Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Control of a Quadrotor Helicopter Using Visual Feedback

Erdinç Altuğ*, James P. Ostrowski*, Robert Mahony**
*GRASP Lab. University of Pennsylvania, Philadelphia, PA 19104, USA
**Dep. of Eng., Australian Nat. Uni., ACT 0200, Australia
E-mail: {erdinc, jpo}@grasp.cis.upenn.edu, mahony@faceng.anu.edu.au

Abstract

We present control methods for an autonomous four-rotor helicopter, called a quadrotor, using visual feedback as the primary sensor. The vision system uses a ground camera to estimate the pose (position and orientation) of the helicopter. Two methods of control are studied — one using a series of mode-based, feedback linearizing controllers, and the other using a backstepping-like control law. Various simulations of the model demonstrate the implementation of feedback linearization and the backstepping controllers. Finally, we present initial flight experiments where the helicopter is restricted to vertical and yaw motions.

1 Introduction

The purpose of this study is to explore the control methodologies that will make an unmanned aerial vehicle (UAV) autonomous. An autonomous UAV will be suitable for applications like search and rescue, surveillance and remote inspection. Rotary wing aerial vehicles have distinct advantages over conventional fixed wing aircrafts on surveillance and inspection tasks, since they can take-off/ land in limited spaces and easily hover above the target. A ***quadrotor*** is a four rotor helicopter. One example is shown in Figure 1. The idea of using four rotors is not new. A full-scale four-rotor helicopter was built by De Bothezat in 1921 [1]. Other examples are the Mesicopter [2] and Hoverbot [3]. Also, related models for controlling the VTOL aircraft are studied by Hauser et al [4] and in [5]. Helicopters are dynamically unstable and therefore suitable control methods are needed to make them stable. Although unstable dynamics is not desirable, it is good for agility. The instability comes from the changing helicopter parameters and the disturbances such as wind.

A quadrotor helicopter is controlled by varying the rotor speeds, thereby changing the lift forces. It is an under-actuated, dynamic vehicle with four input forces and six output coordinates. One of the advantages of using a multi-rotor helicopter is the increased payload capacity. It has more lift therefore heavier weights can be carried. Quadrotors are highly maneuverable, which enables vertical take-off/landing, as well as flying into hard to reach areas. Disadvantages are the increased helicopter weight and increased energy consumption due to the extra motors. Since it is controlled with rotor-speed changes, it is more suitable to electric motors, and large helicopter engines which have slow response may not be satisfactory without a proper gear-box system.

The main concentration of this study is using non-linear control techniques to stabilize and perform output tracking control of a helicopter. In Section 2 the helicopter model and dynamics of quad-rotor is described. The equation of motion of a simplified quadrotor is given here. Feedback linearization and backstepping controllers are described and simulation results are introduced in Section 3. Real-time control and the vision system which is responsible for pose estimation and real-time control are described in Section 4. Experiments on a real quadrotor test-bed are given in Section 5.

2 Helicopter Model

Unlike regular helicopters that have variable pitch angles, a quadrotor has fixed pitch angle rotors and the rotor speeds are controlled to produce the desired lift forces. Basic motions of a quadrotor can be described using Figure 1. Vertical motion of the helicopter can be achieved by changing all of the rotor speeds at the same time. Motion along the x-axis is related to tilt around the y-axis. This tilt can be obtained by decreasing the speeds of rotors 1 and 2 and by increasing speeds of rotors 3 and 4. This tilt also produces acceleration along the x-axis. Similarly y-motion is the result of the tilt around the x-axis. The yaw motions are obtained using the moments that are created as the rotors spin. Conventional helicopters have the tail rotor in order to balance the moments created by the main rotor. With the four-rotor case, spinning directions of the rotor are set to balance and cancel these moments. This is also used to produce the desired yaw motions. To turn in a clock-wise direction, the speeds of rotor 2 and 4 should be increased to overcome the moments created by rotors 1 and 3. A good controller should be able to reach a desired yaw angle while keeping the tilt angles and height constant.

2.1 Dynamics of Quadrotor Helicopter

A body fixed frame is assumed to be at the center of gravity of the quadrotor, where the z-axis is pointing upwards. This body axis is related to the inertial frame by a position vector (x,y,z) and 3 Euler angles, (θ,ψ,ϕ), representing pitch, roll and yaw respectively. A ZYX-Euler angle representation given in Equation 1, has been chosen for the representation of the rotations.

$$R = \begin{pmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{pmatrix}. \quad (1)$$

where c_θ and s_θ represent $\cos\theta$ and $\sin\theta$ respectively.

