Astrobee

Guest Science Guide

IRG-FF029
2017-08-31
National Aeronautics and Space Administration
Ames Research Center
Moffet Field, CA

Contents

1 Introduction

Welcome! We're the NASA Astrobee Research Facility, and our primary goal is to provide guest scientists like you with a powerful new platform for zero-gravity robotics research inside the International Space Station (ISS). Astrobee's success depends on your creativity, and we're excited about helping you fly your own experiments. You can always find our up-to-date contact information and the latest on project schedule and status on our web site: `https://www.nasa.gov/astrobee`.

The Astrobee Research Facility will maintain three identical free-flying Astrobee robots on the ISS. After the Astrobees are launched and commissioned in 2018, they will replace the SPHERES robots that have been operating on the ISS since 2006 (Fig. 2). Over the years, the SPHERES have been among the most-used payloads on the ISS, supporting dozens of experiments from a variety of guest scientists. In the next section, we'll talk about past SPHERES experiments as possible inspiration for your future research on Astrobee. Compared to SPHERES, the Astrobee robots will offer many new capabilities and will require less astronaut time to support, so we hope the new facility will be able to fly experiments much more often.

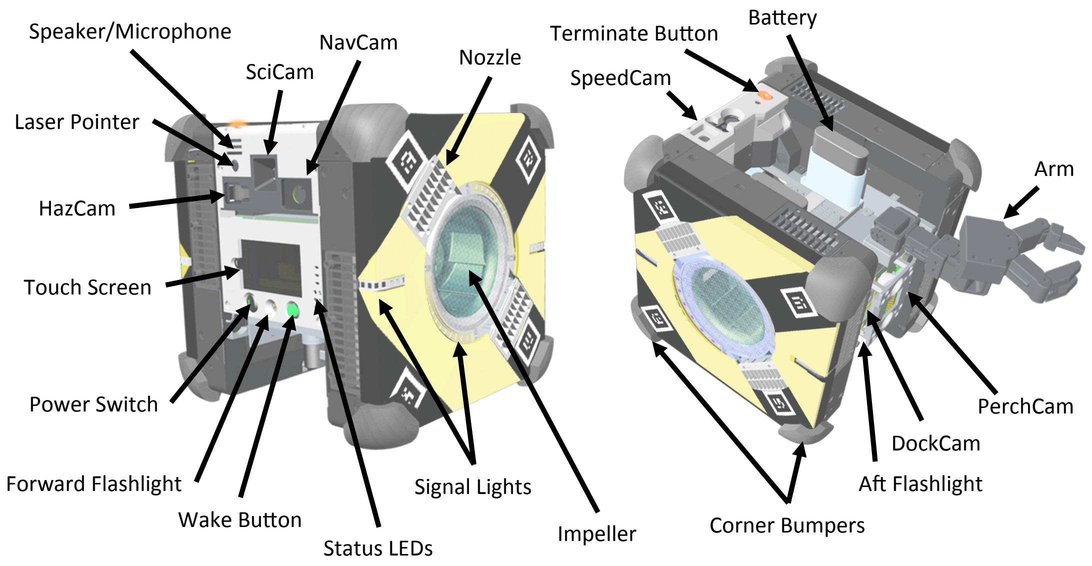

Figure 1: An Astrobee robot.

Each Astrobee is shaped like a cube 12.5 inches (32 cm) wide (Fig. 1). It can fly autonomously throughout most of the US section of the ISS interior, but cannot operate outside the ISS. It is propelled by a pair of battery-operated fans, and can autonomously return to a docking station to recharge, so it can perform most activities without requiring any astronaut support. It carries a suite of six cameras, a two degree-of-freedom (DOF) arm with a gripper that can grasp ISS handrails and other objects (Fig. 2), and three payload bays that provide power and data for guest science hardware. It can autonomously execute hours-long plans (for example, sensor surveys) or be teleoperated live from the ground or by astronauts.

