(Preprint) AAS 19-156

SEEKER FREE-FLYING INSPECTOR GNC SYSTEM OVERVIEW

Samuel Pedrotty,[*] Jacob Sullivan,[†] Elisabeth Gambone, [‡] and Thomas Kirven[§]

Seeker is an ultra-low cost approach to highly automated extravehicular inspection of crewed or uncrewed spacecraft that has been designed and built in-house at the NASA Johnson Space Center (JSC). The first version of Seeker is intended to be an incremental development towards an advanced inspection capability. This effort builds on past free-flying inspector development efforts such as the Autonomous Extravehicular Activity Robotic Camera Sprint (AERCam Sprint) and Mini AERCam. Seeker was funded as an International Space Station (ISS) "X-by" Project, which required delivery of the vehicle approximately one year after authority to proceed and within the budget of $1.8 million. Seeker will fly onboard the NG-11 Cygnus mission in 2019 and will deploy after Cygnus' primary mission is completed. Seeker will perform inspection-like maneuvers within 50m of the target vehicle (Cygnus) and then dispose itself. The Seeker Guidance, Navigation, and Control (GNC) system is composed entirely of commercial off-the-shelf (COTS) and space-rated COTS items, an inertial-relative Multiplicative Extended Kalman Filter, point-to-point guidance (with various additional modes such as stationkeeping), proportional-integral translational control, phase plane rotational control, and a state machine for automated mission moding with minimal ground input.

INTRODUCTION

In-space inspection can be considered to have started with the visual inspection of the Gemini VI and VII spacecraft after their rendezvous in 1965. Since then, the need for inspection has grown along with spacecraft complexity, mission complexity, and the debris environment. Inspection techniques and technologies have also advanced with systems like the ISS's Mobile Servicing System (MSS). However, currently implemented systems and methods for in-space inspection leave much to be desired with regard to cost, responsiveness, and efficacy. For example, ISS robotic inspections require many hours of planning and analysis prior to execution and the MSS has a mass of 4,600 kg.[**,1] Seeker is intended to overcome these limitations. An incremental design philosophy, CubeSat form factor, heavy use of COTS items, advanced degree of automation, and new Class 1E hardware classification all enable Seeker to be well positioned to overcome the pitfalls of current inspection systems.

Seeker was formally given Authority to Proceed (ATP) in late July 2017 with a delivery date of October 2018 and a planned launch in April 2019 onboard the NG-11 mission. Seeker is a 3U

---

[*] Aerospace Engineer, EG/Aeroscience and Flight Mechanics Division, NASA Johnson Space Center, Houston, TX 77058

[†] Aerospace Engineer, EG/Aeroscience and Flight Mechanics Division, NASA Johnson Space Center, Houston, TX 77058

[‡] Aerospace Engineer, EG/Aeroscience and Flight Mechanics Division, NASA Johnson Space Center, Houston, TX 77058

[§] Aerospace Engineer, EG/Aeroscience and Flight Mechanics Division, Jacobs Engineering, Houston, TX 77058

[**] https://www.nasa.gov/mission_pages/station/structure/elements/mobile-servicing-system.html

CubeSat, making it approximately 10cm by 10cm by 30cm. As of this writing, all major development is complete, verification is nearly complete, and spacecraft final assembly is nearly complete. Seeker will deploy from its host Cygnus spacecraft well after Cygnus has departed ISS. Seeker will perform a series of maneuvers in close proximity to the Cygnus to demonstrate a variety of capabilities that are critical to spacecraft inspection. Flight data downlink, collection, and analysis is planned to be completed by the end of 2019. It's hoped that this will be the first of a series of missions, each providing more inspection capability than the previous. This paper is intended to provide a high-level introduction to the Seeker spacecraft and its mission with an emphasis on GNC.

A BRIEF HISTORY OF FREE-FLYER SPACERAFT

In the context of the Seeker project, we define "free-flyer" spacecraft as those that are intended to operate primarily near another spacecraft under their own propulsive control. This includes vehicles like the Autonomous Extravehicular Activity Robotic Camera Sprint (AERCam Sprint), Synchronized Position, Hold, Engage, Reorient, Experimental Satellites (SPHERES), Experimental Satellite System-11 (XSS-11), etc.[*,2,3,4] This does not include vehicles like Gemini, Orion, Cygnus, etc. Free-flyer spacecraft have been around for at least two decades. It can be hard to find information on certain free-flyer missions as only a few have been carried out by civilian space agencies. This paper only discusses the extravehicular free-flyer spacecraft developed by NASA.

AERCam Sprint can be considered to be the forerunner to Mini AERCam (both shown in Figure 1) and Seeker. AERCam Sprint was a prototype that was intended to demonstrate a free-flying light and camera system that could have been used for remote inspection of the ISS. Much of the Sprint hardware was taken or adapted from the Simplified Aid for EVA Rescue. The vehicle was a 14" diameter sphere with two cameras, an illumination light, 12 cold gas nitrogen thrusters, and a padded exterior. On STS-87 in 1997, Sprint was flown around the payload bay by Steve Lindsey from the Shuttle's aft flight deck for about 30 minutes, successfully demonstrating the vehicle's systems.

Mini AERCam was envisioned as the operational version of Sprint, significantly enhancing the prototype. The project would reduce the vehicle's mass and size, add a relative navigation system, add a guidance system, and add additional capabilities. By 2002, a flight-like prototype had been developed. The resulting vehicle was spherical with a 7.5" diameter and a weight of 10 lbs. The system and many of its capabilities were demonstrated on an air bearing table at JSC.[2] Unfortunately, Mini AERCam was not flown as neither the ISS nor Shuttle Programs required its use and it had a significant development cost to make it flight-ready.

