Vision-based Pose Estimation and Control of a Model Helicopter

Erdinç Altuğ
Department of Mechanical Engineering
Istanbul Technical University
Istanbul, TURKEY
Email: altuger@itu.edu.tr

Camillo Taylor
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104, USA
Email: cjtaylor@grasp.cis.upenn.edu

***Abstract*— In this paper, a vision system for pose estimation and stabilization control of a model helicopter has been proposed. This method consists of a pair of ground and onboard cameras, and it is used to estimate the full six degrees of freedom of the helicopter. The pose estimation algorithm is compared through simulation to some other feature based pose estimation methods and is shown to be less sensitive to feature detection errors. The proposed pose estimation algorithm and non-linear control techniques have been implemented on a remote controlled model helicopter.**

I. INTRODUCTION

The purpose of this study is to explore pose estimation algorithms that will make an unmanned aerial vehicle (UAV) autonomous. An autonomous UAV will be suitable for applications like search and rescue, surveillance and remote inspection. Rotary wing aerial vehicles have distinct advantages over conventional fixed wing aircrafts on surveillance and inspection tasks, since they can take-off/land in limited spaces and easily hover above the target. A ***quadrotor*** is a four rotor helicopter. Recent work in quadrotor design and control includes quadrotor [1], X4-Flyer [6], and mesicopter [9]. Also, related models for controlling the VTOL aircraft are studied by Hauser et al. [7].

Quadrotor 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. Disadvantages are 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 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, therefore eliminating the need for a tail rotor. This is also used to produce the desired yaw motions. A good controller should properly arrange the rotor speeds so that only the desired states change.

In order to create an autonomous UAV, precise knowledge of the helicopter position and orientation is needed. This info can be used to stabilize, hover the helicopter or for tracking an object. The pose estimation of a 3D robot has also been studied by [3], [17], [15], [16]. But in these papers, a single onboard camera has 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 on the quadrotor. Limited payload capacity does not permit the use of heavy navigation systems or GPS. Moreover the GPS does not work at indoor environments. One can still setup an indoor GPS system or use small navigation systems but, cost limits the use of these systems. This study utilizes a two camera system for pose estimation. Unlike previous work that either utilizes monocular views or stereo pairs, our two cameras are set to see each other. A ground camera that has pan-tilt capability, and an onboard camera are used to get accurate pose information. The proposed pose estimation algorithm is compared in simulation with other methods like a four-point algorithm [4], a state estimation algorithm [14] and a direct method that uses the area estimations of the blobs. A backstepping like controller [13], [5] shown in [1] has been implemented and shown effective in simulations of the dynamical quadrotor model. The proposed pose estimation algorithm and the control techniques have been implemented on a remote controlled, battery powered model helicopter.

Fig. 1. A Four Rotor Model Helicopter, Quadrotor

II. Helicopter Pose Estimation

For surveillance and remote inspection tasks a relative position and orientation detection is important. Our goal is to obtain the pose from vision rather than complex and heavy navigation systems or GPS. The purpose of the pose estimation method is to obtain the relative position and orientation of the helicopter with respect to the ground camera. Two camera pose estimation method uses a pan/tilt/zoom ground camera and an onboard camera. Previous work on vision based pose estimation utilizes monocular views or stereo pairs. Our two camera pose estimation method involves the use of two cameras that are set to see each other. Colored blobs of 2.5 cm radius are attached to the bottom of the quadrotor and to the ground camera as shown in Figure 2. A blob tracking algorithm is used to get 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) and the yaw angle (ϕ) of the helicopter in real-time relative to the camera frame.