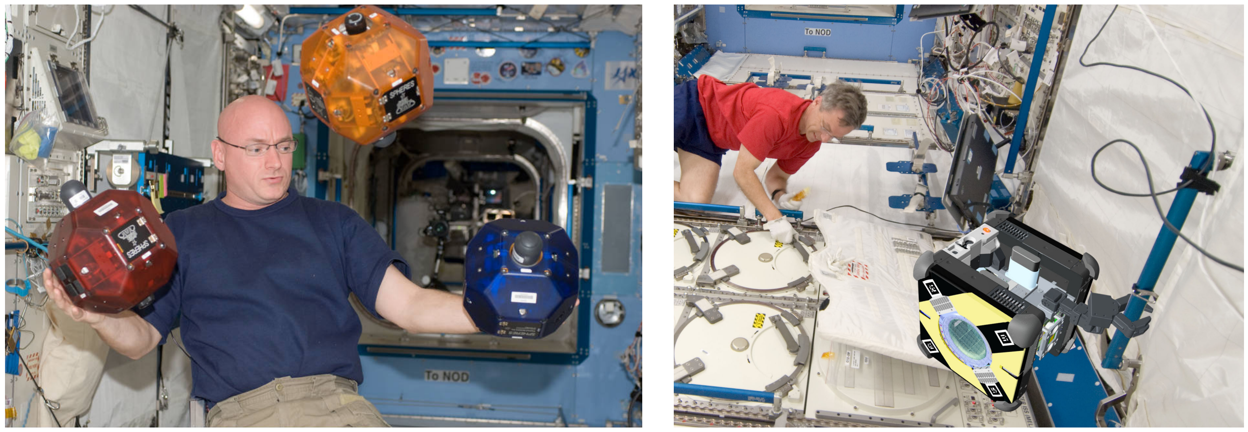

Figure 2: (left) The SPHERES robots. (right) Artist's concept of an Astrobee robot perched on a handrail to take video of a crew activity.

1.1 Research Themes and Capabilities

Astrobee is designed to support research in a wide variety of areas, including:

- **Motion control:** Each Astrobee has 6-DOF holonomic motion capability (instantaneous thrust in any direction and torque about any axis) provided by 12 variable-thrust nozzles. This effectively allows each robot to simulate any space vehicle thruster configuration. Astrobee's baseline flight software will be open sourced and available to you as a guest scientist. It provides a robust controller (implemented in MATLAB SIMULINK and ROS/C++) that you can use as a starting point for your own more advanced experimental controllers. As a guest scientist, you can update nearly all of Astrobee's flight software as needed to support your experiments.

  Background: Airbus/MIT SPHERES tether experiments studied the dynamics of two SPHERES flying together while connected with Kevlar thread (Fig. 3) [MLS+16].

- **Advanced propulsion / mobility hardware:** You can supplement Astrobee's built-in propulsion system with your own experimental mobility hardware (magnetic propulsion, hopping, you name it). Take advantage of Astrobee's baseline mobility capability to maneuver the robot into any desired initial conditions for your experiment, then switch to your mobility hardware, and continue using Astrobee's built-in position tracking to log performance.

  Background: UMD SPHERES RINGS experiments investigated the use of magnetic propulsion to enable spacecraft to fly in a stable formation without using propellant (Fig. 3) [PAS+14].

- **Robotic manipulation:** Each Astrobee carries a 2-DOF arm with a 1-DOF passively underactuated tendon-driven gripper, capable of grasping ISS handrails and other objects. As a guest scientist, you can use the arm as-is, replace the standard gripper with your own advanced gripper design, or replace the whole arm—all the hardware is designed for easy swapping by astronauts. For perception, the arm workspace is

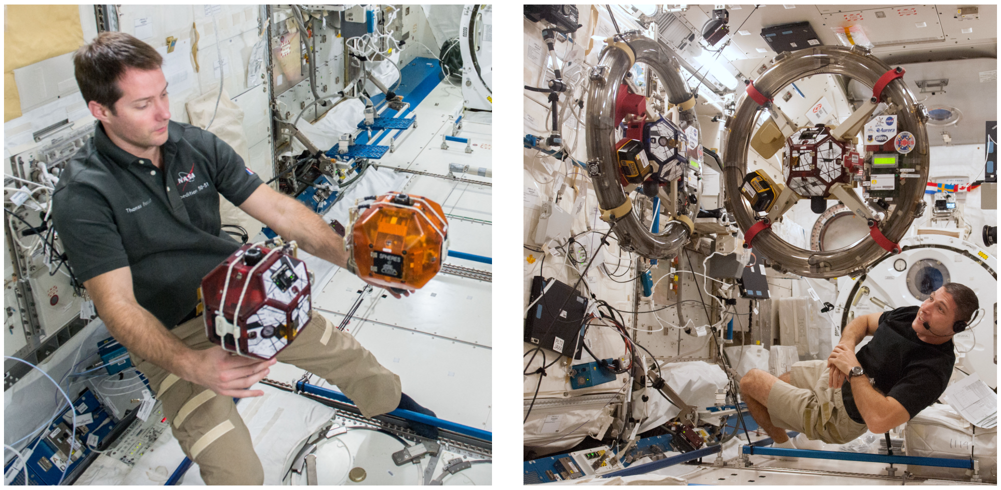

Figure 3: (left) SPHERES tether experiment—the thread between tho two robots is too thin to make out in this photo. (right) The SPHERES RINGS payload.

in view of LIDAR and RGB cameras. The arm controller's firmware (written in PIC microcontroller C) will be open sourced and available to you as a guest scientist. You can update the software as needed to support your experiment.

