

Control Architecture for a UAV-Mounted Pan/Tilt/Roll Camera Gimbal

Ole C. Jakobsen* and Eric N. Johnson†
Georgia Institute of Technology, Atlanta, GA, 30332-0150

This paper presents the architecture of a pan/tilt/roll camera control system implemented on the Georgia Tech's UAV research helicopter, the GTMax. The controller has currently three operating modes available: it can keep the camera at a fixed angle with respect to the helicopter, make the camera point in the direction of the helicopter velocity vector, or track a specific location. The camera is mounted in a large, but relatively light gimbal. Each axis is driven by a modified servo, and optical encoders measure the gimbal orientation. A PID controller with anti-windup and derivative filtering was designed in Simulink based on simple models of the servos and later implemented on the real system. A discussion of results obtained from Hardware-In-The-Loop tests and flight tests is given at the end of the paper.

I. Introduction

UAVs may be used for a variety of civilian and military purposes, of which rescue operations in dangerous areas and surveillance may be mentioned as obvious examples. A lot of these tasks require live video recording and researchers at Georgia Tech are currently working on imaging algorithms making it possible to track moving targets, achieve visual feedback in flight, and to automate landings on moving platforms. Accurate pointing of the onboard camera is vital in order to achieve these tasks, and this paper outlines the architecture of the camera control system implemented on the Georgia Tech UAV lab's research helicopter, the GTMax. (For related work on camera control systems mounted on UAVs, consider for example Ref. 1 and Ref. 2.)

The camera is placed in a gimbal delivered by Neural Robotics and mounted at the front end of the GTMax. Three modified servos from Hitec serve as motors, rotating the camera about each axis and taking velocity commands as inputs. The system is designed so that there are no limitations in the rotation angle about the pan axis, while roll is limited to angles between -100 deg and 100 deg and tilt to -90 deg and 90 deg. Three encoders with indexing delivered by US Digital are used to read each angle. The velocity commands are given at a rate of 50 Hz, as are the angle readings from the encoders. Altogether, these hardware components allow flexible and accurate controlling and results in a relatively light and inexpensive system.

The high level controller can operate in three different modes. Its outputs are desired pan, tilt and roll angles for the gimbal in all modes, but the angles are calculated differently in the three cases. The controller may order the camera to point at a specific location, and the computed angles are in that case based on the given location and GTMax position and attitude as estimated by the integrated navigation system³. Ground station personnel may enter a target position into the system manually and send it to GTMax, but the ground station may also receive position information about a stationary or moving target (like another aircraft) and forward it to GTMax without any human intervention. The second controller mode makes the camera point in the direction of the helicopter velocity vector, while the third and simplest mode keeps the camera at a fixed angle relative to the helicopter.

The low level part of the control is implemented as a simple PID controller; it outputs velocity commands to each motor and angle measurements are fed back from the encoders. The load experienced by each motor differs considerably. In particular, the load on the pan axis motor is larger than that on the other two and also varies in time due to gimbal motion. A simple model partly based on measurement data was therefore made of each motor and implemented in Simulink. Several simulations were then performed to study system behavior for various controller

* Graduate Student, Aerospace Engineering, 270 Ferst Drive.

† Lockheed Martin Assistant Professor of Avionics Integration, Aerospace Engineering, 270 Ferst Drive, Member AIAA.

parameters and find suitable controller gains. The results obtained with the low level controller algorithm implemented in C and tested on the real system are thoroughly discussed in this paper.

II. Hardware

The following sections will give a brief description of the hardware used.

A. GTMax

Georgia Tech's research UAV, the GTMax, is based on the Yamaha R-max airframe. This Yamaha helicopter has a rotor radius of 5.1 ft, weighs about 128 lbs empty, has a payload capability of about 66 lbs, and can fly non-stop for more than an hour. Baseline software, simulation tools and a modular avionics system are other key components of the GTMax. The avionics part currently consists of two Intel processors, an IMU, differential GPS, sonar altimeter, 3-axis magnetometer, and wireless data-links^{3,4}. GTMax, including the camera gimbal discussed in this paper, is shown in Fig. 1.



