Target Acquisition, Localization, and Surveillance using a Fixed-Wing Mini-UAV and Gimbaled Camera

Morgan Quigley
Michael A. Goodrich
Computer Science
Brigham Young University
Provo, UT 84602, USA
mquigley@byu.edu
mike@cs.byu.edu

Stephen Griffiths
Andrew Eldredge
Mechanical Engineering
Brigham Young University
Provo, UT 84602, USA
stepheng@email.byu.edu
andrewe@byu.edu

Randal W. Beard
Electrical and Computer Engineering
Brigham Young University
Provo, UT 84602, USA
beard@ee.byu.edu

***Abstract*— Target acquisition and continuous surveillance using fixed-wing UAVs is a difficult task due to the many degrees of freedom inherent in aircraft and gimbaled cameras. Mini-UAVs further complicate the problem by introducing severe restrictions on the size and weight of electro-optical sensor assemblies. We present a field-tested mini-UAV gimbal mechanism and flightpath generation algorithm as well as a novel human-UAV interaction scheme in which the operator manually flies the UAV to produce an estimation of the target position, then allows the aircraft to fly itself and control the gimbal while the operator refines and moves the target position as required.**

***Index Terms*— UAV, gimbaled camera, flightpath generation, human-UAV interaction**

I. Introduction

Fixed-wing mini-UAVs are an emerging class of aerial vehicles that posess unique strengths and limitations. The portability of these aircraft and their ability to be hand-launched are what define the vehicle class and make them very attractive to a wide range of military and civillian applications, but these same factors place serious restrictions on the size, weight, and electrical power of their payloads. However, even with these constraints, mini-UAVs offer great potential for highly mobile local surveillance. Previous work details our efforts in creating a stable mini-UAV testbed that combines small and highly integrated autopilot assemblies with durable airframes [1]. In this paper, we discuss issues relating to the actual use of these airframes for target acquisition, localization and continous surveillance. Although military applications of these concepts are readily apparant, many civilian applications exist as well, notably wilderness search-and-rescue.

The continual improvement and miniaturization of electro-optical sensors has produced off-the-shelf video cameras that are have volumes well under eight cubic centimeters, weigh less than an ounce, and cost under one hundred dollars. Small audio/video transmitters are similarly inexpensive and capable, operating on the 2.4 GHz band and producing ranges over 10km. With so much hardware available, a major question is how to integrate these system and create a stable and efficient mini-UAV platform for real-world target acquisition and localization tasks.

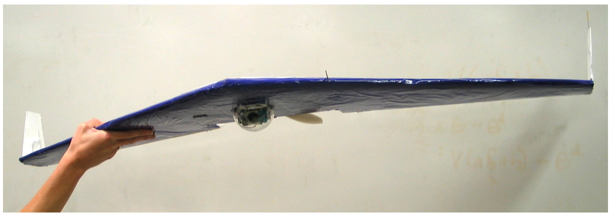

Fig. 1. Direct-drive gimbal mounted on a 40” flying-wing mini-UAV.

Rigidly attaching a miniature camera to a UAV airframe can produce imaging of target points, especially when the UAV is in the hands of a highly skilled operator. However, an unavoidable problem with this configuration is that the UAV must be facing the target to image it. Because fixed-wing UAVs must maintain airspeed, the UAV will soon fly over the target and require significant time to itself for another pass over the target; it is thus impossible to maintain continuous surveillance of a target using a forward-facing camera. Although this configuration is light, streamlined, and durable, it severely limits the imaging that the mini-UAV can produce.

As demonstrated by large operational UAV platforms, gimbaled camera mounts can dramatically improve the quality and quantity of imaging returned from the UAV. The flightpath trajectory need not be a direct flyover of the target, as required by a forward-facing camera. Large UAV platforms typically separate the role of sensor operator from that of pilot, because the gimbal angles and aircraft attitude are independent and the relationship between them is continually adjustable. Using two human operators to perform the roles of aircraft pilot and sensor operator is logical when missions are very long and complex, such as those typically performed by USAF Predator aircrews [3]. However, many current scenarios for mini-UAV deployment are short-range missions with a relatively low number of goals, such as battle damage assessment or searching.