Background: Stanford University is measuring the performance of grippers that are coated in material inspired by gecko feet and are able to grasp flat surfaces. They are planning to integrate a version of the gripper with Astrobee (Fig. 4) [EJN+17].

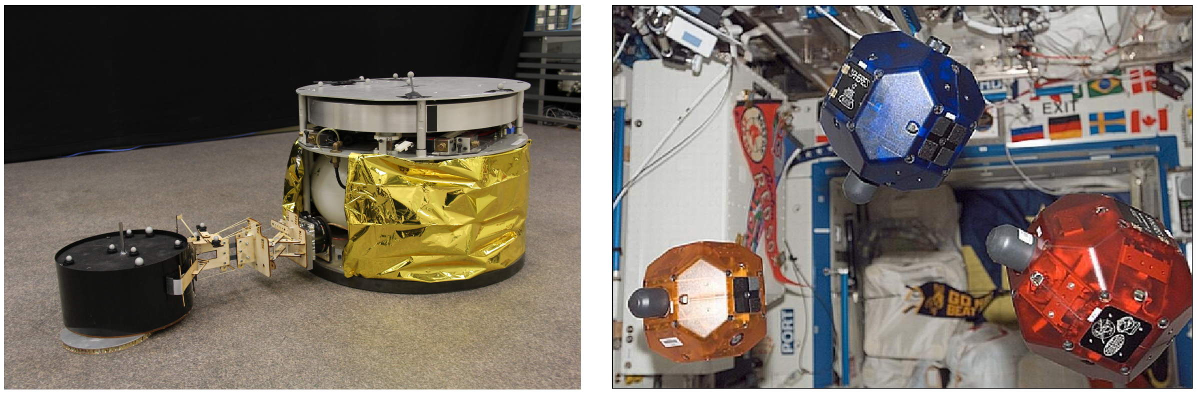

Figure 4: (left) Stanford gecko gripper testing. (right) SPHERES formation flight.

- **Multi-robot teams and formation flight:** The three Astrobee robots are capable of operating together and communicating with each other via the ISS WiFi network in order to support multi-robot experiments. Each Astrobee will be marked with visual fiducials that can be used by other Astrobees to track relative position.

  Background: Some of MIT's first SPHERES experiments on the ISS demonstrated formation flight using a peer-to-peer motion control architecture (Fig. 4) [CASM07].

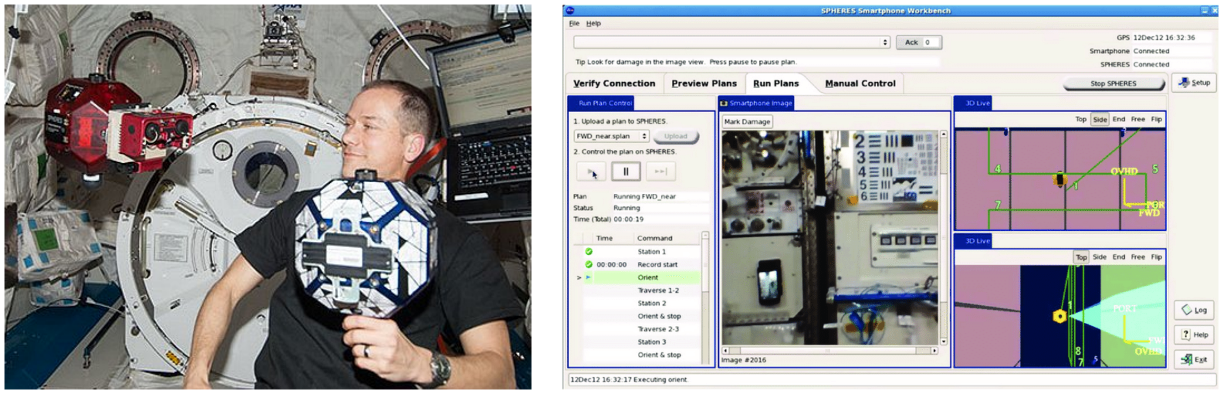

Figure 5: (left) The SPHERES VERTIGO payload. (right) The Smart SPHERES control interface.

- **Satellite inspection and rendezvous:** Each Astrobee carries a suite of six cameras (including LIDAR sensors and a 21 MP RGB camera), and can fly arbitrary holonomic trajectories around another Astrobee or a passive object. It can grasp an object with its arm to simulate rendezvous and maneuvering in a mated configuration.