Figure 1. GTMax with camera gimbal

B. Gimbal

The camera gimbal used is delivered by Neural-Robotics. It is relatively light and inexpensive and is shown in the figure below. Cables needed for three servos, three encoders and the video camera, a total of 24 cables, are fed to the gimbal through a slip ring. The slip ring may introduce cross-talk problems due to the small distance between each cable, but has the advantage that it allows for an unlimited number of rotations about the pan axis. In theory there are no physical constraints on the roll rotation either, but the video cables may wind up, and it was decided to limit its rotation to angles between -100 deg and 100 deg. The servo driving the roll rotation is mounted on the outside of the inner gimbal and creates physical limits for the tilt rotation of about -135 deg and 135 deg. The software limits for this axis are set to -90 deg and 90 deg.

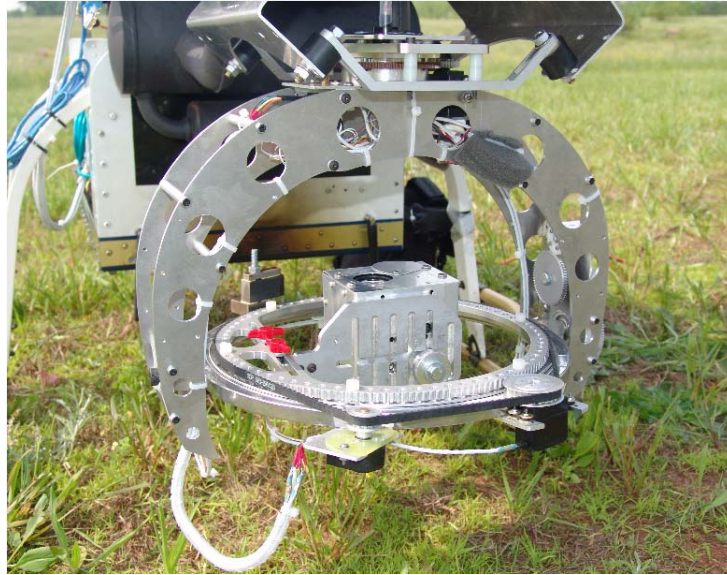


Figure 2. Camera gimbal.

C. Servos

A JR 8311 servo is mounted on each axis. These servos offer a torque of 130 oz/in, a maximal speed of 0.18 s/60 deg and are modified to accept velocity commands instead of position commands as inputs. The speed/command relationship of each servo was found to be linear within a quite limited area, and the motor inputs are therefore also limited from software. The gimbal/motor gear ratio for the pan axis is 77/34, and a speed of about 80 deg/s is achievable. The tilt axis has a gimbal/motor gear ratio of 768/119, resulting in a maximum speed of approximately 40 deg/s. The corresponding numbers for roll are 179/24 and 29 deg/s.



Figure 3. Modified servo

D. Encoders

An H1 encoder from US Digital is used to measure the angle of each gimbal axis. The encoders output signals on three channels, A, B, and Index. A and B provide quadrature codes used to determine turn direction and calculate encoder shaft position, while the Index channel outputs one pulse for each encoder revolution and hence enables pseudo absolute positioning at power up. These particular encoders have a resolution of 1024 CPR and X1 decoding is used. The gimbal/encoder gear ratios for pan, tilt and roll are 7, 6, and 179/11, respectively, so one ends up with overall resolutions of 0.050 deg, 0.058 deg, and 0.022 deg.



Figure 4. Optical encoder

E. Flight Control System 20 (FCS20)

The FCS20 is a small integrated guidance, navigation and control system recently developed at Georgia Institute of Technology. It has previously been tested on GTMax and an 11 inch ducted fan UAV called Helispy and has proven capable of automatic takeoff, landing, hover, and aggressive maneuvering⁴. The responsibilities of the FCS20 onboard GTMax are currently limited to recording helicopter rotor rpm and to run the low level gimbal controller. The main components are, as shown in Fig. 5, a large Field Programmable Gate Array (FPGA), a Digital Signal Processor (DSP), SDRAM and Flash memory.

Signals from channels A and B on each encoder are detected by three 10 bit counters that output a value between 0 and 1023, indicating encoder shaft position relative to the index. The index pulses sent by the encoders are detected by three 6 bit counters, each outputting a value between 0 and 63, indicating the number of full encoder revolutions since start up. When the system is powered up, each axis will rotate until the first index is detected. Whereas all gimbal/encoder gear ratios are greater than 1, there will be a position ambiguity at this point, and there is no guarantee that the gimbal will be at the desired initial (zero) position. The errors can only be an integer number of indexes, however, and can therefore easily be corrected for from ground station. This approach, although requiring some manual work, was preferred to the extra sensors and additional cabling that would be needed to automate the initialization process.

