provided by IntechOper

the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

122,000

ternational authors and editors

135M

Downloads

154

TOP 1%

Our authors are among the

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE^{TA}

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Fly-The-Camera Perspective: Control of a Remotely Operated Quadrotor UAV and Camera Unit

DongBin Lee¹, Timothy C. Burg¹, Darren M. Dawson¹ and Günther Dorn²

¹Clemson University, ²Hochschule Landshut

¹USA, ²Germany

1. Introduction

This Chapter presents a mission-centric approach to controlling the optical axis of a video camera mounted on a *camera positioner* and fixed to a *quadrotor* remotely operated vehicle. The approach considers that for video collection tasks a single operator should be able to operate the system by "flying-the-camera"; that is, collect video data from the perspective that the operator is looking out of and is the pilot of the camera. This will allow the control of the quadrotor and the camera manipulator to be fused into single *robot manipulator* control problem where the camera is positioned using the four degree-of-freedom (DOF) quadrotor and the two DOF camera positioner to provide a full six DOF actuation of the camera view. Design of a closed-loop controller to implement this approach is demonstrated using a *Lyapunov*-type analysis. Computer simulation results are provided to demonstrate the suggested controller.

Historically, the primary driver for UAV reconnaissance capabilities has been military applications; however, we appear to be at the juncture where the cost and capabilities of such systems has become attractive in civilian applications. The success of recent UAV systems has raised expectations for an increased rate of technology development in critical factors such as low-cost, reliable sensors, air-frame construction, more robust and lightweight material, higher energy-density battery technologies, and interfaces that require less operator training. One of the essential technologies is the camera positioner, which includes camera, camera base, and multi-axis servo platform. The potential for UAVs with camera positioners has been well established in many applications as diverse as fire fighting, emergency response, military and civilian surveillance, crop monitoring, and geographical registration. Many research and commercial groups have provided convincing demonstrations of the utility of UAVs in these applications. Most of the commercial systems are equipped with camera positioners as standard equipment; however, the use of the camera is not integrated with the control of the UAV.

The typical structures of a camera positioner include pan-tilt, tilt-roll, or pan/tilt/roll revolute joints or multi-axis gimbals. When considering the actuator of the camera gimbal, rate-gyros or encoders are used to measure the orientations. If the system is small and lightweight, the actuator dynamics can be discounted or neglected in the control of the UAV. Heavier systems, relative to the UAV, may require that the interaction of the camera

positioner with the airframe be considered in the control system. In this chapter we demonstrate an approach that couples the positioning of a camera system to the control of the UAV. We believe this approach will serve as the basis for coupling dynamic control of the two systems.



Figure 1. Clemson UAV with Pan-Tilt Camera

1.1 Motivation

A typical camera system has a 2-axis gimbal or pan-tilt attached to the UAV as shown in Fig. 1. This camera positioner, with actuators, is usually fixed to the UAV rigid-body and generally attached to the front of the aerial vehicle where it can be used for surveillance, monitoring, or object targeting. The system is usually open-loop controlled using a controller/ joystick at the ground station. In this case, when the UAV tilts up and down to meet position control objectives of the airframe, the camera will also point up and down as shown in the upper plots of Fig. 2. If left uncompensated, the camera loses the target and fails to meet the surveillance objective. Compensating the camera field-of-view as shown in the lower half of Fig. 2 means that the camera positioner is moved in reaction to the platform motion. In the simplest system, the UAV pilot manually performs this compensation.

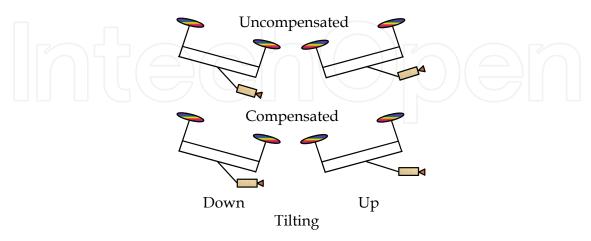


Figure 2. Uncompensated (upper) Comensated (lower) Camera Platform Position for UAV Airframe Tilting Motion

When the navigation or surveillance tasks become complicated, two people may be required to achieve the camera targeting objective that is controlled *independently* from the vehicle: a pilot to navigate the UAV and a camera operator. In automated systems feature tracking software can be used to identify points of interest in the field of view and then generate camera position commands that keep these features in the field of view. It is insightful to consider the actions of the two actors in pilot/camera operator scenario in order to hypothesize a new operational mode. The pilot will work to position the aircraft to avoid obstacles and to put the camera platform, *i.e.*, the aerial vehicle, in a position that then will allow the camera operator to watch the camera feed and move the camera positioner to track a target or survey an area.

An important underlying action on the part of the camera operator that makes this scenario feasible is that the camera operator must compensate for the motions of the UAV that stabilize the camera targeting. Fig. 3 further describes the effect of uncompensated camera platform motion on the camera axis in the upper figures and the bottom figures show a scheme where the camera positioner is used to compensate for the UAV body motion and maintain the camera view. Additionally, there must be communication between the pilot and the camera operator so that the camera platform is correctly positioned or moved to meet the video acquisition objective. More specifically, the camera positioning problem is split between the pilot and the camera operator. Since the operator is not in full control of positioning the camera, she must rely on commands to the pilot to provide movement of the camera platform for motions not included in the camera positioner. For example, if the camera positioner is a simple pan-tilt and the camera operator requires translation of the camera, then a request must be made to the pilot to move the camera platform. The potential shortcomings of this typical operational scenario can be summarized as:

- 1. multiple skilled technicians are typically required,
- 2. the camera operator must compensate for the actions of the pilot, and
- 3. it is not intuitive for a camera operator to split the camera targeting tasks between actions of the camera positioner controlled by the operator and commands to the pilot.

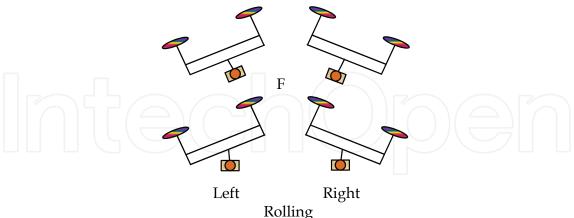


Figure 3. Uncompensated (upper) and Compensated (lower) Camera Platform for Rolling Motion

1.2 Previous Research

Sample research that frames the UAV-camera control problem is now reviewed. The user's complaint of difficulty of locating objects indicates the need for an actuated roll axis as

described in Adelstein & Ellis (2000). They created a 3 DOF camera platform, to promote telepresence, by complementing yaw and pitch motions with roll motion in an experiment. The research in Mahony et al. (2002) examined hardware and software to automatically keep the target in the camera's field-of-view and stabilize the image. This work suggested "eye-in-hand" visual servo approach for a robotic manipulator with a camera on the end effector and introduced nonlinear gain into the orientation feedback kinematics to ensure that the target image does not leave the visual field. Jakobsen & Johnson (2005) presented control of a 3-axis Pan-Tilt-Roll camera system using modified servos and optical encoders mounted on a UAV with three different operating modes. Lee et al (2007) presented the modelling of a complete 3 link robotic positioner and suggested a single robotic body system by combining it with a quadrotor model.

The authors, Yoon & Lundberg (2001), presented equations of motion for a two-axis, pantilt, gimbal system to simplify the gimbal control and to further illustrate the properties of the configuration. In Sharp et al. (2001), the authors implemented a realtime vision system for a rotorcraft UAV during landing by estimating the vehicle state. A pan-tilt camera vision system is designed to facilitate the landing procedure based on geometric calculations from the camera image. The work in Stolle & Rysdyk (2003) proposed a solution to the problem of the limited range of the camera mechanical positioner. The system attempts to keep a target in the camera field-of-view by applying a circle-based flight path guidance algorithm with a nose-mounted pan-tilt camera. Pieniazek (2003) presented a software-based camera control and stabilization for two degree-of-freedom onboard camera so as to stabilize the image when the aircraft attitude was disturbed by turbulence or attitude changes. Quigley et al. (2005) presented a field-tested mini-UAV gimbal mechanism, flightpath generation algorithm, and a human-UAV interface to enhance target acquisition, localization, and surveillance.

Procerus Technologies produces OnPoint™ targeting system for vision-based target prosecution. The system utilizes feature tracking and video stabilization software, a fixed camera, and no GPS sensor to achieve these goals. Micropilot company manufactures a stabilized pan-tilt-zoom camera system which stabilizes the video camera in yawing or rolling and tilting (elevation) providing a stable image at high zoom. This stabilized gimbal system is mounted on a mechanical interface that can be specified to match the UAV objective with the ground control software HORIZON™P.

1.3 Camera Stabilization and Targeting

A 2-axis camera positioner is shown in Fig. 4 which can be used for surveillance or trajectory-tracking. The problem of providing an intuitive interface with which an operator can move a camera positioner to make a video camera follow a target image appears in many places. The difficulty of moving a system that follows a subject with a video camera was recently addressed in Cooke et al. (2003) where operating a multilink, redundant-joint camera boom for the movie and television industry is described. The interesting result from this work is that an integrated control strategy, using a vision servoing approach to reduce the number of links controlled by the operator, can improve the use of the system. The final result shows an unexperienced operator achieving the same tracking result as an experienced operator; hence, the control strategy has rendered the system more friendly to the operator. The salient point of the control strategy is that there is independent macro- and micro- positioning of the camera - the operator controls the coarse positioning and the vision

system controls the fine positioning. The authors suggest that the same approach could be used for other camera platforms; however, it is required that the system have redundant positioning axes. Additionally, an automated vision servoing system may not be desirable for general reconnaissance where the target is not known.

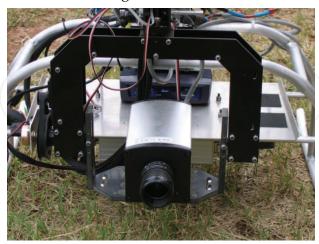


Figure 4 Two-axis Camera Positioner

A different perspective to this basic camera targeting problem was presented in Chitrakaran (2006) and Neff et al. (2007) where the camera platform, a quadrotor UAV, and the camera positioning unit are considered to be controlled concurrently. In this work a controller was developed which simultaneously controls both the quadrotor and the camera positioning unit in a complimentary fashion. Both works show combining the four degrees-of-freedom provided by motion of the quadrotor helicopter with two degrees-of-freedom provided by a camera positioner to provide arbitrary six degree-of-freedom positioning of the on-board video camera. The work in Chitrakaran (2006) is actually directed towards providing an automated means of landing the quadrotor through the vision system but provides an important mathematical framework for analyzing the combined quadrotor/camera system. The work in Neff et al. (2007) builds on Chitrakaran (2006) using a velocity controller and first introduced the "fly-by-camera" concept by combining UAV and 2-axis tilt-roll camera for the combined quadotor/camera system that works from operator commands generated in the camera field-of-view to move both elements, presenting a new interface to the pilot. The research in Lee et al (2007) is exploited to provide an integrated system combining quadrotor UAV with 3-DOF camera system to present a single robotic unit. The paper designed a complete Pan-Tilt-Roll model which can be used for two camera optical axis; looking forward which can be used for surveillance, tracking, or targeting and looking downward axis for vision-based landing and monitoring the ground situation. In addition, the work suggests a position-based controller to show the upper bounds of position and velocity tracking error which yields Globally Uniformly Ultimately Bounded (SGUUB). The stabilization of the camera comes from this proposed perspective, which will be referred to as the fly-the-camera perspective, where the pilot commands motion from the perspective of the on-board camera - it is as though the pilot is riding on the tip of the camera and commanding movement of the camera ala a six-DOF flying camera. This is subtly different from the traditional remote control approach wherein the pilot processes the camera view and then commands an aircraft motion to create a desired motion of the camera view. The work proposed here exploits this new perspective for fusing vehicle and camera control. In

this exposition, a quadrotor UAV model will be combined with a two-DOF camera kinematic model to create a fully actuated camera frame and a positioner controller will be designed.

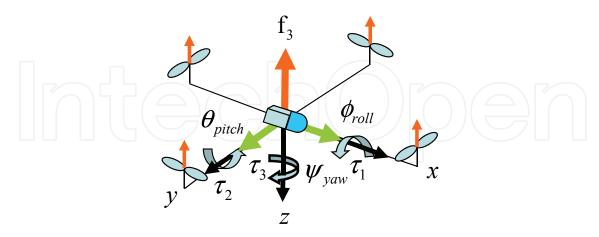


Figure 5. Underactuated Quadrotor Helicopter

2. System Modeling

2.1 Quadrotor UAV

The elements of the quad-rotor unmanned aerial vehicle model are shown in Fig. 5. This system can only directly produce a thrust, f_3 , along the z-axis (vertical) and torques about the roll, pitch, and yaw angle axes (τ_1, τ_2 , and τ_3); hence, the system is underactuated and the translational motion in the x- and y-directions is indirectly achieved. The quadrotor is assumed to be a rigid body on which thrust and torque act uniformly through the body and that the quadrotor body fixed frame, F, is chosen to coincide with the center of gravity, which implies that it has a diagonal inertia matrix. The kinematic model of a quadrotor expressed in the inertial reference frame, denoted as North-East-Downward (NED), is given by

$$\begin{bmatrix} \dot{p}_F \\ \dot{\Theta}_F \end{bmatrix} = \begin{bmatrix} R_F^I & O_3 \\ O_3 & T_F^I \end{bmatrix} \begin{bmatrix} v^F \\ \omega^F \end{bmatrix} \in R^6$$
 (1)

where $v^F(t)$ and $\boldsymbol{\omega}^F = [\boldsymbol{\omega}_x, \, \boldsymbol{\omega}_y, \, \boldsymbol{\omega}_y]^T \in R^3$ denote the linear velocity and angular velocity of the quadrotor body-fixed frame, F, with respect to the earth-fixed inertial frame, I, expressed in the body-fixed frame, F. The position and angle, $p_F(t)$ and $\Theta_F(t)$, and velocities $v^F(t)$ and $\boldsymbol{\omega}^F(t)$ are assumed to be measurable. The $\dot{p}_F(t)$ in (1) is the velocity $v^F(t)$ of the quadrotor transformed by the spatial rotation matrix $R_F^I(\Theta) \in SO(3)$ where $\Theta_F = [\phi, \, \theta, \, \psi]^T \in R^3$ are roll, pitch, and yaw angles about x-, y-, and z-axes (Fossen, 2002)

$$\dot{R}_F^I = R_F^I S(\omega^F). \tag{2}$$

 $\dot{\Theta}_F(t)$ in (1) represents the angular velocity $\omega^F(t)$ transformed into the inertial frame by the angular orientation matrix $T_F^I(\Theta) \in R^{3 \times 3}$ (Fossen, 2002) which represents the modeling assumption that angular velocity of the quadrotor is calculated directly in lieu of modeling the angular dynamics; that is, $\omega^F(t)$ is considered as the system input. The translational dynamic modeling equation of the quadrotor is based on Chitrakaran (2006) and given by

$$m\dot{v}^F = F^f - mS(\omega^F)v^F + N_1(v^F) + G(R_I^F)$$
 (3)

where the rotational angular velocity, $\omega^F(t)$, is implemented by control command (see (57)) and (3) contains the gravitational term, $G(R_I^F)$, which is represented in the body-fixed frame as

$$G(R_I^F) = mg(R_I^F)E_3 \in R^3 \tag{4}$$

where $g \in R^1$ denotes gravitational acceleration, $E_3 = [0,0,1]^T$ denotes the unit vector in the coordinates of the inertial frame, $m \in R^1$ is the known mass of the quad-rotor, $N_1(v^F) \in R^3$ represents a bounded, unknown, nonlinear function (i.e, aerodynamic damping) and $S(\cdot) \in R^{3\times 3}$ is a general form of the skew-symmetric matrix as follows

$$S(\xi) = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix}, \ \xi = [\xi_1, \xi_2, \xi_3]^T \in \mathbb{R}^3.$$
 (5)

As mentioned at the start of this modeling section, the quadrotor shown in Fig. 5 is underactuated in the sense that it has one translational force along the z-axis. The vector $F^f(t) \in \mathbb{R}^3$ refers to the quadrotor translational forces but in reality represents the single translational force which is created by summing the forces generated by the four rotors and is expressed as

$$F^f = B_1 u_1 = \begin{bmatrix} 0 & 0 & u_1 \end{bmatrix}^T \tag{6}$$

where $B_1 = I_3$ is a configuration matrix (actuator dynamics are beyond the scope of this design) and $u_1(t) = f_3(t) \in \mathbb{R}^1$. The modeling nomenclature is summarized in 7.3.

2.2 Model Assumptions

The following assumptions are made regarding the system model:

• A1: $T_F^I(\Theta)$ are full rank, *i.e.*, $\theta(t) \neq \pm \frac{\pi}{2}$. This will ensure that the orientation angle, θ_a , (defined in (38)) remains within the range $-\pi < \theta_a < \pi$ about the rotation axis $\mu(t)$ and will ensure that $\det(L_{\omega})$ exists (see (43)).

• A2: Aerodynamic damping term $N_1(v^F)$ in (3) can be replaced by the parameter linearizable form; i.e., $Y_1(v^F)\theta_1 = N_1(v^F)$ where $\|Y_1(v^F)\| \le \zeta_0 \|v^F\|$, ζ_0 is a positive constant, $Y_1(v^F) \in R^{3 \times n}$ is a known regression matrix, and $\theta_1 \in R^n$ is a known parameter vector. Additionally, $\|Y_1(v^F)\theta_1\| \le \zeta_1 \|v^F\|$ where ζ_1 is a positive function and non-decreasing in $\|v^F\|$ and $\zeta_1 \in R^1$ is a positive constant.

2.3 Camera Positioner

Here we suggest a camera system with two revolute joints to cover forward and downward optical axes as shown in Fig. 6. Considering the integrated system, UAV and camera positioner, the yaw action of the UAV provides the third orientation angle for the camera. To create a general analytic framework for modeling the configurations of a two-link revolute positioner, a 2-axis robot positioner model is proposed. This general two-axis camera positioner will provide the forward kinematics for a tilting-rolling motion and panning-tilting motion where the angles $\theta_{tilt}(t)$ and $\theta_{roll}(t)/\theta_{pan}(t)$ are assumed to be measurable. The Tilt-Roll configuration can be used to compensate for the quadrotor body roll and pitch when the camera is facing forward for tasks such as general navigation or surveillance. On the other hand, the Pan-Tilt camera positioner configuration can be used to compensate for the quadrotor pan and pitch motion when the camera is looking downward for tasks like landing, monitoring, or surveillance.

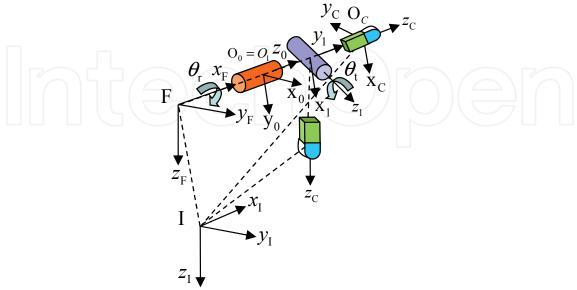


Figure 6. Quadrotor with a Camera Positioner Configurable for Tilt-Roll and Pan-Tilt Operating Modes

The dynamics of the camera unit will be considered negligible. The standard coordinate system definitions are made, specifically, O_0 is used to represent the origin of the coordinate system at the base (B) of the camera, O_1 is the origin of the second link, and O_c is used to represent the camera frame and can be denoted as O_2 . Note that the link lengths are assumed to be zero so O_0 , O_1 , and O_2 are coincident. The camera positioner kinematics will be calculated using Denavit-Hartenberg (D-H) Convention (Spong, 2006). From the convention the rotation matrix R_i^{i-1} from the i^{th} coordinate system to the $(i-1)^{th}$ frame is given

$$R_i^{i-1} = \begin{bmatrix} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$
(7)

where θ is the rotated angle, pan/tilt/roll, and α is the twisted angle between the links.

2.3.1 Case 1: Tilt-Roll Camera Configuration (camera looking forward)

A feasible 2-axis Tilt-Roll camera positioner is pictured in Fig. 7. Note that the matrix ${\cal O}_0$ represents a static mounting on the quadrotor as a camera base frame denoted B and is given by

$$R_{0=B}^{F} = R_{\alpha, \frac{\pi}{2}} R_{\theta, \frac{\pi}{2}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
(8)

where θ and α angles are each $+90^\circ$ in order calculated by (7) and $z_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ where z_{i-1} (i is the link origin) is the last column of the rotation matrix which will be used in the Jacobian matrix. The Denavit-Hartenberg (D-H) table for the Tilt-Roll configuration when the optical axis is looking forward, as shown in Fig. 7, is in Table I. The camera positioner unit is considered to have coincident rotational links, thus the link lengths are zero (a=0 in Table I).

Link (i)	Offset	Length	Twist	Angle
	D	a	α	θ
1	0	0	+90°	$\theta_r + 90^\circ$
2 (forward axis)	0	0	-90°	$\theta_{\scriptscriptstyle t}$

Table 1. D-H for Tilt-Roll Camera Positioner while looking forward optical axis



Figure 7. Quadrotor with Tilt-Roll Camera Positioner

The rotation matrices for first and second links are obtained as

$$R_1^B = \begin{bmatrix} -s\theta_r & 0 & c\theta_r \\ c\theta_r & 0 & s\theta_r \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad R_2^1 = \begin{bmatrix} c\theta_t & 0 & -s\theta_t \\ s\theta_t & 0 & c\theta_t \\ 0 & -1 & 0 \end{bmatrix}$$
(9)

where $c = \cos(\cdot)$, $s = \sin(\cdot)$ were used, and the rotation matrix from the base frame to first link frame is given

$$R_{l}^{F} = R_{B}^{F} R_{l}^{B} = \begin{bmatrix} 0 & 1 & 0 \\ -s \theta_{r} & 0 & c \theta_{r} \\ c \theta_{r} & 0 & s \theta_{r} \end{bmatrix}$$

$$(10)$$

where $z_1 = \begin{bmatrix} 0 & c\theta_r & s\theta_r \end{bmatrix}^T$. The total rotation matrix from the quadrotor through the second link can be obtained as

$$R_{2=C}^{F} = R_{B}^{F} R_{1}^{B} R_{2}^{1} = \begin{bmatrix} \sin \theta_{t} & 0 & \cos \theta_{t} \\ -\sin \theta_{r} \cos \theta_{t} & -\cos \theta_{r} & \sin \theta_{r} \sin \theta_{t} \\ \cos \theta_{r} \cos \theta_{t} & -\sin \theta_{r} & -\cos \theta_{r} \sin \theta_{t} \end{bmatrix}.$$
(11)

The angular velocity of the camera frame can be written as follows

$$\omega_{BC}^F = R_B^F \omega_{BC}^B = J_C \dot{\theta}_C \tag{12}$$

where the positioner joint angles, $\dot{\theta}_{C}(t) \in \mathbb{R}^{3}$, are given by

$$\dot{\theta}_C = \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_t \end{bmatrix}$$
 and $\theta_C = \begin{bmatrix} \theta_r \\ \theta_t \end{bmatrix}$. (13)

The Jacobian matrix $J_C(\theta_C) \in \mathbb{R}^{3\times 2}$ can be built from the rotation matrices with the final result given by

$$J_C = \begin{bmatrix} z_0 & z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & c\theta_r \\ 0 & s\theta_r \end{bmatrix}$$
(14)

where the reader is referred to Murray et al. (1994) for the Jacobian matrix of a manipulator.

2.3.2 Case 2: Pan-Tilt Camera Configuration (camera looking downward)

A similar approach to that for case 1, 2.3.1, can be followed to develop the kinematics for the Pan-Tilt configuration when the optical axis is looking downward as shown in Fig. 8.



Figure 8. Quadrotor with Pan-Tilt Camera Positioner

The systematic Denavit-Hartenberg convention is used to obtain the rotation matrices of the Pan-Tilt camera positioner configuration and the detail procedure is shown in Appendix 7.1.

2.4 Position and Orientation of the Camera Positioner

The objective is to control the motion of the camera optical axis. Towards this end, the kinematic relationships in (1) will be extended to include the action of the camera positioning unit and to obtain the position and orientation of the camera. The derivative of the position of the camera in the camera frame, C, with respect to the inertial frame, I, and expressed in the inertial frame, $\dot{p}_{C}(t) \in R^{3}$, is defined as

$$\dot{p}_C = R_C^I v^C \tag{15}$$

where $v^{C}(t) \in \mathbb{R}^{3}$ is the linear velocity in the camera frame, C, referred to inertial frame, I, expressed in the camera frame, C. The rotation matrix R_{C}^{I} from the camera frame to the inertial frame can be obtained using the previous equations as

$$R_C^I = R_F^I R_C^F. (16)$$

The velocity $v^{C}(t)$ can be divided into two components as follows

$$v^C = v^{IF} + v^{FC} = R_F^C v^F \tag{17}$$

where $v^{IF}(t)$ is the UAV velocity expressed in camera frame (C) and $v^{FC}(t) = 0$ since the camera positioning unit only has rotational axes and does not translate from the quadrotor body. Thus, substituting (17) for $v^{C}(t)$ in (15) yields

$$\dot{p}_{C} = R_{C}^{I} R_{F}^{C} v^{F} = R_{F}^{I} v^{F}. \tag{18}$$

The desired camera position trajectory, $\dot{p}_D(t) \in \mathbb{R}^3$, is generated via

$$\dot{p}_D = R_D^I v^D \tag{19}$$

where $v^D(t) \in R^3$ is a desired input velocity vector and the desired rotation matrix expressed in inertial frame, $R_D^I(\Theta, \theta_C, \theta_d) \in R^{3 \times 3}$, can be defined in the following manner

$$R_D^I = R_F^I R_C^F R_D^C (20)$$

where $R_D^C(\theta_d) \in R^{3 \times 3}$ represents a rotation matrix from D to desired frame C as shown in Fig. 9 for the subsequent control development which is the tracking of the camera frame towards the desired camera frame using angle-axis representation (see Appendix 7.2). Utilizing (20), (19) yields

$$\dot{p}_D = R_F^I R_C^F R_D^C v^D. \tag{21}$$

Hence, to quantify the mismatch between the camera (actual) and desired attitudes, we define the rotation matrix $\tilde{R}(\Theta, \theta_C, \theta_d) \in R^{3\times 3}$ which can be obtained using measurable rotations and using the angle-axis representation (Spong, 2006) as

$$\tilde{R} = R_C^I (R_D^I)^T. \tag{22}$$

A similar result for the camera angles, $\Theta_C(t)$, in the camera frame from the second equation of (1) is now shown as

$$\dot{\Theta}_{C} = T_{F}^{I}(\Theta) \omega_{IC}^{F} = T_{F}^{I}(\Theta) R_{C}^{F} \omega^{C}$$
(23)

where an angular transformation $T_F^I(\Theta)$ is used. Decomposing the angular velocity,

$$\omega^{C}(t) = \omega^{IF} + \omega^{FB} + \omega^{BC} = R_F^C \omega^F + R_F^C \omega_{FB}^F + R_B^C \omega_{BC}^B, \tag{24}$$

yields

$$\dot{\Theta}_C = T_F^I(\Theta) R_C^F (R_F^C \omega^F + R_F^C \omega_{FB}^F + R_B^C \omega_{BC}^B). \tag{25}$$

Then, (25) yields

$$\dot{\Theta}_{C} = T_{F}^{I}(\Theta)\omega^{F} + T_{F}^{I}(\Theta)J_{C}\dot{\theta}_{c} \tag{26}$$

where $R_C^F R_F^C = 1$ is used and $\omega_{FB}^F = 0$ since the camera base is rigidly mounted on the quadrotor frame. Finally, following the same approach the desired camera angle, $\Theta_D(t)$, is obtained from the desired angular velocity of the camera in the quadrotor frame, $\omega^D(t)$, as

$$\dot{\Theta}_D = T_F^I(\Theta) R_D^F \omega^D. \tag{27}$$

The changing rate of $\mathit{R}_{\mathit{C}}^{\mathit{I}}(\Theta)$ is obtained as follows

$$\dot{R}_C^I = R_C^I S(\omega^C) \quad \text{and} \quad \dot{R}_I^C = S^T(\omega^C) R_I^C = -S(\omega^C) R_I^C.$$
 (28)

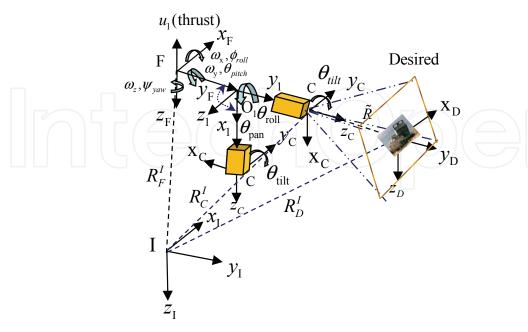


Figure 9. Tracking of the Camera Frame Towards the Desired Camera Frame

3. Control Method

Fig. 10 demonstrates the culmination of the modeling effort to combine the quadrotor and camera positioner to create a means of fully actuating the camera optical axis. In this diagram, it can be seen that the camera is positioned and oriented by using the two camera positioner angles (selected from $\theta_{tilt}(t)$, $\theta_{roll}(t)$, and $\theta_{pan}(t)$, $\theta_{tilt}(t)$, according to the configuration of the camera optical axis), the quadrotor linear force, F(t), and the quadrotor angles $\varphi_{roll}(t)$, $\theta_{pitch}(t)$, and $\psi_{yaw}(t)$ through $\omega_x(t)$, $\omega_y(t)$, and $\omega_z(t)$. In keeping with the fly-the-camera objective, a controller will be designed based on these inputs to move the camera optical axis along a desired trajectory. A control strategy will be proposed to control the camera translational position error, $e_p(t) \in R^3$, and the camera orientation error, $e_p(t) \in R^3$.

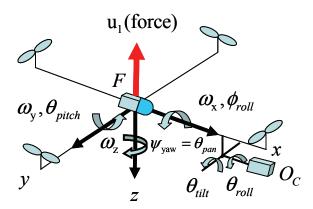


Figure 10. Six-DOF Single Robotic System

3.1 Tracking Error Formulation

The camera translational position error, $e_p(t) \in R^3$, is defined in the camera frame (C) as the transformed difference between the inertial frame based camera position, $p_C(t)$, and the inertial frame based desired camera position, denoted as $p_D(t) \in R^3$, as follows

$$e_p = R_I^C (p_C - p_D). \tag{29}$$

The camera translational position error rate, $\dot{e}_p(t) \in \mathbb{R}^3$, is obtained by taking the time derivative of (1) to yield

$$\dot{e}_{p} = \dot{R}_{I}^{C}(p_{C} - p_{D}) + R_{I}^{C}(\dot{p}_{C} - \dot{p}_{D})$$
(30)

where $\dot{p}_{C}(t)$ and $\dot{p}_{D}(t)$ were introduced in (18) and (19), respectively. Substituting (28) into the first term in (30) yields

$$\dot{R}_{I}^{C}(p_{C}-p_{D}) = -S(\omega^{C})e_{p}, \tag{31}$$

and the term $R_I^C \dot{p}_C(t)$ in (30) can be rewritten in terms of the quadrotor velocity, $v^F(t)$, as

$$R_{I}^{C} \dot{p}_{C} = R_{I}^{C} (R_{C}^{I} v^{C}) = R_{F}^{C} v^{F}$$
(32)

where all translation of the camera is the result of the quadrotor translation. The final term in (30) is rewritten as

$$R_{I}^{C} \dot{p}_{D} = R_{I}^{C} (R_{D}^{I} v^{D}) = R_{D}^{C} v^{D}. \tag{33}$$

Substituting (31), (32), and (33) into (30) yields

$$\dot{e}_{p} = -S(\omega^{C})e_{p} + R_{F}^{C}v^{F} - R_{D}^{C}v^{D}. \tag{34}$$

To further the controller development, a filtered error, $r(t) \in \mathbb{R}^3$, is introduced as

$$r = \begin{bmatrix} r_p^T & e_\theta^T \end{bmatrix}^T \tag{35}$$

where the filtered position error, $r_p(t) \in \mathbb{R}^3$, is defined as

$$r_p = k_p e_p + R_F^C v^F + R_F^C \delta, \tag{36}$$

in which $\delta = [0 \quad 0 \quad \delta_3]^T \in R^3$ is a constant design vector. The orientation tracking signal, $e_{\theta}(t) \in R^3$, is defined in terms of the angle-axis representation of the rotation matrix between the desired and actual camera orientations, $\tilde{R}(\Theta, \theta_C, \theta_d)$, as

$$e_{_{\theta}} = \theta_{a} \mu \tag{37}$$

where the scalar angle $\, heta_a\,$ is obtained from

$$\theta_a = \cos^{-1}(\frac{1}{2}(Tr(\tilde{R}) - 1)) \in R^1,$$
(38)

in which $Tr(\tilde{R})$ defines the trace of the matrix $\tilde{R}(\Theta,\theta_C,\theta_d)$ and the unit length axis of rotation $\mu \in R^3$ defined as

$$\mu = \frac{1}{2\sin\theta_a} \left[(r_{32} - r_{23}), (r_{13} - r_{31}), (r_{21} - r_{12}) \right]^{\mathsf{T}}, \tag{39}$$

for which $\|\mu\|^2 = 1$. In this representation, the rotation angle θ_a is assumed to stay within the range $-\pi < \theta_a < \pi$. Note that the terms on the right-hand side of the definition in (39) come from $\tilde{R}(\Theta, \theta_C, \theta_d)$ as

$$\tilde{R}(\Theta, \theta_C, \theta_d) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{40}$$

Substituting the axis-angle representation (38) and (39) into (37) yields

$$e_{\theta} = \frac{1}{2sinc\{\cos^{-1}(\frac{1}{2}(Tr(\tilde{R}) - 1))\}} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$
(41)

where $sinc(\theta_a) = \frac{\sin(\theta_a)}{\theta_a}$. The angular error rate can be obtained by taking the time derivative of (37) as

$$\dot{e}_{\theta} = L_{\omega} \omega^{CD} \in R^3, \tag{42}$$

and $L_{\omega} \in \mathbb{R}^{3\times3}$ can be defined as

$$L_{\omega} = I_3 - \frac{\theta_a}{2} S(\mu) + \left(1 - \frac{\operatorname{sinc}(\theta_a)}{\operatorname{sinc}^2(\frac{\theta_a}{2})}\right) S^2(\mu), \tag{43}$$

in which $S(\mu)$ is a skew symmetric matrix and details for obtaining

$$S(\mu) = \frac{(\tilde{R} - (\tilde{R})^T)}{2\sin\theta_a},\tag{44}$$

for obtaining (42) can be found in Malis (1998). The term $\omega^{CD}(t) \in \mathbb{R}^3$ introduced in (42) represents the angular velocity of the desired camera frame relative to the actual camera frame as

$$\omega^{CD} = \omega^{CI} + \omega^{ID} = \omega^{D} - \omega^{C} = \omega^{D} - (\omega_{IE}^{C} + \omega_{EC}^{C})$$
(45)

where $\omega^{ID}(t) = \omega^D(t)$ and $\omega^{CI}(t) = -\omega^{IC}(t) = -\omega^C(t)$. Substituting (45) along with (12) - (14) into (42) produces

$$\dot{e}_{\theta} = L_{\omega} \omega^{D} - L_{\omega} R_{F}^{C} \omega^{F} - L_{\omega} R_{F}^{C} J_{c} \dot{\theta}_{c}. \tag{46}$$

Taking the time derivative of $r_n(t)$ in (36) yields

$$\dot{r}_{p} = R_{F}^{C} \dot{v}^{F} + \dot{R}_{F}^{C} v^{F} + k_{p} \dot{e}_{p} + \dot{R}_{F}^{C} \delta. \tag{47}$$

Substituting from (3) and (34) into (47), utilizing the fact that

$$\dot{R}_F^C(\Theta) = R_F^C S(\omega_{CF}^F) = -R_F^C S(\omega_{FC}^F),$$

and subtracting and adding $\mathit{R_{F}^{C}S(\omega^{F})\delta}$ yields

$$\dot{r}_{p} = R_{F}^{C} \left[\frac{1}{m} N_{1}(v^{F}) - S(\omega^{F}) v^{F} + g R_{I}^{F} E_{3} + \frac{1}{m} F^{f} \right] + R_{F}^{C} S(\omega^{F}) \delta$$
(48)

$$+k_{p}[R_{F}^{C}v^{F}-S(\omega^{C})e_{p}-R_{D}^{C}v^{D}]-R_{F}^{C}S(\omega_{FC}^{F})v^{F}-R_{F}^{C}\left(S(\omega_{FC}^{F})+S(\omega^{F})\right)\delta.$$

Combining the angular velocities represented by the second term in the first bracket and the mid-term in the last row of (48) yields

$$-R_F^C S(\omega^F + \omega_{FC}^F) v^F = -R_F^C S(\omega_{IC}^F) v^F.$$
 (49)

The right-hand side of (49) can be further clarified using $\omega_{IC}^F = R_C^F \omega^C(t)$ and

$$S(\omega_{IC}^F) = R_C^F S(\omega^C) R_E^C, \tag{50}$$

to yield

$$-R_F^C R_C^F S(\omega^C) R_F^C v^F = -S(\omega^C) R_F^C v^F.$$
(51)

Combining the angular velocity terms $\omega_{FC}^F(t)$ and $\omega^F(t)$ in the last terms of the last row in (48) with $\omega_{IC}^F(t)$ yields

$$-R_F^C S(\omega^F) \delta - R_F^C S(\omega^{FC}) \delta = -R_F^C S(R_C^F \omega^C) \delta = -S(\omega^C) R_F^C \delta. \tag{52}$$

Multiplying both sides of (36) by $S(\omega^C)$ yields

$$-S(\omega^{C})r_{p} = -S(\omega^{C})(k_{p}e_{p} + R_{F}^{C}v^{C} + R_{F}^{C}\delta).$$
(53)

Substituting (51) through (53) into (48) yields

$$\dot{r}_{p} = \frac{R_{F}^{C} N_{1}(v^{F})}{m} - S(\omega^{C}) r_{p} + g R_{I}^{C} E_{3} + \frac{R_{F}^{C} F^{f}}{m} + k_{p} R_{F}^{C} v^{F} - k_{p} R_{D}^{C} v^{D} - R_{F}^{C} S(\delta) \omega^{F}.$$
 (54)

By taking the time derivative of r(t) in (35) and substituting (46) and (54) it can be obtained that

$$\dot{r} = \begin{bmatrix} \frac{R_F^C F^f}{m} - R_F^C S(\delta) \omega^F \\ -L_{\omega} R_F^C (\omega^F + J_c \dot{\theta}_c) \end{bmatrix} - \begin{bmatrix} S(\omega^C) r_p \\ O_{3x1} \end{bmatrix} + \begin{bmatrix} \frac{R_F^C N_1(v^F)}{m} + k_p (R_F^C v^F - R_D^C v^D) + g R_I^C E_3 \\ -L_{\omega} \omega^D \end{bmatrix}. (55)$$

Arranging the first term in (55) yields

$$\begin{bmatrix} \frac{1}{m} R_F^C F^f - R_F^C S(\delta) \omega^F \\ -L_{\omega} R_F^C \omega^F - L_{\omega} R_F^C J_c \dot{\theta}_c \end{bmatrix} = \overline{L}_{\omega} B \overline{U} = \overline{B} \overline{U}$$
 (56)

where

$$\overline{L}_{\omega} = \begin{bmatrix} R_F^C & O_{3x3} \\ O_{3x3} & -L_{\omega}R_F^C \end{bmatrix}, B = \begin{bmatrix} \frac{B_1}{m}, -S(\delta), O_{3x2} \\ O_{3x1}, I_{3x3}, J_c \end{bmatrix} \in R^{6x6}, \overline{U} = \begin{bmatrix} u_1 \\ \omega^F \\ \dot{\theta}_C \end{bmatrix} \in R^6, \quad (57)$$

along with the new term U(t) where $U_1(t) \in \mathbb{R}^3$ and $U_2(t) \in \mathbb{R}^3$, defined also further clarify the control design procedure.

$$U = \overline{B}\overline{U} = [U_1, U_2]^T \in R^6.$$
(58)

Substitution of (57) to (58) into (56) and then substitution of the resulting form of (56) using (55) produces the open-loop filtered error dynamics as follows

$$\dot{r} = \begin{bmatrix} -S(\omega^{C})r_{p} \\ O_{3x1} \end{bmatrix} + \begin{bmatrix} U_{1} \\ U_{2} \end{bmatrix} - \begin{bmatrix} k_{p}R_{D}^{C}v^{D} \\ L_{\omega}\omega^{D} \end{bmatrix} + \begin{bmatrix} \frac{R_{F}^{C}N_{1}(v^{F})}{m} + k_{p}R_{F}^{C}v^{F} + gR_{I}^{C}E_{3} \\ O_{3x1} \end{bmatrix}$$
(59)

where it is clear that the control inputs $U_1(t)$ and $U_2(t)$ will be designed to meet the control objective. Note that implementation of the control will require $\bar{U}(t)$ which is obtained from

$$\overline{U} = \overline{B}^{-1} U \in R^6, \tag{60}$$

which requires $\overline{B}^{-1} = B^{-1} (\overline{L}_{\omega})^{-1}$ where

$$(\overline{L}_{\omega})^{-1} = \frac{1}{\Delta(\overline{L}_{\omega})} \begin{bmatrix} -L_{\omega}R_F^C & O_3 \\ O_3 & R_F^C \end{bmatrix}, \tag{61}$$

in which $\Delta(\overline{L}_{\omega}) = \left| -R_F^C L_{\omega} R_F^C \right|$ is the determinant of the matrix $\overline{L}_{\omega}(t)$ and $\Delta(\overline{L}_{\omega}) \neq 0$ due to Assumption 1 (A1).

3.2 Lyapunov-based Control Design

The non-negative scalar function V(t) is chosen as

$$V = \frac{1}{2} e_p^T e_p + \frac{1}{2} r^T r.$$

Differentiating yields

$$\dot{V} = e_p^T \dot{e}_p + r_p^T \dot{r}, \tag{62}$$

by substituting (34) and (59) into (62) it can be obtained that

$$\dot{V} = [r_p^T, e_\theta^T]^T \begin{bmatrix} -S(\omega^C) r_p \\ O_{3x1} \end{bmatrix} + \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} + \begin{bmatrix} \frac{R_F^C N_1(v^F)}{m} + k_p R_F^C v^F + g R_I^C E_3 - k_p R_D^C v^D \\ L_\omega \omega^D \end{bmatrix}$$

$$+ e_p^T [-S(\omega^C) e_p + (r_p - k_p e_p - R_F^C \delta) - v^D]$$
(63)

where (36) was utilized. The terms in (63) can be collected to yield

$$\dot{V} = \left[r_{p}^{T}U_{1} + \frac{1}{m}r_{p}^{T}R_{F}^{C}N_{1}(v^{F}) + e_{p}^{T}r_{p} + r_{p}^{T}k_{p}R_{F}^{C}v^{F} + r_{p}^{T}gR_{I}^{C}E_{3} - r_{p}^{T}k_{p}R_{D}^{C}v^{D} - k_{p}e_{p}^{T}e_{p} - e_{p}^{T}R_{F}^{C}\delta - e_{p}^{T}R_{D}^{C}v^{D} + e_{\theta}^{T}U_{2} - e_{\theta}^{T}L_{\omega}\omega^{D}\right].$$
(64)

Equation (64) will be utilized to design the control inputs $U_1(t)$ and $U_2(t)$. From the upper equation in (64) we can design $U_1(t)$ to subtract out four terms, add stabilizing feedback, and add robust compensatation for the unknown nonlinear term as follows

$$U_{1} = -k_{r}r_{p} - \frac{r_{p}\zeta_{1}^{2}(\|v^{F}\|)}{\varepsilon_{0}} - k_{p}R_{F}^{C}v^{F} - gR_{I}^{C}E_{3} - e_{p}$$
(65)

where \mathcal{E}_0 is a positive constant, A2 is used for $\zeta_1(\|v^F\|)$, and the nonlinear term $\overline{N}_1(t)$ is defined by

$$\overline{N}_{1} = \frac{1}{m} R_{F}^{C} N_{1}(v^{F}). \tag{66}$$

From the lower equation in (59), $U_2(t)$ can be designed as

$$U_2 = L_\omega \omega^D - k_\theta e_\theta. \tag{67}$$

Substituting (65) and (67) into (64) yields

$$\dot{V} = r_p^T (\bar{N}_1 - \frac{r_p \zeta_1^2 (\|v^F\|)}{\varepsilon_0}) - k_r r_p^T r_p - r_p^T k_p R_D^C v^D - k_\theta e_\theta^T e_\theta - k_p e_p^T e_p - e_p^T (R_F^C \delta + R_D^C v^D).$$
(68)

According to sufficient condition, finally it yields

$$\dot{V} \le -\lambda_2 \|\eta\|^2 + \varepsilon_4. \tag{69}$$

4. Stability Analysis

The closed-loop control law of (65) and (67) ensure that the tracking error is Globally Uniformly Ultimately Bounded (GUUB) in the manner

$$\|\eta\| \le \sqrt{\|\eta(0)\|^2 e^{-2\lambda_2 t} + \frac{\mathcal{E}_4}{\lambda_2}}$$
 (70)

where

$$\boldsymbol{\eta} = [\boldsymbol{r}_{p}^{T}, \boldsymbol{e}_{\theta}^{T}, \boldsymbol{e}_{p}^{T}]^{T}, \tag{71}$$

 \mathcal{E}_4 is a positive constant, and λ_2 is a positive constant given by the following form

$$\lambda_2 = \min\left\{ (k_r - \frac{\lambda_1}{2}), (k_p - \frac{\lambda_0}{2}), k_\theta \right\}$$
 (72)

where λ_0 , λ_1 are positive constants under the conditions that

$$k_r > \frac{\lambda_1}{2}$$
 and $k_p > \frac{\lambda_0}{2}$. (73)

A proof of the theorem can be proven using Lyapunov-type stability analysis (This is beyond the scope).

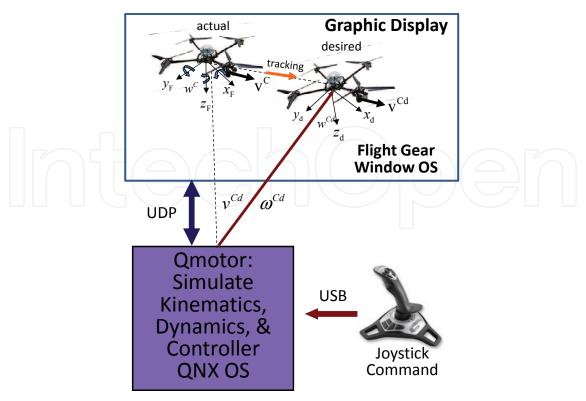


Figure 11. System Overview

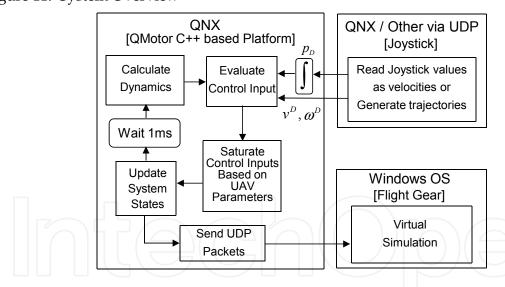


Figure 12. System Interal Outline in Simulation System

5. Simulation

The simulation system overview is given Fig. 11. A two computer system, QNX and Windows, was built to simulate the proposed controller. The QNX software systems are configured to run QNX Real-Time Operating System (RTOS) and host the QMotor (2000) control and simulation package while the Windows XP computer is configured to run and host the FlightGear (v.0.9.10) (2006) open-source flight simulator package. A QMotor program was written to simulate the rigid body kinematics, dynamics, and the proposed

control. The output of the dynamics simulation is sent via UDP to set the aircraft/camera position and orientation in the FlightGear virtual world (note that FlightGear is used only as a graphics processor). The desired trajectories are input by the operator using a 6 DOF joystick (Logitech Extreme 3D Pro, 2006). Specifically, the operator indirectly supplies the desired position trajectory used by the controller through monitoring the simulated camera view and using the joystick to command the velocities that move the camera view in the virtual world. A detailed block diagram showing the three inputs on the joystick, labeled as x, y, and twist, are used to generate and control either three translational velocities, $v^D(t)$, or the three angular velocities, $\omega^D(t)$, depending on trigger position. That is, the magnitude and direction of these quantities is derived from the joystick position. The velocity commands are then integrated to produce the desired position trajectory used by the controller. A typical scene from FlightGear (2006) is shown in Fig. 13 where the quadrotor is tilted but the camera view seen by the operator remains level. The quadrotor simulation was developed to approximate the parameters of the DraganFlyer X-Pro (2005).

$$k_r = k_\theta = diag(1,1,1), k_p = diag(1,1,5), m = 2.72[kg] \text{ and } g = 9.81[m^3 / kgs^2],$$

 $u_{1_{\text{max}}} = 35.586 [N], \theta_{tilt_{\text{max}}} = \theta_{roll_{\text{max}}} = 4.067 [Nm].$

Parameters such as mass (*m*) and saturation limits for control inputs, and the control gains

A short timespan of the simulation was captured to demonstrate the operation of the system. The simulation results are shown in Fig. 14 through Fig. 19. The easiest to appreciate operational point is found at t=50 [sec] in Fig. 17 and Fig. 18 where the orientations of the camera and quadrotor rotate in opposite directions measured from the x-, y-, and z-axes to achieve the fly-the-camera perspective. This also can be shown by the torque inputs of UAV roll, the second plot in Fig. 16, and camera roll, the last plot in Fig. 16, which acts in opposite directions as the UAV torque produces the required lateral motion and the camera compensates for the resulting roll. This is also seen in the pitch input in the third plot of Fig. 16 and the camera tilt input in the fourth plot of Fig. 16. Fig. 14 shows the position tracking of the quad-rotor to the desired trajectory and Fig. 15 shows the position errors about the coordinates. The actual quad-rotor trajectory represented by the dotted line follows the desired trajectory represented by the solid line which is commanded to go up at the first time and then, move to forward and again go forward near the end.

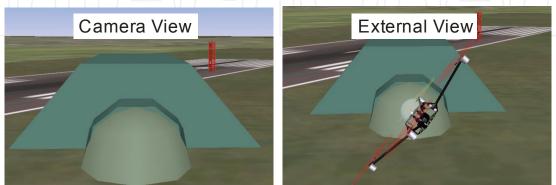


Figure 13. The "fly-the-camera" view used by the operator and an outside view of the quadrotor position

are chosen to be

4

5.1 Simulation Results

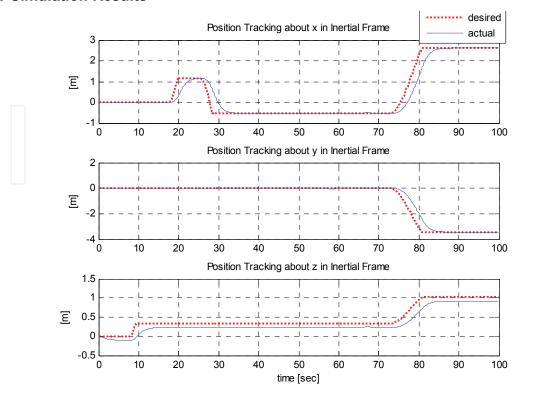


Figure 14. Position Tracking

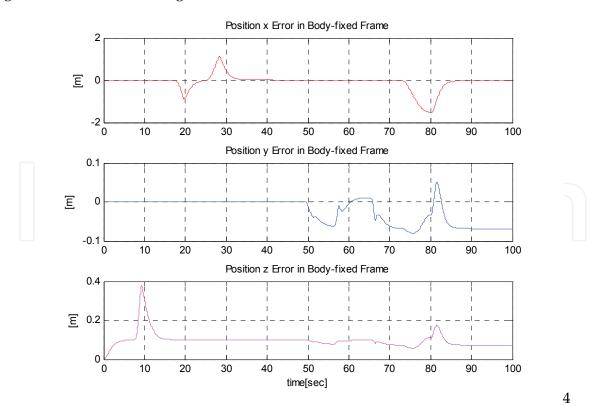


Figure 15. Tracking Errors

www.intechopen.com

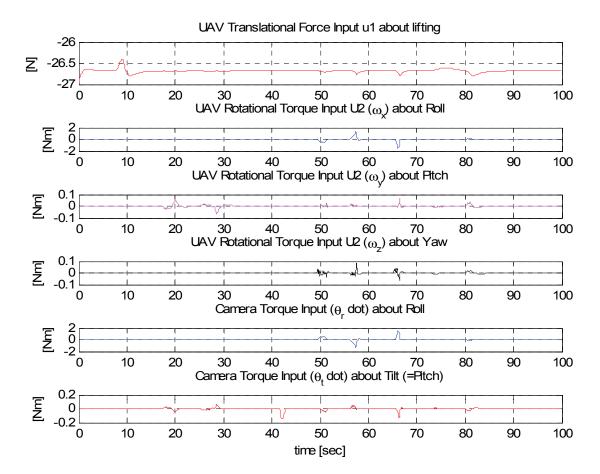


Figure 14. Control Inputs

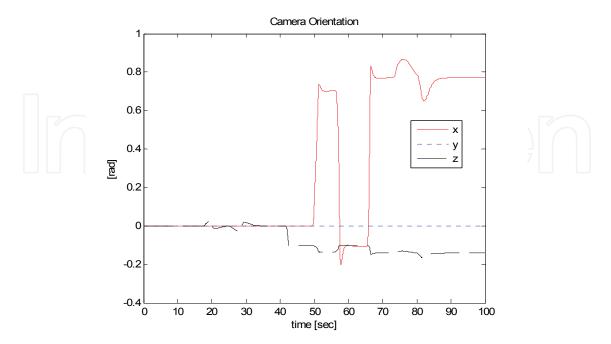


Figure 15. Camera Orientation

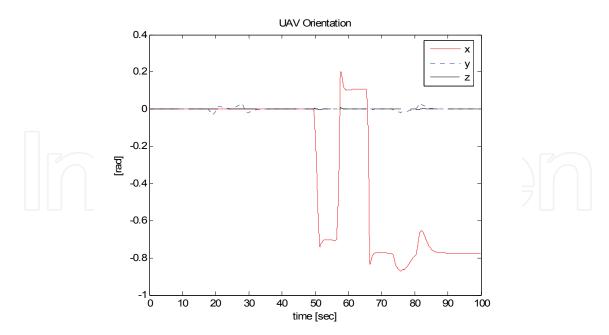


Figure 18. UAV Orientation

6. Conclusion

We have described the fly-the-camera approach to concurrent control of a camera positioner and unmanned aerial vehicle in an operator friendly manner. We detailed the design of a nonlinear controller as one possible embodiment of this philosophy. The fly-the-camera approach considers a single robotic dynamic object which consists of an underactuated aerial vehicle and two complementary camera axes to produce a fully actuated camera targeting platform. The fly-the-camera approach should provide a more intuitive perspective for a single remote pilot to operate the quadrotor vehicle and camera for surveillance, navigation, and landing/monitoring tasks. The approach fuses the often separate tasks of vehicle navigation and camera targeting into a single task where the pilot sees and flies the system as through riding on the camera optical axis. The controller detailed here was shown to provide position and angle based tracking in the form of Globally Uniformly Ultimately Bounded (GUUB) result. The simulation results were shown to demonstrate the proposed system. The development of fly-the-camera perspective system can provide a platform for many areas of commercial, industrial, or research work.

The demonstration of the "fly-the-camera" concept and a control methodology provide the foundation for expanding this approach. Two obvious points would extend the suitability of this work. First, only the kinematic model of the camera has been included. This approximation is only valid when the camera mass and acceleration forces are small relative to the airframe. Second, cognitive loading studies should be performed to evaluate the true potential of this approach.

7. Appendix

7.1 Case 2: Pan-Tilt Camera Configuration (camera looking downward)

The Denavit-Hartenberg (D-H) table for the Pan-Roll configuration when the optical axis is looking downward as shown in Fig. 7 is given in Table II.

Link (i)	Offset	Length	Twist	Angle
	D	a	α	θ
1	0	0	+90°	$\theta_p + 90^{\circ}$
2 (downward axis)	0	0	-90°	$\theta_{\rm t}$ – 90°

Table 2. D-H for Pan-Tilt Camera Positioner while the optical axis is looking forward

The rotation matrices of the Pan-Tilt camera positioning unit for first and second links as shown in Fig. 8 are obtained as

$$R_1^B = \begin{bmatrix} -s\theta_p & 0 & c\theta_p \\ c\theta_p & 0 & s\theta_p \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad R_2^1 = \begin{bmatrix} s\theta_t & 0 & c\theta_t \\ -c\theta_t & 0 & s\theta_t \\ 0 & -1 & 0 \end{bmatrix}$$
(74)

where the first link rotation matrix for panning when the camera looking downward is the same as that of matrix for rolling motion when looking forward as shown in Fig. 6 and 7. Next, we can obtain the matrix from the camera base to first link (also same) but the total rotation matrix by combining all those matrices from quadrotor through the second link to yield

$$R_{1}^{F} = \begin{bmatrix} 0 & 1 & 0 \\ -s\theta_{p} & 0 & c\theta_{p} \\ c\theta_{p} & 0 & s\theta_{p} \end{bmatrix} \quad \text{and} \quad R_{2=C}^{F} = \begin{bmatrix} -c\theta_{t} & 0 & c\theta_{t} \\ -s\theta_{t}s\theta_{p} & -c\theta_{p} & -c\theta_{t}s\theta_{p} \\ s\theta_{t}c\theta_{p} & -s\theta_{p} & c\theta_{t}c\theta_{p} \end{bmatrix}$$

$$(75)$$

where $R_1^F = R_B^F R_1^{B=O_0}$ and $R_{2=C}^F = R_1^F R_{2=C}^1$ were used to calculate and the third column in R_1^F is the vector $Z_1(t)$ in the Jacobian matrix $J_C(t)$. Thus, the positioner joint angles, $\dot{\theta}_C(t) \in R^3$, are given by

$$J_{C} = \begin{bmatrix} z_{0} & z_{1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & c\theta_{p} \\ 0 & s\theta_{p} \end{bmatrix}, \ \dot{\theta}_{C} = \begin{bmatrix} \dot{\theta}_{p=r} \\ \dot{\theta}_{t} \end{bmatrix}, \text{ and } \theta_{C} = \begin{bmatrix} \theta_{p=r} \\ \theta_{t} \end{bmatrix}. \tag{76}$$

Note that Jacobian matrix of the Pan-Tilt camera manipulator in (76) is the same one used in the Tilt-Roll configuration due to the configuration of the same camera base and the first link. In addition, the pan angle in Pan-Tilt configuration is actually the same of roll used in Tilt-Roll configuration due to the actuation of the second link.

7.2 Rotation Matrix

A rotation matrix, $R_D^C(\theta_d)$ denoted in (20), from D to desired frame C for the tracking of the camera frame to desired frame can be obtained by considering the camera frame with regard to the desired frame as another end link as follows:

Link (i)	Offset	Length	Twist	Angle
	D	a	α	heta
C (looking forward)	0	0	+90°	$\theta_d + 90^\circ$
C (looking downward)	0	0	0°	$\theta_d + 90^{\circ}$

Table 3. D-H Table to find the rotation matrices between the camera and desired frames which yields the following rotation matrix, respectively (for simulation, $\theta_d = 0^\circ$)

$$R_D^C = \begin{bmatrix} -\mathbf{s}\,\boldsymbol{\theta}_d & 0 & \mathbf{c}\,\boldsymbol{\theta}_d \\ \mathbf{c}\,\boldsymbol{\theta}_d & 0 & \mathbf{s}\,\boldsymbol{\theta}_d \\ 0 & 1 & 0 \end{bmatrix}_{(forward)} \quad \text{and} \quad R_D^C = \begin{bmatrix} -\mathbf{s}\,\boldsymbol{\theta}_d & -\mathbf{c}\,\boldsymbol{\theta}_d & 0 \\ \mathbf{c}\,\boldsymbol{\theta}_d & -\mathbf{s}\,\boldsymbol{\theta}_d & 0 \\ 0 & 0 & 1 \end{bmatrix}_{(downward)} .$$

7.3 Notation and Nomenclature

 R_i^{i-1} Rotation matrix from the origin i to the origin i-1 coordinate frame

 $T_{\!F}^I$ Orientation matrix from the origin of F to the origin of frame I

 p_i , \dot{p}_i , Camera (C) or UAV (F) position and position rate represented in Inertial frame I - ground-based - quantities denoted using subscript

 Θ_i , $\dot{\Theta}_i$ Camera or UAV angle $([\varphi, \theta, \psi]^T)$ and angular rates about roll, pitch, and yaw represented in I frame -ground-based-quantities using subscript

 Θ_C Camera angles about pan, tilt, and roll $([\theta_n, \theta_t, \theta_r]^T)$ expressed in the Inertial frame

 v^i , ω^i Camera (C), Camera base (B) or UAV linear and angular velocities about x, y, and z-axis which is airborne quantities denoted using superscript

 v^{ij} , ω^{ij} Velocities expressed in Camera frame (C) between the origin i and another origin j coordinate frame

 v_{jk}^i , ω_{jk}^i UAV or Camera base frame velocities (i) denoting specific quantities between j frame and k frame

8. References

Adelstein, B. D.; Ellis, S. R. (2000). Rotation and Direction Judgment from Visual Images Head-Slaved in Two and Three Degrees-of-Freedom, *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 30, No. 2, March 2000, pp. 165 – 173.

Chitrakaran, V.; Dawson, D. M.; Kannan, H. (2006). Vision Assisted Autonomous Path Following for Unmanned Aerial Vehicles, in *Proc of 45th IEEE Conference on Decision and Control*, pp. 63 - 68, Dec. 2006.

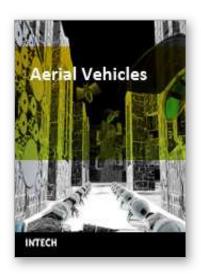
Cooke, N; Pringle, H.; Pedersen, H.; Connor, O. (2006). Human Factors of Remotely Operated Vehicles, *ELSVIER JAI*, 2006

DraganFlyer X-Pro (2005): http://www.rctoys.com/ (cited 10/2008).

FlightGear (2006). http://www.flightgear.org/ (cited 10/2008).

Fossen, T. I. (2003). Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs, and Underwater Vehicles, *Marine Cybernetics*, 2003.

- Jakobsen, O. C.; Johnson, E. N. (2005). Control Architecture for a UAV Mounted Pan/Tilt/Roll Camera Gimbal, AIAA 2005-7145, Infotech@Aerospace, Arlington, Virginia, Sep. 2005, pp. 01 - 10.
- Lee, D.; Chitrakaran, V.; Burg, T.C.; Dawson, D. M.; Xian, B. (2007). Control of a Remotely Operated Quadrotor Aerial Vehicle and Camera Unit Using a Fly-The-Camera Perspective, in *Proc of 46th IEEE Conference on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 6412-6417.
- Logitech Extreme 3D Pro Joystick (2006); http://www.logitech.com
- Mahony, R.; Hamel, T.; Chaumette, F. (2002). A Decoupled Image Space Approach to Visual Servo Control of a Robotic Manipulator, in *IEEE Intl Conference on Robotics and Automation*, Washington D. C., May 2002, pp. 3781 3786.
- Malis, E. (1998) Controbutions a la modelisation et a la commande en asservissement visuel, *Ph.D. Thesis*, University of Rennes I, IRISA, France, 1998.
- Micropilot.; http://www.micropilot.com/ (9/2008 cited)
- Murray, R. M.; Li, Z.; Sastry S. S. (1994). A Mathematical Introduction to Robotic Manipulator, CRC Press, 1994.
- Neff, A.; Lee, D.; Chitrakaran, V.; Dawson, D. M.; Burg, T. C. (2007). Velocity Control of a Quadrotor UAV Fly-By-Camera Interface, in *Proceedings of IEEE Southeast Conference*, pp. 273-278, Richmond, VA, Mar. 2007.
- Pieniazek, J. (2003). Software-based Camera Stabilization on Unmanned Aircraft, Aircraft Engineering and Aerospace Technology: *An International Journal*, Vol. 75, No. 6, 2003, pp. 575 580.
- Procerus Technologies.; http://www.procerusuav.com/ (9/2008 cited)
- Qmotor Simulator (2000). A QNX-based C++ Real-time Control Environment, http://www.ece.clemson.edu/crb/research/realtimesoftware/qmotor/index.htm (cited 10/2008).
- QNX 6, QNX Neutrino 6.2.1. (2006). QNX Software Systems, Real-Time Control Software. http://www.qnx.com (cited 10/2008)
- Quigley, M.; Goodrich, M. A.; Griffiths, S.; Eldredge A.; Beard, R. W. (2005). Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera, in *IEEE Intl Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2611 2616.
- Sharp, C. S.; Shakernia, O. & Sastry S. S. (2001). A Vision System for Landing an Unmanned Aerial Vehicle, in *IEEE Intl Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 1720-1727.
- Spong, M. W.; Hutchinson, S.; Vidyasagar M. (2006). Robot Modeling and Control, John Wiley and Sons, Inc., 2006.
- Stanciu, R. & Oh. P. (2007). Human-in-the-Loop Camera Control for a Mechatronic Broadcast Boom, *IEEE/ASME Transactions on Mechatronics*, Vol. 12, No. 1, February 2007.
- Stolle, S. & Rysdyk, R. (2003). Flight Path Following Guidance for Unmanned Air Vehicles with Pan-Tilt Camera for Target Observation, in *Proceedings of the AIAA Digital Avionics Systems Conference*, Oct. 2003, pp 01 12.
- Yoon, S.; Lundberg, J. B. (2001). Equations of Motion for a Two-Axes Gimbal System, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 3, July 2001, pp. 1083 1091.



Edited by Thanh Mung Lam

ISBN 978-953-7619-41-1
Hard cover, 320 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

This book contains 35 chapters written by experts in developing techniques for making aerial vehicles more intelligent, more reliable, more flexible in use, and safer in operation. It will also serve as an inspiration for further improvement of the design and application of aeral vehicles. The advanced techniques and research described here may also be applicable to other high-tech areas such as robotics, avionics, vetronics, and space.

How to reference

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

DongBin Lee; Timothy C. Burg; Darren M. Dawson and Guenther Dorn (2009). Fly-the-Camera Perspective: Control of a Remotely Operated Quadrotor UAV and Camera Unit, Aerial Vehicles, Thanh Mung Lam (Ed.), ISBN: 978-953-7619-41-1, InTech, Available from: http://www.intechopen.com/books/aerial_vehicles/fly-the-camera_perspective__control_of_a_remotely_operated_quadrotor_uav_and_camera_unit

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 © 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the <u>Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License</u>, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.