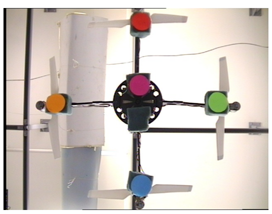

Fig. 2. Quadrotor Tracking with a Camera

The pose estimation can be defined as:

Problem: Find Rotation Matrix, $\mathbf{R} \in R^{3\times3}$, defining the body fixed frame of the helicopter with respect to the fixed frame located at the ground camera frame, where $\mathbf{R}^T\mathbf{R} = \mathbf{I}, det(\mathbf{R}) = \mathbf{I}$, relative position $\vec{p} \in R^3$ which is the position of the helicopter with respect to the ground camera and velocities $\vec{w}$ and $\vec{V}$.

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

A. Two Camera Pose Estimation Method

The two camera pose estimation method uses a pan-tilt ground camera and an on-board camera. Previous work on vision-based pose estimation utilizes monocular views or stereo pairs. Our two camera pose estimation method involves the use of two cameras that are set to see each other. This method is especially useful for autonomous taking 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 3. 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 $i = 1 \cdots 6$.

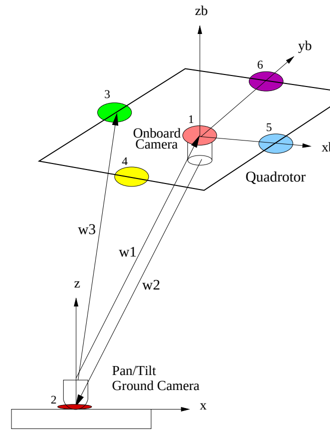

Fig. 3. Two-Camera Pose Estimation Method using a Pair of Ground and Onboard Cameras

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

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

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

$$\lambda_3 \vec{w}_3 = \lambda_1 \vec{w}_1 + \mathbf{R}\vec{L}_a \tag{2}$$

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

$$\lambda_1 (\vec{w}_3 \times \vec{w}_1) = \mathbf{R}\vec{L}_a \times \vec{w}_3. \tag{3}$$

This can be rewritten as

$$(\vec{w}_3 \times \vec{w}_1) \times (\mathbf{R}\vec{L}_a \times \vec{w}_3) = 0. \tag{4}$$

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

$$\mathbf{R} = Rot(\vec{w}_1 \times \vec{w}_2, \theta) \cdot Rot(\vec{w}_1, \alpha) \tag{5}$$

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 $\vec{w}_2$.

$$\theta = acos(\vec{w}_1 \cdot \vec{w}_2) \tag{6}$$

The only unknown is the angle α. Let $\mathbf{M}$ be a matrix described as

$$\mathbf{M} = (\vec{w}_3 \times \vec{w}_1) \times (\vec{w}_3 \times (\mathbf{R}(\vec{w}_1 \times \vec{w}_2, \theta))). \tag{7}$$

Using Rodrigues' formula, Equation 7 can be simplified to

$$\mathbf{M}\cdot\vec{L}_a+\sin\alpha\cdot\mathbf{M}\vec{w}_1\cdot\vec{L}_a+(1-\cos\alpha)\cdot\mathbf{M}\cdot(\vec{w}_1)^2\cdot\vec{L}_a=0. \quad (8)$$

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

$$\alpha=\arcsin\frac{B\cdot C\pm\sqrt{(B^2\cdot C^2-(A^2+B^2)\cdot(C^2-A^2))}}{A^2+B^2} \quad (9)$$

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 ($\phi$, 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 $i=1\cdots6$. λ_1 can be found using Equation 3. 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

$$\lambda_i\vec{w}_i=\lambda_1\vec{w}_1+\mathbf{R}\vec{L}_i. \quad (10)$$

L_i is the position vector of i^{th} blob 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

$$\begin{aligned}X&=(\lambda_3\vec{w}_3(1)+\lambda_4\vec{w}_4(1)+\lambda_5\vec{w}_5(1)+\lambda_6\vec{w}_6(1))/4\\Y&=(\lambda_3\vec{w}_3(2)+\lambda_4\vec{w}_4(2)+\lambda_5\vec{w}_5(2)+\lambda_6\vec{w}_6(2))/4\\Z&=(\lambda_3\vec{w}_3(3)+\lambda_4\vec{w}_4(3)+\lambda_5\vec{w}_5(3)+\lambda_6\vec{w}_6(3))/4.\end{aligned} \quad (11)$$

B. Comparing the Pose Estimation Methods

The proposed two camera pose estimation method is compared to other methods using a MATLAB simulation. Other methods used were a four-point algorithm [4], a state estimation algorithm [14], 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.

