AAS 03-0XX

American Astronautical Society

Design, Development and Testing of the Miniature Autonomous Extravehicular Robotic Camera (Mini AERCam) Guidance, Navigation, and Control System

J. Wagenknecht, NASA Johnson Space Center
S. Fredrickson, NASA Johnson Space Center
T. Manning, Titan Systems Corporation
B. Jones, Titan Systems Corporation

26th ANNUAL AAS GUIDANCE AND CONTROL CONFERENCE

February 5-9, 2003
Breckenridge, Colorado

Sponsored by
Rocky Mountain Section

American Astronautical Society
Rocky Mountain Section

AAS Publications Office, P.O. Box 28130 - San Diego, California 92198

DESIGN, DEVELOPMENT AND TESTING OF THE MINIATURE AUTONOMOUS EXTRAVEHICULAR ROBOTIC CAMERA (MINI AERCAM) GUIDANCE, NAVIGATION, AND CONTROL SYSTEM

J. Wagenknecht, NASA Johnson Space Center
S. Fredrickson, NASA Johnson Space Center
T. Manning, Titan Systems Corporation
B. Jones, Titan Systems Corporation

Engineers at NASA Johnson Space Center have designed, developed, and tested a nanosatellite-class free-flyer intended for future external inspection and remote viewing of human spaceflight activities. The technology demonstration system, known as the Miniature Autonomous Extravehicular Robotic Camera (Mini AERCam), has been integrated into the approximate form and function of a flight system. The primary focus has been to develop a system capable of providing external views of the International Space Station. The Mini AERCam system is spherical-shaped and less than eight inches in diameter. It has a full suite of guidance, navigation, and control hardware and software, and is equipped with two digital video cameras and a high resolution still image camera. The vehicle is designed for either remotely piloted operations or supervised autonomous operations. Tests have been performed in both a six degree-of-freedom closed-loop orbital simulation and on an air-bearing table. The Mini AERCam system can also be used as a test platform for evaluating algorithms and relative navigation for autonomous proximity operations and docking around the Space Shuttle Orbiter or the ISS.

INTRODUCTION

Engineers at NASA Johnson Space Center have designed, developed, and tested a nanosatellite-class free-flyer intended for future external inspection and remote viewing of human spaceflight activities. The technology demonstration system, known as the Miniature Autonomous Extravehicular Robotic Camera (Mini AERCam), has been integrated into the approximate form and function of a flight system.

Scope

This paper provides an overview of the Mini AERCam vehicle design, and a detailed description of the GN&C system design, development, and testing.

Background

The predecessor to Mini AERCam, called AERCam "Sprint," was flight tested on STS-87 in December 1997. As shown in Figure 1, Sprint was released in the Space Shuttle Orbiter Payload Bay by astronaut Winston Scott during an extravehicular activity (EVA). Astronaut Steve Lindsay controlled the free-flyer using a combination rotational and translational hand controller from inside the Orbiter aft cockpit. Lindsay was able to see Sprint at all times through the Orbiter windows, and also monitored the video on an internal display. The flight test on STS-87 lasted about an hour and fifteen minutes and was highly successful.

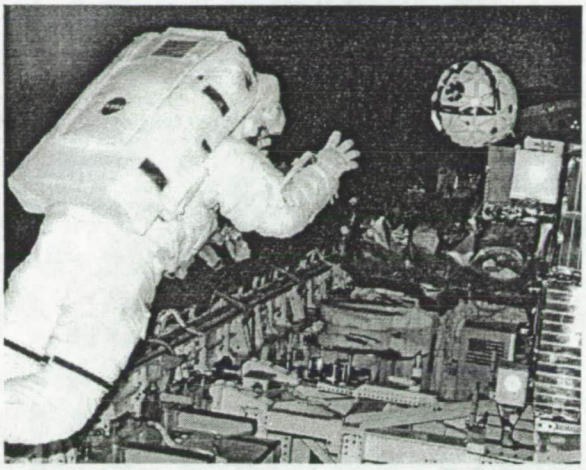

Figure 1 AERCam Sprint Release on STS-87

Sprint was approximately fourteen inches in diameter and weighed about thirty-five pounds. The propulsion system consisted of twelve cold gas thrusters with nitrogen as the propellant. Sprint had angular rate gyros for providing angular rate feedback and automatic attitude hold. The video camera could be switched between two focal lengths for a regular or "telephoto" view. A light was available for illumination. Sprint had a soft covering for impact protection.

From 1997 to 1998, JSC engineers designed, developed, and integrated a ground-test version of AERCam for the "AERCam Integrated Ground Demonstration (IGD)." The AERCam IGD, demonstrated on an indoor air-bearing table, was intended to demonstrate the availability of advanced technologies that could be used to enhance AERCam. The GN&C technology advancements were a significant departure from the limited GN&C capabilities on Sprint. Then in 1998, a crew evaluation was conducted in JSC's Virtual Reality Laboratory for the purpose of evaluating the suitability of Sprint for use as an operational vehicle outside the International Space Station. The study by nine astronaut participants yielded a set of recommendations for how to improve safety, situational awareness, and handling qualities for a future version of AERCam.

In 2000 Mini AERCam development began with the goal of reducing free-flyer size while simultaneously adding capabilities that would contribute to overall system controllability, reliability, and utility. For situational awareness, it was desired to add an additional camera for an orthogonal view, as well as relative navigation so that crewmembers would know where the free-flyer is with respect to the very large and complex ISS structure. To improve the handling qualities and reduce crew workload, engineers would incorporate automatic position hold and point-to-point maneuvering. By 2002, a fully functional prototype has been designed, developed, integrated and tested as described below in the following sections.

Challenges for the Mini AERCam Design Team

The Mini AERCam design team was challenged by NASA management to design a version of Sprint with advanced capabilities that would be "about the size of a grapefruit." The team was also directed to select advanced technologies that were at the near edge of the technology envelope, such that if they were not available yet, they would be mature in time for a flight experiment within a few years. Finally, the design team was asked to design the ground demonstration system without making significant compromises such that each subsystem was only one step away from a flight configuration. These challenges were met as described in the following sections.

MINI AERCAM VEHICLE SYSTEMS

The Mini AERCam free-flyer is spherical-shaped and approximately 7.5 inches in diameter (closer to the size of a cantaloupe than a grapefruit). Figure 2 shows a size comparison between AERCam Sprint and Mini AERCam. Mini AERCam achieves its 'nanosatellite' size by a combination of dense packaging and miniaturized components.

