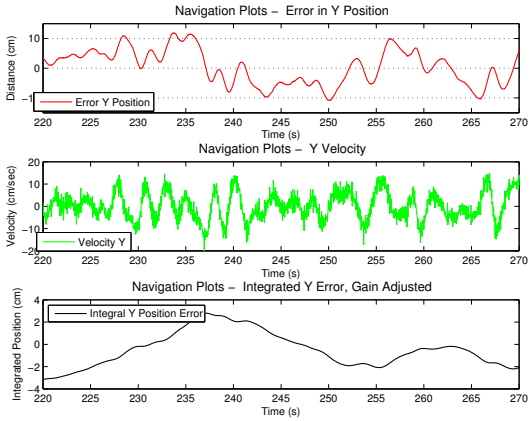
An additional test for disturbance rejection was conducted by placing a fan next to the quadrotor while it was hovering. The fan was placed about 1.5 m from the quadrotor and the stability of the quadrotor was tested at three different fan speeds corresponding to 1.7 m/s, 2.2 m/s, and 2.8 m/s measured at a distance of 1.5 m from the fan using a wind gauge. A picture of the setup is shown in Fig. 5.9(a). The quadrotor maintained a stable and tight hover at wind speeds of 1.7 m/s and 2.2 m/s, however, it was found to be jittery at 2.8 m/s, which led to the use of lower gains. The quadrotor was able to keep a hover within $\pm 20cm$ after lowering the gains as can be seen in Fig. 5.9(b). In this particular flight, the first 20 s the fan is off, the second 20 s the fan is on level 1 (1.7 m/s windspeed), the third 20 s the fan is on level 2 (2.2 m/s windspeed), the fourth 20 s the fan is on level



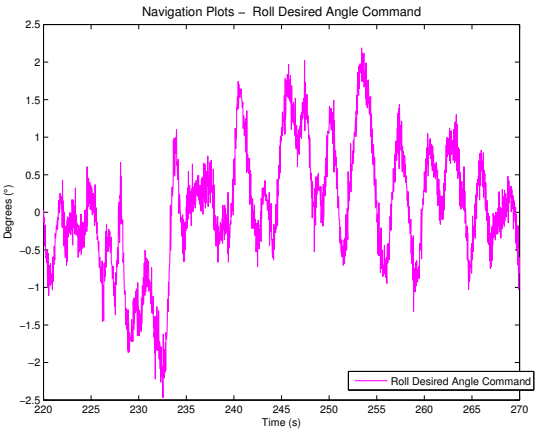
(a) Position and velocity in the roll axis.



(b) Desired angle commanded in the roll axis.



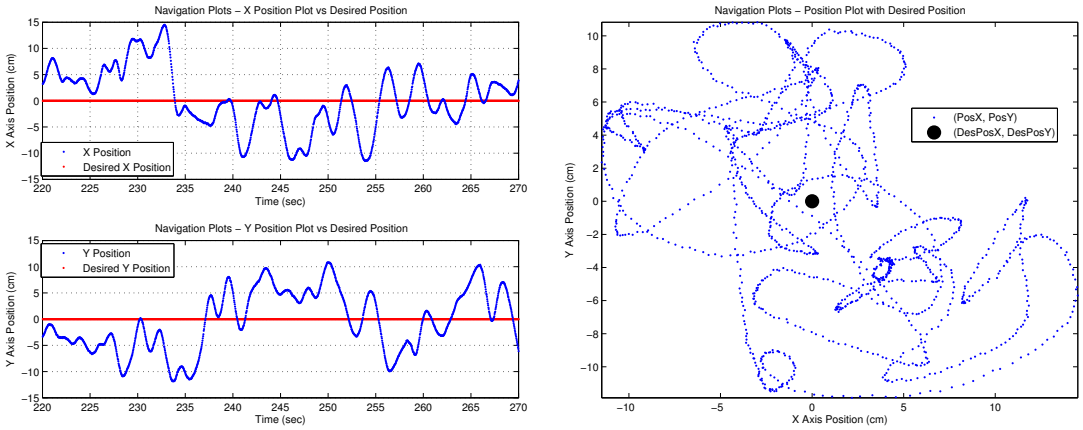
(c) Position and velocity in the pitch axis.



(d) Desired angle commanded in the pitch axis.

Fig. 5.6: Navigation measurements in the roll and pitch axes for a hover.

3 (2.8 m/s windspeed). The remainder of the flight is on level 2, except for the very end which is on level 3 again. This specific flight is with the fan blowing only in the direction of the x axis of the quadrotor. The fact that the y axis maintains near its non-disturbance hover quality demonstrates the general decoupling between the two axes of the quadrotor during flight. Detailed flight data is shown in Fig. 5.10. Another experiment had the fan blowing in the diagonal direction, and as expected, both the x and y axis behaved roughly the same. The quadrotor has also been flown outdoors in slight windy conditions; however, the wind speed was not measured.



(a) x and y separate over time. (b) Coordinate plot.

Fig. 5.7: Desired vs actual position for a hover.

5.9.3 Vision-Based Navigation and Trajectory Tracking

In the navigation mode, the quadrotor is given waypoints and a fixed desired velocity. An example of the measurement results from a square path are shown in Figs. 5.11(a), 5.11(b), 5.11(c), and 5.11(d).

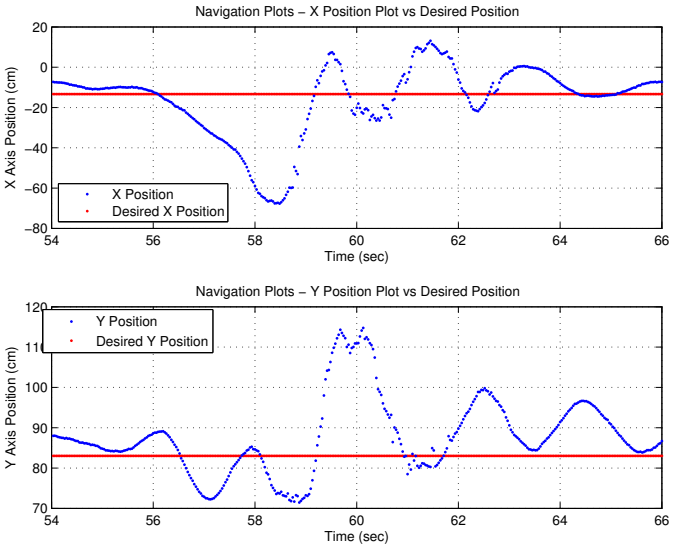
The performance of the navigation system during a square path is indicated in Figs. 5.12(a), 5.12(b), showing the x and y measured positions versus the desired position as separate variables over time, as well as together on a coordinate plot. The maximum position error on such a square path for the x axis is 37 cm, and 20 cm for the y axis. The y axis has a tighter error due to the differences between the roll and pitch axes, as described in Chapter 4.

5.9.4 Vision-Based Outdoor Flight

The quadrotor was flown outdoors to demonstrate the capabilities of the system in GPS-denied environments such as urban canyons. An action shot of the quadrotor hovering outdoors is shown in Fig. 5.13. Successful flights of the quadrotor hovering and following waypoints over grass were achieved. Grass being somewhat self-similar does not aid in global localization, therefore, the quadrotor was made to took-off on the pavement and then commanded to fly over grass. While successful flights were achieved, this experience

(a) Quadrotor being hit by a stick.

(b) Quadrotor with cable being pulled.



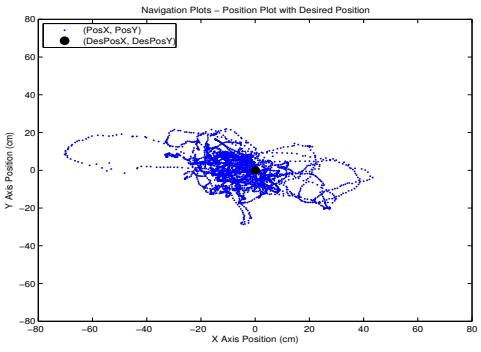
(c) Quadrotor position in x and y during disturbance and recovery.

Fig. 5.8: Disturbance rejection tests.

also exposed the various pitfalls of the system.