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

$$\begin{aligned}e_{ang}&=\parallel log(\mathbf{R}^{-1}\cdot\mathbf{R}^{est})\parallel\\e_{pos}&=\parallel\vec{p}-\vec{p}^{est}\parallel.\end{aligned} \quad (12)$$

$\mathbf{R}^{est}$ 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 $\mathbf{R}$ to $\mathbf{R}^{est}$.

Figures 4, 5 show the motion of the quadrotor and the pose estimation errors. 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* were added on image values. The blob areas were also added a random error of magnitude ±2. The comparison of the pose estimation methods and the average angular and positional errors are given on Table 1.

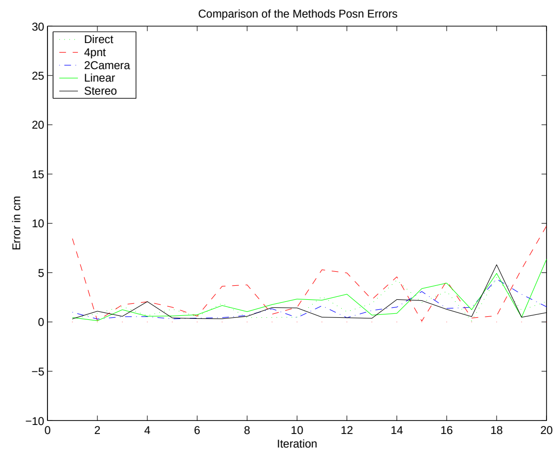

Fig. 4. Comparison of the Position Estimation Errors

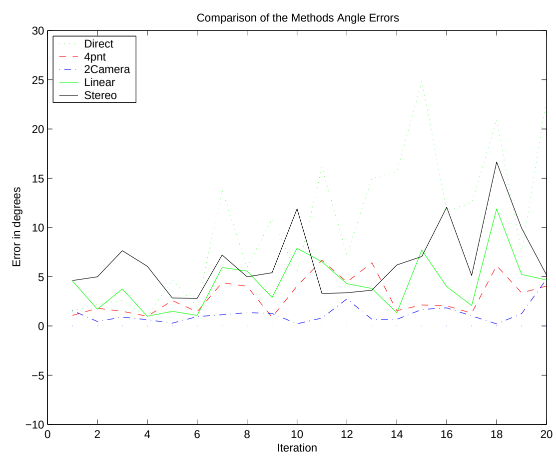

Fig. 5. Comparison of the Orientation Estimation Errors

It can be seen from the plots and 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. 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.

III. HELICOPTER MODEL AND CONTROL

The quadrotor helicopter model is shown in Figure 6. A body-fixed 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 (**O**) by a position vector p=(x, y, z) $\in\mathbf{O}$ and a rotation matrix $R:O\rightarrow B$, where $R\in\mathbf{SO(3)}$. A ZYX Euler angle representation has been chosen to represent the rotations. It is composed of three Euler angles,

Method	Ang. E. (*deg*)	Posn. E. (*cm*)
Direct M.	10.2166	1.5575
4 Pnt. M.	3.0429	3.0807
2 Camera M.	1.2232	1.2668
Linear M.	4.3700	1.8731
Stereo M.	6.5467	1.1681

TABLE I
COMPARISON OF THE ANGULAR AND POSITIONAL ERRORS OF DIFFERENT POSE ESTIMATION METHODS

(ϕ, θ, ψ), representing yaw, roll (rotation around y-axis) and pitch (rotation around x-axis), respectively.

