

Vision-Only Aircraft Flight Control Methods and Test Results

Alison A. Proctor* and Eric N. Johnson†

Georgia Institute of Technology, Atlanta, GA, 30332, USA

An unmanned aerial vehicle usually carries an array of sensors whose output is used to estimate the vehicle's attitude, velocity and position. This paper details the development of control strategies for a glider, which is capable of flying from a starting point to a ending location using only a single vision sensor. Using vision to control an aircraft presents a few unique challenges. Firstly, absolute state measurements are not available from an image. Secondly, in order to maintain adequate control of the aircraft, the images must be processed at a fast rate. The image processor utilizes an integral image representation and a rejective cascade filter to find and classify simple features in the images, reducing the image to the most probable pixel location of the objective. The navigation algorithms use an extended Kalman filter to generate state estimates based on measurements obtained from the imagery. The algorithms are tested through the flight testing of a glider instrumented only with a single camera.

Nomenclature

B_{δ_r}	Coefficient for Sideslip from Rudder
d	distance to the window (ft)
dt	time step (sec)
dx	horizontal displacement of window from the center of the image (pixels)
dy	vertical displacement of window from the center of the image (pixels)
EKF	extended Kalman filter
FOV	field of view (rad)
GCS	Ground Control Station
GNC	guidance, navigation and control
$HITL$	Hardware-in-the-loop
K	Control Gain
N	width of the window (ft or pixels)
PFC	primary flight control computer
px	unrolled horizontal displacement of window from the center of the image (pixels)
py	unrolled vertical displacement of window from the center of the image (pixels)
Q	Coefficient for Elevator effectiveness
R	Coefficient for Rudder effectiveness
RC	radio controlled
SF	Scale factor relating pixels to angular displacement
$SITL$	Software-in-the-loop
UAV	Unmanned Aerial Vehicle
v	velocity (ft/sec)
y	lateral distance from the trajectory (ft)
Z	width or height of the image (pixels)

*Graduate Research Assistant for Aerospace Engineering, Atlanta GA, Student Member, alison_proctor@ae.gatech.edu.

†Lockheed Martin Assistant Professor of Avionics Integration in Aerospace Engineering, Atlanta GA, AIAA Member, Eric.Johnson@aerospace.gatech.edu.

Symbols

β_r	sideslip due to rudder deflection (rad)
δ	command signal
γ	track angle (rad)
ϕ	roll angle (rad)
ψ	yaw angle or heading (rad)
ψ'	modified heading angle (rad)
θ	pitch attitude angle (rad)
$\hat{\cdot}$	estimated
a	aileron
e	elevator
H	horizontal in the local frame
p	pixels
r	rudder
$trim$	trim settings
ur	uncoupled from roll estimate
V	vertical in local frame
W	window
X	horizontal in the camera image
Y	vertical in the camera image
c	crab

I. Introduction

Autonomous mechanical flight has historically utilized a wide array of sensors to determine the position and attitude of the vehicle. This is in contrast to what is found in nature. Human beings do not use GPS and infra-red sensors to find their way through a door, and birds of prey do not require ground personnel to send up the coordinates of game before they attack. Although we all have other senses that we use, vision is often dominant. These ideas are what prompted this research. If a human pilot can visually acquire a specified destination, he or she can fly to it; therefore, a computer presumably should be able to do the same.

The concept of using vision to control a robot is not new, researchers have been discussing the use of vision in the feedback control loop since the 1970's.¹ In 1979 Hill and Park first coined the term *visual servoing* to refer to a control system that uses vision directly in the feedback loop as opposed to *look then move* systems.² The theories developed in this field have been successfully implemented in manufacturing robots for many years. The demand for more precise visually controlled robots has spawned research in many different directions including real-time image processing techniques and high-precision camera calibration methods.

Until recently, most visually controlled robots have had a fixed base which gives it holonomic kinematics and a well defined set of possible configurations. This has enabled researchers to develop control theory for vision sensors and the dynamic effects of using a vision sensor in a feedback loop.³ However, in the case of an aircraft, the position and attitude are unconstrained; therefore, these theories do not necessarily apply. There has been some interest in expanding the theory for visual servoing to non-holonomic systems such as mobile robots.⁴ However, this concept has by no means been completely explored,. Therefore, there is little theory to describe the use of vision for aircraft flight control.

Using vision to augment pilot aids is currently being pursued by several groups. The German Aerospace Center, DLR is working towards developing an ILS system which uses images from an uncalibrated camera to calculate roll, lateral displacement and the glideslope of the aircraft.⁵ Their work is complemented by new research into Integrated Enhanced Vision Systems (IEVS), which is currently being conducted by Max-Viz, Inc in conjunction with Oregon Health Sciences University's Oregon Graduate Institute (OGI) in Portland, OR. This work extends their current line of enhanced vision products and deals with using multi-sensor image data to verify imagery from other on-board avionics systems.⁶

Unmanned aerial vehicle's (UAV's) are also making use of vision; the most prevalent area of research is in the development of UAV's that utilize vision in conjunction with other sensors. In this case, the vision is used for trajectory planning or for precise determination of a relative position.^{7,8} Vision has even been used as the sole sensor for guiding an unmanned blimp, whose dynamics can be approximated by a three degree

of freedom system, through a controlled environment.⁹

The research presented in this paper differs from previous work in that it utilizes only vision for determining the position, velocity, and attitude of the vehicle, the desired flight path, and the required control commands, in an outdoor environment. Therefore, this is a more intricate estimation problem than has been previously investigated. In contrast to other flight control systems, a completely vision based flight control system has the advantage of having a simple hardware configuration. This makes it a low cost solution for an autopilot, and an attractive solution for a completely isolated back-up system. Initial investigations into this type of controller using simulation has been performed already.¹⁰ This paper details methods that are capable of performing this task in a real-world scenario. In addition, test results, using both simulation and a real aircraft, are presented which verify their success.

II. Preliminary Test Scenario

The task chosen to demonstrate the precision of the control strategies is to fly an autonomous glider, instrumented with only a camera, at an open window. Since the geometry of the destination is well defined, this objective provides a clear framework for developing guidance, navigation, and control (GNC) laws using only vision. Initiating the flight from a stable platform simplifies the problem slightly, as the operator can ensure that the image processor has properly acquired the target destination prior to launch. To accomplish this, the aircraft needs to be launched from a stationary location at the correct elevation relative to the window.

III. Test Vehicle

Four distinct systems are used to fly this scenario: the vehicle, the control hardware, the video acquisition/transmission system, and the flight control computer (PFC). Since this technology is still in a primitive stage, it is desirable to perform the initial flight tests on a small vehicle with relatively stable flight dynamics. In order to eliminate the need for a customized flight control computer, the flight control calculations and image processing are performed on a remote computer that can be located separately from the aircraft. This eliminates the size and weight constraints for this component, and reduces the required payload for the aircraft. The video is transmitted to the PFC using a wireless transmitter; similarly, the control commands are transmitted back to the vehicle through a standard radio-controlled (RC) transmitter.

