Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback

Erdinç Altuğ

Istanbul Technical University
Istanbul, Turkey
altuger@itu.edu.tr

James P. Ostrowski

Evolution Robotics
Pasadena, CA, USA

Camillo J. Taylor

University of Pennsylvania
Philadelphia, PA, USA

Abstract

In this paper we propose a vision-based stabilization and output tracking control method for a model helicopter. A novel two-camera method is introduced for estimating the full six-degrees-of-freedom pose of the helicopter. One of these cameras is located on-board the helicopter, and the other camera is located on the ground. Unlike previous work, these two cameras are set to see each other. The pose estimation algorithm is compared in simulation to other methods and is shown to be less sensitive to errors on feature detection. In order to build an autonomous 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 demonstrate the implementation of these controllers. Finally, we present flight experiments where the proposed pose estimation algorithm and non-linear control techniques have been implemented on a remote-controlled helicopter.

KEY WORDS—helicopter control, pose estimation, unmanned aerial vehicle, vision-based control

1. Introduction

The purpose of this study is to explore control methodologies and pose estimation algorithms that will provide some level of autonomy to an unmanned aerial vehicle (UAV). An autonomous UAV will be suitable for applications such as search and rescue, surveillance, and remote inspection. Rotary wing aerial vehicles have distinct advantages over conventional fixed wing aircraft on surveillance and inspection tasks, since they can take-off/land in limited spaces and easily hover above any target. Moreover, rotary wing aerial vehicles (such as helicopters) are highly maneuverable making them preferable for various tasks. Unfortunately this agility makes control harder, due to the dynamical instabilities and sensitivity to disturbances.

A quadrotor (Altuğ, Ostrowski, and Mahony 2002; Altuğ, Ostrowski, and Taylor 2003), as shown in Figure 1, is a four-rotor helicopter. This helicopter design dates back to the early twentieth century, with the first full-scale four-rotor helicopter being built by De Bothezat in 1921 (Gessow and Myers 1967). Recent work in quadrotor design and control includes much smaller scale versions of quadrotors, such as the X4-Flyer (Hamel, Mahony, and Chriette 2002) and the Mesicopter (Kroo and Printz 2005). Also, related models for controlling the VTOL aircraft are studied by Hauser, Sastry, and Meyer (1992) and Martin, Devasia, and Paden (1996).

A quadrotor is an underactuated, dynamic vehicle with four input forces (basically, the thrust provided by each of the propellers) and six output coordinates (fully spatial movements). Unlike regular helicopters that have variable pitch angle rotors, a quadrotor helicopter has fixed pitch angle rotors. It can be controlled by varying the rotor speeds of all four rotors, thereby changing the lift forces. In a quadrotor, rotors are paired to spin in opposite directions in order to cancel out the resultant moment found in single-rotor helicopters. This eliminates the need for a tail rotor to stabilize the yaw motions of the quadrotor. Some of the advantages of using a multi-rotor helicopter include high maneuverability and increased payload capacity, due to the increased load area. Also, as the number of engines on an aircraft increase, the total weight of the engines (which give the same total thrust) tend to decrease (Gessow and Myers 1967). The most important disadvantage is the increased energy consumption due to the extra motors.

The International Journal of Robotics Research
Vol. 24, No. 5, May 2005, pp. 329-341,
DOI: 10.1177/0278364905053804

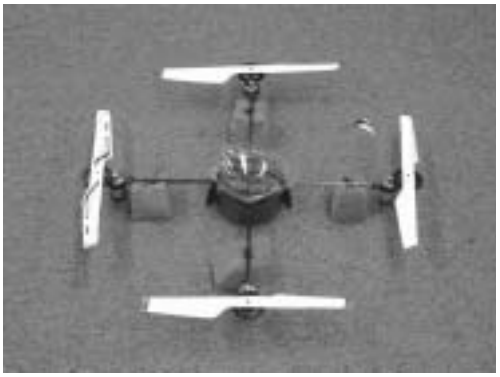

Fig. 1. Quadrotor helicopter.

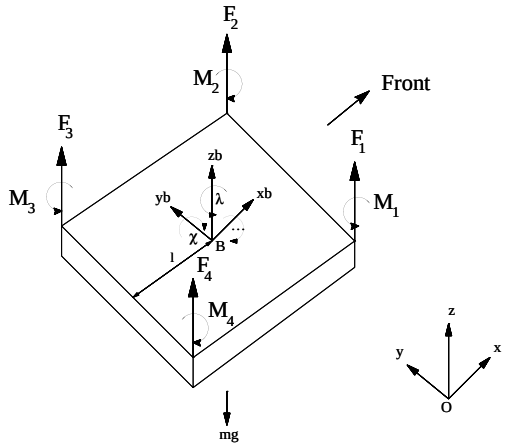

Fig. 2. The three-dimensional quadrotor model.

The basic motions of a quadrotor are generated by tilting the helicopter. As the rotor speeds decrease, the helicopter tilts toward that direction, which enables acceleration along that direction. For this reason, control of the tilt angles and the motion of the helicopter are closely related and estimation of orientation (roll and pitch) is critical. Helicopter motions are coupled. Slowing one rotor results in not only tilting toward that direction, but also a change in total yaw moment and thrust. Because of the coupled nature of the helicopter, to create any motion all of the rotor speeds should be controlled.

The main contribution of this work is the use of non-linear control techniques to stabilize and perform output tracking control of a helicopter. A feedback linearizing controller and a backstepping-like (Sastry 1999) controller have been implemented and shown to be effective on simulations. Since motions along the x- and y-axis are related to tilt angles θ and ψ, respectively, a backstepping controller can be used to control tilt angles, which enables the precise control of the x and y motions. Convergence of the backstepping controllers is guaranteed with iterative development of the controller, where at each step a successful Lyapunov function is generated.

In this study we also use a novel two-camera system for pose estimation. Unlike previous work that either utilizes monocular views or stereo pairs (Amidi 1996; Ma, Kosecká, and Sastry 1998; Shim 2000), our two cameras are set to see each other. A ground camera and an on-board camera are used to obtain accurate pose information. The proposed pose estimation algorithm is compared in simulation with other methods such as the four-point algorithm (Ansar et al. 2001), a state estimation algorithm (Sharp, Shakernia, and Sastry 2001), and the direct area method that uses the area estimations of the blobs. The proposed pose estimation algorithm and the control techniques have been implemented on a remote-controlled, battery-powered model helicopter. The helicopter was attached to a tether in order to provide power and to capture video more easily, as well as to enhance safety by bounding the translation motion in the lateral (x and y) directions.

2. Helicopter Model

It is not an easy task to model a complex helicopter such as a quadrotor. In this section, our goal is to model a four-rotor helicopter as realistically as possible, and to derive control methodologies that would stabilize and control its motions.

For a rigid body model of a three-dimensional quadrotor given in Figure 2, a body-fixed frame (frame B) 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 $\vec{p} = (x, y, z) \in O$ where O is the inertial frame and a rotation matrix $\mathbf{R} : O \rightarrow B$, where $\mathbf{R} \in SO(3)$. A ZYX (Fick angles) Euler angle representation has been chosen for the representation of the rotations, which is composed of three Euler angles, (ϕ, θ, ψ), representing yaw, pitch, and roll respectively:

$$RPY(\phi, \theta, \psi) = Rot(z, \phi) \cdot Rot(y, \theta) \cdot Rot(x, \psi). \quad (1)$$

Let $\vec{V}$ and $\vec{w} \in O$ represent the linear and angular velocities of the rigid body with respect to the inertial frame. Similarly, let $\vec{V}^b$ and $\vec{w}^b \in B$ represent the linear and angular velocities of the rigid body with respect to the body-fixed frame. Let $\vec{\zeta}$ be the vector of Euler angles, $\vec{\zeta} = [\psi, \theta, \phi]^{\mathrm{T}}$ and $\vec{w}^b = [p, q, r]^{\mathrm{T}}$. The body angular velocity is related to Euler angular velocity by $\vec{w}^b = \text{unskew}(\mathbf{R}^{\mathrm{T}}\dot{\mathbf{R}})$, where the unskew( ) term, represents obtaining vector $\vec{w}_b$ from the skew symmetric matrix, skew($\vec{w}^b$). The skew($\vec{w}$) $\in so(3)$ is the skew symmetric matrix of $\vec{w}$.