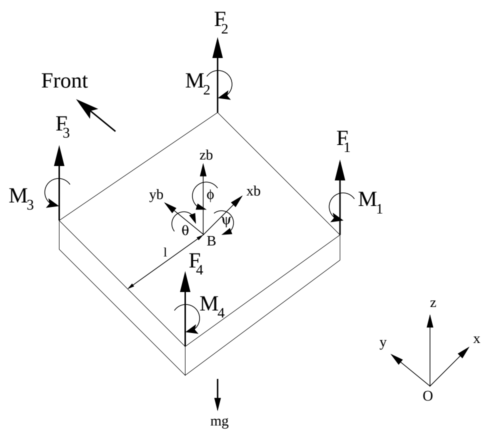

Fig. 6. 3D Quadrotor Model

A spinning rotor produces moment as well as thrust. Let F_i be the thrust and M_i be the moment generated by rotor i, that is spinning with rotational speed of w_i.

Let $V_b \in \mathbf{B}$ be the linear velocity in body-fixed frame and $w_b \in \mathbf{B}$ the angular velocity. Therefore the velocities will be

$$V_b = R^T \dot{p} \tag{13}$$
$$skew(w_b) = R^T \dot{R} \tag{14}$$

where $skew(w) \in \mathbf{so(3)}$ is the skew symmetric matrix of w. To represent the dynamics of the quadrotor, one can write the Newton-Euler equations as follows

$$m\dot{V}_b = F_{ext} - w_b \times mV_b \tag{15}$$
$$I_b \dot{w}_b = M_{ext} - w_b \times I_b w_b. \tag{16}$$

F_{ext} and M_{ext} are the external forces and moments on the body-fixed frame. I_b is the inertia matrix, and m is the mass of the helicopter.

Drag on a moving object [12] is given by $Drag = \frac{1}{2} C_d \rho v^2 A$, in which ρ is the density of air, A is the frontal area, C_d is the drag coefficient, and V is the velocity. Assuming constant ρ, the constants at the above equation can be combined to form C, which simplifies drag to $Drag = Cv^2$.

The force generated by a rotor [12] which is spinning with rotational velocity of w is given by $F = bL = \frac{\rho}{4} w^2 R^3 abc(\theta_t - \phi_t)$, where b is the number of blades on a rotor, θ_t is the pitch at the blade tip, ϕ_t is the inflow angle at the tip. By combining the constant terms as constant variable D, this equation simplifies to $F_i = D{w_i}^2$.

Therefore F_{ext} and M_{ext} will be

$$F_{ext} = -C_x \dot{x}^2 \hat{i} - C_y \dot{y}^2 \hat{j} + (T - C_z \dot{z}^2)\hat{k} - R \cdot mg\hat{k} \tag{17}$$
$$M_{ext} = M_x \hat{i} + M_y \hat{j} + M_z \hat{k} \tag{18}$$

where C_x, C_y, C_z are the drag coefficients along x, y and z axes, respectively. T is the total thrust and M_x, M_y, M_z are the moments generated by the rotors. The relation of thrust and moments to the rotational velocities of rotors is given as follows

$$\begin{pmatrix} T \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} D & D & D & D \\ -Dl & Dl & Dl & -Dl \\ -Dl & -Dl & Dl & Dl \\ CD & -CD & CD & -CD \end{pmatrix} \begin{pmatrix} {w_1}^2 \\ {w_2}^2 \\ {w_3}^2 \\ {w_4}^2 \end{pmatrix}. \tag{19}$$

The above matrix $M \in \mathbb{R}^{4\times 4}$ is full rank for $l, C, D \neq 0$. The rotational velocity of rotor i (w_i), can be related to the torque of motor i (τ_i) as

$$\tau_i = I_r \dot{w}_i + K{w_i}^2 \tag{20}$$

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

Motor torques τ_i should be selected to produce the desired rotor velocities (w_i) in Equation 20, which will change the external forces and moments in Equation 18. This will produce the desired body velocities and accelerations in Equation 14.