Based on desired flying characteristics and the payload requirements, a 2-meter 2-axis glider was chosen. The glider is equipped with elevator and rudder, and the two meter wing span provides sufficient payload capacity to ensure a very slow stall speed with the additional weight from the video system. The glider is shown in Figure 1.

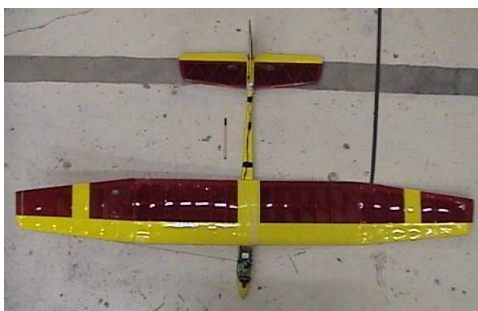


Figure 1. Test Glider

A. Imaging System

The imaging system uses an analog camera and wireless video transmitter. The 460 line color CCD board camera used is shown in Figure 2-a, the camera is photographed next to a quarter to indicate scale. The camera has a standard micro lens which provides a high contrast picture in most lighting conditions. The

small dimensions of the camera makes it suitable for the glider. The camera casing was constructed from latex rubber to minimize weight.

The choice of video transmitter was made primarily on size, range, and frequency. A 1200 MHz system was chosen because it does not interfere with the 900 MHz and 2.4 GHz bands which are used in the wireless communication systems on other vehicles operated at the Georgia Tech UAV Research Facility.¹¹ This ensures that the video transmission will not suffer adverse effects from the datalinks associated with other vehicles, which may be flown in tandem. The VTRM-1200 video transmitter and receiver set is shown in Figure 2-b. The transmitter is 0.875" x 1.125" x 0.33" and weighs 15 g with antenna, which makes it suitable for use onboard the glider.

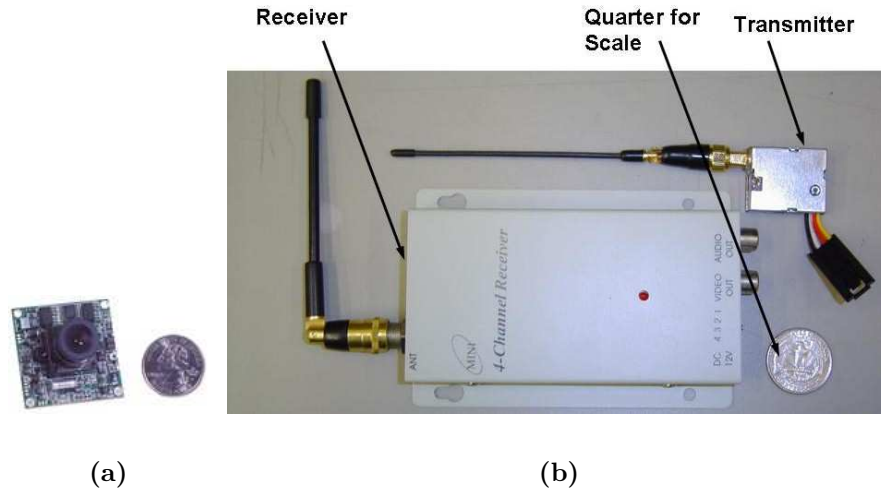


Figure 2. (a) Video Camera, (b) Transmitter and Receiver

B. Primary Flight Control Computer

The PFC must perform three functions: acquire an image, process the image, and calculate and distribute the actuator commands. These tasks are run on a Versa Logic VSBC-8 Pentium III 850 MHz single board computer, which has been augmented with the Sensoray 311 PC/104+ framegrabber. The framegrabber is used to digitize individual frames from the analog video signal coming from the imaging system. This enables frame acquisition at speeds up to 30 frames/sec to be achieved without the need for a complex video interface.

In order for the PFC to control the servos, the commands must be transmitted to the RC receiver onboard the glider. This is done through a custom PC/Futaba interface. During manual flight the glider is controlled using a Futaba FP-7UAFS transmitter, which is equipped with a trainer chord that can take external servo commands encoded as a pulse-position modulated (PPM) signal. Since commercial interfaces between a PC and Futaba transmitters are not available, a custom interface was constructed. The conversion is performed by an Altera APEX20KE - FPGA board. The FPGA reads actuator commands from the PFC serial output (RS-232) and outputs the PPM signal to the RC transmitter.

IV. Guidance, Navigation, and Control

The flight control software is comprised of an image processor, the navigation filter, and the controller. Once an image has been obtained from the video stream, it is processed to find the most probable window candidate. Then the location of the window candidate in the camera image is passed to the navigation system. The state estimate is updated, and new servo commands are generated by the controller.

A. Image Processing

There are many ways of extracting relevant information from an image, but doing it quick enough is critical. Many standard image processing routines involve filtering the image and then performing some sort of edge detection. This type of routine can be computationally expensive, especially in noisy images. Then, once the window is acquired, the image processor must track it. This means that image processor cannot jump between multiple window candidates. Ultimately, high speed and high precision are opposing requirements.

A strategy for finding and tracking the window was developed in previous work.¹⁰ The strategy uses high speed pattern matching to find the square window; the speed is achieved by using an integral image of pixel intensities.¹² This representation of the image is used to quickly compare features in a single layer image. Then the probability that a feature is actually the destination point is determined using a rejective cascade of classifiers, as shown in Figure 3-a. This classifier evaluates the probability of each pixel being the center of the window, using successively more rigorous tests. By wisely choosing the evaluation criteria, up to 70% of the pixels can be ruled out with very simple and fast comparisons.

The classifier used in this work differs slightly from that presented in the previous work by deWagter, Proctor, and Johnson, in that there is a third comparison made for each potential window candidate. The final comparison estimates where the window should be in the image based on its movement through previous frames. This comparison augments the score for all the window candidates which are close to the estimated position even if they did not pass the first two classifications, using the proximity function shown in Figure 3-b. This function is a truncated normal distribution, where the radius of the upper surface is representative of the uncertainty in the position estimation. This prevents the image processor from jumping around when the video is corrupted during transmission and helps localize the image processor when there are multiple windows in view at one time.