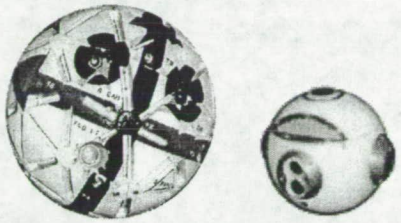

Figure 2 AERCam Sprint and Mini AERCam

Figure 3 highlights some of the external features of the Mini AERCam prototype. Two cameras are aligned with the +X direction of the vehicle. One camera provides NTSC-quality color video, and the other camera can be used for high-resolution still images, when selected. A third color video camera is positioned in the +Y direction for an orthogonal view.

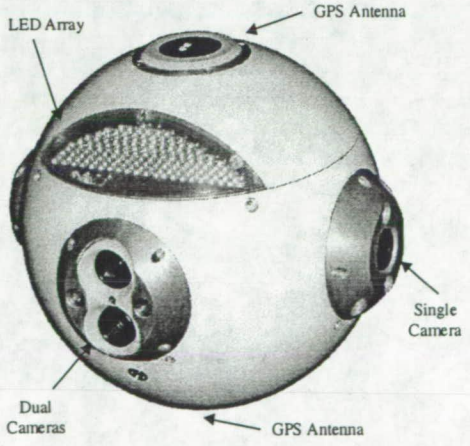

Figure 3 Mini AERCam External Features

The Mini AERCam prototype is designed to have two custom GPS antennas, one on top and one on the bottom. An LED array provides illumination. A communications antenna, not visible in the figure above, is near the top GPS antenna.

As shown in Figure 4, the vehicle is designed with a central ring that houses the power and propulsion system. The batteries are lithium-ion and provide at least four hours of operational time. The propulsion system is designed for cold-gas xenon, which packs more densely than nitrogen, but is compatible with low-cost nitrogen in the current ground test configuration. Attitude and position control are achieved with the use of twelve thrusters, distributed across four thruster pods around the central ring. The batteries are rechargeable and a port is provided for refueling.

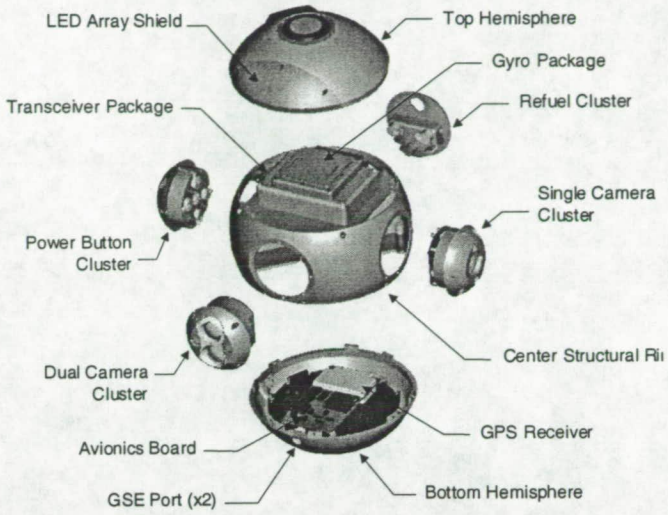

Figure 4 Mini AERCam Exploded View

The Mini AERCam navigation system includes a GPS receiver for position and velocity determination, and Draper Micro-Electro-Mechanical System (MEMS) gyros for angular rate sensing. In order to compute precise relative navigation using GPS, the parent vehicle (e.g. the ISS) must also be equipped with a GPS receiver. The current Mini AERCam GPS receiver is a Thales Navigation Systems (formerly Ashtech) G-12 HDMA GPS receiver.

The Mini AERCam avionics system includes an avionics processor board, a video compression and imaging system, wireless ethernet (802.11b standard) for communications, and LED illuminators. The avionics processor board is a 22-layer board.. The board houses a PowerPC 740 which runs at 266 MHz, with 64 Mbytes of RAM. The board includes a field programmable gate array (FPGA) which implements most of the hardware interface functions. Data buses include RS-232, I^2C for instrumentation, RS-422, and LVDS. The vision subsytem consists of three complementary metal oxide semiconductor (CMOS) imagers, an onboard video compression system, and an off-board video decompression and display system. The video compression system performs wavelet compression in both hardware and custom software. The video is transmitted as part of a single multiplexed data stream. A summary of Mini AERCam specifications is provided below in Table 1.

Size	7.5 inch diameter sphere
Mass	5 kg
Power (avg)	<20 Watts (>4 hours run time)
Attitude and	12 thrusters mounted around central

position control	ring
Propellant	Xenon gas (about 40 ft/s delta-V) (nitrogen for ground test vehicle)
Attitude rate sensors	MEMS gyros
Relative position sensors	Precise relative GPS
Avionics processor	Custom built board with PowerPC 740
Instrumentation	I^2C network
Communications	Wireless Ethernet (802.11b)
Lighting	LED array
Video	2 NTSC-quality video cameras (+X and +Y) and a 1 megapixel inspection camera for still images

Table 1 Mini AERCam Specifications

The vehicle is designed for either remotely piloted operations or supervised autonomous operations. A set of rotational and translational hand controllers are used to operate the vehicle manually. The operator may also input commands through the Control Station, which also displays system status and engineering data. Multiple Mini AERCam camera views are available to the operator. For the ISS application, a Situational Awareness display provides a high fidelity graphical representation of the Mini AERCam location with respect to the ISS. Figure 5 shows the hand controllers and selected displays.

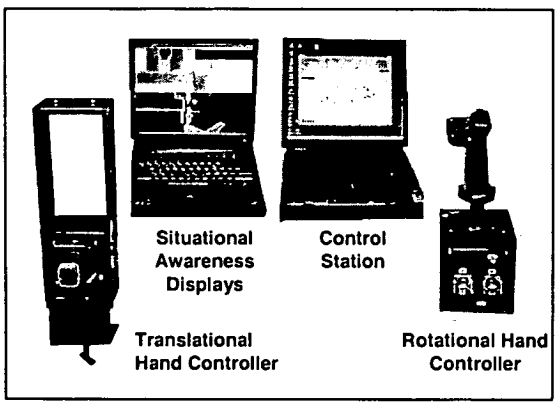

Figure 5 Operator Controls and Displays

NAVIGATION DESIGN

Translational Navigation

The navigation subsystem is the cornerstone of the Mini AERCam GN&C capabilities. AERCam Sprint's capabilities were significantly limited by the absence of any translational navigation systems to provide relative navigation information that could be used for operator situational awareness and on-board autonomous functionality. Thus, precise translational navigation was a high priority for providing advanced capabilities on Mini AERCam.