---

[*] https://spaceflight.nasa.gov/station/assembly/sprint/

Figure 1: AERCam Sprint on STS-87 (Left) and Mini AERCam in Lab at JSC (Right)

THE SEEKER DEMONSTRATION MISSION

Seeker and its demonstration mission have been carefully designed to incorporate as many critical capabilities related to spacecraft inspection as could be accommodated in its $1.8m budget and 14 month schedule. At a high level, these capabilities include safely operating around the target vehicle, visually inspecting the target vehicle, and minimizing required human input. The mission has three tiers of success denoted as minimum (the bare minimum capability we advertised), full (the external goal capability for this mission), and stretch (an extended set of capability that would be nice to demonstrate, but that is far beyond the advertised capability). Put simply, ‘minimum success’ is when Seeker deploys, stops, and takes a picture. ‘Full success’ is when seeker performs a series of translational and rotational maneuvers. ‘Stretch’ is an additional set of demonstrations such as avoiding a keep-out sphere (KOS) and holding position during a loss of signal (LOS).

Mission segments were selected, ordered, and designed in order to consider host vehicle preferences, inspection and safety-related demonstrations, lighting, and minimizing propellant usage. As noted above, the Seeker mission requires minimal human interaction, with operators merely providing a command to proceed after planned holds that allow them to review the vehicle’s status. This is enabled by the mission’s linear design and the Automated Flight Manager (AFM). In order to accomplish the major mission phases outlined above, the mission is broken down into 41 mission sub-phases. The Seeker mission (including stretch objectives) demonstrates several safety features including low kinetic energy (accomplished by limiting maximum relative velocity), the ability to disable the vehicle (via the inhibit command), the ability to self-dispose (via the dispose command), avoidance of keep-out spheres, and safe response to a LOS. Seeker’s operational envelope, being approximately 30m away from Cygnus, was selected as a compromise between the inspection capability demonstration (which drives a reduction in range to the target spacecraft) and safety (which drives an increase in range to the target spacecraft). The translational and rotational maneuvers demonstrate the vehicle’s ability to precisely control itself across all six degrees of freedom (DOF).

The Seeker Concept of Operations (ConOps) begins with its supporting communication relay unit, known as Kenobi, powering up when Cygnus is in a Local Vertical Local Horizontal (LVLH) attitude hold with the Kenobi GPS antenna pointing zenith and the deployment vector in the direction of the orbital velocity vector. The ground will command Seeker to power on after Kenobi acquires GPS lock. On the ground, telemetry is seen from both vehicles. When it looks like both are running in their idle states and the Kenobi GPS fix is stable, a command will be sent to initialize the Seeker navigation system. The AFM will then be sent a command to prepare for deployment. Soon after, the deployment command will be sent to the NanoRacks CubeSat Deployer (NRCSD) and Seeker will be deployed. Kenobi remains within the NRCSD. After several seconds, Seeker will begin actively controlling itself to null tipoff attitude rates incurred during the deploy event and to actively point at the target (called target tracking). As it nears the targeted waypoint of +30m along the Cygnus velocity vector, it will begin to slow down and then initiate stationkeeping. This fulfills the minimum success criteria. On the ground, the telemetry should indicate that Seeker is in a relative position hold, taken several high-resolution images, the propulsion and power systems are performing nominally, and that the GNC systems are performing as expected. If those conditions are met, the ground will send a command to the AFM for Seeker to continue the mission. Seeker will then execute a series of translations while remaining in the target tracking attitude mode. Seeker will travel 5m nadir and then 5m out of plane, drawing a backwards "L" from Cygnus' perspective, then translate a few meters towards Cygnus and then back out. At this point, the ground will review the vehicle systems again and make a call to continue. The AFM will be sent another command to proceed and the vehicle will begin a series of attitude maneuvers. Upon completion, the ground will again review the vehicle systems. This fulfills the full success criteria. After passing further system checks, the ground will again send the AFM the command to proceed. The vehicle will then be provided a waypoint that is within the predefined KOS. If the vehicle rejects the waypoint, continuing to stationkeep, the AFM will then be sent another command to proceed. The vehicle will then execute its LOS demo. During the demo, Kenobi will stop broadcasting a flag that will cause Seeker to respond as if it has lost communication. However, both vehicles will still be in communication. Seeker will hold position and wait for the aforementioned flag to resume broadcasting. The flag broadcast will resume, which the AFM will detect and mode Seeker out of LOS mode. The ground will command the AFM to proceed, which will cause the vehicle to do a 90 degree pitch, demonstrating handoff between its communication antennas. After a short time, the vehicle will pitch back -90 degrees to its prior orientation. This fulfills the stretch success criteria. Figure 2 shows a visualization of the 3D trajectory relative to Cygnus from an out-of-plane position, highlighting the in-plane motion. The final action of the mission is a ground-commanded disposal maneuver where Seeker is commanded to a waypoint further along the positive orbital velocity vector and nadir from its current position. Seeker will then travel towards that waypoint until it has exhausted its propellant.