In addition to encoder measurements, the FPGA soft core CPU (nios) also receives a message from the main flight computer (through the serial ports) containing reference commands for the low level controller. These messages are received by the nios, forwarded to the DSP and finally sent back to the nios where they are read and used in the controller algorithm. Note that the DSP does not do any processing with regards to this controller.

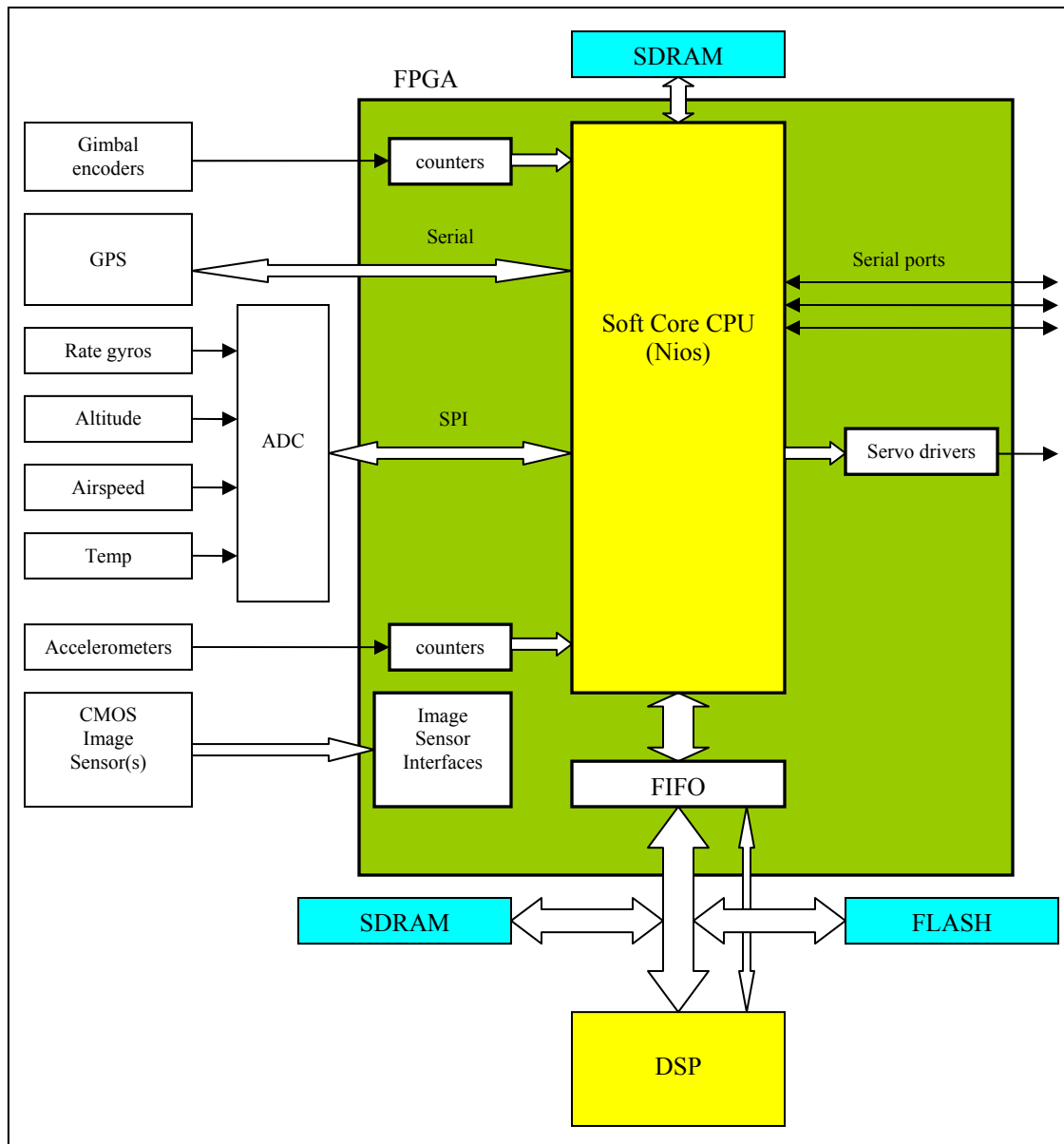


Figure 5. FCS20 hardware⁴

The encoders and servos mounted on the gimbal operate in a rather noisy environment due to onboard electronics and communication between helicopter and ground station. The Ethernet antenna is currently mounted close to the gimbal and thus poses a particular challenge regarding measurement noise. Early testing showed that the noise could result in erroneous detections of the Index signal and therefore also more or less random resets of the 10 bit counter. Such errors would cause serious problems if happening during flight, and a reset-functionality is therefore implemented to make it possible to repeat the gimbal initialization procedure while in air.

III. Software

The following sections summarize the controller software.

A. Low Level Controller

The low level controller reads each encoder value and calculates a new PWM command for each servo at a rate of 50Hz. It receives three desired angles from the high level controller and also three inputs representing index offsets. These offsets are, as mentioned above, only used if the gimbal starts out (at power-up) in the wrong position,

or if something unexpected happens while in air. Note also that controller calculated gimbal angles are recorded by the ground station for after-flight analysis.