The Mini AERCam translational navigation system uses precise relative GPS for estimating relative position and velocity between the free-flyer and parent vehicle (in this case, the ISS). Precise relative GPS is based on differencing the pseudorange and carrier phase measurements between two vehicles' GPS receivers. Throughout this paper, the GPS receiver that would be on-board the ISS is referred to as the "ISS GPS," but this should not be confused with the current ISS GPS, which is a Trimble Force 19 housed in a Honeywell Space Integrated GPS/INS (SIGI). The ISS SIGI's GPS receiver is a 6-channel GPS receiver and the Force 19 is nearly obsolete. For our implementation and ground testing, we selected a different COTS GPS receiver. It is envisioned that a receiver like the G-12 would be integrated with either an existing ISS GPS antenna or with a new one, in order to provide an acceptable level of functionality for precise relative GPS.

INSERT ILLUSTRATION OF NAV SYSTEM

The GPS measurements are processed and filtered using JSC's Flexible Inertial/Relative Estimator (FIRE) software, which includes a Kalman filter consisting of 12 + N states, where N is the number of GPS satellite vehicles (SV's) in common between both GPS receivers. FIRE was originally based on the Generic Kalman Filter (GKF) algorithms which are used for many NASA navigation applications. FIRE was initiated in 1998 in cooperation with the United States Air Force Research Laboratory (AFRL), as part of a risk mitigation activity for the XSS-11 microsatellite program. FIRE development was continued when Mini AERCam began in 2000. FIRE is more than just the Kalman filter, and also includes the real-time data collection and synchronization software. The FIRE navigation software that runs on the free-flyer is hand-coded in ANSI C and runs under VxWorks.

On-board the free-flyer, FIRE samples the GPS receiver data and processes it. Three Kalman filters run simultaneously on-board the free-flyer: one filter estimates the WGS-84 position and velocity of the parent vehicle based on its GPS data; another filter estimates the WGS-84 position and velocity of the free-flyer based on its GPS data; and the third filter estimates the position and velocity of the free-flyer in its local-vertical-local-horizontal (LVLH) frame. FIRE provides the position and velocity estimates at 1 Hz. It also generates detailed performance information for display and recording for post-flight analysis.

Another version of FIRE also runs on the Control Station laptop computer, under Microsoft Windows 98. The Control Station version samples data from the parent vehicle GPS receiver, processes the data, and writes it to a Common Data Area on the laptop for packaging into the communications data structure. The data is transmitted to the free-flyer across the wireless Ethernet interface.

Attitude Navigation

The Mini AERCam attitude navigation system uses angular rate measurements from the Draper MEMS gyros to estimate inertial attitude and attitude rate for the free-flyer. The MEMS gyro package outputs data at a rate of 300 Hz, but the data are only sampled at 25 Hz. The body axis roll, pitch, and yaw attitude rate measurements are converted to quaternions for use in computations. In order to estimate relative attitude of the free-flyer with respect to the ISS, the inertial attitude is transformed to other reference frames, including the Mini AERCam LVLH-to-body frame, J2000 Inertial-to-Mini AERCam LVLH, Body-to-J2000, Body-to-ISS LVLH, J2000-to-ISS LVLH, and Body-to-an arbitrary inertial frame.

The quaternions are integrated using a first-order Runge-Kutta integrator. Higher-order integrators were considered, but are not used in order to reduce error accumulation over time. Before the quaternions are passed to the integrators, measurement noise is reduced by applying a single-pole filter to the rate gyro outputs. This includes using feed-forward estimation based on

previously commanded thruster on-times and the previous body rates to produce an estimated body rate. A weighted average is then computed with the measurements and the estimated values, and output to the integrators. Due to relatively high noise levels on the MEMS rate gyros, the estimated body rates are weighed more heavily than the measured rates. A body rate with respect to the vehicle LVLH is also produced by correcting for the orbital rotation rate, which is computed using the inertial states from the translation navigation and transformed to the body frame. After the quaternions have been updated, the Euler angles are extrapolated in pitch-yaw-roll order and output for pilot telemetry.

The attitude navigation uses a slightly different process when producing the quaternion used for inertial attitude hold. Instead of always holding with respect to some established inertial frame, like J2000 or M50, the body-to-arbitrary inertial transformation is reset with each attitude hold command, and then continues propagation like all other quaternions. This ensures that errors accumulated over time are dropped from the estimate used for inertial attitude holds.

The free-flyer attitude can also be updated from an external source, when such information is available. As described in a later section, the free-flyer attitude is initialized from an external source during air-bearing table testing.

Analysis of the AERCam attitude navigation versus truth can be performed in real-time, or by post-processing recorded data. Since the inertial attitude is typically reset several times during a test, the body attitude with respect to the vehicle LVLH frame is used to determine navigation accuracy. Currently, the attitude performance is an attitude drift rate of about 1 degree per minute in the yaw axis, with much better performance in the pitch and roll axes. External tests have shown that noise on the yaw gyro is an order of magnitude greater than the other two gyros, and this axis is where we experience the most drift.

GUIDANCE AND CONTROL DESIGN

Attitude Guidance

The Attitude Guidance subsystem on MiniAERCam is designed to accommodate either manual (crew-piloted) or autonomous attitude and attitude rate commands. The Attitude Guidance system accepts inputs from either source and computes errors that are then passed on to the Attitude Control subsystem. The current modes supported in the Attitude Guidance are described in Table 2 below.

Attitude Guidance Mode	Description
Drift	No attitude or attitude rate guidance
Decel	Null attitude rates
Limit Cycle	Maintain attitude to within specified deadbands
Attitude Maneuver	Execute commanded attitude
Discrete Rate	Execute commanded attitude rate

Table 2 Translational Guidance Modes

Except for free drift, which is not reference-frame-dependent, all modes can be executed with respect to either inertial or relative (LVLH) reference frames. The pilot makes the selection on the Control Station before the mode is executed. Free drift mode is used when the AERCam is completely under the pilot's control and no autonomous capabilities are desired. Decel mode is only used to null the angular rates to within a predetermined threshold. In Decel mode, the attitude errors are set to zero. After Decel mode, the free-flyer automatically enters Limit Cycle mode and holds the attitude in the desired frame until the pilot issues another command. The

Attitude Maneuver mode is used to issue a command to rotate the vehicle to a pilot-selected attitude. Once the vehicle arrives at the commanded attitude it enters Limit Cycle mode. Discrete rate mode increments the commanded body rate by a specified discrete amount with each rotation command issued by the pilot.

INCLUDE ATTITUDE GUIDANCE MODING STATE TRANSITION DIAGRAM HERE

Except for the Attitude Maneuver mode, all attitude guidance modes are initiated by issuing a commanded body rate to propagate the commanded attitude quaternion. This is required because the coupled nature of the rotation axes requires that the commanded quaternion use information from all axes, even if they are each in different modes.