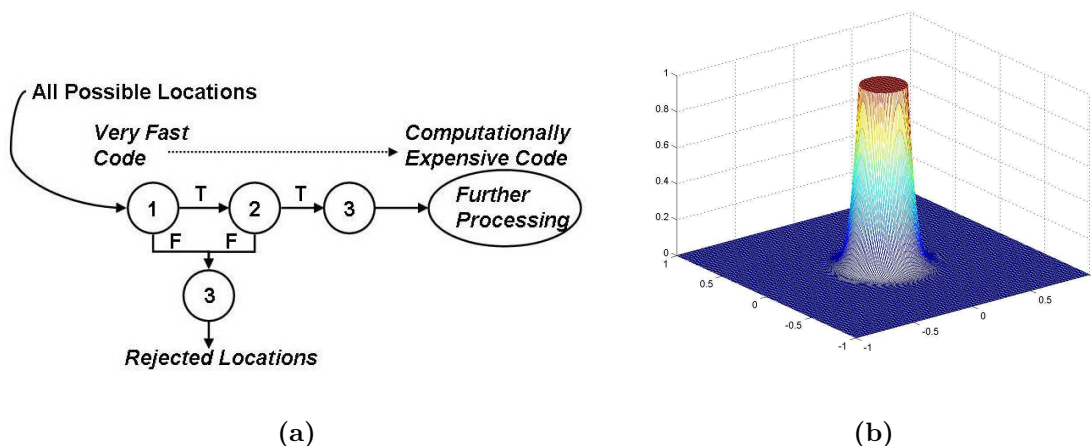


Figure 3. (a) Rejective Cascade of Classifiers, (b) Third Comparison - Proximity Classification Function

V. Navigation

The navigation process uses the measurements obtained in the image processing to estimate the position, velocity, and attitude of the aircraft relative to the window. The measurements are a function of one or more states, and it may or may not be possible to obtain enough independent measurements to find a unique solution for the position. The state estimate is obtained by using an extended Kalman filter (EKF). Then the state estimate can be used to guide and control the aircraft. In this case, complete knowledge of the aircrafts position is not required; only its position relative to the window is needed.

A. Extended Kalman Filter

1. Image Processing Measurements

The measurements obtained from the images are slightly different from the standard set of measurements in that they are not direct measurements of the state of the aircraft, rather they are measurements that are dependant on the state in a nonlinear way. Extremely precise measurements require a high resolution image and rigorous image processing techniques; therefore, the types of measurements that can be obtained from the images are limited by the speed at which the image must be processed.

There are only four measurements that can readily be obtained with this image processor and camera system; these are shown in Figure 4. Although additional quantities, such as skew, would help the position estimate, the current image processor cannot realistically make these measurements with accuracy that is practical for this situation. All measurements are in pixels, denoted by the subscript p , unless otherwise noted.

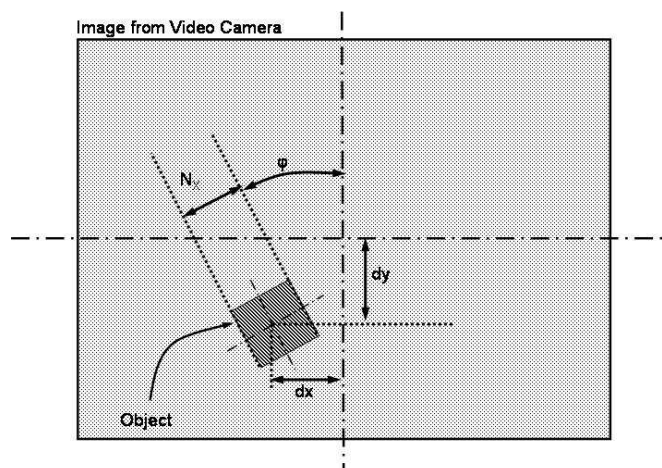


Figure 4. Measurements from an Image

The horizontal and vertical position of the window's centroid, dx_{W_p} and dy_{W_p} , come directly from the cascade filter. The linear size, N_{X_p} , can be found by counting the number of dark pixels that surround the center. Since the lighting can change from one frame to the next, the darkness threshold can be found dynamically by averaging the intensity of the inside of the window with that of the border. This method is easy to implement and works well for homogenous windows with sufficiently large lighter borders.

Assuming that the window is a square, there will be one corner located in each quadrant. Therefore, the corners can be found by recording the farthest dark point from each quadrant. Since each corner should be at a 45° angle from the x-axis of the window, the average deviation from this gives a measure of the rotation of the window, ϕ_W . Since the window may take up as few as 15 pixels in the initial portions of the flight, the initial rotation measurements can be rough.

2. Nonlinear Measurement Model

One of the key requirements of this EKF is that it converge quickly. Therefore, it is important to wisely choose the entries of the covariance matrices. From Figure 4, it is clear that dx_{W_p} and dy_{W_p} are functions of the roll angle of the aircraft as well as lateral displacement and pitch angle. Therefore, a number of potentially harmful side effects, such as sensitivity to crosswind and errors in the camera alignment, can be removed by uncoupling the roll and position measurements before they are used in the EKF. However, before the two measurements can be uncoupled an accurate estimate of the roll of the aircraft must be obtained. In this situation, the orientation of the window with respect to the earth is directly measured; therefore, with the help of the EKF the roll of the aircraft can be accurately estimated by the orientation of the window in

the image. The uncoupled horizontal and vertical position of the window can be found as:

$$\begin{bmatrix} px_{W_p} \\ py_{W_p} \end{bmatrix}_{ur} = \begin{bmatrix} \cos(\hat{\phi}) & \sin(\hat{\phi}) \\ -\sin(\hat{\phi}) & \cos(\hat{\phi}) \end{bmatrix} \begin{bmatrix} dx_{W_p} \\ dy_{W_p} \end{bmatrix} \quad (1)$$

Now the measurement model can be constructed. Since all of the measurements are made in pixels and all of the state estimates are in a local coordinate system, it is necessary to convert between the two systems. The focal length of the lens can be used as a scale factor that relates the tangent of the angular distance to a distance in pixels. The geometry for defining this scale factor is shown in Figure 5.

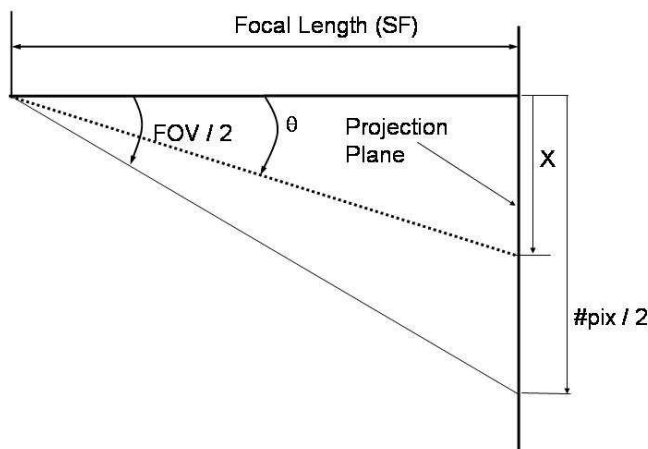


Figure 5. Geometry for Determining the Scale Factor

Therefore the scale factor, SF , can be found as:

$$SF = \frac{Z}{(2 \cdot \tan(\frac{FOV}{2}))} = \frac{X}{\tan(\theta)} \quad (2)$$

$$X = SF \cdot \tan(\theta) \quad (3)$$

where FOV is defined as the field of view, and Z is the width or height of the image in pixels.

The horizontal window size can be directly related to the distance the vehicle is from the window. Assuming that the angle between the window and the center of the image is small, the relation between the window size measurement in pixels and distance to the window in feet is:

$$\hat{N}_{X_p} \approx \frac{N_H}{d} \cdot SF \quad (4)$$

where N_H is the width of the actual window in feet, and SF is given by Eq. (2).

