18

Vision Based Control of Model Helicopters

Erdinç Altuğ[1], James P. Ostrowski[2], Camillo J. Taylor[3]
[1]*Istanbul Technical University, Turkey*
[2]*Evolution Robotics, USA*
[3]*University of Pennsylvania, USA*

1. Introduction

The purpose of this paper is to explore control methodologies and vision algorithms to develop an autonomous unmanned aerial vehicle (UAV). UAVs include unmanned aircrafts, helicopters, blimps and other flying vehicles. An autonomous UAV brings enormous benefits and is suitable for applications like search and rescue, surveillance, remote inspection, military applications, therefore saving time, reducing costs and keeping human pilots away from dangerous flight conditions. UAVs are especially useful when (i) the working environment is inaccessible or hard to reach (planetary environments), (ii) flight is dangerous (due to war, contaminated environmental conditions), (iii) flight is monotonous, (vi) flight time is extended (atmospheric observations, data relay), (v) flight is not possible even by a skilled pilot (movie making, flight of experimental vehicles).

Various unmanned vehicles are in service in military and civilian areas. Fixed-wing vehicles have long-range since they are energy efficient, but they lack the maneuverability required for many UAV tasks. Blimps are easy to control when there are fewer disturbances like wind, and lift comes from natural buoyancy, but they lack maneuverability as well. The rotary-wing aerial vehicles (also called rotorcraft) - such as helicopters - have distinct advantages over conventional fixed-wing aircraft and blimps on surveillance and inspection tasks, since they can take-off and land in limited space and can easily hover above any target. Moreover, helicopters have the advantage of superior maneuverability. Unfortunately, this advantage comes from the dynamically unstable nature of the vehicle and it makes helicopters very hard to control. Sophisticated sensors, fast on-board computation, and suitable control methods are required to make rotorcraft based UAV stable.

Autonomy defined as "the quality or state of being self-governing". Most of the commercial UAVs involve little or no autonomy. The goal is to increase the level of autonomy to fully autonomous operation including take-off, landing, hover (if platform capable of), way point navigation to more advanced autonomy modes such as searching, avoiding danger, combat, refueling, returning to base, etc. In addition, autonomy requires cooperation and communication with other vehicles. Currently a military UAV is supported with almost a dozen personnel to perform piloting, communications, and to control payload systems. The goal in the future is to reduce the personnel/UAV ratio (which is greater-equal to one currently) to lower than one, meaning that one personnel controlling various UAVs. Moreover, the vehicles will be autonomous to control their actions, interact with other vehicles to perform missions. This can be a commander controlling a fleet of military vehicles for combat, or a group of fire fighting UAVs commanded to extinguish a fire.

Source: Mobile Robots, Moving Intelligence, ISBN: 3-86611-284-X, Edited by Jonas Buchli, pp. 576, ARS/plV, Germany, December 2006

In order to create an autonomous UAV, precise knowledge of the helicopter position and orientation is needed. In the previous work involving autonomous flying vehicles, this information was obtained from Inertial Navigation Systems (INS), Global Positioning Systems (GPS) or other sensors like sonar sensor. Typically, multiple sensors are used to overcome limitations of individual sensors, thereby increasing the reliability and decreasing the errors. Vision sensors are primarily used for estimating the relative positions of some target, like a landing site or a ground vehicle. In other words, vision is used to answer the question "Where is the target?". The basic reason for this is the fact that special objects can be easily identified on the visual data relatively fast. Unfortunately, the vision system is not as fast as a gyro, and it is not as reliable as other sensors due to sensitivity to changes in lighting conditions. Vision is used for various research projects involving flying vehicles (Amidi, 1996; Shim, 2000; Sharp et al., 2001). In these papers, vision systems consisting of a single on-board camera or stereo cameras have been used, and the estimates were obtained by combining image data with readings from the inertial navigation systems, GPS or gyros. Our primary goal is to investigate the possibility of a purely vision-based controller. Limited payload capacity may not permit the use of heavy navigation systems or GPS. Moreover, GPS does not work in indoor and clustered environments. One can still setup an indoor GPS system with beacons or use small navigation systems though, cost considerations limit the use of these systems. Vision information can also be used to stabilize, to hover the helicopter, and also to track a moving object. In other words, we are asking the questions "Where is it?'" and "Where am I?" at the same time. This study utilizes a two-camera system for pose estimation. Unlike previous work that utilizes either monocular views or stereo pairs, our two cameras are set to see each other. A ground camera with pan-tilt capabilities and an on-board camera are used to get accurate pose information.

There are various configurations of rotorcrafts. Conventional main rotor/tail rotor configuration; single rotor configuration; coaxial twin rotor configuration; side-by-side twin rotor configuration and multi-rotor configuration. The most popular configuration is the conventional main rotor/tail rotor configuration. This configuration has good controllability and maneuverability. However, the mechanical structure is complex and it requires a large rotor and a long tail boom (Castillo et al., 2005). Our interest in this paper will be on multi-rotor rotorcrafts. We have selected a remote-controlled, commercially available multi-rotor helicopter as our test bed. A quadrotor is a four-rotor helicopter, shown in Figure 1. The first full-scale quadrotor has been built by De Bothezat in 1921 (Gessow & Myers, 1967). It is an under-actuated, dynamic vehicle with four input forces and six output coordinates. Unlike regular helicopters that have variable pitch angle rotors, a quadrotor helicopter has four fixed-pitch angle rotors. Advantages of using a multi-rotor helicopter are the increased payload capacity and high maneuverability. However, multi-rotor helicopters are disadvantaged due to the increased helicopter weight and increased energy consumption due to the extra motors. The basic motions of a quadrotor are generated by varying the rotor speeds of all four rotors, thereby changing the lift forces. The helicopter tilts towards the direction of the slow spinning rotor, which enables acceleration along that direction. Therefore, control of the tilt angles and the motion of the helicopter are closely related and estimation of orientation (roll and pitch) is critical. Spinning directions of the rotors are set to balance the moments and eliminate the need for a tail rotor. This principle is also used to produce the desired yaw motions. A good controller should properly arrange the speed of each rotor so that only the desired states change.

Fig. 1. A commercially available four-rotor rotorcraft, Quadrotor.

Recent work in quadrotor design and control includes the quadrotor (Altuğ, 2003), and X4-Flyer (Hamel et al., 2002). Moreover, related models for controlling the VTOL aircraft are studied by Hauser et al. (1992), and Martin et al. (1996). The main concentration of this study is to use non-linear control techniques to stabilize and perform output-tracking control of a helicopter using vision based pose estimation.