When the pilot commands an attitude maneuver, the current estimated attitude quaternion is compared to the commanded attitude, and the guidance generates a quaternion describing the transformation from the current body frame to the commanded body frame. This quaternion is then reduced to an Eigenvalue and an Eigenvector to determine the angle and axis around which the vehicle must be rotated. The angle is compared to the maximum allowed magnitude of the maneuver rate to determine the time required to complete the maneuver. The commanded attitude rate is found by projecting the maximum maneuver rate into the body frame using the Eigenaxis. This rate is then used to propagate the commanded quaternion that is initialized to the attitude state when the maneuver command is issued. When the time to completion has been reached, the commanded attitude rate is dropped to zero and the guidance enters limit cycle mode using the pilot commanded attitude. This method effectively generates a "bang-bang" solution and only requires additional thruster commands to make small corrections during the maneuver, thereby minimizing fuel requirements.

The errors that are output to the control logic are generated by comparing the quaternion which describes the transformation from the reference frame to the commanded body frame with the current state. This generates a new current body-to-commanded body frame transformation, and Euler angles are extrapolated to generate the attitude errors. Using quaternion multiplication also eliminates singularities and angle rollovers at 180 degrees. Attitude rate errors are determined by subtracting the actual body rates from the commanded body rates.

Minimal analysis is needed to determine the performance of the attitude guidance. Tests have verified that guidance performs as instructed, and that all commanded values and errors are correct.

Attitude Control System

The Attitude Control System (ACS) uses a simple phase plane design for each independent body axis. It is designed to work in such a manner as to center the vehicle in the phase plane rapidly. The phase plane contains switching lines as seen below.

PHASE PLANE PICTURE GOES HERE

Because of the granularity of the thruster valve drive electronics (1ms) and the FPGA interface we can send a time command to the thrusters to stay on until a desired switching line is crossed. This is accomplished by estimating the minimum acceleration that the thrusters will provide. The ACS does not send a command longer than one 25 ms control cycle. Otherwise, the system would support the maximum burn duration and not be able to be overridden. The unique nature of this thruster command system allows the ACS to run at a rate that is slower than the thruster granularity but still retains all the advantages of the granularity. Now the system can run at a frequency necessary to control it under disturbance forces. Since, in space, a vehicle that is essentially a rigid sphere does not have much in the way of disturbance torques, the ACS could

probably run at a slower rate. Until there is a better understanding of translational cross-coupling on the flight vehicle, however, we will continue to support a 25 Hz ACS cycle.

The most important characteristic of the phase plane control is the trim burn. A "perform_trim_burn" flag is set whenever any phase plane command is issued. A trim burn is commanded whenever the rate axis is crossed inside of the phase plane and the perform_trim_burn flag is on. The trim burn continues to be commanded until the rate errors are zero. Once the trim burn is accomplished, the perform_trim_burn flag is set to 0. While the Pitch perform_trim_burn flag is on, the flight software refuses to support any Z body translation commands, either from the Hand Controller or Automatic Translational Controller. It will, however, queue Z hand controller commands and support them once the vehicle Pitch is centered (i.e. the trim burn is complete.). This logic is necessary to support a problem with Z/Pitch contention encountered as a result of our thruster configuration.

By waiting until the Pitch axis is centered in the phase plane, a more pure Z pulse is obtained. The figure below shows one way of handling Z/Pitch contention where Z thruster pulses are queued until the Pitch thruster pulses stop. The problem with this approach is that the vehicle is still not centered in Pitch when a Z pulse is allowed. Cross-coupling from a Z translation will cause a machine gunning effect in the Z/Pitch control.

SHOW MACHINE GUNNING PHASE PLANE FIGURE

Translational Guidance

The Translational Guidance subsystem on Mini AERCam is designed to accommodate either manual (crew-piloted) or autonomous commands. The Translational Guidance system accepts inputs from either source and computes errors that are then passed on to the Translational Control subsystem. The current modes supported in the Translational Guidance are described in Table 3 below.

Translational Guidance Mode	Description
Drift	No translational commands
Brake	Null relative velocity between free-flyer and parent vehicle
Station Keep	Maintain relative position between free-flyer and parent vehicle to within specified deadbands
Commanded Maneuver	Execute commanded relative position or velocity maneuver

Table 3 Attitude Guidance Modes

In Drift mode all of the position control errors are zeroed. Brake mode is used in support of a station-keeping command from the pilot. When the pilot commands the vehicle to stop, it first enters Brake mode until its velocity slows to within a specified limit. In Brake mode the position errors are zeroed and the velocity errors are set equal to the most recent FIRE LVLH velocity estimate. The control will attempt to null the velocity. Once the velocity is within a specified limit, the Translational Guidance enters Station Keeping mode. In this mode the position errors are the difference between the LVLH position estimate upon entering Station Keeping mode and the current LVLH position estimate. The velocity errors are set to the current LVLH velocity estimate. The Brake and Station Keeping algorithms effectively slow the vehicle and stop it in a short amount of time instead of slowing it and returning to the position where the pilot commanded it to

stop. If the pilot wishes to return to the position where the stop was commanded, he may fly it there himself using hand controller commands or a Commanded Maneuver.

INCLUDE TRANSLATIONAL GUIDANCE PICTURE HERE

The Commanded (Translational) Maneuver mode allows the pilot to select any point in the vicinity of the target vehicle and have the Mini AERCam maneuver to that point with a specified LVLH velocity. A targeting algorithm computes the commanded position as a function of time over the length of the trajectory. As the maneuver continues, the Translational Guidance computes the current commanded position as determined by the targeting logic and subtracts the estimated state to obtain errors. Those errors are then shipped to the Translational Control. Once a maneuver is complete, the vehicle enters Station Keeping mode. As a future enhancement, GN&C will provide the ability to end a maneuver with a commanded velocity. This will provide a useful interface for waypoint navigation with multiple waypoints. This type of commanded maneuver will be completely independent of the Rotational Guidance Mode.

INCLUDE A PICTURE OF ACTUAL vs. COMMANDED LVLH X FOR OUR DEMO MANEUVER.

If the Flight Control system at any time receives a THC command it will immediately mode the Translational Guidance into Drift mode. There are two THC modes currently supported on MiniAERCam. In ***pulse mode***, a predetermined pulse duration is commanded every time a hand controller is deflected. The second mode is ***Accel mode***. In Accel mode, the commanded thrusters remain actively firing as long as the hand controller is deflected. Accel mode in translation is important because control authority in translation is much (how much?) less than in rotation. This mode allows the pilot to force the vehicle to thrust as long as he deems comfortable with the vehicle response. The current implementation of Pulse mode also forces a translation pulse to be three times as long as a rotation pulse.

Translational Control

The Translational Control subsystem accepts as inputs the errors from the Translational Guidance subsystem and the LVLH-to-Body quaternion estimate from the Rotational Navigation subsystem. The translational errors are then converted from the LVLH frame to the body frame.