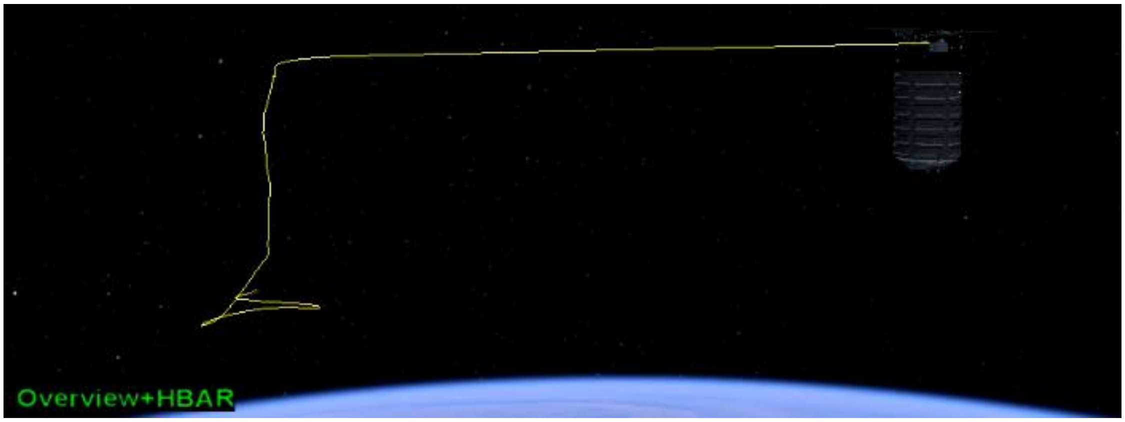

Figure 2: Side-view of Seeker trajectory trail relative to Cygnus

GNC-RELATED DESIGN

Seeker is meant to be the first step of an evolving design that adds capability, as required, in each subsequent mission. The vehicle for this first mission is intended to have a minimal capability that is able to operate safely around the target vehicle, visually inspect the target vehicle, and do so with minimal human input. It should be noted that a significant market for CubeSat components exists with many space-rated COTS products for common vehicle subsystems. The Seeker GNC system consists of an AFM Core Flight Software (CFS) application, a suite of sensors and their associated input/output (I/O) CFS applications, a Kalman filter CFS application, a state propagator CFS application, a guidance CFS application, and a control CFS application. This is a fairly typical design where the AFM configures the applications for the current mission sub-phase, the navigation system incorporates sensor inputs and provides states of the chaser and target spacecraft, the guidance system computes errors between the current and desired states, and the control system commands the effectors to reduce the errors guidance has calculated. Subsystems that impact GNC are also described in brief.

Flight Software and Simulation

Given its wide-spread use, heritage, flexibility, familiarity, and support, CFS was selected as the flight software architecture. CFS consists of an Operating System Abstraction Layer (OSAL), Platform Support Package (PSP), core Flight Executive, and various libraries and applications. The OSAL and PSP components enable a large variety of hardware and operating systems to be used without significant reconfiguration. The CFS framework has applications communicate via a publish-subscribe architecture, where these messages are all put onto and pulled from a Software Bus Network.[5] This allows applications to be built quickly as developers can focus on their functions instead of on the inter-application communication. This also allows developers with limited knowledge of CFS to flesh out application templates into full-fledged applications very quickly that require minimal integration into the rest of the CFS framework. Given this segmented, modular application approach, it is very easy to reuse applications in future development. CFS is often visualized as a "bubble" chart, where each application is represented by its own "bubble." This is shown in Figure 3.

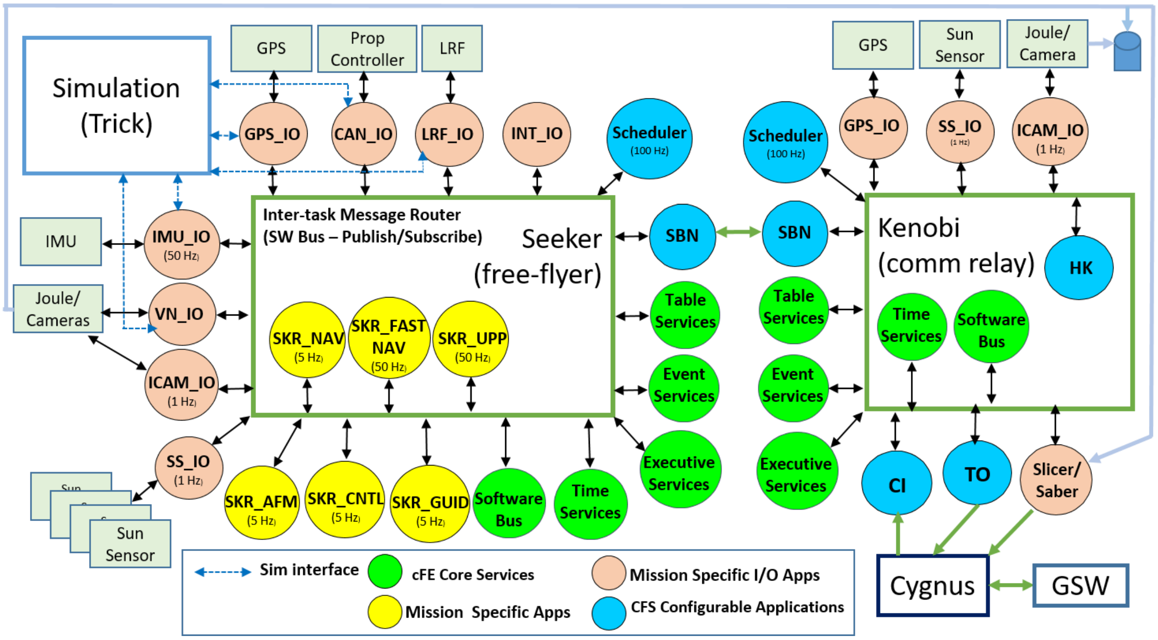

Figure 3: CFS architecture "bubble" chart

The GNC subsystem required an integrated simulation for development. This drove the development of an integrated environment where FSW could drive a physics-based CubeSat model. The Trick simulation development environment was selected due to its capability, common usage at JSC, and in-house support. A simulation was created that modeled the Cygnus target vehicle, the

Seeker chaser vehicle, effectors for both, and celestial bodies. The Valkyrie generic model package was used for the sensor models and the JSC Engineering Orbital Dynamics package was used for the dynamics.[*] Trick allows for data logging, faster than real-time runs (with an option for real-time), and has a Monte Carlo capability.