2. Computer Vision

The estimation of motion (relative 3D position, orientation, and velocities) between two frames is an important problem in robotics. For autonomous helicopters, estimation of the motion of objects relative to the helicopter is important as well as estimation of the motion of the helicopter relative to a reference frame. This information is critical for surveillance and remote inspection tasks or for autonomous landing - taking off from a site. This information can be obtained using on-board sensors (like INS, GPS) or cameras. Usually the best sensor can be chosen based on the specific application. For a pose estimation in space for docking operations a camera system would be necessary since, other sensors like INS or GPS are not functional at space. Similarly, for a surveillance UAV used for military purposes, the estimation should not depend entirely on GPS or active sensors that could be manipulated, detected, or disturbed by the enemy.

The pose estimation problem has been a subject of many research projects for many years. The methods proposed use single-vision cameras, stereo cameras or direct 3D measuring techniques such as sonar sensors or laser range finders. Most of the pose estimation techniques are image based and they fall into these two categories: (i) point-based methods and (ii) model-based methods. Point-based methods use the feature points identified on a 2D image while model-based methods use the geometric models (e.g. lines, curves) and its image to estimate the motion. Moreover, the image based pose estimation (IBPE) methods that are point based can also be divided into two categories based on the number of the

cameras used: i) Single-cam methods and ii) Dual-camera methods. In this paper, we will describe the direct method, which is a single-cam method, and the two-camera method, which is a dual-camera method.

For our project, the goal is to obtain the pose from vision rather than complex navigation systems, INS or GPS. We are interested in point-based techniques that are real-time. For this purpose, pair of color cameras are being used to track the image features. These cameras track multi-color blobs located under the helicopter and ground. These blobs are located on a known geometric shape as shown in Figure 2. A blob-tracking algorithm is used to obtain the positions and areas of the blobs on the image planes. Therefore the purpose of the pose estimation algorithm is to obtain (x, y, z) positions, tilt angles (θ, ψ), the yaw angle (ϕ) and the velocities of the helicopter in real-time relative to the ground camera frame.

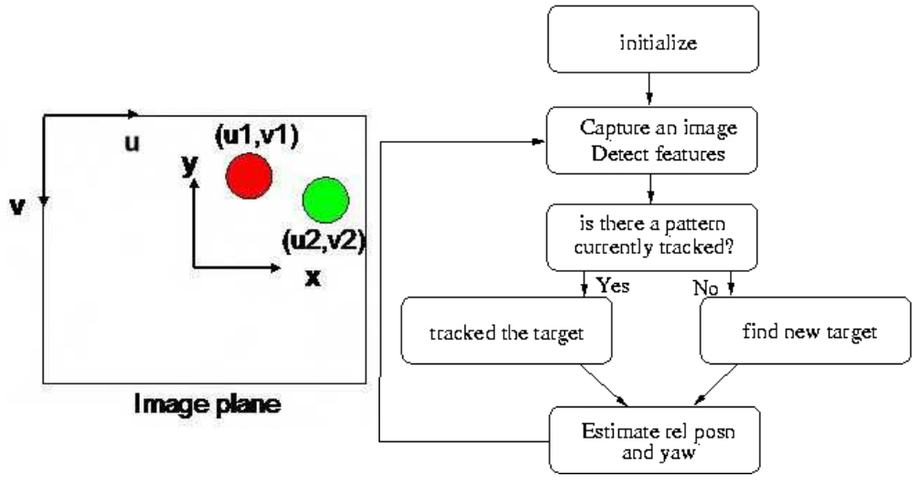

Fig. 2. Feature based pose estimation using color blobs (left). Tracking algorithm to estimate relative motion (right).

For the estimation of the pose (x, y, z, and heading) of a flying robot, such as a blimp or a helicopter, an on-board camera, and multi-color blobs that are equally spaced grids on the floor can be used. A ground camera can also be used for the pose estimation of a flying vehicle. If the pitch and the roll angles are approximately zero, in that case two blobs will be enough for successful pose estimation. The way such a pose estimation method works is, the blobs are tracked with a ground camera and the blob separation on image plane is compared to distance L, the real blob separation, to estimate the altitude. Tracking is the action where a particular blob's whereabouts are known by successfully identifying it all time steps. Estimation of the relative motion and the absolute position and yaw needs a systematic approach; estimate the position of the pattern at each time step and update the absolute position of the pattern based on the estimated motion of the pattern. The biggest disadvantage of such a ground based pose estimation method is the fact that the estimation is limited to camera view area. A pan/tilt camera can be used to not only estimate the pose but also track the pattern as it moves. This increases the limited view area of the ground

camera. With the input of the pan and tilt angles, measured from the camera, the estimated relative position values should be translated due to the motion of the camera system.
The pose estimation can be defined as finding a rotation Matrix, R, $R \in SO(3)$, defining the body fixed frame of the helicopter with respect to the fixed frame located at the ground camera frame, where $R^T R = I$, $\det(R) = 1$, and also the relative position $\bar{p} \in R^3$ of the helicopter with respect to the ground camera and also the velocities w and V of the helicopter as shown in Figure 3.

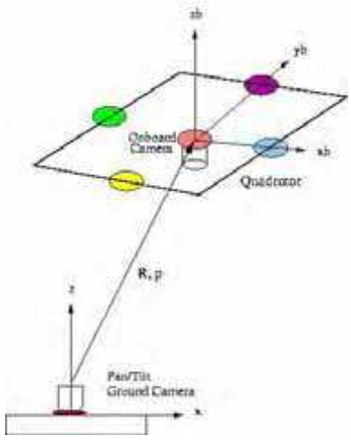

Fig. 3. Quadrotor helicopter pose estimation and tracking using color blobs.

In this section, we will introduce two methods to estimate pose of the helicopter in real-time. These methods are the direct method and the two-camera method. The methods will then be compared in simulations.

2.1. Direct Method

The purpose of the direct 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. Four different color blobs can be placed as a square pattern under the helicopter as shown in Figure 4. A ground camera is used to track the blobs to estimate the helicopter pose. Input of the camera intrinsic parameters (f_x, f_y, O_x, O_y) and image coordinates of the blobs are required. Moreover, the blob size and blob separation *L* is predetermined. A blob-tracking algorithm can be used to get the positions and areas of the blobs on the image plane (u_i, v_i, A_i).The position of each blob with respect to fixed frame is calculated as

$$z_i = \frac{(f_x + f_y)\sqrt{C\pi}}{2\sqrt{A_i}}, \quad x_i = (u_i - O_x)\frac{z_i}{f_x}, \quad y_i = (v_i - O_y)\frac{z_i}{f_y} \tag{1}$$

where *C* is the number of pixels per unit area. The position of the helicopter is estimated by averaging the four blob positions. Normalization is performed using the real center difference between blobs. The yaw angle, ϕ, can be obtained from blob positions and the tilt angles can be estimated from the height differences of the blobs.

