Auton Robot (2012) 33:41–68
DOI 10.1007/s10514-012-9286-z

On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor

Vaibhav Ghadiok · Jeremy Goldin · Wei Ren

Received: 1 August 2011 / Accepted: 9 March 2012 / Published online: 29 March 2012

Abstract This paper presents the design and development of autonomous attitude stabilization, navigation in unstructured, GPS-denied environments, aggressive landing on inclined surfaces, and aerial gripping using onboard sensors on a low-cost, custom-built quadrotor. The development of a multi-functional micro air vehicle (MAV) that utilizes inexpensive off-the-shelf components presents multiple challenges due to noise and sensor accuracy, and there are control challenges involved with achieving various capabilities beyond navigation. This paper addresses these issues by developing a complete system from the ground up, addressing the attitude stabilization problem using extensive filtering and an attitude estimation filter recently developed in the literature. Navigation in both indoor and outdoor environments is achieved using a visual Simultaneous Localization and Mapping (SLAM) algorithm that relies on an onboard monocular camera. The system utilizes nested controllers for attitude stabilization, vision-based navigation, and guidance, with the navigation controller implemented using a nonlinear controller based on the sigmoid function. The efficacy of the approach is demonstrated by maintaining a stable hover even in the presence of wind gusts and when manually hitting and pulling on the quadrotor. Precision landing on inclined surfaces is demonstrated as an example of an aggressive maneuver, and is performed using only onboard sensing. Aerial gripping is accomplished with the addition of a secondary camera, capable of detecting infrared light sources, which is used to estimate the 3D location of an object, while an under-actuated and passively compliant manipulator is designed for effective gripping under uncertainty.

The quadrotor is therefore able to autonomously navigate inside and outside, in the presence of disturbances, and perform tasks such as aggressively landing on inclined surfaces and locating and grasping an object, using only inexpensive, onboard sensors.

Keywords Quadrotor · SLAM · Micro air vehicle · Aerial gripping · GPS-Denied environment · Indoor navigation

This research was supported by the National Science Foundation under CAREER Award ECCS-0748287.

Electronic supplementary material The online version of this article (doi:10.1007/s10514-012-9286-z) contains supplementary material, which is available to authorized users.

V. Ghadiok (✉) · W. Ren
Department of Electrical Engineering, University of California, Riverside, Riverside, CA 92521, USA
e-mail: vaibhav.ghadiok@ieee.org

W. Ren
e-mail: ren@ee.ucr.edu

J. Goldin
Electronic Systems Center, Hanscom Air Force Base, Bedford, MA 01731, USA
e-mail: jeremy.goldin@us.af.mil

1 Introduction

The past few years have seen a focus on the area of navigation in indoor GPS-denied environments for MAVs (He et al. 2008; Grzonka et al. 2009; Blösch et al. 2010). In the absence of global positioning, an MAV needs to rely on onboard sensing modalities such as laser range finders or cameras in order to determine its position. MAVs with such capabilities have applications in scenarios requiring sensing and observation, such as in search and rescue, surveillance or inspection. The extension of these functionalities to include active manipulation of entities external to the vehicle would vastly expand the applications of these systems, as

they move from mere passive observation and sensing to dynamic interaction with the environment. This would allow activities of gripping objects from places not suitable for landing, such as vertical surfaces, water, and radio towers, with potential applications in object retrieval and improved observation through manipulating barriers, objects or switches, as well as deployment and retrieval of sensor nodes in a sensor network. Aside from the requirements of functionality that are needed for commercial applications, a high value to price ratio will increase usage of new devices.

The choice of vision for navigation research, as opposed to laser range finders, is driven by the goals above, along with the research challenge involved with extracting useful information from such a rich sensor to perform navigation. In regards to the objectives above, cameras have the benefits of being low cost, low power, small size and light weight, and the capability of being used for multiple tasks, such as for navigating as well as performing surveillance.

1.1 Related work

Some of the early successes with quadrotors involved the use of 3D tracking systems, which became more popular with the availability of the Vicon system (Vicon 2011). These systems showed the initial promise of autonomous flights by allowing accurate 6-DOF state estimation without the problems of noise inherent in IMUs, including mechanical vibration noise, as in Castillo et al. (2005), Valenti et al. (2006). Initial implementations that used only onboard sensing for attitude stabilization are described in Bouabdallah et al. (2004), Guenard et al. (2005), Gurdan et al. (2007), Bouabdallah and Siegwart (2007), Hoffmann et al. (2007), Pounds et al. (2010) Some of these efforts culminated in several successful designs that are still in use.

Once autonomous attitude stabilization was achieved, navigation was the next challenge. Some early camera-based implementations for maintaining accurate hover flight using known targets or artificial markers are presented in Altuğ et al. (2005), Tournier et al. (2006), Romero et al. (2006), Guenard et al. (2008), Bourquardez et al. (2009), Rudol et al. (2010). Work on maintaining a hover using optical flow is presented in Fowers et al. (2007), Kendoul et al. (2009), Romero et al. (2009). A commercial quadrotor capable of maintaining a stable hover has been implemented (Parrot 2011). Work on localization and navigation in pre-mapped unstructured environments such as He et al. (2008), Grzonka et al. (2009), Soundararaj et al. (2009) allowed greater flexibility of the quadrotor as an autonomous MAV. Lately, with accomplishments in navigation in unknown environments, such as Achtelik et al. (2009), Celik et al. (2009), Blösch et al. (2010), the quadrotor has become a promising option for autonomously completing desired tasks.

Our work on aggressive landing on inclined surfaces has two analogs in the literature: that of landing on inclined surfaces, either static or dynamic surfaces such as landing on an aircraft carrier, and that of aggressive perching. An autonomous landing system for a radio-controlled hobby helicopter hovering above an inclined plane is presented in Moore (1994). In Oh et al. (2006), an autopilot is designed to land a tethered helicopter on a rocking ship. The specific maneuver of aggressive perching has seen only very recent research. Using a hobby traditional helicopter and Vicon, Bayraktar and Feron (2008) demonstrated perching on inclined surfaces using velcro and a state machine to transition between level flight and an open-loop perching maneuver. Recently in Mellinger et al. (2010), aggressive perching on inclined, vertical, and inverted surfaces was demonstrated using velcro and Vicon, with trajectory controllers defined by a sequence of segments, each with a goal state based on parameter adaptation from measured errors after experimental testing. Given the extremely precise information provided by motion capture systems, there are still real world issues to address with perching, despite the successful perching demonstrations of these works. Our work addresses the problem of landing aggressively on inclined surfaces using onboard sensing without the use of an active/passive perching mechanism.

Aerial manipulation on an MAV has seen very little published research. Some early work using magnets for picking objects is presented in Amidi et al. (1998). In Kuntz and Oh (2008), results using hoops to pick up objects are presented using a test rig that is simulated to move like an MAV. Some theoretical results on aerial vehicles interacting with the environment are presented in Gentili et al. (2008), while some experimental results of cooperative manipulation using cables are presented in Michael et al. (2009). In Pounds and Dollar (2010a, 2010b), experimental results of gripping using a commercial electric helicopter are shown along with theoretical results proving the stability of PID control for gripping, by modeling the gripper as an elastic linkage. 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 (Pounds et al. 2011). Additionally, some work has been done involving gripping with quadrotors in Lindsey et al. (2011), using the Vicon Motion Capture System. However, given the extremely precise information provided by the Vicon, many of the real world issues seen in gripping still need to be tackled.

1.2 Description of the proposed quadrotor system

In this paper, we present the design and development of a complete quadrotor system, which addresses attitude stabi-

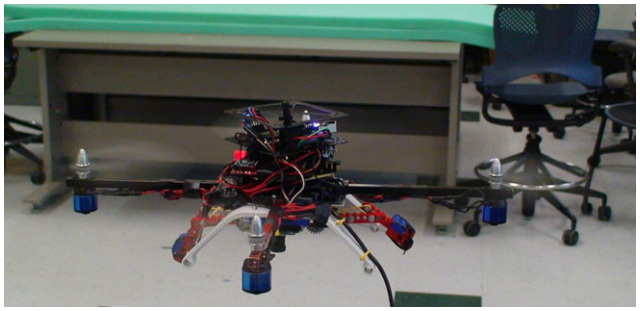


Fig. 1 The quadrotor, in flight, used to perform aerial gripping

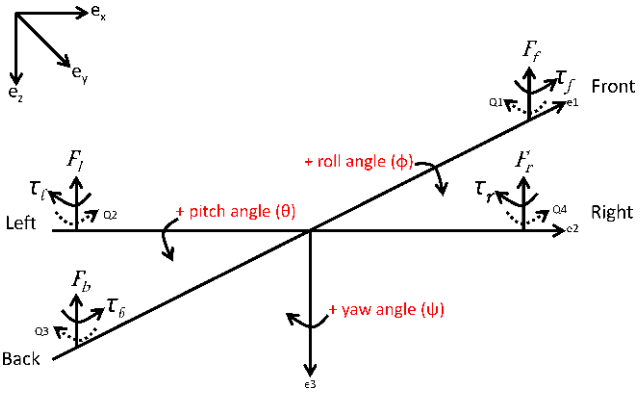


Fig. 2 Model of a quadrotor

lization and vision-based navigation with the use of inexpensive onboard sensors. One of our objectives is to extend previous findings on attitude stabilization and vision-based navigation to specifically address the issues that arise when lower quality, off-the-shelf, consumer components are used. We additionally propose solutions to advanced flight behaviors of stability in the presence of wind gusts and manual disturbance, aggressive landing on inclined surfaces, and aerial gripping. With this research, we hope to bring the use of such vehicles in everyday life closer to reality.

In order to reach the goals outlined above, we designed an attitude estimation and control system capable of effective stabilization in the presence of sensor and mechanical noise. Heavy low pass filtering of the onboard attitude sensors was needed in order to reduce sensor noise and mechanically induced noise during flight. An attitude estimation filter recently proposed in the literature is implemented in order to provide accurate information on the state of the quadrotor, while the attitude is stabilized using PD control. This is further explained in Sect. 3. The vision-based navigation system utilizes a nonlinear controller based on the sigmoid function that gives improved noise rejection and overall consistent performance despite the use of low-quality sensors. A downward-facing monocular camera, using a simultaneous localization and mapping (SLAM) algorithm is used to estimate the position and yaw of the quadrotor. The navigation controller is implemented as an outer controller, sending reference commands to the inner attitude controller. This is described in Sect. 4. Using path tracking control, aggressive landing on inclined surfaces is accomplished using only onboard sensors and is discussed in Sect. 5.

With the addition of another camera, as well as a third outer control loop, aerial gripping is accomplished with a gripper attached underneath the quadrotor and is discussed in Sect. 6. Experimental results demonstrating the attitude stabilized vehicle, vision navigation accuracy, and autonomous gripping capabilities of the quadrotor are presented in Sect. 7. Figure 1 shows the quadrotor and the gripper attached underneath it.

2 Quadrotor

The complete quadrotor system is developed from scratch; the physical system is built with low-cost constraints, necessitating adaptations of the control systems, noise filtering and attitude estimation. The attitude control system is developed based on the successful PID controller, with gains tuned empirically. The development of the quadrotor starts with modeling the vehicle as a rigid-body, and a linearized dynamic model is used for control.

2.1 Dynamic model

Modeling the quadrotor as a rigid-body, using Newtonian mechanics, let $\mathcal{I} = e_x, e_y, e_z$ denote the inertial frame, and $\mathcal{B} = e_1, e_2, e_3$ the aircraft body frame, as shown in Fig. 2. Then the model is:

$$\dot{\xi} = \upsilon \tag{1}$$

$$\dot{\upsilon} = g e_z - \frac{1}{m} t R e_z \tag{2}$$

$$t = b \sum_{i=1}^{4} \omega_i^2. \tag{3}$$

Here the vector $\xi = [x\ y\ z]^T$ represents the position of the origin of the body-fixed frame, with respect to the inertial frame; 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; $e_z = [0\ 0\ 1]^T$ is the unit vector in the inertial frame, $\mathcal{I}$; g is the acceleration from gravity; m is the mass of the vehicle; the orientation of the vehicle frame is given by the direction cosine 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 using (3), with b a constant of proportionality parameter that depends on aerodynamic effects, including the density of the air, and the size, shape, and pitch angle of the rotor blades; ω_i is the speed of

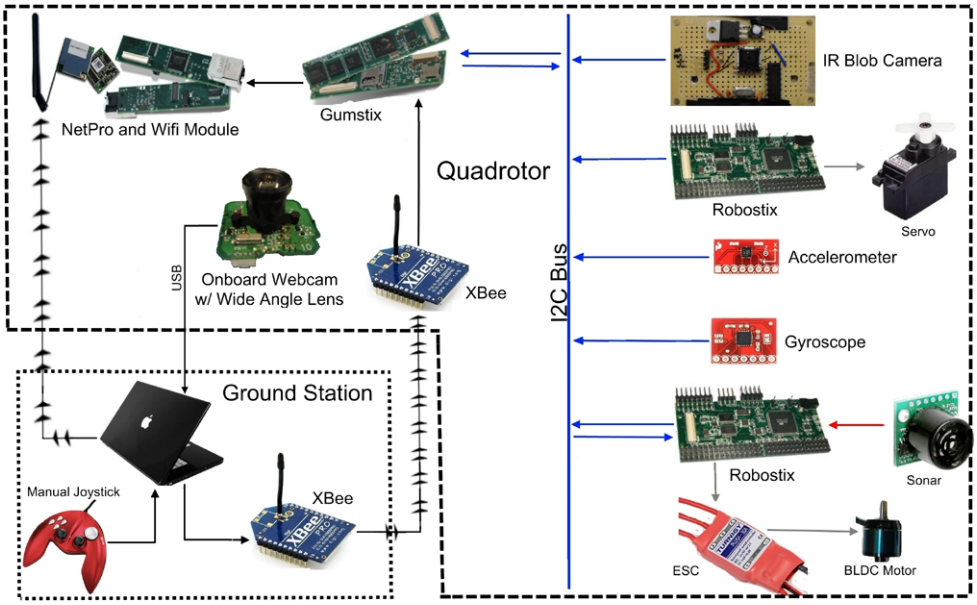


Fig. 3 Component level breakdown of the quadrotor system. *Blue lines* (I^2C bus in/out lines) represent I^2C communication, the *red line* (Sonar) represents analog input, *grey lines* (Robostix) represent PWM outputs, and *black lines* (USB/Zigbee/Gumstix) represent serial connections (Color figure online)

the rotors, $i \in 1, 2, 3, 4$. The dynamic model is

$$\dot{R} = R \cdot sk(\Omega) \tag{4}$$

$$I_f \dot{\Omega} = -\Omega \times I_f \Omega - G_a + \tau_a \tag{5}$$

$$I_r \dot{\omega}_i = \tau_i - Q_i, \quad i \in \{1, 2, 3, 4\} \tag{6}$$

$$Q_i = k\left(\omega_i^2\right) \tag{7}$$

$$G_a = \sum_{i=1}^{4} I_r (\Omega \times e_z)(-1)^{i+1} \omega_i \tag{8}$$

where Ω is the angular velocity, in roll, pitch and yaw of the vehicle in the body frame; $sk(X)$ denotes the creation of the skew-symmetric matrix; I_f is the inertia matrix of the airframe, where the center of mass is considered to coincide with the origin of the frame, $\mathcal{B}$; I_r signifies the moment of inertia of the rotor blades; Q_i is the reactive torque generated in free air by the rotor due to drag, with k a constant of proportionality parameter that depends on aerodynamic effects; G_a is the gyroscopic torque due to the combination of the rotation of the airframe and the four rotors; τ_a is the airframe torque generated by the rotor; τ_i represents the four control inputs to the system, in the form of motor torques.