1. One of the outdoor tests was conducted in the middle of the day with the sun overhead. The shadow of the quadrotor was cast almost directly underneath it and the outline of the shadow provided a rich set of features compared to the relatively featureless pavement used for take-off. With a majority of the features stationary with respect to the quadrotor, the navigation system failed. This precludes the flying of the quadrotor

(a) Action shot of the quadrotor flying in front of the fan.



(b) Coordinate plot of the quadrotor position.

Fig. 5.9: Quadrotor hovering in the presence of a fan.

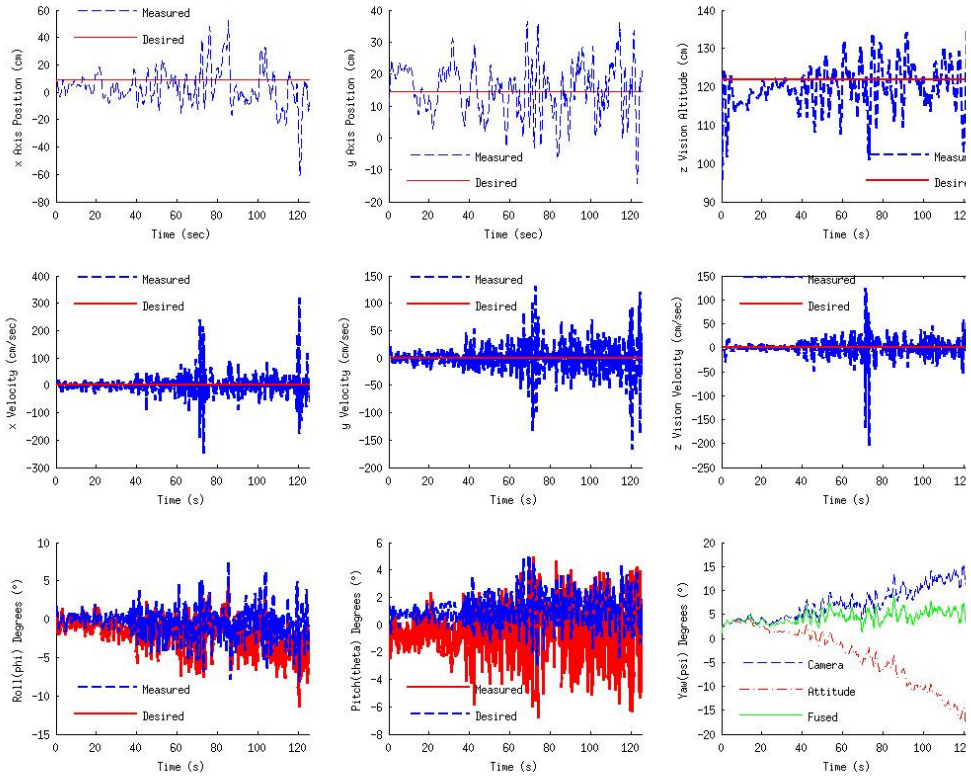
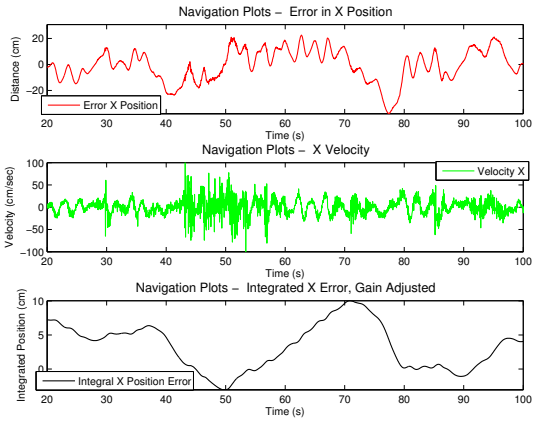


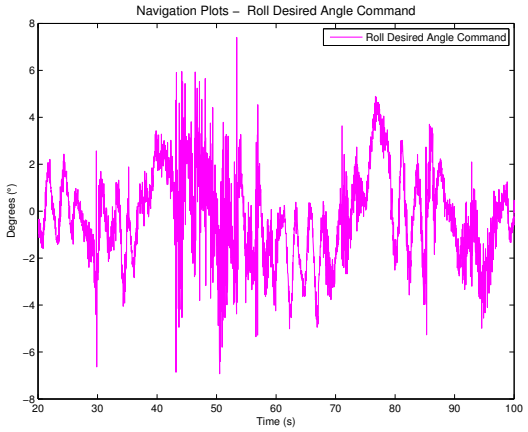
Fig. 5.10: Data for quadrotor flying in front of the fan. The top three graphs are the x, y, and z axis positions (desired and actual); the middle three graphs are the x, y, and z axis velocities (desired and actual); and the bottom three graphs are the roll (ϕ) and pitch (θ) angles (desired angle commanded and measured angles).

near the ground under a directly overhead sun. Corner features such as Harris or FAST often suffer under lighting changes and this was found to be particularly true when flying the quadrotor outdoors or even in different lighting conditions. This is an area that deserves further investigation as extracting features that enable good data-association is a fundamental requirement for any mapping algorithm. Features such as CenSure or SIFT (if computational ability allows) should be explored.

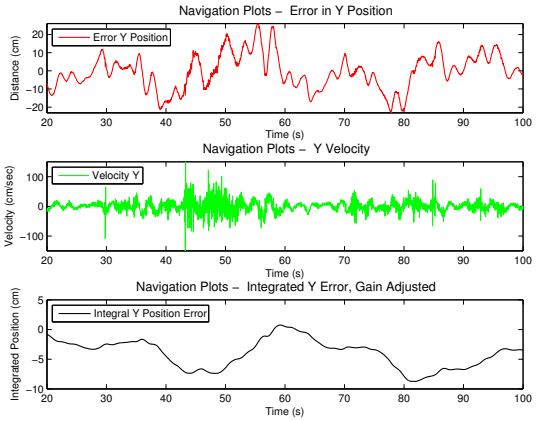
2. Another observation when operating in bright sunlight is over-exposed camera images and this problem becomes even more severe with the use of a wide-angle lens. This makes feature extraction, especially using corner detectors, extremely challenging. These problems are found to be less severe when operating with a downward looking camera, however, if a forward looking camera is to be used, care needs to be taken to



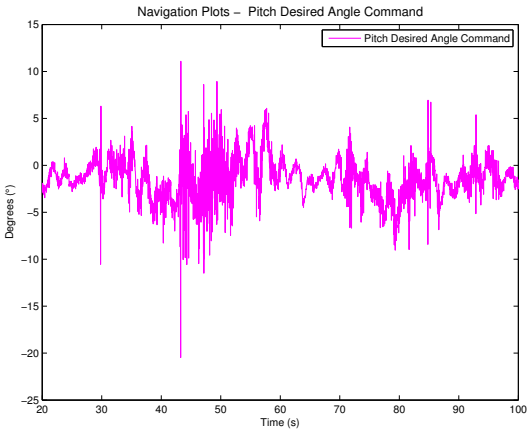
(a) Position and velocity in the roll axis.



(b) Desired angle commanded in the roll axis.



(c) Position and velocity in the pitch axis.



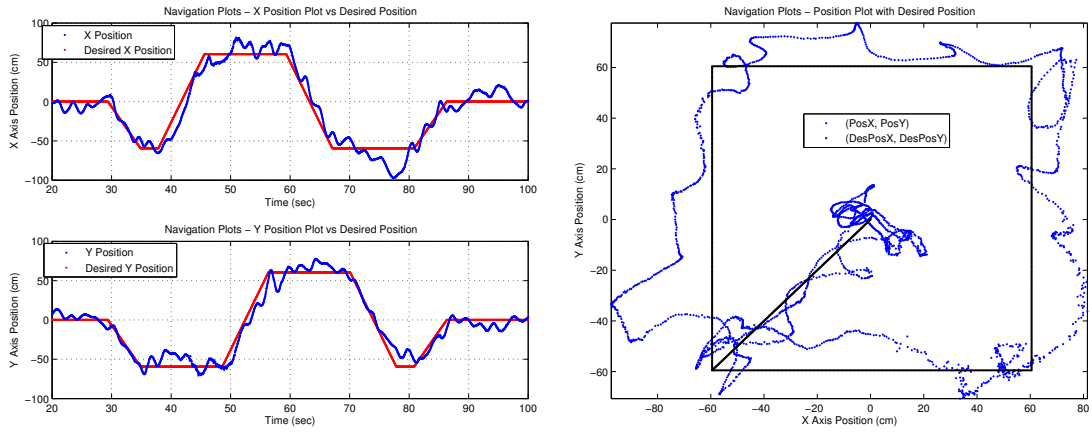
(d) Desired angle commanded in the pitch axis.