The Euler angle velocities to body angular velocities are mapped by $\dot{\vec{\zeta}} = \mathbf{J}\vec{w}^b$

$$\begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 1 & s_\psi t_\theta & c_\psi t_\theta \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi/c_\theta & c_\psi/c_\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad (2)$$

where s_ϕ denotes $\sin(\phi)$ and c_ϕ denotes $\cos(\phi)$.

Using the Newton–Euler equations, we can represent the dynamics of the quadrotor as follows

$$\dot{\vec{V}}^b = \frac{1}{m}\vec{F}_{ext} - \vec{w}^b \times \vec{V}^b \quad (3)$$

$$\mathbf{I}_b \dot{\vec{w}}^b = \vec{M}_{ext} - \vec{w}^b \times \mathbf{I}_b \vec{w}^b \quad (4)$$

$$\dot{\vec{\zeta}} = \mathbf{J}\vec{w}^b. \quad (5)$$

Here, $\mathbf{I}_b$ is the inertia matrix, and $\vec{F}_{ext}$ and $\vec{M}_{ext}$ are the external forces and moments on the body-fixed frame given as

$$\vec{F}_{ext} = -drag_x\hat{i} - drag_y\hat{j} + (T - drag_z)\hat{k} - \mathbf{R}\cdot mg\hat{k} \quad (6)$$

$$\vec{M}_{ext} = M_x\hat{i} + M_y\hat{j} + M_z\hat{k}, \quad (7)$$

where T is the total thrust, M_x, M_y, and M_z are the body moments, and $\hat{i}$, $\hat{j}$, and $\hat{k}$ are the unit vectors along the x-, y-, and z-axis, respectively. A drag force acts on a moving body opposite to the direction it moves. The terms $drag_x$, $drag_y$, and $drag_z$ are the drag forces along the appropriate axis. Letting ρ be the density of air, A the frontal area perpendicular to the axis of motion, C_d the drag coefficient and V the velocity, then the drag force on a moving object is

$$drag = \frac{1}{2}C_d\rho V^2 A. \quad (8)$$

Assuming the density of air is constant, then the constants at above equation can be collected, and the equation can be written as

$$drag = C_d V^2. \quad (9)$$

The total thrust (Prouty 1995) is

$$F = bL = \frac{\rho}{4}w^2R^3abc(\theta_t - \phi_t), \quad (10)$$

where a is the slope of the airfoil lift curve, b is the number of blades on a rotor, c is the lift coefficient, L is the lift of a single blade, θ_t is the pitch at the blade tip, ϕ_t is the inflow angle at the tip, w is the rotor speed, and R is the rotor radius. Note that the quadrotor has fixed pitch rotors, and therefore the angle θ_t is constant. Also, we assume that $\phi_t = 0$, implying that we ignore the change in direction of the airflow due to the motion of the quadrotor through the air. By collecting the constant terms as D, for hover or near-hover flight conditions, this equation simplifies to

$$F_i = Dw_i{}^2. \quad (11)$$

Successful control of the helicopter requires direct control of the rotor speeds, w_i. Rotor speeds can be controlled by controlling the motor torque. The torque of motor i, M_i, is related to the rotor speed w_i as

$$M_i = I_r w_i{}^2 + K w_i{}^2, \quad (12)$$

where I_r is the rotational inertia of rotor i, and K is the reactive torque due to the drag terms.

For simplicity, we assume that the inertia of the rotor is small compared to the drag terms, so that the moment generated by the rotor is proportional to the lift force, i.e., $M_i = CF_i$, where C is the force-to-moment scaling factor. For simulations, a suitable C value has been experimentally calculated.

The total thrust force T and the body moments M_x, M_y, and M_z are related to the individual rotor forces through

$$\begin{pmatrix} T \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -l & -l & l & l \\ -l & l & l & -l \\ C & -C & C & -C \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix}, \quad (13)$$

where F_i are the forces generated by the rotors, as given by eq. (11). The matrix above, which we denote by $\mathbf{N} \in R^{4\times4}$, is full rank for $l, C \neq 0$. This is logical since $C = 0$ would imply that the moment around the z-axis is zero, making the yaw-axis control impossible. When $l = 0$, this corresponds to moving the rotors to the center of gravity, which eliminates the possibility of controlling the tilt angles, again implying a lack of control over the quadrotor states.

In summary, to move the quadrotor, motor torques M_i should be selected to produce the desired rotor velocities (w_i) in eq. (12), which will change the body forces and moments in eq. (13). This will produce the desired body velocities and accelerations in eqs. (3) and (4).

3. Control of a Quadrotor Helicopter

In this section we present some control methods for the helicopter. We also give details of the implementation of feedback linearization and a backstepping controller to the three-dimensional quadrotor model and show that it can be stabilized and controlled.

3.1. Introduction

Development of the controllers for a model helicopter is a complex task. There are two basic types of controllers. The

first class considers the system as a whole and attempts to solve full non-linear problem to obtain suitable controllers. This method involves highly complex equations. The second class attempts to break the control into easily controllable modes and finds suitable control methods for each of the modes. This approach to make the quadrotor helicopter autonomous involves 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. Each mode requires only a simple controller, so that the overall problem is broken down into simpler subproblems. The helicopter is stabilized at the hover mode by keeping the positions (x, y, z) constant and angles (θ, ψ, ϕ) zero. 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. The 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. The forward/reverse mode tilts around the y-axis by changing the ψ angle. The hover mode will switch to the landing mode when the flight is complete. These low-level control tasks can be connected to a higher-level controller that does the switching between modes, setting the goal points and performing the motion planning. Similarly, Koo et al. (1998) used hybrid control methodologies for autonomous helicopters. A natural starting place for this is to ask what modes can be controlled.

The helicopter model given in the previous section is a complicated, non-linear system. It includes rotor dynamics, Newton–Euler equations, dynamical effects, and drag. Under some assumptions, it is possible to simplify the above model. Such a simplified model will be useful for derivation of the controllers.

Let us assume that

1. the higher-order terms can be ignored ($\vec{F}_{ext.} \gg m\vec{w}^b \times \vec{V}^b$ and $\vec{M}_{ext.} \gg \vec{w}^b \times \mathbf{I}_b\vec{w}^b$);

2. the inertia matrix ($\mathbf{I}_b$) is diagonal.

To simplify further, let us assume that the pitch (ψ) and roll (θ) angles are small, so that $\mathbf{J}$ in eq. (5) is the identity matrix, giving $\dot{\vec{\zeta}} = \vec{w}^b$.

This leads to the following dynamical equations:

$$\dot{\vec{V}}^b = \frac{1}{m}\vec{F}_{ext.} \tag{14}$$

$$\mathbf{I}_b\dot{\vec{w}}^b = \vec{M}_{ext.} \tag{15}$$

$$\dot{\vec{\zeta}} = \vec{w}^b. \tag{16}$$

The equations of motion can be written using the force and moment balance on the inertial frame:

$$\begin{aligned}
\ddot{x} &= \left[\left(\sum_{i=1}^{4} F_i\right)(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - K_1\dot{x}\right]/m \\
\ddot{y} &= \left[\left(\sum_{i=1}^{4} F_i\right)(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) - K_2\dot{y}\right]/m \\
\ddot{z} &= \left[\left(\sum_{i=1}^{4} F_i\right)(\cos\theta\cos\psi) - mg - K_3\dot{z}\right]/m \\
\ddot{\theta} &= l(-F_1 - F_2 + F_3 + F_4 - K_4\dot{\theta})/J_1 \\
\ddot{\psi} &= l(-F_1 + F_2 + F_3 - F_4 - K_5\dot{\psi})/J_2 \\
\ddot{\phi} &= (M_1 - M_2 + M_3 - M_4 - K_6\dot{\phi})/J_3.
\end{aligned} \tag{17}$$

J_i given above are the moments of inertia with respect to the corresponding axes, and K_i are the drag coefficients. In the following, we assume the drag is zero, since drag is negligible at low speeds.