Because the UAV is tasked with a relatively straightforward objective that is directly related to its imaging, we propose an alternative aircrew division: one operator controls the gimbal and aircraft using high-level autonomy while solely focusing on accomplishing the mission, while the other operator serves as the “flight engineer,” monitoring critical systems such as battery (or fuel) levels, range, altitude, and the like. The UAV must then be equipped with enough autonomy that the traditional piloting role is all but eliminated, except in takeoff and landing, which are unique flight segments that can either be automated or performed manually by either aircrew member. The benefits of this aircrew configuration include the following:

- The high-level operator is able to devote full attention to the video stream and readily evaluate and react to its information (i.e. track moving targets, etc.). Aircraft automation maintains constant-radius orbits around the target point and controls the gimbal angles.
- The flight engineer devotes full attention to critical aircraft telemetry, such as battery/fuel levels, airspeed, and altitude. Because the flight engineer is solely responsible for keeping the aircraft aloft, he or she is able to perform this task efficiently and precisely.
- Higher human:UAV ratios can be obtained simply by adding more high-level operators. The flight engineer will likely be able to monitor the critical systems of two or more aircraft simultaneously. Scaling up the number of UAVs allows the human:UAV ratio to be 3:2, 4:3, 5:4, etc., gradually approaching a multiple of the 1:1 ratio. (Although there is sure to be a limit on the number of UAVs that one flight engineer can efficiently support.)

The separation of the roles of high-level operator and flight engineer reflects the premise that obtaining useful intelligence and keeping a UAV in the air are two fundamentally different tasks. Our model includes the assumption that the aircraft autonomy, as produced by the onboard autopilot, comprises an implicit third member of the aircrew, performing much of the role of the traditional pilot.

In this paper, we use the previous concepts as a foundation for the implementation details of a complete mini-UAV system which is capable of performing target acquisition, localization, and surveillance. The next section discusses the issues pertaining to camera gimbal assemblies for mini-UAVs.

II. Mini-UAV Gimbal Design

We have constructed many prototypes of various gimbal configurations and flown them extensively on our mini-UAV platforms. These have included the traditional altitude-azimuth hemispherical assembly as well as several novel retractable designs. A summary of the lessons we have learned from the design process is given at the end of this section.

A. General Considerations

Our in-house autopilot design provides four channels of PWM output which are reserved for aileron, elevator, rudder, and throttle controls. We use an external servo controller connected over an I2C bus to generate the control signals for the gimbal. To promote modularity, increase reliability, and reduce size and weight, our current designs employ off-the-shelf micro servo assemblies to provide position control.

Locating the camera assembly in a hemispherical dome on the underside of the aircraft has proven successful on a variety of aircraft, including conventional airplanes, helicopters, and even airships. The equations of motion of the two-axis gimbal system are detailed in previous publications and are well understood [2]. However, unlike large and high-powered aircraft, the small size of mini-UAVs makes them highly sensitive to drag induced by external assemblies. In particular, takeoffs and landings become much more difficult as the gimbal-induced disruptions to airflow dramatically increase the stall speed of the aircraft. For a time, we experimented with a bungee-catapult launch scheme to significantly increase the takeoff speed of the aircraft over that possible with hand launches. Although this method definitely produced a higher takeoff airspeed, it was far from elegant and ignored the crux of the matter: the small size and low Reynolds numbers of mini-UAVs greatly amplify the problems produced by drag on the gimbal assembly, so airflow distruptions induced by gimbal assemblies must be as small as possible.

Our initial gimbal prototype introduced so much drag on the airframe that it was necessary to continually run the motor at maximum current, which, on occasion, would overheat to the point of unsoldering its power connectors and/or melting its stator coil. This example is somewhat extreme, but serves to illustrate the dangers of adding sizable components to a mini-UAV. We have made progress in overcoming this problem by embedding as much as possible of the gimbal assembly into the airframe, with only the clear hemispherical dome projecting into the airstream, as shown in Figure 1.

After observing the severe performance losses produced by our initial gimbal design, we sought to overcome the high stall speed of gimbal-equipped mini-UAV airframes by constructing a retractable-arm gimbal mount. In this design, the camera is unprotected from the wind and is directly attached to its elevation servo, which in turn is directly attached to the horn of the azimuth servo. The entire assembly can then be “stowed” flush on top of the airframe or “deployed” several centimeters in front of the nose. This “stowing” motion was provided by an additional servo which pivoted a small arm upon which the gimbal assembly was mounted. When stowed, this assembly is very compact when compared to our hemispherical domes, and we hypothesized that it would result in superior low-speed performance of the airframe. Unfortunately, the assembly proved insufficiently streamlined to increase the performance of the airframe, and in field testing several prototypes were severely damaged in inverted landings resulting from stalls during low-speed landing approaches. Although our initial efforts were unsuccessful, the idea of eliminating the drag induced by a hemispherical dome is very appealing, and we are currently designing an airframe