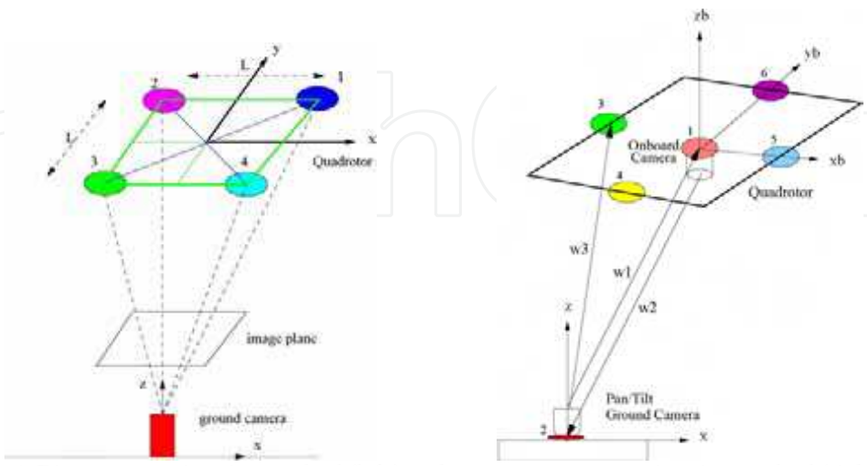

Fig. 4. Direct pose estimation method (left) and two-camera pose estimation method (right)

$$\phi = \text{atan}(y_1 - y_3 / x_1 - x_3), \quad \psi = \text{asin}(z_3 - z_2 / L), \quad \theta = \text{asin}(z_2 - z_1 / L) \tag{2}$$

These estimates depend on area calculations; therefore, they are sensitive to noise. This method works well on estimating yaw angle and positions of the helicopter, but it suffers greatly on estimating the tilt angles.

2.1. 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 at 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 are attached to the bottom of the quadrotor and to the ground camera as shown in Figure 4. 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 all of the features. The cameras have matrices of intrinsic parameters, A_1 and A_2 Let be $\vec{w}_i$ a unit vector from each camera to the blobs, and λ_i unknown scalars.

$$\vec{w}_i = inv(A_1)\cdot[u_i \quad v_i \quad 1]^{'}, \ \vec{w}_i = \vec{w}_i / norm(\vec{w}_i) \ \text{ for i=1,3,4,5,6} \tag{3}$$

$$\vec{w}_2 = inv(A_2)\cdot[u_2 \quad v_2 \quad 1]^{'}, \vec{w}_2 = \vec{w}_2 / norm(\vec{w}_2) \tag{4}$$

Let $\vec{L}_a$ be the vector pointing from blob-1 to blob-3 in Figure 4. Vectors $\vec{w}_1$ and $\vec{w}_3$ are related by

$$\lambda_3 \vec{w}_3 = \lambda_1 \vec{w}_1 + R\vec{L}_a \tag{5}$$

To simplify, let us take the cross product of this equation with $\vec{w}_3$

$$\lambda_1(\vec{w}_3 \times \vec{w}_1) = (R\vec{L}_a \times \vec{w}_3) \tag{6}$$

This can be rewritten as

$$(\vec{w}_3 \times \vec{w}_1) \times (R\vec{L}_a \times \vec{w}_3) = 0 \tag{7}$$

In order to solve the above equation, let the rotation matrix *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

$$R = Rot(\vec{w}_1 \times \vec{w}_2, \theta) \cdot Rot(\vec{w}_1, \alpha) \tag{8}$$

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 $\theta = a\cos(\vec{w}_1 \cdot \vec{w}_2)$. Alternatively, one can use the cross product of $\vec{w}_1$ and $\vec{w}_2$, to solve θ angle. The only unknown left in Equation 8 is the angle α. Rewriting Equation 7 gives

$$(\vec{w}_3 \times \vec{w}_1) \times (\vec{w}_3 \times (Rot(\vec{w}_1 \times \vec{w}_2, \theta) \cdot Rot(\vec{w}_1, \alpha))\vec{L}_a) = 0. \tag{9}$$

Let *M* be given as

$$M = (\vec{w}_3 \times \vec{w}_1) \times \{\vec{w}_3 \times [R(\vec{w}_1 \times \vec{w}_2, \theta)]\} = 0\,. \tag{10}$$

Using Rodrigues' formula, $Rot(\vec{w}_1, \alpha)$ can be written as

$$Rot(\vec{w}_1, \alpha) = I + \hat{w}_1 \sin\alpha + \hat{w}_1^2(1 - \cos\alpha) \tag{11}$$

Pre-multiplying Equation 11 with *M* and post-multiplying it with $\vec{L}_a$ gives the simplified version of the Equation 9

$$M \cdot \vec{L}_a + \sin\alpha \cdot M\vec{w}_1 \cdot \vec{L}_a + (1 - \cos\alpha) \cdot M \cdot (\vec{w}_1)^2 \cdot \vec{L}_a = 0\,. \tag{12}$$

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

$$\alpha = a\sin\frac{B \cdot C \pm \sqrt{(B^2 \cdot C^2 - (A^2 + B^2) \cdot (C^2 - A^2)}}{A^2 + B^2}\,. \tag{13}$$

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. The estimated rotation matrix will be found from Equation 8. Euler angles (ϕ, θ, ψ) defining the orientation of the quadrotor can be obtained from the rotation matrix, *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. The λ_1 can be found using Equation 6. The other λ_i values can be found from the relation of the blob positions

$$\lambda_i \vec{w}_i = \lambda_1 \vec{w}_1 + R\vec{L}_i \tag{14}$$

where $\vec{L}_i$ is the position vector of the blob i, in body-fixed frame. To reduce the errors, λ_i values are normalized using the blob separation, *L*. The center of the quadrotor will be

$$[X \quad Y \quad Z]' = [\lambda_3 \vec{w}_3 + \lambda_4 \vec{w}_4 + \lambda_5 \vec{w}_5 + \lambda_6 \vec{w}_6]/4\,. \tag{15}$$

2.3 Comparing the Pose Estimation Methods

The proposed direct method and the two-camera pose estimation methods are compared to other methods using a MATLAB simulation. Other methods used were a four-point algorithm (Ansar et al., 2001), a state estimation algorithm (Sharp et al., 2001), and a stereo pose estimation method that uses two ground cameras that are separated by a distance d. The errors are calculated using angular and positional distances, given as

$$e_{ang} = \left|\log(R^{-1} \cdot R^{est})\right| \quad e_{pos} = \left|\vec{p} - \vec{p}^{\,est}\right| . \tag{16}$$

R^{est} and $\vec{p}^{\,est}$ are the estimated rotational matrix and the position vector. Angular error is the amount of rotation about a unit vector that transfers R to R^{est}. In order to compare the pose estimation methods, a random error up to five pixels was added on image values. The blob areas were also added a random error of magnitude ± 2. During the simulation helicopter moves from the point (22, 22, 104) to (60, 60, 180) cm, while (θ,ψ,ϕ) angles change from (0.7, 0.9, 2) to (14, 18, 40) degrees. The comparison of the pose estimation methods and the average angular and positional errors are given on Table 1. The values correspond to average errors throughout the motion of the helicopter.