There are two methods of Translational Control. The first uses a phase plane controller similar to that on the Space Shuttle Orbiter. Like the Attitude Control System controller, it uses the anticipated thruster accelerations to force the vehicle back toward the center of the phase plane in a fuel-efficient manner.

INSERT TRANSLATIONAL PHASE PLANE

The second method is using a Linear Quadratic Regulator (LQR) controller developed for Mini AERCam by Charles Stark Draper Laboratories. This controller takes into account the expected relative motion environment using Clohessy-Wiltshire equations, and controls the vehicle within an ellipse for fuel optimality.

Z/Pitch Control

Z/Pitch control is handled differently from other control modes due to a contention problem between Z and pitch, as mentioned earlier. This is due to the way that the thrusters are configured on the vehicle. The types of conflict and the management strategy for each are described in Table 4.

Z Command Source	Conflicts with Pitch Command Source	Conflict Management Strategy
Z hand controller	Pitch hand controller	Support pitch first, and then support Z
Z hand controller	Pitch phase plane	Support pitch phase plane until the trim burn is complete, and then support Z hand controller
Z automatic translation	Pitch hand controller	Support pitch hand controller commands first and ignore Z translation commands. It is assumed that the translation controller errors will grow and that it will respond accordingly when the vehicle finishes its Pitch maneuver
Z automatic translation	Pitch automatic attitude	Support pitch automatic translation commands and ignore Z automatic translation commands. It is assumed that the translation controller errors will grow and that it will respond accordingly when the vehicle finishes its Pitch maneuver

Table 4 Z/Pitch Contention Description

Z/PITCH CONTROL BLOCK DIAGRAM GOES HERE

Jet Selection

The inputs to the jet selection logic for MiniAERCam are the commanded thrust duration times (in milliseconds) for all six degrees of freedom. Each thrust duration input is either positive or negative depending on the direction of thrust. X/Yaw contention is addressed by turning off one thruster to allow for both simultaneously. Y/Roll contention is addressed with in the same fashion. The only potential issue with this approach is that the desired amount of thrust may not be achieved because instead of an anticipated number of milliseconds from two thrusters, only one thruster would fire. However, the MiniAERCam jet selection logic will queue the unsupported thrust duration and direction and thus be able to provide thrust for the required number of milliseconds with two thrusters. The Z/Pitch contention problem is handled by only supporting Pitch in the jet select logic. It is assume that this is properly taken care of in the Z/Pitch control logic.

INCLUDE JET SELECTION LOGIC TABLES

GN&C SOFTWARE DEVELOPMENT

Software Architecture

The GN&C software architecture is designed to be compatible with the overall Mini AERCam software architecture. The Mini AERCam system executive, which schedules all tasks, runs at a high rate of XX Hz and a low rate of YY Hz. The backbone of the Mini AERCam software is the Common Data Area (CDA), which manages all data that is passed between software subsystems. All sensor data is queued in the CDA, including GPS and rate gyro data. All GN&C telemetry also passes through the CDA. However, the GN&C software issues thruster commands directly to the FPGA and the commands do not pass through the CDA. The GN&C software runs at a high rate of XX Hz and a low rate of YY Hz.

ILLUSTRATION OF THE CDA

Several commercial-off-the-shelf software packages are used on the Mini AERCam project. The operating system running on the free-flyer avionics processor is WindRiver's VxWorks real-time operating system. The Real Time Innovations Network Data Delivery System (NDDS) software package is used to format communication data. The compiler is WindRiver's Diab compiler. The GN&C team uses the MathWorks MATRIXx development environment for GN&C algorithm development and autocode generation. MathWorks Matlab is also used for data processing and plotting. National Instruments' LabWindows is used for real-time data displays.

The MiniAERCam is controlled remotely from the Control Station, which is a laptop computer running Microsoft Windows 98. This computer has RS-232 serial interfaces to two hand controllers (separate rotational and translational) and to the "parent vehicle" GPS receiver. Data is transmitted between the Control Station and the free-flyer via wireless ethernet.

Software Development Process

At the beginning of the project, the Mini AERCam project management set a series of incremental milestones for demonstrations of capabilities, as shown in Table 5. These milestones were achieved by using a spiral development process, similar to that used on many other major programs, including X-38 and the ISS GN&C.

Orbital Simulation Demonstration Milestones	Air-Bearing Table Demonstration Milestones
1: Manual 6-DOF Control	1: Manual 3-DOF Control
2: Manual 6-DOF Control with Automatic Attitude Hold (Inertial and Relative)	2: Manual 3-DOF Control with Automatic Attitude Hold
3: Relative Navigation with GPS Hardware in the Loop	3: Relative Navigation with Indoor Navigation System
4: Automatic Translation Hold (ATH) with GPS Hardware in the Loop	4: Automatic Translation Hold (ATH) with Indoor Navigation System
5: Automatic Point to Point Maneuvering in 6-DOF	5: Automatic Point to Point Maneuvering in 3-DOF

Table 5 Incremental Demonstration Milestones

Mini AERCam GN&C software includes both hand-coded and autocoded C code. The translational navigation software, including FIRE, is entirely hand-coded in C. The guidance and control software is designed in MATRIXx's SystemBuild, which is a graphical programming environment, and then C code is auto-generated using MATRIXx. MATRIXx autocode has a legacy within the AERCam program. MATRIXx autocode was first used on AERCam Sprint; the entire motion control system was built and generated using MATRIXx[1]. The auto-generated C code was then integrated with the hand-coded system executive and other routines. Similarly, the guidance and control software for the AERCam IGD vehicle was also generated using MATRIXx. The navigation software for the AERCam IGD was mostly hand-coded. The biggest difference between the software for AERCam Sprint and the AERCam IGD vehicles is that the AERCam Sprint processor (a simple microcontroller, the Intel 80C196K) was so limiting that the autocode had to be post-processed by hand to improve efficiency. This was not the case for the AERCam IGD processor, which was a 166 MHz Pentium processor.

MATRIXx SystemBuild made it easy to reuse much of the guidance and control logic from the AERCam IGD for Mini AERCam. The most significant change is that the AERCam IGD had a set of two-dimensional algorithms that were designed for a terrestrial air-bearing table, and the Mini AERCam algorithms are designed to work on-orbit in three dimensions.