For convenience, we define the inputs to be

$$\begin{aligned}
u_1 &= (F_1 + F_2 + F_3 + F_4)/m = T/m \\
u_2 &= (-F_1 - F_2 + F_3 + F_4)/J_1 = T_x/J_1 \\
u_3 &= (-F_1 + F_2 + F_3 - F_4)/J_2 = T_y/J_2 \\
u_4 &= C(F_1 - F_2 + F_3 - F_4)/J_3 = T_z/J_3,
\end{aligned} \tag{18}$$

where C is the force-to-moment scaling factor. u_1 represents a total thrust/mass on the body in the z-axis, u_2 and u_3 are the pitch and roll inputs, and u_4 is the input to control yawing motion. Therefore, the equations of motion become

$$\begin{aligned}
\ddot{x} &= u_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) & \ddot{\theta} &= u_2 l \\
\ddot{y} &= u_1(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) & \ddot{\psi} &= u_3 l \\
\ddot{z} &= u_1(\cos\theta\cos\psi) - g & \ddot{\phi} &= u_4.
\end{aligned} \tag{19}$$

3.2. Feedback Linearization

One can use exact input–output linearization and pick the outputs to be z, x, y, and ϕ, which results in a complex equation with higher derivatives. Our goal is to use a vision system which is subject to noise, and therefore the use of higher-order derivatives of the states is not desirable.

We can alternatively choose outputs to be z, θ, ψ, and ϕ, in order to control the altitude, yaw, and tilt angles of the quadrotor. However, this controller introduces zero dynamics, which results in the drift of the helicopter in the x–y-plane. The main reason for the zero dynamics is the fact that even small tilt angles result in acceleration along the x- or y-axis, and the only way to control these accelerations is to tilt in the opposite direction to generate negative accelerations. The zero dynamics for this system are

$$\begin{aligned}
\ddot{x} &= g(\sin\theta + \tan\phi\tan\psi) \\
\ddot{y} &= g(\tan\phi\sin\theta - \tan\psi).
\end{aligned} \tag{20}$$

These zero dynamics are not desirable, and so another controller or a combination of controllers is needed.

3.3. Proportional Derivative Controllers

One can design controllers separately by considering the helicopter motions. Noticing that the motion along the y-axis is related to the ψ tilt angle by eq. (19), one can design a proportional derivative (PD) controller to control the ψ angle in order to control y motions. From the $\ddot{y}$ term in eq. (19), setting $\theta = \phi = 0$ and $u_1 = 1$ gives

$$\ddot{y} = -K_p y - K_d \dot{y} = -\sin\psi. \tag{21}$$

The desired tilt angle (ψ_d) can be written as

$$\psi_d = \arcsin(K_p y + K_d \dot{y}). \tag{22}$$

By taking the derivative of this expression, one can obtain the expression for $\dot{\psi}_d$, the desired tilt angle velocity. The maximum tilt angle depends on the helicopter model. Let us assume that the maximum tilt angle is $Tilt_{Max}$. Therefore, for experiments the ψ_d value calculated will be bounded with the appropriate $Tilt_{Max}$ value. If we can select the desired tilt angle and tilt angle velocity based on the y position and $\dot{y}$, we can control the motion along that axis with a PD controller of the form

$$u_3 = K_{p1}(\psi_d - \psi) + K_{d1}(\dot{\psi}_d - \dot{\psi}), \tag{23}$$

where

$$\begin{aligned} \psi_d &= \arcsin(K_p y + K_d \dot{y}) \\ \dot{\psi}_d &= \frac{K_p \dot{y} + K_d \ddot{y}}{\sqrt{1 - K_p^2 y^2 - 2K_p K_d y\dot{y} - K_d^2 \dot{y}^2}}. \end{aligned} \tag{24}$$

Similarly a θ–x controller of the form can be used to generate the u_2 input

$$u_2 = K_{p2}(\theta_d - \theta) + K_{d2}(\dot{\theta}_d - \dot{\theta}) \tag{25}$$

where

$$\begin{aligned} \theta_d &= \arcsin(-K_p x - K_d \dot{x}) \\ \dot{\theta}_d &= -\frac{K_p \dot{x} + K_d \ddot{x}}{\sqrt{1 - K_p^2 x^2 - 2K_p K_d x\dot{x} - K_d^2 \dot{x}^2}}. \end{aligned} \tag{26}$$

The altitude and the yaw, on the other hand, can be controlled by PD controllers

$$\begin{aligned} u_1 &= \frac{g + K_{p1}(z_d - z) + K_{d1}(\dot{z_d} - \dot{z})}{\cos\theta\cos\psi} \\ u_4 &= K_{p2}(\phi_d - \phi) + K_{d2}(\dot{\phi}_d - \dot{\phi}). \end{aligned} \tag{27}$$

Figure 3 shows the quadrotor simulation, where it moves from (40, 20, 60) to the origin with initial zero yaw and tilt angles.

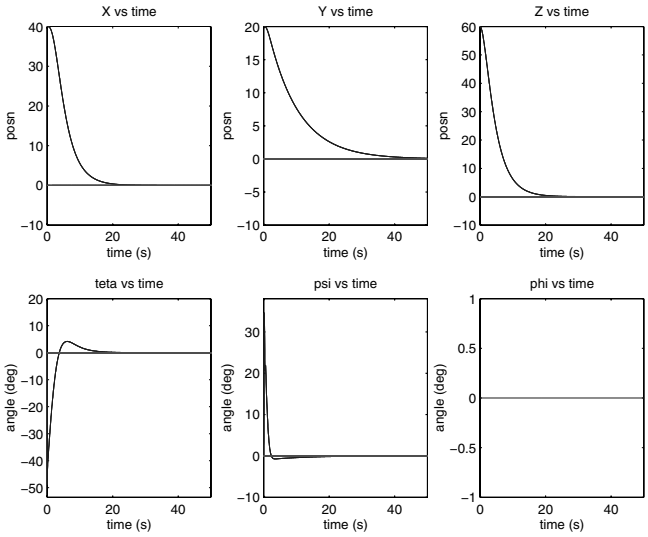

Fig. 3. PD controller simulation results.

3.4. Backstepping Controller

Backstepping controllers (Sastry 1999) are especially useful when some states are controlled through other states. To move the quadrotor along the x- and y-axis, the tilt angles need to be controlled. One can use backstepping controllers to control these motions. Similar ideas of using backstepping with visual servoing have been developed for a traditional helicopter by Hamel and Mahony (2000). The approach here is slightly simpler in implementation, and relies on simple estimates of pose.

The x motion of the helicopter is related to the θ angle, and similarly the y motion is related to the ψ angle. A backstepping controller can be used to control these angles to control the x and y motions of the helicopter. The use of a small angle assumption on ψ in the $\ddot{x}$ term, and a small angle assumption on θ in the $\ddot{y}$ term of eq. (19) gives

$$\begin{aligned} \ddot{x} &= u_1 \cos\phi \sin\theta \\ \ddot{y} &= -u_1 \cos\phi \sin\psi. \end{aligned} \tag{28}$$

The detailed derivation of the controllers is given in the Appendix. After simplifications of the above equations, this leads to a backstepping controller for x–θ of

$$\begin{aligned} u_2 = \frac{1}{u_1 \cos\theta \cos\phi}(&-5x - 10\dot{x} - 9u_1 \sin\theta\cos\phi \\ &-4u_1\dot{\theta}\cos\theta\cos\phi + u_1\dot{\theta}^2 \sin\theta\cos\phi \\ &+2u_1\dot{\phi}\sin\theta\sin\phi + u_1\dot{\theta}\dot{\phi}\cos\theta\sin\phi \\ &-u_1\dot{\phi}\dot{\theta}\cos\theta\sin\phi - u_1\dot{\phi}^2\sin\theta\cos\phi). \end{aligned} \tag{29}$$

This controller controls the input u_2, and therefore angle θ, in order to control x motions. To develop a controller for