Fig. 5.11: Navigation measurements in the roll and pitch axes for path tracking.

adjust the aperture and shutter speed. This is not possible with most cheap cameras such as the webcam used in this work.

5.10 GPS as a Drop-in Replacement

GPS can be used as a drop-in replacement for the vision-based navigation system when operating in GPS-available locations. However, the ideal scenario would be to fuse the position reported by the GPS receiver with the position reported by the SLAM algorithm to obtain higher accuracy. Moreover, GPS could be used as a fallback option if the vision system fails. It could also be used as an initial estimate when performing global localization.



(a) x and y separate over time. (b) Coordinate plot.

Fig. 5.12: Desired vs actual position for path tracking.

Fig. 5.13: Quadrotor hovering outdoors.

Chapter 6

Autonomous Aerial Manipulation

This chapter discusses extending the capabilities of quadrotors from being mere observers to actively interacting with their environment and objects contained within it. Unlike fixed-wing UAVs incapable of driving their velocity to zero, quadrotors are ideally suited to the task of aerial manipulation or grasping. The quadrotor can grip objects from places not suitable for landing such as power lines, water surface, radio towers, etc. Potential applications include object retrieval, intelligence gathering, and explosives disposal.

However, three major challenges need to be overcome: precise positioning, object sensing and manipulation, and stabilization in the presence of disturbance due to object interaction. Gripping an object results in a change in the flight dynamics often leading to instability of an aerial vehicle. This is even more pronounced in the case of a nonlinear and naturally unstable system such as the quadrotor. Maintaining flight stability under these conditions is challenging and requires robust disturbance rejection. Aside from this already difficult prerequisite, the vehicle will need to be capable of precisely navigating to the object (discussed in the previous chapter) and then have some means of sensing and interacting with it. An off-the-shelf camera capable of detecting IR light sources is used to sense the objects and an underactuated passively compliant gripper capable of gripping despite positional accuracies is used.

This chapter discusses the object sensing setup, the design of the gripper, control system architecture; and concludes with experimental results demonstrating completely autonomous aerial gripping. This work in its current form can also be used for precision landing.

6.1 Related Work

Some early work using magnets for picking objects is presented by Amidi et al. [93]. Kuntz and Oh use hoops to pick up objects, however, only results using a test rig are presented [94]. Gentili et al. present some theoretical results on UAVs interacting with the environment [95]. Michael et al. present some experimental results of cooperative manipulation using cables [96]. Pounds and Dollar present some experimental results of gripping using an electric helicopter along with theoretical results proving the stability of PID control for gripping by modeling the gripper as an elastic linkage [16,97]. However, the helicopter in this implementation is under manual control, requiring an expert pilot, and the system provides no capability of gripping while hovering. Recently, this work has been extended to enable gripping while hovering [98]. Additionally, some work has been done involving gripping with quadrotors, using the Vicon Motion Capture System [99]. However, given the extremely precise information provided by the Vicon, many of the real-world issues seen in gripping still need to be tackled.

The design of the object sensing method is limited to the same low-cost constraints as the entire quadrotor system. In addition, due to the computational requirements of the SLAM algorithm and the limitation to just a single dual-core processing ground station, utilizing additional off-board processing is not feasible. This quickly reduced the number of options, based upon consumer-grade sensors that can be processed onboard without limiting the speed of the attitude controller.

After evaluating a number of sensors for object detection, a decision was made to use a camera extracted from a Nintendo WiiMote owing to its low-cost, light-weight, and low-power consumption [31]. These are critical requirements for any sensing modality to be used on an aerial vehicle. The camera consists of a 1024 x 768 pixels Charged Coupled Device (CCD) sensor and a custom system-on-a-chip that is capable of tracking up to four IR light sources simultaneously. It reports the x and y position of the IR light sources or blobs along with the estimated blob size as a value ranging from one to six at a rate of up to 200 Hz over an I^2C line. It can detect IR blobs up to a distance of 5 m and has a FOV

of 41° vertical and 3° horizontal. Parameters such as the minimum and maximum blob size and camera gain can be set over I^2C. The gain parameter is related to the sensitivity of the camera and a gain of 255 was experimentally found to work best.

6.2 Object Sensor

6.2.1 Interfacing

There are two ways to interface to these cameras. Lee demonstrated connecting to the camera over a Bluetooth link, however, this setup used the microcontroller and Bluetooth chipset on the Wii Board, necessitating the use of the entire Wii board [100]. This lead to the concern of accommodating a relatively large board on the quadrotor. Moreover, there were concerns of latency over a Bluetooth link and it was finally decided to interface with the camera directly.

Kako has documented the interfacing of the camera over I^2C [101]. Using schematics obtained from his site, a board housing the camera and supporting components was built and is shown in Fig. 6.1. It only requires an enable and a clock signal besides the I^2C lines and operates at 3.3 V.

6.2.2 Challenges

The camera is attached to the I^2C bus enabling the gumstix to directly talk to it.

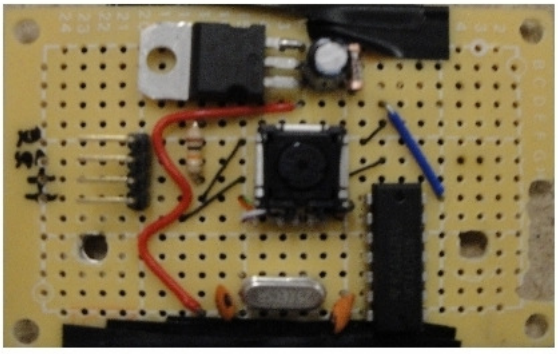


Fig. 6.1: Wii Camera along with supporting circuitry mounted on a board.

One of the key challenges in interfacing the camera was the flouting of the I^2C protocol by the camera. A typical read using the I^2C protocol involves the sending of a start bit, followed by the device address, register address, and then a repeated start to be able to read the contents of the register. Surprisingly, the camera expected a stop bit instead of a repeated start contravening the I^2C protocol. This anomaly was found after observing the waveforms of the I^2C lines using Kako's program on an Arduino Board [102]. It was resolved by splitting the read cycle into a separate read and write phase.

6.3 Gripper

This section discusses the design and integration of the gripper with the quadrotor. The design is highly dependent upon the structure of the platform and the space constraints. The gripper is shown in a closed position, separate from the vehicle, in Fig. 6.2.

6.3.1 Requirements

Some of the key application specific requirements for the gripper are enumerated below.

1. Compliance - Due the limited positional accuracy achievable using quadrotors, a gripper is needed that is capable of manipulation under uncertainty.

2. Ability to flatten itself - This unique requirement is due to the small landing gear on the current system, requiring a gripper able to be accommodated under the landing gear for takeoff and landing. Additionally, the gripper needs to avoid occluding the IR sensor, necessitating the ability to flatten out of view for object identification and tracking.

3. Light-Weight - The limited payload capacity and already high energy costs drive this requirement.

4. Minimal actuation - For similar reasons, an underactuated system reduces the need for more motors, alleviating power and weight constraints.

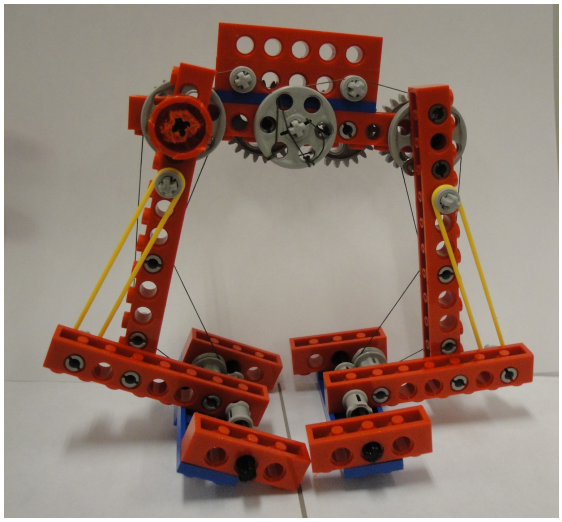


Fig. 6.2: Gripper showing the cables, pulleys, and elastic bands used to provide compliance and under-actuation.