Each rotor produces moments as well as vertical forces. These moments have been experimentally observed to be linearly dependent on the forces for low speeds. There are four input forces and six output states (x, y, z, θ, ψ, ϕ) therefore the quadrotor is an under-actuated system. The rotation direction of two of the rotors are clockwise while the other two are counterclockwise, in order to balance the moments and produce yaw motions as needed.

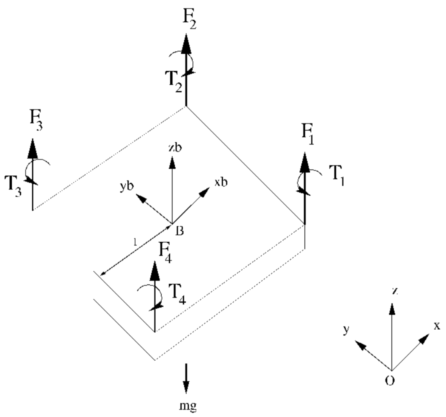

Figure 1: 3D Quadrotor Model.

The equations of motion can be written using the force and moment balance.

$$\begin{aligned}
\ddot{x} &= \frac{(\sum_{i=1}^{4} F_i)(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - K_1\dot{x}}{m} \\
\ddot{y} &= \frac{(\sum_{i=1}^{4} F_i)(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) - K_2\dot{y}}{m} \\
\ddot{z} &= \frac{(\sum_{i=1}^{4} F_i)(\cos\phi\cos\psi) - mg - K_3\dot{z}}{m} \qquad (2) \\
\ddot{\theta} &= l(-F_1 - F_2 + F_3 + F_4 - K_4\dot{\theta})/J_1 \\
\ddot{\psi} &= l(-F_1 + F_2 + F_3 - F_4 - K_5\dot{\psi})/J_2 \\
\ddot{\phi} &= (M_1 - M_2 + M_3 - M_4 - K_6\dot{\phi})/J_3
\end{aligned}$$

The K_i's given above are the drag coefficients. In the following we assume the drag is zero, since drag is negligible at low speeds. For convenience, we will define the inputs to be

$$\begin{aligned}
u_1 &= (F_1 + F_2 + F_3 + F_4)/m \\
u_2 &= (-F_1 - F_2 + F_3 + F_4)/J_1 \qquad (3) \\
u_3 &= (-F_1 + F_2 + F_3 - F_4)/J_2 \\
u_4 &= C(F_1 - F_2 + F_3 - F_4)/J_3.
\end{aligned}$$

where J_i's are the moment of inertia with respect to the axes and C is the force-to-moment scaling factor. The u_1 represents a total thrust on the body in the z-axis, u_2 and u_3 are the pitch and roll inputs and u_4 is a yawing moment. Therefore the equations of motion become

$$\begin{aligned}
\ddot{x} &= u_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) & \ddot{\theta} &= u_2 l \\
\ddot{y} &= u_1(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) & \ddot{\psi} &= u_3 l \\
\ddot{z} &= u_1(\cos\theta\cos\psi) - g & \ddot{\phi} &= u_4. \quad (4)
\end{aligned}$$

The center of gravity is assumed to be at the middle of the connecting link. As the center of gravity moves up (or down) d units, then the angular acceleration becomes less sensitive to the forces, therefore stability is increased. Stability can also be increased by tilting the rotor forces towards the center. This will decrease the roll and pitch moments as well as the total vertical thrust.

3 Control of a Quadrotor

Our goal is to use an external camera as the primary sensor and use onboard gyros to get the tilt angles and stabilize the helicopter in an inner control loop. Due to the weight limitations we can not add GPS or other accelerometers on the system. Therefore our controller should be able to get the positions and speeds from the camera only. One other aspect of the controller selection depends on the method of control of the UAV. It can be mode-based or non-mode based. For the mode based controller, independent controllers for each state are needed, and a higher level controller decides how these interact. On the other hand for a non-mode based controller, a single controller controls all of the states together. In this section we will present the implement of feedback linearization and a backstepping controller to the quadrotor model and show that it can be stabilized and controlled.