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

Table 1. Comparison of the Pose Estimation Methods using the angular and positional distances.

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 due to noisy blob area readings. For the stereo method, the value of the baseline is important for pose estimation. The need for a large baseline for stereo pairs is the drawback of the stereo method. 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.

3. 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, so that we can derive control methodologies that would stabilize and control its motions. As shown in Figure 5, quadrotor helicopter has two clock-wise rotating and two counter-clock-wise rotating rotors, which eliminates the need for a tail rotor. Basic motions of the quadrotor are achieved by the trusts generated by its four rotors.

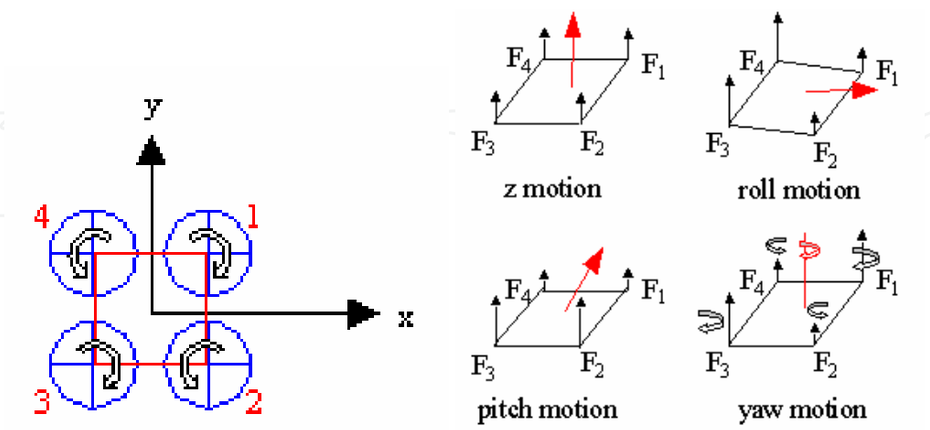

Fig. 5. Quadrotor helicopter can be controlled by individually controlling the rotor thrusts.

For a rigid body model of a 3D quadrotor given in Figure 6, 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 $\bar{p} = [x \quad y \quad z]' \in O$ where O is the inertial frame and a rotation matrix $R : O \to B$, where $R \in SO(3)$. 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.

$$RPY(\phi, \theta, \psi) = Rot(z, \phi) \cdot Rot(y, \theta) \cdot Rot(x, \psi) \tag{17}$$

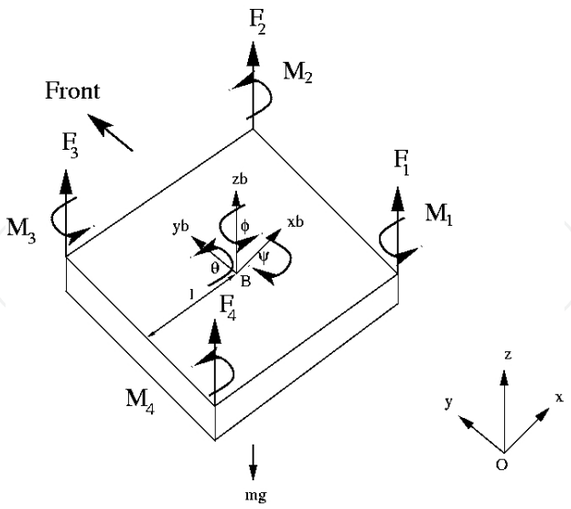

Fig. 6. 3D quadrotor helicopter rigid body model.

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, $\vec{\zeta} = [\psi \quad \theta \quad \phi]^T$ and $\vec{w}^b = [p \quad q \quad r]^T$. The body angular velocity is related to Euler angular velocity by $\vec{w}^b = unskew(R^T \dot{R})$, where *unskew()* term, represents obtaining vector $\vec{w}^b$ from 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, $\dot{\vec{\zeta}} = J\vec{w}^b$

$$\begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 1 & S_\psi T_\theta & C_\psi T_\theta \\ 0 & C_\psi & -S_\psi \\ 0 & S_\psi / C_\theta & C_\psi / C_\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{18}$$

where S_ϕ denotes $Sin(\phi)$, C_ϕ denotes $Cos(\phi)$, and T_θ denotes $Tan(\theta)$. Using the Newton-Euler equations, we can represent the dynamics of the quadrotor as follows

$$\dot{\vec{V}} = \frac{1}{m}\vec{F}_{ext} - \vec{w}^b \times \vec{V}^b, \quad I_b \dot{\vec{w}}^b = \vec{M}_{ext} - \vec{w}^b \times I_b \vec{w}^b, \quad \dot{\vec{\zeta}} = J\vec{w}^b \tag{19}$$

where I_b is the inertia matrix, F_{ext} and M_{ext} are the external forces and moments on the body fixed frame given as

$$\vec{F}_{ext} = -drag_x \vec{i} - drag_y \vec{j} + (T - drag_z)\vec{k} - R \cdot mg\vec{k}, \quad \vec{M}_{ext} = M_x \vec{i} + M_y \vec{j} + M_z \vec{k}. \tag{20}$$

In this equation, *T* is the total thrust, M_x, M_y, and M_z are the body moments, $\vec{i}, \vec{j}, \vec{k}$ are the unit vectors along *x*, *y* and *z* axes respectively. A drag force acts on a moving body opposite to the direction it moves. The terms $drag_x, drag_y, drag_z$ are the drag forces along the appropriate axis. Let 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 $drag = \frac{1}{2} C_d \rho V^2 A$. Assuming the density of air is constant then, the constants at above equation can be collected, and the equation can be written as $drag = C_d V^2$. The total thrust is (Prouty, 1995)

$$F = bL = \frac{\rho}{4} \omega^2 R^3 abc(\theta_t - \phi_t) \tag{21}$$

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, ω is the rotor speed, and R is the rotor radius. Note that the angle θ_t is constant for a quadrotor helicopter that has fixed pitch rotors. In addition, 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 $F_i = D\omega_i^2$. Successful control of the helicopter requires direct control of the rotor speeds, ω_i. Rotor speeds can be controlled by controlling

the motor torque. The torque of motor i, M_i, is related to the rotor speed ω_i as $M_i = I_r\omega_i^2 + K\omega_i^2$, where I_r is the rotational inertia of rotor i, 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

$$\begin{pmatrix} T \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} l & l & l & l \\ -l & -l & l & l \\ -l & l & l & -l \\ C & -C & C & -C \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} \tag{22}$$

where F_i's are the forces generated by the rotors. The matrix above which we denote by $N \in R^{4\times4}$ is full rank for $l, C \neq 0$. This is logical since $C = 0$ would imply that the moment around 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, which again implies 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 ω_i, which will change the thrust and the body moments in Equation 22. This will change the external forces and moments in Equation 20. This will lead to the desired body velocities and accelerations as given in Equation 19.