  Background: MIT's SPHERES VERTIGO payload demonstrated using stereo vision to build a 3D model of a satellite, then navigating relative to the satellite using the model (Fig. 5) [TMSOM12].

- **Human-robot interaction:** For interacting with co-located astronauts, each Astrobee robot has a touch screen, a speaker and microphone, signal lights, and a laser pointer. Astrobee operators use a control station interface that supports 3D visualization of the robot location inside the ISS, generating plans, monitoring execution, live teleoperation, and custom guest science commands. The control station software will be open sourced and available to you as a guest scientist. It is written in Java using the Eclipse RCP framework, which makes it easy for you to extend the control station with new interface elements. The control station software is designed to run anywhere, ranging from astronaut laptops, to the ISS Mission Control Center in Houston, to a computer at your home institution, so you can help operate your own experiments running live on the ISS.

  Background: NASA Ames' Smart SPHERES experiment demonstrated a ground operator guiding a SPHERES robot through an inspection task using a survey plan status display, live video, and 3D position views (Fig. 5) [FMM$^+$13].

- **Surprise us!** When the SPHERES robots launched in 2006, nobody could have anticipated the wide variety of guest science they would eventually enable. We expect the same from Astrobee. We are committed to supporting your most innovative advanced concepts.

1.2 Becoming a Guest Scientist

A great first step is to reach out to contact us at the facility and have an initial discussion about the feasibility of your research plan. You can also ask us for an invitation to meet the larger community of guest scientists through the SPHERES/Astrobee Working Group, which meets four times per year (you can attend these meetings virtually through screen sharing and a call-in number).

There are a wide variety of ways you can use Astrobee, ranging in difficulty from simply specifying a plan for an Astrobee to run (for example, if your research requires a certain kind of visual inspection of the ISS interior), all the way up to developing your own hardware payload that might have complex interactions with core Astrobee capabilities such as propulsion. We recommend starting small and working your way up toward bigger experiments—after your first time, you will have a much clearer idea how the guest science process works and how to realistically plan for bigger experiments.

To try out Astrobee technically, you can start by downloading our open-source Astrobee simulator, based on ROS/Gazebo, and working through the tutorials.[1] The simulator framework is highly extensible. You can plug in your own modified flight software, or extend the simulator to model your a new hardware payload.

You can also request ground testing time in our labs at NASA Ames Research Center in Mountain View, California (Fig. 6). Three Astrobee robot ground test units will be available. In the Granite Lab, Astrobee robots can use their propulsion systems to fly around in three degrees of freedom, riding on air carriages on a level granite surface, with a two meter square workspace. In the Micro-Gravity Test Facility (MGTF), a single Astrobee can ride on a 6-DOF motion platform, with a workspace similar to a large ISS module. The ground test units can be used to test your software or hardware at an early stage of prototyping, giving you vital performance information prior to finalizing your design. These facility services are typically available to guest scientists free of charge, subject to practical limitations of schedule, staff, and equipment availability.

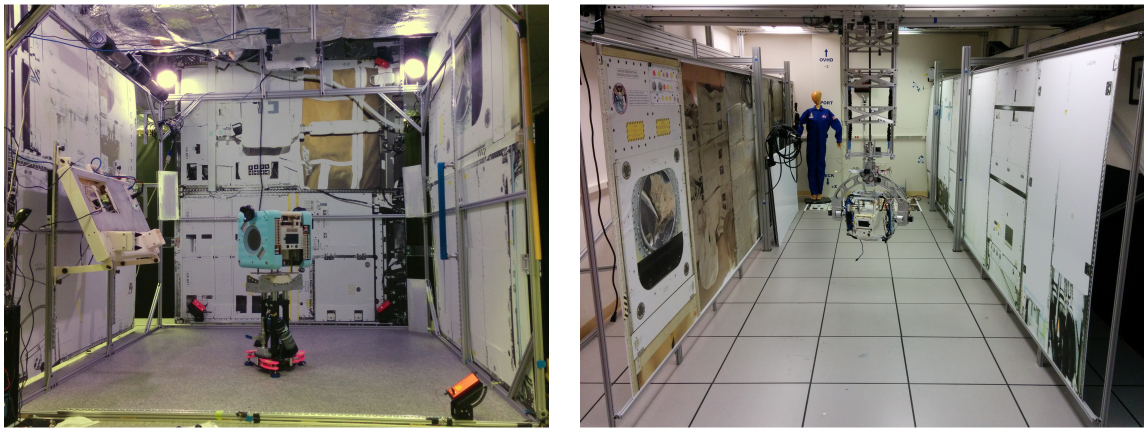

Figure 6: (left) The Granite Lab. (right) The MGTF.

