Autonomous Indoor Aerial Gripping Using a Quadrotor

Vaibhav Ghadiok, Jeremy Goldin and Wei Ren

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

1. INTRODUCTION

The past couple of years has seen a focus on navigation in indoor GPS-denied environments for MAVs. In the absence of global positioning, a MAV needs to rely on onboard sensing modalities such as laser range finders or cameras in order to determine its position [1], [2], [3]. 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, radio towers, etc, with potential applications in object retrieval and improved observation through manipulating barriers, objects or switches. Applications also include deployment and retrieval of sensor nodes in a sensor network.

Aerial manipulation on a MAV has seen very little published research. Some theoretical results on aerial vehicles interacting with the environment are presented in [4], while some experimental results of cooperative manipulation using cables are presented in [5]. In [6], [7], 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 [8]. Additionally, some work has been done involving gripping with quadrotors in [9], 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.

Unlike fixed wing UAVs 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.

Our approach to tackle the above outlined challenges is to utilize a highly capable navigation system, a simple object sensing setup, and a gripper designed for grasping even under positional inaccuracy. Fig. 1 shows the quadrotor and the gripper attached underneath it. The quadrotor is built with specific restrictions on the system, including requirements for low-cost components. As such, a custom platform is designed specifically utilizing off-the-shelf components, which necessitated certain adaptations to the attitude stabilization and navigation control systems. A monocular camera is used for vision-based navigation using a sophisticated simultaneous localization and mapping (SLAM) algorithm. A secondary camera capable of detecting infrared (IR) light
sources is used to sense the object, which is equipped with a small IR LED and battery pack. The design of an under-actuated, passively compliant gripper capable of gripping despite positional uncertainty is developed, allowing completely autonomous aerial gripping.

This paper presents our system in II, III and IV, covering the base platform, navigation setup and gripping system. We conclude with experimental results which demonstrate the autonomous gripping capabilities of the quadrotor and a discussion of the limitations and scope for future work.

II. QUADROTOR BASE PLATFORM

The complete quadrotor system is developed from scratch with low-cost constraints in mind. The attitude control system uses a PD controller and the gains are tuned empirically.

A. Hardware Architecture

The quadrotor is custom-made from available consumer-grade components. Fig. 2 gives a breakdown of the primary parts of the system and how they communicate with each other. 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 I2C 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 camera transmits images through a USB cable linked to the ground station, which consists of an Intel Core 2 CPU 2x2.4GHz processor running Ubuntu.

The quadrotor weighs 1.4 kg and measures 50 cm from end to end of the frame. 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.

B. Dynamic Model

Modeling the quadrotor as a rigid-body, 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. Then the model is:

\begin{align}
\dot{\xi} & = \nu \\
\dot{\nu} & = ge_z - \frac{1}{m} T e_z \\
T & = b \sum_{i=1}^{4} \omega_i^2
\end{align}

where 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 $\nu = [v_x \ v_y \ v_z]^T$ represents the linear velocity of the origin of $\mathcal{B}$, expressed in the inertial frame, and $e_z = [0 \ 0 \ 1]^T$ is the unit vector in the inertial frame, $\mathcal{F}$; $g$ is the acceleration from gravity and $m$ 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, $\phi, \theta$ and $\psi$ 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. $\omega_i$ is the speed of the rotors, $i \in \{1, 2, 3, 4\}$. The dynamic model continues

\begin{align}
\dot{R} & = R \cdot sk(\Omega) \\
I_f \dot{\Omega} & = -\Omega \times I_f \Omega - G_a + \tau_a \\
I_i \omega_i & = \tau_i - Q_i, \ i \in \{1, 2, 3, 4\} \\
Q_i & = k(\omega_i^2) \\
G_a & = \sum_{i=1}^{4} I_i (\Omega \times e_z)(-1)^{i+1} \omega_i
\end{align}

where $\Omega$ 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, $\mathcal{B}$. $I_i$ signifies the moment of inertia of the rotor blades and $Q_i$ 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. $\tau_a$ is the airframe torque generated by the rotor, while $\tau_i$ represents the 4 control inputs to the system, in the form of motor torques.