In order to have an integrated simulation and FSW environment that can be used faster than real-time, an interface between Trick and CFS was required. This had been done in the past with an application called TrickCFS, which was upgraded to interface the newer versions of Trick and CFS that the Seeker project was using.

The Trick/CFS/TrickCFS framework was used to develop and analyze the GNC and FSW system. The Monte Carlo capability within Trick was used to assess the impacts of various combinations of environmental conditions, sensor noise, and other events. This produces an enormous amount of data that is difficult to parse with the simple Python plotting scripts that were used for single-run analysis. For quickly loading and assessing data, an internal JSC data analysis package called Koviz was used. Koviz is designed to quickly load and plot large volumes of data. A key feature of Koviz is its ability to load the associated dispersed parameters and then sort the plotted lines based on them, clearly highlighting the driving parameter (if there are any). For automated requirements checking, another internal JSC tool was used, known as VERAS. VERAS loads the run data, parses it, compares it to requirements, and then creates pdf reports that show how the requirements are (or are not) met. The reports also detail the exact software version used to create the data, eliminating confusion as the number of run sets increases.

The ability to visualize 3D representations of the physical vehicles and nearby planetary bodies allows for rapid assessment of performance. This created a strong desire for a visualization package that could be paired with the integrated simulation and FSW environment. This was done with the Engineering DOUG Graphics for Exploration software package.[†] High fidelity Seeker and Cygnus models were added along with approximate models of the inspection and navigation cameras. This not only provided a faster-than-real-time, in-the-loop way to assess the vehicle's behavior during simulation runs, but also provided a visualization of behavior during hardware-in-the-loop demos and a quick way to assess the lighting environment and impact of various deployment configurations.

AFM

The AFM is essentially a state machine that ensures the GNC software is appropriately configured for the current mission sub-phase. The AFM gets its knowledge of phase and sub-phase configuration from a user-created CFS table, known as an initialization load (iLoad). This file contains descriptions of each mission phase and sub-phase that define the behavior of the applications that receive information and/or commands from the AFM. After it initializes, the AFM only receives ground commands and guidance error. While the AFM typically only advances through states in an incremental fashion (e.g. state 1, state 2, state 3), it can be moded via ground command from any phase into the terminal "dispose" and "inhibit" states. During normal operation, the AFM is looking for "triggers" to transition from the current sub-phase to the next. The "triggers" that AFM supports includes time-based, ground command-based, and guidance error-based (referred to as a "deadband" and can be cued from rotational and translational position and rate). In addition to the GNC applications, the AFM also drivers the behavior of the inspection camera and the applications involved in the LOS demo.

---

[*] https://www.nasa.gov/centers/johnson/techtransfer/technology/MSC-24532-1-jeod.html

[†] https://software.nasa.gov/software/MSC-24663-1

Sensors

Seeker has a sensor suite that includes a Sensonor STIM 300-400-5 Inertial Measurement Unit (IMU), Jenoptik DLEM-SR laser rangefinder (LRF), four Solar MEMS nanoSSOC-D60 sun sensors, and a Sony FCB-MA130 camera paired with algorithms developed by the University of Texas at Austin (UT). Some of these sensors can be seen integrated into the sensor bracket in Figure 4. The sensor types required for this mission were selected by performing a study using the linear covariance analysis tool (LinCov) to determine the minimum set of required sensors that met the mission requirements, including size, weight, and power (SWaP), performance, schedule, and budget. Once the sensor types had been selected, candidate units were identified from a study of COTS and space-rated COTS items that have demonstrated space heritage. In certain cases where sensors were less than $10k, several units were purchased to evaluate in parallel in an effort to reduce the schedule and maximize performance. On the COTS side, special preference was given to tactical-grade units as they already have a significant amount of environmental robustness. The STIM IMU was on-orbit onboard ISS as part of the Raven package on STP-H5 and the DLEM-SR was on-orbit onboard AeroCube 7. While these units are more rugged than standard COTS items and had limited on-orbit heritage, concern about the possibility of the unpublicized changes in the assembly and makeup of these units necessitated a brief environmental test campaign in order to qualify them for the space environment. The units were subjected to thermal, vacuum, vibration, and blinding tests. After each test, it was confirmed that their operation was within their specifications. The units were also tested for shock and through other environmental tests similar to before when integrated into the vehicle with their functionality and performance verified afterward.

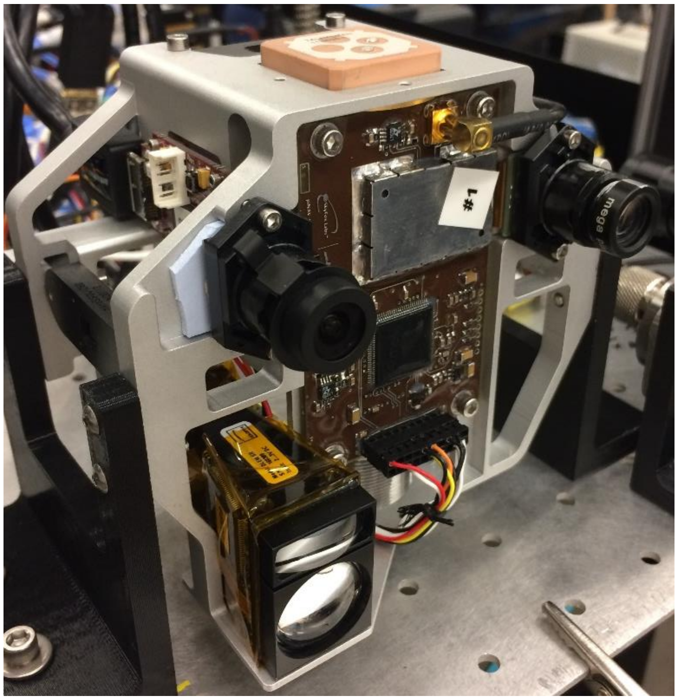

Figure 4. Integrated sensor bracket on the FlatSat with sun sensors (left and top), LRF (lower left), vision-based navigation camera (center left), GPS receiver and antenna (center and top), and IMU (occluded behind right edge).