3.1 Feedback Linearization

One approach to make the quadrotor helicopter autonomous is the use of a controller that can switch between many modes such as; hover, take-off, landing, left/right, search, tilt-up, tilt-down etc. These low level control tasks can be connected to a higher level controller that sets the goal points and does the motion planning. Similarly, Koo et al. used hybrid control methodologies in [6] for autonomous helicopters. A natural starting place for this is to ask what modes can be controlled using feedback linearizing controllers.

We can use exact input-output linearization and choose outputs to be z, θ, ψ and ϕ, in order to control the altitude, yaw and tilt angles of the quadrotor. But this controller introduces zero dynamics which results in the drift of the helicopter in the x-y plane. Therefore such a controller is unstable.

The zero dynamics for this system are

$$\ddot{x} = \tan\theta \qquad \ddot{y} = -(\tan\psi/\cos\theta). \quad (5)$$

These zero dynamics are not desirable, and so another controller or a combination of controllers is needed. One can pick the outputs to be z,x,y,ϕ, which results in a complex equation with higher derivatives. Alternatively, we may pick two outputs, $z-x$, and use separate controllers for controlling ϕ and y-motion. The problem with this approach is the need of switching between controllers. To get the inputs, we differentiate the equations until the inputs appear. A fourth order derivative of the states is necessary.

$$\begin{aligned} \ddot{u}_1 &= \sin\theta(v_1 - 2\dot{\theta}\dot{u}_1\cos\theta + u_1\dot{\theta}^2\sin\theta) + \\ &\quad \cos\theta(v_2 - 2\dot{\theta}\dot{u}_1\cos\theta + u_1\dot{\theta}^2\sin\theta) \\ u_2 &= (\cos\theta(v_1 - 2\dot{\theta}\dot{u}_1\cos\theta + u_1\dot{\theta}^2\sin\theta) - \\ &\quad \sin\theta(v_2 - 2\dot{\theta}\dot{u}_1\cos\theta + u_1\dot{\theta}^2\sin\theta))/u_1, \end{aligned} \tag{6}$$

where v_1 and v_2 are given as

$$\begin{aligned} v_1 &= -K_1\ddot{x} - K_2\dot{x} - x \\ v_2 &= -K_3\ddot{z} - K_4\dot{z} - z. \end{aligned} \tag{7}$$

We can control the y-axis motion and yaw by PD controllers. Motion along the y-axis can be related to the ψ tilt angle by Equation 9. Therefore the tilt angle is selected based on the y position and the velocity:

$$\begin{aligned} u_3 &= K_{p1}(\psi_d - \psi) + K_{d1}(\dot{\psi}_d - \dot{\psi}) \\ u_4 &= K_{p2}(\phi_d - \phi) + K_{d2}(\dot{\phi}_d - \dot{\phi}), \end{aligned} \tag{8}$$

where

$$\begin{aligned} \psi_d &= \arcsin(K_p y + K_d \dot{y}) \\ \dot{\psi}_d &= \frac{K_p\dot{y} + K_d\ddot{y}}{\sqrt{1 - K_p^2 y^2 - 2K_pK_d y\dot{y} - K_d^2\dot{y}^2}} \end{aligned} \tag{9}$$

Figure 2 shows the quadrotor simulation, where it moves from (40,20,60) to the origin with initial zero yaw and tilt angles. Note the tilt-up motion of the quadrotor in order to slow down and reach the origin with zero velocity.

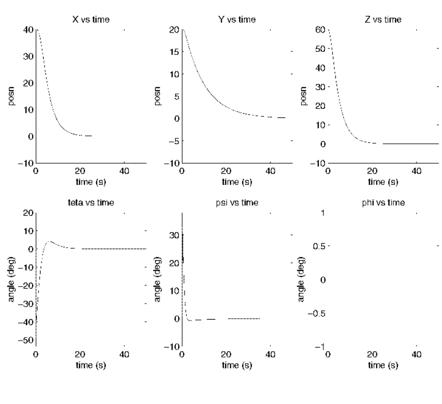

Figure 2: Feedback linearization simulation results.

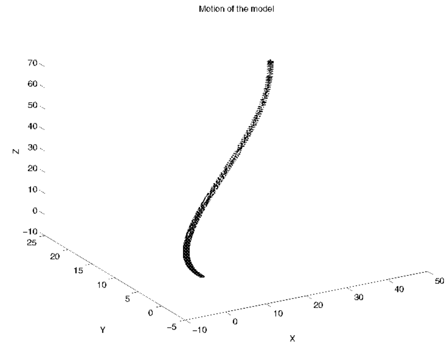

Figure 3: Feedback linearization path.