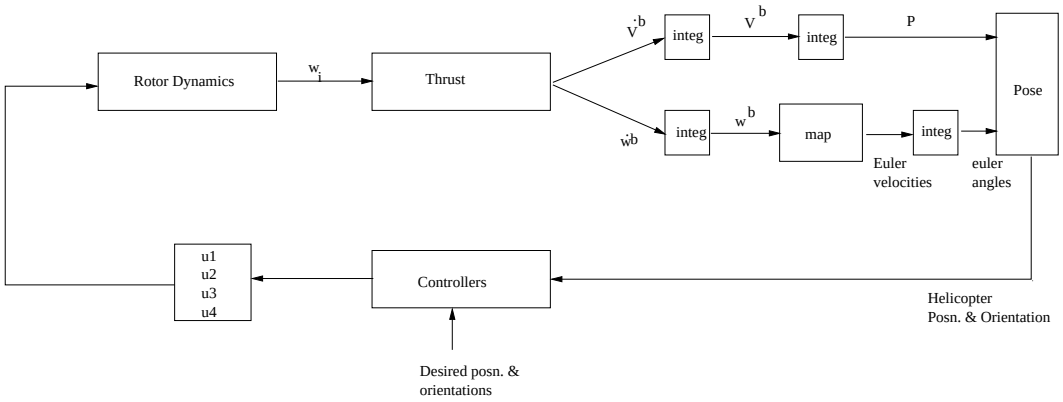

Fig. 4. Simulation model.

motion along the y-axis, similar analysis is needed. This leads to a backstepping controller for y–ψ control given by

$$\begin{aligned} u_3 = \frac{1}{u_1 \cos\psi \cos\phi}(&-5y - 10\dot{y} - 9u_1 \sin\psi \cos\phi \\ &-4u_1\dot{\psi}\cos\psi\cos\phi + u_1\dot{\psi}^2 \sin\psi\cos\phi \\ &+2u_1\dot{\psi}\sin\psi\sin\phi + u_1\dot{\psi}\dot{\phi}\cos\psi\sin\phi \\ &-u_1\dot{\phi}\dot{\psi}\cos\psi\sin\phi - u_1\dot{\phi}^2\sin\psi\cos\phi). \end{aligned} \tag{30}$$

The sequential derivation of the backstepping controller involves finding suitable Lyapunov functions, and therefore the controllers are guaranteed to exponentially stabilize the helicopter. The altitude and the yaw, on the other hand, can be controlled by PD controllers given in eq. (27). If these controllers are placed in eq. (19), this gives

$$\begin{aligned} \ddot{z} &= K_{p1}(z_d - z) + K_{d1}(\dot{z_d} - \dot{z}) \\ \ddot{\phi} &= K_{p2}(\phi_d - \phi) + K_{d2}(\dot{\phi}_d - \dot{\phi}). \end{aligned} \tag{31}$$

Therefore, the proposed controllers exponentially stabilize the helicopter.

The proposed controllers are implemented on a MATLAB Simulink simulation as shown in Figure 4. The helicopter model is based on the model given by eqs. (3)–(5). The following values are used for the simulation. The force-to-moment ratio, C, was found experimentally to be 1.3. The length between rotors and center of gravity, l, was taken as 21 cm. The inertia matrix elements are calculated with a point mass analysis as $I_x = 0.0142$ kg m^2, $I_y = 0.0142$ kg m^2 and $I_z = 0.0071$ kg m^2. The mass of the helicopter is taken as 0.56 kg. The drag coefficients are taken as $C_x = 0.6$, $C_y = 0.6$, and $C_z = 0.9$. Gravity is $g = 9.81$ m s^{-2}.

In reality, thrust forces are limited. Therefore, the maximum thrust is taken as 10 N, and the inputs are limited by

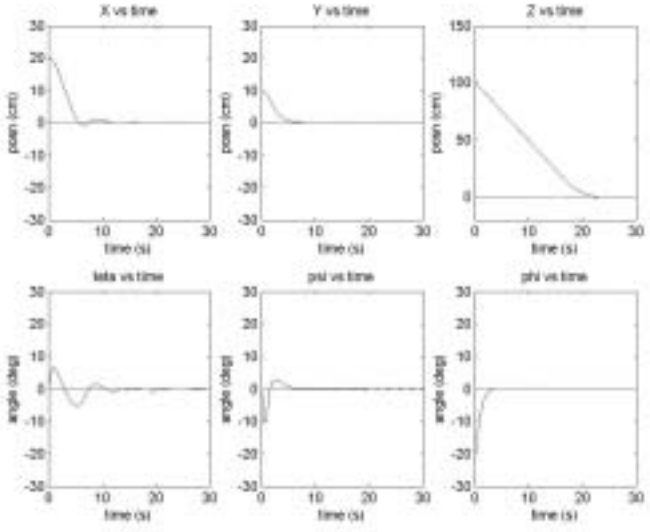

Fig. 5. Backstepping controller simulation results.

$$\begin{aligned} -F_{max}/m \le u_1 \le 4F_{max}/m \\ -2F_{max}/l \le u_2 \le 2F_{max}/l \\ -2F_{max}/l \le u_3 \le 2F_{max}/l \\ -2CF_{max} \le u_4 \le 2CF_{max}. \end{aligned} \tag{32}$$

The simulation results in Figure 5 show the motion of the quadrotor from position (20, 10, 100) to the origin, while reducing the yaw angle from -20° to zero. Figure 6 represents the motion of the quadrotor during this motion.

The controller is strong enough to handle random errors that 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 the error on z has variance of 2 cm. The yaw

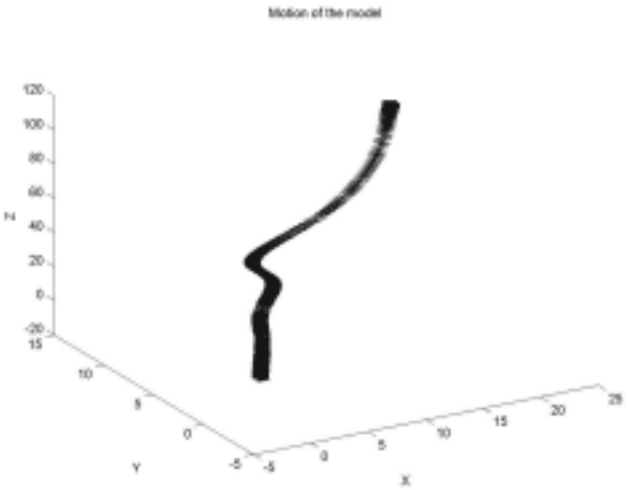

Fig. 6. Backstepping controller path.

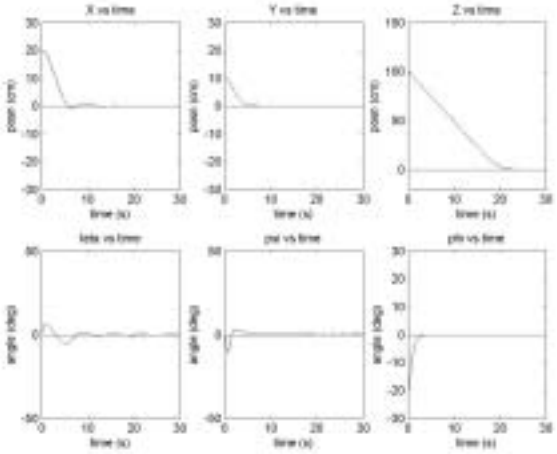

Fig. 8. The effect of the delay on the simulation.

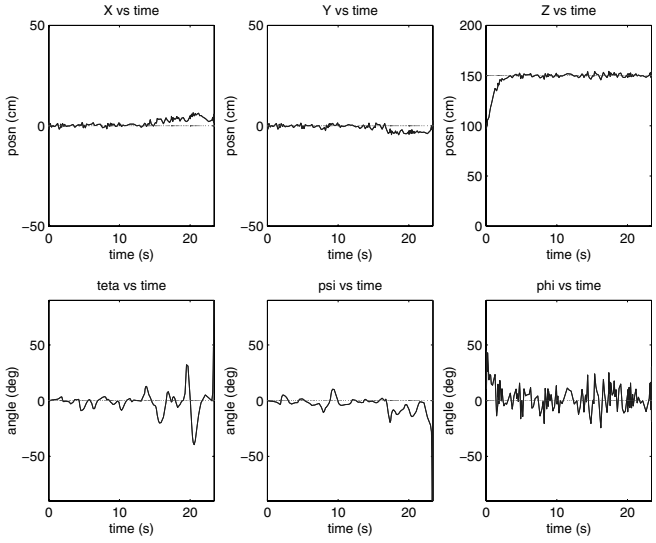

Fig. 7. Backstepping controller simulation with random noise at x, y, z, and ϕ values.

variance is 1.5°. The helicopter moves from 100 to 150 cm, while reducing the yaw angle from 30° to zero. The mean and standard deviations are found to be 150 and 1.7 cm for z and 2.4° and 10.1° for ϕ, respectively.