2.2 Hardware architecture

The quadrotor is custom-made from available consumer-grade components. Figure 3 gives a breakdown of the primary parts of the system and how they communicate with each other. The quadrotor, including the gripping system, weighs 1.4 kg and measures 50 cm from end to end of the frame. The total system is very low cost, at a prototype price of less than $1000, with the IMU up to two orders of magnitude less than other quadrotor implementations found in the literature. The system uses the following major components:

- Turnigy Plush 30 A Electronic Speed Controllers (ESCs) (reflashed with modified firmware)
- KDA20-22L Hacker-style brushless DC motors
- Inertial Measurement Unit (IMU) consisting of gyroscopes and accelerometers:
  - InvenSense ITG3200 MEMS 3-axis gyroscope
  - Bosch Sensortec BMA180 3-axis accelerometer
- MaxBotix LV-EZ2 sonar module
- Gumstix Verdex Pro XL6P (w/netpro and wi-fi module)
- two Robostix using Atmega128 microcontrollers
- Logitech Quickcam Pro 5000 camera retrofitted with a 2.1 mm wideangle lens
- IR blob detecting camera
- micro servo

The Robostix, IR camera, and IMU sensors communicate over I^2C to the Gumstix, which is the master. One of the Robostix controllers reads in the sonar on its ADC and also outputs PWM signals to the ESCs for control of the motors. The second Robostix operates the micro servo. The servo and IR camera are used for gripping, with the IR LED detection aspects used for guidance control. Attitude stabilization and indoor/outdoor navigation are performed by the rest of the system. The monocular camera transmits images through a USB cable linked to the ground station, which consists of an Intel Core 2 CPU 2 × 2.4 GHz processor running Ubuntu.

It is important to note that this Gumstix single board computer does not have a floating point unit, so in this imple-

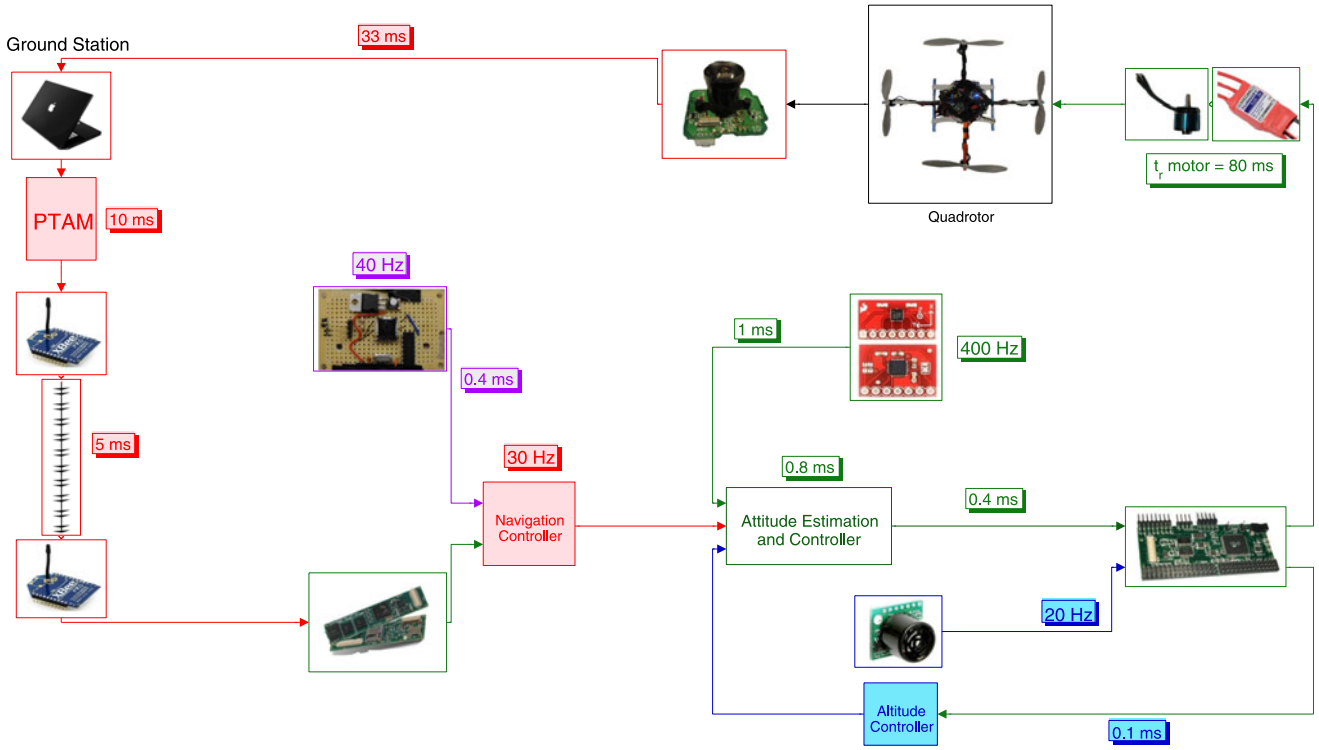


Fig. 4 Quadrotor system latency diagram by component and communication. The *green blocks* indicate the attitude control system; the *blue blocks* are for the sonar altitude controller; *red* indicates the navigation system utilizing the SLAM algorithm on the ground station, which returns position information. The *purple* refers to the gripping controller (Color figure online)

mentation, software floating point is used to obtain greater accuracy at the cost of computation speed.

2.2.1 *ESC firmware filter*

The hobbyist-grade ESCs arrive with a manufacturer developed motor control firmware, which contains a low-pass filter, primarily for power savings for use on a fixed wing aircraft. Such a filter dramatically diminishes the responsiveness of the motors in a quadrotor application, due to the necessity to quickly adjust thrust on individual motors by directly changing the motor speed. The firmware must thus be modified to eliminate the low pass filter, so that quickly changing commands from the microcontroller to the ESC will be equivalently driven to the motor.

2.2.2 *Center of gravity*

The center of gravity (CG) of the quadrotor is (−0.182 cm, −0.265 cm, −3.573 cm) relative to the center of the rotor plane. The weight of the USB wire that hangs from the quadrotor to the ground is included, and so brings the CG much lower. During development, the rotor plane and CG were adjusted relative to each other, from CG above rotor plane to CG below rotor plane, as well as roughly coincident. Empirically, having the CG slightly below the rotor plane yielded the most stable and accurate flights.

2.3 System latencies

System latencies and delays can contribute to instability and poor performance. A full timing analysis was done in order to understand what delays impact the system and what can be reduced or adjusted for. A diagram of the system delays is shown in Fig. 4. Some of these latencies are discussed in the relevant sections.

2.3.1 *Actuation*

A simple setup consisting of a photodiode and an infrared (IR) Light-emitting diode (LED), with the propeller set to spin between the two was employed to measure the speed of the propeller. The motor response to a step input was estimated by logging the measured speed data using a National Instruments Data Acquisition Card and analyzing the data in Labview. The time constant was measured over a variety of thrust ranges, but primarily around the actual flight envelope, as that is where the system dynamics are taking place. Upon system identification, the motor response time was found to be around 80 ms.

Another constant that was estimated using this setup relates to the mapping between the output of the PD controller and the RPM achieved. This is further discussed in Sect. 3.2. A load cell was added to the setup described above to measure the thrust generated by the propeller. The thrust generated at various speeds was logged and the coefficient b in (3) was estimated.

2.3.2 Onboard computation

The I^2C protocol forms the backbone for all onboard communications and connects almost all the devices, such as the Robostix, accelerometer, gyroscope and IR camera to the Gumstix. I^2C is set to operate at 400 kHz in the Gumstix code (maximum clock rate supported by the devices connected to the Gumstix), giving approximately a net throughput of 27.5 Kbytes/sec with an estimated overhead of 0.3 ms. These numbers give an estimate of the time taken to transmit/receive X number of bytes, accounting for all overheads. In this protocol, the master (Gumstix) addresses a slave and then waits for the slave to respond. The response time is dependent on the slave and is not deterministic. However, the response time in our setting was found to be small. To get a more realistic idea of the time taken for each I^2C read/write, small pieces of Gumstix code were written to read/write several thousand bytes from/to a device to calculate an average, since the time taken by an individual read/write is very small. We ultimately observed the waveform of the I^2C protocol for various data reads and writes on an oscilloscope to get a final confirmation of the latency indicated in the figure. The sonar sensor data is received by the Robostrix via an ADC. The ADC overhead is very small, so the rate of sonar data is taken from the manufacturer data on the sonar used. The frequency of each control loop was measured using timers placed within the C code.

2.3.3 Ground station communication and offboard computation

The latencies for these sub-systems are discussed in Sect. 4.3

3 Attitude stabilization

The MEMS-based accelerometers and gyroscopes provide linear accelerations and angular rates, which are read by the Gumstix over I^2C at a rate of about 400 Hz. The reported accelerations and angular rates are low pass filtered before being read by the Gumstix, using the individual sensors' built-in customizable filters. These filters are set to 10 Hz cutoff for the accelerometers, and 25 Hz cutoff for the gyroscopes. These values were determined through analysis of FFT plots and empirical study of the behavior of our quadrotor in hand and in flight, iteratively progressing through the possible filter bandwidths to determine the lowest acceptable cutoff frequency that did not compromise actual signal integrity. The sensor data is further FIR low pass filtered on the Gumstix using very short filter lengths of length 8 in order to minimize delay. These software filters were designed in Matlab using a weighted least squares fitting for the desired filter length. Chosen bandwidths are also 10 Hz for the accelerometers and 25 Hz for the gyroscopes. Such heavy filtering is required due to the extensive noise coupling into the sensors, as well as the general lower quality of MEMS-based sensors. Moreover, MEMS-based gyroscopes suffer from drift and this problem is exacerbated when trying to integrate to obtain angles, due to integration drift. Nonetheless MEMS-based sensors are preferred due to their small weight, size and cost.

3.1 Attitude estimation filter

Attitude estimation is performed by fusing linear accelerations given by the accelerometers with the angular rates reported by the gyroscopes. The two classical approaches to attitude extraction using multiple sensors are the Kalman filter and the Linear Complementary Filter.

3.1.1 Kalman filter

The Kalman filter is a standard solution to this problem, however, real systems are neither linear nor suffer from gaussian noise.

3.1.2 Linear complementary filter

The linear complementary filter exploits the differing spectral characteristics of the measurement sources and fuses these using a fixed gain. However, the filter relies on the measurement system being able to be accurately linearized.

3.1.3 Nonlinear complementary filter

The nonlinear complementary filter developed in Mahony et al. (2008) provides a superior method to fuse the accelerometer and gyro measurements, exploiting the known properties of the underlying system. The specific form of the nonlinear complementary filter that is used for our implementation is one of the three nonlinear complementary filter types designated by Mahony et al. (2008), which they term the passive nonlinear complementary filter. The rotation of the quadrotor can be described using a rotation matrix and all possible orientations of the quadrotor live in the space of 3×3 orthogonal matrices known as the Special Orthogonal Group $SO(3)$. The rotational kinematics of the quadrotor is given by (4) and this filter was specifically designed on the $SO(3)$ group to exploit this fact. We summarize in this section the functioning of this filter and how we use it.

The proposed observer/filter is posed as a kinematic system

$$\dot{\hat{R}} = (\hat{R}\Omega + k_p \hat{R}\omega)_X \hat{R} \tag{9}$$

to give an estimate, $\hat{R}$, of the attitude of the system. Here, $\hat{R}$ 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 $\hat{R}^T R_y$, k_p is a non-zero positive gain, and $()_X$ 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. A block diagram of the nonlinear filter is shown in Fig. 5.

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 (see (10) and (11) for details). The innovation term, ω, can be seen as a nonlinear approximation of the error between R and $\hat{R}$. Through Lyapunov analysis, Mahony et al. (2008) shows 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.

Initial angular estimates are provided by the accelerometer, using

$$\phi = \arctan\left(\frac{\ddot{z}}{\ddot{y}}\right) + \frac{\pi}{2}, \tag{10}$$

$$\theta = -\arctan\left(\frac{\ddot{z}}{\ddot{x}}\right) - \frac{\pi}{2}, \tag{11}$$

where the angles are measured in radians, ϕ and θ are the roll and pitch angles, respectively. The accelerations, $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ are for the x, y, and z measured accelerations, in m/s^2. Note that in practice, the *atan2* function is used for robustness. Angles exceeding π and below $-\pi$ are explicitly taken care of.

From the block diagram of the nonlinear filter (Fig. 5), one can notice the structural similarity with the linear/classical complementary filter, and hence the name they gave it. 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

$$B = k_p \cdot \pi_a(\tilde{R}) + (\hat{R}\Omega)_X. \tag{12}$$

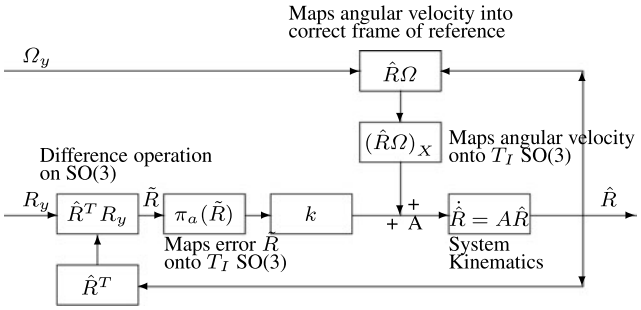


Fig. 5 Block diagram of the nonlinear complementary filter

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,

$$A = I_3 + B\frac{\sin(|vex(B)|\,dt)}{|vex(B)|dt} + B^2\frac{1-\cos(|vex(B)|\,dt)}{|vex(B)|\,dt}. \tag{13}$$

The final form of the filter along with the bias estimator is given by,

$$\dot{\hat{R}} = \hat{R} \cdot (\Omega_y - \hat{b} + \omega)_X \tag{14}$$

$$\dot{\hat{b}} = -k_b \cdot vex\big(\pi_a(\tilde{R})\big), \quad k_b > 0, \tag{15}$$

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 (14) 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.