Once the inputs are obtained, one can use Equation 3 to find the forces and set the motor speeds to get the desired lift from each of the rotors.

We can put together these controllers into a hybrid controller of the form shown in Figure 4. Hover mode is the central mode, where the model is stabilized by keeping the positions (x, y, z) constant and (θ,ψ,ϕ) angles zero. The basic commands will be switched from this hover mode.

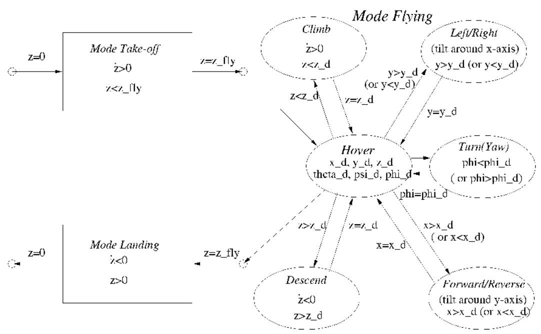

Figure 4: Hybrid Controller Model.

Climb and descend modes change the z-value, while keeping the other values constant. These modes will terminate only when the desired z-value is achieved. Left/Right mode is responsible for controlling the y-axis motions. As the model tilts around the x-axis, it will start moving on the y-axis. Forward/Reverse mode tilts around y-axis by changing the ψ angle. The hover mode will switch to the landing mode when the flight is complete.

3.2 Backstepping Controller

Backstepping controllers [7] are especially useful when some states are controlled through other states. As it was observed in the previous section, in order to control the x and y motion of the quadrotor, tilt angles need to be controlled. Therefore a backstepping controller has been developed in this section. Similar ideas of using backstepping with visual servoing have been developed for a traditional helicopter by Hamel and Mahony [8]. The approach here is a bit simpler in implementation, and relies on only very simple estimates of pose.

We will use a small angle assumption on the yaw angle, ϕ, to justify neglecting certain terms from Equation 4 to give

$$\begin{aligned} \ddot{x} &= u_1 \sin\theta \cos\psi \cos\phi \\ \ddot{y} &= -u_1 \sin\psi \cos\phi. \end{aligned} \tag{10}$$

First we notice that motion in the y-direction can be controlled through changes in the roll. This leads to a backstepping controller for $y - \psi$ control given by

$$\begin{aligned} u_3 = \frac{1}{u_1 \cos\psi \cos\phi}&(-5y - 10\dot{y} - 9u_1 \sin\psi \cos\phi \\ &-4u_1\dot{\psi} \cos\psi \cos\phi + u_1\dot{\psi}^2 \sin\psi \cos\phi \\ &+2u_1\dot{\psi} \sin\psi \sin\phi + u_1\dot{\psi}\dot{\phi} \cos\psi \sin\phi \\ &-u_1\dot{\phi}\dot{\psi} \cos\psi \sin\phi - u_1\dot{\phi}^2 \sin\psi \cos\phi). \end{aligned} \tag{11}$$

To develop a controller for motion along the x-axis, we assume the tilt ψ is slowly varying or $\cos\psi \approx const.$ This leads to a backstepping controller for $x - \theta$ of

$$\begin{aligned} u_2 = \frac{1}{u_1 \cos\theta \cos\psi \cos\phi}&(-5x - 10\dot{x} - 9u_1 \sin\theta \cos\psi \cos\phi \\ &-4u_1\dot{\theta} \cos\theta \cos\psi \cos\phi + u_1\dot{\theta}^2 \sin\theta \cos\psi \cos\phi \\ &+2u_1\dot{\phi} \sin\theta \cos\psi \sin\phi + u_1\dot{\theta}\dot{\phi} \cos\theta \cos\psi \sin\phi \\ &-u_1\dot{\phi}\dot{\theta} \cos\theta \cos\psi \sin\phi - u_1\dot{\phi}^2 \sin\theta \cos\psi \cos\phi). \end{aligned} \tag{12}$$

The altitude and the yaw on the other hand, can be controlled by a PD controller.