One of the biggest problems in vision-based control is the fact that the vision system is not a continuous feedback device. Unlike sensors that have a much higher rate than the vision updates, such as an accelerometer or potentiometer, the readings (images) have to be captured, transferred, and analyzed. Therefore, to simulate the discrete nature of the feedback system, this problem has to be included in the model. Usually, the frame rate of many cameras is 20–30 Hz. A frame rate of 15 Hz will be used for the overall sampling rate of this sensory system. Figure 8 shows the results of the simulation, where the x, y, and z positions are sampled at 15 Hz. The controllers are robust enough to handle the discrete inputs. A simple comparison of the plots shows that discrete sampling causes an increased settling time.

Considering the methods introduced in Section 3, the feedback linearizing controllers have the disadvantage of complexity. The controllers generated usually involve higher derivatives, which are sensitive to sensor errors. The results on PD controllers depend on the correct selection of gains K_p and K_d. Considering the settling time, and the ability to perform with noisy or delayed data, the backstepping controllers are much better than the PD controllers. Moreover, the backstepping controllers are guaranteed to exponentially stabilize the helicopter.

4. Pose Estimation

Control of a helicopter will not be possible if its position and orientation are not known. The position and orientation of the helicopter with respect to a known reference frame are required for the controllers in order to generate the control inputs and the motion trajectories. Also, for surveillance and remote inspection tasks a relative position and orientation detection is important. This may be the location of the landing pad with respect to the helicopter or the area of interest that is inspected. For these tasks, a vision system can be used to provide position and orientation information.

The purpose of the pose estimation is to obtain the relative position and orientation of the helicopter with respect

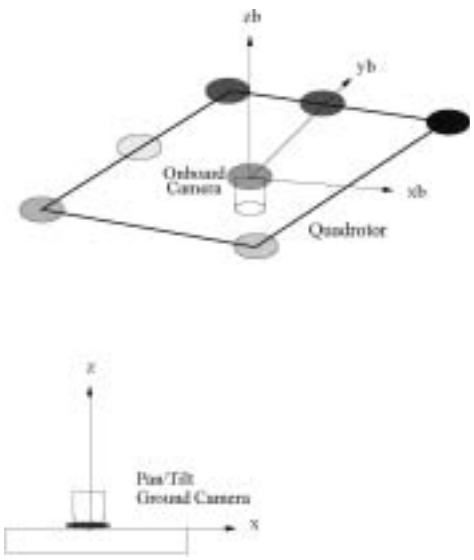

Fig. 9. Helicopter pose estimation.

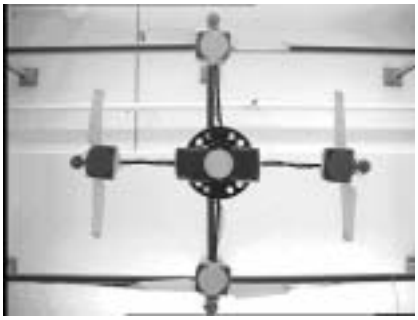

Fig. 10. Quadrotor tracking with a camera.

to a ground frame using vision. Our goal is to obtain the pose from vision rather than complex and heavy navigation systems or global positioning system (GPS). Previous work on vision-based pose estimation utilizes monocular views or stereo pairs. The two-camera pose estimation method proposed in this section uses a pan/tilt/zoom ground camera and an on-board camera, which are set to see each other.

The pose estimation can be defined as: finding the rotation matrix, $R \in SO(3)$, defining the body-fixed frame of the helicopter with respect to the fixed frame located at the ground camera frame and the relative position $\vec{p} \in R^3$, which is the position of the helicopter with respect to the ground camera, as well as velocities $\vec{w}$ and $\vec{V}$ in real time.

In this section, we introduce the two-camera pose estimation algorithm, and compare it in simulations to other pose estimation algorithms.

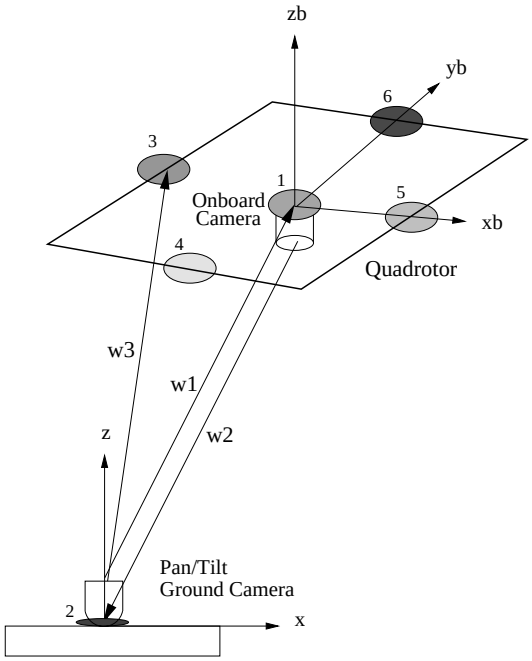

Fig. 11. Two-camera pose estimation method using a pair of ground and on-board cameras.

4.1. Two-camera Pose Estimation Method

The two-camera pose estimation method involves the use of two cameras that are set to see each other. One of the cameras is located on the ground and the other is an on-board camera looking downwards. This method is useful for autonomous take-off or landing, especially when the relative motion information is critical, such as landing on a ship at rough seas. Colored blobs of 2.5 cm radius are attached to the bottom of the quadrotor and to the ground camera, as shown in Figure 11. A blob tracking algorithm is used to obtain the positions and areas of the blobs on the image planes. Tracking two blobs on the quadrotor image plane and one blob on the ground image frame is found to be enough for accurate pose estimation. To minimize the error as much as possible, five blobs are placed on the quadrotor and a single blob is located on the ground camera. The blob tracking algorithm tracks the blobs and returns image values (u_i, v_i) for $i = 1 \ldots 6$.

The cameras have matrices of intrinsic parameters, $\mathbf{A_1}$ and $\mathbf{A_2}$. Let $\vec{w}_i \in \mathbf{R}^3$ be a unit vector, and λ_i an unknown scalar. The unit vector $\vec{w}_i$ from each camera to the blobs can be found as

$$\begin{aligned} \vec{w}_i = inv(\mathbf{A_1}).[u_i \quad v_i \quad 1]', \quad \vec{w}_i = \vec{w}_i/norm(\vec{w}_i) \\ \text{for} \quad i = 1, 3, 4, 5, 6 \\ \vec{w}_2 = inv(\mathbf{A_2}).[u_2 \quad v_2 \quad 1]', \quad \vec{w}_2 = \vec{w}_2/norm(\vec{w}_2). \end{aligned} \tag{33}$$

Let $\vec{L}_a$ be the vector pointing from blob 1 to blob 3 in Figure 11. Vectors $\vec{w}_1$ and $\vec{w}_3$ are related by

$$\lambda_3 \vec{w}_3 = \lambda_1 \vec{w}_1 + \mathbf{R}\vec{L}_a, \tag{34}$$

where λ_1 and λ_3 are unknown scalars. Taking the cross-product with $\vec{w}_3$ gives

$$\lambda_1(\vec{w}_3 \times \vec{w}_1) = \mathbf{R}\vec{L}_a \times \vec{w}_3. \tag{35}$$

This can be rewritten as

$$(\vec{w}_3 \times \vec{w}_1) \times (\mathbf{R}\vec{L}_a \times \vec{w}_3) = 0. \tag{36}$$

In order to solve the above equation, let the rotation matrix $\mathbf{R}$ be composed of two rotations: the rotation of θ degrees around the vector formed by the cross-product of $\vec{w}_1$ and $\vec{w}_2$, and the rotation of α degrees around $\vec{w}_1$. In other words

$$\mathbf{R} = Rot(\vec{w}_1 \times \vec{w}_2, \theta) \cdot Rot(\vec{w}_1, \alpha), \tag{37}$$

where $Rot(\vec{a}, b)$ means the rotation of b degrees around the unit vector $\vec{a}$. The value of θ can be found from the dot product of vectors $\vec{w}_1$ and $\vec{w}_2$:

$$\theta = a\cos(\vec{w}_1 \cdot \vec{w}_2). \tag{38}$$

Alternatively, one can use the cross-product of w_1 and w_2, to solve θ angle. The only unknown left in eq. (36) is the angle α.