4. Helicopter Control

Unmanned aerial vehicles bring enormous benefits to applications like search and rescue, surveillance, remote inspection, military applications and saving human pilots from dangerous flight conditions. To achieve these goals, however, autonomous control is needed. The control of helicopters is difficult due to the unstable, complex, non-linear, and time-varying dynamics of rotorcrafts. Rotor dynamics, engine dynamics, and non-linear variations with airspeed make the system complex. This instability is desired to achieve the set of motions that could not be achieved by a more stable aircraft. In this work, our goal is to use external and on-board cameras as the primary sensors and use onboard gyros to obtain 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. The controller should be able to obtain the relative positions and velocities from the cameras only. The selection of suitable control method for an UAV requires careful consideration of which states need to be observed and controlled, which sensors are needed and the rate of sensors. In this paper, we will introduce 3D quadrotor model and explain the control algorithms that are developed for these vehicles.

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. One can under some assumptions simplify the above model. Such a simplified model will be useful for derivation of the controllers.

Let us assume that

- The higher order terms can be ignored ($\vec{F}_{ext} \gg m\vec{w}^b \times \vec{V}^b$ and $\vec{M}_{ext} \gg \vec{w}^b \times I_b \vec{w}^b$)
- The inertia matrix I_b is diagonal
- The pitch (ψ) and roll (θ) angles are small, so that J in Equation 19 is the identity matrix

This leads to the following dynamical equations

$$\dot{\vec{V}} = \frac{1}{m}\vec{F}_{ext}, \quad I_b \dot{\vec{w}}^b = \vec{M}_{ext}, \quad \dot{\vec{\zeta}} = \vec{w}^b. \tag{23}$$

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

$$\begin{aligned} \ddot{x} &= \left[(\sum_{i=1}^{4} F_i)(C_\phi S_\theta C_\psi + S_\phi S_\psi) - K_1 \dot{x} \right] / m \\ \ddot{y} &= \left[(\sum_{i=1}^{4} F_i)(S_\phi S_\theta C_\psi - C_\phi S_\psi) - K_2 \dot{y} \right] / m \\ \ddot{z} &= \left[(\sum_{i=1}^{4} F_i)(C_\theta C_\psi) - mg - K_3 \dot{z} \right] / m \end{aligned} \tag{24}$$

$$\ddot{\theta} = \frac{l}{J_1}(-F_1 - F_2 + F_3 + F_4), \quad \ddot{\psi} = \frac{l}{J_2}(-F_1 + F_2 + F_3 - F_4), \quad \ddot{\phi} = \frac{l}{J_3}(M_1 - M_2 + M_3 - M_4) \tag{25}$$

The J_i's given above are the moments of inertia with respect to the corresponding axes, and the K_i's 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

$$\begin{aligned} u_1 &= (F_1 + F_2 + F_3 + F_4)/m = T/m \\ u_2 &= (-F_1 - F_2 + F_3 + F_4)/J_1 = T_x/J_1 \\ u_3 &= (-F_1 + F_2 + F_3 - F_4)/J_2 = T_y/J_2 \\ u_4 &= C(F_1 - F_2 + F_3 - F_4)/J_3 = T_z/J_3 \end{aligned} \tag{26}$$

where C is the force-to-moment scaling factor. The 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

$$\begin{array}{lll} \ddot{x} = u_1(C_\phi S_\theta C_\psi + S_\phi S_\psi) & \ddot{y} = u_1(S_\phi S_\theta C_\psi - C_\phi S_\psi) & \ddot{z} = u_1 C_\theta C_\psi \\ \ddot{\theta} = u_2 l & \ddot{\psi} = u_3 l & \ddot{\phi} = u_4 \end{array} \tag{27}$$

Noticing that the motion along the y-axis is related to the ψ tilt angle, similarly motion along the x-axis is related to the θ angle, one can design a backstepping controller. Backstepping controllers (Sastry, 1999) are especially useful when some states are controlled through other states. Similar ideas of using backstepping with visual servoing have also been developed for a traditional helicopter (Hamel & Mahony, 2000).

Considering Equation 27, 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 gives $\ddot{x} = u_1 C_\phi S_\theta \quad \ddot{y} = -u_1 C_\phi S_\psi$. From this equation, backstepping controllers for u_2 and u_3 can be derived (Altuğ et al., 2005). Controller u_2 controls angle θ in order to control x motions and controller u_3 controls angle ψ in order to control motions along the y-axis.

$$u_2 = \frac{1}{u_1 C_\theta C_\phi}(-5x - 10\dot{x} - 9u_1 S_\theta C_\phi - 4u_1 \dot{\theta} C_\theta C_\phi + u_1 \dot{\theta}^2 S_\theta C_\phi + 2u_1 \dot{\phi} S_\theta S_\phi - u_1 \dot{\phi}^2 S_\theta C_\phi) \tag{28}$$

$$u_3 = \frac{1}{u_1 C_\psi C_\phi}(-5y - 10\dot{y} - 9u_1 S_\psi C_\phi - 4u_1 \dot{\psi} C_\psi C_\phi + u_1 \dot{\psi}^2 S_\psi C_\phi + 2u_1 \dot{\psi} S_\psi S_\phi - u_1 \dot{\phi}^2 S_\psi C_\phi) \tag{29}$$

The sequential derivation of the backstepping controller involves finding suitable Lyapunov functions therefore; the controllers are guaranteed to exponentially stabilize the helicopter.
PD controllers on the other hand, can control the altitude and the yaw.

$$u_1 = \frac{g + K_{p1}(z_d - z) + K_{d1}(\dot{z}_d - \dot{z})}{C_\theta C_\psi}, \quad u_4 = K_{p2}(\phi_d - \phi) + K_{d2}(\dot{\phi}_d - \dot{\phi}) \tag{30}$$

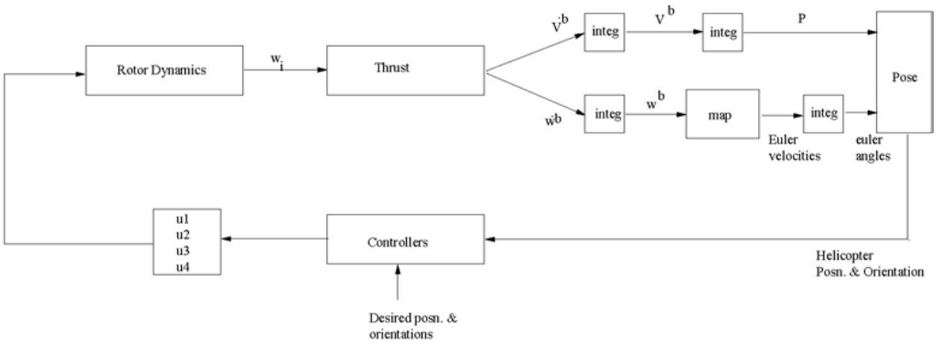

Fig. 7. Helicopter simulation model developed in MATLAB Simulink.