$$\begin{aligned} u_1 &= \frac{g + K_{p1}(z_d - z) + K_{d1}(\dot{z}_d - \dot{z})}{\cos\theta \cos\psi} \\ u_4 &= K_{p2}(\phi_d - \phi) + K_{d2}(\dot{\phi}_d - \dot{\phi}) \end{aligned} \tag{13}$$

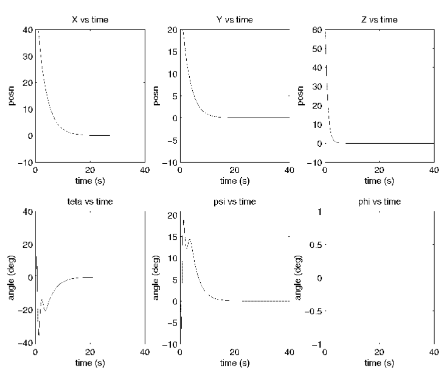

Figure 5: Backstepping controller simulation results.

The simulation results in Figures 5 and 6 show the motion of the quadrotor from position $(40, 20, 60)$ to origin. The

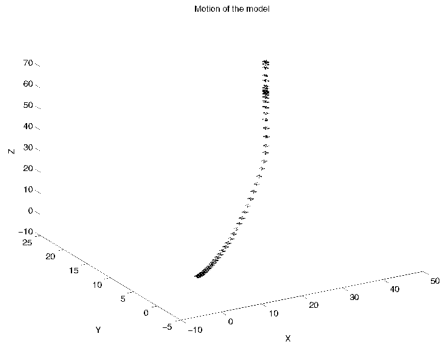

Figure 6: Backstepping controller path.

controller is strong enough to handle random errors which simulate the pose estimation errors and disturbances as shown in Figure 7. The error introduced on x and y has variance of 0.5 cm and error on z has variance of 2 cm. The yaw variance is 1.5 degrees. The helicopter moves from 100 cm to 150 cm while reducing the yaw angle from 30 degrees to zero. The mean and standard deviation are found to be 150 cm and 1.7 cm for z and 2.4 degrees and 10.1 degrees for ϕ respectively.

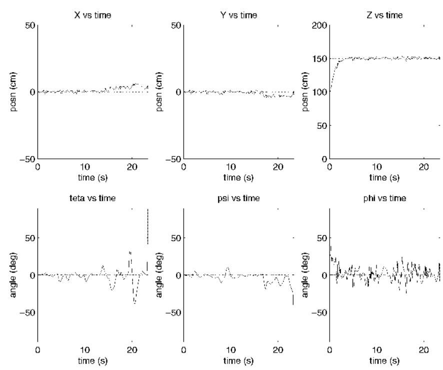

Figure 7: Backstepping controller simulation with random noise at x, y, z and yaw values.

Moving to an arbitrary heading

A backstepping controller is used in the previous section to perform tilts around the x and y axes of the helicopter, assuming the desired yaw angle is zero. Using the invariances of the system, it is straightforward to see that the same controller can be applied for an arbitrary ψ_d.

The x' and y' axes shown in Figure 8 are

$$\begin{aligned} x' &= x \cos\phi_d + y \sin\phi_d \\ y' &= -x \sin\phi_d + y \cos\phi_d. \end{aligned} \tag{14}$$

The equation of motion relative to the initial frame can be rotated to align the body frame to the desired yaw angle.

$$R_z(-\phi_d) \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = u_1 R_z(\phi - \phi_d) \begin{pmatrix} \sin\theta \cos\psi \\ -\sin\psi \end{pmatrix} \tag{15}$$

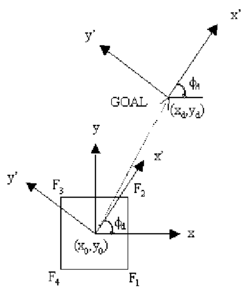

Figure 8: Moving along arbitrary direction to goal.

The backstepping controller described above will be used to perform control on the new $x' - y'$ frame and go to the desired point of that frame. Controllers used for yaw and altitude control can also be used in this part. This controller can deal with any desired yaw angle. Simulation results are given in Figure 9 for moving to origin with $\psi_d = 60^o$.