A simple PID controller (position form) was implemented for each of the servos. The initial designs for the controllers were based on plant models of the form:

$$G(s) = Ke^{-Ts} \left(\frac{1 + \tau_n s}{1 + \tau_d s} \right) \quad (1)$$

where K has unit rad/rad/s, and T , τ_n and τ_d are given in seconds. Simulations in Simulink suggested that anti-windup would be needed for the integral part of the controllers due to the actuator saturation, and later testing also showed that additional filtering was necessary for the derivative signals. The implemented digital controllers were hence given by the following equations:

$$E(k) = r(k) - x(k) \quad (2)$$

$$I(k) = I(k-1) + \frac{h}{T_i} E(k-1) - \frac{h}{T_i K_p} (u(k-1) - v(k-1)) \quad (3)$$

$$D(k) = \frac{1}{T_f + h} \left(\frac{T_f}{K_p} D(k-1) - T_d (x(k) - x(k-1)) \right) \quad (4)$$

$$u(k) = K_p (E(k) + I(k) + D(k)) \quad (5)$$

where we have that $r(k)$ is the reference command in radians, $x(k)$ is the measured position in radians, $E(k)$ is the position error, T_i is the integral time constant, T_t is a tracking time constant for the integral reset, $I(k)$ represents the integrated error in radians, $u(k)$ and $v(k)$ are the actual (saturated) and desired controller outputs, respectively, both given in radians per second. T_d is the derivative time, T_f is the filter time constant, $D(k)$ represents the derivative part of the measured position and is in radians, K_p is a gain with unit rad/s/rad, and finally, h represents the time step size. Figure 6 shows one simulation result for each axis, while a block diagram of the simulated system is given in Fig 7.