The proposed controllers are implemented on a MATLAB, Simulink simulation as shown in Figure 7. The helicopter model is based on the full non-linear model given by Equation 19. 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/m², I_y = 0.0142 kg/m² and I_z = 0.0071 kg/m². 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/s². The thrust forces in real flying vehicles are limited. Therefore, the maximum and minimum inputs are defined by

$$\frac{-F_{\max}}{m} \le u_1 \le \frac{4F_{\max}}{m}, \quad \frac{-2F_{\max}}{l} \le u_2 \le \frac{2F_{\max}}{l}, \quad \frac{-2F_{\max}}{l} \le u_3 \le \frac{2F_{\max}}{l}, \quad -2CF_{\max} \le u_4 \le 2CF_{\max} \tag{31}$$

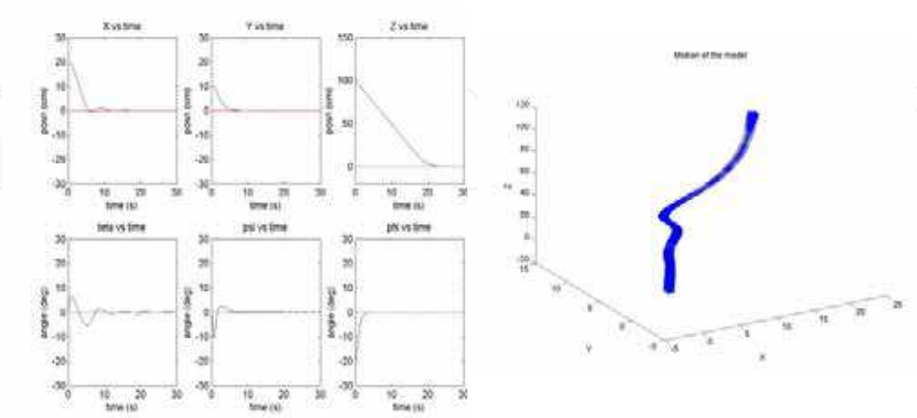

Fig. 8. Helicopter control simulation, using backstepping and PD controllers, and helicopter motion during this simulation.

In the first simulation, helicopter is being controlled with the proposed PD and backstepping controllers. The simulation results in Figure 8 shows the motion of the quadrotor from position (20, 10, 100) to the origin, while reducing the yaw angle from -20 to zero degrees.
The controller should be strong enough to handle random errors that are occurring because of the pose estimation error and disturbances. In order to simulate the robustness of the controllers to error, random error have been introduced on x, y, z, and yaw values. Error on x, y was with variance of 0.5 cm., error on z was with variance of 2 cm., and error on yaw was with variance of 1.5 degrees. In the simulation helicopter moves from 100 cm. to 150 cm. while reducing the yaw angle from 30 degrees to zero as shown in Figure 9. Although there were considerable error on the states, controllers were able to stabilize the helicopter. The mean and standard deviation are found to be 150 cm. and 1.7 cm. for z and 2.4 and 10.1 degrees for yaw respectively.

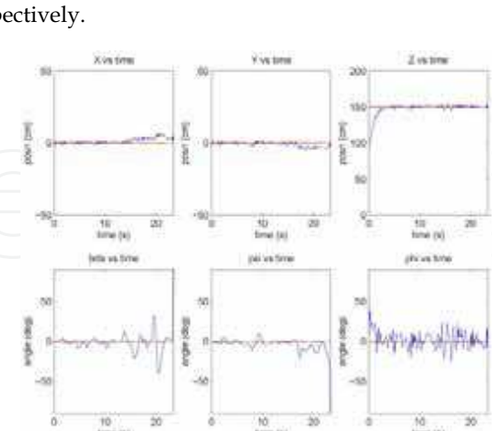

Fig. 9. Backstepping controller simulation with random noise at x, y, z and ϕ values.

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 much higher rate than the vision updates such as accelerometer, or potentiometer, the readings – images - has 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 rates of many cameras are 20 to 30 Hz. A frame rate of 15 Hz. will be used for the overall sampling rate of this sensory system. The Figure 10 shows the results of the simulation, where the x, y and z positions are sampled at 15 Hz. The controllers are robust enough to handling the discrete inputs. A simple comparison of the plots shows that discrete sampling causes an increased settling time.

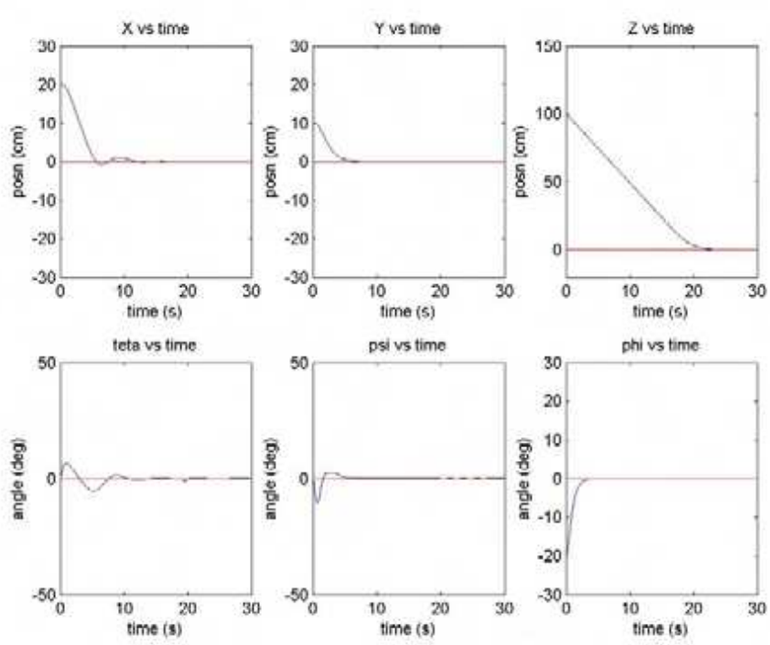

Fig. 10. The effect of the delay on the simulation.

Considering the simulations performed, PD control was successful to control altitude and heading of the helicopter. 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.