Vision-Based Navigation

The vision-based navigation system (VizNav) was envisioned to provide a bearing measurement to the target vehicle. Frequently changing ConOps meant that the algorithm would need to be robust to various lighting conditions and potentially the Earth in the background of the image. Given the aggressive schedule, there also would not be an opportunity to gather imagery of the Cygnus with the baselined camera in simulated on-orbit lighting, driving the need for increased robustness. With these significant demands and a very tight schedule, a parallel development with traditional computer vision and neural-network-type approaches was used to develop three algorithmic approaches. By the time a selection was made, only two were still being considered.

A test campaign was developed similar to the Orion Program's approach to verification of vision-based navigation algorithms where a high resolution monitor was placed in front of the camera such that it filled its field of view. Synthetic and real video with resolution as high as possible were displayed that included the target vehicle with various backgrounds: an effectively solid background, a busy background, some with the target vehicle at various ranges and attitudes, and some without the target vehicle at all. The traditional algorithm failed to differentiate the target from the background in nearly all cases. The algorithm developed by UT was able to identify and bound the target in nearly all cases with very few false positives. The UT algorithm uses a neural network to identify the Cygnus in the image and then bound it. Within the bounding box, a traditional algorithm is then used to attempt to outline the vehicle silhouette. The resulting region within the outline is then centroided, which is then used to calculate the bearing relative to the center of the image plane. The output of the debug mode of the software highlights several of these mechanisms in action, as shown in Figure 5.

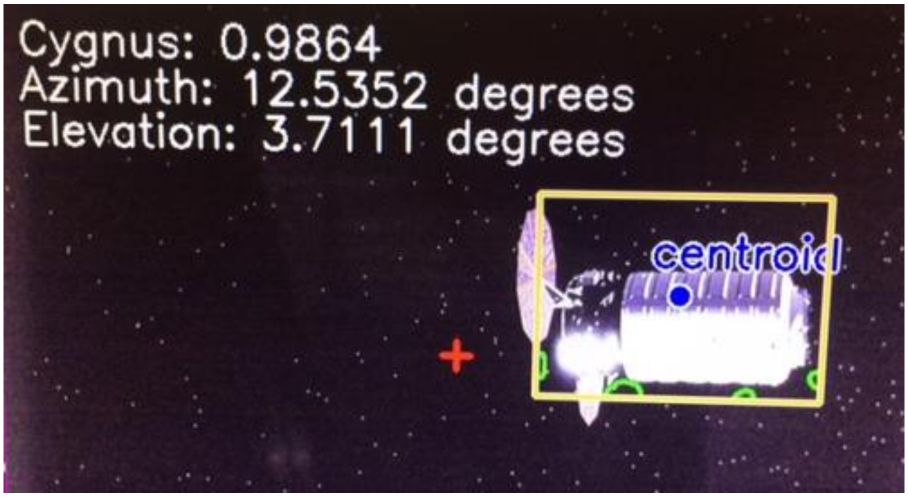
Cygnus: 0.9864
Azimuth: 12.5352 degrees
Elevation: 3.7111 degrees

Figure 5. Debug GUI of the VizNav software showing the confidence of the Cygnus in the upper left along with the bearing. The yellow box indicates the algorithm has located the Cygnus and the blue dot labeled 'centroid' is the calculated centroid.

Navigation

The Seeker navigation system leveraged work done on Project Morpheus at JSC in the early 2010's. The navigation architecture consists of three parts, the IMU Preprocessor (IMUPre), the fast propagator (FASTNAV), and the Kalman Filter (NAV). IMUPre and FASTNAV each run at 50 Hz, while NAV runs at 5 Hz. This architecture is shown in Figure 6.

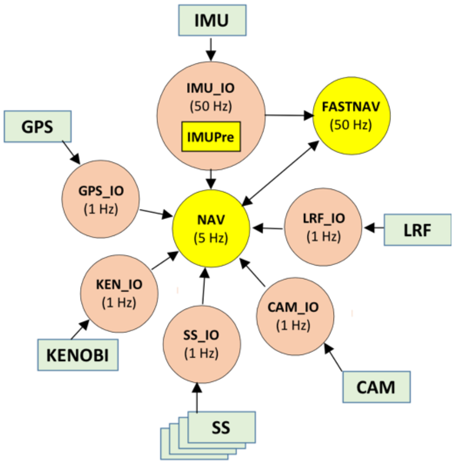

Figure 6: Seeker Navigation Architecture

The IMUPre application has Morpheus and Resource Prospector heritage and was largely used as it was originally designed. It is responsible for down-sampling the high-rate data from the IMU, performing coning and sculling corrections, and producing a single 50 Hz output for the other navigation components. The IMUPre application maintains its own inertial reference frame which is snapped at the system startup and enables the use of multiple IMUs, if desired. The accumulated delta-V and an updated vehicle body-to-reference frame quaternion are output from the application, along with instantaneous sensed acceleration and angular rates in the vehicle body frame. By combining the IMUPre inertial reference frame with an assumption of the Seeker position and orientation at deployment, the vehicle is able to obtain an initial estimate of its attitude in a J2000 reference frame.