Rewriting eq. (36) gives

$$(w_3 \times \vec{w}_1) \times (\vec{w}_3 \times (Rot(\vec{w}_1 \times \vec{w}_2, \theta) \cdot Rot(\vec{w}_1, \alpha))L_a) = 0. \tag{39}$$

Let $\mathbf{M}$ be given as

$$\mathbf{M} = (\vec{w}_3 \times \vec{w}_1) \times \{\vec{w}_3 \times [\mathbf{R}(\vec{w}_1 \times \vec{w}_2, \theta)]\}. \tag{40}$$

Using Rodrigues' formula, $Rot(w_1, \alpha)$ can be written as

$$Rot(w_1, \alpha) = \mathbf{I} + \hat{w_1}\sin\alpha + \hat{w_1}^2(1 - \cos\alpha). \tag{41}$$

Pre-multiplying eq. (41) with $\mathbf{M}$ and post-multiplying it with L_a gives the simplified version of eq. (39)

$$\mathbf{M} \cdot \vec{L}_a + \sin\alpha \cdot \mathbf{M}\vec{w}_1 \cdot \vec{L}_a + (1 - \cos\alpha) \cdot \mathbf{M} \cdot (\vec{w}_1)^2 \cdot \vec{L}_a = 0. \tag{42}$$

This is a set of three equations in the form of $A\cos\alpha + B\sin\alpha = C$, which can be solved by

$$\alpha = \arcsin \frac{B \cdot C \pm \sqrt{(B^2 \cdot C^2 - (A^2 + B^2) \cdot (C^2 - A^2))}}{A^2 + B^2}. \tag{43}$$

One problem here is that $\alpha \in [\pi/2, -\pi/2]$, because of the arcsin function. Therefore, one must check the unit vector formed by two blobs to find the heading, and pick the correct α value.

Thus, the estimated rotation matrix will be $\mathbf{R} = Rot(\vec{w}_1 \times \vec{w}_2, \theta) \cdot Rot(\vec{w}_1, \alpha)$. Euler angles (ϕ, θ, ψ) defining the orientation of the quadrotor can be obtained from rotation matrix, $\mathbf{R}$.

In order to find the relative position of the helicopter with respect to the inertial frame located at the ground camera frame, we need to find scalars λ_i, for $i = 1 \ldots 6$. λ_1 can be found using eq. (35). The other λ_i values $(\lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6)$ can be found from the relation of the blob positions

$$\lambda_i \vec{w}_i = \lambda_1 \vec{w}_1 + \mathbf{R}\vec{L}_i. \tag{44}$$

L_i is the position vector of the ith blob in the body-fixed frame. To reduce the errors, λ_i values are normalized using the blob separation, L.

The center of the quadrotor will be

$$\begin{aligned} X &= \left[\lambda_3 \vec{w}_3(1) + \lambda_4 \vec{w}_4(1) + \lambda_5 \vec{w}_5(1) + \lambda_6 \vec{w}_6(1)\right]/4 \\ Y &= \left[\lambda_3 \vec{w}_3(2) + \lambda_4 \vec{w}_4(2) + \lambda_5 \vec{w}_5(2) + \lambda_6 \vec{w}_6(2)\right]/4 \\ Z &= \left[\lambda_3 \vec{w}_3(3) + \lambda_4 \vec{w}_4(3) + \lambda_5 \vec{w}_5(3) + \lambda_6 \vec{w}_6(3)\right]/4. \end{aligned} \tag{45}$$

4.2. Comparing the Pose Estimation Methods

The proposed two-camera pose estimation method is compared with other methods using a MATLAB simulation. Other methods used were a four-point algorithm (Ansar et al. 2001), a state estimation algorithm (Sharp, Shakernia, and Sastry 2001), a direct method that uses the area estimations of the blobs, and a stereo pose estimation method that uses two ground cameras that are separated by a distance d.

In this simulation, the quadrotor moves from the point (22, 22, 104) to (60, 60, 180) cm, while (θ, ψ, ϕ) changes from (0.7, 0.9, 2) to (14, 18, 40) degrees. A random error up to five pixels was added on image values, to simulate the errors associated with the blob extractor. A random error of magnitude ± 2 was also added to the blob area estimates on image plane.

The errors are calculated using angular and positional distances, given as

$$\begin{aligned} e_{ang} &= \| \log(\mathbf{R}^{-1} \cdot \mathbf{R}^{\mathbf{est}}) \| \\ e_{pos} &= \| \vec{p} - \vec{p}^{est} \| . \end{aligned} \tag{46}$$

$\mathbf{R}^{\mathbf{est}}$ and $\vec{p}^{est}$ are the estimated rotational matrix and the position vector. The angular error is the amount of rotation about a unit vector that transfers $\mathbf{R}$ to $\mathbf{R}^{\mathbf{est}}$.

In the simulation, the helicopter and blobs moved according to the path given. The projection of those points on the image plane are corrupted with random errors and the resulting image values are used at each step to estimate the pose for each method; the results are compared. The comparison of the pose estimation methods and the average angular and positional errors are given in Table 1.

It can be seen from Table 1 that the estimation of orientation is more sensitive to errors than position estimation. The direct method uses the blob areas, which leads to poor pose estimates

Table 1. Comparison of the Angular and Positional Errors of Different Pose Estimation Methods

Method	Angular error (degree)	Position error (cm)
Direct	10.2166	1.5575
Four point	3.0429	3.0807
Two camera	1.2232	1.2668
Linear	4.3700	1.8731
Stereo	6.5467	1.1681

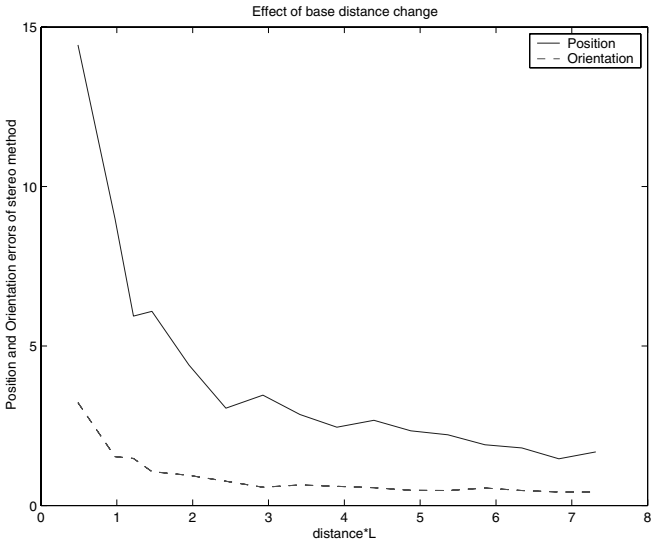

Fig. 12. Effects of baseline distance change on pose estimation at the stereo method.

due to noisy blob area readings. Based on the simulations, we can conclude that the two-camera method is more effective for pose estimation, especially when there are errors on the image plane. For the stereo method, the value of the baseline is important for pose estimation. Figure 12 shows the effects of the baseline distance change on the estimation. As the baseline distance approaches three times the distance between blobs ($6L$), the stereo method appears to give good estimates. The need for a large baseline for stereo pairs is the drawback of the stereo method. Placing a stereo pair that has big baseline on a mobile robot is not desirable.

The two-camera method is powerful for estimating the pose of a flying vehicle. Basically, the position of the quadrotor can be estimated very well from the ground-based camera, and the orientation (rotation) of the quadrotor can be estimated very well using the on-board camera on the quadrotor, and a central blob seen by the ground camera. The reason behind the effectiveness of the two-camera method is that it combines the strengths of ground-based and on-board estimation methods in obtaining the best estimate of the quadrotor's pose.