C. Attitude Stabilization

The IMU provides angular rates and accelerations, which are read by the Gumstix over I2C at a rate of 400 Hz. The measurements are low pass filtered before being read by the Gumstix, using the individual sensors’ built-in customizable filters. The sensor data is further FIR low pass filtered on the Gumstix using very short filter lengths to minimize delay. Such heavy filtering is required due to the extensive noise coupling into the sensors, as well as the general low quality of the components used.

Angles extracted from the acceleration values are then fused with the integrated gyro rates using a nonlinear passive
complementary filter, developed in [10]. This filter is found to perform significantly better than the Kalman filter for this application. The yaw angle is obtained by integrating the gyro after being bias adjusted using the complementary filter.

1) Attitude Controller: Attitude is stabilized using a PD controller where the roll, pitch and yaw control values, \( u_\phi \), \( u_\theta \) and \( u_\psi \), are governed by

\[
\begin{align*}
  u_\phi &= k_{p,\phi} \cdot (\phi^{des} - \phi) + k_{d,\phi} \cdot (\dot{\phi}^{des} - \dot{\phi}) \\
  u_\theta &= k_{p,\theta} \cdot (\theta^{des} - \theta) + k_{d,\theta} \cdot (\dot{\theta}^{des} - \dot{\theta}) \\
  u_\psi &= k_{p,\psi} \cdot (\psi^{des} - \psi) + k_{d,\psi} \cdot (\dot{\psi}^{des} - \dot{\psi}),
\end{align*}
\]

where the superscript, \( \text{des} \), 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.

A rigid body model, described in II, 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, the model takes the form

\[
\begin{align*}
  \dot{\phi}^{des} &= \frac{4k_f L \omega_b}{I_{xx}} (u_\phi k_{rpm} + c_{rpm}), \\
  \dot{\theta}^{des} &= \frac{4k_f L \omega_b}{I_{yy}} (u_\theta k_{rpm} + c_{rpm}), \\
  \dot{\psi}^{des} &= \frac{8k_M \omega_b}{I_{zz}} (u_\psi k_{rpm} + c_{rpm}),
\end{align*}
\]

where \( L = 23.2 \text{ cm} \) is the distance from the axis of rotation of the rotors to the center of the quadrotor. \( k_{rpm} \) converts the \( u_\phi, u_\theta, u_\psi \) PD commands to RPMs, as they are in terms of PWM values and is determined to be equal to 10 when around the hover thrust region. An offset, \( c_{rpm} \), is needed to match the nominal RPM with the nominal PWM. The values, \( u_\phi, u_\theta, u_\psi \), are determined using (9,10,11) and the terms, \( I_{xx}, I_{yy}, I_{zz} \), are the diagonal elements of the inertia matrix.

2) Altitude Controller: Altitude is measured using a downward-facing sonar, which is sampled at 20 Hz. The sampled data is run through a median filter to reject outliers. A PID controller (15), along with a feedforward term, is found to compensate effectively for the nonlinear effects involved in the altitude dynamics of the quadrotor, using

\[
\begin{align*}
  u_{alt} &= k_{p,alt} \cdot (z^{des} - z) + k_{d,alt} \cdot \int_0^t (z^{des} - z) dt \\
  &\quad + k_{i,alt} \cdot (z^{des} - z) + u_{nom} \tag{15}
\end{align*}
\]

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

\[
  z = (\cos \phi \cdot \cos \theta) \cdot z_{sonar}. \tag{16}
\]

In (15), the velocity, \( \dot{z} \), is obtained by finite differentiation of the input sonar measurements.

3) System Control: The controller outputs, \( u_\phi, u_\theta, u_\psi \) and \( u_{alt} \), are inputs to the ESCs in terms of PWM values, and they are converted to individual motor commands using

\[
\begin{bmatrix}
  u_{i,\text{PWM}}^1 \\
  u_{i,\text{PWM}}^2 \\
  u_{i,\text{PWM}}^3 \\
  u_{i,\text{PWM}}^4
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & -1 & 1 \\
  1 & 1 & 0 & -1 \\
  0 & 1 & 1 & 1 \\
  -1 & 0 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
  u_{alt} \\
  u_\phi \\
  u_\theta \\
  u_\psi
\end{bmatrix}, \tag{17}
\]