The core G&C software for Mini AERCam is primarily developed in MATRIXx SystemBuild. First, unit test cases are developed for each algorithm, and then the system is designed. After design, the unit test cases are executed. Once an algorithm passes the unit tests,

the software is autocoded in C and compiled for the development board test bed, which houses a real-time embedded processor similar to the actual vehicle Avionics processor board but without the avionics interfaces. The development board test bed software (under VxWorks) is then run through a battery of functional tests. After this it is integrated with the FIRE software and ported to the Simulation Avionics Processor Board Test Bed where it is integrated with the Mini AERCam system executive software. In this arrangement the software is then run through a series of tests to insure that the demonstration mission will run. Finally, the GN&C software is ported to the Air Bearing Test Bed where it is again run through a series of maneuvers designed for demonstration. These maneuvers test out the hand controller interface as well as the position and attitude control logic.

Most of the hardware and software interface software written for FIRE was first written as part of the XSS-11 project. Each version of FIRE is initially developed to run in UNIX under the Solaris Operating system, with a serial interface to the GPS receivers. It is then ported to the development board testbed. The only essential change in FIRE from Solaris to the VxWorks environment is at the application level. Device interfaces are configured within their respective components at compile time. For Mini AERCam, the free-flyer interface from the GPS receivers to FIRE is not directly serial, but through the Common Data Area. The system executive on Mini AERCam provides the interface between the CDA and FIRE. The only significant rework for porting to another Operating System was done by the system executive programmer to migrate the ISS (parent vehicle) portion of FIRE to Microsoft Windows. As for the G&C system, each individual interface component of FIRE has unit test cases that are run under both Solaris and VxWorks. Current work consists of developing a larger functional test suite and a set of unit test cases for the Filter proper. FIRE also has the ability to read GPS receiver data from a file for off-line testing. FIRE executes as a separate thread on the free flyer and as such can be run stand-alone.

(this paragraph needs to be restructured. Topic sentence and supporting statements and all of that. It's junky.)

TEST FACILITIES

Test Facility Description

The Mini AERCam GN&C testing has four levels. As described in the Software Development section, the first level of testing is simple unit testing either in SystemBuild or using hand written C code. All of the software is originally developed using some variety of UNIX as the development platform, including Solaris and Apple workstations. Once developed and tested on the development platform, the software is ported to the development board system. Once it passes a series of tests in that environment it is then ported to the Orbital Simulation. Again it must pass a series of tests in that environment before it is finally installed on the Air Bearing Table system. Each of these test facilities are described in more detail in the following sections.

Development Board System

INSERT AN ILLUSTRATION OF THE DEVELOPMENT BOARD CONFIG

The development board system includes the development board, which is a real-time embedded system similar to the avionics processor board, and also a real-time on-orbit dynamics simulation of the MiniAERCam. The simulation was developed using the Trick dynamic simulation environment, which is used for many other simulations throughout the Johnson Space Center. Trick contains all of the environment models needed, as well as a database of selectable sensor and effector models. The Trick simulation runs on a Sun Ultra 80 workstation. The development board is a flight-like PPC 750 Single-Board VME computer running in a separate VME chassis.

The communication to the development board from the simulation workstation is achieved with a PCI/VME bridge. The flight software runs asynchronously on the development board computer, sampling simulated rate gyro data over the bridge and writing thruster software commands to the bridge. A set of hand controllers is connected to the system through a serial port on the Ultra 80. These hand controller commands are then passed to the Flight Control System through the PCI/VME Bridge. The Trick simulation also connects to a JSC Virtual Reality Laboratory (VRL) graphics package called DOUG (Dynamic Onboard Ubiquitous Graphics) over the Local Area Network to display the position of the Mini AERCam in real-time as it maneuvers around the ISS. This system also shows us a simulated view as it would be seen from the Mini AERCam cameras. The current configuration of the graphics package contains a very simple Mini AERCam graphical model and a full model of the International Space Station. The development board system allows the GN&C team to perform a variety of guidance and control functional tests before attempting to use the Simulation Avionics Processor Board (Sim APB). The development board system also provides a way for GN&C to perform tests even when the Sim APB is required for other types of testing or for upgrades. Finally, in the development board configuration, all data can be logged and plotted using the Trick Logging and Data Processing utilities.

Orbital Simulation Testbed

The orbital simulation testbed is designed for testing guidance, navigation, and control capabilities in a high fidelity simulation with much of the Mini AERCam avionics in the loop. In the orbital simulation, the free-flyer can perform rotational and translational maneuvers, including automatic attitude hold, automatic translation hold, and point-to-point maneuvering. The MEMS gyros and thrusters are emulated in hardware, and a high-fidelity GPS signal generator provides realistic Radio Frequency (RF) signals to the GPS receivers. This is illustrated in Figure 6.