5. Tall gripper - The height of the gripper signifies the vertical distance from the object to the quadrotor base. The specific need for a tall gripper stems from the camera dead-zone, where at a distance of 5 cm it is incapable of detecting IR blobs and even just above this distance the FOV contains a very small area. This additionally necessitates mounting of the camera as high as possible.

6.3.2 Design and Integration

A decision to design a custom gripper was made owing to the unavailability of grippers that satisfied the criteria listed above, especially, the ability to flatten. Conventional robot grippers typically require high precision, , which is not feasible for a hovering quadrotor. The approach outlined in this thesis employs an underactuated gripper with passive mechanical compliance leading to insensitivity to positional uncertainty. This provides the benefits of minimal actuation and no gripper-based sensing requirements, unlike other complicated control schemes as force control, which provide active compliance.

The design is inspired from the gripper presented by Dollar and Howe [103]. The gripper is shown in Fig. 6.2 and it is mounted vertically under the quadrotor, as shown in Fig. 6.3. It is constructed using Lego for rapid prototyping and uses a combination of

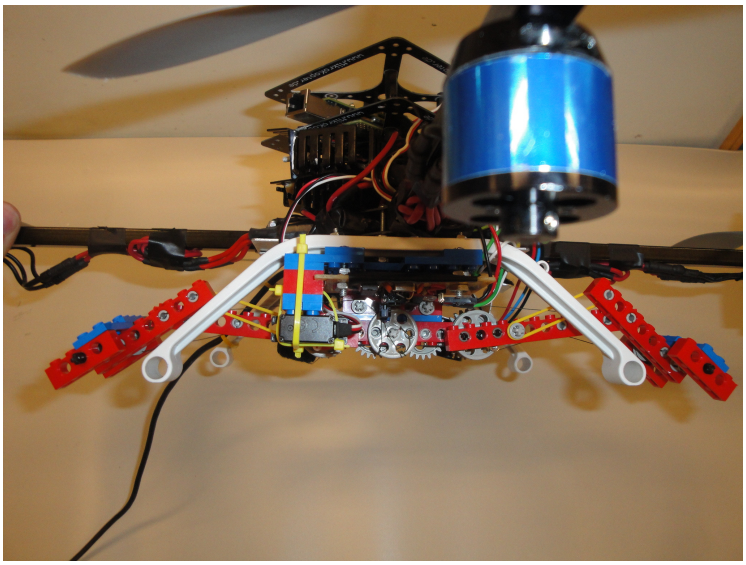


Fig. 6.3: Gripper shown mounted under the quadrotor.

pulleys and elastic-bands to achieve under-actuation and compliance [104]. It is capable of grasping objects up to 7.5 cm wide.

6.3.3 Actuation

Actuation of the gripper is achieved using a micro servo from Futaba, weighing just 8 g. It accepts PWM signals varying in width from 1 to 2 ms at a rate of 50 Hz, with the indicated pulse widths signifying the positions for the servo. The torque provided by the motor is found to be sufficient to actuate the gripper and maintain a closed position when grasping objects.

6.4 Implementation Details

The IR camera is found to be most sensitive to the 940 nm wavelength, therefore an IR LED with a peak emittance at 940 nm is used as a marker. The LED is extremely small measuring 3mm x 3mm x 4mm with a viewing angle of 160°. It can be unobtrusively placed along with the object to be gripped and can be powered using a small button cell.

The sonar altitude sensor cannot be used during gripping maneuvers, since it will see a shorter distance when it is over the object. Instead, altitude is controlled based off of the

navigation system, as discussed in Chapter 5.

6.5 Control System Architecture

A third outer control loop is added that can be viewed as a guidance loop, shown in Fig. 6.4, which changes the desired position of the quadrotor based upon the measured position of the IR blob. A feedback loop is required due to the quantized and noisy blob location data returned by the camera and the lack of true relative positioning information. Initially a PD loop was tested for changing the desired position, but gave poor results due to rapid changes in the desired position. Since the translational dynamics of the quadrotor are relatively slow, it was surmised that an integrator alone would be sufficient for convergence to the desired position in x and y and experiments proved this hypothesis to be true. The outputs of the integrator loop

$$x_{\text{offset}} = k^x_{i,IR} \cdot \int_0^t (x^{des}_{IR} - x_{IR}), \tag{6.1}$$

$$y_{\text{offset}} = k^y_{i,IR} \cdot \int_0^t (y^{des}_{IR} - y_{IR}), \tag{6.2}$$

are the offsets for the desired positions of the navigation loop, with $k^x_{i,IR}, k^y_{i,IR}$ being the integrator gains.

This augments a PID loop for positional control and a PD loop for attitude stabilization and is found to be robust enough to deal with disturbances caused by gripping. These disturbances are mainly caused due to the contact forces of the object acting on the quadrotor and the aerodynamic effects.

6.6 Experimental Observations

For the results presented here, a stuffed toy weighing 150 g was placed about 50 cm below the commanded height of the quadrotor. The entire sequence of actions from the quadrotor first sighting the object, decreasing altitude to grip the object and then returning to a hover, took less than 4 seconds. This quick response behavior prevents the translational dynamics of the quadrotor from being impacted by the aerodynamic effects, leading to a