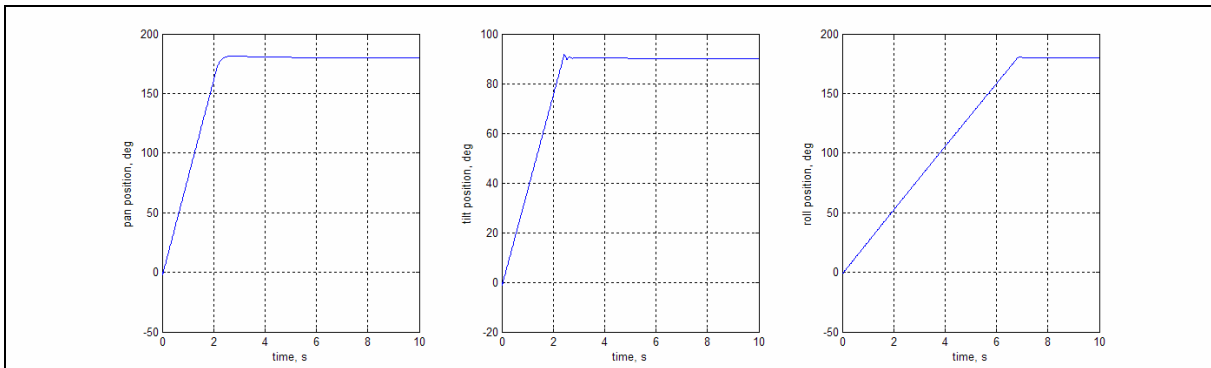


Figure 6. Simulated step responses

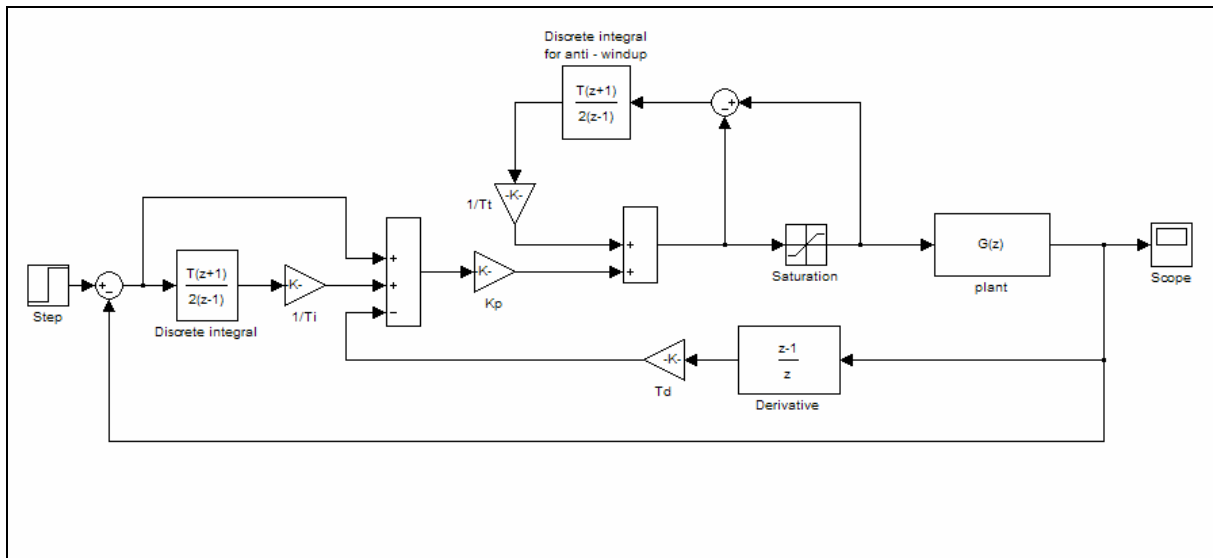


Figure 7. Block diagram of simulated system

Overshoot was not considered acceptable in general because it would degrade the video quality, but in order to achieve a reasonably fast system, some overshoot was accepted for large step-inputs. Initial testing showed that step-inputs would result in some chattering by the controller, so a second order trajectory is now calculated based on the step-input command and used as a reference to achieve a smoother response.

B. High Level Controller

The high level controller has currently three operating modes available: it can output angle commands to the low level controller making the camera track a given position, track the helicopter velocity vector, or stay at a given, fixed angle with respect to the helicopter. Each mode serves a highly different purpose and a great variety of test conditions are possible. Note also that ground station personnel can change mode at any time during flight.

The least complex functionality is clearly to make the camera stay at a fixed angle with respect to the helicopter, and in this mode the high level controller merely passes three angles specified by the ground station on to the low level controller – no calculations are necessary.

The upper left part of Fig. 8 shows the command window where ground personnel can enter angle commands manually, change mode and specify index offsets, while the other three “scene” windows show the user interface needed for the position tracking mode. In particular, the lower left window shows a map of the test area with the planned flight trajectory indicated as a brown line and the helicopter as a yellow circle. A user can make the camera point at a specific location by simply marking that position in the window with a mouse click – creating, as may be seen in the figure, a small, green circle at that point. This position to track is then input to the high level controller, which uses it and the estimated helicopter position to calculate angle commands for the low level controller. The lower right window in Fig. 8 shows the same situation only at a shorter distance, while the upper right window indicates the camera-view. Observe that the green circle is in the center of the camera view as should be expected. Note also that the ground station may receive position information about a stationary or moving target and send the position to track to GTMax automatically, that is; without human intervention.

To make the camera point along the helicopter velocity vector requires no user action other than changing the “mode” input in the command window. The high level controller will then use information about the estimated helicopter velocity to calculate angle commands.