3.1.4 Performance and motivation

The nonlinear passive complementary filter is found to perform significantly better than the Kalman filter, as can be seen in Fig. 6 from data taken during a typical hover. 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, giving the advantage of avoiding corrupting $\hat{R}$ with the noise and errors in the reconstructed pose from the accelerometers.

3.2 Attitude controller

With the attitude of the quadrotor accurately estimated, the attitude is then stabilized using a PD controller, where the

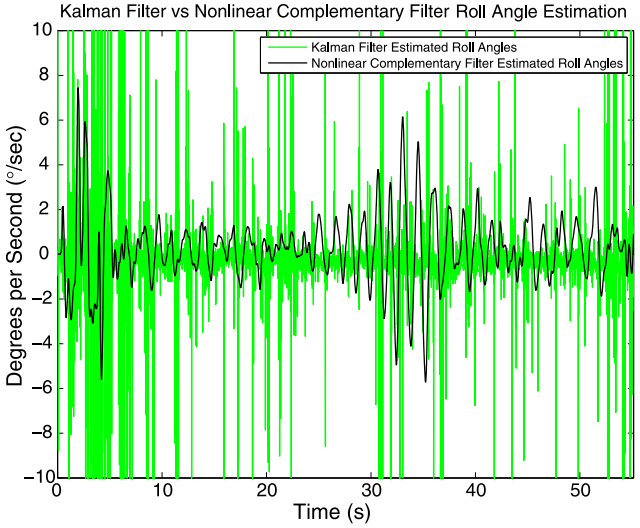


Fig. 6 Comparison of the attitude of the quadrotor, for the roll angle, extracted using the nonlinear passive complementary filter and the Kalman filter. This data is from a typical hover flight maintained using vision-based navigation

roll, pitch and yaw control values, u_ϕ, u_θ and u_ψ, are governed by

$$u_\phi = k_{p,\phi} \cdot \left(\phi^{des} - \phi\right) + k_{d,\phi} \cdot \left(\dot{\phi}^{des} - \dot{\phi}\right) \tag{16}$$

$$u_\theta = k_{p,\theta} \cdot \left(\theta^{des} - \theta\right) + k_{d,\theta} \cdot \left(\dot{\theta}^{des} - \dot{\theta}\right) \tag{17}$$

$$u_\psi = k_{p,\psi} \cdot \left(\psi^{des} - \psi\right) + k_{d,\psi} \cdot \left(\dot{\psi}^{des} - \dot{\psi}\right) \tag{18}$$

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.

The dynamic model, (2)–(8) described in Sect. 2, is used for calculating the desired angular rates, $\dot{\phi}^{des}$, $\dot{\theta}^{des}$, $\dot{\psi}^{des}$. Using the approximation that the rotation matrix in (4) is identity, and linearizing (5) about the hover point with small angle approximations, we get,

$$\dot{\phi}^{des} = \frac{4k_F L\omega_h}{I_{xx}}(u_\phi k_{\text{rpm}} + c_{\text{rpm}}), \tag{19}$$

$$\dot{\theta}^{des} = \frac{4k_F L\omega_h}{I_{yy}}(u_\theta k_{\text{rpm}} + c_{\text{rpm}}), \tag{20}$$

$$\dot{\psi}^{des} = \frac{8k_M\omega_h}{I_{zz}}(u_\psi k_{\text{rpm}} + c_{\text{rpm}}), \tag{21}$$

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_ϕ, u_θ, u_ψ, are determined using (16)–(18); and the terms, I_{xx}, I_{yy}, I_{zz}, are the diagonal elements of the inertia matrix. The inertia matrix was calculated using direct 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

$$\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} = \begin{bmatrix} 0.02509 & 0.00016 & -0.00276 \\ 0.00016 & 0.02610 & 0.00070 \\ -0.00276 & 0.00070 & 0.02262 \end{bmatrix}. \tag{22}$$

3.3 Altitude controller

Altitude is measured using a downward-facing sonar, which is sampled at 20 Hz. The sampled data is run through a length-3 median filter to reject outliers. A PID controller along with a feedforward term, (23), is found to compensate effectively for the nonlinear effects involved in the altitude dynamics of the quadrotor, using

$$u_{alt} = k_{p,alt} \cdot \left(z^{des} - z\right) + k_{i,alt} \cdot \int_0^t \left(z^{des} - z\right)dt + k_{d,alt} \cdot \left(\dot{z}^{des} - \dot{z}\right) + u_{nom}, \tag{23}$$

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

$$z = (\cos\phi \cdot \cos\theta) \cdot z_{\text{sonar}}. \tag{24}$$

In (23), the velocity, $\dot{z}$, is obtained by finite differentiation of the input sonar measurements.

3.4 Actuation

The controller outputs, u_ϕ, u_θ, u_ψ and u_{alt}, are inputs to the ESCs in terms of PWM values, and they are converted to individual motor commands using

$$\begin{bmatrix} u^1_{\text{PWM}} \\ u^2_{\text{PWM}} \\ u^3_{\text{PWM}} \\ u^4_{\text{PWM}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_{alt} \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix}, \tag{25}$$

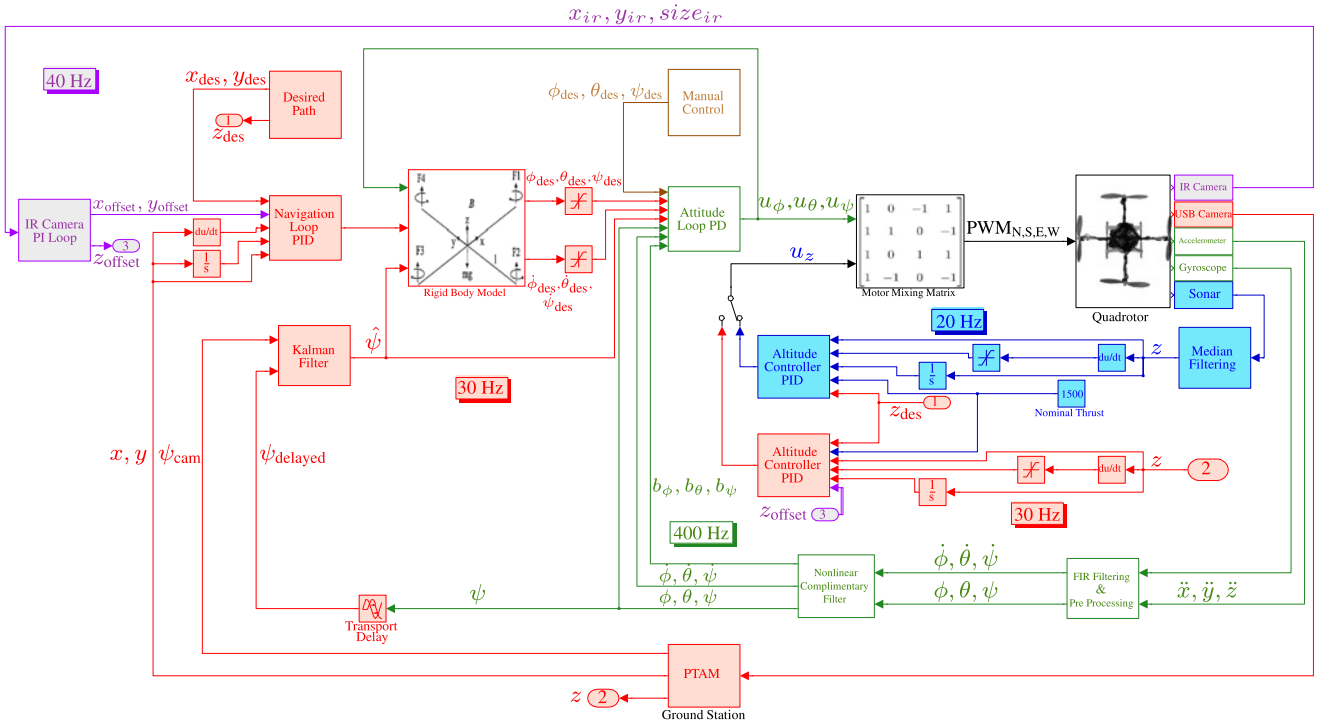


Fig. 7 Control block diagram. The *green blocks* indicate the attitude control system; the *blue blocks* are for the sonar altitude controller; *red* indicates the navigation system utilizing the SLAM algorithm on the ground station, which returns position information. The *purple* refers to the gripping controller. A backup manual controller for safety is shown in *tan* (Color figure online)

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 (5), with

$$\omega_i = k_{\text{rpm}} \cdot u^i_{\text{PWM}} + c_{\text{rpm}}, \quad i \in 1,2,3,4 \tag{26}$$

$$\tau_a = \left(\tau_a^1, \tau_a^2, \tau_a^3\right)^T \tag{27}$$

$$\tau_a^1 = L \cdot b\left(\omega_2^2 - \omega_4^2\right) \tag{28}$$

$$\tau_a^2 = L \cdot b\left(\omega_1^2 - \omega_3^2\right) \tag{29}$$

$$\tau_a^3 = k\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right). \tag{30}$$

This attitude controller achieves successful stabilization of the roll, pitch and yaw of the quadrotor, allowing the addition of a vision navigation loop for position control of the vehicle. The attitude controller, as the innermost controller of the complete quadrotor control system, is shown in Fig. 7. Attitude system gains used to achieve stabilization are shown in Table 1.

Table 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 Navigation

The navigation system uses a visual SLAM algorithm to obtain position and heading measurements from an onboard, downward-facing monocular camera. The algorithm runs off-board on the ground station and transmits the position and yaw estimates to the quadrotor over a Zigbee link. A PID controller is implemented onboard for tracking the desired references. The current system uses a wired onboard camera that sends the camera images over USB for offboard processing, with offboard chosen for rapid-prototyping simplicity, although the SLAM algorithm is capable of being run onboard the quadrotor using one of the widely available

dual-core computers. The hanging USB cable does interfere with efficient stabilization and positioning of the quadrotor, acting as an external disturbance and limiting the accuracy of the system. Despite this handicap we are still able to show that the quadrotor can accurately stabilize and navigate.

4.1 SLAM using a camera

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 (Davison et al. 2007), in which the state vector consists of the 6-DOF pose of the vehicle, and is augmented to include distinctive landmarks in the environment as they are observed. 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) (Lowe 2004) are employed, however, this leads to a very sparse map. 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 (Montemerlo et al. 2003) that maintains multiple hypotheses of the vehicle pose and map and is computationally more efficient.

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 (Strasdat et al. 2010). This was the motivation of choosing the second approach, SfM, over Kalman Filter based techniques.

4.2 Overview of the vision-based SLAM algorithm

The outlined approach uses the visual SLAM algorithm developed in Klein and Murray (2007). 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. 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 non-linear 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 efficient algorithms, such as Sparse Bundle Adjustment (Lourakis and Argyros 2009), exploit the sparsity of the Hessian Matrix.

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 a stereo technique based on a specific translation between two locations during map generation, as implemented in Klein and Murray (2007). 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 outlined in Klein and Murray (2008), before moving to and mapping new locations.

As a side note, the OpenCV implementation of the pyramidal KLT Tracker (Bouguet 1999) was also tested but found to be less robust than the tracking algorithm employed in this work.

4.3 Controller design

The navigation and attitude (and altitude) controllers are cascaded with the attitude (and altitude) controller being the inner loop. The attitude (and altitude) controllers take in the desired roll ϕ, pitch θ, heading ψ and the desired height z^{des} as inputs from the navigation controller. 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. The delay is calculated based on the estimated total time for acquiring an image, performing feature tracking (for a reasonably large map) and wirelessly transmitting the position data to the onboard controller. These delays were shown in Sect. 2, Fig. 4. Note that the time taken for feature tracking would increase with an increase in feature points and that bundle adjustment essentially serves to

refine the map and is not accounted for in the latency calculation. The time for acquiring an image and performing feature tracking is calculated by timing the relevant code sections and the time for wireless transmission is calculated based on information provided by the manufacturer of the transceiver.

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 (25) to

$$\begin{bmatrix} u^1_{\mathrm{PWM}} \\ u^2_{\mathrm{PWM}} \\ u^3_{\mathrm{PWM}} \\ u^4_{\mathrm{PWM}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \left(\begin{bmatrix} u_{alt} \\ u_{\phi} \\ u_{\theta} \\ u_{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ u_{nav,x} \\ u_{nav,y} \\ 0 \end{bmatrix} \right). \quad (31)$$

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.

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

$$\ddot{x}^{des} = k^x_{p,nav} \cdot \left(x^{des} - x\right) + k^x_{i,nav} \cdot \left(\int_0^t \left(x^{des} - x\right) dt \right) + k^x_{d,nav} \cdot \left(\dot{x}^{des} - \dot{x}\right) \quad (32)$$

$$\ddot{y}^{des} = k^y_{p,nav} \cdot \left(y^{des} - y\right) + k^y_{i,nav} \cdot \left(\int_0^t \left(y^{des} - y\right) dt \right) + k^y_{d,nav} \cdot \left(\dot{y}^{des} - \dot{y}\right), \quad (33)$$

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 here can effectively account for small offsets since the measurement data is relatively accurate even for small measurements (unlike in the attitude system). These offsets can be due to in-flight effects such as actuator friction buildup in the bearing and efficiency changes due to heating, and an integrator can account for them as well as for the reduction of steady-state error. 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. 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.

Table 2 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	$\dot{k}^x_{p,nav}$	0.70
Y Derivative	$\dot{k}^y_{p,nav}$	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

Navigation gains used to achieve accurate hover and path following are shown in Table 2.

The outputs of the PID controllers in (32) and (33) are desired accelerations, based on linearization of (3) about the hover region for the acceleration of the center of mass in the inertial frame, and the desired angles are calculated after accounting for the yaw of the vehicle, using

$$\begin{bmatrix} \phi^{des} \\ \theta^{des} \end{bmatrix} = \frac{1}{g} \begin{bmatrix} \sin\psi & -\cos\psi \\ \cos\psi & \sin\psi \end{bmatrix} \begin{bmatrix} \ddot{x}^{des} \\ \ddot{y}^{des} \end{bmatrix}. \quad (34)$$

ϕ^{des} and θ^{des} are then sent to the attitude loop using (16) and (17).

Integration of the yaw angular rate is found to give insufficiently accurate estimates of the yaw angle over time. 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. The height, z, obtained using the SLAM algorithm can be directly used for visual altitude control, using a similar PID controller as in (23).

The navigation controller also has a specific path tracking mode, in which the quadrotor tracks a path at a desired velocity. The controller works on normal and tangent components of the path as described below.