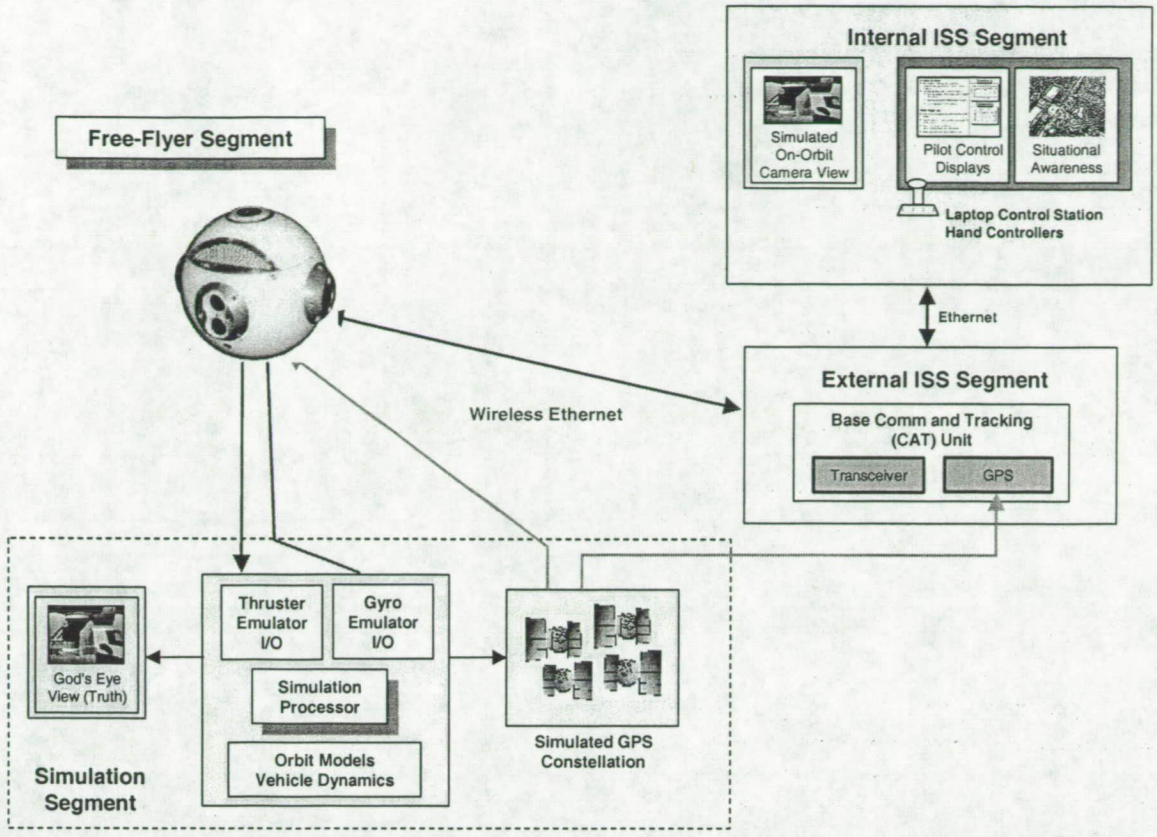

Figure 6 Orbital Simulation Testbed Configuration

In the orbital simulation, the free-flyer is controlled using separate rotational and translational hand controllers and a control station. In place of live video, the images as seen from the free-flyer are simulated with high-fidelity graphics. Engineering data and state information are displayed on the control station. In addition, a "God's Eye View" is generated for the Situational Awareness display, which provides the operator with a graphical view of the Mini AERCam location with respect to ISS structure. Figure 7 shows an example scene from the high-fidelity graphics that are used for the orbital simulation.

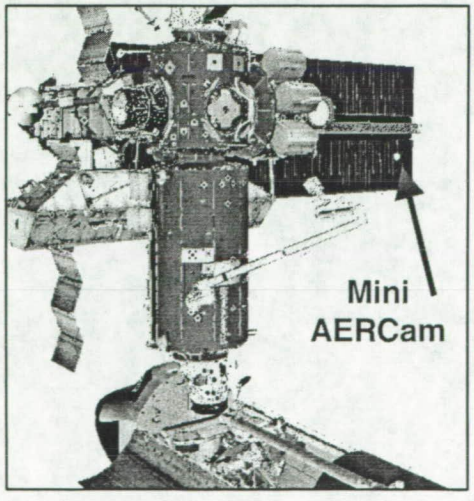

Figure 7 "God's Eye View" for the Orbital Simulation

The simulation avionics processor board system uses a full fidelity avionics processor like that in the free-flyer prototype. Using a duplicate avionics processor board allows orbital simulation and air-bearing table tests to be conducted in parallel with no resource conflicts. The orbital simulation configuration is similar to the development board testbed, but with more fidelity. In the orbital simulation, the Trick simulation also is connected to a Spirent GPS signal generator through a UDP socket connection to an interface PC. The Trick simulation sends vehicle state information for both the free-flyer and ISS to the interface PC. The interface PC packages the data in a format that the GPS signal generator can interpret, and ships the information over an IEEE-488 connection. The signal generator then uses that information to generate the RF signals, which are transmitted via RF cables to both the Mini AERCam and the ISS GPS receivers. The Mini AERCam SimAPB is connected serially to the G-12 GPS receiver. The control station laptop provides an interface to the ISS GPS receiver and packages the data to send across the wireless Ethernet interface to the free-flyer, along with operator commands.

In this test bed, data is logged and displayed using LabWindows in real-time by tapping into the telemetry stream. Since the communications for the MiniAERCam is Ethernet, the engineering data can be displayed on a separate computer on the local area network. The logged data is also post-processed and analyzed using Matlab.

INCLUDE A PLOT OF COMMANDED POSITION AND ACTUAL POSITION VS. TIME FROM LABWINDOWS

Air Bearing Table Testbed

An air-bearing table test facility is used for hardware and avionics testing. A special sled, illustrated in Figure 8, produces a cushion of air to float the free-flyer on an air-bearing table, which simulates microgravity. The vehicle is then able to translate in two directions and rotate about one axis, using the vehicle's own thrusters, propelled with nitrogen for the ground demonstration.

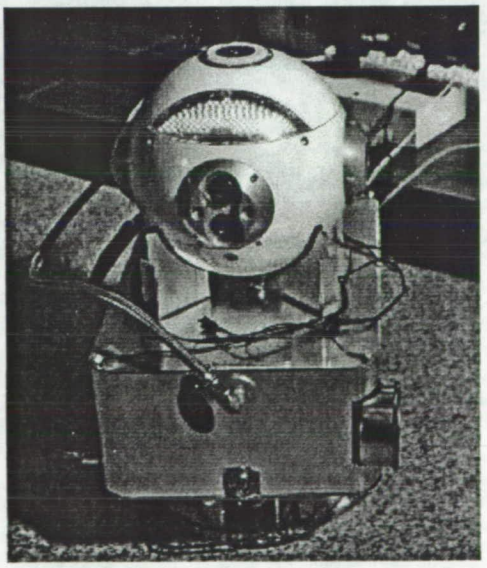

Figure 8 Mini AERCam on Air-Bearing Table Sled

The free-flyer is controlled on the air-bearing table using separate rotational and translational hand controllers and a control station. Live video and images from the free-flyer are provided on two video displays. Engineering data and state information are displayed on the control station. On the air-bearing table, the free-flyer can perform rotational and translational maneuvers, including automatic attitude hold, automatic translation hold, and point-to-point maneuvering.

Since GPS constellation signals are not available inside the laboratory, a commercial-off-the-shelf (COTS) image-based navigation system called Visualeyez is used for position determination. Visualeyez provides indoor relative navigation using an LED tracking system. The LEDs are mounted on the free-flyer sled. The state (position only) observations from the Visualeyez system are transmitted to the free-flyer on the air-bearing table from off-board, and then are processed through a version of FIRE that is similar to the orbital version but configured for indoor, terrestrial operations. When executed on the table, FIRE utilizes a separate set of filter tunings, and absolute positions are not calculated. The GN&C main software uses a separate set of initial values for parameters on the table for the change in conditions, i.e. no orbital rate, mass property changes, etc. Rotation navigation does not compute some transformations, and a different translation controller is used for the table.

The air bearing test bed for the Mini-AERCam provides a fully integrated three degree-of-freedom test platform for hardware/software interaction and other hardware systems in a controlled environment. It utilizes all hardware in the loop, including the rate gyros, thrusters, video systems, and communications. It also allows for demonstrating, profiling and debugging these systems.

Testing a fully integrated vehicle also allows the developers to test hardware interfaces, thermal effects, and other hardware systems. One primary example is testing the video system, for evaluating how the picture quality is affected by bandwidth reduction and loss of communication.