The rotation angle can be predicted using the roll estimate for the aircraft itself.

$$\hat{\phi}_W = \hat{\phi} \quad (5)$$

Figure 6 shows the geometry for estimating the horizontal or vertical position of the window in the image without roll. In this figure, γ_W is the track angle from the aircraft to the center of the window, ψ is the yaw angle relative to a vector normal to the window, and d is the distance to the plane containing the window. From this geometry, it is possible to define:

$$\begin{aligned} \tan(\gamma_W - \psi) &= \frac{x}{R} = \frac{(y + d \cdot \tan(\psi)) \cos(\psi)}{(d - d') \cos(\psi)} \\ &= \frac{y \cdot \cos(\psi) + d \cdot \sin(\psi)}{d \cdot \cos(\psi) - y \cdot \sin(\psi)} \end{aligned} \quad (6)$$

Then, assuming that $d \gg y$ and using small angle approximations, the unrolled horizontal position of the window can be estimated as:

$$\begin{aligned} \hat{p}x_{ur} &= -\tan(\gamma_W - \hat{\psi}) \cdot SF \\ &= -\frac{\hat{y} + \hat{d} \cdot \hat{\psi}}{\hat{d}} \cdot SF \end{aligned} \quad (7)$$

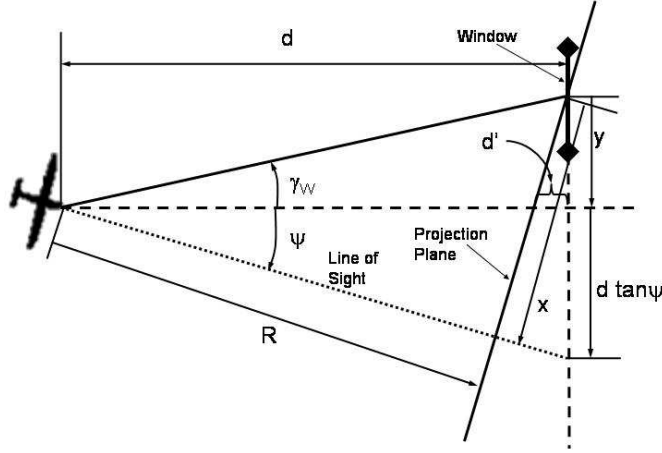


Figure 6. Geometry for the Unrolled Vertical and Horizontal Position Estimate

Similarly, using the same geometric interpretation but assuming that the altitude error is zero, the vertical position can be estimated by:

$$\hat{p}y_{ur} = -\hat{\theta} \cdot SF \quad (8)$$

3. Integrating the Crab and Sideslip Angles

The measurement model presented in the previous section is geometrically valid. However, when the rudder is deflected there is a resulting sideslip angle. This will cause the vehicle to yaw slightly and the window to move laterally in the image. The sideslip dynamics could simply be folded into the crab angle, however, the dynamics are much faster than those being captured by the crab angle. Therefore, in order to aid the convergence of the lateral dynamics, an estimate of the sideslip angle due to rudder deflection is subtracted out of the measurement. The estimate for the sideslip angle due to rudder deflection, β_r , is calculated using a simple first order system, shown in (9). For this glider a gain of 2 provided a fairly accurate sideslip angle due to rudder deflection.

$$\dot{\beta}_r = K_\beta (B_{\delta_r} \delta_r - \beta_r) \quad (9)$$

The crab angle is the third state that is estimated from the lateral measurement. The crab angle can be integrated in two ways, the traditional way would be to introduce the crab angle into the lateral dynamics in the process model. However, in practice this created coupling between the lateral offset and the crab angle which resulted in a poor estimation of both states. This problem was solved by subtracting the crab angle out of the horizontal measurement, in a similar manner to the sideslip angle, and not incorporating it into the lateral dynamics. This has ramifications in the process model which will be discussed in the next section. Therefore, the augmented estimate for the unrolled horizontal position of the window is give by:

$$\hat{p}x_{ur} = -\frac{\hat{y} + \hat{d} \cdot (\hat{\psi} + \hat{\psi}_c - \hat{\beta}_r)}{\hat{d}} \cdot SF \quad (10)$$

4. Nonlinear Process Model for the Glider

In order to accurately guide the aircraft to intercept the window, the position relative to the window, the velocity, and the attitude of the vehicle are required. By only estimating the most important states the

convergence time is minimized; the states shown in Table 1 are estimated in the EKF. Since this glider will be flown outdoors, it is important to compensate for the effects of wind. Defining a crab angle takes care of two error sources: crosswind and errors in the camera alignment. The crab angle is defined as the angle of yaw that is required to counteract the velocity of the air mass and enable the aircraft to fly straight with respect to the ground. The crab angle is defined as positive when a negative yaw angle is required.

Table 1. Navigation Filter States

Sym	Description	Sym	Description
d	Distance to the Window	ϕ	Roll Angle
y	Lateral Offset	θ	Pitch Angle
v	Velocity	ψ'	Modified Heading Angle
		ψ_c	Crab Angle

In this scenario, there is only 2-3 seconds for the EKF to converge on a solution. The lateral offset, modified heading angle, and crab angle all come from the unrolled horizontal position measurement.

The vertical flight path has a similar ambiguity to that seen in the lateral direction. The vertical measurement obtained encompasses both the pitch and an altitude error above the initial glide path. However, if the velocity is allowed to fluctuate freely, it was determined that by allowing the flight path to move with the aircraft and recalculating the desired pitch angle so that the glider intercepts the window, acceptable controller performance can be achieved. This approach simplifies many aspects of the guidance, but eliminates speed control. However, since the final velocity in this scenario is not important, it is feasible to design a flyable trajectory by selecting a launch altitude and then simply allow the velocity to vary.

The choice was made to keep the dynamic model as simple as possible. The glider model in the navigation filter is propagated forward in time using forward euler integration and is completely described using only the five parameters listed in Table 2.

Table 2. Navigation Filter Model Parameters

Sym	Description	Sym	Description
Q_{δ_e}	Elevator effectiveness	$\delta_{e_{trim}}$	Elevator trim setting
R_{δ_r}	Rudder effectiveness	$\delta_{r_{trim}}$	Rudder trim setting
P_{δ_a}	Aileron effectiveness		

The forward translational portion of the process model is very simple. The distance estimation assumes that the yaw angle is small, and the velocity is constant.

$$\hat{d}_{k+1} = \hat{d}_k - \hat{v}_k dt \quad (11)$$

$$\hat{v}_{k+1} = \hat{v}_k \quad (12)$$

where dt is the discrete time step for the update.