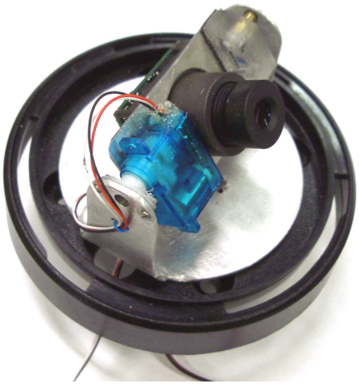

Fig. 2. Direct-drive gimbal

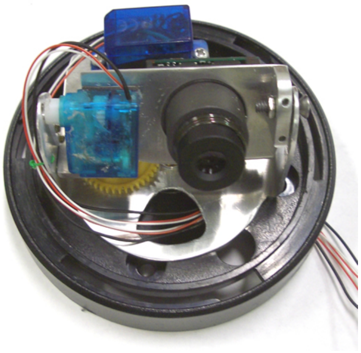

Fig. 3. Accelerated-Gearing Gimbal

in which the gimbal will "deploy" downward similarly to the front landing gear on a conventional aircraft, and "stow" in a similar fashion fully inside the fuselage. We anticipate that this combined airframe-gimbal design will dramatically increase low-speed flight performance while also increasing the survivability of the gimbal assembly.

Detailed descriptions of our current hemispherical-dome mini-UAV gimbal designs are given in the following sections.

B. *Direct-Drive Hemispherical Dome Assemblies*

A simple and light gimbal assembly can be constructed by connecting one servo directly to the gimbal armature. This allows for azimuth control, and another servo can function as the pivot for the elevation arm to provide elevation control. Using a programmable servo controller allows the servos to be driven up to their mechanical limits, but unfortuantely most micro servos possess no more than 135 degrees of electronically stabilized travel. If the servo is directly connected to the gimbal armature without intervening gears, this results in severe restrictions on the azimuth slew range.

However, we have obtained very favorable results with this configuration by introducing automatic flightpath generation to keep the target bearing within the constraints of the gimbal's azimuth servo, a process that will be discussed in a following section. It is important to note that this "direct-drive" configuration, however limited it may seem, is extremely durable. Our direct-drive prototypes have withstood numerous hard crashes, which are virtually inevitable in mini-UAV research and development. Even with a strategically located landing skid, gimbals placed on the underside of our "flying-wing" mini-UAVs experience very significant mechanical forces during rough landings and crashes; actual real-world deployment will likely involve similar mishaps, and gimbal assemblies on mini-UAVs must be able to withstand moderate impacts with minimal damage.

C. *Accelerated-Gearing Hemispherical Dome Assemblies*

To obtain full range of motion in the azimuth plane, we have constructed several prototype gimbals which use intermediate gears to amplify the motion of a standard micro servo, expanding the azimuth slew range to 360 degrees. This results in the ability to observe and track targets regardless of the orientation of the UAV, although azimuth resolution suffers somewhat. Because the servo controller in our current design accepts 8-bit position codes, there are only 2^8 possible positions. When the azimuth travel is expanded to 360 degrees through gearing, this results in azimuth resolution of approximately 1.4 degrees. Although this is a very coarse discretization, it is important to remember that a mini-UAV flying at 100 meters can obtain satisfactory imaging with relatively wide-angle lenses, and a 40-degree field of view can provide a surprisingly good quality image. A difficulty with this type of assembly, however, is creating a mechanism which can withstand significant mechanical shock while not exhibiting significant free-play in its normal operation. Our future work looks toward replacing the gears with belts in the hope of reducing free-play while maintaining mechanical integrity in the event of hard landings.

D. *Implementation Details*

Tuning the gimbal is a major problem when using off-the-shelf micro servos. Although each servo will likely be similar to others of its class, small discrepencies exist which require fine tuning when used in a camera-pointing application. A simple constrained linear control equation suffices:

$$o = min\left(max\left(i \cdot s + b, o_{min}\right), o_{max}\right)$$

where o equals the output value, i equals the input value, o_{min} and o_{max} are limits on the travel to prevent hardware damage, and s and b are empirically determined scale and bias values, respectively. Arriving at proper values for s and b is currently a process of trial-and-error, and these values (particularly the bias) will likely deviate somewhat from servo to servo.

E. *Mini-UAV Gimbal Design Summary*

Our efforts at constructing a variety of gimbals for mini-UAVs using micro servo assemblies can be summarized in the following "lessons" that we have learned:

- The most critical factor to the usefulness and survivability of a gimbal design for a mini-UAV is the aerodynamic drag it induces on the airframe.
- Hemispherical-dome designs, if carefully constructed, can be successfully and reliably flown on mini-UAVs.
- Retractable-arm designs offer the potential of improving the low-speed aerodynamics of the airframe, but they require a purpose-built airframe to ensure complete protection for the camera assembly.
- Full azimuth travel is gained at the expense of additional complexity and weight, and lowers angular resolution.

We have currently settled on the "direct-drive" gimbal assembly as our current operational configuration, as it provides adequate flight performance and excellent durability due to its very simple construction. We hasten to add, however, that to obtain satisfactory imaging from this gimbal design, it is necessary to place the target within the limited azimuth range of the gimbal assembly. This task can be performed through automatic flightpath generation, which is discussed in the following section.