5. Experiments

This section discusses the applications of the image based pose estimation and control methods. One logical starting point is to decide where to put cameras, how many cameras

to use, location of the vision computer and computation time. If a remote computer will process the images, transferring images to this computer and transfer of commands to the helicopter will be required. The use of an onboard camera and processing images locally not only eliminates the need of information transfer, but also is very useful for other task that are usually required by the vehicle, such as locating a landing pad. The disadvantage of this is the increased vehicle weight, the need of more powerful computers onboard the vehicle.

The proposed algorithms implemented on a computer vision system. We used off the shelf hardware components for the system. Vision computer is a Pentium 4, 2 GHz machine that had an Imagination PXC200 color frame grabbers. Images can be captured at 640×480 resolution at 30 Hz. The camera used for the experiments was a Sony EVI-D30 pan/tilt/zoom color camera. The algorithm depends heavily on the detection of the color blobs on the image. When considering color images from CCD cameras, there are a few color spaces that are common, such as 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 channel. Color tables are generated for each color in MATLAB. Multiple images and various lighting 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. We use the blob tracker routines. The blob tracking routings use the images and the pregenerated color tables to identify the color blobs in real-time. It returns 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 could be able to track the blobs at about 20 Hz.

To make a helicopter fully autonomous, we need a flight controller as shown in Figure 11. An off-board controller receives camera images, processes them, and sends control inputs to the on-board processor. 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.

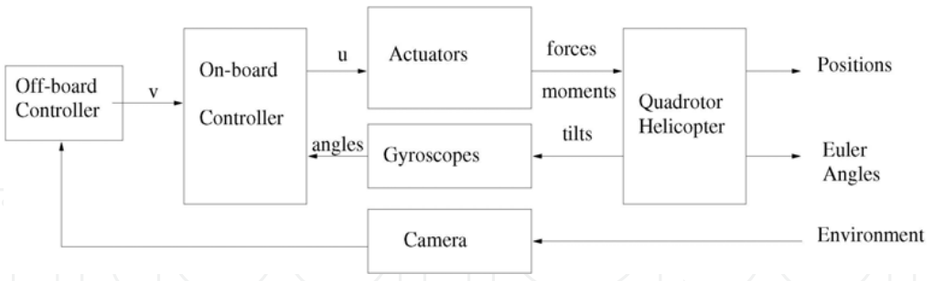

Fig. 11. Diagram of the on-board controller.

The off-board controller (the ground system) is responsible for the main computation. It processes the images, sets the goal positions, and sends them to the on-board controller using the remote controller transmitter as shown in Figure 12.

The proposed controllers and the pose estimation algorithms have been implemented on a remote-controlled battery-powered helicopter shown in Figure 13. 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 three minutes flight time. This helicopter has three gyros on board to stabilize itself. An experimental setup shown in Figure 13 was prepared to prevent the helicopter from moving too much on the x-y plane, while enabling it to turn and ascend/descend freely. Vision based stabilization experiments were performed using two different methods; direct method, which is using a single ground camera, and the two-camera pose estimation method.

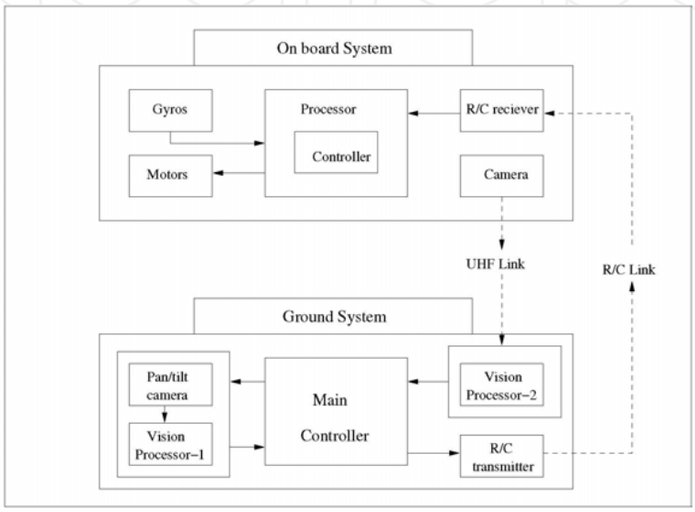

Fig. 12. Experimentation system block diagram, including the helicopter and the ground station.

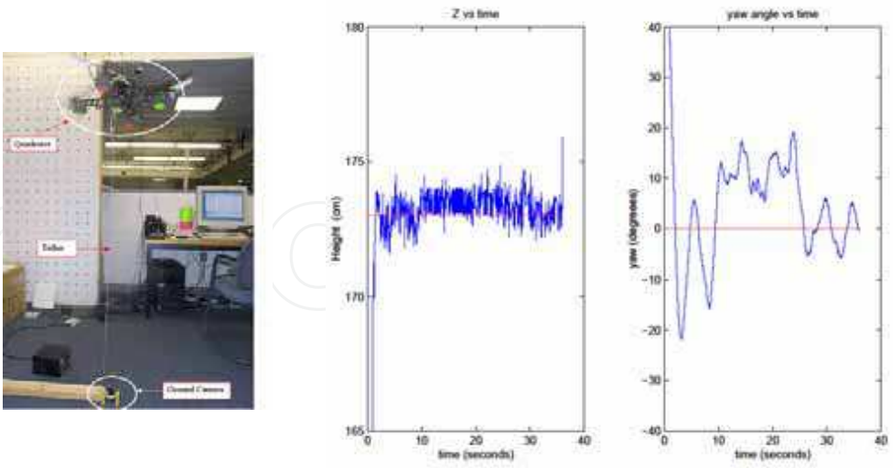

Fig. 13. Altitude and yaw control experiment performed with a single ground camera based on direct pose estimation method, and the results.

The first experiment involves the use of a single ground camera and the direct pose estimation method. The helicopter pose is estimated using image features, as well as areas of the image features. The goal is to control the helicopter at the desired altitude and the desired heading. In this experiment altitude and yaw angle are being controlled with PD controllers. Figure 13 shows the results of the altitude and yaw control experiment using this method.

The second experiment is the control of the quadrotor helicopter using the two-camera pose estimation method. In this experiment, two separate computers were used. Each camera was connected to separate computers that were responsible for performing blob tracking. PC-1 was responsible for image processing of the on-board camera. The information is then sent to PC-2 via the network. PC-2 was responsible for the ground pan/tilt camera control, image processing, and calculation of the control signals for the helicopter. These signals were then sent to the helicopter with a remote control device that uses the parallel port. The backstepping controllers for x and y motions and PD controllers for altitude and heading were implemented for the experiment. Figure 14 shows the results of this experiment using the two-camera pose estimation method. The mean and standard deviation are found to be 106 cm. and 17.4 cm. for z, 4.96 degrees, and 18.3 degrees for heading 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.