5. Experiments

The proposed controllers and the two-camera pose estimation algorithm have been implemented on a remote-controlled battery-powered helicopter as shown in Figure 14. It is a commercially available model helicopter called HMX-4. It is about 0.7 kg, 76 cm long between rotor tips, and has about 3 min flight time. This helicopter has three gyros on board to enhance stability, by providing damping on the rotational motion of the quadrotor. An experimental setup shown in Figure 15 was prepared to prevent the helicopter from moving too much on the x–y-plane, while enabling it to turn and ascend/descend freely.

We used off-the-shelf hardware components for the system. The computers processing vision are Pentium 4, 2 GHz machines with Imagination PXC200 color frame grabbers. Images can be captured at 640 $\times$ 480 resolution at 30 Hz. The cameras used for the experiments were a Sony EVI-D30 pan/tilt/zoom color camera as the ground camera, and a tiny CCD color camera as the on-board camera. The pose estimation algorithms depend heavily on the detection of the features, in this case color blobs on the image. When considering color images from CCD cameras, there are several color spaces that are common, including RGB, HSV, and YUV. The YUV space has been chosen for our application. The gray-scale information is encoded in the Y channel, while the color information is transmitted through the U and V channels. Color tables are generated for each color in MATLAB. Multiple images and various lighting conditions have to be used to generate the color tables, to reduce the effect of lighting condition changes. The ability to locate and track various blobs is critical. The blob tracking routines return the image coordinates of all color blobs as well as the sizes of the blobs. It can track up to eight different blobs at a speed depending on the camera, computer, and frame grabber. Our system can track the blobs at about 20 Hz.

Vision-based stabilization experiments were performed using the two-camera pose estimation method. In these experiments, two separate computers were used. Each camera was connected to a separate computer, which was responsible for performing blob tracking. PC 1 was responsible for image processing of the on-board camera and the information then transferred to PC 2 via the network. PC 2 was responsible for the ground pan/tilt camera image processing and control, as well as the calculation of the control signals for the helicopter control. These signals were then sent to the helicopter with a remote control device that uses the parallel port. The rotor speeds are set accordingly to achieve the desired positions and orientations. The system block diagram is shown in Figure 13.

The backstepping controllers described in Section 3.4 were implemented for the experiments. Figure 16 shows the

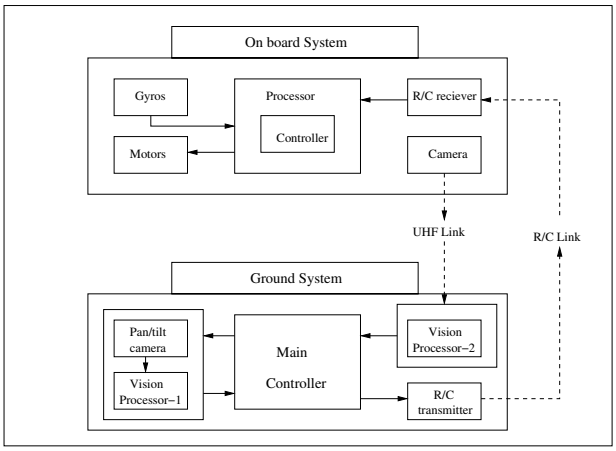

Fig. 13. System block diagram.

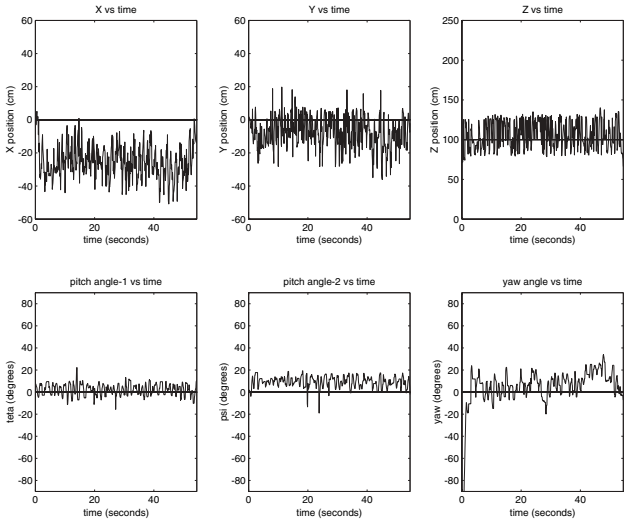

Fig. 16. Results of the height, x, y, and yaw control experiment with the two-camera pose estimation method.

Fig. 14. Quadrotor helicopter.

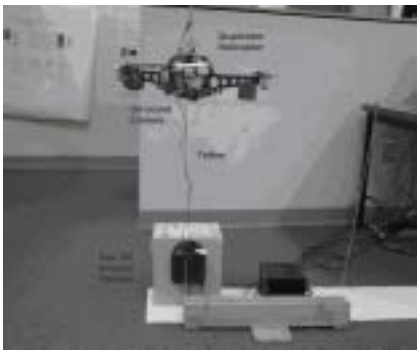

Fig. 15. Experimental setup.

results of the experiment using the two-camera pose estimation method, where height, x, y, and yaw angle are being controlled. The mean and standard deviations are found to be 106 and 17.4 cm for z, -25.15 and 10.07 cm for x, -7.57 and 9.51 cm for y, and 4.96° and 18.3° for ϕ, respectively. The results from the plots show that the proposed controllers do an acceptable job despite the pose estimation errors and errors introduced by the tether.

6. Conclusions and Future Work

In this paper we have presented pose estimation algorithms and non-linear control techniques to build and control autonomous helicopters. A novel two-camera method for helicopter pose estimation has been introduced. The method has been compared to other pose estimation algorithms, and has been shown to be more effective especially when there are errors on the image plane. Feedback linearization and backstepping controllers have been used to stabilize and perform output tracking control on the two-dimensional and three-dimensional quadrotor models. Simulations performed on MATLAB Simulink show the ability of the controller to perform output tracking control even when there are errors on state estimates. The proposed controllers and the pose estimation method have been implemented on a remote-controlled, battery-powered model helicopter. Initial experiments on a tethered system have shown that the vision-based control is effective in controlling the helicopter.

One of the drawbacks of the vision system is that it is not reliable when the lighting on the scene changes. Similarly, GPS is not reliable in certain locations, such as indoors, near

trees or near tall buildings. Therefore, it is better to use multiple sensors as much as possible to overcome limitations of individual sensors. One possibility is to investigate methods to reduce the errors on the feature detection. For indoor environments, an indoor GPS system can be built to increase the reliability of the system.

A quadrotor helicopter can be teamed up with a ground mobile robot to form a heterogeneous marsupial robotic team (Murphy et al. 1999), where the quadrotor rides on the mobile robot for deployment in the field. By placing a camera on top of the mobile robot, mobility will be introduced to the two-camera system. Moreover, a platform can be built to enable the helicopter to take-off/land on the mobile robot. The ground robot will be responsible for transportation, communication relay, assistance in landing and take-off with the camera, and will provide assistance on computation of the visual information. Such functionalities will be useful for remote sensing, chase, and other ground–air cooperation tasks.

Appendix: Detailed Derivation of the Controllers

In this appendix we give details of the derivation of the backstepping controller for the x motions.

The equations of motions of the quadrotor on x and y are given as

$$\ddot{x} = u_1(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \tag{47}$$

$$\ddot{\theta} = u_2 \tag{48}$$

$$\ddot{y} = u_1(\sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi) \tag{49}$$

$$\ddot{\psi} = u_3. \tag{50}$$

From eqs. (47) and (49), it is clear that the θ angle also affects the y motion, and similarly the ψ angle also affects the x motion. In order to reduce this effect, we use a small angle assumption on the ψ angle in eq. (47), and a small angle assumption on the θ angle in eq. (49) to give

$$\begin{aligned} \ddot{x} &= u_1 \sin\theta \cos\psi \cos\phi \\ \ddot{y} &= -u_1 \sin\psi \cos\phi. \end{aligned} \tag{51}$$

Let us design the x–θ controller. Let the system be

$$\begin{aligned} x_1 &= x \\ x_2 &= \dot{x} \\ x_3 &= \theta \\ x_4 &= \dot{\theta}. \end{aligned} \tag{52}$$

Then

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u_1 C_\phi S_{x_3} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= u_2. \end{aligned} \tag{53}$$

For $i = 1$, let $z_1 = x_1$ and $z_2 = x_2 + x_1$, then $x_2 = z_2 - z_1$. Integrating these gives, $\dot{z}_1 = z_2 - z_1$ and $\dot{z}_2 = \dot{x}_2 + \dot{x}_1$. A Lyapunov function of the form, $V_1 = (1/2){z_1}^2$, gives

$$\dot{V}_1 = -{z_1}^2 + z_1 z_2. \tag{54}$$

Also

$$\dot{z}_2 = u_1 C_\phi S_{x_3} + x_2 = \bar{f}_2 + x_3 \Rightarrow \bar{f}_2 = u_1 C_\phi S_{x_3} + x_2 - x_3. \tag{55}$$

For $i = 2$, define

$$z_3 = x_3 - \alpha_2 \tag{56}$$

$$\dot{z}_2 = z_3 + \alpha_2 + \bar{f}_2. \tag{57}$$

One can use a Lyapunov function of the form, $V_2 = V_1 + (1/2){z_2}^2$.