In order to actually operate on the ISS, in general, some entity will need to sponsor you

[1] As of this writing (Aug 2017), the simulator is not yet released, and tutorials are still in development.

(i.e., vouch for the value of your research). One typical path to sponsorship is to win a NASA research grant based on proposing to use Astrobee for your research. Other U.S. research funding agencies, such as NSF and DARPA, can also sponsor you in this way. Foreign researchers are also welcome; your use of Astrobee would likely involve an international agreement between your country's space agency and NASA. Commercial research projects with their own funding can apply for ISS support through the CASIS organization. Once your payload is sponsored, the ISS program typically provides a wide variety of services (integration, testing, launch) free of charge. Please contact us for more information about paths to sponsorship.

If you are planning to fly your own hardware, you should be aware that there is an involved ISS integration process that starts when your concept is assigned an ISS Payload Integration Manager, and typically takes *at least* several months before your payload is approved to launch. Ideally the integration process starts early in your project before your detailed design is finalized. The process considers a wide variety of issues, from safety, to compatibility with other hardware on ISS, to any astronaut procedures and training needed to support your payload. The effort required for integration depends greatly on the complexity of your payload. Some integration issues are specific to Astrobee payloads, and we will work closely with the ISS program to streamline that part of your experience as much as possible.

The rest of this guest science guide (still under construction) will go into much more detail about the Astrobee platform and the guest science process. But remember, feel free to contact us if you have further questions. The Astrobee Research Facility exists to support you!

2 Platform Overview

In future releases of the Guest Science Guide, this section will summarize the Astrobee platform capabilities. For the time being, please refer to our previous overview papers covering the concept of operations [BBF+15] and preliminary design [SBB+16]. Although some details have changed, these papers still provide a useful overview.

3 Guest Science Lifecycle

The Astrobee Research Facility welcomes scientists from around the world to work with the Astrobee platform. It's been designed from the bottom up to best support a Guest Science Program. This program allows anyone with new science to interface with the facility team as well as the Astrobee robot hardware.

A good first step is to reach out to the facility and begin communicating your interests. Contact information can be found on our website at `www.nasa.gov/astrobee`. The sooner the Astrobee Research Facility and the ISS Program know about potential paylaods, the better for scheduling of resoures both at NASA Ames and on ISS.

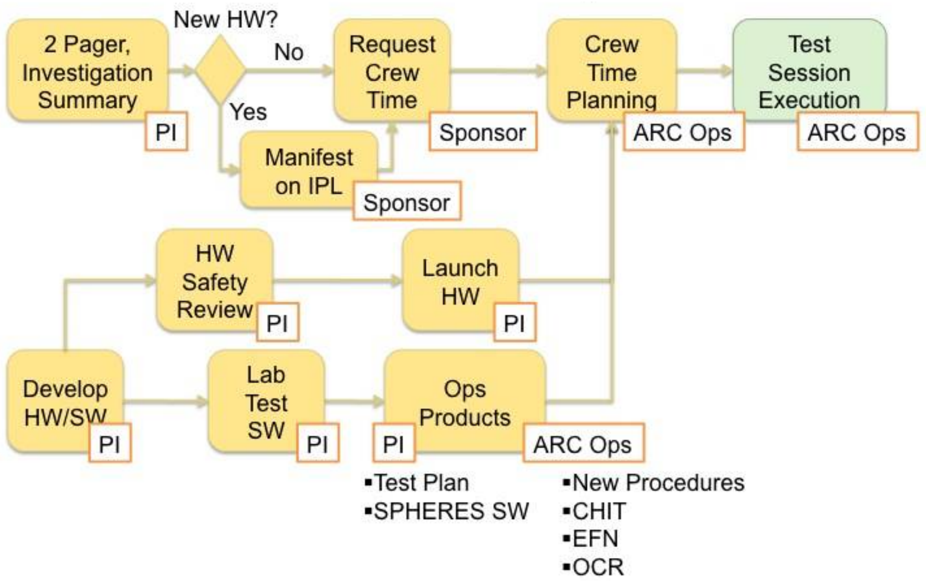

Figure 10: ISS Tactical Phase

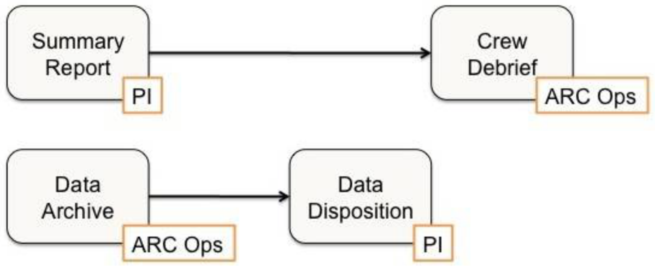

Figure 11: ISS Post-Operations Phase

costs. NASA HEOMD-AES has committed to ensuring the Astrobee Research Facility is available for use. So in all likelihood, both of these costs can be transparent to the PI. In the rare case that payload scope and integration effort is big enough, it is possible that the Astrobee Research Facility would require augmented funding from the PI.