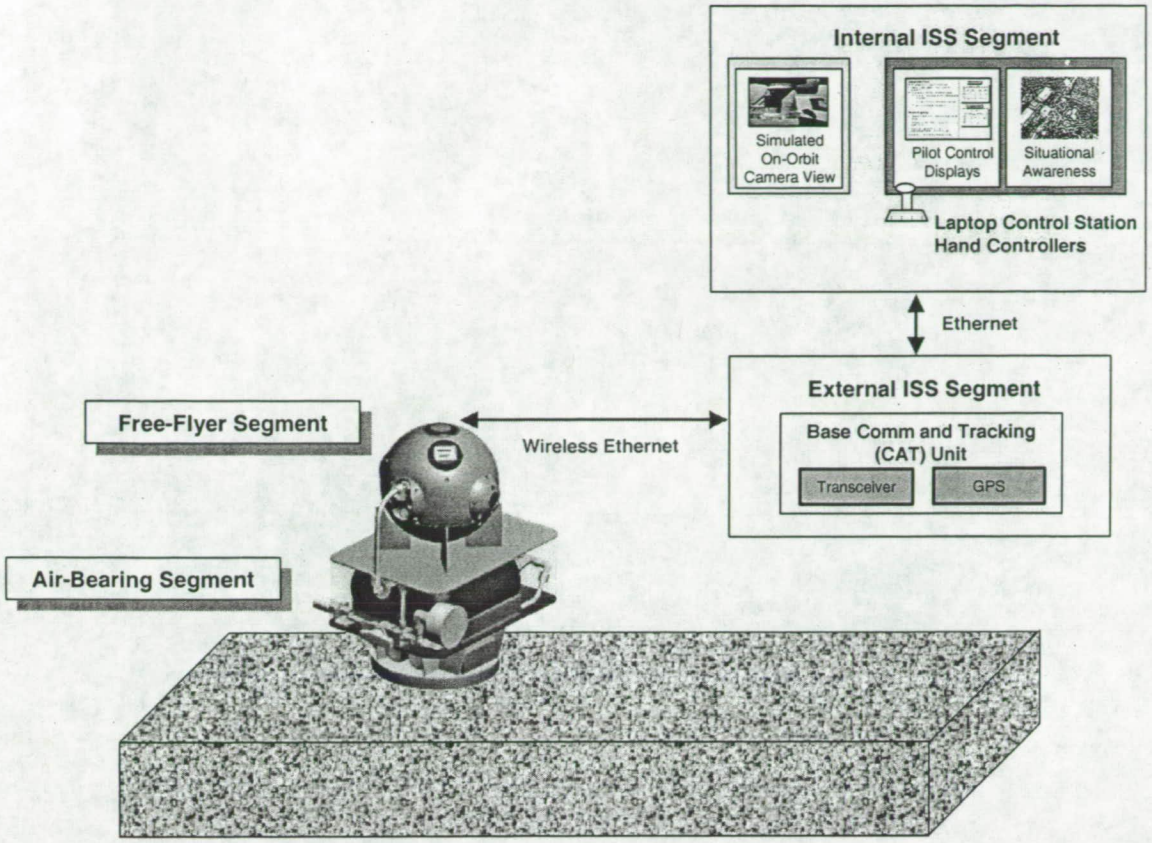

Figure 9 Air-Bearing Table Testbed Configuration

Comparison Between Air Bearing Table and Orbital Simulation Test Beds

The Orbital Simulation and Air Bearing Table facilities provide a nearly complete complement of test capabilities for Mini AERCam testing and verification. As shown in Table 6 below, most of the hardware functions are thoroughly tested on the air-bearing table, while most of the orbit GN&C software functions are tested in the Orbital Simulation. However, the avionics processor and hardware interfaces are tested in both facilities. Additionally, the orbital simulation provides a significant hardware-in-the-loop capability.

		Indoor Air Bearing Table	Orbital Simulation	Other Functional Test
Systems	End-to-end wireless communications	■		
	Camera source selection and attitude control	■		
	Illumination	■		
	Rate sensors	■		
	GPS antennas			■
	GPS		■	
	Sensor network	■		
	RF Tracking*	■		
	Thrusters	■		

	Batteries	■		
Capabilities	Teleoperation of untethered free-flyer	■		
	Representative ISS mission		■	
	3D situational awareness for operator		■	
	GPS relative navigation		■	
	Combined GPS and RF tracking navigation			■
	6DOF motion simulation		■	
	Automatic inertial attitude hold		■	
	Automatic relative attitude hold		■	
	Automatic position hold		■	
	Point to point guidance and control		■	

■ Primary demonstration
Secondary demonstration

Table 6 Test Allocation

The primary difference between the on-orbit simulation and the air bearing test bed is restriction of vehicle motion to three DOF during a single test: two translation axes and one rotation axis. However, the orientation of the AERCam can be changed within the cradle to test other axes. This also gives the developers the chance to easily test the performance of individual rate gyros. The reduction in free movement does affect the performance of the attitude guidance, particularly the rotation maneuvers, since it is based on a single rotation axis solution. When the attitude has drifted during a long test, the rotation axis may not be perpendicular to the table, affecting the efficiency of the rotation maneuver.

Even though the on-orbit test bed is a better platform for GN&C development, the air bearing test bed exposed several bugs within the software, including communication dropouts, Z-pitch thruster "machine gunning," and rate gyro filter errors. In some cases, observing the vehicle as it reacts to certain events has provided information that could not have been obtained through saved telemetry data. The visual navigation system also provides, on average, a more consistent state estimation than the GPS filter, and allows for more realistic testing of the translation guidance and control without leaping from true state data inputs to GPS filter results immediately.

The air bearing test bed also provides a simpler test, since the number of computers required for interfaces is greatly reduced. Whereas the on-orbit version uses over 7 computers, the air bearing test bed uses only three computers when video is not used, and 4 when it is enabled. This increase in simplicity allows the developers to more easily isolate problems within the system.

A suite of verification tests has been developed for both test facilities, and verification testing is underway at the Johnson Space Center.

GN&C CAPABILITIES

Add a couple of paragraphs to describe GN&C capabilities and how they provide services for various mission types. Include holds and maneuvers and what they are used for.

CONCLUSIONS

The current Mini AERCam prototype has demonstrated GN&C capabilities that can be of significant benefit to NASA's human spaceflight program for external visual inspections of space vehicles. In addition to the primary ISS viewing mission, these capabilities allow Mini AERCam to serve as a test platform for evaluating algorithms and relative navigation for autonomous proximity operations and docking around the Space Shuttle Orbiter or the ISS, or for evaluating candidate sensors and technologies.

In the future, small free-flyers like Mini AERCam may serve many functions around the ISS, such as visual or non-visual inspections, communications relay, or viewing for human and robotic extravehicular activities, both in low earth orbit and during exploration missions.

ACKNOWLEDGEMENTS

REFERENCES

[1] Reference AIAA Service Vehicles Conference Paper