The attitude dynamics take into account the effect of the control surfaces. With this type of aircraft it is acceptable to approximate the effects of a rudder deflection directly in the rolling dynamics as if it were ailerons. Therefore, the roll angle is estimated by:

$$\hat{\phi}_{k+1} = \hat{\phi}_k + P_{\delta_a}(\delta_{r_k} - \delta_{r_{trim}}) \cdot dt \quad (13)$$

This simple model gives a satisfactory propagation of roll. The rudder trim setting is used mainly to counteract any adverse roll due to asymmetries created in the aircraft during construction. Similarly, the pitching dynamics can be captured by:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + Q_{\delta_e}(\delta_{e_k} - \delta_{e_{trim}}) \cdot dt \quad (14)$$

The lateral dynamics are more intricate. Firstly, the crab angle can simply be approximated as a constant, since it will typically have fairly slow dynamics.

$$\hat{\psi}_{c_{k+1}} = \hat{\psi}_{c_k} \quad (15)$$

Since the crab angle is removed from the measurement instead of being included in the lateral dynamics, the resultant yaw angle is essentially the angle of the velocity vector, or track angle. This quantity is referred to as the modified heading angle. However, since the crab angle is assumed to be constant, the modified heading angle dynamics will be identical to that of the actual heading angle. The yawing motion is a byproduct of the rolling and pitching dynamics as well as the rudder control itself. Assuming a coordinated turn, the modified heading angle is propagated by:

$$\begin{aligned}\hat{\psi}'_{k+1} &= \hat{\psi}'_k + g \cdot \tan\left(\frac{\hat{\phi}_k}{\hat{v}}\right) \cdot dt \\ &+ Q_{\delta_e}(\delta_{e_k} - \delta_{e_{trim}})\sin(\hat{\phi}_k) \cdot dt \\ &+ R_{\delta_r}(\delta_{r_k} - \delta_{r_{trim}}) \cdot dt\end{aligned}\quad (16)$$

Since the modified heading angle represents our approximation of the angle of the velocity vector. The lateral displacement, y can be approximated as:

$$\hat{y}_{k+1} = \hat{y}_k + \sin(\hat{\psi}'_k)\hat{v}_k dt \quad (17)$$

B. Control

Once the EKF has a adequate estimate of the aircrafts position relative to the window, guidance and control are relatively simple. The guidance laws determine a desired attitude to intercept the window and the appropriate rudder and elevator commands to achieve that attitude.

Longitudinal control must maintain a tight enough trajectory to intercept the window in a very short period of time and, it must ensure that changes in attitude are slow enough that the image processing can retain a lock on the window. The pitch controller is simply a proportional feedback controller which drives the relative pitch, $\hat{\theta}$, estimate to zero.

$$\delta_e = -K_\theta \hat{\theta} + \delta_{e_{trim}} \quad (18)$$

Through simulation it was determined that the upper bound on the gain of the pitch controller is $K_\theta = 3.0$. This provides a sufficiently accurate pitch command without moving the aircraft so drastically as to undermine the image processing.

The roll/yaw controller must determine the rudder command required to fly the horizontal flight path. For simplicity, the modified heading angle was transformed back into the heading angle by adding the estimate of the crab angle. This allows the desired roll command to be determined using a standard formulation with y , ψ , and ψ_c .

These three components are weighted by gains and added together. The desired roll angle is in radians; therefore the gain for y needs to be in units of ft^{-1} . Additionally, the desired contribution of y will be dependant on the velocity. Therefore, by scaling K_y by v in the control equation, the gain can be chosen as a number with units of sec^{-1} , which can roughly represent how quickly the glider should minimize the lateral displacement y . Then the desired roll angle can be formulated as:

$$\phi_{des} = -\frac{K_y}{\hat{v}}\hat{y} - K_\psi\psi - K_{\psi_c}\psi_c \quad (19)$$

Once the desired roll angle is determined it is clipped to ensure that it remains between -20 and 20 degrees. If the aircraft rolls more than this the results of the image processing will begin to degrade, and the aircraft will become difficult to recover. Then a simple feedback controller can be used to maintain this roll position.

$$\delta_r = K_\phi(\phi_{des} - \hat{\phi}) + \delta_{r_{trim}} \quad (20)$$

In this case, simulation has shown that using an outerloop gain $K_\phi = 2.5$ with innerloop gains K_y between 0.1 and 0.5 and K_ψ and K_{ψ_c} between 0.2 and 0.75, generates acceptable lateral control.

VI. Simulation and Test Results

The flight control method presented in this paper, has been tested in simulation and through flight tests of the glider shown in Figure 1. Two types of simulation are used for testing software-in-the-loop (SITL) and

hardware-in-the-loop (HITL). The SITL simulation is a completely self contained simulation that can run on a personal computer. The simulation generates a synthetic scene using OpenGL, which can be viewed from multiple viewpoints. In order to make the simulation more realistic, the environment includes trees, buildings, wind and air turbulence. The aircraft dynamics are determined using a nonlinear rigid body model of the glider, with an actuator model that includes saturation and rate limits. The flight control code that normally runs on the PFC along with the code that runs the ground station is compiled into the simulation. This allows a developer to test new algorithms and additions without tying up the flight hardware. The camera and video transmission system is simulated using a scene window from the viewpoint of the camera. The pixel information is read from this scene at the same rate that images would normally come from the framegrabber.

The first set of results shows the output of the navigation filter with a known initial position and an unknown cross wind of 5 ft/sec. Figure 7-a shows the lateral and longitudinal position estimates versus the actual position of the aircraft. The longitudinal position estimate is quite accurate, and the error in \hat{y} due to the unknown crosswind decreases as it approaches the window.

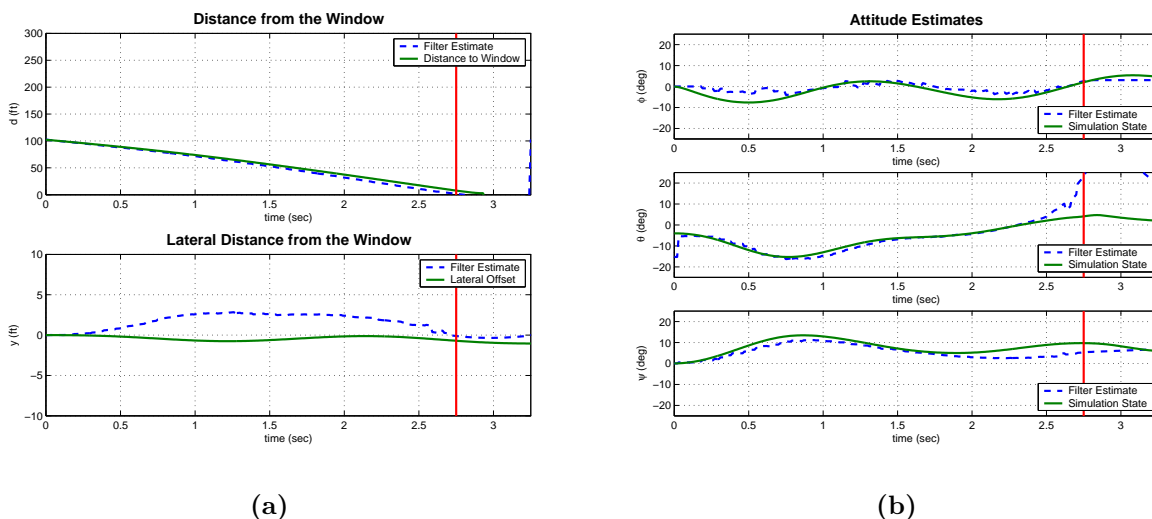


Figure 7. Comparison between the output of the EKF and the recorded state of the dynamic model. (a) Position estimate, (b) Attitude estimate

The attitude estimate shown in Figure 7-b. In the second graph the relative pitch angle has been transformed back into a local pitch angle for comparison purposes. In addition, the modified yaw angle has been transformed back into heading angle. In this graph, it is possible to see how closely the heading angle tracks the heading of the aircraft.