4.4 Path following control

A path is defined, $P \in N \times \mathbb{R}^2$, by a sequence of N desired way-points in two dimensions, $[x^{des}_i \; y^{des}_i]^T$, along a path segment P_i connecting way-point i to $i+1$, with a constant desired speed of travel for the path, $[\dot{x}^{des}_{pos} \; \dot{y}^{des}_{pos}]^T$. This path definition is shown visually in Fig. 8. Let t_i be the unit tangent vector in the direction of travel along the path from $[x^{des}_i \; y^{des}_i]^T$ to $[x^{des}_{i+1} \; y^{des}_{i+1}]^T$, and n_i be the unit normal vector to the path. Then, given the actual current position of

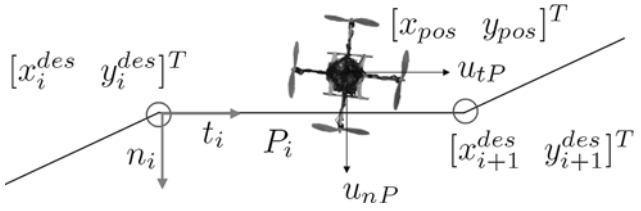


Fig. 8 Path definition, with way-points i and $i+1$ and corresponding path segments, noting the tangent and normal path components

the vehicle, $[x_{pos} \; y_{pos}]^T$, the normal path errors, $e_{nP}, \dot{e}_{nP}$ and tangent path errors, $e_{tP}, \dot{e}_{tP}$ are

$$e_{nP} = \left(\begin{bmatrix} x_i^{des} \\ y_i^{des} \end{bmatrix} - \begin{bmatrix} x_{pos} \\ y_{pos} \end{bmatrix} \right) \cdot n_i, \tag{35}$$

$$e_{tP} = \left(\begin{bmatrix} x_i^{des} \\ y_i^{des} \end{bmatrix} - \begin{bmatrix} x_{pos} \\ y_{pos} \end{bmatrix} \right) \cdot t_i, \tag{36}$$

$$\dot{e}_{nP} = - \begin{bmatrix} \dot{x}_{pos} \\ \dot{y}_{pos} \end{bmatrix} \cdot n_i, \tag{37}$$

$$\dot{e}_{tP} = \left(\begin{bmatrix} \dot{x}_i^{des} \\ \dot{y}_i^{des} \end{bmatrix} - \begin{bmatrix} \dot{x}_{pos}, \\ \dot{y}_{pos} \end{bmatrix} \right) \cdot t_i, \tag{38}$$

where the tangent path components of the desired velocities must be taken since the desired velocities are implemented independently from the desired path, in terms of x and y components.

Both the along path error and error rates are used in order to simplify the controller operation between hover and path modes. This allows for an essentially seamless transition between the two modes, utilizing the same controller but just different desired positions and velocities. In the path, the error is used to drive the system to the desired position, while the error rate is used to keep the quadrotor to a specified velocity. Thus, the along path tracking employs PD control, while the normal path regulation employs PID control using

$$\begin{aligned} u_{tP} &= \begin{bmatrix} u_{nav,x} \\ u_{nav,y} \end{bmatrix}_{tP} \\ &= \begin{bmatrix} k^x_{ptP,nav} & 0 \\ 0 & k^y_{ptP,nav} \end{bmatrix} e_{tP} \\ &\quad + \begin{bmatrix} k^x_{dtP,nav} & 0 \\ 0 & k^y_{dtP,nav} \end{bmatrix} \dot{e}_{tP} \\ u_{nP} &= \begin{bmatrix} u_{nav,x} \\ u_{nav,y} \end{bmatrix}_{nP} \\ &= \begin{bmatrix} k^x_{pnP,nav} & 0 \\ 0 & k^y_{pnP,nav} \end{bmatrix} e_{nP} \\ &\quad + \begin{bmatrix} k^x_{dnP,nav} & 0 \\ 0 & k^y_{dnP,nav} \end{bmatrix} \dot{e}_{nP} \end{aligned} \tag{39}$$

$$+ \begin{bmatrix} k^x_{inP,nav} & 0 \\ 0 & k^y_{inP,nav} \end{bmatrix} \int_0^t e_{nP} dt. \tag{40}$$

The path gains and the hover gains are set equal, limiting the need to retune. Although task specific gains could be explored in future work, the hover gains were found to have a good carryover to the path following. Since, u_{tP} and u_{nP} are ultimately implemented in terms of x and y axes, they need to be merged to obtain the desired accelerations in x and y, using,

$$\ddot{x}^{des} = u^{tP}_{nav,x} + u^{nP}_{nav,x}, \tag{41}$$

$$\ddot{y}^{des} = u^{tP}_{nav,y} + u^{nP}_{nav,y}. \tag{42}$$

Completion of segment i for transition to segment $i+1$ occurs at the point that the quadrotor reaches way-point $i+1$. Upon completion of P_i, the integrators for the normal path error are reset if the direction to way-point $i+2$ is not tangent to t_i. This is to account for the fact that the built up error for the normal path is tied to the path direction which is based upon the components of x and y, and will not carry over directly to a different path direction. Alternatively, low curvature paths where the angle between t_i and t_{i+1} is small, the integrator could be kept. Additionally, it might be beneficial in future work to take components of the built up integrator for appropriate application to the current direction, allowing for reduction in the time required for the integrator to build up for the new path segment. The path does not incorporate changes in the vertical path dimension. Paths are generated in Matlab, but could easily be generated onboard the quadrotor. A distance gain is used in the path generation, the value for which sets the spacing between way-points, and is related to the speed at which the quadrotor is intended to travel the path and must be adjusted concurrently with the desired speed setting.

4.5 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. Another major reason for the quadrotor displaying overshoot and slowly damped oscillations when given step inputs or disturbances was due to measurement noise, finite differentiation effects and the over-hanging cable. For example, when the quadrotor was commanded to increase its height by a large step input, the hanging cable could create a transient disturbance

due to its weight or even occlude the sonar attached underneath reporting a smaller height than the true value. Similarly, when navigating in the x and y axes, spikes in the position data from the visual SLAM algorithm were observed implying a large change in position even though the quadrotor had moved a small distance. These lasted long enough to cause a jerk in the quadrotor motion. These spikes could be attributed to the cheap off-the-shelf webcam used that could suffer from blurred frames under motion leading to a momentary failure in tracking.

4.6 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. With the output required to be zero when the input is zero, the function takes the simplified form

$$Y(t) = \frac{A}{1 + e^{-Bt}} - A/2, \tag{43}$$

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

4.6.1 Altitude regulation

The altitude is regulated using a PID controller, with the P and D commands 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

$$u_{alt} = \frac{A_{kp}^{alt}}{1 + e^{-B_{kp}^{alt}(z^{des} - z)}} - \frac{A_{kp}^{alt}}{2} + k_{i,alt} \int_0^t (z^{des} - z) dt + \frac{A_{kd}^{alt}}{1 + e^{-B_{kd}^{alt}(\dot{z}^{des} - \dot{z})}} - \frac{A_{kd}^{alt}}{2}, \tag{44}$$

where u_{alt} is used as described in Sect. 3, with the motor actuation for the PWM to the motors given by (25). The gain constants used for the nonlinear altitude controller are indicated in Table 3.

4.6.2 Navigation

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,

$$S_{kp}^{*} = \frac{A_{kp}^{nav,*}}{1 + e^{-B_{kp}^{nav,*}(*_{pos}^{ref} - *_{pos})}} - \frac{A_{kp}^{nav,*}}{2},$$

Table 3 Altitude nonlinear 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

Table 4 Navigation nonlinear 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

$$u_{nav,x} = S_{kp}^{x} + ki_{sig}^{nav,x} \int_0^t S_{kp}^{x} dt + \frac{A_{kd}^{nav,x}}{1 + e^{-B_{kd}^{nav,x}(\dot{x}_{pos}^{ref} - \dot{x}_{pos})}} - \frac{A_{kd}^{nav,x}}{2}, \tag{45}$$

$$u_{nav,y} = S_{kp}^{y} + ki_{sig}^{nav,y} \int_0^t S_{kp}^{y} dt + \frac{A_{kd}^{nav,y}}{1 + e^{-B_{kd}^{nav,y}(\dot{y}_{pos}^{ref} - \dot{y}_{pos})}} - \frac{A_{kd}^{nav,y}}{2}, \tag{46}$$

where $*$ is either x or y depending on the context, and $u_{nav,x}, u_{nav,y}$ are used as described in Sect. 4 with the output being interpreted as a desired acceleration and then used to determine the desired angle according to (34). The gain constants used for the nonlinear navigation controller are indicated in Table 4. A similar controller has been applied in Ahn and Thanh (2005), however, the saturation is applied to the output of the PID rather than each of the components.

Note that a linear integrator is not used here, as unlike for the altitude control, the integrator will begin to build up excessively if far from the desired position, causing an unrecoverable situation. As shown, the integrator term is determined from integrating the position error after it has been modified through the nonlinear proportional function, making it somewhat of a hybrid linear-nonlinear integrator, as it linearly accumulates the nonlinear system output. Although if constantly far from the nominal position, this integrator would get large, in general it stays small enough due to the

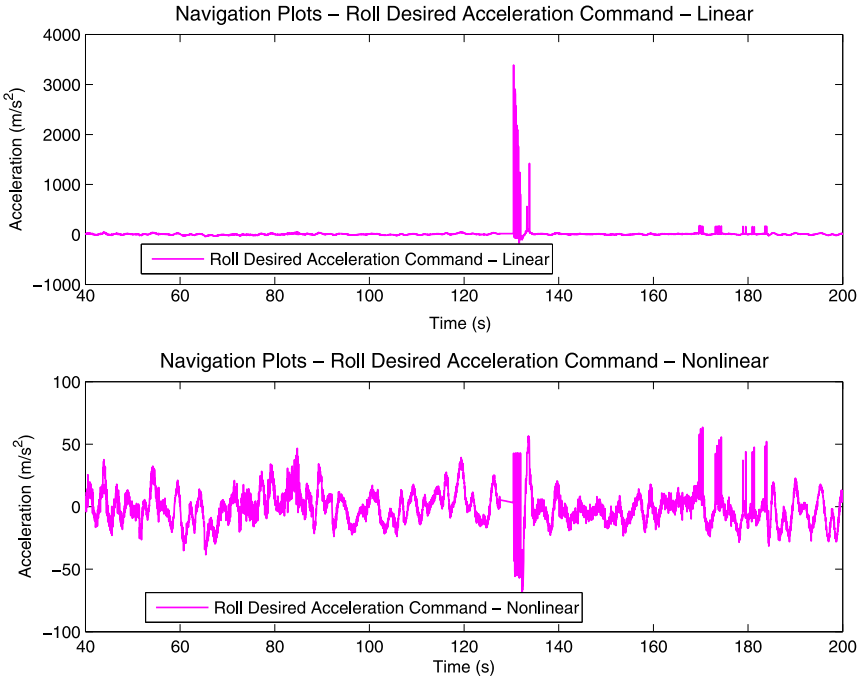


Fig. 9 Desired accelerations for the linear vs. nonlinear controllers—x axis

gain value and the integration of nonlinear saturated outputs. Calculating the integrator term via integration of the actual position error and then run through its own nonlinear function for growth rate and saturation level tunings is not a good option. This is because when far from the desired position, the integrated position error will get large very fast, and if this value is used for integration, it will be very large most of the time, and thus when run through the nonlinear controller, will always be at the saturation level. An alternative method is to blend the two, where the integration is performed on the nonlinear system output, as it is now, but then that value is operated on by its own nonlinear function, with separate growth rate and saturation level tunings. This could be explored in future work for possible improvements in performance.

4.7 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 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. 9, 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. This reduction in large changes due to measurement noise provided a large improvement in the quality 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. We would like to note that while using a saturation greatly improved the performance of our system, we do not claim that a saturation-based controller outperforms a standard PID controller in general. In circumstances where a large disturbance is received, the saturation might curtail the ability of the quadrotor to respond by not allowing greater control effort. In such cases, a standard PID might work better.

5 Aggressive landing on inclined surfaces

Building upon the capabilities presented in the previous sections, the idea of pushing the flight envelope assumptions of near hover region control becomes feasible. Although the capabilities already presented yield a very capable autonomous MAV, it becomes apparent upon the study

of manually controlled MAVs flown by expert pilots that autonomous MAVs are capable of performing more maneuverable flight, and such capability is a desirable trait for the ultimate goals of these vehicles.

The demonstration of aggressive and precise landing on inclined surfaces illustrates a maneuver that would allow such abilities as landing in constrained spaces. The applications of this capability include surveillance, inspection, and mobile sensor networks. The practical use of such abilities (using an engagement mechanism) is already being examined, including the capability to perch on power lines for recharging (Moore and Tedrake 2009).

More recently, research has focused on aggressive perching examining specifically the type of maneuver that many birds routinely perform, which is that of an aggressive aerobatic maneuver characterized by a rapid reduction in speed through the use of a high angle of attack of the aircraft wing surfaces such that a point landing can be achieved (Desbiens et al. 2011; Cory 2010). Aggressive maneuvers using a rotorcraft, rather than a fixed-wing, presents some unique challenges stemming from various aerodynamic effects that become significant at rotor descent and translation speeds that are comparable to the induced wind speed, including blade flapping, translational lift, and toroidal vortexes (Hoffmann et al. 2007; Huang et al. 2009). The maneuver performed in this work is analogous to this type of aggressive perching in every respect except for the absence of an engagement mechanism, albeit with less accuracy than would typically be required.

In this work, initial results on landing on inclined surfaces is demonstrated using an onboard camera as the only navigation sensor. The landing surface is a hand-made wooden podium, about 2 feet high with a landing surface of 2 feet by 2 feet, making it roughly 9 times the area of the quadrotor landing gear, and is similar to the perching setup in Bayraktar and Feron (2008), although without the use of velcro. The angle of the podium landing surface is 30 degrees and is covered with a half inch layer of high-density foam, to provide some friction for landing, as well as for protection of the quadrotor. The location of the landing pad is known a priori, and a trajectory is generated ahead of time. To accomplish this, the path to the landing pad, and the landing pad itself, were pre-mapped for location determination and to prevent navigation delays due to mapping updates. An example of the feature map generated for this maneuver is shown in Fig. 10.

Initially, perching on a string using a hook underneath the quadrotor was considered, but was determined to be unfeasible due to two reasons: point landing on the string would cause the quadrotor to tip over and fall to the ground; and landing on the string would prevent the quadrotor from taking off again, which could be attempted from an inclined surface.