A. Control

A controller should pick suitable rotor speeds w_i for the desired body accelerations. Let us define the control inputs to be

$$\begin{aligned} u_1 &= (F_1 + F_2 + F_3 + F_4) \\ u_2 &= l(-F_1 + F_2 + F_3 - F_4) \\ u_3 &= l(-F_1 - F_2 + F_3 + F_4) \\ u_4 &= C(F_1 - F_2 + F_3 - F_4). \end{aligned} \tag{21}$$

C is the force-to-moment scaling factor. The u_1 represents a total thrust on the body in the z-axis, u_2 and u_3 are the pitch and roll inputs and u_4 is a yawing moment. Backstepping controllers [13] are useful when some states are controlled through other states. Since motions along the x and y axes are related to tilt angles θ and ψ respectively, backstepping controllers given in [1] can be used to control tilt angles enabling the precise control of the x and y motions (inputs u_2 and u_3). The altitude and the yaw, can be controlled by PD controllers

$$\begin{aligned} u_1 &= \frac{g + K_{p1}(z_d - z) + K_{d1}(\dot{z}_d - \dot{z})}{\cos\theta \cos\psi} \\ u_4 &= K_{p2}(\phi_d - \phi) + K_{d2}(\dot{\phi}_d - \dot{\phi}). \end{aligned} \tag{22}$$

IV. Experiments

The proposed controllers and the two-camera pose estimation algorithm have been implemented on a remote-controlled battery-powered helicopter shown in Figure 7. 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 8 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. Vision computer is a Pentium 4, 2 *Ghz* machine which had a Imagination PXC200 color frame grabbers. Images can be captured at 640 $\times$ 480 resolution at 30 *Hz*. The camera used for the experiments was a Sony EVI-D30 pan/tilt/zoom color 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 a couple of color spaces that are common 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 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. 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*.

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 separate computers which were 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. 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.

Controllers described in Section 3 were implemented for the experiments. Figure 9 shows the results of the altitude and yaw control experiment that is using a single ground camera only. In this experiment altitude and yaw angle are being controlled. The pose is estimated using a pose estimation method that uses image features as well as feature areas to estimate the helicopter pose.

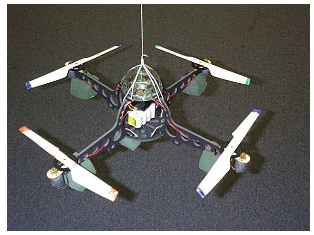

Fig. 7. Quadrotor Helicopter

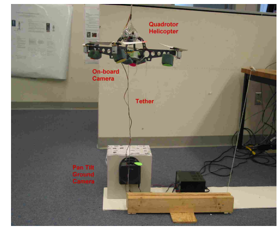

Fig. 8. Experimental Setup

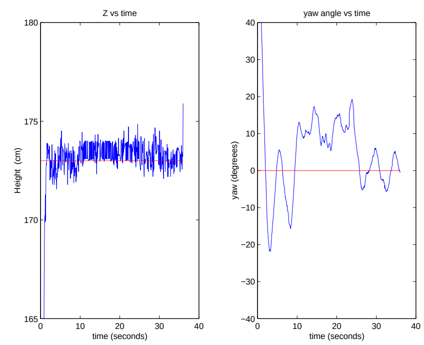

Fig. 9. The Results of the altitude and yaw Control Experiment with a Ground Camera based Direct Pose Estimation Method

Figure 10 shows the results of the experiment that is using the two camera pose estimation method, where height, *x*, *y* and yaw angle are being controlled. 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 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.