There are several ways a payload gets added to the official Integrated Payload List (IPL), which is what kicks off the ISS Program support and the assignment of an ISS Payload Integration Manager (PIM). The ISS Technology Demo Office (TDO) is in charge of approving all NASA sponsored payload science on ISS. CASIS (for privately funded efforts) and US Defense (DOD) sponsored efforts are other ways. It's expected that other space agencies involved in the ISS Program (ESA, JAXA, etc.) have their own processes for doing this.

A payload project timeline can vary between six months to two years. It's split between four phases; Strategic, Tactical, Operations, and Post-Operations as described above.

Notional Schedule:

- Requirements development (2 months)
- Design (6 months)
- Integration & hardware delivery (8 months)
- Simulation & software (4 months)
- Science plan (4 months)

Example list of deliverables:

- All flight units shipped for integration and launch to ISS
- Copy of Final Acceptance Tests data package (including all verification results)
- Payload Safety Review Panel (PSRP) Phase 0/1 Safety Data Package (SDP) at Preliminary Design Review (PDR)
- PSRP Phase 3 SDP at hardware delivery
- Crew training materials, if any
- Crew procedures
- ISS Test Session reports
- Final Report

Designing payload hardware to meet ISS safety requirements can be costly, and it is best to account for it early in the design process. It should be possible to get a hold of the actual safety requirements documents (SSP 57000), but going through them and then figuring out what does and doesn't apply to your specific payload can be a lengthy process. The Astrobee Research Facility team has gone through the process a few times, so it would be wise to meet with them to discuss design concepts. The entire payload project and process involve a close partnership between the SPHERES/Astrobee team and PI team.

4 Software Interfaces and Communication Protocols

4.1 Software Hosting Environments

4.1.1 Free Flyer High-Level Processor Environment

When you want to run your own software onboard an Astrobee robot, creating a guest science Android app on the High-Level Processor (HLP) is usually the first place to start.

The High-Level Processor is a quad-core processor running Android/ARM that is primarily dedicated to guest software. It also manages some human-robot interaction hardware (speaker, microphone, touch screen) and off-loads some compute-intensive tasks from the other processors (e.g. compressing SciCam HD video).

The nominal concept of operations for guest science is built around an Android Java app running on the HLP. The operator control station provides features for uplinking, managing, executing, and commanding guest science apps. HLP guest apps have several capabilities:

- **Command the robot.** The Astrobee Command Dictionary (IRG-FF022) defines a set of high-level commands that are available to the guest science, as well as to the operator control station, and any software running on the ground. These high-level commands focus on common tasks such as "move to specified location", and do not provide access to all hardware capabilities, such as high-rate force/torque control. Your commands will be forwarded as ROS messages to Astrobee's built-in software.

- **Subscribe to robot telemetry.** Astrobee's built-in flight software is structured as a group of ROS nodes that communicate via network messages, and your app can subscribe to any of this ROS message traffic down to the lowest level, subject to bandwidth limitations.

- **Communicate to the ground.** The HLP guest app can send arbitrary (small) data packets to the ground. Your app will publish its data within a ROS message, on a designated guest science telemetry topic. Astrobee's built-in software will then repackage the data as a DDS message and forward it to the ground.

- **Manage experiment life cycle.** HLP guest apps can define custom commands relevant to their specific experiment, and document these commands in a configuration file. These commands are then available in the Astrobee operator control station, both in the plan editor and in the guest science interface. This enables experiment planners and operators to manage experiments within the larger Astrobee ops context.

- **Human-robot interaction.** Some human-robot interaction components are managed by the Android OS on the HLP, including the touch screen, microphone, and speaker. Guest apps can access these components directly through standard Android APIs.

- **Payload communication.** If you have a hardware payload mounted on Astrobee, your guest app can talk to it via USB. See §4.1.2.

4.1.2 Free Flyer Payload Processor Environment

Your hardware payload may have its own avionics, with USB data connectivity to your guest app running on the HLP. Using the USB drivers in the HLP's Android OS, it is straightforward to treat the USB connection either as a serial device or as a network interface. If you plan to treat the USB connection as a network interface, your payload would either use its own network-over-USB drivers, or you may find it convenient to use an Ethernet-to-USB hardware adapter.