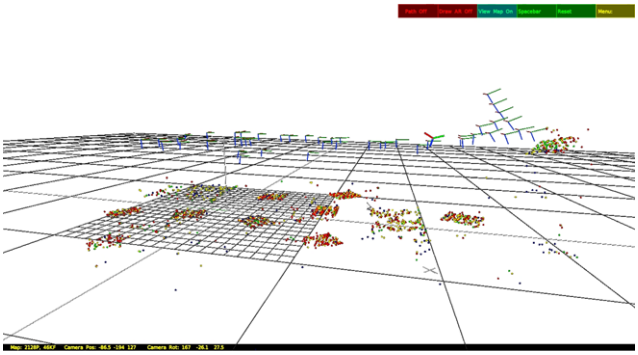


Fig. 10 Map example of the features making up the landing pad and the path

Table 5 Perching nonlinear controller gains

Gain Term	Notation	Value
Navigation y Prop. Asymptote Bound	$A_{kp}^{nav,y}$	200
Navigation y Prop. Growth Rate	$B_{kp}^{nav,y}$	0.013
Navigation y Deriv. Asymptote Bound	$A_{kd}^{nav,y}$	360
Navigation y Deriv. Growth Rate	$B_{kd}^{nav,y}$	0.0128
Navigation y Integral Gain	$ki_{sig}^{nav,y}$	0
Navigation y Desired Velocity	$(\dot{y})_{nav}^{des}$	350 cm/s

The controller used is the nonlinear saturation controller described in Sect. 4.6. This controller is used due to the noise within the tracking measurements from the navigation system (relying on a camera) from the rapid dynamic transitions needed for this maneuver. The only modification to the nonlinear saturation controller needed is specifically allowing more control authority to the navigation system, for allowing an attitude change near 30° in order to obtain high speed flight. To achieve this, the saturation asymptotes are increased to an estimated and then empirically tuned value, based upon the needed desired angle for the speeds being achieved. The gains and saturation levels used for the maneuver different from Table 4 are shown in Table 5. Only gains on the y axis were changed since perching was performed on this axis.

The path generated for aggressive landing is characterized by high desired speeds for the initial fast translational movement followed by a transition to a zero desired speed at the point at which the maneuver must take place, so as to reach the landing pad at nearly zero velocity and a high angle of attack. As the desired speed is increased, to values above 300 cm/s, and with a distance gain, k_d near 4 cm, over a translational distance to the landing pad of about 3 meters, the quadrotor actually begins to overpass the way points in the path due to its rapidly increasing speed from the initial commands. This is actually used advantageously for this maneuver, for as the crossing point is reached, the

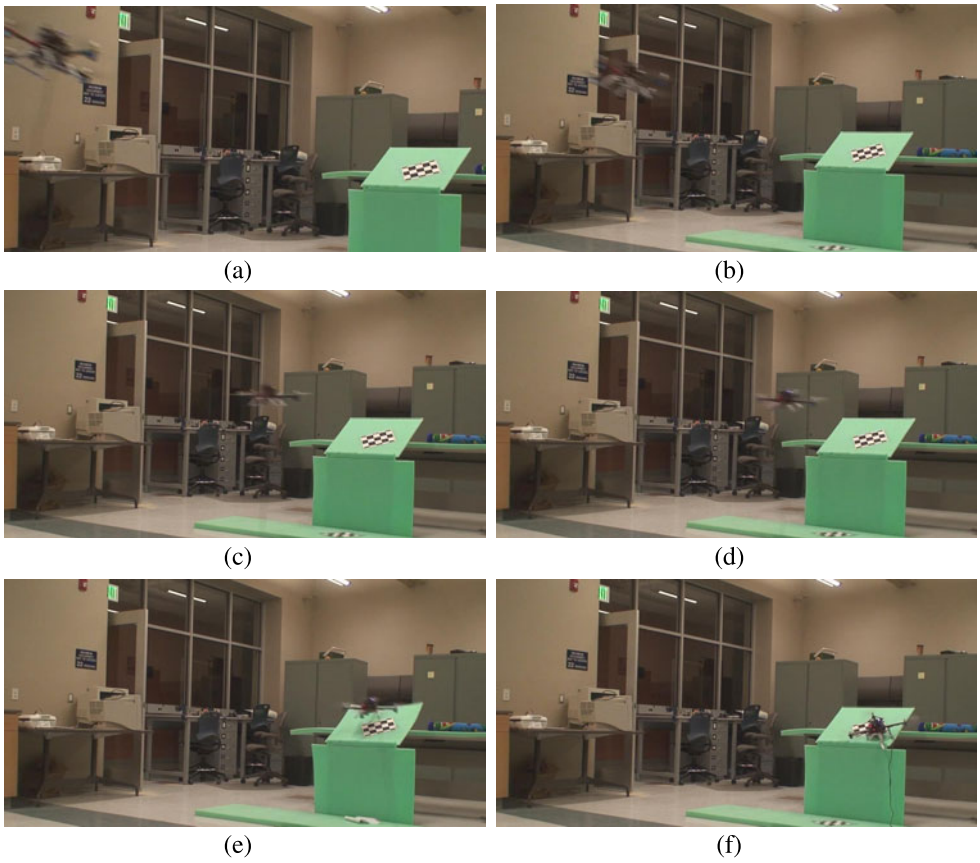


Fig. 11 Perching maneuver action shot sequence

command values from the controller are reduced over a short period of time until they become reversed. This then causes a rapidly increasing controller command in the opposite direction of travel, initializing the post-stall maneuver. A simple state transition is used upon reaching the landing pad, at which point the thrust is rapidly decreased and then turned off. During this entire maneuver, the system relies entirely on visual height control to maintain altitude, as using a sonar would report the incorrect height approaching the landing pad leading to an increase in thrust. The quadrotor was able to successfully land on a 30 degree inclined surface, achieving angles greater than 10 degrees on the high-speed translational portion, and angles up to 15 degrees for the post-stall maneuver. Translational speeds of 3 m/s were achieved.

Although the angles and speeds are not large, the indoor testing area was limited to only 4 meters, restricting the speeds, and thus the angles, of the maneuver. It is expected that these slower speeds allow for adequate trajectory tracking even with linearized models, as aerodynamic effects such as blade flapping and vehicle drag start becoming significant at velocities slightly higher than the ones achieved (Huang et al. 2009). A sequence of action shots of the maneuver are shown in Fig. 11. In the video of this maneuver, the quadrotor exhibits buffeting behavior in the horizontal plane normal to the maneuver. This is due to the effects of the camera cable hanging from one location on the quadrotor, which act as a significant disturbance during such a high speed flight.

6 Gripping

Unlike fixed wing MAVs that are incapable of driving their velocity to zero, quadrotors are ideally suited to the task of aerial manipulation or grasping. 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 and then have some means of sensing and interacting with it.

The design of the object sensing methods and gripper 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 offboard 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. The design of the gripper is highly dependent upon the structure of the platform and the space constraints.

6.1 Object sensor

A camera extracted from a Nintendo WiiMote is used, owing to its low cost, light weight, low power consumption, and specialized sensing capability. The camera consists of a 1024×768 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 pixel positions of the IR light sources, or blobs, along with the estimated blob size, as a value ranging from one to six. These measurements can be obtained at a rate of up to 200 Hz.

The camera has two ways with which it can be interfaced: Bluetooth and I^2C. Due to latency concerns from using bluetooth, the camera is interfaced directly, over I^2C. We built a board that houses the camera and supporting components,[1] shown in Fig. 3.

The camera can detect IR blobs up to a distance of 5 m and has a field of view (FOV) of 41 degrees horizontal and 31 degrees vertical. 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, with a gain of 255 experimentally found to provide the optimal performance for our implementation.

6.2 Gripper

The gripper is shown in a closed position, separate from the vehicle, in Fig. 12.

The key application specific requirements for the gripper are:

- Compliance: Due the limited positional accuracy achievable using quadrotors, a gripper is needed that is capable of manipulation under uncertainty.
- 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.
- Light-Weight: The limited payload capacity and already high energy costs drive this requirement.

[1]Camera I^2C interfacing are discussed by Kako on his website (http://www.kako.com).

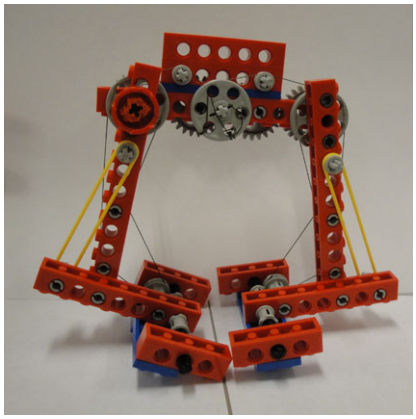


Fig. 12 Gripper showing the cables, pulleys and elastic bands used to provide compliance and under-actuation

- Minimal actuation: For similar reasons, an under-actuated system reduces the need for more motors, alleviating power and weight constraints.
- 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 Design and integration

A custom gripper is designed owing to the unavailability of grippers that satisfied the criteria listed above, especially the ability to flatten. In general, conventional robot grippers require high precision, which is not feasible for a hovering quadrotor. Initially, a very basic pincer-type single-joint gripper was constructed, which proved unworkable due to the need for compliance in order to allow insensitivity to uncertainty. After a couple iterations, an under-actuated gripper with two-link fingers was designed. A string-pulley system is installed with the string running along the length of the finger to allow both passive compliance and under-actuation, with one motor being able to actuate both the joints. The string-pulley system additionally improved the force transmission to the tips of the gripper. A rubber band is placed to keep the gripper open in the unactuated state and additionally produce a gripping action that is more smooth, accurate, and compliant. Two "finger" attachments were rigidly attached offset to the main "finger" to handle uncertainty in the direction perpendicular to the plane of gripper motion. This gripper design 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 foundation of the design is inspired from the gripper presented in Dollar and Howe

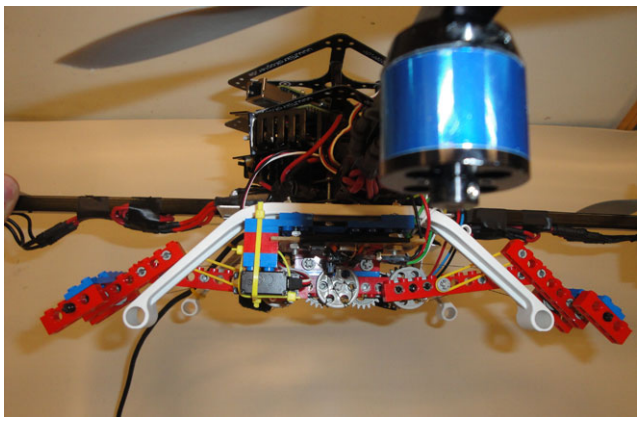


Fig. 13 Gripper shown mounted under the quadrotor

(2006). Owing to the unavailability of other options, Lego components were the choice for rapid-prototyping construction.

The final design is shown in Fig. 12. It uses a combination of pulleys and elastic-bands to achieve under-actuation, compliance, insensitivity to positional inaccuracy and is capable of grasping objects up to 7.5 cm wide. The gripper is mounted vertically underneath the quadrotor, as shown in Fig. 13. Unfortunately, the way this gripper is used, the quadrotor cannot land with an object enclosed in the gripper; it must be dropped first.

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 and can be unobtrusively placed along with the object to be gripped.

The sonar altitude sensor described in Sect. 2 cannot be used during gripping maneuvers, since it will see a shorter distance when it is over the object. Instead, altitude is controlled only based off of the navigation system, as mentioned in Sect. 4.

6.5 Control system architecture

A third outer control loop is added that can be viewed as a guidance loop, shown in Fig. 7, 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 it causing rapid changes in the desired position. Since the translational dynamics of the quadrotor are relatively slow, an integrator alone is sufficient for convergence to the desired position. Moreover, the gripping control loop is only activated once the LED/object is detected. At the time of detection, the quadrotor is roughly positioned over the object and needs only a gentle nudge to accurately position itself such that the object is in the center of the gripper. Therefore, the gripping loop only needs to send small commands to the navigation loop and for this purpose an integrator seemed to be a good choice. The outputs of the integrator loop,

$$x_{\text{offset}} = k^x_{i,IR} \cdot \int_0^t \left(x^{des}_{IR} - x_{IR}\right) \tag{47}$$

$$y_{\text{offset}} = k^y_{i,IR} \cdot \int_0^t \left(y^{des}_{IR} - y_{IR}\right), \tag{48}$$

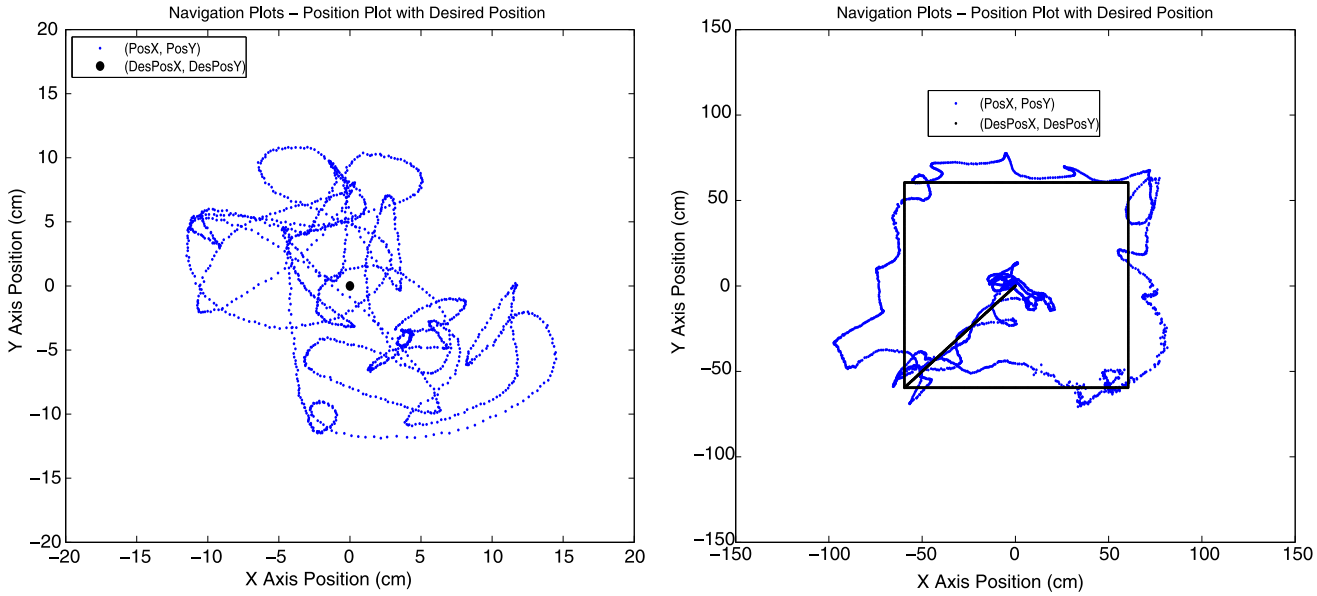
are the offsets for the desired positions of the navigation loop, with $k^x_{i,IR}, k^y_{i,IR}$ being the integrator gains.