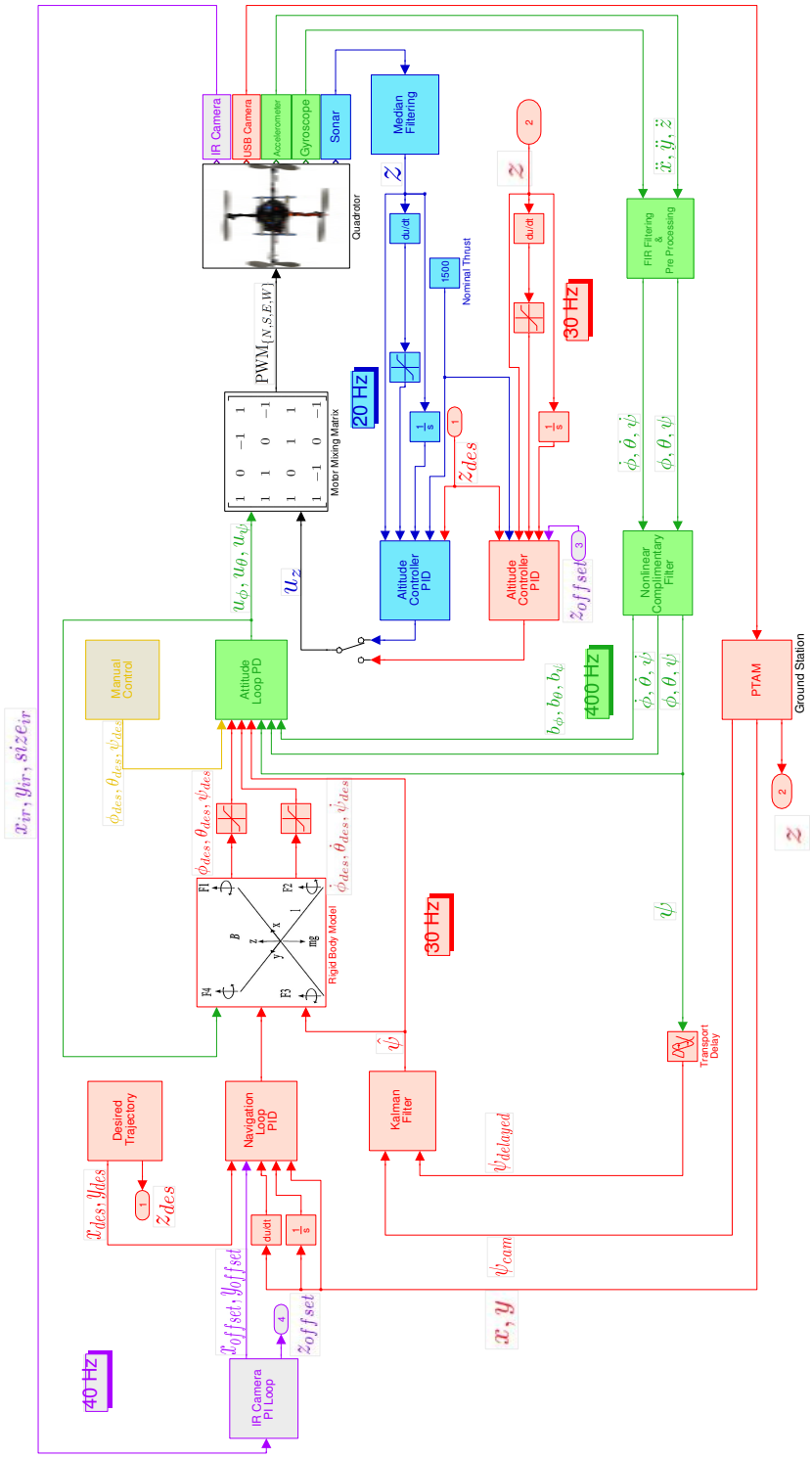


Fig. 6.4: Cascaded control scheme used for gripping.

successful grip. Figure 6.5 shows the path taken to the object. Takeoff occurs in the upper left quadrant of the graph, with gripping occurring in the lower right. Way-points were given incrementally to the quadrotor until gripping was activated over the object. Figure 6.6 shows the position, velocity, and angles for the entire flight. Fig. 6.7 shows the desired offsets in x, y, and z, and Fig. 6.8 shows the change in desired position along with the actual position of the quadrotor on the map. Figure 6.9 shows the x and y pixel positions of the IR blob (as seen by the IR camera), and Fig. 6.10 shows these same positions as a 3D plot with time on the vertical axis, and the detected blob size shown below. An action shot sequence of the gripping maneuver is shown in Fig. 6.11.

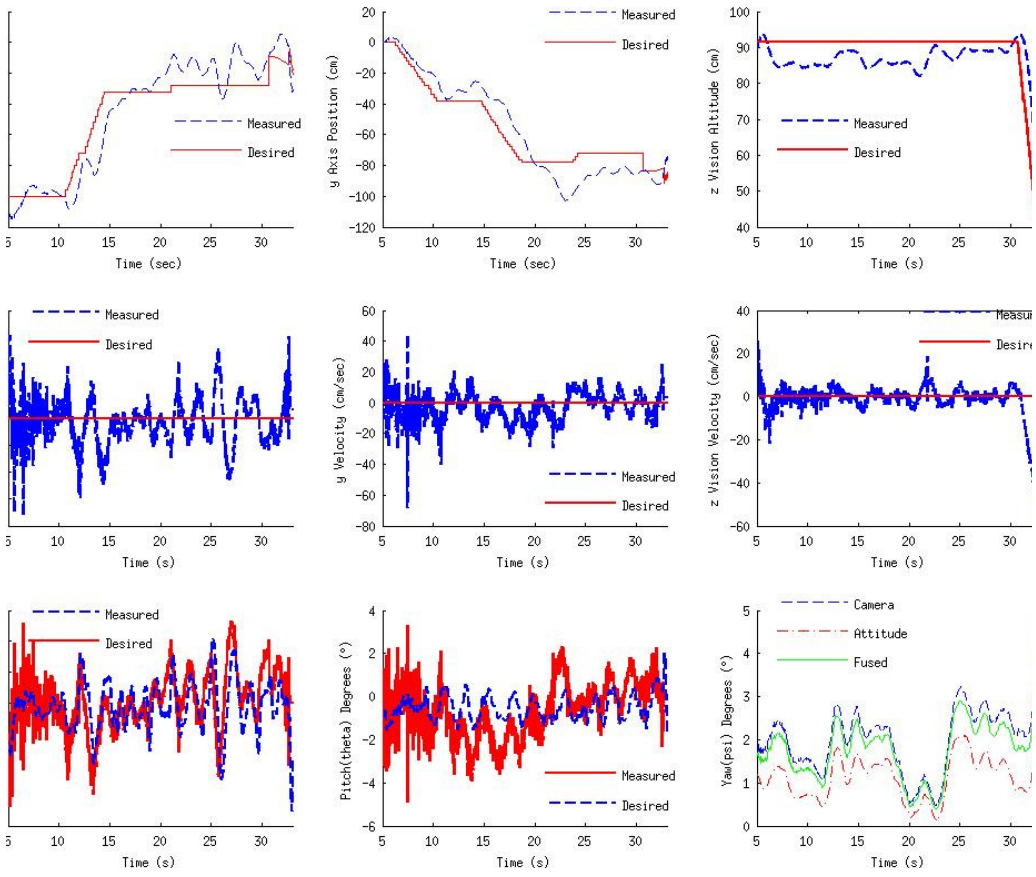


Fig. 6.5: The top three graphs are the x, y, and z axis positions (measured and reference); the middle three graphs are the x, y, and z axis velocities (measured and reference); and the bottom three graphs are the roll (ϕ) and pitch (θ) angles (desired angle generated by the navigation system, and the measured angle from the attitude estimation system), and the yaw (ψ) angle (measured value from the camera, gyroscope, and the fused value).

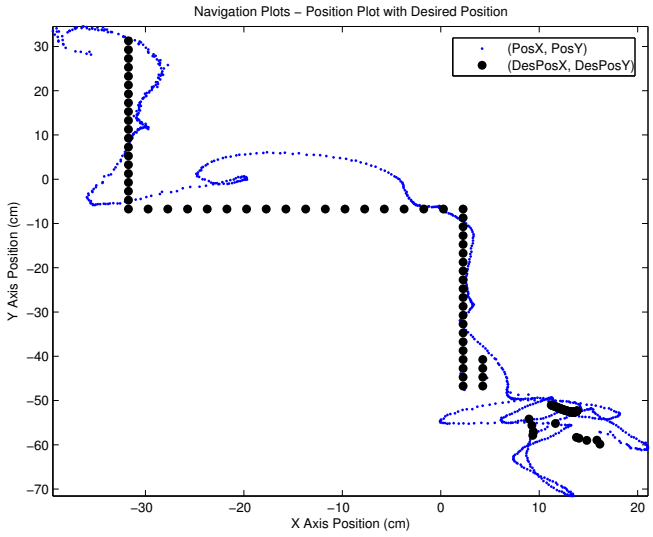


Fig. 6.6: Coordinate plot showing the path to the desired gripping location.

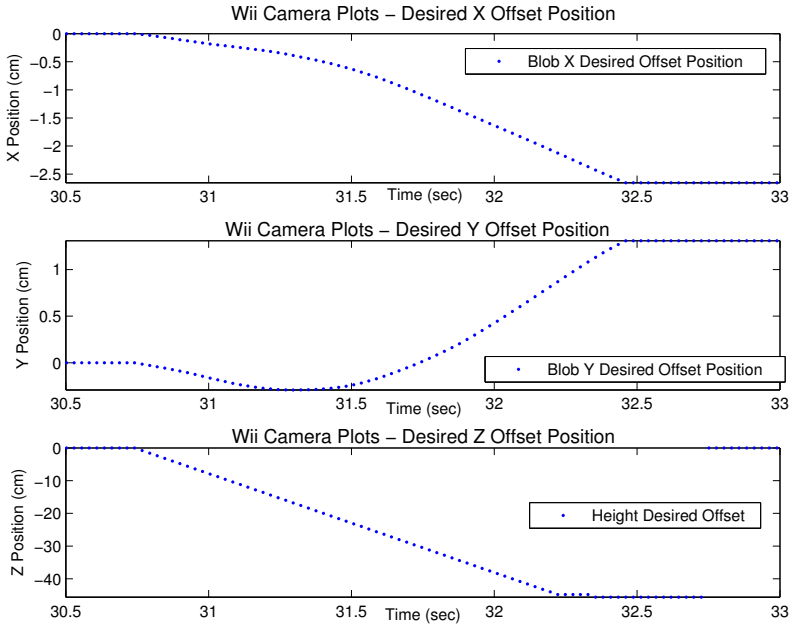


Fig. 6.7: These plots show desired offsets in x, y, and z (in cm) from the hover position to the blob and are commanded by the outer-most loop to the navigation loop using an integral controller (6.1, 6.2). The blob is detected by the IR camera at 30.7 sec, whereupon the offsets begin generating commands to the navigation controller for maneuvering over the blob, horizontally as well as vertically. Gripping is activated at 32.4 sec, after which the x and y offsets remain unchanged because the outermost controller is deactivated, while the z offset is reset to 0 in order to return to the initial altitude.