The velocity and crab angle estimates are shown in Figure 8-a, along with the actuator commands in Figure 8-b. In the top graph the crab angle quickly attains the desired value and maintains it throughout the flight. This is a remarkable result as it does not exhibit any coupling with the other lateral dynamics.

A. Hardware-in-the-Loop Simulation Results

HITL simulations use the same premise as SITL but include varying amounts of the flight hardware in the loop. The results presented in this section are from HITL simulations. In these cases, the actual PFC, camera, and GCS are used in the simulation. The hardware setup is depicted in Figure 9-a. In this case, the GCS and PFC are configured as normal, but the camera is pointed at a third computer. The third computer runs a copy of the simulation, and feeds the actuator commands from the PFC into the dynamic model. Then the camera view of the synthetic scene is captured by the camera.

This type of testing provides several advantages over SITL simulation. In SITL, the simulation will not run in real time, the update rate required for the scene generation prohibits this. Secondly, in SITL the image acquisition is perfect, while in reality there is distortion from the lens, video dropouts and bad lighting. The HITL simulation, however, does run in realtime, as the processing is distributed across several

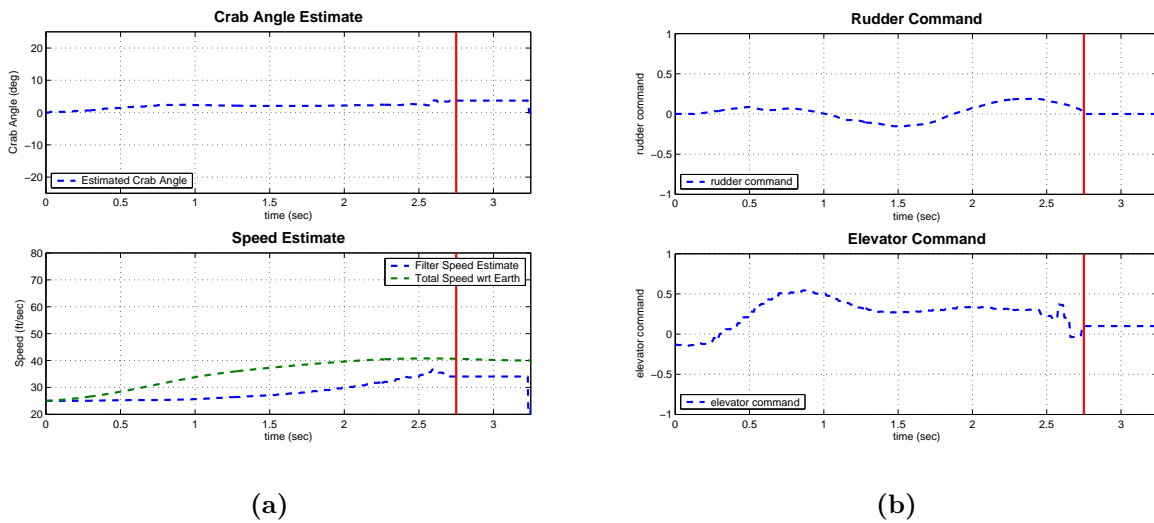


Figure 8. (a) Crab angle and Velocity Comparison, (b) Control Commands (-1 to 1)

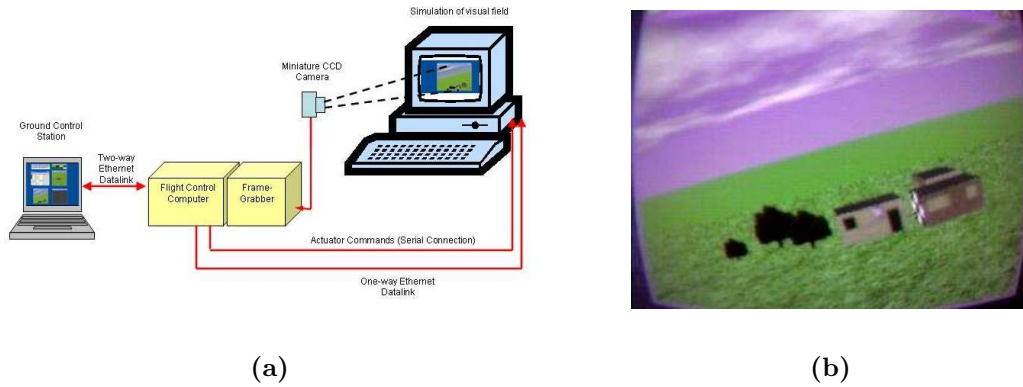


Figure 9. (a) HITL hardware setup, (b) Typical HITL image with distortion

computers. Therefore, it gives a more rigorous test to the timing algorithms, and the processing time for the image processing. Additionally, it is useful for testing the communication links between the computers and the vehicle, as well as incorporating many of the issues associated with image acquisition. The distortion in a typical image with HITL is shown in Figure 9-b. The bowing of the image is caused by the camera being so close to the monitor. This creates far more distortion than is normally seen in flight, which is a very rigorous test of the EKF. In addition, the interference between the frequency of the monitor and the camera create an effect which is similar to video dropouts, which tests the robustness of the tracking algorithm.

One of the difficulties that the GNC algorithms must handle, is unknown initial conditions. Errors in the estimate of the initial condition require the EKF to converge quick enough to guide the glider through the window. The test shown in the figures below is a typical real world scenario where there is an unknown adverse roll angle off the launch. The glider starts 100 ft away from the window and 30 ft above the ground. In this scenario, there is no cross wind.

Figure 10-a shows the position and lateral offset estimates. The error in the lateral estimate is far less than seen in Figure 7-a where there was a crosswind. The error in the distance are primarily due to the error in the window size from the lens distortion.

Figure 10-b shows the attitude estimates. Once again, as was seen in Figure 7-b the attitude estimates are very good. The error scene in the pitch attitude is primarily due to an unavoidable vertical misalignment between the camera and the screen. The roll estimate has a little more noise than in the results from the SITL tests, due to the degradation of the image quality in HITL, but the overall trend is well captured.