The FASTNAV application performs the high-rate integration of the vehicle state. Once the NAV application initializes the vehicle state vector and computes the gravity vector and the gravity gradient matrix for both Seeker and the Cygnus vehicle (based on a known initial relative state), FASTNAV is able to integrate both the Seeker inertial state and the Seeker-to-Cygnus relative state through time. Gyroscope and accelerometer bias is compensated by removing the bias estimates (Gauss-Markov states) from the measurement. The body frame attitude change and the accumulated delta-V output from IMUPre are then used to integrate the vehicle state. The Cygnus state is derived from the Seeker inertial state and the estimated relative state, and both the Seeker and Cygnus states are propagated forward using a second-order Taylor series expansion, incorporating the sensed velocity change applied to Seeker only.

Once the vehicle states are integrated, FASTNAV computes the dynamics partial derivative matrix and integrates the vehicle state transition matrix (STM), again using a second-order expansion. A frame counter between FASTNAV and NAV monitors for each time the NAV application will run and resets the STM back to identity after passing out the propagated matrix for NAV to use, along with the dynamics partials for measurement back-propagation. Additionally on this frame, FASTNAV integrates the last state vector correction from NAV to the current time using the STM and updates the state vector to incorporate the measurements.

The NAV application utilizes a Multiplicative Extended Kalman Filter (MEKF) formulation to perform the measurement update to the state vector at 5 Hz. The filter carries 24 states that cover Seeker's inertial position, velocity, attitude (deviation), relative position, relative velocity, gyro bias, accel bias, LRF bias, and camera bias. At startup, NAV waits for the FSW to receive data from the Kenobi GPS receiver. Upon receipt of valid GPS data, NAV uses the ECEF state from the GPS to initialize the Seeker navigation state vector. In the event the Kenobi GPS fails to acquire, the project has implemented the capability to command an estimated ECEF state and time from the ground to initialize navigation.

In each cycle, NAV begins by performing its time update, updating the state vector using the FASTNAV propagated state, computing the process noise matrix, and updating the state covariance matrix. Once the vehicle state is integrated to the time tag of the most recent IMU measurement, NAV checks for new measurements from each of the vehicle sensors by comparing the time tag of the data packet received from the sensor IO application to the time tag of the last known measurement. If a new measurement was received, NAV uses the current dynamics matrix from FASTNAV to back-propagate the vehicle state to the measurement time, compute the estimated measurement, residuals, and measurement partials. The measurements are processed one at a time to avoid matrix inversion and accumulated into a single state update vector using the Joseph form to update the covariance matrix. It is considered best practice to use the Joseph update in order to assure a symmetric covariance matrix. The filter has configurable underweighting parameters for each measurement type to ensure smooth convergence and an iLoaded residual edit threshold, which is compared to the square of the measurement residual over the measurement innovation, to reject spurious measurements. After the attitude correction is rectified, the state update is passed out to be used by FASTNAV along with an updated gravity and gravity gradient vector to perform the next propagation steps.

Guidance

Seeker's guidance algorithm was designed and integrated with the control algorithm to achieve waypoint seeking, position and attitude holds, target tracking, and to limit Seeker's relative kinetic energy. Since Seeker will operate in close proximity (approximately 30 m) to the target vehicle, relative orbital dynamics were assumed to be negligible, which simplified the guidance algorithm. The guidance application runs at 5 Hz, receiving mode and waypoint information from the AFM as well as the current state from the navigation system. Using this data, guidance computes velocity, attitude, and angular rate errors in the LVLH frame. The velocity error is generated using logic similar to potential field approaches, which generate a force as a function of position within an artificial potential field.

The method implemented on Seeker computes a velocity command as a function of distance to the desired waypoint. This function outputs a constant magnitude velocity command if Seeker is farther than an iLoaded distance to the current waypoint and a linearly decreasing velocity command as Seeker approaches the waypoint within the iLoaded distance. Effectively, this is an outer loop controller that limits Seeker's kinetic energy. The velocity error is then just the difference between the velocity command and Seeker's velocity relative to Cygnus.

Target tracking is achieved by computing a desired attitude that orients Seeker's +X body axis along the position vector to the target vehicle. This ensures the LRF and navigation camera stay pointed at the target vehicle to provide range and bearing measurements to the navigation system. The attitude error is the rotation between the current and desired attitude, and is converted to Euler angles which are used by the phase plane control algorithm. The rate error is computed using the phase plane control logic and attitude errors. Position and attitude holds are achieved by treating Seeker's position and attitude (at the moment the holds are initiated) as the desired waypoint and the desired attitude respectively, which permits re-use of waypoint seeking and attitude maneuver logic. Additionally, the guidance algorithm checks AFM-provided waypoints in relation to an iLoaded KOS to ensure Seeker is not commanded into or through the KOS. Future implementation of this hazard avoidance approach will likely use dynamically generated KOSs as vehicle environmental/situational awareness improves.

Control

The control flight software application runs at 5 Hz and receives new inputs each cycle to calculate thruster firing commands. The main function reads the current errors from guidance and the current control modes from AFM, as well as the control inputs. The control inputs can be updated based on the AFM flight phase. The main function calls the control and thruster selection algorithms and publishes the resulting thruster firing commands for the propulsion and navigation applications. A proportional-integral function calculates the command per axis from the translational error, and a phase plane function calculates the command per axis from the rotational error. The thruster selection function calculates the firing time per thruster from both the translational and rotational commands. A final thruster combination function combines the translational and rotational firing commands and limits the number of thrusters firing to meet vehicle limitations.

A thruster test function can be run to verify the thruster mapping and interface between GNC and the flight hardware. If the control mode is set to OFF in AFM, the control application continuously publishes zeros as its output. Once the control mode is set to FLIGHT, the control application commands the thruster isolation valve open and begins processing the errors from guidance. The control application zeros stale guidance data and resets the integral term after free drift to prevent extraneous thruster firings. The functions are generically designed and can be implemented in other spacecraft GNC systems with updated inputs.