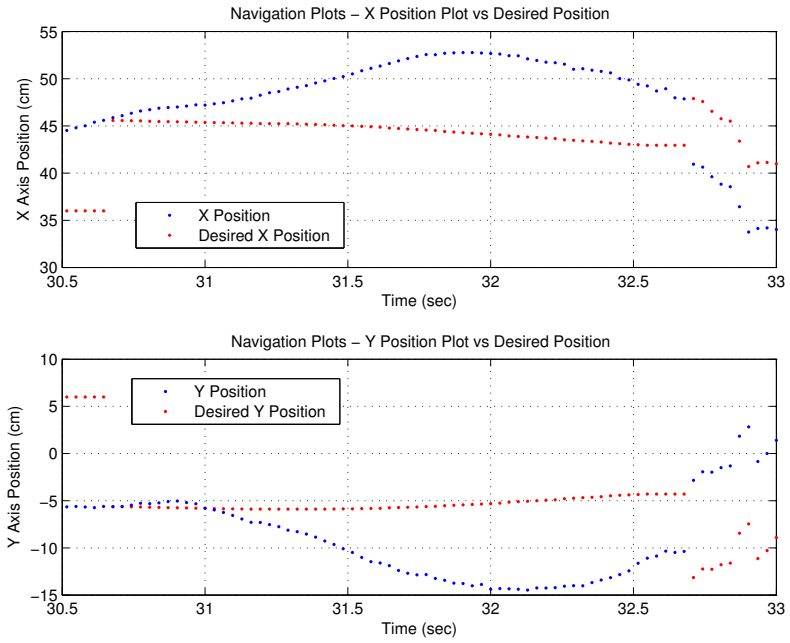


Fig. 6.8: Desired and actual positions of the quadrotor as seen by the navigation controller, in x and y. The blob is seen by the IR camera at 30.7 sec, at which point the desired positions (in red) are adjusted by the desired offsets, and gripping is activated at about 32.4 sec. After gripping, the navigation controller returns to a hover mode.

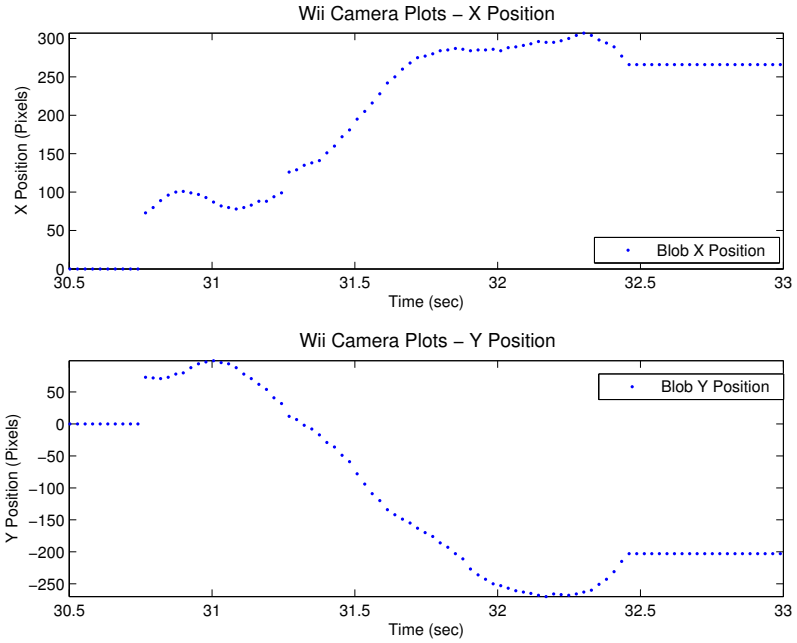


Fig. 6.9: The x and y pixel positions of the IR blob while maneuvering to center over the blob and gripping.

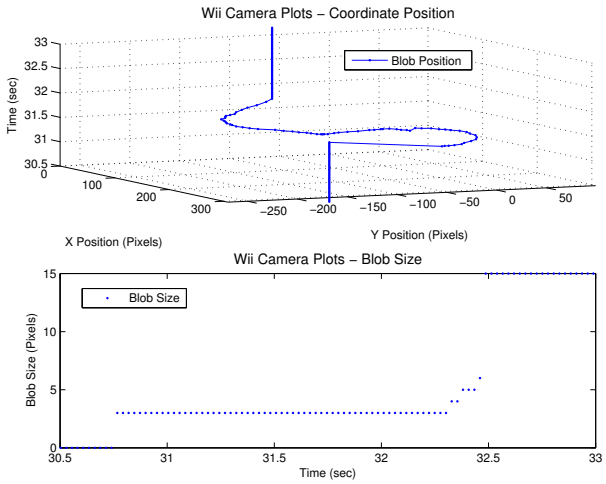
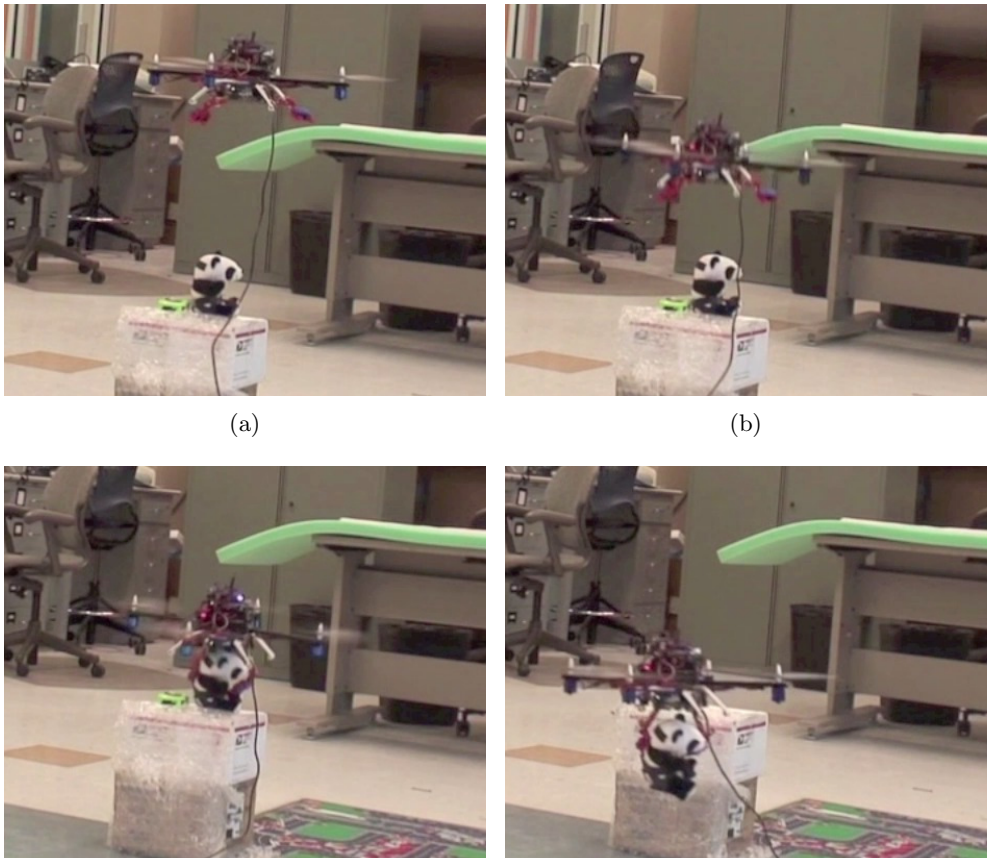


Fig. 6.10: Blob pixel position vs time (top) and blob size (bottom) as seen by the IR camera. The blob is seen by the IR camera at 30.7 sec, and once a blob size of 5 is detected within a restricted pixel area, gripping is activated at 32.4 sec, shortly after which the outermost controller is deactivated and so the blob size goes to 15.