III. Flightpath Generation

Fully automatic flight control is extremely useful in reducing operator workload when the GPS point of the target is known. Continuous surveillance of a known target point is possible by flying constant-radius orbits at a constant velocity and altitude while aiming the camera at the center of the orbit, which corresponds to ninety degrees away from the direction of travel. To create an acceptable level of performance using this style of autonomous flight, it is necessary to generate the heading parameter in real time onboard the aircraft. Although the problem of orbit generation has many valid solutions, real-world implementation on mini-UAV systems are subject to several restrictions which may be more critical than the corresponding issues on large UAV platforms:

- Very limited available processing power.
- Sizable flightpath anomolies due to wind turbulence or transient state estimation errors.
- Low cruise speed of mini-UAVs means that wind speeds may often be $> 50\%$ or more of airspeed.

Mini-UAV systems with a single onboard processor must operate under a very high processor load. Many off-the-shelf embedded processors are cooperative multitasking environments, and computation-intensive orbit generation routines have the potential of reducing the amount of processor time available for state estimation. Future embedded systems will certainly offer higher computational performance, but the quandry of processor time allocation will likely remain; if the processor cannot be scheduled to handle all accelerometer and rate gyroscope updates, the quality of state estimation is reduced. Because path planning is only one of many background tasks the autopilot processor must perform, reducing path computation cost is of paramount importance.

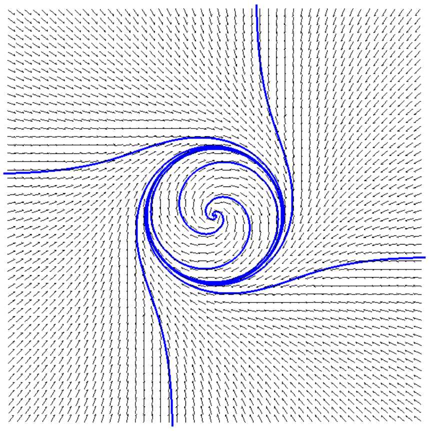

Fig. 4. Trajectories through the slope field of a supercritical Hopf bifurcation always converge to the limit cycle.

To satisfy this condition, we have adapted a relatively simple system of equations: the Hopf bifurcation. Supercritical Hopf bifurcations are well understood mathematically, and produce spiral trajectories which converge to a limit cycle attractor [4]. When inside of the limit cycle, trajectories spiral outward from the center point and converge to the cycle; when outside of the limit cycle, trajectories initially aim for the center point, and spiral in to the limit cycle as they approach it. Figure 4 shows several trajectories through the X-Y plane to illustrate the effects of the direction field and the resultant limit cycle.

We have found that the Hopf bifurcation satisfies the constraints on mini-UAV flightpath generation listed previously. In particular, the system of equations is extraordinarily simple to compute:

$$\frac{dx}{dt} = y + x\left(\mu - x^2 - y^2\right)$$
$$\frac{dy}{dt} = -x + y\left(\mu - x^2 - y^2\right)$$

Parameters can be added to move the orbital center, control the radius of the limit cycle, and adjust the "gain," or deviation tolerance, of the system. Heading targets can be obtained through the arctangent of the parametric output.

$$x_c = \hat{x} - x_t$$
$$y_c = \hat{y} - y_t$$
$$\frac{dx}{dt} = y_c + \frac{x_c}{\alpha r^2}\left(r^2 - x_c^2 - y_c^2)\right)$$
$$\frac{dy}{dt} = -x_c + \frac{y_c}{\alpha r^2}\left(r^2 - x_c^2 - y_c^2)\right)$$
$$\psi = arctan\left(\frac{dy}{dx}\right)$$

Where $(\hat{x}, \hat{y})$ is the (possibly estimated) GPS position of the UAV, (x_t, y_t) is the target GPS position, α is the deviation tolerance, r is the desired orbital radius. ψ is the output of the system: the desired aircraft heading. With a some algebraic simplifications, the operation takes only the arctangent (`atan2`) call, 8 multiplies, and 8 additions. The routine can thus be scheduled to run quite often on the processor without noticible degredation of other mission-critical tasks which must be multitasked cooperatively.

A. GPS Extrapolation

Mini-UAV platforms using small single-package GPS units which provide position updates only once per second will experience similarly timed "jerks" in the heading target. Airframes orbiting a target point with low-frequency GPS updates will thus experience periodic rocking of the airframe as the autopilot strives to match the sudden changes in desired heading. We have found that integrating the gyroscope estimates of heading and the previous groundspeed (provided by the GPS unit) to produce predicted GPS positions while awaiting readings from the GPS unit helps immensely in "smoothing out" the orbit trajectory. Although this method is somewhat crude, mini-UAVs fly at such low velocities that the predicted position error is not overly severe, and the net effect is a far smoother trajectory. Higher-order models would allow for more precise trajectory projection, but at the cost of a higher computational load.