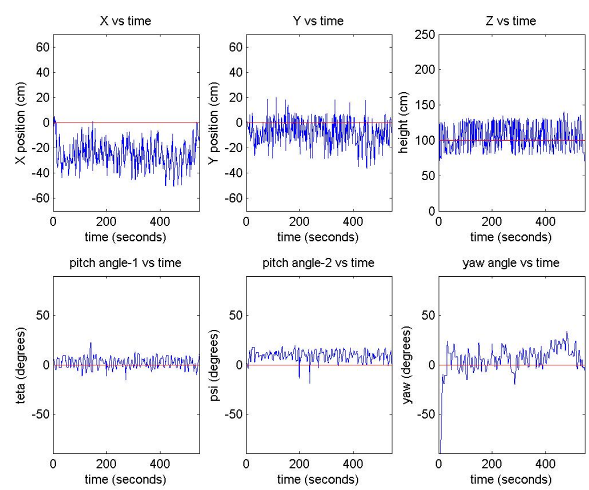

Fig. 10. The Results of the Height *x*, *y* and yaw Control Experiment with Two-Camera Pose Estimation Method

V. Conclusion

We have presented a novel two camera method for 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. Backstepping controllers have been used to stabilize and perform output tracking control. The proposed controllers and the pose estimation method have been implemented on a remote-control, battery-powered model helicopter. Experiments on a tethered system showed that the vision-based control is effective in controlling the helicopter. Our future work will include placing the ground camera on top of a mobile robot and enabling take-off/landing from a mobile robot. Such functionalities will be useful for inspection, chase and other ground-air cooperation tasks.

Acknowledgment

The authors gratefully acknowledge support from Air Force Office of the Scientific Research (AFOSR) and Ministry of National Education of Turkey. We also would like to thank Aveek Das and John Spletzer for their help in image processing, and feature extractor.

References

[1] E. Altuğ, J. P. Ostrowski, R. Mahony, *Control of a Quadrotor Helicopter using Visual Feedback*, Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, D.C., May 2002, pp 72-77.

[2] E. Altuğ, J. P. Ostrowski, C. Taylor, *Quadrotor Control using Dual Visual Feedback*, Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September 2003, pp 4294-4299.

[3] O. Amidi, *An Autonomous Vision Guided Helicopter*, Ph.D. Thesis, Carnegie Mellon University, 1996.

[4] A. Ansar, D. Rodrigues, J. Desai, K. Daniilidis, V. Kumar, M. Campos, *Visual and Haptic Collaborative Tele-presence*, Computer and Graphics, Special Issue on Mixed Realities Beyond Convention, Oct 2001.

[5] T. Hamel, R. Mahony, *Visual Servoing of a Class of Under-actuated Dynamic Rigid-body Systems*, Proceedings of the 39th IEEE Conference on Decision and Control, 2000.

[6] T. Hamel, R. Mahony, A. Chriette, *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., May 2002, pp 2781-2786.

[7] Hauser, Sastry, Meyer, *Nonlinear Control Design for Slightly Non-Minimum Phase Systems: Application to V/STOL Aircraft*, Automatica, vol 28, No:4, pp 665-679, 1992.

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

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

[10] Martin, Devasia, Paden *A Different Look at Output Tracking Control of a VTOL Aircraft*, Automatica 32, pp 101-107.

[11] R. Murphy, M. Ausmus, M.Bugajska, T. Ellis, T. Johnson, N.Kelley, J. Kiefer, L.Pollock, *Marsupial-like Mobile Robot Societies*, Proceedings of the Third International Conference on Autonomous Agents, 1999.

[12] R. Prouty, *Helicopter Performance, Stability, and Control*, Krieger Publishing Company, June 1995.

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

[14] C. S. Sharp, O. Shakernia, S. S. Sastry, *A Vision System for Landing an Unmanned Aerial Vehicle*, IEEE Conference on Robotics and Automation, 2001.

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

[16] Y. Ma, J. Kosecka, S. Sastry, *Optimal Motion from image Sequences: A Riemannian viewpoint*, Research Report, University of California Berkeley, UCB/ER/M98/37.

[17] H. Zhang, *Motion Control for Dynamic mobile robots*, Ph.D. Thesis, University of Pennsylvania, June 2000.