(a) (b)

(c) (d)

Fig. 6.11: Action shots of the quadrotor gripping a stuffed toy.

There are several parameters that were varied to assess the effect on the quality of the grip, including: object height from ground, distance from quadrotor nominal height to object height, placement/attachment quality of the LED, and object shape and material.

1. Object height from ground - The farther the object is from the ground, the less that ground effect turbulence prevents the quadrotor from steady flight during the grip. As a corollary to this, the type of platform on which the object is placed will also affect the amount of air flow affects on the quadrotor.

2. Distance from quadrotor nominal height to object height - The farther the quadrotor has to travel in order to get to the object, the more difficult it is to stay directly over the object during the descent, and thus the more the quadrotor has to recover to the desired position. This problem can mostly be attributed to the overhanging cable.

3. Placement/attachment quality of the LED - The brightness and location of the LED with respect to the quadrotor affects the field of view of the gripping IR camera, as well as the value of the detected blob size. If the LED does not point straight up to the camera, then the locations at which the camera can detect the LED become restricted, and the blob size value will not be accurate.

4. Object shape and material - The better the object can fit in the gripper from different orientations, and the better the gripper can grasp the object due to the material, the more likely a successful grip will take place. Strangely shaped objects and those of slippery material make gripping much more difficult with the current gripper.

The success in gripping lies on effectively managing the possible issues described above. Gripping is kept simple by means of just an extra sensor and a upper feedback controller over the navigation system in order to reduce the error seen by the camera without requiring camera calibration for absolute measurement information.

Chapter 7

Conclusion and Future Work

7.1 Summary

This paper presents the development of a complete quadrotor helicopter capable of autonomous navigation in GPS-denied environments using inexpensive sensors. It is capable of a precise hover and has strong disturbance rejection capabilities. The quadrotor is shown to gracefully transition from outdoor to indoor environments and vice-versa, extending the current state-of-the-art. A novel nonlinear controller using model-based control enables robust path tracking performance. Moreover, the quadrotor is shown to autonomously navigate to the location of an object without needing any prior information of the environment, sense the object and then ultimately grip it.

7.2 Future Work

This section presents the possibles avenues for future research. The proposed future work is categorized by chapter.

7.2.1 Hardware and Software Architecture

A shift to Surface Mount Devices (SMD) would enable the circuitry to be made more compact and lighter. It would lower the cost of the system and would allow for more payload to be carried. The use of a processor with a FPU would allow more processes to be run on the onboard computer.

7.2.2 Attitude Estimation and Altitude Control

The quaternion form of the nonlinear complementary filter could be implemented to avoid singularities. Quaternion based nonlinear PID Control has been shown to be exponen-

tially stable and could be implemented for even better attitude stabilization performance. Accurate system identification would aid in the design of better controllers.

7.2.3 Vision-Based Hover and Navigation

Currently, the range of the quadrotor is limited by the length of the USB cable. This issue can be effectively addressed by implementing the visual SLAM algorithm on an additional computer onboard the quadrotor. Outdoor navigation could be improved through the fusion of GPS, vision, and IMU data and additionally, GPS could provide initial estimates for global localization. SLAM algorithms with the goal of providing life-long mapping could be implemented to enable large-scale mapping. The currently used visual SLAM algorithm assumes a mostly static environment. Extending visual SLAM algorithms to dynamic environment is an important area for future research. The autonomy of the quadrotor could be further enhanced by implementing online path-planning, especially, taking into account the dynamics of the quadrotor.

7.2.4 Autonomous Aerial Manipulation

One of the key limitations of this work is the need for an IR light source marker to be placed with the object to be gripped. Even though the IR LED can be powered using just a button cell and is itself small and unobtrusive, our system precludes the ability to grip objects in adverse environments by not specifically addressing the perception problem. A depth camera could be used to provide a dense 3D point cloud for identifying a variety of objects. Maps generated using PTAM are not dense enough to enable object recognition. Additionally, the object is assumed to be stationary for gripping from a hover position. Devising new controllers to account for moving objects would further enhance the capabilities of the system.

Currently, the descent phase for quadrotor and the activation of the gripper itself is based on the blob size as perceived by the camera. The values reported for blob size vary from 1 to 6 giving a highly quantized estimate of the blob size. This could also be address by perceiving the object using a depth camera.

The payload capacity of the quadrotor is 800 g, but is currently limited by the gripper design. A custom manufactured gripper would enable carrying heavier loads and utilizing the full payload capacity of the quadrotor.

A taller landing gear would enable the quadrotor to land with the object in its grasp. Currently, the object needs to be released and then the quadrotor lands.

References

[1] J. Leishman, "The Breguet-Richet Quad-Rotor Helicopter of 1907," [http://www.enae.umd.edu/AGRC/Aero/Breguet.pdf], 2002.

[2] P. M. Lambermont, *Helicopters and Autogyros of the world*. New York: Philosophical Library, 1958.

[3] Northrop Grumman Corporation, "RQ-4 Global Hawk," [http://www.northropgrumman.com/], 2011.

[4] iRobot Corporation, "510 Packbot and 710 Warrior," [http://www.irobot.com/], 2011.

[5] Honeywell International, "T-Hawk," [http://www.thawkmav.com/], 2011.

[6] Ascending Technologies, "AscTec Hummingbird," [http://www.asctec.de], 2011.

[7] Microdrones GmbH, [http://www.microdrones.com], 2011.

[8] MikroKopter, [http://www.mikrokopter.com], 2011.

[9] Quanser, "Qball-X4," [http://www.quanser.com], 2011.

[10] Parrot SA., "AR.Drone," [http://ardrone.parrot.com], 2011.

[11] Vicon MX Systems, "Vicon," [http://www.vicon.com/products/viconmx.html], 2011.

[12] Draganfly Innovations, "Draganflyer V Ti Pro," [http://www.rctoys.com], 2010.

[13] FlightGear Flight Simulator, [http://www.flightgear.org], 2011.

[14] P. Bristeau, P. Martin, E. Salaün, and N. Petit, "Role of propeller aerodynamics in model of quadrotor UAV," in *Proceedings of the IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 683–688, Aug. 2009.

[15] P. Pounds, R. Mahony, and J. Gresham, "Towards dynamically favourable quad-rotor aerial robots," in *Proceedings of the ARAA Australasian Conference on Robotics and Automation (ACRA)*, 2004.

[16] P. Pounds and A. Dollar, "Hovering stability of helicopters with elastic constraints," in *Proceedings of the ASME Dynamic Systems and Control Conference (DSCC)*, 2010.

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

[18] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 562–571, May 2006.

[19] Align Corp., Ltd., "T-Rex 600 Landing Gear," [http://www.align.com.tw], 2011.

[20] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quad-rotor robot," in *Proceedings of the ARAA Australasian Conference on Robotics and Automation (ACRA)*, 2006.

[21] RCgroups, "Brushless Motor Comparison Table," [http://www.rcgroups.com], 2011.

[22] Landing Products, "10x47 Slow Flyer Propeller," [http://www.apcprop.com], 2011.

[23] National Instruments Corporation, "Labview," [http://www.ni.com], 2011.

[24] D. Gurdan, J. Stumpf, M. Achtelik, K. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 361–366, Apr. 2007.

[25] Atmel Corporation, "ATmega128 Microcontroller," [http://www.atmel.com], 2011.

[26] Gumstix, Inc., "Robostix-TH," [http://www.gumstix.com], 2011.

[27] InvenSense, Inc., "ITG-3200 3-Axis Gyroscope," [http://www.invensense.com], 2011.

[28] Bosch Sensortec GmbH, "BMA180 3-Axis Accelerometer," [http://www.bosch-sensortec.com], 2011.

[29] Maxbotix, Inc., "LV-MaxSonar-EZ2," [http://www.maxbotix.com], 2011.

[30] Logitech, "QuickCam Pro 5000," [http://www.logitech.com], 2011.

[31] Nintendo of America, Inc., "Wii Remote," [http://www.nintendo.com], 2011.