B. Wind Tolerance

We have extensively tested this method of orbit generation in conditions ranging from total calm to winds that nearly exceeded the maximum airspeed of the UAV. We have found that performance degrades acceptably with moderate winds; even with winds at 65% of maximum airspeed, orbits are still usable (Figure 5). As winds increase beyond 75% of maximum airspeed, however, the groundspeed of the aircraft drops less than 5 meters per second, which is so low that the heading vector returned by the GPS unit becomes increasingly unreliable, and the uncorrected heading gyroscope drift becomes a major problem. This results in erratic and unacceptable trajectories (Figure 6). With our current autopilot design, in such extreme conditions, the aircraft must be flown manually. However, we are developing guidance routines which incorporate small two-axis magnetometers to correct for heading gyroscope drift, and we anticipate that this design will allow for superior flight control in high winds.

IV. Low-level UAV-Centric Control

Many autonomy modes and interaction schemes can be created using a mini-UAV and a gimbaled camera mount. The following sections discuss several interaction schemes which serve as benchmarks in the "gimbaled human-UAV" interaction space.

A. Fixed Gimbal

The simplest way to handle the gimbaled human-UAV interaction problem is to fix the orientation of the gimbal, and allow the operator to guide the UAV in a semi-autonomous interaction mode [5]. The gimbal can be aimed slightly downward when near the target to give more of the frame to ground-based targets, and aimed straight out the nose of the UAV to provide maximum situation awareness during takeoff and landing. The deficiency of this approach, however, is that much of the flight time is spent lining up passes over the target, which can only be kept in the camera's field of view for several seconds at a time.

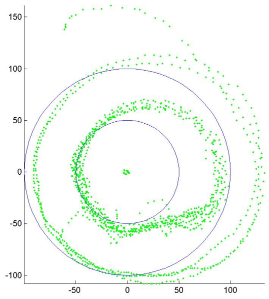

Fig. 5. Winds of 65% airspeed distort the orbits, but they are still usable. Two orbital radii are shown in this figure, the outer orbits being 100m and the inner being 50m. Ideal trajectories are shown in thin blue lines, while real-world data is shown in green dots.

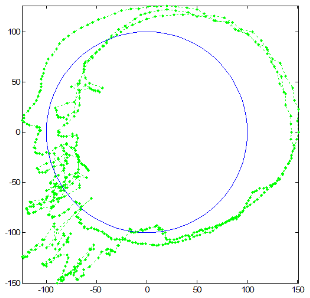

Fig. 6. Winds >75% airspeed cause heading estimation errors that lead to trajectories that are wildly off-course, including many segments where the actual heading differs from the ideal by 90 degrees.

B. Manual Control

Off-the-shelf handheld controllers are readily available that provide a small joystick under each thumb and a number of control buttons (figure 7) [6]. It is possible to use the left stick for attitude control of the aircraft and the right stick for gimbal position. To map the right joystick to the range of travel of the direct-drive gimbal assembly, we must choose where the limited range of travel of the gimbal azimuth will be placed.

Although the choice is somewhat arbitrary, we place the leftmost extreme of the azimuth travel in the straight-ahead position. This produces the maximum compatibility with our orbit generation algorithm, which under ideal conditions will produce a constant gimbal azimuth angle of 90 degrees (i.e. out the right wing of the UAV). Because the direct-drive assembly can produce approximately 135 degrees of travel, there is ample room for error due to wind or other factors.

To map this range of azimuth travel to the right joystick of the handheld controller, we have the upper-left extreme

Fig. 7. Very inexpensive off-the-shelf controller with two analog joysticks, one digital "direction pad," and ten buttons.

of the right joystick travel correspond to the "straight-ahead" gimbal position. Pulling the right joystick all the way down produces a 90-degree gimbal pitch (i.e. straight down), and pushing the right joystick fully right causes the gimbal to slew clockwise as much as possible. When the right joystick returns to center, it produces as 90-degree azimuth angle and a 30-degree pitch angle. Therefore, it is possible to keep the target in the field of view if the aircraft is flown manually in an ideal orbit and no pressure is applied to the right joystick.

Real-world tests using this interaction scheme have demonstrated that even with experienced operators, it is virtually impossible to control both joysticks at once due to communications lags and spatial disorientation. However, alternating between joysticks is feasible and effective; the operator first uses the right joystick to place the gimbal in the straight-ahead position, and then flies the UAV with the left joystick so that it is headed toward an object of interest. Releasing all pressure from the left joystick commands the UAV to maintain level flight, during which the right joystick can be used to track and observe the target.