In order to reach within gripping distance, a change in altitude of the quadrotor is performed using the blob size measurement from the IR camera. With our gain settings, a blob size of one indicates over 1 m away, while the closest measurable blob size is six, which is very close to the 5 cm minimum detectable distance from the camera. A proportional height change is utilized for any blob size other than five, as long as the measured blob position is within a centered area of the camera. In case the camera loses sight of the IR blob, the quadrotor stops descending and tries to relocate the light source. At a blob size of at least five, the gripper is activated and the quadrotor returns to the original height and hovers.

The control complexities involved with gripping, including the disturbance effects from grasping and carrying an object are handled effectively using the cascaded PID control structure. A PID loop for positional control and a PD loop for attitude stabilization is found to be robust enough to deal with the disturbances caused by gripping. These disturbances are mainly due to aerodynamic effects and the contact forces of the object acting on the quadrotor.

7 Experimental results

This section gives a selection of results showcasing different aspects of the quadrotor's capabilities, along with some images during flight. The results presented include reference angle tracking of the attitude system, navigation



(a) Hover flight. (b) Square path flight.

Fig. 14 Coordinate plots for hover and square path flights

accuracy measurements from a typical hover and square path, vision-based height performance, behavior during disturbances, and aerial gripping. Note that the measurements of x, y and z are close to metric units but not quite true metric units as they are dependent on the initial calibration procedure for the SLAM algorithm. Every effort has been made to make sure that these are close to metric units. All these experiments can be seen in the video in the Electronic Supplementary Material (or at: http://www.youtube.com/watch?v=2NKb8jhpY2M&hd=1).

7.1 Hovering and way-point navigation

A graph showing the x and y measured positions versus the desired position together on a coordinate plot is shown in Fig. 14(a). The approximated position error on a hover for the x axis is ± 13 cm, and ± 11 cm for the y axis. The y axis has a slightly tighter error due to the differences between the roll and pitch axes of the physical quadrotor. The performance of the navigation system during a hover is indicated in Fig. 15. These graphs show the x and y components regulation, the performance of altitude regulation using the sonar, the velocity regulation for all three axes, plus the performance of the attitude system in tracking the desired roll and pitch angles generated by the navigation system, as well as the behavior of the yaw, between the camera measurement, the gyroscope measurement, and the Kalman filtered estimation based on the two. The performance of the attitude controller during a hover for tracking the desired angles passed in from the navigation system demonstrates a responsive attitude system. Altitude regulation using the sonar had an error of ± 6 cm.

Performance for a square path is shown in Figs. 14(b) and 16, showing the same graphs mentioned above for the hover, except that altitude tracking is performed using just the vision-based z measurement. The approximated position error during the path for the x axis is ± 25 cm, and ± 15 cm for the y axis. The vision-based altitude performance shows an error of ± 7 cm.

7.2 Disturbance rejection

The vision-based navigation system and attitude controller are, in addition to being capable of accurate regulation and tracking, also very robust to disturbances. This robustness is demonstrated by recovery of the quadrotor after hitting one axis of the quadrotor with a stick as well as pulling the camera cable, displacing the quadrotor by over half a meter. Figure 17(a) shows an image of the quadrotor being hit by a stick, and Fig. 17(b) shows the quadrotor being pulled by the cable. The response of the system to this disturbance is indicated in the graphs of Fig. 17(c) for the x and y positions during the cord pulling disturbance and the recovery.

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 3 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. 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.

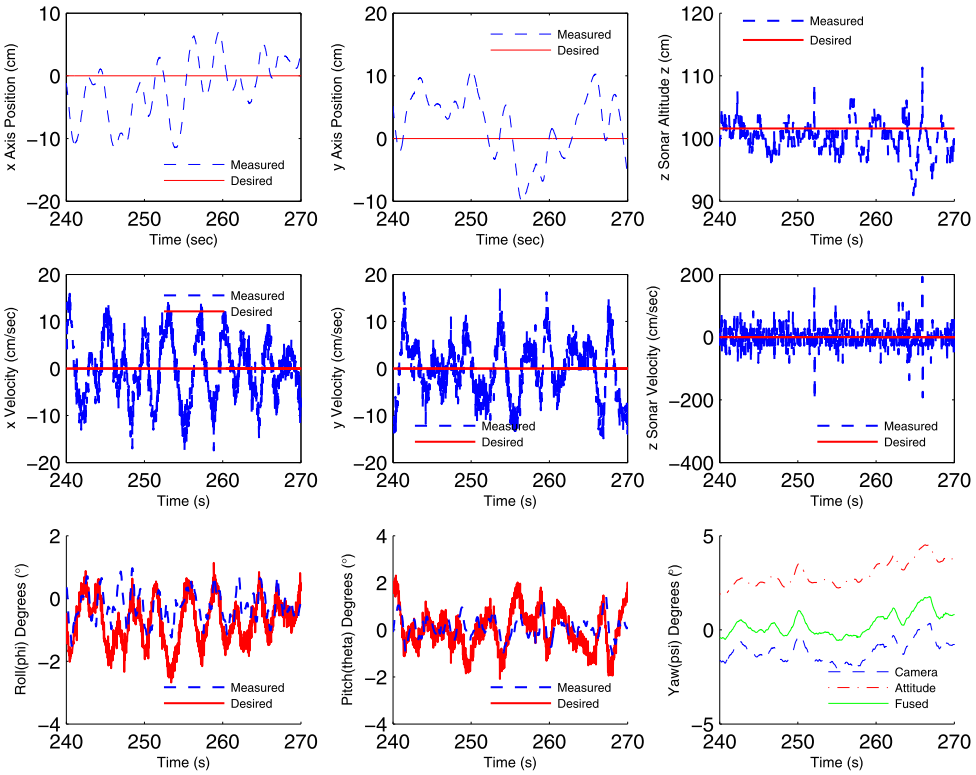


Fig. 15 Flight data for hover, using sonar to control altitude. The *top three graphs* are the x, y, and z axis positions (both measured and reference), the *middle three graphs* are the x, y, and z axis velocities (both measured and reference), and the *bottom three graphs* are the roll (ϕ) and pitch (θ) angles (showing the desired angle generated by the navigation system, and the measured angle from the attitude estimation system) and the yaw (ψ) angle (showing the measured value from the camera and attitude gyroscope, and the Kalman filtered combination of the two)

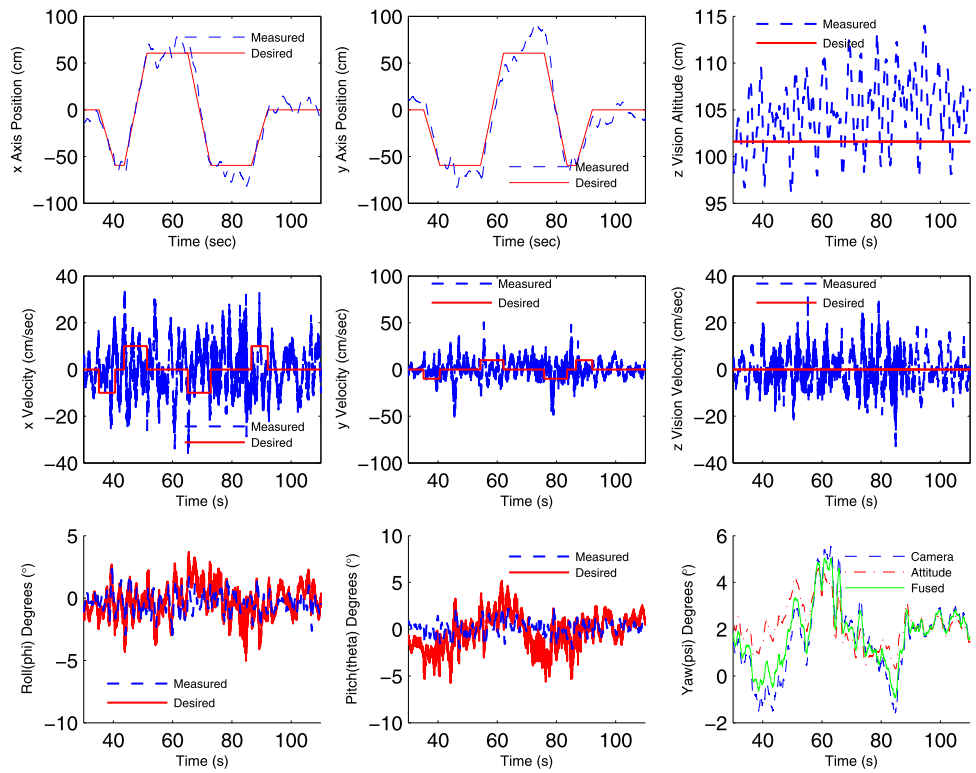
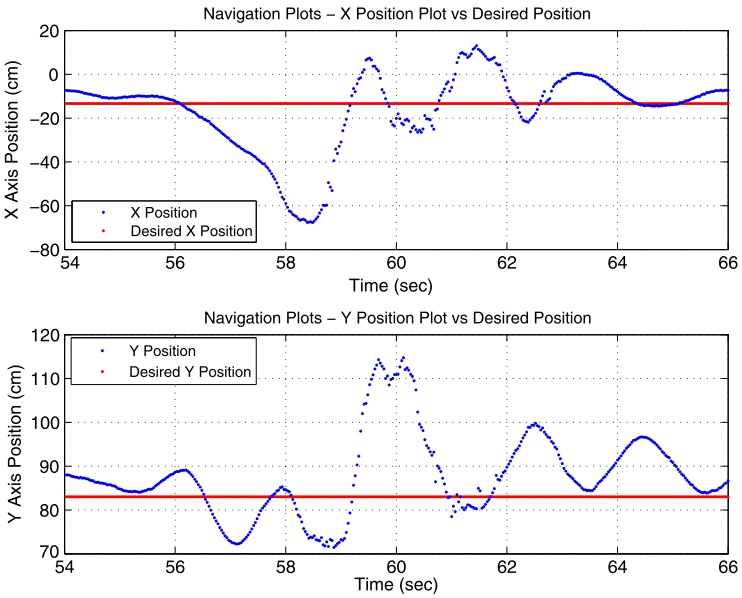


Fig. 16 Flight data for square path, using vision to control altitude. The *top three graphs* are the x, y, and z axis positions (both measured and reference), the *middle three graphs* are the x, y, and z axis velocities (both measured and reference), and the *bottom three graphs* are the roll (ϕ) and pitch (θ) angles (showing the desired angle generated by the navigation system, and the measured angle from the attitude estimation system) and the yaw (ψ) angle (showing the measured value from the camera and attitude gyroscope, and the Kalman filtered combination of the two)

The quadrotor was able to keep a hover within ± 20 cm after lowering the gains as can be seen in Fig. 18. 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

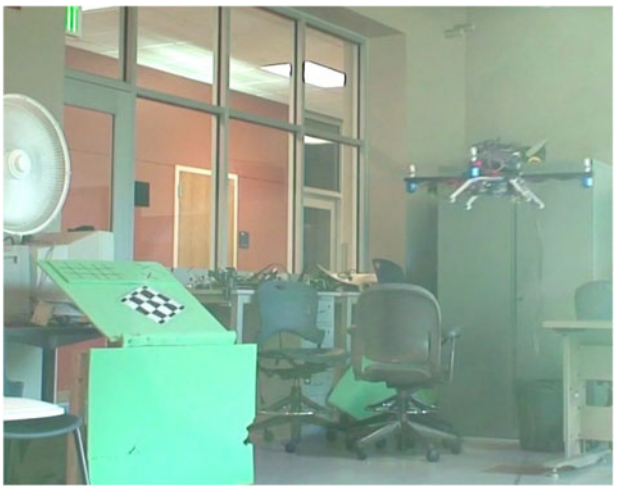
(a) Quadrotor being hit by a stick

(b) Quadrotor with cable being pulled



(c) Positions of x and y during disturbance and recovery

Fig. 17 Disturbance rejection pictures and graph



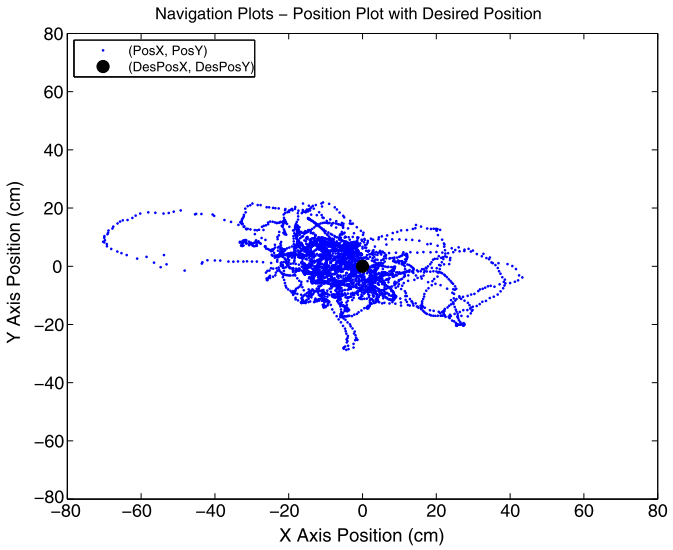


Fig. 18 Quadrotor hovering in presence of a blowing fan and a coordinate plot of the hover

flight. Another experiment had the fan blowing in the diagonal direction, and as expected, both the x and y axis behaved roughly the same as if individually they each had a fan blowing on their axis. Figure 19 shows flight data for the quadrotor hovering in the presence of a blowing fan. The quadrotor has also been flown outdoors in slight windy conditions, however, the wind speed was not measured. It should be noted that under certain circumstances such as large disturbances, the saturation used in the controller could have detrimental effects and a standard PID could work better by allowing higher control authority in order for the quadrotor to recover from these large disturbances. In our setting and implementation, we observed the saturation-based PID to work better and more consistently due to reasons outlined in Sects. 4.6 and 4.7.