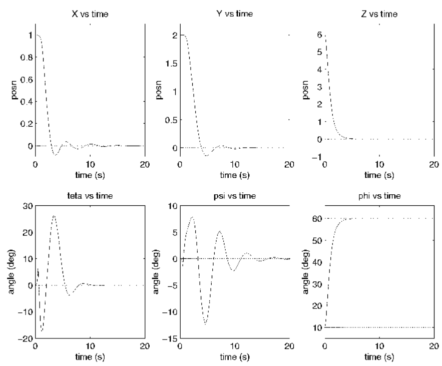

Figure 9: Simulation results for backstepping controller with arbitrary heading.

4 Real Time Control and Vision System

To make a helicopter fully autonomous, we need a flight controller as shown in Figure 10. There is an off-board controller that receives camera images, processes them and sends control inputs to the on-board processor. The on-board processor stabilizes the model by checking the gyroscopes and listens for the commands sent from the off-board controller. The rotor speeds are set accordingly to achieve the desired positions and orientations.

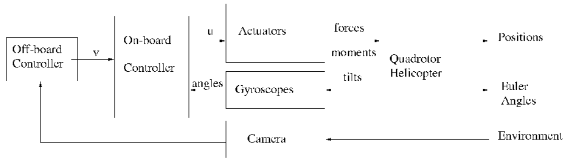

Figure 10: Control Diagram.

The off-board controller shown in Figure 11 is responsible for the main computation. It processes the images and sets the goal positions and sends them to the on-board controller via a radio link.

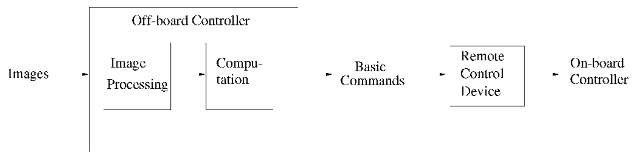

Figure 11: Off-board controller.

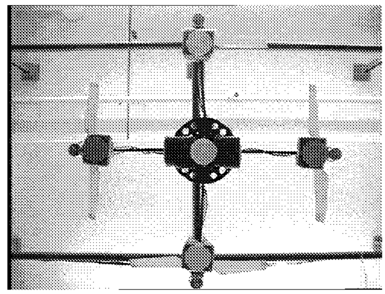

Figure 12: Quadrotor tracking with a camera.

Helicopter pose is estimated by a pose estimation algorithm. This algorithm uses 2.5 cm radius colored blobs that are attached to the bottom of the quadrotor as shown in Figure 12. A blob tracking algorithm is used to get the positions and areas of the blobs on the image plane. Then the purpose of the pose estimation algorithm is to obtain (x,y,z) positions, pitch angles (θ, ψ) and the yaw angle (ϕ) of the helicopter in real-time relative to the camera frame. The position of each blob is calculated as

$$\begin{aligned} z_i &= (f_x + f_y)\sqrt{C\pi}/(2\sqrt{A_i}) \\ x_i &= (u_i - O_x)z_i/f_x \\ y_i &= (v_i - O_y)z_i/f_y, \end{aligned} \tag{16}$$

where f_x and f_y are the focal lengths in x and y respectively, C is the number of pixels per unit area, O_x and O_y are the image center coordinates, A_i are the area of the blobs and u_i and v_i are the image coordinates.

$$\begin{aligned} \phi &= \arctan(y_1 - y_5/x_1 - x_5) \\ \psi &= \arcsin(z_4 - z_2/d) \\ \theta &= \arcsin(z_5 - z_1/d). \end{aligned} \tag{17}$$

The position of the helicopter is estimated by averaging the five blob positions. A normalization is performed using the real center difference between blobs. Yaw angle can be obtained from blob positions and the tilt angles can be estimated from the height differences of the blobs, where d is given as the distance between blobs in Equation 17. These estimates depend on area calculations, therefore they are sensitive to noise. Other methods that do not depend on the area measurements will be implemented to lower the errors in the pitch and roll angle estimation.

5 Quadrotor Experiments

The proposed controllers and the pose estimation algorithm have been implemented on a remote controlled battery powered helicopter shown in Figure 13a. It is a commercially available hobby helicopter called HMX-4. It is about 0.7 kg, 76 cm long between rotor tips and has about 3 minutes flight time. This helicopter has three gyros on board to stabilize the helicopter. There is a R/C receiver and colored blobs at the bottom. The commands can be sent from a PC to the helicopter by a remote control device that uses the parallel port of a PC.