V. High-Level Target-Centric Control

Given a target GPS point, the Hopf bifurcation flightpath generator discussed in the previous section can enter and maintain a stable orbit around the target. Camera slew angles can be generated by rotating the UAV→target vector by the airframe attitude rotations and extracting the angles using inverse trigonometric functions, as shown in Figure V, where **P** is the vector pointing from the UAV to the target in the world frame, **B** is **P** rotated into the UAV body frame, and ψ_g is the gimbal azimuth angle (positive meaning clockwise), and θ_g is the gimbal elevation angle (zero meaning level, $\pi/2$ meaning straight down).

(Note that many optimizations are possible to the previous equations, the easiest and most effective of which is to store the sines and cosines of ϕ, θ, and ψ after initially calculating them.)

The great caveat with this approach is that the target GPS point must be known. In some situations, this may be possible, but in many mini-UAV applications, the whole point of deploying the mini-UAV is to obtain the position of one or more targets, whether they be enemy vehicles, lost hikers, forest fire flashpoints, etc.

Requiring the user to manually create GPS positions of unlocalized targets in the camera frame through trial and error is a tall order, if not completely impractical. However, we have had success with this approach when the relative position of targets from the launch point is known (i.e. "500 meters west of the launch point"). We have also had success in using one mini-UAV to image another mini-UAV in flight, when the "target" mini-UAV actively broadcasts its GPS position to the other UAV. These experiments have shown the feasibility of using fully-autonomus gimbaled mini-UAVs for continuous surveillance of targets, but it does not address the problem of target acquisition. We have moved to a method of adjustable autonomy for this problem, which is discussed in the next section.

$$
\begin{aligned}
P_x &= T_x - \hat{x} \\
P_y &= T_y - \hat{y} \\
P_z &= T_z - \hat{z} \\
\\
B_x &= cos(\theta)cos(\psi)P_x \\
&\quad + cos(\theta)sin(\psi)P_y \\
&\quad - sin(\theta)P_z \\
B_y &= (sin(\phi)sin(\theta)cos(\psi) - cos(\phi)sin(\psi))\,P_x \\
&\quad + (sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi))\,P_y \\
&\quad + sin(\phi)cos(\theta)P_z \\
B_z &= (cos(\phi)sin(\theta)cos(\psi) + sin(\phi)sin(\psi))\,P_x \\
&\quad + (cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi))\,P_y \\
&\quad + cos(\phi)cos(\theta)P_z \\
\psi_g &= arctan\left(\frac{B_y}{B_x}\right) \\
\theta_g &= -arcsin\left(\frac{B_z}{\|B\|}\right)
\end{aligned}
$$

Fig. 8. Gimbal control equations

VI. High-Level Camera-Centric Control

When the UAV operator is searching through terrain attempting to find a target, the easiest form of interaction is using a forward-facing camera. The UAV operator uses a joystick to control the attitude of the aircraft, which aims the camera while controlling the translational direction of the aircraft and allows the operator's sense of telepresence to greatly aid navigation and target acquisition. Autopilot autonomy allows the operator to only be concerned with pitch and roll during this process; airspeed is maintained automatically through a PID loop closed by the motor throttle.

Once the target is in view the UAV operator can cease flying the UAV and allow the autopilot to maintain straight and level flight. The operator then shifts attention to aiming the gimbal rather than the airframe, using the right joystick to aim the camera at the target as discussed previously. The goal is for the operator to notify the ground-based target acquisition algorithm when the target is precisely in the middle of the video feed. The easiest way for this to occur is for the operator to aim the camera slightly ahead of the target and allow the target to pass through cross-hairs superimposed on the display, clicking the joystick trigger when the target passes through cross-hairs drawn in the center of the video feed.

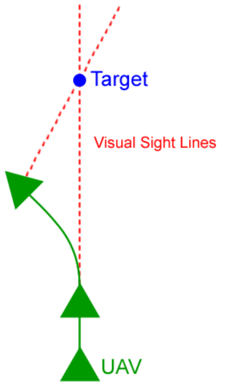

Fig. 9. Target sight lines will originate from the UAV position at time of lock, and will theoretically intersect at the target position.

When the user clicks the joystick button, the ground control station saves the roll, pitch, and yaw of the airframe, the gimbal azimuth and elevation angle, and the current GPS position estimate. The user then waits until the UAV has flown further in its current trajectory, or alters the flightpath by banking for several seconds and then returning to level flight. Either way, the UAV will have flown to a new position and the operator again clicks the joystick trigger when the target is in the cross-hairs of the camera.