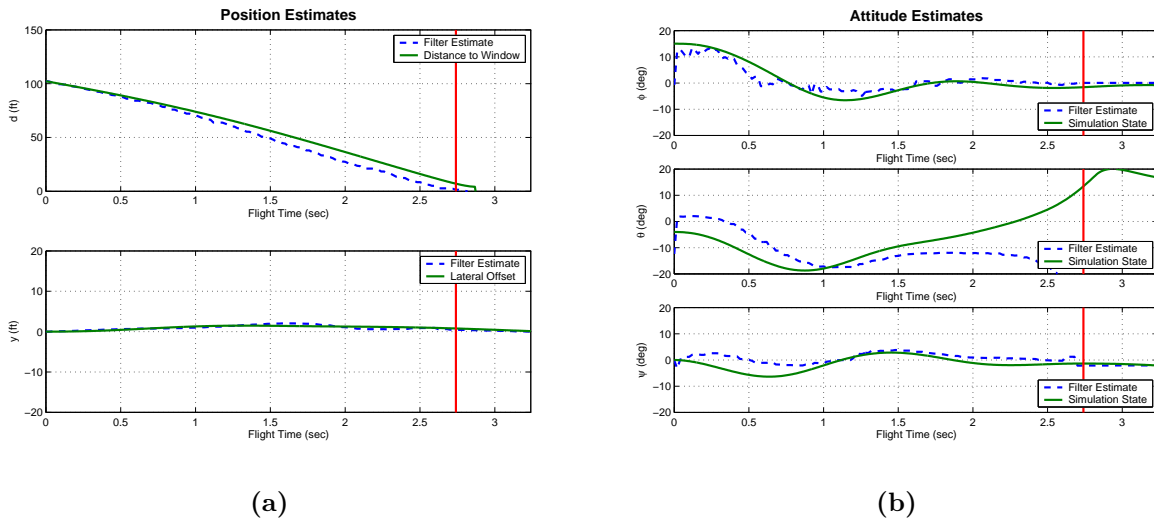


Figure 10. Comparison between the output of the EKF and the recorded state of the dynamic model. (a) Position estimate, (b) Attitude estimate

The velocity estimate and crab angle estimates are shown in Figure 11-a. In this scenario, there isn't a crosswind and the EKF correctly estimates the crab angle as near zero. Figure 11-b shows a comparison of the actuator commands that were issued by the controller and those that were received by the dynamic model on the third computer. The differences between the two signals are representative of how consistently the control commands are delivered and how well the actuators can follow them.

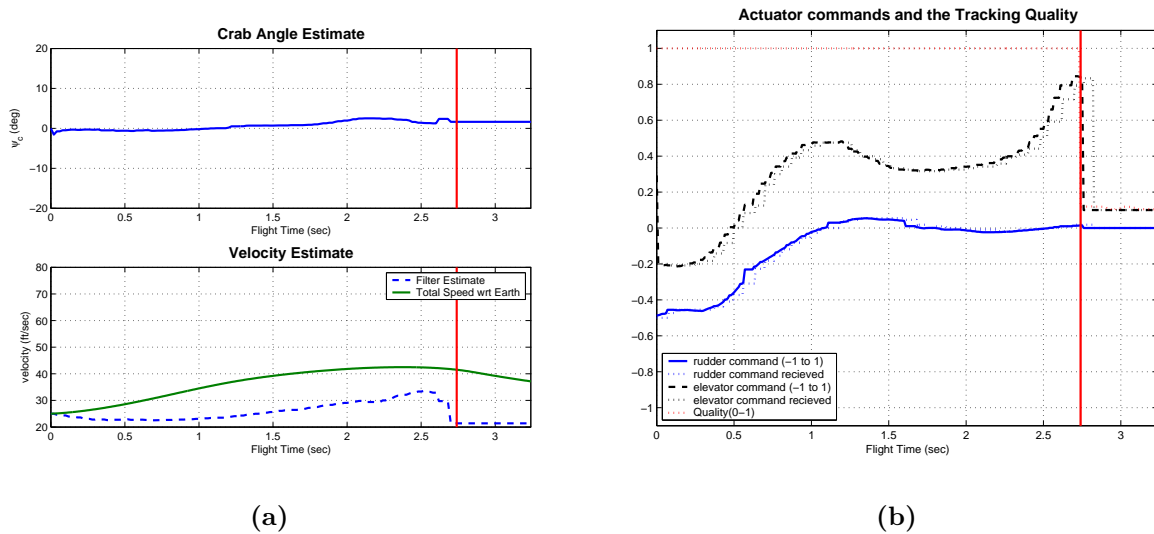


Figure 11. (a) Crab angle and Velocity Comparison, (b) Control Commands (-1 to 1)

B. Flight Test Results

Each of the flight tests performed with the glider has led to insight into making the system more robust, and making improvements in the performance of the ground station and the flight control system. Figure 12 shows a photo taken during the flight depicted in the following plots and onboard images with the results of the image processor. During this flight the image processor tracked the window all the way to the end of the flight, and the glider hit the center of the window. The glider was launched by hand, from an altitude

of 15 ft above the window, and the window was approximately 100 ft from the launch site.

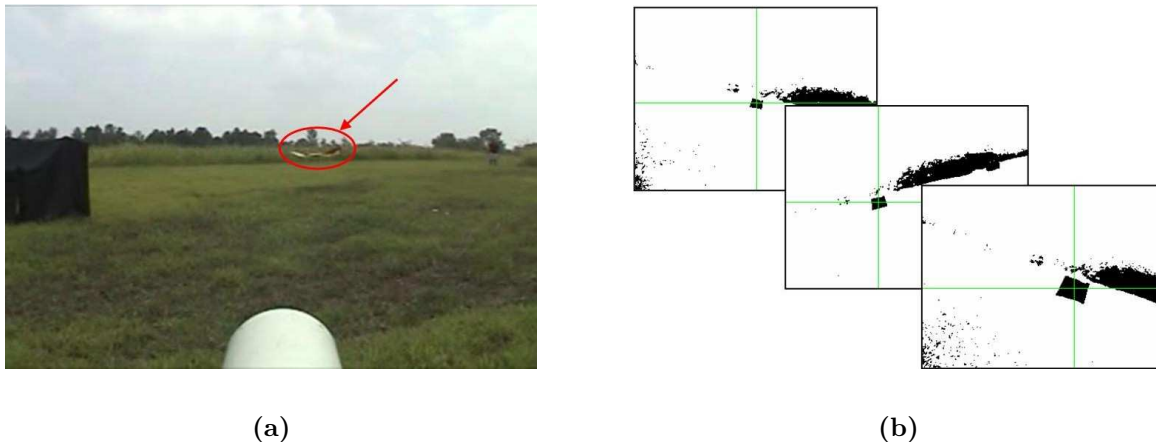


Figure 12. Flight Test of the Glider (a) Still image taken during flight, (b) Onboard Images with image processing results

Figure 13-a shows the distance estimate from the flight. The top plot show the distance to the window. The second plot shows the lateral offset estimate, and the final plot shows the size of the window in the image as the flight progresses. This shows that by the end of the flight the window take up the entire camera image.