Avionics, Communication, and Power

The avionics, communication, and power subsystems consist of a combination of COTS, space-rated COTS, and custom boards that provide all the electrical interfaces, processing power, wireless communication, and power for the other subsystems. The integrated power system, computers, and other avionics boards are shown in Figure 7.

Seeker and Kenobi are both designed to each have two computers onboard. The primary computer is the National Science Foundation’s Center for Space, High-performance, and Resilient Computing (CHREC) Space Processor (CSP). The CSP is a space-rated COTS item, balancing cost and reliability in the space environment. The CSP has a dual-core ARM Cortex A9 processor, 250 MB of RAM, and a Field Programmable Gate Array (FPGA), used for various interfaces on Seeker and the high speed serial connection on Kenobi. An Intel Joule 570X was selected as the secondary computer, also known as the Camera Image Processor (CIP) as it was intended to be used for the CPU-intensive VizNav algorithms. This selection leveraged the prior assessment done by the High Definition EVA Mobility Unit Camera Assembly project. The Joule has a quad-core Intel Atom processor at 2.4 GHz per core and 4 GB of RAM in a 24x48mm form factor (not including interface board). The translation of FSW control commands into valve commands is done by a custom board internally referred to as the prop controller, which uses an FPGA.

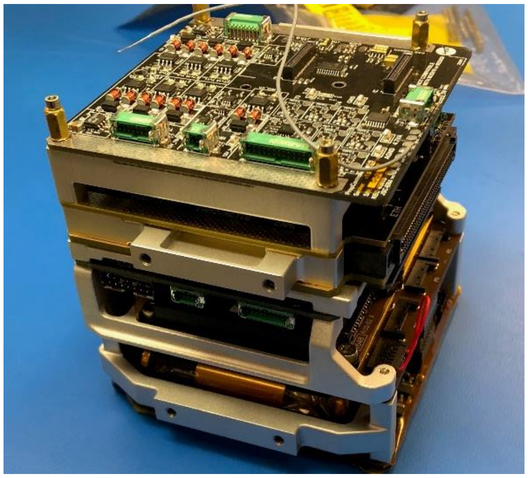

Figure 7: Seeker avionics stack

Seeker has two Sony FCB-MA130 cameras on board. One has a 7.2 mm lens and the other has a 20 mm lens. The one with the 7.2 mm lens is used for the VizNav algorithms (known as the navigation camera) and then other (known as the inspection camera) is used for taking high resolution images of the target spacecraft. The CSP, Joule, and cameras, are connected with a COTS USB hub.

Seeker and Kenobi communicate wirelessly via 5 GHz wifi. The Joule onboard Kenobi is connected to a COTS Netis WF2190 USB wifi dongle, which is then connected to a COTS Tecom C-band antenna. Seeker has another Tecom C-band antenna that is connected directly to its onboard Joule. A spring finger connection on the upper surface of Seeker allows Kenobi to actuate Seeker's latching relay, powering the vehicle, while commands and data are exchanged wirelessly through a patch antenna embedded into Seeker's NRCSD tube.

Power for Seeker is provided by a space-rated COTS GomSpace NanoPower BP4 Lithium-Ion battery pack distributed via two GomSpace P60 PDU-200 power distribution units all integrated on a GomSpace P60 dock. This configuration is advertised to provide 38.5 Wh of energy, which is expected to provide approximately 90 minutes of operation. Power for Kenobi is provided directly from the Cygnus host vehicle.

Propulsion

The propulsion subsystem uses cold, gaseous nitrogen and twelve 0.1N thrusters to provide control authority in all six DOF. The system was designed entirely in-house at JSC and is comprised of COTS fluid components, custom machined manifolds, and custom additively manufactured thrusters. The propulsion subsystem is advertised to provide approximately 5m/s delta-V to a 5.75kg vehicle with the subsystem itself packed within an approximately 1.25U form factor. The integrated system is shown in Figure 8. Even though the Seeker project had greater flexibility in a variety of requirements due to the Class 1E classification, the propulsion subsystem still had to meet a number of ISS safety and performance requirements.

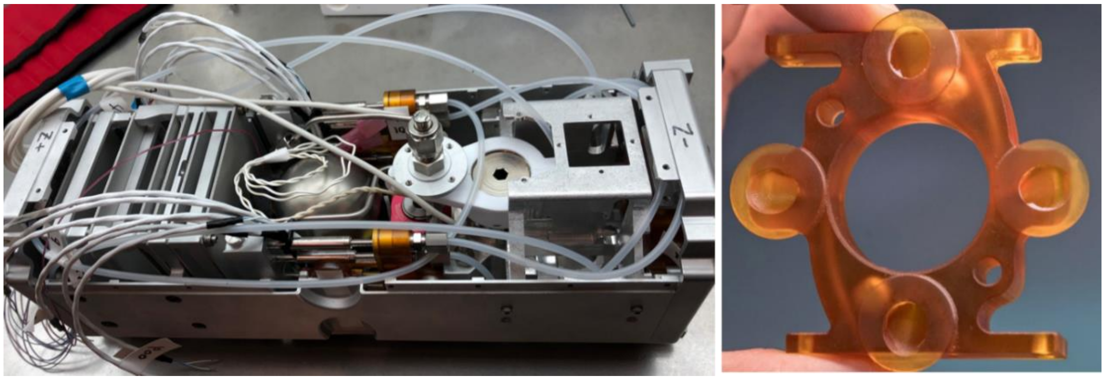

Figure 8: Integrated propulsion system in its pre-vibe configuration (left) and an additively manufactured thruster (right)

The titanium Arde tank is a Mini AERCam-heritage component. The remaining downstream components are COTS (including the TESCOM BB7012 regulator, Lee Co PRRX relief valve and IEPX thruster valves). The thrusters are additively manufactured with Cyanate Ester 221 using the continuous liquid interface production (CLIP) process. The resulting part is translucent, making visual inspection easier, as can be seen in Figure 8.