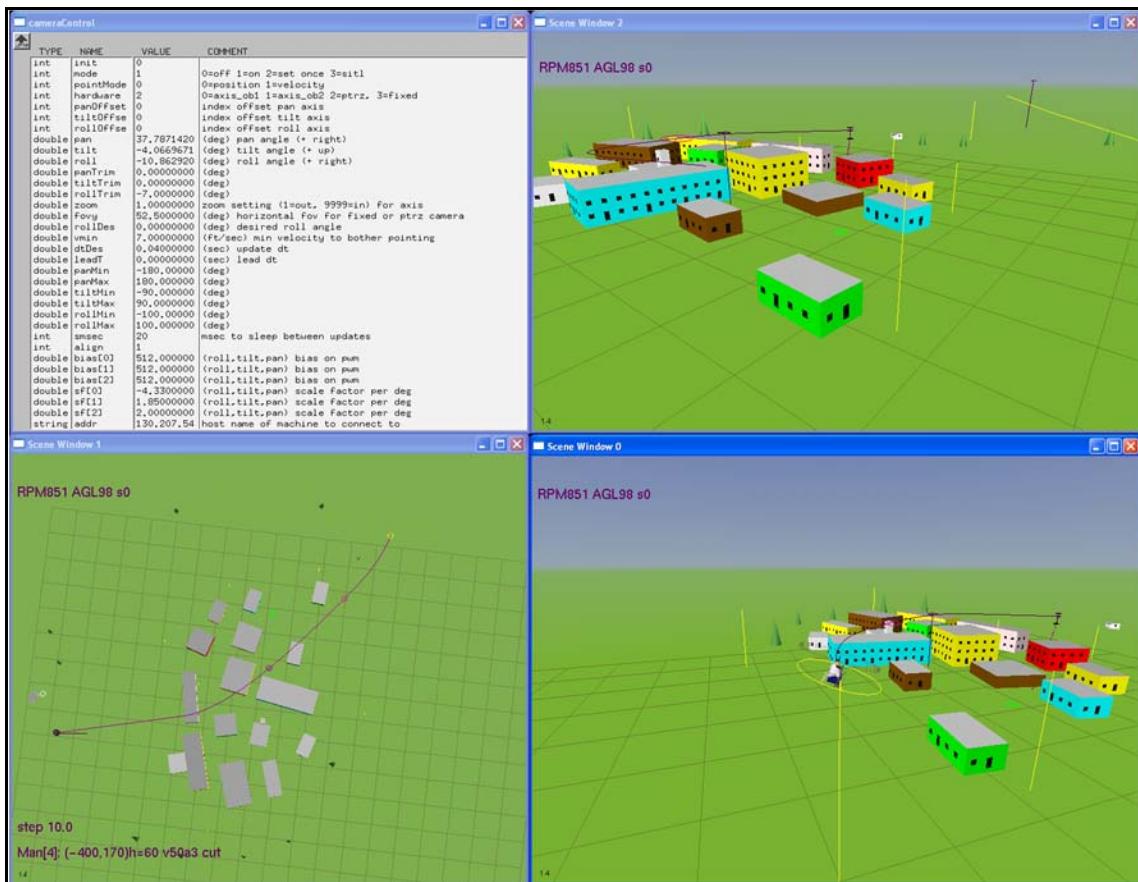


Figure 8. User interface

IV. Results

The GTMax system allows for Software-In-The-Loop (SITL) and Hardware-In-The-Loop (HITL) testing as well as flight testing⁵. In order to perform HITL tests for this project, the main flight computer was connected via a serial link to a laptop running the flight simulation. Figure 9 shows a scene window from one of the simulations. The helicopter, the trajectory “flown“ by the helicopter, and the point at which the camera should be pointing are indicated as in Fig. 8. The results of this particular HITL test are shown in Fig. 10. One can see that the controller performs quite well even when the commands from the high level controller are changing rapidly, but there will be some tracking error due to the simplicity of the low level controller.

The flight test results shown in Fig. 11 confirm the observations made during the HITL tests, that is; the system has good regulation capabilities, but the simple low level PID controller is not able to perfectly track a constantly changing reference command.