[32] Digi International, Inc., "Xbee-PRO 802.15.4 Module," [http://www.digi.com], 2011.

[33] Advance Energy, Inc., "TP2600-3SPL Battery," [http://www.thunderpowerrc.com], 2011.

[34] J.-Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1148–1162, Oct. 1991.

[35] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2451–2456, Sept. 2004.

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

[37] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Towards autonomous indoor micro VTOL," *Autonomous Robots*, vol. 18, pp. 171–183, 2005.

[38] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3255–3260, Oct. 2006.

[39] G. Hoffmann, D. Rajnarayan, S. Waslander, D. Dostal, J. Jang, and C. Tomlin, "The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *Proceedings of the AIAA/IEEE Conference on Digital Avionics Systems Conference (DASC)*, Oct. 2004.

[40] N. Guenard, T. Hamel, and V. Moreau, "Dynamic modeling and intuitive control strategy for an "x4-flyer"," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 141–146, June 2005.

[41] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, July 2004.

[42] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control Systems Magazine*, vol. 25, no. 6, pp. 45–55, Dec. 2005.

[43] M. Valenti, B. Bethke, G. Fiore, and J. P. How, "Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference (GNC)*, Aug. 2006.

[44] S. Waslander, G. Hoffmann, J. S. Jang, and C. Tomlin, "Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3712–3717, Aug. 2005.

[45] Sparkfun Electronics, [http://www.sparkfun.com], 2011.

[46] InvenSense, Inc., "IDG-500 2-Axis Gyroscope," [http://www.invensense.com], 2011.

[47] Analog Devices, "ADXL335 3-Axis Accelerometer," [http://www.analog.com], 2011.

[48] C. Dobler, "Development of flight hardware for a next generation autonomous micro air vehicle," Master's thesis, Swiss Federal Institute of Technology - Zurich, Zurich, Switzerland, 2010.

[49] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[50] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, June 2008.

[51] K. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[52] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference (GNC)*, Aug. 2007.

[53] E. Altug, J. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 72–77, 2002.

[54] E. Altug, J. Ostrowski, and C. Taylor, "Quadrotor control using dual camera visual feedback," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 4294–4299, Sept. 2003.

[55] E. Altuğ, J. P. Ostrowski, and C. J. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," *International Journal of Robotics Research*, vol. 24, pp. 329–341, May 2005.

[56] G. P. Tournier, M. Valenti, and J. P. How, "Estimation and control of a quadrotor vehicle using monocular vision and moiré patterns," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference (GNC)*, pp. 2247–2252, Aug. 2006.

[57] H. Romero, R. Benosman, and R. Lozano, "Stabilization and location of a four rotor helicopter applying vision," in *Proceedings of the AACC American Control Conference (ACC)*, June 2006.

[58] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, Apr. 2008.

[59] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743–749, June 2009.

[60] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter trajectory tracking control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference (GNC)*, Aug. 2008.

[61] P. Rudol, M. Wzorek, and P. Doherty, "Vision-based pose estimation for autonomous indoor navigation of micro-scale unmanned aircraft systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1913–1920, May 2010.

[62] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," in *Proceedings of the Computer Vision Winter Workshop*, Aug. 2007.

[63] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Computing Surveys*, vol. 27, pp. 433–466, Sept. 1995.

[64] S. Fowers, D. Lee, B. Tippetts, K. Lillywhite, A. Dennis, and J. Archibald, "Vision aided stabilization and the development of a quad-rotor micro UAV," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pp. 143–148, June 2007.

[65] H. Romero, S. Salazar, and R. Lozano, "Real-time stabilization of an eight-rotor UAV using optical flow," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 809–817, Aug. 2009.

[66] Intel Corporation, "OpenCV Computer Vision Library," [http://www.intel.com], 2011.

[67] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," *Technical Report, Intel Corporation*, 1999.

[68] J. Conroy, G. Gremillion, B. Ranganathan, and J. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," *Autonomous Robots*, vol. 27, pp. 189–198, 2009.

[69] F. Kendoul, K. Nonami, I. Fantoni, and R. Lozano, "An adaptive vision-based autopilot for mini flying machines guidance, navigation and control," *Autonomous Robots*, vol. 27, pp. 165–188, 2009.

[70] L. R. Garcia-Carrillo, E. Rondon, A. Dzul, A. Sanche, and R. Lozano, "Hovering quad-rotor control: A comparison of nonlinear controllers using visual feedback," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pp. 1662–1667, Dec. 2010.

[71] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2878–2883, May 2009.

[72] D. Hahnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 496–501, 2002.

[73] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1322–1328, 1999.

[74] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unstructured and unknown indoor environments," in *Proceedings of the European Conference on Micro Air Vehicles (EMAV)*, 2009.

[75] R. He, S. Prentice, and N. Roy, "Planning in information space for a quadrotor helicopter in a GPS-denied environment," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1814–1820, May 2008.

[76] E. B. Olson, *Robust and efficient robotic mapping*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[77] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[78] B. Steder, G. Grisetti, C. Stachniss, and W. Burgard, "Visual SLAM for flying vehicles," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1088–1093, Oct. 2008.

[79] S. Ahrens, D. Levine, G. Andrews, and J. How, "Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2643–2648, May 2009.

[80] K. Celik, S.-J. Chung, M. Clausman, and A. Somani, "Monocular vision SLAM for indoor aerial vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1566–1573, Oct. 2009.

[81] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 21–28, May 2010.

[82] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," in *Proceedings of the International Society for Optical Engineering (SPIE)*, Apr. 2009.

[83] S. Soundararaj, A. Sujeeth, and A. Saxena, "Autonomous indoor helicopter flight using a single onboard camera," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5307–5314, Oct. 2009.

[84] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[85] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[86] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the International Conference on Artificial Intelligence (IJCAI)*, pp. 1151–1156, 2003.

[87] H. Strasdat, J. Montiel, and A. Davison, "Real-time monocular SLAM: Why filter?" in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2657–2664, May 2010.

[88] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1–10, 2007.

[89] B. Triggs, P. F. McLauchlan, R. Hartley, and A. W. Fitzgibbon, "Bundle adjustment–a modern synthesis," *Vision algorithms theory and practice*, vol. 34099, pp. 298–372, 2000.

[90] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 430–443, 2006.

[91] M. I. A. Lourakis and A. A. Argyros, "SBA: a software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, vol. 36, pp. 2:1–2:30, Mar. 2009.

[92] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 802–815, Oct. 2008.

[93] O. Amidi, T. Kanade, and J. R. Miller, "Vision-based autonomous helicopter research at carnegie mellon robotics institute 1991-1997," in *American Helicopter Society International Conference*, Apr. 1998.

[94] N. R. Kuntz and P. Y. Oh, "Development of autonomous cargo transport for an unmanned aerial vehicle using visual servoing," in *Proceedings of the ASME Dynamic Systems and Control Conference (DSCC)*, 2008.

[95] L. Gentili, R. Naldi, and L. Marconi, "Modeling and control of VTOL UAVs interacting with the environment," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pp. 1231–1236, Dec. 2008.

[96] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," in *Proceedings of the Robotics: Science and Systems Conference (RSS)*, June 2009.

[97] P. Pounds and A. Dollar, "Aerial grasping from a helicopter UAV platform," in *Proceedings of the IFRR International Symposium on Experimental Robotics (ISER)*, 2010.

[98] P. Pounds, D. R. Bersak, and A. Dollar, "Grasping from the air: Hovering capture and load stability," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Beijing, China, July 2011.

[99] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," in *Proceedings of the Robotics: Science and Systems Conference (RSS)*, Los Angeles, CA, June 2011.

[100] J. Lee, "Hacking the Nintendo Wii remote," *IEEE Transactions on Professional Communication*, vol. 7, no. 3, pp. 39–45, July 2008.

[101] Kako, "Interfacing Wii Camera over I^2C," [http://kako.com/neta/2008-009/2008-009.html], 2011.

[102] Arduino, [http://www.arduino.cc], 2011.

[103] A. Dollar and R. Howe, "A robust compliant grasper via shape deposition manufacturing," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 2, pp. 154–161, Apr. 2006.

[104] LEGO Group, [http://www.lego.com], 2011.