After these two "target sight lines" have been acquired, the ground control station will calculate the three-dimensional point that represents the location where the target sight lines came the nearest to intersecting and upload this point to the UAV. Note that the target sight lines are rays originating from the position of the UAV at the time the joystick button was pressed and traveling through the air along the axis of the camera (Figure 9).

A simple way to describe a sight line is to use the attitude of the aircraft and gimbal to form a vector parallel to the sight line, and to later add this vector to the GPS position estimate of the aircraft at the time the sight line was recorded. The initial vector is aiming due north and parallel with the ground (corresponding to a UAV flying north and completely level, with the gimbal aiming out the nose of the UAV). This vector is then pushed through five successive linear transformations to produce a vector P_i parallel to the sight line SL_i: one rotation each for the UAV ϕ, θ, ψ, and for the gimbal ψ and θ:

$$\begin{aligned}\mathrm{P}_i = & R(\vec{z}, -\psi_u) \cdot R(\vec{x}, \theta_u) \cdot R(\vec{y}, \phi_u) \\ & \cdot R(\vec{z}, -\psi_g) \cdot R(\vec{x}, \theta_g) \cdot [0 \quad 1 \quad 0]^{\mathrm{T}},\end{aligned}$$

where $R(\vec{x}, \delta)$, $R(\vec{y}, \delta)$, $R(\vec{z}, \delta)$ are the rotation matricies about the x-, y-, and z-axes of δ radians, respectively.

A numerical approximation algorithm to find an estimate of the coordinate where the n target sight lines intersect can be created by creating m planar slices through the Z-axis that are perpendicular to the X-Y plane. The set of slices can start at the current altitude of the UAV and end at the lowest possible altitude of the target (for example, 0 meters above launch point).

The sight lines can be projected to intersect with each planar slice, forming n points, $p_1, ..., p_n$ on each plane, and the center of mass c can be computed on each plane. Then, the sum of the squared errors from the center of mass can be summed from each plane, and the center of mass of the plane having the least squared error will serve as the estimate of the target point.

The z-plane with the lowest squared error e_z corresponds to the z-plane that has the tightest grouping of sight line projections, and thus an estimate of the desired target triangulation point of the operator. The x-, y-, and z-coordinates of the center of mass on the z-plane with the lowest squared error are uploaded to the UAV to serve as the estimated target point.

When the UAV operator feels that he or she has presented the algorithm with enough target sight lines to calculate a target point (typically two or three, depending on precision), a joystick button is pressed which activates the flightpath generation algorithm around the estimated target position and automatically aims the camera gimbal at this point. Ideally, the UAV would then orbit the object of interest while continually imaging it.

However, due to attitude estimation errors, communications delays, and gimbal inaccuracies, the calculated target point will likely be incorrect. We have found that using the "direction pad" on the handheld controller to move the target position greatly helps the user to precisely define the target point. Because the precision of the target point estimation is directly correlated with the quality of the imaging returned, assisting the user in defining the target point is of paramount importance. Assuming that the UAV is maintaining zero pitch, the motions of the "target position" controller with respect to the camera frame can be mapped into changes in the world-relative target coordinates:

$$\begin{aligned}\Delta T_x = & \quad \sigma \cdot J_x \cdot cos\left(\psi_{uav} + \psi_{gimbal}\right) \\ & + \sigma \cdot J_y \cdot sin\left(\psi_{uav} + \psi_{gimbal}\right) \\ \Delta T_y = & \quad \sigma \cdot J_x \cdot -sin\left(\psi_{uav} + \psi_{gimbal}\right) \\ & + \sigma \cdot J_y \cdot cos\left(\psi_{uav} + \psi_{gimbal}\right)\end{aligned}$$

Where ΔT is the change in the world-relative target coordiates, J_x and J_y are the x- and y-positions of the "target position" controller (normalized into [-1, 1]), ψ_{uav} is the heading of the UAV (north = 0, east = $\pi/2$), and ψ_{gimbal} is the airframe-relative gimbal heading (straight ahead = 0, straight out the right wing = $\pi/2$).

The end result is that the user need not maintain a mental model of the world-frame position of the UAV in relation to the target point; instead, the user can move the target point with the "target position" controller, thereby moving the target *in relation to the camera frame*. If the target is on the right side of the camera frame, the operator can center the target by pushing the "target position" controller right, and the UAV will slew the gimbal right as well as adjusting the flightpath to adapt to the newly changed target position.

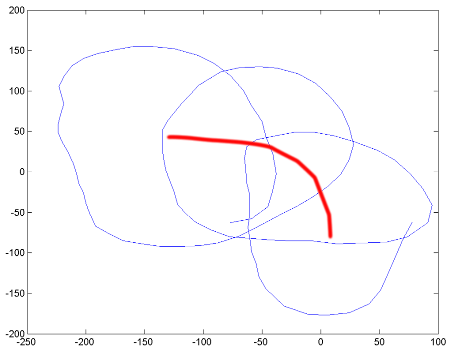

Fig. 10. Real-world automatically generated flightpath (thin blue line) of the UAV seeking constant-radius orbits around a user-controlled moving target point (thick red line).