7.3 Outdoor navigation

The quadrotor was flown outdoors to demonstrate the capabilities of the system in outdoor GPS-denied environments

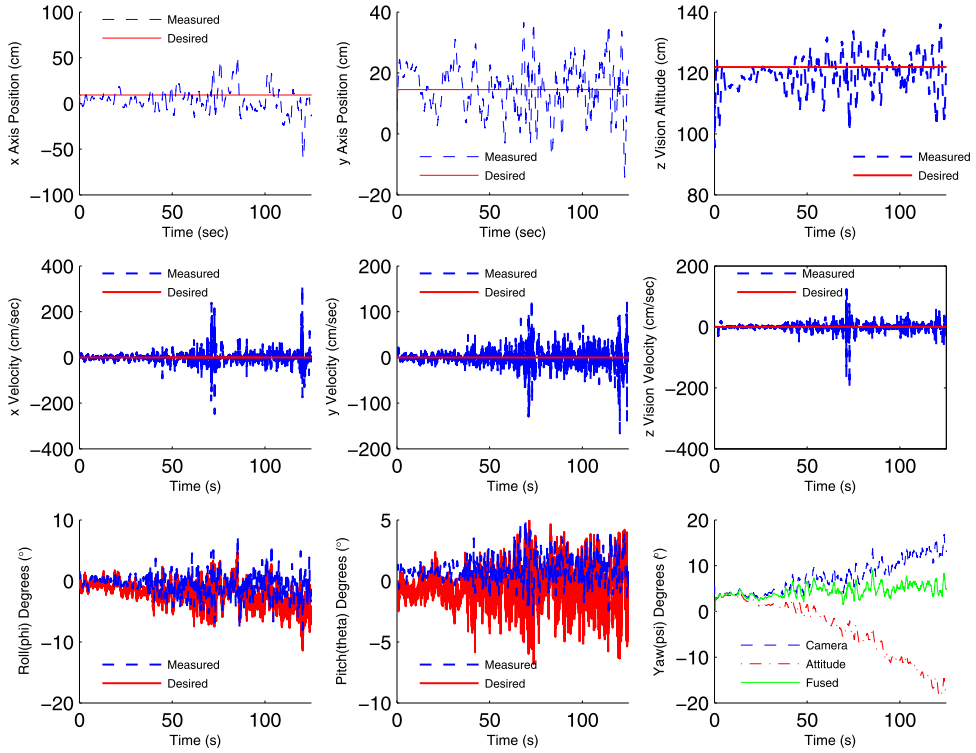


Fig. 19 Flight data graphs for the quadrotor hovering in the presence of a blowing fan with altitude control using vision. In this flight, the first 20 s the fan is off, the second 20 s the fan is on level 1 (1.7 m/s windspeed), the third 20 s the fan is on level 2 (2.2 m/s windspeed), the fourth 20 s the fan is on level 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. The *top three graphs* are the x, y, and z axis positions (both measured and reference), the *middle three graphs* are the x, y, and z axis velocities (both measured and reference), and the *bottom three graphs* are the roll (ϕ) and pitch (θ) angles (showing the desired angle generated by the navigation system, and the measured angle from the attitude estimation system) and the yaw (ψ) angle (showing the measured value from the camera and attitude gyroscope, and the Kalman filtered combination of the two)

such as urban canyons. Successful flights of the quadrotor hovering and following waypoints over grass were achieved. Figure 20 shows an image of the quadrotor in flight outside. Grass being somewhat self-similar does not aid in global localization, therefore, the quadrotor took-off on the pavement and was then commanded via waypoints to fly over grass. While successful flights were achieved, this experience also exposed the various pitfalls of the system.

– 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 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.
– 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

Fig. 20 Quadrotor hovering outdoors

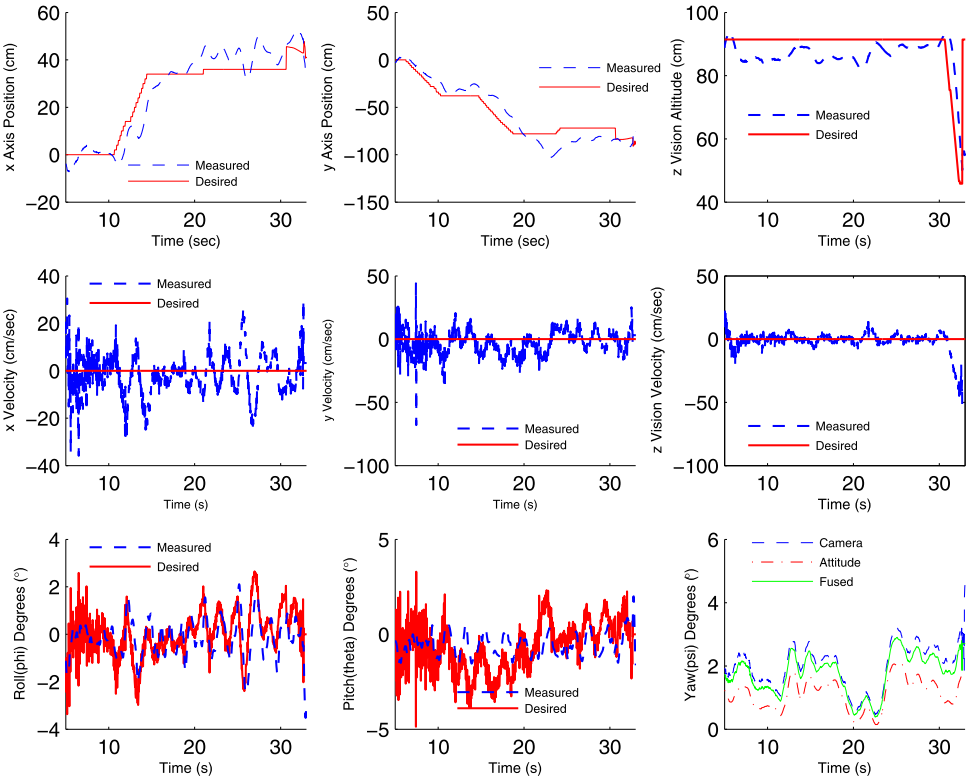


Fig. 21 Flight data for flight with gripping. Way-points were given incrementally to the quadrotor until gripping was activated over the object. The *top three graphs* are the x, y, and z axis positions (both measured and reference), the *middle three graphs* are the x, y, and z axis velocities (both measured and reference), and the *bottom three graphs* are the roll (ϕ) and pitch (θ) angles (showing the desired angle generated by the navigation system, and the measured angle from the attitude estimation system) and the yaw (ψ) angle (showing the measured value from the camera and attitude gyroscope, and the Kalman filtered combination of the two)

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

7.4 Gripping

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 successful grip. Figure 21 shows the navigation flight data for the flight and Fig. 22 shows a coordinate plot indicating the movement of the quadrotor to the object. With the onset of gripping, Fig. 23 shows the desired offsets in x, y and z, and Fig. 24 shows the change in desired position along with the actual position of the quadrotor on the map. Figure 25 shows the x and y pixel positions of the IR blob (as seen by the IR camera), and Fig. 26 shows these same positions as a 3D plot with time on the vertical axis, and the detected blob size shown below. An

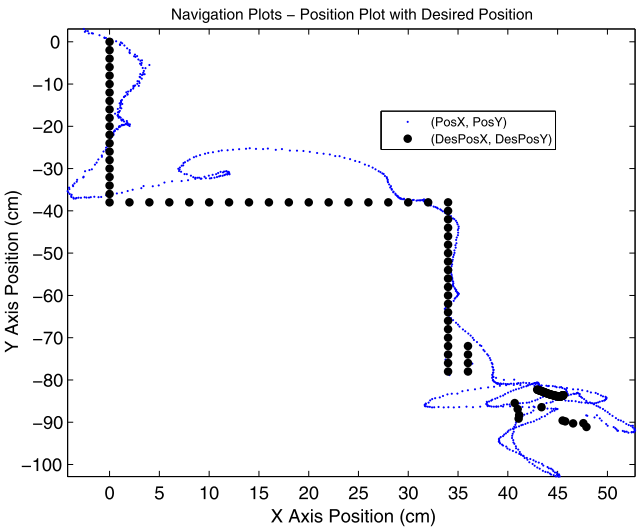


Fig. 22 Coordinate plot showing the quadrotor navigating to the object. Takeoff occurs in the *upper left quadrant* of the graph, with gripping occuring in the *lower right*. Way-points were given incrementally to the quadrotor until gripping was activated over the object

action shot sequence of the gripping maneuver is shown in Fig. 27.

There are several factors that we experimented with that affect 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.

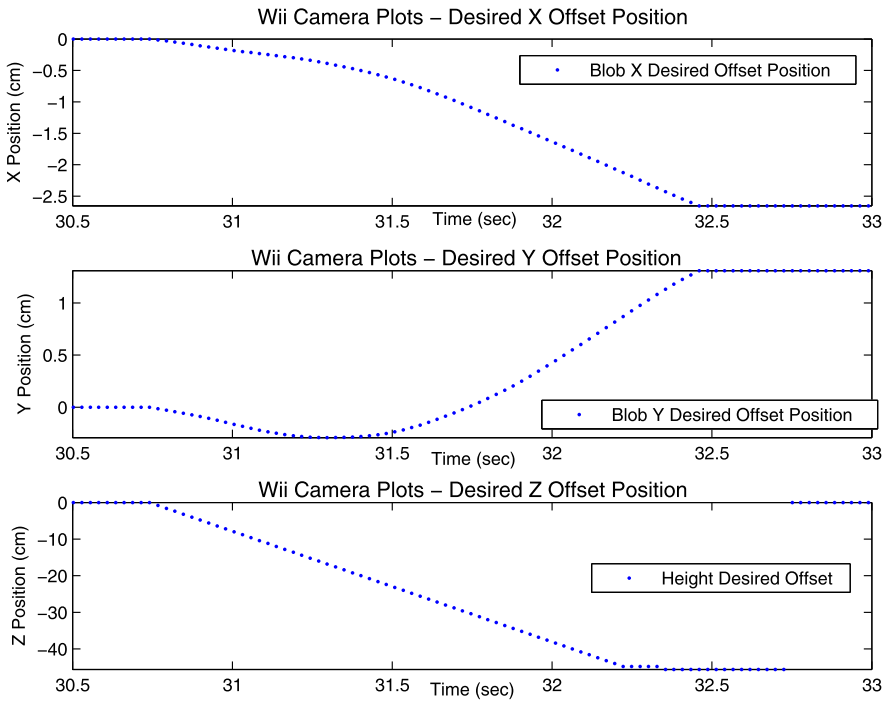


Fig. 23 These plots show desired offsets in x, y and z (in cm) from the hover position to the blob and are commanded by the outer most loop to the navigation controller using an integral controller (47, 48). The blob is detected by the IR camera at 30.7 sec, whereupon the offsets begin generating commands 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

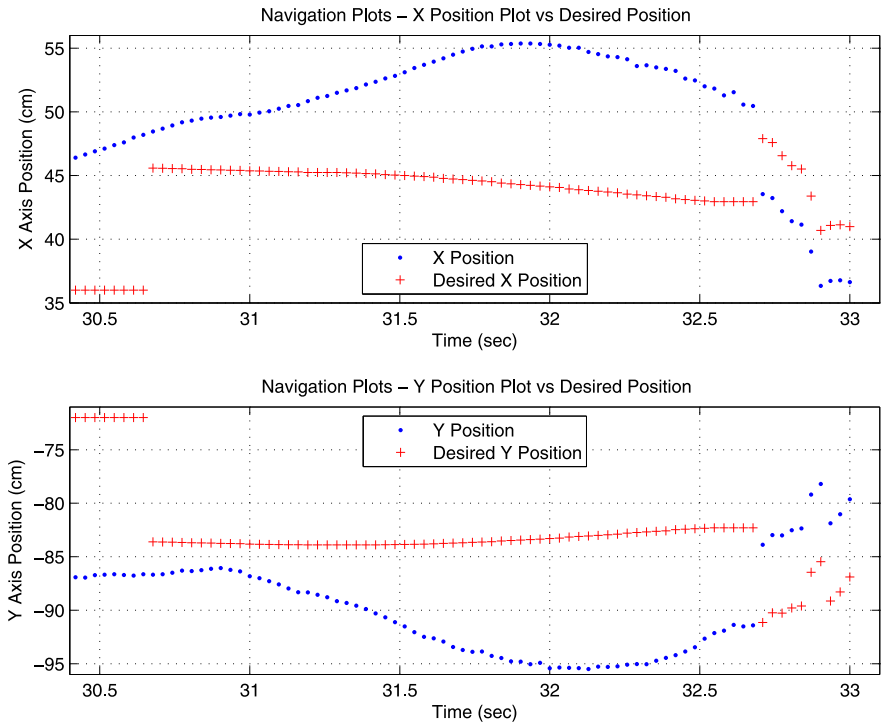


Fig. 24 Desired and actual positions of the quadrotor as seen by the navigation controller, in x and y. The blob is seen by the IR camera at 30.7 sec, at which point the desired positions (in *red*) are adjusted by the desired offsets, and gripping is activated at about 32.4 sec. After gripping, the navigation controller returns to a hover mode (Color figure online)

– 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.

– 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.

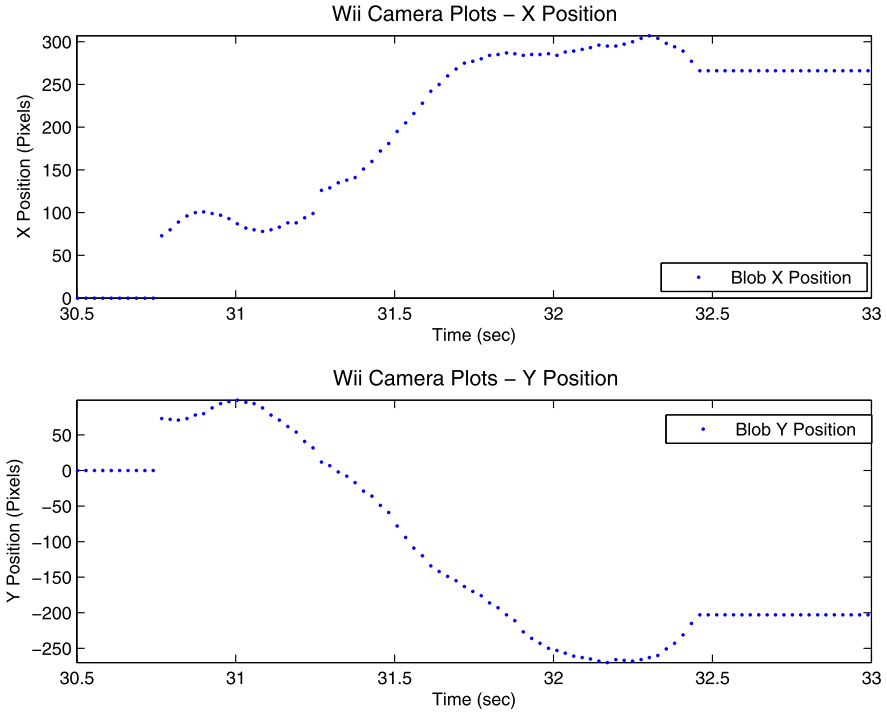


Fig. 25 The x and y pixel positions of the IR blob while maneuvering to center over the blob and gripping

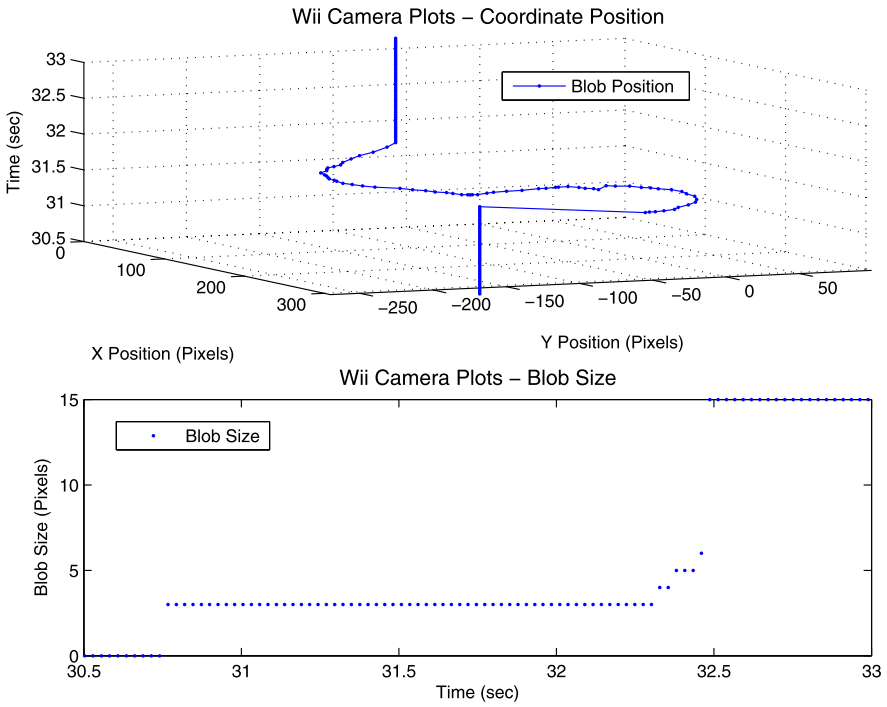


Fig. 26 Blob pixel position vs. time (*top*) and blob size (*bottom*) as seen by the IR camera. The blob is seen by the IR camera at 30.7 sec, and once a blob size of 5 is detected within a restricted pixel area, gripping is activated at 32.4 sec, shortly after which the outermost controller is deactivated and so the blob size goes to 15

– 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.