Let us pick

$$\alpha_2 = -z_1 - z_2 - \bar{f}_2 = -z_1 - z_2 - u_1 C_\phi S_{x_3} - x_2 + x_3. \tag{58}$$

Then, z_3 becomes

$$z_3 = z_1 + z_2 + u_1 C_\phi S_{x_3} + x_2 = 2x_2 + 2x_1 + u_1 C_\phi S_{x_3}. \tag{59}$$

$\dot{V}_2$ becomes

$$\dot{V}_2 = -{z_1}^2 - {z_2}^2 + z_2 z_3. \tag{60}$$

For $i = 3$,

$$z_4 = x_4 - \alpha_3 \tag{61}$$

$$V_3 = \frac{1}{2}({z_1}^2 + {z_2}^2 + {z_3}^2) \tag{62}$$

$$\dot{z}_3 = z_4 + \alpha_3 + \bar{f}_3 = 2\dot{x}_2 + 2\dot{x}_1 + u_1 \dot{x}_3 C_\phi C_{x_3} - u_1 \dot{\phi} S_\phi S_{x_3}. \tag{63}$$

Then, $\bar{f}_3$ becomes

$$\bar{f}_3 = 2u_1 C_\phi S_{x_3} + 2x_2 + u_1 \dot{x}_3 C_\phi C_{x_3} + x_4 - u_1 \dot{\phi} S_\phi S_{x_3}. \tag{64}$$

$\dot{V}_3$ becomes

$$\dot{V}_3 = -{z_1}^2 - {z_2}^2 - {z_3}^2 + z_3 z_4. \tag{65}$$

For $i = 4$,

$$z_5 = x_5 + \alpha_4 \tag{66}$$

$$V_4 = \frac{1}{2}({z_1}^2 + {z_2}^2 + {z_3}^2 + {z_4}^2) \tag{67}$$

$$z_4 = x_4 - \alpha_3 \tag{68}$$

where

$$\alpha_3 = -3x_1 - 5x_2 - 3u_1 C_\phi S_{x_3} - u_1 \dot{x}_3 C_\phi + x_4 - u_1 \dot{\phi} S_\phi S_{x_3}. \tag{69}$$

Taking the derivative of z_4 gives

$$\begin{aligned} \dot{z}_4 = {} & 3\dot{x}_1 + 5\dot{x}_2 + 3u_1 C_\phi \dot{x}_3 C_{x_3} - 3u_1 \dot{\phi} S_\phi S_{x_3} \\ & - u_1 \dot{x}_3^2 S_{x_3} C_\phi - u_1 \dot{x}_3 C_{x_3} \dot{\phi} S_\phi + u_1 \dot{\phi} S_\phi \dot{x}_3 C_{x_3} \\ & + u_1 \dot{\phi}^2 C_\phi S_{x_3} + u_1 \ddot{x}_3 C_{x_3} C_\phi. \end{aligned} \tag{70}$$

$\dot{V}_4$ becomes

$$\dot{V}_4 = -{z_1}^2 - {z_2}^2 - {z_3}^2 - {z_4}^2. \tag{71}$$

Also

$$\dot{z}_4 = \bar{f}_4 + u_2 \tag{72}$$

and

$$u_2 = -z_3 - z_4 - \bar{f}_4. \tag{73}$$

After placing z_3, z_4, and $\bar{f}_4$ into the above equation and performing simplifications, u_2, the backstepping controller for x–θ, will be obtained

$$\begin{aligned} u_2 = {} & \frac{1}{u_1 \cos\theta \cos\phi} (-5x - 10\dot{x} - 9u_1 \sin\theta \cos\phi \\ & - 4u_1 \dot{\theta} \cos\theta \cos\phi + u_1 \dot{\theta}^2 \sin\theta \cos\phi \\ & + 2u_1 \dot{\phi} \sin\theta \sin\phi + u_1 \dot{\theta} \dot{\phi} \cos\theta \sin\phi \\ & - u_1 \dot{\phi} \dot{\theta} \cos\theta \sin\phi - u_1 \dot{\phi}^2 \sin\theta \cos\phi). \end{aligned} \tag{74}$$

Acknowledgements

We gratefully acknowledge support from the National Science Foundation (NSF) and the Air Force Office of Scientific Research (AFOSR). We would like to thank Aveek Das and John Spletzer for their help in image processing and feature extractor. We also would like to thank all the reviewers for their valuable comments.

References

Altuğ, E., Ostrowski, J. P., and Mahony, R. 2002. Control of a quadrotor helicopter using visual feedback. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May, pp. 72–77.

Altuğ, E., Ostrowski, J. P., and Taylor, C. 2003. Quadrotor control using dual visual feedback. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September, pp. 4294–4299.

Amidi, O. 1996. An autonomous vision guided helicopter. Ph.D. Thesis, Carnegie Mellon University.

Ansar, A., Rodrigues, D., Desai, J., Daniilidis, K., Kumar, V., Campos, M. 2001. Visual and haptic collaborative telepresence. *Computer and Graphics* 25(5):789–798 (special issue on mixed realities beyond convention).

Gessow, A. and Myers, G. 1967. *Aerodynamics of the Helicopter*, 3rd edition, Frederick Ungar, New York.

Hamel, T. and Mahony, R. 2000. Visual servoing of a class of underactuated dynamic rigid-body systems. *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, Vol. 4, pp. 3933–3938.

Hamel, T., Mahony, R., and Chriette, A. 2002. Visual servo trajectory tracking for a four rotor VTOL aerial vehicle. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May, pp. 2781–2786.

Hauser, J., Sastry, S., and Meyer, G. 1992. Nonlinear control design for slightly non-minimum phase systems: application to V/STOL aircraft. *Automatica* 28(4):665–679.

Koo, T. J., Hoffmann, F., Sinopoli, B., and Sastry, S. 1998. Hybrid control of an autonomous helicopter. *Proceedings of IFAC Workshop on Motion Control*, Grenoble, France, September.

Kroo, I. and Printz, F. 2005. Mesicopter project, Stanford University, http://aero.stanford.edu/mesicopter.

Ma, Y., Kosecká, J., and Sastry, S. 1998. Optimal motion from image sequences: a Riemannian viewpoint. Research Report, University of California Berkeley, UCB/ER/M98/37.

Martin, P., Devasia, S., and Paden, B. 1996. A different look at output tracking control of a VTOL aircraft. *Automatica* 32:101–107.

Murphy, R., Ausmus, M., Bugajska, M., Ellis, T., Johnson, T., Kelley, N., Kiefer, J., and Pollock, L. 1999. Marsupial-like mobile robot societies. *Proceedings of the 3rd International Conference on Autonomous Agents*, Seattle, WA, pp. 364–365.

Prouty, R. 1995. *Helicopter Performance, Stability, and Control*, Krieger, Malabar, FL.

Sastry, S. 1999. *Nonlinear Systems; Analysis, Stability and Control*, Springer-Verlag, New York.

Sharp, C. S., Shakernia, O., and Sastry, S. S. 2001. A vision system for landing an unmanned aerial vehicle. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, May.

Shim, H. 2000. Hierarchical flight control system synthesis for rotorcraft-based unmanned aerial vehicles. Ph.D. Thesis, University of California, Berkeley.