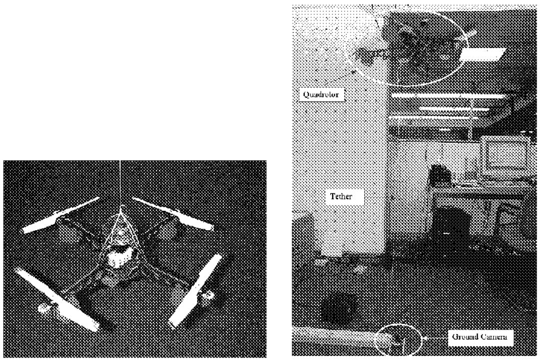

Figure 13: a) Quadrotor Helicopter, b) Experimental Setup.

An experimental setup shown in Figure 13b was prepared to prevent the helicopter from moving too much in the x-y plane, but letting it be able to turn and ascend/descend. Controllers given in Equations 11, 12 and 13 are implemented on the experiment. Figure 14 shows results of this experiment, where height, x ,y and yaw angles are being controlled. The mean and standard deviation are found to be 144 cm and 4.26 cm for z and -10.2 degrees and 11.7 degrees for ϕ respectively. The results from the plots show that the proposed controllers control the helicopter's yaw and height well despite the pose estimation errors and the errors introduced by the tethering system.This also agrees with the simulation given in Figure 7 where random noise was introduced.

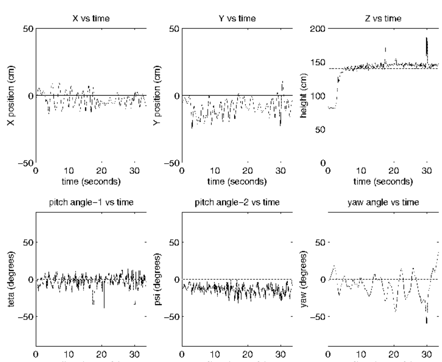

Figure 14: The results of the height x, y and yaw control experiment.

6 Conclusions and Future Work

We have presented a model of a quadrotor helicopter, and introduced several control methods. Simulations of feedback linearization and backstepping controllers were implemented and compared using Matlab Simulink. As it can be seen from the simulations the backstepping controller works better than the feedback stabilization. The height and yaw control experiment result shown in Section 5 prove the control system and the vision system's ability to perform the tasks. A helicopter can not be fully autonomous if it depends on an external camera. Therefore our next goal is to use a combination of two cameras; one on-board the helicopter and an other on the ground. This will help to decrease the errors in estimated tilt angles as will the use of other pose estimation algorithms that do not depend on area estimates. Our future interest will be to use this helicopter for ground-air cooperation tasks. Quadrotors' superior maneuverability makes it a good candidate for inspection, chase and other tasks. Another basic advantage of using such a helicopter is the increased payload, therefore cooperative manipulation of objects with flying robots can also be an interesting and challenging research direction.

7 Acknowledgements

We would like to thank John Spletzer for his help for the blob tracking algorithm. We also gratefully acknowledge support from NSF grant NSF-IIS-9876301 and DARPA ITO Grant 130-1303-4-534328-xxxx-2000-0000.

References

[1] A. Gessow, G. Myers, *Aerodynamics of the helicopter*, Frederick Ungar Publishing Co, New York, third edition, 1967.

[2] I. Kroo, F. Printz, *Mesicopter Project*, Stanford University, http://aero.stanford.edu/mesicopter.

[3] J. Borenstein, *Hoverbot Project*, University of Michigan, www-personal.engin.umich.edu/~johannb/hoverbot.htm.

[4] Hauser, Sastry, Meyer, *Nonlinear control design for slightly non-minimum phase systems: application to V/STOL Aircraft*, Automatica, vol 28, No:4, pp 665-679, 1992.

[5] Martin, Devasia, Paden, *A different look at output tracking: Control of a VTOL aircraft*, Automatica, vol 32, No:1, pp 101-107, 1996.

[6] T. J. Koo, F. Hoffmann, B.Sinopoli, S. Sastry, *Hybrid Control of An Autonomous Helicopter*,Proceedings of IFAC Workshop on Motion Control, Grenoble, France, September 1998.

[7] S. Sastry, *Nonlinear Systems; Analysis, Stability and Control*, Springer-Verlag, 1999.

[8] T. Hamel, R. Mahony, *Visual servoing of a class of under-actuated dynamic rigid-body systems*, Proceedings of the 39th IEEE Conference on Decision and Control, 2000.