where \( u_{i,\text{PWM}} \) denotes the output PWM value to motor \( i \). The \( u_{i,\text{PWM}} \) outputs to each motor are related to the airframe torque of the dynamic model (5), with

\[
\begin{align*}
  \omega_i &= k_{rpm} \cdot u_{i,\text{PWM}} + c_{rpm}, \quad i \in \{1,2,3,4\} \tag{18} \\
  \tau_a &= (\tau_a^1, \tau_a^2, \tau_a^3)^T \tag{19} \\
  \tau_a^1 &= L \cdot b (\omega_3^2 - \omega_2^2) \tag{20} \\
  \tau_a^2 &= L \cdot b (\omega_1^2 - \omega_3^2) \tag{21} \\
  \tau_a^3 &= k (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2). \tag{22}
\end{align*}
\]

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, shown in Fig. 3.
III. NAVIGATION

The navigation system uses a sophisticated visual SLAM algorithm to obtain position and heading measurements from an onboard, downward-facing monocular camera. The algorithm runs offboard on the ground station, which sends the measurements to the quadrotor system using Zigbee. A PID controller is implemented onboard for tracking the desired references.

A. Overview of the Vision-Based SLAM Algorithm

The outlined approach uses the visual SLAM algorithm developed in [11]. A similar approach has been used on a commercial quadrotor platform in [3]. This SLAM algorithm is highly capable of tracking and mapping from a single camera, and the motion model employed copes well with 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 [11]. 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 [12] before moving to and mapping new locations.

B. Controller Design

The navigation and attitude controllers are hierarchically separated as part of an integrated, cascaded design. The attitude and altitude controllers take in the desired angular tilt, $\phi$ and $\theta$, and heading, $\psi$, references as well as the desired height, $z^{\text{des}}$, with the navigation controller determining the desired references using the ground station SLAM output, and passing the references directly on to the attitude controller.

The navigation controller runs asynchronously at an order of magnitude smaller rate than the attitude controller, and operates only when positioning information is received from the ground station. 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.

The navigation system utilizes a PID controller for obtaining the desired accelerations prior to determining desired angular references, with the controller outputs determined by

\[
x^{\text{des}} = k^x_p \cdot (x^{\text{des}} - x) + k^x_i \cdot \left( \int_0^t (x^{\text{des}} - x) dt \right) + k^x_d \cdot \dot{x}^{\text{des}}
\]

\[
y^{\text{des}} = k^y_p \cdot (y^{\text{des}} - y) + k^y_i \cdot \left( \int_0^t (y^{\text{des}} - y) dt \right) + k^y_d \cdot \dot{y}^{\text{des}},
\]

Fig. 3. 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.
where the superscript, \( \text{des} \), refers to the desired position and velocity references. \( k^*_{p, \text{nav}}, k^*_{i, \text{nav}} \) and \( k^*_{d, \text{nav}} \) are the proportional, integral and derivative navigation control gains. Here, the outputs of the PID controllers are desired accelerations, based on linearization of (2) 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.

\[
\begin{bmatrix}
\theta^\text{des} \\
\psi^\text{des}
\end{bmatrix} = \begin{bmatrix}
1/s & -\cos \psi \\
\cos \psi & \sin \psi
\end{bmatrix} \begin{bmatrix}
x^\text{des} \\
y^\text{des}
\end{bmatrix},
\]  
(25)

These references are then sent to the attitude loop using (9, 10). The yaw angle, \( \psi \), is determined from fusing an appropriately delayed value of the gyro integrated yaw with the yaw determined from the SLAM algorithm, using a Kalman filter.

The \( x^\text{des} \) and \( y^\text{des} \) inputs to (23, 24), as well as \( \psi^\text{des} \) in (11), are generated based upon the desired path or trajectory to carry out.

The \( z \) value obtained from the SLAM algorithm can be directly used for visual altitude control, using a PID controller similar to the one described in II.

The current system uses a wired onboard camera that sends the camera images over USB for onboard processing. Offboard computation is chosen for rapid-prototyping simplicity, although the SLAM algorithm is quite capable of being implemented and 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 accurately stabilize and navigate.

The navigation system forms the intermediate control loop and is shown in Fig. 3.

IV. OBJECT DETECTION AND GRIPPING

The design of the object sensing method 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.