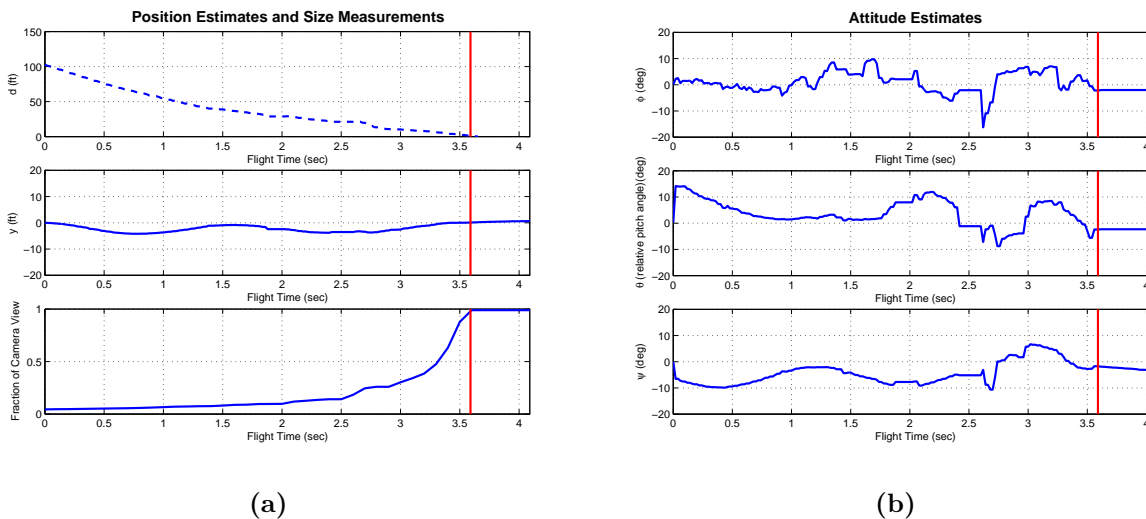


Figure 13. Flight Test of the Glider (a) Position Estimate, (b) Attitude Estimate

The attitude estimates are shown in Figure 13-b. The estimates are rather noisy. Once again, this is probably due to the fact that the output of the image processor from real images has significantly more noise than with the synthetic images. However, this does show that the results of the image processor were playing a significant roll in updating the EKF. One of the drawbacks of actual flight, is that there is nothing to validate the output of the navigation filter, except the impact location and recorded video from other cameras.

Figure 14 shows the control commands for the flight. It is clear from this plot that the controller was trying to center navigate towards the window.

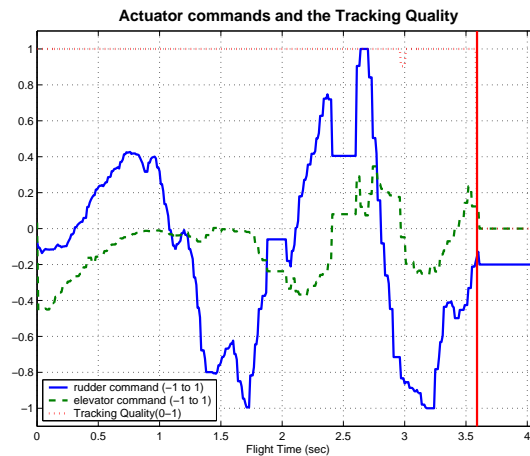


Figure 14. Actuator Commands during Flight of the Glider

VII. Conclusion and Future Work

Simulation tests showed that vision-only aircraft flight control is effective, even with very few image analysis measurements and a simple glider model. These ideas have been successfully implemented and tested in real flight scenarios using a glider. The flight tests to this point have proven that controlling an aircraft is possible using the vision system described in this paper.

A. Future Work

This topic has by no means been completely explored. There are still a lot of new scenarios to fly in simulation and in actual flights. In the future, these algorithms will be flown on a larger aircraft, where reference data (from GPS) is available to verify the accuracy of the position estimation from the EKF. This will enable researchers to explore new settings for the EKF and attempt to improve its performance in real flight scenarios.

VIII. Acknowledgements

This work was supported in part by the Air Force Office of Scientific Research through the Multi-University Research Initiative in Active-Vision Control Systems and by Lockheed Martin Corporation. The authors would also like to acknowledge the contributions of Thomas Apker, Tobias Breithaupt, Henrik Christophersen, Christophe De Wagter, Jeong Hur, Suresh Kannan, Adrian Koller, Kriangsiri Malasri, Wayne Pickell, and Nimrod Rooz, who made this research possible.

References

- ¹Shirai, Y. and Inoue, H., "Guiding a robot by visual feedback in assembling tasks," *Pattern Recognition*, Vol. 5, 1973, pp. 99–108.
- ²Hill, J. and Park, W. T., "Real-time control of a robot with a mobile camera," *9th Int. Symp. Indus. Robots*, Washington, D.C., March 1979, pp. 233–246.
- ³Corke, P. I. and Good, M. C., "Dynamic effects in visual closed-loop systems," *IEEE Trans. on Robotics and Automation*, Vol. 12, October 1996, pp. 671–683.
- ⁴Mikawa, Y. M. M., Maru, N., and Miyazaki, F., "Visual servoing for non-holonomic mobile robots," *IEEE Conference on Intelligent Robots and Systems*, Vol. 2, September 1994, pp. 1133–1140.
- ⁵Doehler, H.-U., "Robust Position Estimation Using Images from an Uncalibrated Camera," *Digital Avionics Systems Conference*, No. 2003-9D2 in DASC, Indianapolis, IN, August 2003.
- ⁶Kerr, J. R., Luk, C. H., Hammerstrom, D., and Pavel, M., "Neural Net Based Processor for Robust, High-Integrity Multisensor & Synthetic Vision Fusion," *Digital Avionics Systems Conference*, No. DASC-2003-9D4, Indianapolis, IN, August 2003.

⁷Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., "Vision Based Navigation for an Unmanned Air Vehicle," *IEEE Conference on Robotics and Automation*, May 2001, pp. 1757–1765.

⁸Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Vision-based autonomous landing of an unmanned aerial vehicle," *IEEE Conference on Robotics and Automation*, Vol. 3, May 2002, pp. 2799–2804.

⁹Zhang, H. and Ostrowski, J., "Visual servoing with dynamics: Control of an unmanned blimp," *IEEE Conference on Robotics and Automation*, Vol. 1, May 1999, pp. 618–623.

¹⁰Wagter, C. D., Proctor, A. A., and Johnson, E. N., "Vision-Only Aircraft Flight Control," *AIAA Digital Avionics Conference*, No. 8B2, Indianapolis, IN, October 2003.

¹¹Johnson, E. and Schrage, D., "The Georgia Tech Unmanned Aerial Research Vehicle: GTMax," *AIAA Guidance, Navigation and Control Conference*, Austin, TX, August 2003.

¹²Viola, P. and Jones, M., "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, December 2001, pp. 1511–1518.