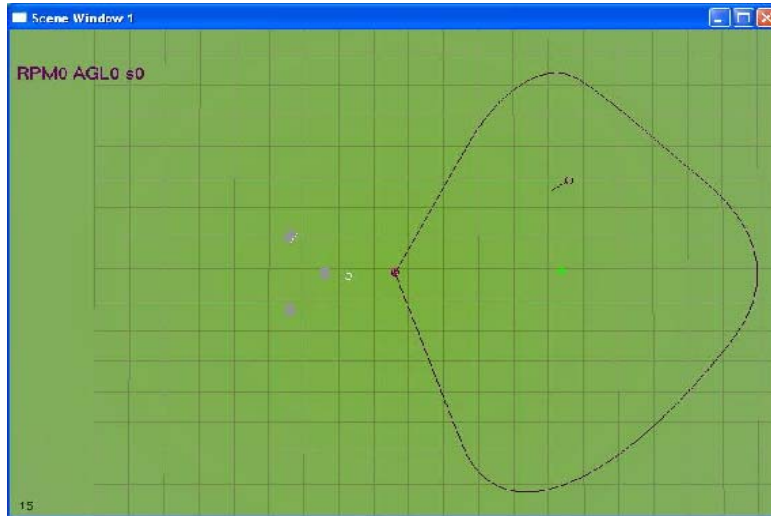


Figure 9. Scene window from HITL test

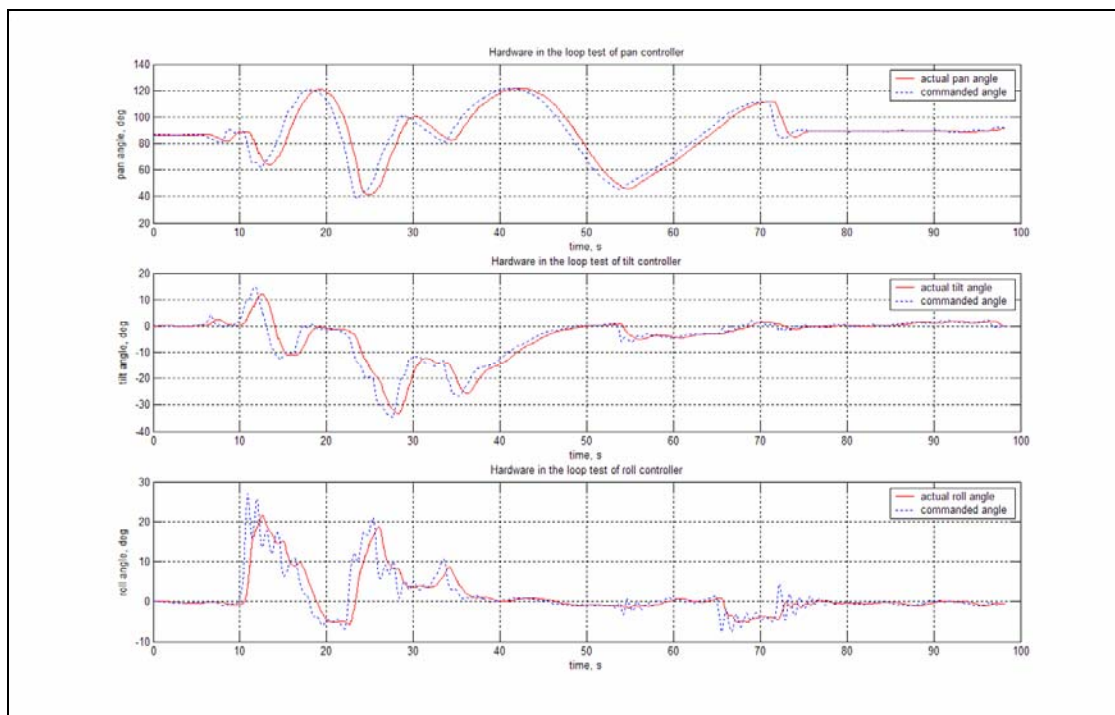


Figure 10. HITL test results for the pan, tilt and roll controller

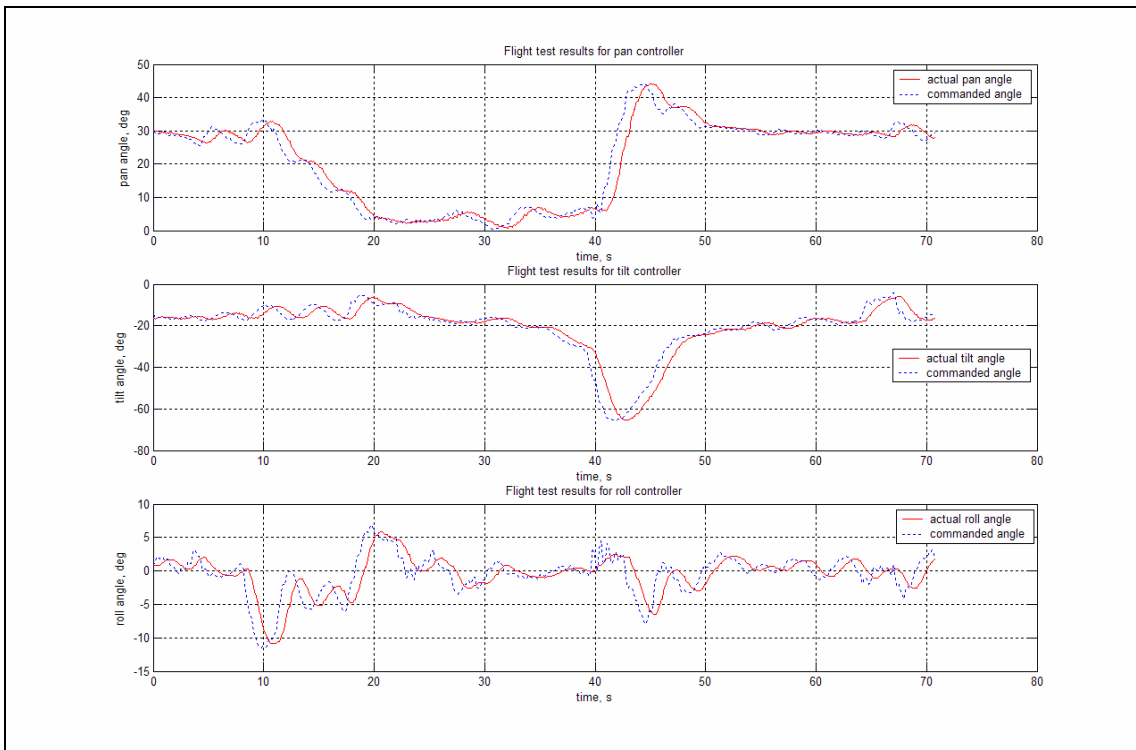


Figure 11. Flight test results for the pan, tilt and roll controller

V. Conclusion and Future Work

The pan/tilt/roll camera control architecture implemented on Georgia Tech's research UAV, GTMax, has been outlined. The controller's performance was demonstrated through both HITL tests and flight tests. In both cases, the controller proved good regulation capabilities and did also track constantly changing reference commands quite well. Future work will seek to find out how the encoder measurement noise can be further reduced, and a more sophisticated low level controller might be considered in order to achieve better tracking results.

Acknowledgments

The authors would like to acknowledge Phillip Jones, Allen Wu, Claus Christmann, Stephen Card, Alison Proctor, Adrian Koller, Henrik Christophersen and Alex Moodie for their contributions to this research.

References

- ¹Sharp, C.S, Shakernia, O., Sastry, S., "A Vision System for Landing an Unmanned Aerial Vehicle", *International Conference on Robotics and Automation*, Seoul, Korea, May 2001.
- ²Stolle, S., Rysdyk, R., "Flight Path Following Guidance for Unmanned Air Vehicles with Pan-Tilt Camera for Target Observation," *Digital Avionics Systems Conference-2003*, April 2003.
- ³Johnson, E.N. and Schrage, D.P., "The Georgia Tech Unmanned Aerial Research Vehicle: GTMax,," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003
- ⁴Christophersen, H.B., Pickell, W.J., Koller, A.A., Kannan, S.K., and Johnson, E.N., "Small Adaptive Flight Control Systems for UAVs using FPGA/DSP Technology," *Proceedings of the AIAA Unmanned Unlimited Technical Conference, Workshop, and Exhibit*, 2004.
- ⁵Johnson, E.N., Schrage, D.P., Prasad, J.V.R., and Vachtsevanos, G.J., "UAV Flight Test Programs at Georgia Tech," *Proceedings of the AIAA "Unmanned Unlimited" Technical Conference, Workshop, and Exhibit*, 2004.