However you use the USB connection, it is up to your HLP app to act as a bridge between your payload and other interfaces such as robot commanding and telemetry and the control station guest science commanding and telemetry.

4.1.3 Other Free Flyer Processor Environments

Astrobee's primary goal is to be a highly capable, flexible platform that enables great guest science. Deploying a guest science app on the HLP is the easiest way to run your software onboard the robot, but in some cases it may not satisfy your needs.

For example, you may want to command the Astrobee propulsion modules or the arm with your own high-rate control loop. The command dictionary available to your guest science app does not provide the appropriate commands for that functionality, and in any case, the HLP environment and the robot's internal network aren't designed to provide the low latency you may need.

For cases like these, we plan to provide a process for you to make experiment-specific modifications to the Astrobee platform software running on processors other than the HLP. As much as possible, the custom software for all processors will be released open source, so you can write the modifications yourself, with guidance from the Astrobee Research Facility staff. In general, we expect this process to be more complicated than deploying a guest science app, and it is likely to require more testing prior to experiments on the ISS, especially if your changes affect more critical functions like basic mobility.

This is an advanced topic. If you think you need to modify the software on other processors, please contact us for further discussion.

4.1.4 Dock Processor Environment

The Dock Processor is a dual-core Linux/ARM processor (same as the Free Flyer Low-Level Processor). You could theoretically modify the Dock Processor software to support your experiment, but we don't realistically foresee any need for that.

4.1.5 Ground Data System Workstation Environment

You may need a software interface on the ground for controlling your experiment.

If your interface needs are simple enough, you can use the guest science tab of the Astrobee control station, as follows:

- **Commanding:** You can pre-define a set of custom commands for your guest app on the HLP, and register them in the control station's configuration file. During your experiment, operators using the control station can select which custom command to send from a menu, and also edit command parameters if needed.

- **Telemetry:** Your guest app can publish its telemetry on a guest science topic. These messages will be downlinked to the ground via the DDS network protocol. If your messages are formatted properly, the control station will display them to the operator.

If your interface needs are more sophisticated, there are two basic approaches you can follow:

- **Modify the control station:** The Astrobee control station software will be released open source. It is a modular Java application based on the Eclipse RCP framework, meaning you can easily add your own interface widgets without significantly modifying the baseline control station code.
- **Fully custom interface:** You can write your own stand-alone application that communicates with the Astrobee robots using the same DDS protocol as the Astrobee control station.

The Astrobee control station software can run in multiple hosting environments, ranging from crew laptops onboard ISS, to workstations in the ISS Mission Control Center (MCC) in Houston or the Astrobee Research Facility workstations in the Multi-Mission Operations Center at NASA Ames, to a workstation at your home institution.

If you plan to extend the control station or provide your own stand-alone ground application, you should consider which of these hosting environments you need to target. In future release of the guest science guide, we will begin to discuss some of the resulting operating system requirements, portability issues, and testing processes.

5 Hardware Interfaces

5.1 Mechanical Interface

Astrobee has three payload bays available for your use. Their structure is defined in the Astrobee Payload Mechanical Interface Control Description (A9SP-1500-M501).

There are two standard ways to attach to a payload bay:

- **Quick-release:** At the bottom of each payload bay, there are two clevises designed to be captured by a pair of quick-release levers that are part of your payload. We will provide a reference design for the levers that you can integrate into your payload design. This is the recommended approach, because it enables quick payload swapping with minimum crew time.
- **Bolts:** For payloads that can't use the quick-release mechanism for some reason, the payload bay also provides a pattern of four bolt-holes; your payload would need to include matching captive bolts.

5.2 Electrical Interface

We plan to release an Astrobee Payload Electrical ICD as a companion to the mechanical ICD (not available yet).

5.3 Thermal Interface

In a future release, this section will discuss guidelines for your payload's thermal design.

6 ISS Integration Considerations

In general, you will work with the ISS program to handle integrating your experiment with the ISS environment. The master reference document describing ISS integration requirements for payloads is SSP 57000. In this section, we briefly mention a few ISS integration issues that are specific to Astrobee.

6.1 Safety

Ensuring collision safety has been a major challenge for Astrobee development. You can benefit from our effort by keeping your payload entirely within the defined volume of the payload bay. In that case, Astrobee's corner bumpers will protect it from impacts with the walls.

If your payload needs to extend outside the payload bay for some reason, you will need to work with the ISS Payload Safety Review Panel to find some other approach for showing that your payload can't damage the ISS in an impact. (For example, you may need to provide your own padding.)