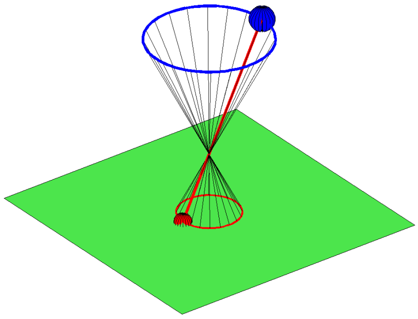

Fig. 11. Estimation error of the altitude of the target point causes the actual target (red, lower-left) to traverse a circle through the camera frame as the UAV (blue, upper-right) moves in its orbit.

In our current implementation, the desired target point is uploaded twice per second to the UAV in response to operator input. The autopilot continually slews the camera to aim at the target point, and the autopilot continually seeks a constant-radius orbit around the target point. A key point is to note that the interaction between the operator and the UAV has completely changed; the operator is no longer flying the UAV, but rather controlling the target point and allowing all other flight control parameters to be generated by the autonomy of the intervening computer systems. In Figure 10, the target point begins in the lower-right of the figure. The trajectory of the target point, as created by an operator who was monitoring the UAV video feed to direct the target point down a rural street, is represented by the heavy red line. The actual flightpath of the UAV, as generated in real-time on the UAV's onboard microprocessor, is shown by the thin blue lines. The data was gathered by polling the UAV twice a second and recording the reported position data.

In addition to the north and east coordinates of the target, the target's altitude is a very important factor as well. Flat target areas would eliminate the need for estimating target altitude, but unfortunately large mountains are found in many areas of military importance and a very high percentage of locations searched by wilderness search-and-rescue. Although digital elevation maps could be indexed using the target GPS coordinate triangulated from the target sight lines, we sought to develop a process that would work in the absence of highly detailed digital elevation maps. Errors in target altitude estimation are readily apparant: if the estimated target altitude is above or below the actual target, the target will appear to trace a circle in the camera frame. This is due to the conic sections being produced by the UAV camera system: the orbiting UAV traces a circle in the sky, and its camera sight lines form the point of an inverted cone. If the object of interest is above or below the estimated target point, the object will traverse a circle through the camera frame, as shown in figure 11.

To allow the user to correct for this error, we have mapped two buttons on the controller to raising and lowering the estimated altitude of the target point. In an ideal world, the initial target estimation produced using the target sight line acquisition method would be perfect; however, real-world flight tests have demonstrated that errors in the barometric altitude estimation of the UAV, imprecision in attitude estimation and gimbal position, and operator error can produce an incorrect target altitude estimation. The user can adjust the estimated altitude of the target point until the target ceases to precess through the camera frame, which corresponds to effectively sliding the estimated target point (the intersection of the upper and lower cones in Figure 11) up and down the world z-axis until the estimated target point is placed precisely on the object of interest.

This interactive target acquisition method allows the mini-UAV operator, with minimal airborne hardware, to precisely localize the UAV-launch-relative coordinate of the target, as well as its altitude with respect to the launch point of the UAV. This relative target point can then be quickly and easily transformed into a global GPS coordinate for communication to other members of the search team.

VII. Conclusions

This paper has described gimbal hardware, flightpath generation algorithms, and a human-UAV interaction scheme that operates in both simulation and real-world hardware. These techniques combine to enable mini-UAVs to perform target acquisition, localization, and continuous surveillance in real-world environments.

References

[1] R. Beard, et al, "Autonomous Vehicle Technologies for Small Fixed-Wing UAVs," AIAA Journal of Aerospace Computing, Information, and Communication, (to appear).

[2] S. Yoon and J.B. Lundberg, "Equations of Motion for a Two-Axes Gimbal System," IEEE Trans. on Aerospace and Electronic Systems, vol. 37, no. 3, pp. 1083-1091, July 2001.

[3] M. Draper and H. Ruff, "Multi-Sensory Displays and Visualization Techniques Supporting the Control of Unmanned Air Vehicles," in "Vehicle Teleoperation Interfaces," a workshop of ICRA 2000.

[4] J.E. Marsden and M. McCracken, The Hopf Bifurcation and Its Applications, New York, Springer-Verlag, 1976.

[5] M. Quigley, M. Goodrich, and R. Beard, "Semi-Autonomous Human-UAV Interfaces for Fixed-Wing Mini-UAVs," IROS 2004 (to appear).

[6] www.thrustmaster.com