– 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. Our success in gripping lies with effectively manag-

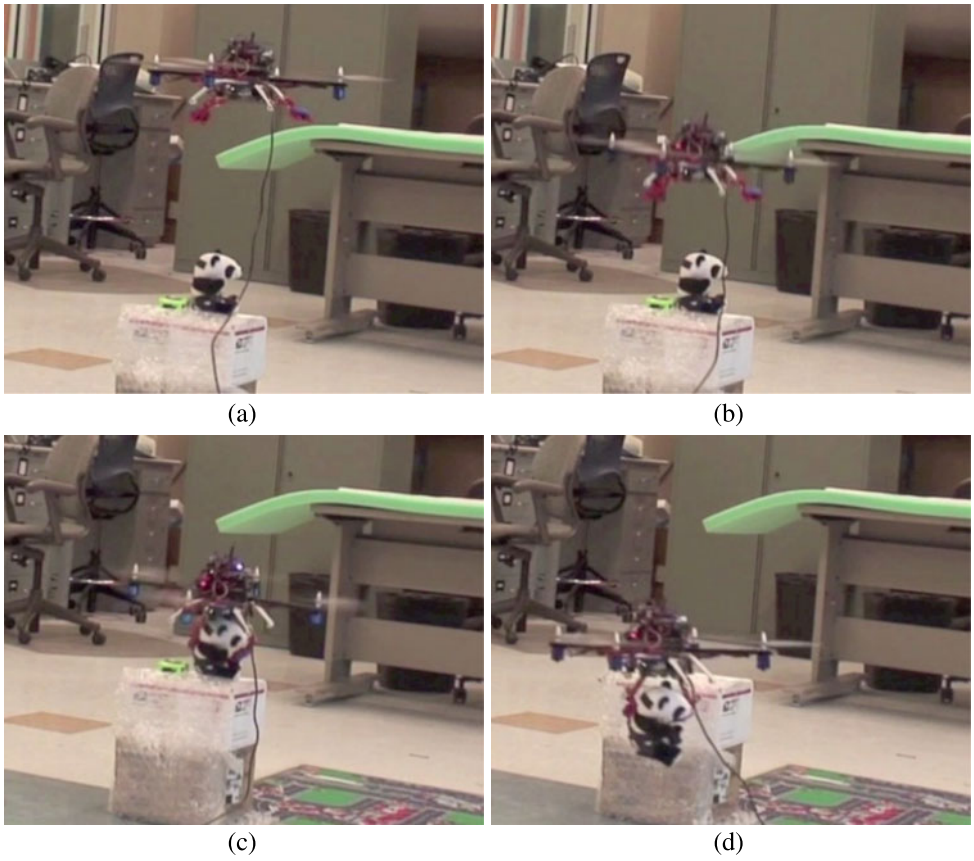


Fig. 27 Action shots of the quadrotor gripping a stuffed toy

ing the possible issues described above. Gripping is kept simple by means of just an extra sensor and an 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.

8 Conclusions and future works

8.1 Conclusions

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 operate in both outdoor and indoor environments. A nonlinear controller using model-based control enables robust path tracking performance and aggressive maneuvers such as precision landing on inclined surfaces, using only onboard sensing. 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.

8.2 Limitations and future work

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. The autonomy of the quadrotor could be further enhanced by implementing online path-planning, especially taking into account the dynamics of the quadrotor. Also, the work on aggressive precision landing on inclined surfaces is being extended to include the sensing of the landing pad.

One of the key limitations in gripping 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 3-D point cloud for identifying a variety of objects. Additionally, the object is assumed to be stationary for gripping from a hover position. Devising new controllers to account

for moving objects and large disturbances would further enhance the capabilities of the system.

References

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

Ahn, K., & Thanh, T. (2005). Nonlinear pid control to improve the control performance of the pneumatic artificial muscle manipulator using neural network. *Journal of Mechanical Science and Technology*, *19*, 106–115.

Altuğ, E., Ostrowski, J. P., & Taylor, C. J. (2005). Control of a quadrotor helicopter using dual camera visual feedback. *The International Journal of Robotics Research*, *24*, 329–341.

Amidi, O., Kanade, T., & Miller, J. R. (1998). Vision-based autonomous helicopter research at Carnegie Mellon Robotics Institute 1991–1997. In *American helicopter society international conference*.

Bayraktar, S., & Feron, E. (2008). Experiments with small helicopter automated landings at unusual attitudes. arXiv:0709.1744.

Blösch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. (2010). Vision based MAV navigation in unknown and unstructured environments. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 21–28).

Bouabdallah, S., & Siegwart, R. (2007). Full control of a quadrotor. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 153–158).

Bouabdallah, S., Noth, A., & Siegwart, R. (2004). 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)* (pp. 2451–2456).

Bouguet, J. Y. (1999). *Pyramidal implementation of the Lucas Kanade feature tracker* (Tech. Rep.), Intel Corp.

Bourquardez, O., Mahony, R., Guenard, N., Chaumette, F., Hamel, T., & Eck, L. (2009). Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, *25*(3), 743–749.

Castillo, P., Lozano, R., & Dzul, A. (2005). Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, *25*(6), 45–55.

Celik, K., Chung, S. J., Clausman, M., & Somani, A. (2009). Monocular vision SLAM for indoor aerial vehicles. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1566–1573).

Cory, R. E. (2010). Supermaneuverable perching. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Davison, A., Reid, I., Molton, N., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067.

Desbiens, AL, Asbeck, A. T., & Cutkosky, M. R. (2011). Landing, perching and taking off from vertical surfaces. *The International Journal of Robotics Research*, *30*(3), 355–370.

Dollar, A., & Howe, R. (2006). A robust compliant grasper via shape deposition manufacturing. *IEEE/ASME Transactions on Mechatronics*, *11*(2), 154–161.

Fowers, S., Lee, D., Tippetts, B., Lillywhite, K., Dennis, A., & Archibald, J. (2007). 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).

Gentili, L., Naldi, R., & Marconi, L. (2008). Modeling and control of VTOL UAVs interacting with the environment. In *Proceedings of the IEEE conference on decision and control (CDC)* (pp. 1231–1236).

Grzonka, S., Grisetti, G., & Burgard, W. (2009). Towards a navigation system for autonomous indoor flying. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 2878–2883).

Guenard, N., Hamel, T., & Moreau, V. (2005). Dynamic modeling and intuitive control strategy for an "x4-flyer". In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 141–146).

Guenard, N., Hamel, T., & Mahony, R. (2008). A practical visual servo control for an unmanned aerial vehicle. *IEEE Transactions on Robotics*, *24*(2), 331–340.

Gurdan, D., Stumpf, J., Achtelik, M., Doth, K., Hirzinger, G., & Rus, D. (2007). 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).

He, R., Prentice, S., & Roy, N. (2008). 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).

Hoffmann, G. M., Huang, H., Waslander, S. L., & Tomlin, C. J. (2007). Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proceedings of the AIAA guidance, navigation, and control conference (GNC)*.

Huang, H., Hoffmann, G., Waslander, S., & Tomlin, C. (2009). Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 3277–3282).

Kendoul, F., Nonami, K., Fantoni, I., & Lozano, R. (2009). An adaptive vision-based autopilot for mini flying machines guidance, navigation and control. *Autonomous Robots*, *27*, 165–188.

Klein, G., & Murray, D. (2007). 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).

Klein, G., & Murray, D. (2008). Improving the agility of keyframe-based SLAM. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 802–815).

Kuntz, N. R., & Oh, P. Y. (2008). 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 (pp. 731–738).

Lindsey, Q., Mellinger, D., & Kumar, V. (2011). Construction of cubic structures with quadrotor teams. In *Proceedings of the robotics: science and systems conference (RSS)*, Los Angeles, CA, USA.

Lourakis, MIA, & Argyros, A. A. (2009). SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, *36*(2), 1–30.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Mahony, R., Hamel, T., & Pflimlin, J. (2008). Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, *53*(5), 1203–1218.

Mellinger, D., Michael, N., & Kumar, V. (2010). Trajectory generation and control for precise aggressive maneuvers with quadrotors. In *Proceedings of the IFRR international symposium on experimental robotics (ISER)*.

Michael, N., Fink, J., & Kumar, V. (2009). Cooperative manipulation and transportation with aerial robots. In *Proceedings of the robotics: science and systems conference (RSS)*.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Pro-*

ceedings of the international conference on artificial intelligence (IJCAI) (pp. 1151–1156).

Moore, J., & Tedrake, R. (2009). Powerline perching with a fixed-wing UAV. In *Proceedings of the AIAA infotech@aerospace conference*.

Moore, J. M. (1994). *A system for landing an autonomous radio controlled helicopter on sloped terrain*. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Oh, S. R., Pathak, K., Agrawal, S., Pota, H., & Garratt, M. (2006). Approaches for a tether-guided landing of an autonomous helicopter. *IEEE Transactions on Robotics*, *22*(3), 536–544.

Parrot, S. A. (2011) AR.Drone. http://ardrone.parrot.com.

Pounds, P., & Dollar, A. (2010a). Aerial grasping from a helicopter UAV platform. In *Proceedings of the IFRR international symposium on experimental robotics (ISER)*

Pounds, P., & Dollar, A. (2010b). Hovering stability of helicopters with elastic constraints. In *Proceedings of the ASME dynamic systems and control conference (DSCC)*.

Pounds, P., Mahony, R., & Corke, P. (2010). Modelling and control of a large quadrotor robot. *Control Engineering Practice*, *18*(7), 691–699. Special Issue on Aerial Robotics.

Pounds, P.E.I., Bersak, D.R., & Dollar, A.M. (2011). Grasping from the air: Hovering capture and load stability. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Romero, H., Benosman, R., & Lozano, R. (2006). Stabilization and location of a four rotor helicopter applying vision. In *Proceedings of the AACC american control conference (ACC)*.

Romero, H., Salazar, S., & Lozano, R. (2009). Real-time stabilization of an eight-rotor UAV using optical flow. *IEEE Transactions on Robotics*, *25*(4), 809–817.

Rudol, P., Wzorek, M., & Doherty, P. (2010). 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).

Soundararaj, S., Sujeeth, A., & Saxena, A. (2009). 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).

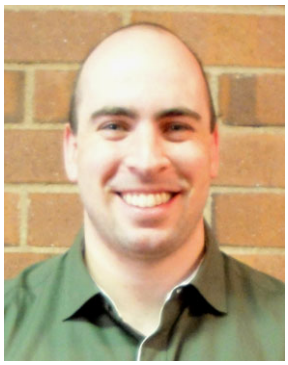
Strasdat, H., Montiel, J., & Davison, A. (2010). Real-time monocular SLAM: Why filter? In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 2657–2664).

Tournier, G. P., Valenti, M., & How, J. P. (2006). 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).

Valenti, M., Bethke, B., Fiore, G., & How, J. P. (2006). Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In *Proceedings of the AIAA guidance, navigation, and control conference (GNC)*.

Vicon (2011) Vicon MX Systems. http://www.vicon.com/products/viconmx.html.

Vaibhav Ghadiok received the B.S. degree in Electrical Engineering from RVCE, VTU, India, in 2007 and the M.S. degree in Electrical Engineering from Utah State University, USA, in 2011. He is currently pursuing his Ph.D. at the University of California, Riverside in the area of controls. His research interests are in perception, mapping, and control for Unmanned Aerial Vehicles (UAVs) and multi-agent systems.



Jeremy Goldin received the B.S. degree in Electrical Engineering from Johns Hopkins University, Maryland, in 2004, and the M.S. degree in Electrical Engineering from Utah State University in 2011. He was employed by the United States Air Force for 5 years as the brake control systems engineer for Landing Gear. He has recently transitioned to the Electronic Systems Center at Hanscom Air Force Base. His research interests are in control and planning for unmanned aerial vehicles.

Wei Ren received the B.S. degree in electrical engineering from Hohai University, China, in 1997, the M.S. degree in mechatronics from Tongji University, China, in 2000, and the Ph.D. degree in electrical engineering from Brigham Young University, Provo, UT, in 2004. From October 2004 to July 2005, he was a Postdoctoral Research Associate with the Department of Aerospace Engineering, University of Maryland, College Park, MD. He was an assistant professor (August 2005 to June 2010) and an associate professor (July 2010 to June 2011) with the Department of Electrical and Computer Engineering, Utah State University, Logan. Since July 2011, he has been with the Department of Electrical Engineering, University of California, Riverside, where he is currently an Associate Professor. His research focuses on distributed control of multi-agent systems and networked control systems. He is an author of two books: Distributed Coordination of Multi-agent Networks (Springer-Verlag, 2011) and Distributed Consensus in Multi-vehicle Cooperative Control (Springer-Verlag, 2008). Dr. Ren was the recipient of a National Science Foundation CAREER award in 2008. He is currently an Associate Editor for Automatica, and Systems and Control Letters and an Associate Editor on the IEEE Control Systems Society Conference Editorial Board.