6.2 Crew Privacy

Astrobee payloads are unusual in that Astrobee can carry them anywhere in the ISS without crew supervision. Astrobee robots can approach sensitive areas such as the crew sleeping quarters and exercise equipment. If your payload includes sensors such as cameras or microphones that could impact crew privacy, you will need to work with the ISS crew office to clear your experiment and plan how to control your data collection, archiving, and distribution.

7 References

[BBF$^+$15] Maria Bualat, Jonathan Barlow, Terrence Fong, Chris Provencher, and Trey Smith. Astrobee: Developing a free-flying robot for the International Space Station. In *Proc. AIAA Space Forum*, 2015.

[CASM07] Soon-Jo Chung, Umair Ahsun, Jean-Jacques Slotine, and David Miller. Application of synchronization to cooperative control and formation flight of spacecraft. In *Proc. AIAA Guidance, Navigation, and Control*, 2007.

[EJN+17] Matthew A. Estrada, Hao Jiang, Bessie Noll, Elliot W. Hawkes, Marco Pavone, and Mark R. Cutkosky. Force and moment constraints of a curved surface gripper and wrist for assistive free flyers. In *Proc. Int. Conf. Rob. Autom. (ICRA)*, 2017.

[FMM+13] Terrence Fong, Mark J Micire, Ted Morse, Eric Park, Chris Provencher, Vinh To, DW Wheeler, David Mittman, R Jay Torres, and Ernest Smith. Smart SPHERES: A telerobotic free-flyer for intravehicular activities in space. In *Proc. AIAA SPACE*, 2013.

[MLS+16] R. Mantellato, E. Lorenzini, D. Sternberg, D. Roascio, A. Saenz-Otero, and H. Zachrau. Simulation of a tethered microgravity robot pair and validation on a planar air bearing. In *Proc. Tethers in Space Conference*, 2016.

[PAS+14] Allison K. Porter, Dustin J. Alinger, Raymond J. Sedwick, John Merk, Roedolph A. Opperman, Alexander Buck, Gregory Eslinger, Peter Fisher, David W. Miller, and Elisenda Bou. Demonstration of electromagnetic formation flight and wireless power transfer. *Journal of Spacecraft and Rockets*, 51(6):1914–1923, Sep 2014.

[SBB+16] Trey Smith, Jonathan Barlow, Maria Bualat, Terrence Fong, Christopher Provencher, Hugo Sanchez, and Ernest Smith. Astrobee: A new platform for free-flying robotics research on the International Space Station. In *Proc. iSAIRAS*, 2016.

[TMSOM12] Brent E. Tweddle, E. Muggler, Alvar Saenz-Otero, and David W. Miller. The SPHERES VERTIGO goggles: Vision based mapping and localization onboard the International Space Station. In *Proc. iSAIRAS*, 2012.

A Astrobee Technical Specifications

Property	Value	Notes
Size	12.5 inch cube	Each Astrobee can fit in a cube 12.5 inches wide. The soft bumpers are 0.5 inches thick, so the rigid structure is 11.5 inches wide.
Mass	10 kg	Reported value is the not-to-exceed mass requirement for Astrobee's minimum-mass "performance" configuration (only two of the four batteries installed, and arm removed). A final as-built mass measurement will not be available until after the certification unit is integrated. Of course, the mass will increase when more hardware is installed.
Operating time	~2 hours	Reported value is the approximate maximum sortie duration with four batteries installed, based on known battery energy limits and measured power consumption with a prototype. This value assumes the robot is constantly flying in teleoperated mode (streaming video to the ground). Battery life can be extended by perching, or cut shorter if more power-hungry components are active (e.g. flashlights, payloads).
Thrust (X axis)	0.6 N	Reported value is the minimum thrust requirement on Astrobee's "best" axis, which is the X axis or forward/aft motion direction. A final as-built thrust measurement will not be available until after the certification unit is integrated. This thrust value assumes the impeller is running at its maximum nominal RPM rate; when running in lower power modes, less thrust will be available.
Thrust (Y, Z axis)	0.3 N	Same caveats as the X-axis thrust.
Payload power voltage	14.4 VDC	Payloads draw power directly from the raw Vbatt power bus. Reported value is nominal voltage for 4S Li-ion cells. Payloads must be able to tolerate 11.0-17.0 VDC without being damaged. It is acceptable if a payload requires ≥12.5 VDC to operate normally.
Payload power max current	2 A	There is a 3 A current limiter upstream of each payload power connector. For the time being, we are giving payloads a 2 A max current guideline, so as to provide some margin and avoid power interruptions. Of course, greater payload power consumption shortens battery life.

Table 1: Astrobee specifications