A. 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 x 768 pixels Charged Coupled Device (CCD) sensor and a custom system-on-a-chip that is capable of tracking up to four IR light sources simultaneously. It reports the \( x \) and \( y \) 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\(^2\)C. Due to latency concerns from using Bluetooth, the camera is interfaced directly, over I\(^2\)C. We built a board that houses the camera and supporting components, shown in Fig. 2.

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\(^2\)C. The gain parameter is related to the sensitivity of the camera, with a gain of 255 experimentally found to provide the best performance.

B. Gripper

The gripper is shown in a closed position, separate from the vehicle, in Fig. 4. 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.
- 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.

\(^1\)Camera \( \text{I}^2\text{C} \) interfacing are discussed by Kako on his website (http://www.kako.com)
C. 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. An under-actuated gripper with passive mechanical compliance is employed, leading to insensitivity to positional uncertainty. This provides the benefits of minimal actuation and no gripper-based sensing requirements, unlike other complicated control schemes as force control, which provide active compliance. The design is inspired from the gripper presented in [13], and is constructed using Lego for rapid prototyping; it is shown in Fig. 4. It uses a combination of pulleys and elastic-bands to achieve under-actuation and compliance, and is capable of grasping objects up to 7.5 cm wide. The gripper is mounted vertically underneath the quadrotor, as shown in Fig. 1.

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.

D. 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 II cannot be used during gripping maneuvers, since it will see a shorter distance when it is over the object. Instead, altitude is controlled based off of the navigation system, as mentioned in III.

E. Control System Architecture

A third outer control loop is added that can be viewed as a guidance loop, shown in Fig. 3, which changes the desired position of the quadrotor based upon the measured position of the IR blob. A feedback loop is required due to the quantized and noisy blob location data returned by the camera and the lack of true relative positioning information. Initially a PD loop was tested for changing the desired position, but gave poor results due to rapid changes in the desired position. Since the translational dynamics of the quadrotor are relatively slow, an integrator alone is sufficient for convergence to the desired position in $x$ and $y$. The outputs of the integrator loop,

$$x_{\text{offset}} = k_{x,IR}^i \int_0^t (x_{\text{IR}}^{des} - x_{\text{IR}}) \quad (26)$$

$$y_{\text{offset}} = k_{y,IR}^i \int_0^t (y_{\text{IR}}^{des} - y_{\text{IR}}), \quad (27)$$

are the offsets for the desired positions of the navigation loop, with $k_{x,IR}^i$, $k_{y,IR}^i$ being the integrator gains.

Fig. 5. These plots show desired offsets in $x$, $y$, and $z$ (in cm) from the hover position to the blob and are commanded by the outer-most loop to the navigation loop using an integral controller (26, 27). 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.

In order to reach within vertical 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 a blob size smaller than five, as long as the measured 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 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 are 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.

V. EXPERIMENTAL RESULTS

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. Fig. 5 shows the desired offsets in $x$, $y$, and $z$, and Fig. 6 shows the change in desired position along with the actual position of the quadrotor on the map, during a gripping maneuver. Fig. 7 shows the $x$ and $y$ positions of the IR blob (as seen by the IR camera), and Fig. 8 shows the same positions as a 3D plot with time on the vertical axis, and the detected blob size shown below. An action shot sequence of the gripping maneuver is shown in Fig. 9, and a video of the quadrotor gripping a stuffed toy is available at:

http://www.youtube.com/watch?v=pjTMZNRunSw

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we presented the gripping of an object using a quadrotor. The position of the quadrotor, and generated map, are determined using an onboard monocular camera. This allows the quadrotor to navigate to the location of the object without needing any prior information of the environment. Utilizing model-based control, the desired position is effectively tracked. A gripper is designed specifically to address effective object grasping even under position uncertainty. With the careful selection and integration of these components and controllers, we have successfully built a system which is able to autonomously navigate, locate, and grasp an object, using only onboard sensors.

B. Limitations and Future Work

One of the key limitations of this work is the need for an IR light source marker to be placed with the object to be gripped. Even though the IR LED can be powered using just a button cell and is itself small and unobtrusive, our system precludes the ability to grip objects in adverse environments by not specifically addressing the perception problem. A depth camera could be used to provide a dense 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 would further enhance the capabilities of the system.

REFERENCES