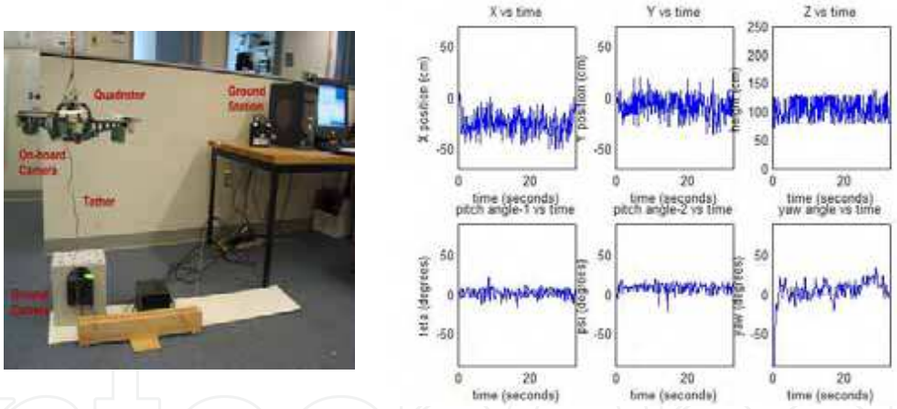

Fig. 14. The experimental setup and the results of the height x, y and yaw control experiment with two-camera pose estimation method.

6. Conclusions and Future Work

In this work, we have presented pose estimation algorithms and non-linear control techniques to build and control autonomous helicopters. We introduce a novel two-camera method for helicopter pose estimation. The method has been compared to other pose estimation algorithms and shown to be more effective, especially when there are errors on the image plane. A three dimensional quadrotor rotorcraft model has been developed. Nonlinear backstepping and PD controllers have been used to stabilize and

perform output-tracking control on the 3D quadrotor model. With the simulations performed on MATLAB - Simulink, controllers shown to be effective even when there are errors in the estimated vehicle states. Even in the presence of large errors in camera calibration parameters or in image values, global convergence can be achieved by the controllers. The proposed controllers and the pose estimation methods have been implemented on a remote control, battery-powered, model helicopter. Experiments on a tethered system show that the vision-based control is effective in controlling the helicopter.

Vision system is the most critical sensory system for an aerial robot. No other sensor system can supply relative position and relative orientation information like it. Especially tracking a moving target can only be possible with a vision system. One of the drawbacks of the vision system is that, it is not reliable when the lighting on the scene changes, and it is sensitive to vibration. In addition, weight and power consumption are other important parameters limiting the use of vision in mini UAVs. With recent advances in microcomputers and cameras, it will become further possible to achieve real-time feature extraction and control with commercial of the shelf parts. Our future work will concentrate on development of a bigger helicopter UAV for outdoor flight. A Vario model helicopter has already been integrated with a suite of sensors including IMU, GPS, barometric pressure altimeter, sonar, camera, and a Motorola MPC-555 controller. A helicopter ground test platform has been developed to test the helicopter system. Our future work will be to further explore control and vision algorithms using this helicopter. It is expected that further improvement of the control and the vision methods will lead to highly autonomous aerial robots that will eventually be an important part of our daily lives.

7. References

Altuğ, E. (2003). Vision Based Control of Unmanned Aerial Vehicles with Applications to an Autonomous Four Rotor Helicopter, Quadrotor. Ph.D. Thesis, University of Pennsylvania, USA

Altuğ, E.; Ostrowski, J. P. & Taylor, C. J. (2005). Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback. *The International Journal of Robotics Research,* Vol. 24, No. 5, May 2005, pp. 329-341

Amidi, O. (1996). An Autonomous Vision Guided Helicopter. Ph.D. Thesis, Carnegie Mellon University, USA

Ansar, A.; Rodrigues, D.; Desai, J.; Daniilidis, K.; Kumar, V. & Campos, M. (2001). Visual and Haptic Collaborative Tele-presence. *Computer and Graphics, Special Issue on Mixed Realities Beyond Convention,* Vol. 25, No. 5, pp. 789-798

Castillo, P.; Lozano, R. & Dzul, A. (2005). *Modelling and Control of Mini-Flying Machines*, Springer-Verlag London Limited

Hamel, T.: Mahony, R. (2000). Visual Servoing of a class of under actuated 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. & Chriette, A. (2002). Visual Servo Trajectory Tracking for a Four Rotor VTOL Aerial Vehicle. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, D.C., pp. 2781-2786

Hauser, J.; Sastry, S. & Meyer, G. (1992). Nonlinear Control Design for Slightly Non-Minimum Phase Systems: Application to V/STOL Aircraft. *Automatica*, vol. 28, No:4, pp. 665-679

Gessow, A. & Myers, G. (1967). *Aerodynamics of the helicopter*, Frederick Ungar Publishing Co, New York, Third Edition

Martin, P.; Devasia, S. & Paden, B. (1996). A Different Look at Output Tracking Control of a VTOL Aircraft, *Automatica*, vol. 32, pp. 101-107

Prouty, R. *Helicopter Performance, Stability, and Control*. (1995). Krieger Publishing Company

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

Sharp, C. S.; Shakernia, O. & Sastry, S. S. (2001). A Vision System for Landing an Unmanned Aerial Vehicle, *IEEE Conference on Robotics and Automation*, Seoul, Korea

Shim, H. (2000). Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles. Ph.D. Thesis, University of California, Berkeley

Mobile Robotics, Moving Intelligence

Edited by Jonas Buchli

ISBN 3-86611-284-X

Hard cover, 586 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers many aspects of the exciting research in mobile robotics. It deals with different aspects of the control problem, especially also under uncertainty and faults. Mechanical design issues are discussed along with new sensor and actuator concepts. Games like soccer are a good example which comprise many of the aforementioned challenges in a single comprehensive and in the same time entertaining framework. Thus, the book comprises contributions dealing with aspects of the Robotcup competition. The reader will get a feel how the problems cover virtually all engineering disciplines ranging from theoretical research to very application specific work. In addition interesting problems for physics and mathematics arises out of such research. We hope this book will be an inspiring source of knowledge and ideas, stimulating further research in this exciting field. The promises and possible benefits of such efforts are manifold, they range from new transportation systems, intelligent cars to flexible assistants in factories and construction sites, over service robot which assist and support us in daily live, all the way to the possibility for efficient help for impaired and advances in prosthetics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Erdinc Altug, James P. Ostrowski and Camillo J. Taylor (2006). Vision Based Control of Model Helicopters, Mobile Robotics, Moving Intelligence, Jonas Buchli (Ed.), ISBN: 3-86611-284-X, InTech, Available from: http://www.intechopen.com/books/mobile_robotics_moving_intelligence/vision_based_control_of_model_helicopters

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen