JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 2, January 2005

A New Approach to Future Mars Missions Using Bioinspired Technology Innovations

Randal Beard,* and D. J. Lee,* and Morgan Quigley*
Brigham Young University, Provo, Utah 84602

Sarita Thakoor†
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109

and
Steve Zornetzer ‡
NASA Ames Research Center, Moffet Field, California 94035

Bioinspired technology breakthroughs of insect inspired navigation are enabling small aircraft that will allow exploration of Mars. The bioinspired technologies under development consist of a bioinspired navigation-control system, human vision inspired and birds of prey inspired search and track systems. Two classes of new missions for Mars exploration: 1) the long range exploration missions and 2) critical ephemeral phenomena observation missions can be enabled by incorporation of these bioinspired technology breakthroughs in such flyers. In this paper we describe our implementation of an image based guidance algorithms that can be used for imaging the descent and landing of mission payloads on Mars, thus potentially enabling a "black box" capability as a unique risk mitigation tool for landed missions of all types. Further on, such capabilities could enable observation of a variety of other ephemeral phenomena such as tracking dust storms in real-time.

I. Introduction

We are currently developing autonomous biologically inspired unmanned flyers that can potentially be used in a variety of missions to Mars. As previously described in Refs. 1-8 unmanned flyers can be used to provide closeup imaging of the Mars surface, explore canyons and other regions on Mars that are not accessible to rovers, enhance map resolution thus increasing path planning capabilities for rovers, and provide ad hoc communication relays between landers on the Mars surface. There is also significant interest in observing the descent and landing of mission payloads sent to Mars. Accordingly, an unmanned flyer could be launched during the descent of a Mars mission package, and used to observe the descent and landing of the main payload. Used in this capacity, the unmanned flyer performs a "black box" function: following the descent of the lander, imaging the event, storing the critical data to provide an onlookers view of descent and landing of the payload.[4,8]

With current state-of-the-art technologies, flyers on Mars are not feasible. Challenges that must be addressed include airframe design, Mars suitable sensor design, low Reynolds number flight control, and autonomous navigation

This paper is part of the December Special Section on Intelligent Systems. Received 17 September 2004; revision received 5 November 2004; accepted for publication 10 November 2004.

*Associate Professor; beard@ee.byu.edu.
†Manager, "BEES for MARS," NASA Ames-JPL Joint Project, MS 1710; sarita.thakoor@jpl.nasa.gov.
‡Deputy Director for Research, Mail Stop I:200-3; szornetzer@mail.arc.nasa.gov.

without GPS and magnetometers. Since GPS will not be available and magnetometers designed for the earth's magnetic field will not likely be effective on Mars, additional sensors must be used. In Refs. 2, 5, 7, 4, 6, 8 development of sensors to address these needs has been described. In this paper we focus on the use of image directed control of an autonomous unmanned flier to image a Mars payload during its descent and landing. In Section 6, we develop a novel guidance and control algorithm that uses image data in the feedback loop to maintain a specific orbit radius around the payload. The image data is used to identify and track critical features of the descending payload and will be used to control the UAV to maintain the image in the center of the camera field-of-view. A computer vision algorithm based on color segmentation will be used to track the payload. To explore the feasibility of this mission, an earth-based experiment that uses two unmanned mini-UAVs is described in Section 2. One of the UAVs, termed the imager, is equipped with a camera mounted on a pan-tilt gimbal. The other UAV, termed the payload, is equipped with a parachute that can be deployed from a safe altitude. The imager will use camera data to image the payload as it descends. The results of this development are being presented here as the latest proof-of-concept of the capabilities of such flyers utilizing bioinspired algorithms and technologies which hold the potential for enabling close-up observation of descent and landing as a possible risk mitigation tool for all future landed missions. A study to assess the relative meritcomplexity of deploying such an observation flyer is underway and will be presented elsewhere.

2 Potential Mars Scenario

We will model the flight tests described in this paper after a potential Mars landing scenario based on the Mars Pathfinder Mission.[9] A possible time line for an actual Mars mission is listed below, where EOL is the estimated-time-of-landing.

1) $EOL =$ 204 seconds. The spacecraft enters the Mars atmosphere at a speed of 7600 m/s and an altitude of 125 km.

2) $EOL =$ 140 seconds. The parachute is deployed. The spacecraft velocity is approximately 400 m/s, with an altitude of 7 km.

3) $EOL =$ 120 seconds. The heat shield separates from the payload and the imager is separated from the main payload. Descent velocity of the payload is 100 m/s and initial velocity of the imager is 120 m/s. Altitude is 5 km.

4) $EOL =$ 80 seconds. The payload separates and the bridle is deployed. Descent velocity of the payload is 65 m/s. Velocity of the flyer is 100 m/s. Altitude of the payload is 3 km. Altitude of the imager is approximately 3.5 km.

5) $EOL =$ 10 seconds. The airbags inflate. Descent velocity of the payload is 60 m/s. Velocity of the imager is 100 m/s. Altitude of the payload is 300 m. Altitude of the imager is 800 m.

6) $EOL =$ 0 seconds. The payload impacts the Mars surface. Velocity of the imager is 100 m/s. Altitude of the imager is 500 m.

The key technical challenges that must be addressed by the autonomous guidance and control algorithms are: (a) the initial acquisition of the payload in the camera field-of-view of the imager, (b) the use of computer vision to identify and track features on the payload, (c) correctly servo the pan and tilt gimbal to maintain the image in the

center of the camera field-of-view, and (d) use the image data to control the imager to maintain a circular orbit around the descending payload.

In this paper we will focus on items (b), (c), and (d). We envision the following demonstration scenario which is depicted graphically in Figure 1.

1) Two unmanned autonomous flyers are launched from home base. The imager is equipped with a camera mounted on a pan and tilt gimbal. The payload flyer is equipped with a deployable parachute.

2) The imager is launched first and commanded to autonomously orbit the payload flyer with a vertical separation of 100 meters and an orbit radius of 100 meters. The payload flyer is then launched and commanded to fly to a waypoint with a vertical height of 500 meters.

3) When the payload flyer is at its destination, the orbit of the imager is tightened to 50 meter vertical separation and 75 meter orbit radius.

4) The payload flyer deploys its parachute and within 20 seconds reaches a constant descent velocity of 0.78 m/s.

5) The imager continues to orbit the payload, maintaining the image of the payload flyer in its camera field-of-view with a constant vertical separation of 50 m.

6) When the payload touches ground, the imager continues to image the payload for an additional sixty seconds, and then lands autonomously.

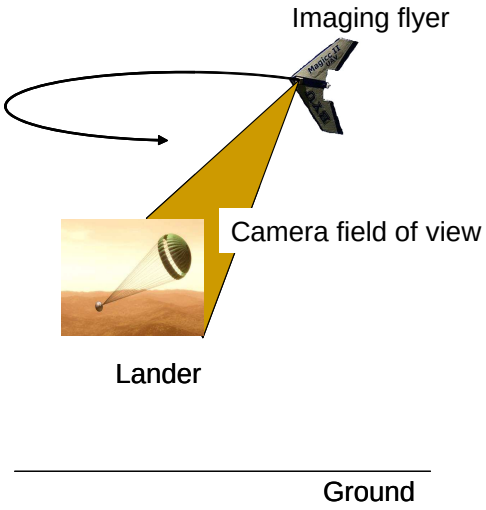

Fig. 1 Demonstration scenario. The imager maintains the payload and parachute constantly in its field of view.

The earth mock-up scenario will use GPS signals to initially acquire the image. In an actual Mars scenario, GPS signals will not be available to guide the acquisition process. One possible approach to image acquisition is as follows. When the imager is separated from the payload, it could use a Kalman filter to predict the location of the payload until the vision system is able to acquire it. Alternatively, if the payload becomes lost, the imager would execute a climb

maneuver to increase the region visible to the camera, and the pan-tilt gimbal would execute a search pattern until the payload is acquired.

A simulation of the observation scenarios is shown in this AVI movie.

3 Biological Inspiration

The vision algorithms described in this paper are inspired by the biological vision systems of birds of prey. Motion detection is an important and fundamental visual function of biological vision systems. It requires frequent eyeball movement in making small visual shifts in order to capture the image of an interesting target on the retinal fovea.[10,11,12] Frequent eyeball movement and target motion create optical flow on the retina. Bio-inspired vision and control algorithms mimicking these operations will be developed to measure the motion, and the target size and position in the image. While these operations could be performed with stereo vision, the limited payload of the imager necessitates single camera techniques.

Birds essentially perform two tasks while hunting: seeking and tracking. To seek prey, they employ two primary senses: smell (olfactory) and sight (vision); however, smell does not contribute nearly as much as vision.[13,14] Both birds and insects use UV light to augment their vision in the visible light spectrum.[14,15,16] Insects and birds use the UV spectrum to identify food sources, so a frequency (color) based algorithm for seeking targets seems appropriate. Once tracking something in-flight, insects generally use optical flow to keep themselves oriented correctly.[17,18] Birds also use optical flow to stabilize their eyes when their body is moving.[19,20] Birds demonstrate specialization at different fovea in the eye for extremely high-resolution tracking. This actually contributes to a spiraling approach toward the prey, since there are only certain positions of the head and body that allow this high-resolution tracking.[21] Coordination between optical flow and inertial measurements (the vestibular system, or ”inner ear”) helps birds in flight[19,20] and provides stabilization for later viewing of the video captured by the imaging flyer.

The tracking algorithm described in this paper is also inspired by the tracking behavior of birds of prey. Since birds travel at a much greater speed than their prey, they execute an orbiting motion about their prey during the tracking phase. However, the orbiting motion is not constrained by time stamped positions along the intended flight

path. Therefore traditional trajectory generation techniques are not appropriate. Alternatively, birds maintain a fixed distance to their prey, modifying their flight path for high wind conditions and other environmental disturbances. In accordance with these principles, we will develop a navigation algorithm that is based on the distance to the desired orbit, adjusting the flight path to accommodate high wind conditions and other environmental disturbances. Simulation results for high wind conditions demonstrate motion that is qualitatively similar to the motion of birds in heavy winds.

4 Motion Extraction Using Color Segmentation

The first task is to identify the location of the payload in the image. In this paper we will focus on identification of the parachute in the camera image. The parachute has been colored to aid the computer vision algorithms. Color is the perceptual result of light in the visible range of the spectrum incident upon the human retina. The human retina has three types of color photo-receptor cone cells which respond to incident radiation with different spectral response curves. Because there are three color photo-receptors, three numerical components are necessary and sufficient to describe colors for color image acquisition or display. Therefore, most color spaces are three-dimensional. Each pixel of the image requires three bytes to store its color information.

Using 3-D data sets for color grading is compute-intensive because large amounts of data must be processed to evaluate colors. It also requires extensive memory for processing. A 3-D color space would not be efficient for machine vision applications, especially for those requiring real-time performance. Additionally, separating colors in a 3-D space is not an easy task, either visually or numerically.

Most off-the-shelf machine vision camera systems digitize color information into 3 channels of 8-bit data, one for each of the red, green, and blue channels. These 24-bit data can represent over 16 million colors. However, in our application only a single color is of interest suggesting a segmentation approach. There are two issues that complicate the color segmentation task. One is the illumination variations and the other is the imperfection of color image acquisition. The segmentation algorithm must be invariant to illumination variations as well as slight color variations due to digitization error.

Inspired by the human vision system's ability to segment specific colors under different lighting conditions, we have developed a color segmentation algorithm that is not affected by illumination variations and is capable of detecting similar colors. Our method converts the red, green, and blue (RGB) data into hue, saturation, and intensity (HIS) color space. We then discard the intensity component that is sensitive to illumination variations. Unlike most other color segmentation techniques that use only hue component for segmentation, we use both hue and saturation and apply two-dimensional thresholds to these two color components. By carefully selecting these thresholds, through the help of a simple calibration procedure, the targeted color can be robustly segmented.

Figure 2 shows a flowchart of the proposed algorithm. Input RGB is first converted to HIS color space. H and S components are then converted to binary using two thresholds (low and high). The two binary H and S images are then passed through an $< AND >$ operation to provide a single binary image of the segmented color. This image is used to compute the parachute's size and centroid which are used to determine (1) the size of the parachute in the image, denoted r_{img}, and (2) the location of the parachute in the image, denoted (y_{img}, z_{img}) .

Our color segmentation algorithm was tested on six parachutes with different colors: red, blue, yellow, white, purple, and light green. Red seemed to give the best results. A flight experiment was conducted where a UAV was directed to pass over the parachutes and record the image data. The image data was then post processed to tune the vision algorithm.

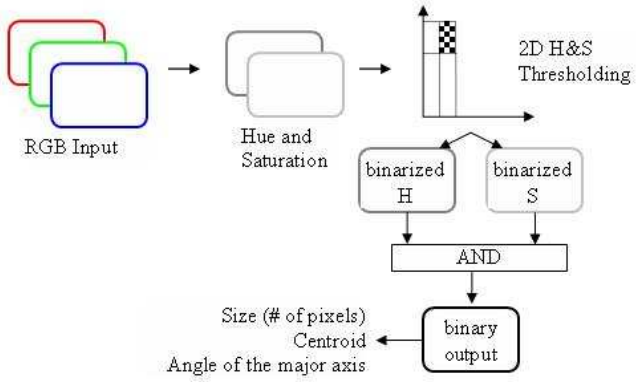

Fig. 2 Computer vision color segmentation algorithm.

The link below shows a movie file of the computer vision results using a video camera mounted in a UAV. The first segment of the movie shows the parachutes on the ground, imaged from a UAV flying at approximately 100 meter. The second segment shows the color segmentation algorithm which has extracted the red parachute. The large smudge in the image is due to a scratched cover on the camera gimbal. The vision algorithm was tuned so that the parachute can be tracked in real time, finding the image in approximately 60% of the frames.

An AVI movie showing computer vision results.

5 Camera Point Algorithm

When birds-of-prey are hunting small animals, they maintain their prey in the center of their view to take advantage of the visual acuity provided by the deep fovea in the center of their eyes.[21] In a similar way, we will use a camera mounted on a pan and tilt gimbal to maintain the parachute in the center of the image. The objective of this section is to briefly describe an algorithm that can be used to point the camera at a world coordinate expressed in the vehicle frame.

There are several coordinate frames that are involved in the problem.

The inertial frame $\mathscr{C}_I$. The inertial coordinate system is a Mars (Earth) fixed coordinate system with origin at the defined home location. The x-axis of $\mathscr{C}_I$ points North, the y-axis points East, and the z axis points toward the center of Mars (Earth).

The vehicle frame $\mathscr{C}_v$. The origin of the vehicle frame is at the center of mass of the UAV. However, the axes of $\mathscr{C}_v$ are aligned with the axis of the inertial frame $\mathscr{C}_I$. In other words, the x-axis points North, the y-axis points East, and the z-axis points toward the center of Mars (Earth).

The body frame $\mathscr{C}_b$. The origin of the body frame is also at the center of mass of the UAV. The x-axis points out the nose of the UAV, the y-axis points out the right wing, and the z-axis point out the belly. As the attitude of the

UAV moves, the body frame remains fixed with respect to the airframe.

The camera frame $\mathscr{C}_c$. The origin of the camera frame is located at the center of the image, which we will assume to be at the center of mass of the UAV. The *x*-axis points along the view axis of the camera, the *y*-axis is located in the image plane and points to the right as seen by a person viewing the image. The *z*-axis is also located in the image plane and points down as seen by a person viewing the image. A graphical representation of the camera frame is shown in Figure 3.

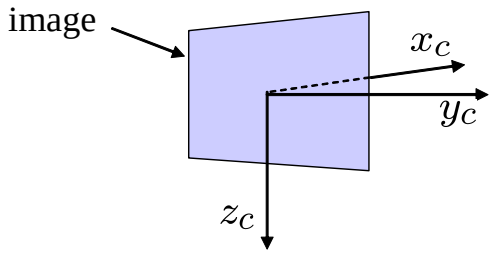

Fig. 3 A representation of the coordinate frame attached to the camera.

.

The transformation between the vehicle frame and the body frame is given by

$$\begin{aligned} R_{v\to b} &= R_{\text{roll}}(\phi)R_{\text{pitch}}(\theta)R_{\text{yaw}}(\psi) \\ &= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix}, \end{aligned}$$

where $c\varphi = \cos(\varphi)$ and $s\varphi = \sin(\varphi)$. Therefore, if $\mathbf{p}$ is a vector, and $p_b, p_v \in \mathbb{R}^3$ are its representation in $\mathscr{C}_b$ and $\mathscr{C}_v$ respectively, then

$$\begin{aligned} p_b &= R_{v\to b}p_v \text{ and} \\ p_v &= R_{b\to v}p_b = R_{v\to b}^T p_b. \end{aligned}$$

If the location of the origin of the vehicle frame in the inertial frame is given by $\mathbf{z}$, then the transformation from the inertial frame to the vehicle frame is given by

$$\mathbf{p}_v = \mathbf{p}_I - \mathbf{z}.$$

As discussed in the previous section, the vision system identifies (1) the pixel coordinates of the parachute, denoted (y_{img}, z_{img}), and (2) the size of the parachute in the image, denoted as r_{img}. We will also assume that we know the actual size of the parachute, denoted R_{img}. From projective geometry,[22] we know that the location of the object in the camera frame is given by

$$p_c = \begin{pmatrix} f\frac{R_{obj}}{r_{img}} \\ y_{img}f^2\frac{R_{obj}}{r_{img}} \\ z_{img}f^2\frac{R_{obj}}{r_{img}} \end{pmatrix}, \tag{1}$$

where f is the focal length of the camera. Therefore, the location of the object in the body frame is given by

$$
\begin{aligned}
p_b &= R_{c\to b} p_c \\
&= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \begin{pmatrix} f\frac{R_{obj}}{r_{img}} \\ y_{img} f^2 \frac{R_{obj}}{r_{img}} \\ z_{img} f^2 \frac{R_{obj}}{r_{img}} \end{pmatrix}.
\end{aligned}
$$

Given the body frame coordinates of the object, the next objective is to command the gimbal servos so that the camera points directly at the object. The orientation of the camera frame with respect to the body frame is given by two angles: the azimuth angle α_{az} and the elevation angle α_{el}. An arbitrary orientation of the camera, represented by $(\alpha_{az}, \alpha_{el})$ is defined as follows. The camera and body axes are originally aligned, and the camera is first rotated by an angle of α_{az} about the body z-axis as shown in Figure 4. Label the new (intermediate) coordinate frame as $\mathscr{C}_1$ as shown in Figure 4. The camera is then rotated an angle of $-\alpha_{el}$ about the y-axis of $\mathscr{C}_1$ as shown in Figure 5 to obtain the camera frame. Therefore, we have

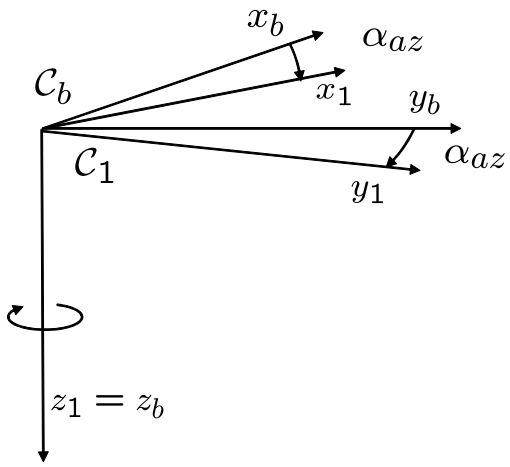

Fig. 4 The azimuth angle is defined as a rotation about the body z-axis.

$$
\begin{aligned}
R_{b\to c} &= \begin{pmatrix} \cos\alpha_{el} & 0 & \sin\alpha_{el} \\ 0 & 1 & 0 \\ -\sin\alpha_{el} & 0 & \cos\alpha_{el} \end{pmatrix} \begin{pmatrix} \cos\alpha_{az} & \sin\alpha_{el} & 0 \\ -\sin\alpha_{el} & \cos\alpha_{el} & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \cos\alpha_{el}\cos\alpha_{az} & \cos\alpha_{el}\sin\alpha_{az} & \sin\alpha_{el} \\ -\sin\alpha_{az} & \cos\alpha_{az} & 0 \\ -\sin\alpha_{el}\cos\alpha_{az} & -\sin\alpha_{el}\sin\alpha_{az} & \cos\alpha_{el} \end{pmatrix}.
\end{aligned}
$$

The objective is to find α_{az} and α_{el} that align the image axis, i.e., x_c with the vector p_b. In other words, we would like

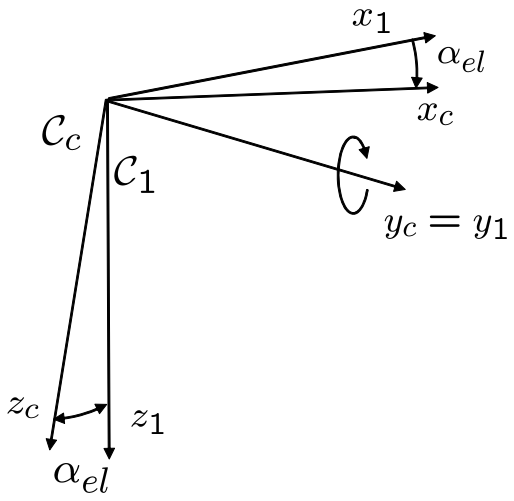

Fig. 5 The elevation angle is defined as a rotation about the *y*-axis in $\mathscr{C}_1$.

to find α_{az} and α_{el} that satisfy the equation

$$
\begin{aligned}
p_b = \begin{pmatrix} p_{bx} \\ p_{by} \\ p_{bz} \end{pmatrix} &= R_{c\to b} \begin{pmatrix} \|p_b\| \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \cos\alpha_{el}\cos\alpha_{az} & -\sin\alpha_{az} & -\sin\alpha_{el}\cos\alpha_{az} \\ \cos\alpha_{el}\sin\alpha_{az} & \cos\alpha_{az} & -\sin\alpha_{el}\sin\alpha_{az} \\ \sin\alpha_{el} & 0 & \cos\alpha_{el} \end{pmatrix} \begin{pmatrix} \|p_b\| \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \cos\alpha_{el}\cos\alpha_{az} \\ \cos\alpha_{el}\sin\alpha_{az} \\ \sin\alpha_{el} \end{pmatrix} \|p_b\| .
\end{aligned}
$$

Dividing the second element of this equation by the first and solving for α_{az} we get

$$\alpha_{az} = \tan^{-1}\left(\frac{p_{by}}{p_{bx}}\right).$$

From the last element of this equation we get

$$\alpha_{el} = \sin^{-1}\left(\frac{p_{bz}}{\|p_b\|}\right).$$

6 Image Directed Vulture Algorithm

This section describes our approach to image directed guidance algorithms that establish and maintain a prescribed orbit radius in the presence of moving obstacles, high wind conditions, and other environmental disturbances. We assume that the motion of the payload to be tracked is slower than the motion of the imager. The guidance algorithm described below uses as its input the location of the payload in the body frame of the imager, and is therefore ideally suited to the Mars lander scenario using image directed control. We also assume that the imager autopilot maintains airspeed, altitude and roll attitude hold, and that a constant airspeed, sufficient to maintain lift in the Martian atmosphere is maintained by the autopilot.

6.1 Mathematical Model of the Unmanned Autonomous Flyer

Let $\mathbf{r} = (r_x, r_y)^T$ be the inertial position of the imager and let h be the altitude. Let (ϕ, θ, ψ) be the roll, pitch, and yaw angles of the imager respectively. Also, let (u, v, w) be the velocities along the body axes, and let (p, q, r) be the associated rates. Then the kinematic equations of motion are given by[23,24]

$$\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi s\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{2}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \tag{3}$$

where the shorthand notation $c\phi \triangleq \cos(\phi)$, $s\phi \triangleq \sin(\phi)$, $t\phi \triangleq \tan(\phi)$ has been used for convenience.

The first simplification will be to express the velocity vector in the wind frame of the imager, which implies that

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} V \\ 0 \\ 0 \end{pmatrix}. \tag{4}$$

The pitch angle θ is related to the flight path angle γ as

$$\theta = \alpha + \gamma$$

where α is the angle of attack. In the wind frame, the angle of attack is zero and we have that $\theta = \gamma$. Therefore, Eq. (2) becomes

$$\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{h} \end{pmatrix} = V \begin{pmatrix} \cos\phi\cos\gamma \\ \sin\phi\cos\gamma \\ \sin\gamma \end{pmatrix}. \tag{5}$$

Equation (5) is valid if V is the ground speed of the imager. However, if V is the airspeed, then this equation must be augmented with the speed of the wind as follows:

$$\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{h} \end{pmatrix} = V \begin{pmatrix} \cos\phi\cos\gamma \\ \sin\phi\cos\gamma \\ \sin\gamma \end{pmatrix} + \begin{pmatrix} w_x \\ w_y \\ w_h \end{pmatrix}, \tag{6}$$

where w_x is the speed of the wind in the direction of the inertial x-axis (North).

At constant altitude we have that $\gamma = 0$ and h equals a constant. Therefore ignoring the altitude dynamics gives

$$\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \end{pmatrix} = V \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} + \begin{pmatrix} w_x \\ w_y \end{pmatrix}. \tag{7}$$

The following derivation draws on the discussion in [24, p. 224–226]. From Eq. (3) we get that

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r.$$

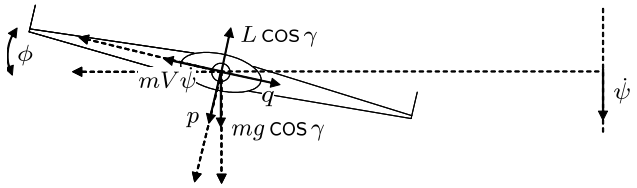

Figure 6: Free body diagram indicating forces on the imager in the $y-z$ plane. The nose of the imager is out of the page. The imager is assumed to be pitched at the flight path angle γ.

In the wind axis we have $\theta = \gamma$ so

$$\dot{\psi} = \frac{\sin\phi}{\cos\gamma} q + \frac{\cos\phi}{\cos\gamma} r. \tag{8}$$

Figure 6 shows a free body diagram of the imager indicating forces in the $x-z$ plane during a cork-screw maneuver. Writing the force equations we get

$$L\cos\phi\cos\gamma = mg\cos\gamma \tag{9}$$

$$L\sin\phi = mV\dot{\psi}. \tag{10}$$

Dividing Eq. (10) by Eq. (9) and solving for$\dot{\psi}$ gives

$$\dot{\psi} = \frac{g}{V}\tan\phi, \tag{11}$$

which is the equation for a coordinated turn.

We will assume that the imager's autopilot is tuned such that the closed-loop behavior of these loops is essentially first order. Therefore, the closed loop behavior of the autopilot is given by

$$\begin{aligned} \dot{V} &= \alpha_V (V^c - V) \\ \dot{\phi} &= \alpha_\phi (\phi^c - \phi) \end{aligned} \tag{12}$$

where α_* are positive constants, and V^c and ϕ^c are the inputs to the autopilot.

In summary, the equations of motion for the imager are given by

$$\dot{r}_x = V\cos\psi + w_x \tag{13}$$

$$\dot{r}_y = V\sin\psi + w_y \tag{14}$$

$$\dot{\psi} = \frac{g}{V}\tan\phi \tag{15}$$

$$\dot{V} = \alpha_V (V^c - V) \tag{16}$$

$$\dot{\phi} = \alpha_\phi (\phi^c - \phi). \tag{17}$$

6.2 Reachable Trajectories

The equations of motion listed in Eqs. (13)–(17) impose limitations on the feasible paths of the imager. The path tracking algorithm developed in this paper depends, at each time instant, on the reachable set T seconds into the future. The objective of this section is to mathematically define this set.

The primary constraint on the imager is the limited feasible roll angle. Assume that the roll angle is constrained as

$$-\bar{\phi} \le \phi \le \bar{\phi}.$$

From Eq. (15) we have

$$-\frac{g}{V}\tan\bar{\phi} \leq \dot{\psi} \leq \frac{g}{V}\tan\bar{\phi}.$$

Given $\dot{\psi}$, the associated turning radius is given by

$$R_t = \frac{V}{\dot{\psi}}.$$

Therefore, the minimum turning radius is given by

$$\bar{R}_t = \frac{V^2}{g\tan\bar{\phi}}.$$

Assuming a constant velocity and a constant roll angle, Eq. (15) can be integrated to obtain

$$\bar{\psi}(t,\sigma) = \psi(t) + \sigma\frac{g}{V}\tan\phi.$$

Plugging into Eqs (13)-(14) and integrating gives

$$\bar{\mathbf{r}}(t,\sigma,\phi) = \mathbf{r}(t) + \frac{V^2}{g\tan\phi}\begin{pmatrix} \sin\left(\psi(t)+\sigma\frac{g}{V}\tan\phi\right) - \sin\psi(t) \\ -\cos\left(\psi(t)+\sigma\frac{g}{V}\tan\phi\right) + \cos\psi(t) \end{pmatrix} + \sigma\begin{pmatrix} w_x(t) \\ w_y(t) \end{pmatrix}. \tag{18}$$

The reachable set T seconds in the future is given by

$$\mathscr{R}_T = \bigcup_{-\bar{\phi}\leq\phi\leq\bar{\phi}} \bar{\mathbf{r}}(t,T,\phi).$$

The reachable set for $T = 1,2,3$ seconds are shown in Figure 7 for $w_x = w_y = 0$.

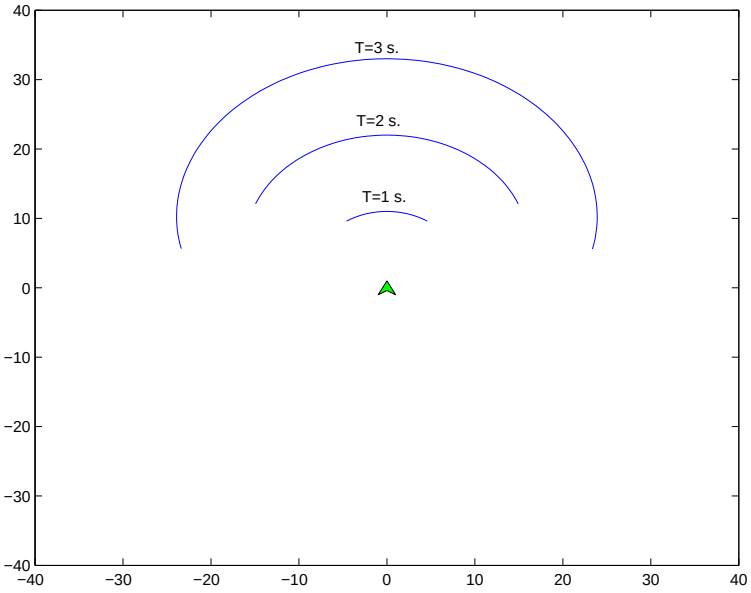

Fig. 7 The reachable set at times $T = \quad 1,2,3$ seconds.

6.3 Distance between the Reachable Set and the Desired Orbit

In this section we will derive an algorithm to find the closest point between the reachable set $\mathscr{R}_T$ and the desired orbit. Suppose that the center of the desired orbit is $\mathbf{r}_o$ and that the radius of the desired orbit is R_o. The reachable set at time T is parameterized by the roll angle ϕ. An objective function that penalizes distance from the desired orbit is given by

$$g(\phi) = \frac{1}{2}\left[\|\bar{\mathbf{r}}(t,T,\phi) - \mathbf{r}_o\|^2 - R_o^2\right]^2. \tag{19}$$

The objective is to find $\phi \in [-\bar{\phi}, \bar{\phi}]$ that minimizes $g(\phi)$.

A simple minded approach is to quantize $[-\bar{\phi}, \bar{\phi}]$ into M elements, evaluate d at each element and pick the minimum value. However if M is small, this strategy leads to quantization error in ϕ and results in chattering behavior.

Figure 8 shows the roll angle command for tracking a reference orbit and the flight path, where ϕ^c was selected by evaluating $g(\phi)$ at 30 evenly spaced angles between $[-35, 30]$ degrees. In addition, this scheme shows sensitivity to the value of α_ϕ. Figure 8, shows results for both $\alpha_\phi = 10$, a fairly quick response time for the roll attitude hold autopilot, and $\alpha_\phi = 1$, a more reasonable response time. Note the sensitivity to α_ϕ.

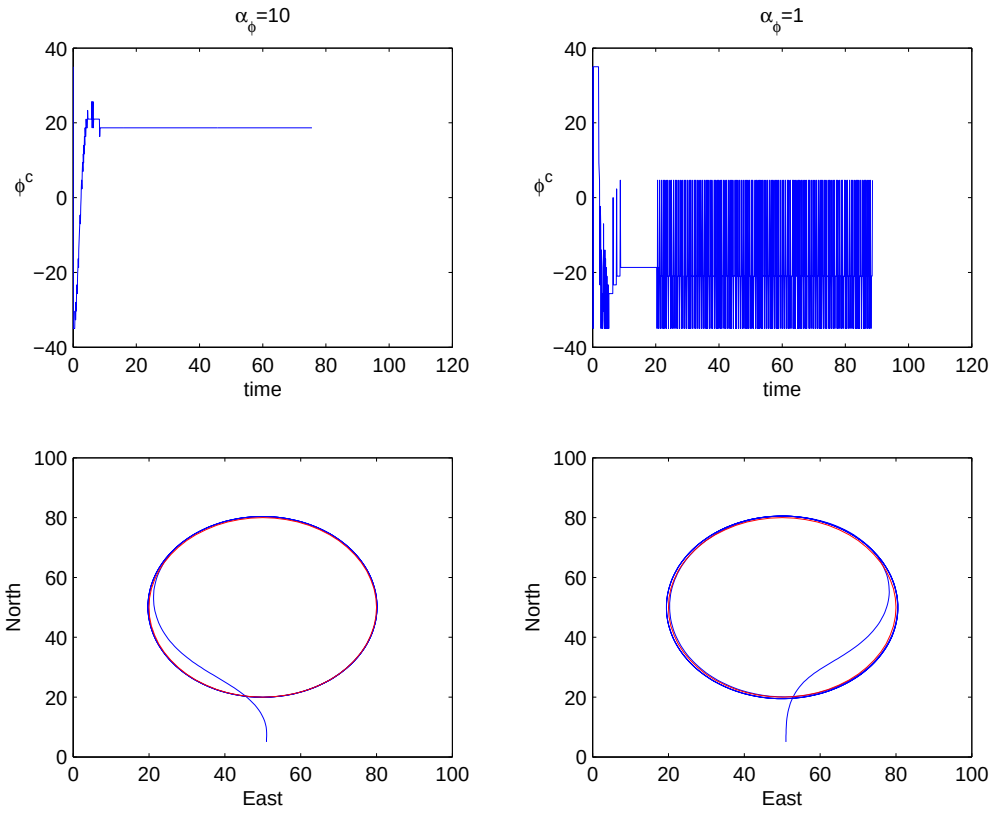

Fig. 8 The commanded roll angle exhibits chattering if the optimization step is done with a brute force technique. This figure also shows the sensitivity of the brute force method to α_ϕ. Note the poor tracking quality.

To eliminate the chatter, we need to embed some memory into the optimization process. In addition, we would like to avoid the chattering that results from quantization error. To do so, we propose introducing a nonlinear inne

loop that dynamically seeks the minimum of Eq. (19). In particular, let

$$\dot{\phi}^* = -\gamma \frac{\partial g}{\partial \phi}(\phi^*), \tag{20}$$

where γ is a positive constant that determines the convergence rate to the minimum of g. The gradient of g is given by

$$\begin{aligned}\frac{\partial g}{\partial \phi} &= \left[\|\bar{\mathbf{r}}(t,T,\phi)-\mathbf{r}_o\|^2 - R_o^2\right]\frac{\partial}{\partial \phi}\left(\|\bar{\mathbf{r}}(t,T,\phi)-\mathbf{r}_o\|^2 - R_o^2\right)\\ &= \left[\|\bar{\mathbf{r}}(t,T,\phi)-\mathbf{r}_o\|^2 - R_o^2\right](\bar{\mathbf{r}}(t,T,\phi)-\mathbf{r}_o)^T\frac{\partial}{\partial \phi}\bar{\mathbf{r}}(t,T,\phi),\end{aligned}$$

where

$$\begin{aligned}\frac{\partial}{\partial \phi}\bar{\mathbf{r}}(t,T,\phi) &= \frac{\partial}{\partial \phi}\left[\mathbf{r} + \frac{V^2}{g\tan\phi}\begin{pmatrix}\sin\left(\psi+\frac{gT}{V}\tan\phi\right)-\sin\psi\\ -\cos\left(\psi+\frac{gT}{V}\tan\phi\right)+\cos\psi\end{pmatrix}\right]\\ &= -\frac{V^2}{g}\csc^2\phi\begin{pmatrix}\sin\left(\psi+\frac{gT}{V}\tan\phi\right)-\sin\psi\\ -\cos\left(\psi+\frac{gT}{V}\tan\phi\right)+\cos\psi\end{pmatrix} + \frac{V^2}{g\tan\phi}\begin{pmatrix}\cos\left(\psi+\frac{gT}{V}\tan\phi\right)\frac{gT}{V}\sec^2\phi\\ \sin\left(\psi+\frac{gT}{V}\tan\phi\right)\frac{gT}{V}\sec^2\phi\end{pmatrix}\\ &= -\frac{V^2}{g\sin^2\phi}\begin{pmatrix}\sin\left(\psi+\frac{gT}{V}\tan\phi\right)-\sin\psi\\ -\cos\left(\psi+\frac{gT}{V}\tan\phi\right)+\cos\psi\end{pmatrix} + \frac{VT}{\sin\phi\cos\phi}\begin{pmatrix}\cos\left(\psi+\frac{gT}{V}\tan\phi\right)\\ \sin\left(\psi+\frac{gT}{V}\tan\phi\right)\end{pmatrix}\\ &= \frac{\begin{pmatrix}-V^2\sin\left(\psi+\frac{gT}{V}\tan\phi\right)\cos\phi + V^2\sin\psi\cos\phi + gVT\sin\phi\cos\left(\psi+\frac{gT}{V}\tan\phi\right)\\ V^2\cos\left(\psi+\frac{gT}{V}\tan\phi\right)\cos\phi - V^2\cos\psi\cos\phi + gVT\sin\phi\sin\left(\psi+\frac{gT}{V}\tan\phi\right)\end{pmatrix}}{g\sin^2\phi\cos\phi}.\end{aligned}$$

The last equation indicates that there will be numerical problems as $\phi \to 0$. By l'Hospitals rule, we get after some lengthy algebra that

$$\lim_{\phi\to 0}\frac{\partial \bar{\mathbf{r}}}{\partial \phi} = \frac{gT^2}{2}\begin{pmatrix}-\sin\phi\\ \cos\phi\end{pmatrix}.$$

Note also, that using l'Hospitals rule yields

$$\lim_{\phi\to 0}\bar{\mathbf{r}}(t,T,\phi) = \mathbf{r}(t) + VT\begin{pmatrix}\cos\phi\\ \sin\phi\end{pmatrix}.$$

Figure 9 shows the roll angle command for tracking a reference orbit and the flight path, where ϕ^c was selected using Eq. (20). Note the relative insensitivity to α and the quality of the tracking error as opposed to the results shown in Figure 8.

6.4 Roll Input Command.

The previous section produces a desired roll angle ϕ^*. The objective of this section is to translate that into the roll input command ϕ^c. Recall that ϕ satisfies

$$\dot{\phi} = \alpha_\phi(\phi^c - \phi),$$

where α_ϕ is the time constant of the autopilot. Defining $\tilde{\phi} \stackrel{\triangle}{=} \phi - \phi^*$ we get

$$\begin{aligned}\dot{\tilde{\phi}} &= \dot{\phi} - \dot{\phi}^*\\ &= \alpha_\phi(\phi^c - \phi) - \dot{\phi}^*.\end{aligned}$$

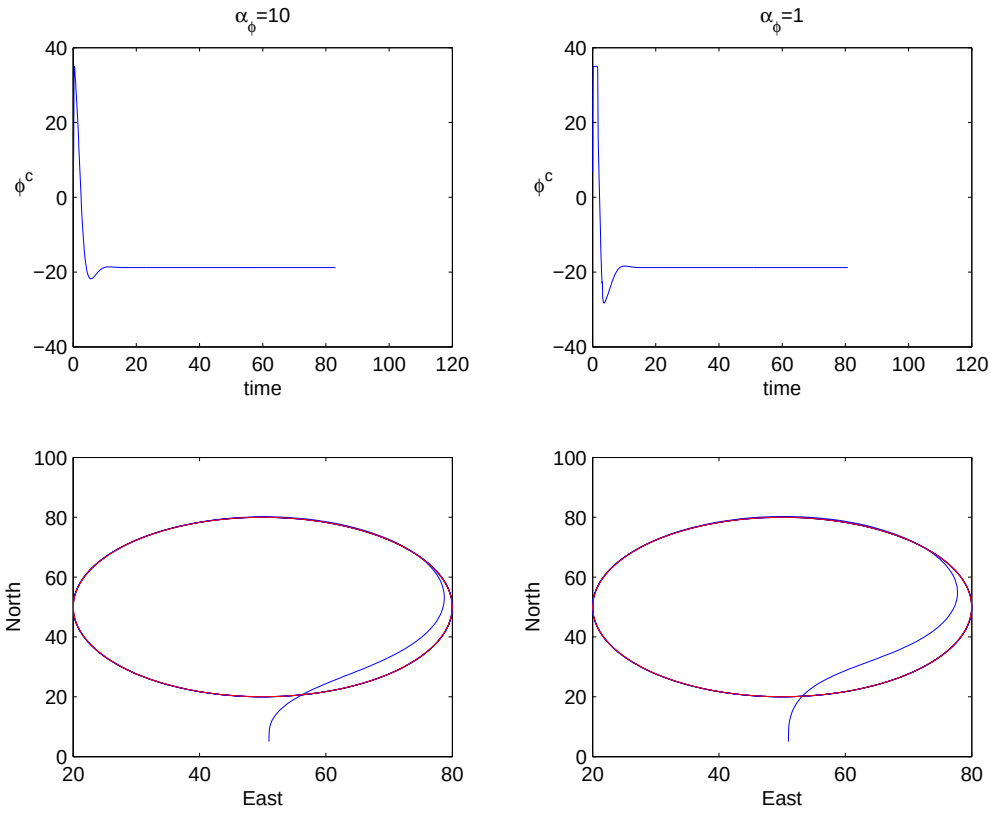

Fig. 9 The roll command and flight path using Eq. (20).

Assuming that the autopilot has already been tuned to extract the best possible time constant for the roll command, we cannot hope to do any better. Therefore we would like

$$\dot{\tilde{\phi}} = -\alpha_\phi \tilde{\phi},$$

which results in

$$\phi^c = \phi^* + \frac{\dot{\phi}^*}{\alpha_\phi}.$$

An Euler approximation of $\dot{\phi}^*$ is used to give

$$\dot{\phi}^*[n] = \frac{\phi^*[n] - \phi^*[n-1]}{T_s},$$

where T_s is the sample rate between roll commands.

6.5 Altitude Command

Given the size of the parachute in the image r_{obj}, the known size of the object R_{obj} and the focal length of the camera f, we get that the distance to the object is given by

$$h_{obj} = f\frac{R_{obj}}{r_{obj}}. \tag{21}$$

The objective of the altitude controller is to drive $\tilde{h} = h - h_{obj}$ to zero. Differentiating $\tilde{h}$ twice with respect to time gives

$$\begin{aligned}\dddot{\tilde{h}} &= \ddot{h} - \ddot{h}_{obj} \\ &= -\alpha_{\dot{h}}\dot{h} - \alpha h(h - h^c) - \ddot{h}_{obj}.\end{aligned}$$

Since the autopilot has already been tuned for maximum performance on the airframe, the best that we can hope for, is that the altitude error satisfies

$$\dddot{\tilde{h}} + \alpha_{\dot{h}}\ddot{\tilde{h}} + \alpha_h\dot{\tilde{h}} + \alpha_h\tilde{h} = 0.$$

Accordingly, we obtain

$$h^c = \frac{1}{\alpha_h}\left[\ddot{h}_{obj} + \alpha_{\dot{h}}\dot{h}_{obj} + \alpha_h h_{obj}\right]. \tag{22}$$

Note that Eq. (22) does not require knowledge of the actual altitude of the imager, but is instead dependent on the relative distance between the imager and the payload.

7 Summary of Vulture Algorithm.

We have termed the algorithm that results by combining the previous sections, the "vulture" algorithm, since it is very similar to the behavior displayed by birds of prey as they hunt.[21]

The inputs to the algorithm are: (1) the radius of the desired orbit R_o, (2) the time constant of the roll attitude hold loop α_ϕ, (3) the number of gradient descent iterations N, (4) the gradient descent parameter γ, (5) the look ahead window T, (6) the roll saturation $\bar{\phi}$, The outputs of the algorithm are: (1) the gimbal servo commands α_{az} and α_{el}, (2) the roll altitude command ϕ^c. A concise summary of the method is given in Algorithm 1. For the simulations shown in Figures 8 and 9 we used $R_o = 30$, $\alpha_\phi = 1(\text{or } 10)$, $N = 5$, $\gamma = 0.0001$, $T = 2$, and $\bar{\phi} = 35$(degrees).

Algorithm 1 Vulture Algorithm (Constant velocity and altitude)

Obtain y_{img}, z_{img}, r_{img} from the computer vision system.

Compute: $p_b \leftarrow \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \begin{pmatrix} f\frac{R_{\text{obj}}}{r_{\text{img}}} \\ y_{\text{img}} f^2 \frac{R_{\text{obj}}}{r_{\text{img}}} \\ z_{\text{img}} f^2 \frac{R_{\text{obj}}}{r_{\text{img}}} \end{pmatrix}$.

Command camera gimbal servos as:

$\alpha_{az} \leftarrow \tan^{-1}\left(\frac{p_{by}}{p_{bx}}\right)$

$\alpha_{el} \leftarrow \sin^{-1}\left(\frac{p_{bz}}{\|p_b\|}\right)$

Start gradient descent at the last roll angle: $\phi^*[n] \leftarrow \phi^*[n-1]$

for $i = 1$ to N **do** {Implement N steps of gradient descent.}

  Compute: $\xi \leftarrow \psi + \frac{Tg}{V}\tan\phi^*[n]$

  if $|\phi^*[n]| < 0.01$ **then** {Separate computation for small roll angle.}

    $(\overline{\mathbf{r}} - \mathbf{r}(t)) \leftarrow VT \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix}$

    $\frac{\partial \overline{\mathbf{r}}}{\partial \phi} \leftarrow \frac{gT^2}{2} \begin{pmatrix} -\sin\psi \\ \cos\psi \end{pmatrix}$

  else

    $(\overline{\mathbf{r}} - \mathbf{r}(t)) \leftarrow \frac{V^2}{g\tan\phi^*[n]} \begin{pmatrix} \sin\xi - \sin\psi \\ -\cos\xi + \cos\psi \end{pmatrix}$

    $\frac{\partial \overline{\mathbf{r}}}{\partial \phi} \leftarrow \frac{-V^2}{g\sin^2\phi^*[n]} \begin{pmatrix} \sin\xi - \sin\psi \\ -\cos\xi + \cos\psi \end{pmatrix} + \frac{VT}{\sin\phi^*[n]\cos\phi^*[n]} \begin{pmatrix} \cos\xi \\ \sin\xi \end{pmatrix}$

  end if

  Compute: $(\overline{\mathbf{r}} - \mathbf{r}_o) \leftarrow p_b + (\overline{\mathbf{r}} - \mathbf{r}(t))$

  Compute gradient: $\frac{\partial g}{\partial \phi} \leftarrow 2\left[\|\overline{\mathbf{r}} - \mathbf{r}_o\|^2 - R_o^2\right](\overline{\mathbf{r}} - \mathbf{r}_o)^T \frac{\partial \overline{\mathbf{r}}}{\partial \phi}$

  Move $\phi^*[n]$ along negative gradient: $\phi^*[n] \leftarrow \text{sat}_{\bar{\phi}}\left(\phi^*[n] - \gamma\frac{T_s}{N}\frac{\partial g}{\partial \phi}\right)$.

end for

Estimate: $\dot{\phi}^* \leftarrow \frac{\phi^*[n] - \phi^*[n-1]}{Ts}$

Command Roll attitude hold as: $\phi^c \leftarrow \text{sat}_{\bar{\phi}}\left(\frac{\dot{\phi}^*}{\alpha_\phi} + \phi^*\right)$

8 Simulation Results

The objective of this section is to demonstrate the effectiveness of the algorithm in strong wind conditions. The airspeed of the imager is simulated at $V = 10$ (m/s). In the simulations, we will use a constant wind blowing from West to East. The strength of the wind will be represented by W which is given in meters/second. Figures 10 and 11 show the flight path of the imager and the distance from its desired orbit when (a) $W = 2$, (b) $W = 5$, (c) $W = 7$, (d) $W = 9$. Note that Figure 11 indicates that reasonable performance is obtained even in the presence of wind speeds that are 90% of the airspeed of the imager.

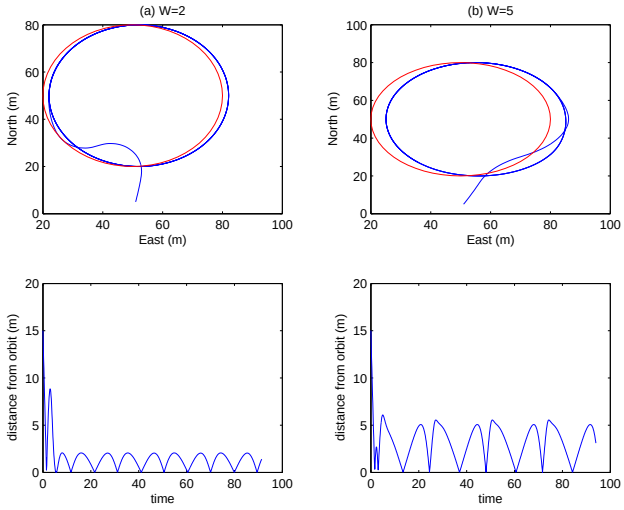

Figure 10: The vulture algorithm in wind blowing from West to East. The strength of the wind is (a) $W = 2(m/s)$, and (b) $W = 5(m/s)$. The airspeed of the imager is $V = 10(m/s)$. The bottom plots show the distance of the imager from the desired orbit.

Movie files that demonstrate the motion of the imager in simulated wind conditions can be viewed in the following avi files. Note the similarity of the motion to the flight paths of large birds in high wind conditions.

An AVI movie showing 6DOF simulation of the imager in wind that is 20% of the airspeed of the imager.

An AVI movie showing 6DOF simulation of the imager in wind that is 50% of the airspeed of the imager.

An AVI movie showing 6DOF simulation of the imager in wind that is 70% of the airspeed of the imager.

An AVI movie showing 6DOF simulation of the imager in wind that is 90% of the airspeed of the imager.

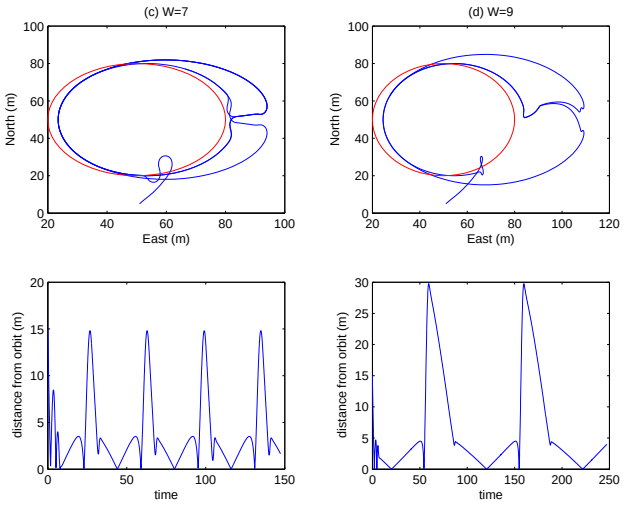

Fig. 11 The vulture algorithm in wind blowing from West to East. The strength of the wind is (c) $W = \quad 7(m/s)$, and (b) $W = 9(m/s)$. The airspeed of the imager is $V = 10(m/s)$. The bottom plots show the distance of the imager from the orbit.

9 Experimental Setup and Results

9.1 Mini-UAV Airframe

Figure 12 shows the airframe used to demonstrate the image directed control concept. The airframes are modified versions of the standard Zagi THL flying wing,[25] and are therefore inexpensive and versatile. While this airframe would not be appropriate for an actual Mars mission, it provides a low-cost, low-risk research platform that can be used to demonstrate image directed control technologies that will be critical for a Mars mission. The Zagi platform has a wingspan of 48 inches, a wing area of 2.83 square feet, an unloaded weight of 11.5 ounces, and a wing loading factor of 4 ounces per square foot.

Fig. 12 Mini-UAVs used for the Mars lander project.

The airframe has been modified in several ways. (1) Both the imager and the payload flyers were retrofitted with autopilots and 900 MHz wireless modems. (2) The payload flyer was outfitted with a parachute and release mechanism (described below). (3) The imaging flyer was outfitted with a video camera mounted on a pan-tilt gimbal and a video transmitter (described below). The airframe is actuated with Hitec HS-55 servos which are inexpensive, light weight, robust and responsive. It is powered by a Hacker b12-15l electric brushless motor, with 4:1 gear reduction. Batteries were selected to achieve a 30 minute flight test. The maximum level flight velocity is approximately 45 mph, with a cruise velocity of approximately 30 mph.

9.2 BYU Kestrel Autopilot

Figure 14 shows a picture of the BYU Kestrel autopilot which was used to obtain the experimental results. The autopilot contains a 30 MHz Rabbit micro-controller, 3-axis rate gyros, 3-axis accelerometers, a differential pressure sensor to measure airspeed, an absolute pressure sensor to measure altitude, an interface to a GPS antenna, and a suite of digital and analog I/O channels. The autopilot implements altitude hold, velocity hold, and roll attitude hold[26,27] as needed to implement Algorithm 1. The autopilot interfaces to ground station software that is capable of commanding

multiple UAVs simultaneously. The ground station software is used to command both the imager as well as the payload flyer.

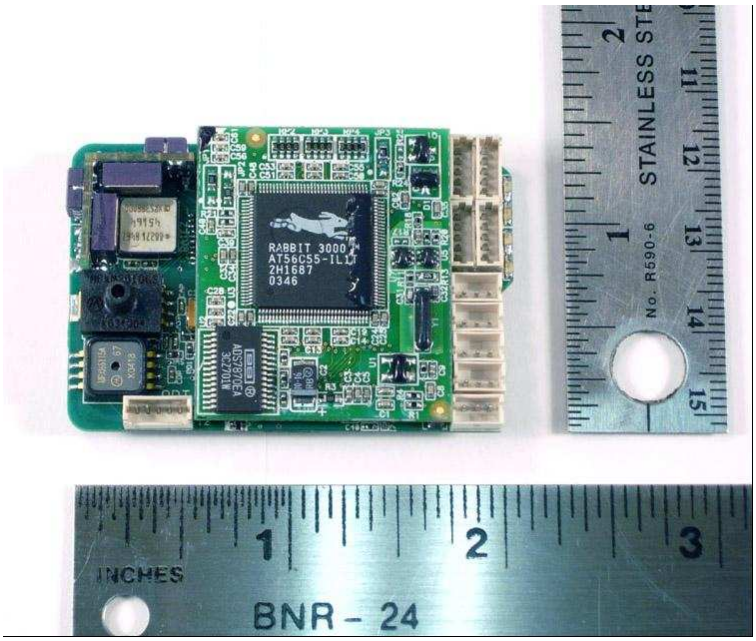

Fig. 13 The Kestrel autopilot developed by BYU.

The BYU Kestrel Autopilot weighs 16.5 grams and contains 3-axis rate gyros, 3-axis accelerometers, absolute and differential pressure sensors, and interfaces to a GPS receiver.

9.3 Pan-Tilt Gimbal

Figure 15 shows the pan-tilt gimbal and the camera system used for the demonstration. The camera is positioned by the use of two servos: one to control pan and one to control tilt. The tilt (elevation) servo directs the camera at angles between the forward horizon and rear horizon. Gearing on the pan (azimuth) servo allows the camera to pivot a full 360 degrees. These capabilities enable the surrounding scene to be captured through two independent configurations. For example, the rear horizon can be captured by tilting to the rear, or alternately by panning 180 degrees to capture images to the rear. The device is designed so that the entire assembly is self-contained. This configuration allows only minimal impact on the design of the actual aircraft structure. The only allowance needed within the aircraft itself is that wiring be available directly beneath the mounting location. Furthermore, the components of the fixture are lightweight and durable (high density plastic and aluminum).

9.4 Parachute and Payload Flyer

As shown in Figure 15, the payload flyer has been equipped with a parachute deployment pod mounted to the top wing surface of the airframe. When commanded by the ground station, the spring loaded deployment pod is activated, releasing the parachute canopy, which pulls the parachute into the air stream. The deployed parachute is shown in Figure 16. The 36 inch parachute slows the payload to a descent of approximately 0.78 m/s. From an altitude of

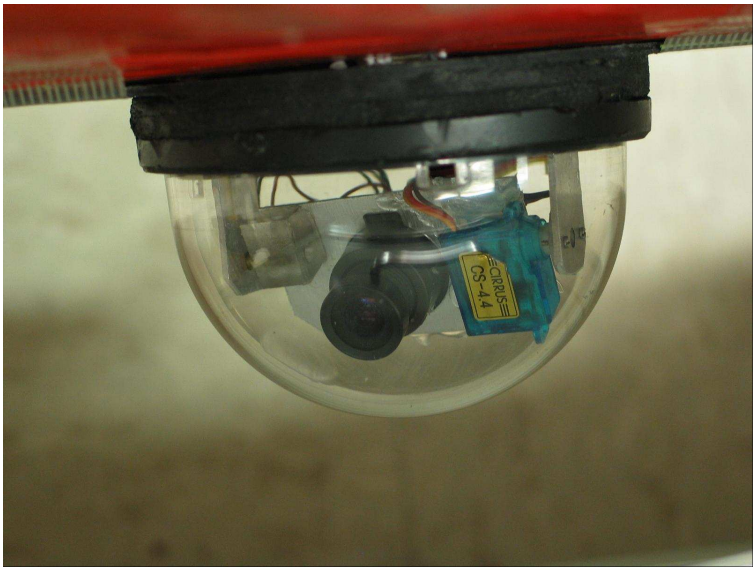

Fig. 14 The pan-tilt gimbal used for the project. The gimbal is actuated by two RC-servos that provide 360 degrees of pan and 180 degrees of tilt.

500 meters it will take the payload about 6.5 minutes to reach the ground. Figure 17 shows the payload in descent mode.

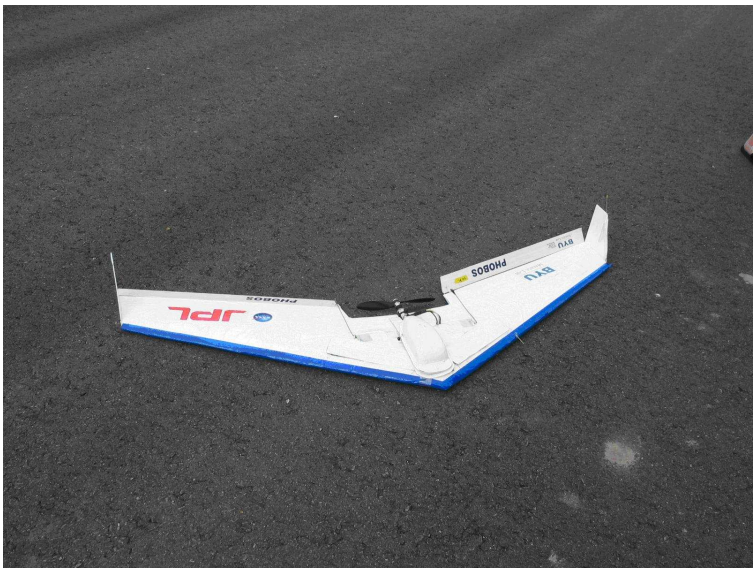

Fig. 15 The payload flyer is an autonomous UAV with a parachute pod mounted on the top of the airfoil.

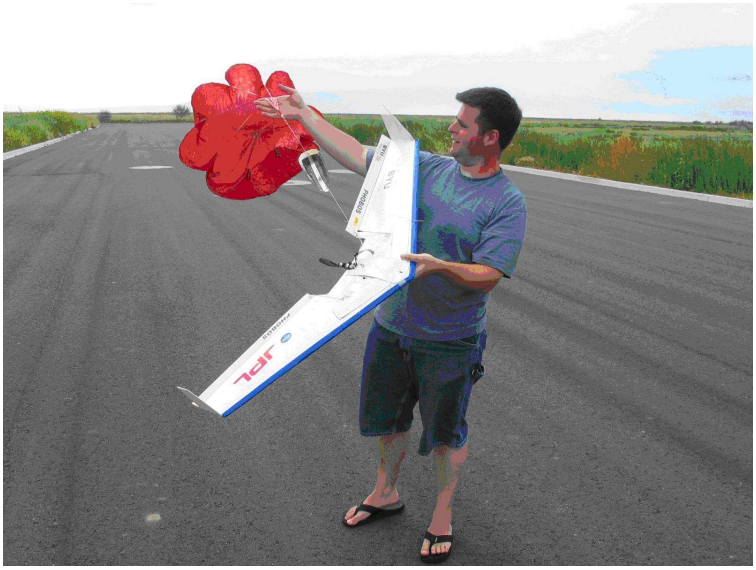

Fig. 16 This figure demonstrates the deployment of the parachute. The pod that holds the parachute remains attached to the UAV.

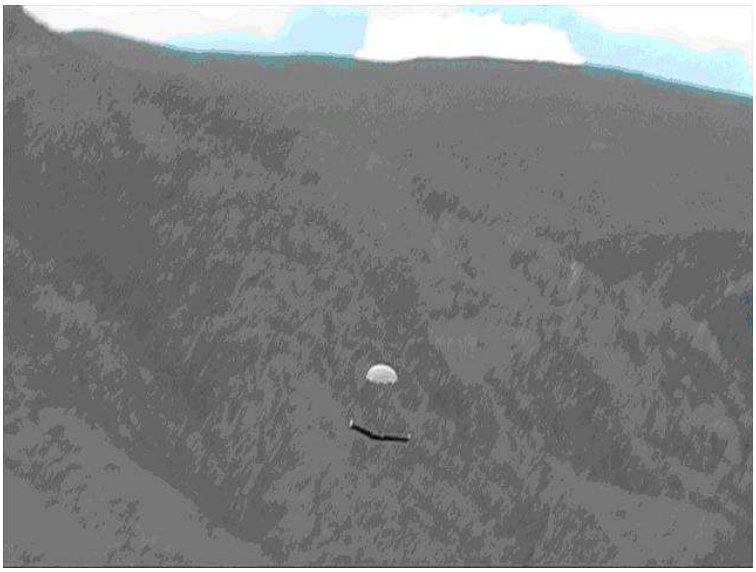

Fig. 17 The payload flyer is shown in descent mode after the parachute has been deployed. When the parachute is deployed, the motor is disengaged.

9.5 Framegrabber and Computer Vision

Figure 18 shows our software architecture for image directed control. At this stage, the computer vision algorithms are implemented at the ground station. For an actual Mars mission, this functionality will need to be moved on-board the flyer. As shown in Figure 18, an NTSC analog video signal is transmitted to the ground station via a 2.4 GHz wireless transmitter. A PCMCIA video capture card plugged into a laptop captures the images at 30 frames per second. We are using an ImperX VCE-P 0 video capture card. The images are processed in real time and the pixel location and size of the parachute are passed to the guidance algorithm which send α_{az}, α_{el}, h^c, V^c, and ϕ^c to the autopilot via a 900 MHz transceiver. The sample rate of the guidance loop is approximately 20 Hz with a delay of approximately 0.5 seconds. We should point out that the sample rate of the autopilot is approximately 120 Hz with insignificant delay.

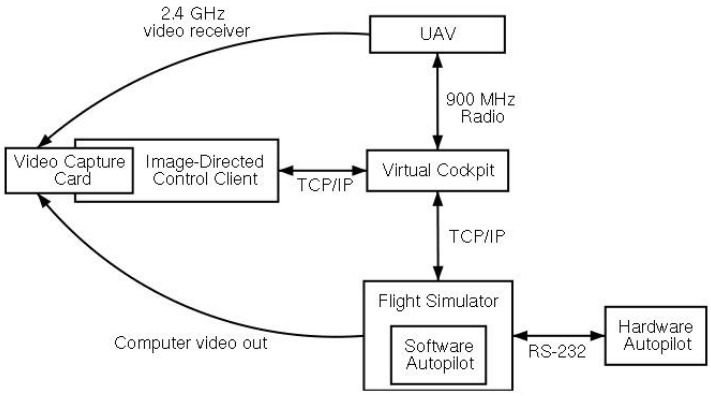

Fig. 18 Software architecture for image directed control. An analog NTSC video signal is transmitted to the ground via a 2.4 GHz transmitter. The analog signal is captured by the video capture card located at the ground station. Image data is processed and autopilot commands are transmitted to the imager.

9.6 Flight Tests and Results

A simplified version of the vulture algorithm was used in our flight tests. The simplified vulture algorithm is based on the limit cycle behavior of a supercritical Hopf bifurcation. The equations for a Hopf bifurcation are given by

$$\dot{r}_x = r_y + r_x\left(\mu - r_x^2 - r_y^2\right)$$
$$\dot{r}_y = -r_x + r_y\left(\mu - r_x^2 - r_y^2\right).$$

A Hopf bifurcation is considered sub-critical when $\mu < 0$, with a single equilibrium at the origin. When $\mu = 0$, it forms a slow decay to a spiral point at the origin, and when $\mu > 0$, the equation is considered supercritical, and the

trajectory is a limit cycle centered about the origin. The desired heading of the imager is generated as follows:

$$\dot{r}_x = \tilde{r}_y + \frac{\tilde{r}_x}{\alpha r^2}\left(r^2 - \tilde{r}_x^2 - \tilde{r}_y^2\right)$$
$$\dot{r}_y = -\tilde{r}_x + \frac{\tilde{r}_y}{\alpha r^2}\left(r^2 - \tilde{r}_x^2 - \tilde{r}_y^2\right)$$
$$\psi^c = \tan^{-1}\left(\frac{\dot{r}_y}{\dot{r}_x}\right),$$

where $\tilde{r} = (\tilde{r}_x, \tilde{r}_y)^T$ is the estimated position vector from the desired orbit location, and r is the desired orbit radius. Figure 19 shows telemetry data from a flight test where the wind was 63% of the airspeed of the flyer, demonstrating the robustness of the algorithm to wind.

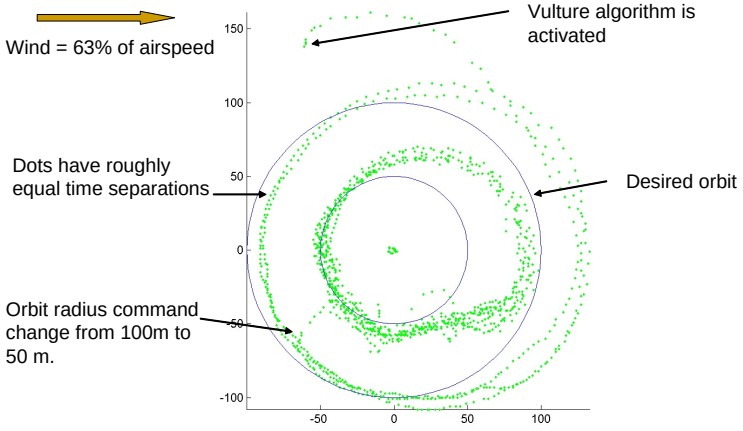

Fig. 19 Telemetry data from a flight test where wind speed is 63% of the flyers airspeed. The flyer is initially commanded to maintain a 100 m orbit. At the indicated location, it is commanded to maintain a 50 m orbit.

A mock demonstration described in Section 2 was flight tested using GPS signals on both the payload and imager. The following WMV and AVI movies demonstrate the effectiveness of our approach. The flyers in these movies operate autonomously without human interaction.

This WMV movie shows the parachute being launched from the payload flyer.

This WMV movie shows the video from the gimbaled camera on the imager as it tracks the parachute launch of the payload. The separation between imager and payload in this movie is approximately 100 meters.

This AVI movie show a Matlab™ visualization of the telemetry data. The imager is initially launched and commanded to orbit the payload flyer while it is still on the ground. The payload flyer is subsequently launched and commanded to fly to a waypoint with an altitude of 500 meters. When the parachute is launched, the payload flyer free falls to the ground and the imager orbits the payload keeping it in the camera field of view.

10 Conclusion

In this paper we have defined a Mars lander scenario where a flyer is used to track and image the descent of the main payload to Mars. Color segmentation algorithms have been developed to track the payload in the camera

image. A novel trajectory generation algorithm has been developed to track the payload flyer based on biological considerations. The salient feature of this algorithm is its robustness to strong wind conditions. An earth based mock scenario was developed and flight tested using miniature UAVs. The flight demonstrations show the effectiveness of the approach.

Acknowledgments

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology and Brigham Young University under a contract with the National Aeronautics and Space Administration (NASA) and was sponsored by the NASA Intelligent Systems Program. We would like to acknowledge the assistance of Steve Griffiths and Joe Jackson who developed the gimbal and flyers, David Hubbard who implemented the color segmentation algorithm, Derek Nelson who implemented the Vulture algorithm, and Andrew Eldridge and Blake Barber who supported the flight demonstrations. We would like to acknowledge Dr. Butler Hine, NASA CICT Program Manager for valuable suggestions on this work. The first author was supported by AFOSR grants F49620-01-1-0091 and F49620-02-C-0094.

References

[1]Thakoor, S., "Bio-Inspired Engineering of Exploration Systems," *Journal of Space Mission Architecture*, No. 2, 2000, pp. 49-79.

[2]Chahl, J., Thakoor, S., Le Bouffant, N., Strange, G., and Srinivasan, M. V., "Bioinspired Engineering of Exploration Systems: A Horizontal Sensor/Attitude Reference System Based on the Dragonfly Ocelli for Mars Exploration Applications," *Journal of Robotic Systems*, Vol. 20, No. 1, 2003, pp. 35-42.

[3]Thakoor, S., Cabrol, N., Soccol, D., Chahl, J., Lay, N., Hine, B., and Zornetzer, S., "Benefits of Bioinspired Flight," *Journal of Robotic Systems*, Vol. 20, No. 12, Dec. 2003, pp. 687-706.

[4]Thakoor, S., Chahl, J., Hine, B., and Zornetzer, S., "Biologically-Inspired Navigation and Flight Control for Mars Flyer Missions," *Proceedings of Space Mission Challenges for Information Technology*, July 2003.

[5]Thakoor, S., Chahl, J., Srinivasan, M., Werblin, F., Young, L., Hine, B., and Zornetzer, S., "Bioinspired Engineering of Exploration Systems for NASA and DoD," *Artificial Life Journal*, Vol. 8, No. 4, 2002, pp. 357-369.

[6]Thakoor, S., Chahl, J., Soccol, D., Hine, B., and Zornetzer, S., "Bioinspired Enabling Technologies And New Architectures For Unmanned Flyers," *Proceedings of the AIAA 2nd Unmanned Unlimited Systems, Technologies and Operations -Aerospace, Land, and Sea Conference*, AIAA Paper 2003-6580, Sept. 2003,

[7]Thakoor, S., Morookian, J. M., Chahl, J., Hine, B., and Zornetzer, S., "BEES: Exploring Mars with Bioinspired Technologies," *IEEE Computer Magazine*, Sept. 2004, pp. 38–47.

[8]Thakoor, S., Hine, B., and Zornetzer, S., "Bioinspired Engineering of Exploration Systems (BEES) Its Impact on Future Missions," *Proceedings of the 1st AIAA Intelligent Systems Conference*, Sept. 2004.

[9]Mars Pathfinder Mission landing profile, available online at http://marsrovers.jpl.nasa.gov/home/ (cited Dec. 2004).

[10]Kandel, E. R., Schwartz, J. H., and Jessell, T. M., *Principles of Neural Science*, Prentice Hall International, London, 1991.

[11]Ohtani, M., Asai, T., Yonezu, H., and Ohshima, N., "Analog Velocity Sensing Circuits Based on Bioinspired Correlation Neural Networks," *Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*, April 1999, pp. 366–373.

[12]Ohtani, M., Asai, T., Yonezu, H., and Ohshima, N., "Analog MOS Circuit Systems Performing the Tracking with Bio-inspired Simple Network," *Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*, April 1999, pp. 240–246.

[13]Franz, M. and Mallot, H., "Biomimetic Robot Navigation," *Robotics and Autonomous Systems*, Vol. 30, 2000, pp. 133-153.

[14]Viitala, J., Korpimaki, E., Palokangas, P., and Koivula, M., "Attraction of Kestrels to Vole Scent Marks Visible in Ultraviolet Light," *Nature*, Vol. 373, Feb. 1995, pp. 425-427.

[15]Honkavaara, J., Koivula, M., Korpimki, E., Siitari, H., and Viitala, J., "Ultraviolet Vision and Foraging in Terrestrial Vertebrates," *Oikos*, Vol. 98, 2002, pp. 504-510.

[16]Moller, R., "Insects Could Exploit UVGreen Contrast for Landmark Navigation," *Journal of Theoretical Biology*, Vol. 214, 2002, pp. 619–631.

[17]Santos-Victor, J., Sandini, G., Curotto, F., and Garibaldi, S., "Divergent Stereo for Robot Navigation: Learning From Bees," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1993, pp. 434–439.

[18]Santos-Victor, J., Sandini, G., Curotto, F., and Garibaldi, S., "Divergent Stereo in Autonomous Navigation: From Bees to Robots," *International Journal of Computer Vision*, Vol. 14, March 1995, pp. 159–177.

[19]Green, P. R., "Head Orientation and Trajectory of Locomotion During Jumping and Walking in Domestic Chicks," *Brain, Behavior and Evolution*, Vol. 51, 1998, pp. 48–58.

[20]Green, P. R., "Head Orientation is Aligned with Takeoff Trajectory as Chicks Jump," *Experimental Brain Research*, Vol. 122, 1998, pp. 295–300.

[21]Tucker, V. A., "The Deep Fovea, Sideways Vision and Spiral Flight Paths in Raptors," *The Journal of Experimental Biology*, Vol. 203, 2000, pp. 3745–3754.

[22]Ma, Y., Soatto, S., Kosecka, J., and Sastry, S., *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer-Verlag, 2003.

[23]Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, John Wiley & Sons, 1965.

[24]Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls, Parts I & II*, DARCorporation, Lawrence, KS, 1998.

[25]Airframe details of standard zagi kit, available online at http://www.zagi.com (cited Dec. 2004).

[26]Beard, R., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., and Goodrich, M., "Autonomous Vehicle Technologies for Small Fixed Wing UAVs," *Journal of Aerospace, Computing, Information, and Communication*, 2004, (to appear).

[27]Kingston, D., Beard, R., McLain, T., Larsen, M., and Ren, W., "Autonomous Vehicle Technologies for Small Fixed Wing UAVs," *AIAA 2nd Unmanned Unlimited Systems, Technologies, and Operations–Aerospace, Land, and Sea Conference and Workshop & Exhibit*, AIAA Paper 2003-6559, Sept. 2003.