Structures

The Seeker and Kenobi primary structures are designed to be as similar as possible. Both are assembled from plates machined from AL 7075 and then hard anodized. The fully assembled vehicles are shown in Figure 9. The avionics boards are all assembled into a compact package with spacers made from anodized AL 6061, known as the avionics stack. The spacers also provide the connections required for passive thermal management. The IMU, cameras, LRF, and three sun sensors are all rigidly mounted to a separate structural component, known as the sensor bracket, which is then fixed to the primary structure. A picture of the sensor bracket is shown in Figure 4.

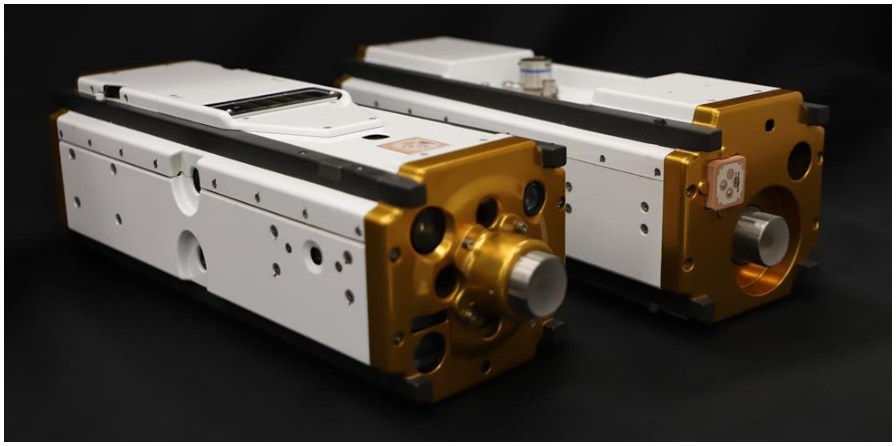

Figure 9: View of assembled Seeker (left) and Kenobi (right) vehicles.

SEEKER FORWARD PLAN

At the time of this writing, the Seeker and Kenobi vehicles are undergoing final assembly and testing. Various reviews are planned over the next couple of months, culminating in delivery of the hardware in March of 2019 to NanoRacks for integration. Launch is slated for mid-April 2019 with deployment and operations occurring in the late July to early August timeframe. Data downlink should be completed over the next several days and analysis (including construction of a best-estimate trajectory from the video and LRF data) should begin shortly afterward, concluding with preliminary flight results by the end of CY 2019. It is hoped that a follow-on mission will be announced shortly after the completion of the current mission.

The Seeker GNC team has already identified several candidate areas for improvement in a follow-on mission. This list will be revised depending on requirement changes and results from the mission. It should be noted that there is no plan to pursue these improvements until a follow-on mission is funded. Foremost is the upgrade of the VizNav software from a bearing measurement to a complete pose measurement. It is thought that this could be done with a relatively small change to the current neural network and should significantly improve the relative state estimate. Next is adding a low cost and SWaP LiDAR to the sensor package as an on-orbit demonstration of such sensors which are robust to lighting and provide many other benefits. The next item on the list is changing the inertial-relative formulation of the MEKF to a kinematic one. This requires further analysis, but may make the GNC system more environment agnostic. Next is upgrading the guidance from point-to-point to potential field-based, which is required for eventual operations in complex spacecraft environments. Other proposed changes include replacing the DLEM-SR with the DLEM20 (improved SWaP), implementing a control for the IMU's noise (either a threshold or a LPF), implementation of the LOS capability as a true capability (as opposed to a demo), upgrading the control system to be optimization-based, trade reaction wheels for attitude control, and adding additional fault detection, identification, and recovery, and non-linearity to the AFM.

CONCLUSION

Seeker is a quickly and relatively cheaply developed 3U CubeSat that is intended to demonstrate basic capabilities required to safely perform visual inspection of crewed and uncrewed spacecraft. The Seeker demonstration mission is slated for mid-2019 and includes a variety of maneuvers and tasks that are intended to pave the way for future missions. It is hoped that further incremental development efforts will build off of this base, eventually resulting in a highly effective and efficient inspection platform for vehicles operating throughout space.

ACKNOWLEDGEMENTS

The authors would like to thank the entire Seeker team (including our awesome interns), the UT vision-based navigation team, Northrop Grumman, NanoRacks, Thilini Schlesinger, John McGregor, Dave Saley, Chris D'Souza, Jack Brazzel, Greg Holt, John Goodman, Bob Scully, Xiang Ni, Ami Yang, Steve Frederickson, Rob Ambrose, Jenny Devolites, Joe Galante, Siegfried Janson, and all of the other people who provided us invaluable support through this whirlwind adventure.

REFERENCES

[1] R. More, "ISS Inspection Capabilities and Challenges." 2014 In-Space Inspection Workshop

[2] J. Wagenknecht, Et al., "Design, Development and Testing of the Miniature Autonomous Extravehicular Robotic Camera (Mini AERCam) Guidance, Navigation, and Control System." *26th AAS Guidance and Control Conference*, 2003

[3] T. Fong, Et al., "Smart SPHERES: a Telerobotic Free-Flyer for Intravehicular Activities in Space." *AIAA SPACE Conference*, AIAA 2013-5338, 2013

[4] *XSS-11 Micro Satellite.* AFRL, Kirtland AFB, September 2011

[5] *core Flight System (cFS) Background and Overview.* NASA GSFC, December 2017