IAC-18,B3,6-A5.3,1,x45274

ASTROBEE: CURRENT STATUS AND FUTURE USE AS AN INTERNATIONAL RESEARCH PLATFORM

Andres Mora Vargas[1], Ruben Garcia Ruiz[2], Peter Wofford[3], Vivek Kumar[3], Bryce Van Ross[4], Aric Katterhagen[5], Jonathan Barlow[6], Lorenzo Flückiger[6], Jose Benavides[7], Trey Smith[7], Maria Bualat[7]

[1]MEI, NASA Ames Research Center, USA, andres.moravargas@nasa.gov
[2]USRA, NASA Ames Research Center, USA, ruben.m.garciaruiz@nasa.gov
[3]Massachusetts Institute of Technology, USA: peterwof@mit.edu, vivekk@mit.edu
[4]California State University LA, USA, bvanros@calstatela.edu
[5]Wyle Labs, NASA Ames Research Center, USA, aric.j.katterhagen@nasa.gov
[6]SGT Inc, NASA Ames Research Center, USA, jonathan.s.barlow@nasa.gov, lorenzo.fluckiger@nasa.gov
[7]NASA Ames Research Center, USA, jose.v.benavides@nasa.gov, trey.smith@nasa.gov, maria.bualat@nasa.gov

The Astrobees are NASA's next generation free-flying robots for the International Space Station (ISS). In this paper we update Astrobee development status, describe its software, and the ground facilities used for testing. We also describe initial uses of Astrobees as a research and educational platform, including the Zero Robotics competition for students ages 12-18. Astrobees will replace the Synchronized Position Hold, Engage, Reorient, Experimental Satellites (SPHERES) on board the ISS. Astrobees will operate inside the ISS, where they will assist astronauts, ground controllers, and researchers. The Astrobee Robot Software is the open-source, Robot Operating System (ROS) based software in charge of Astrobees' autonomous operation. The Astrobee Robot Software can be used interchangeably with an Astrobee Simulator or as Astrobee's on board software, facilitating the robot's development. In addition to simulation-based testing, further evaluation is performed using the SPHERES "Granite Lab" and "Micro-Gravity Test Facility" (MGTF) at NASA Ames. The Granite Lab consists of a 3m x 3m granite table used to mimic microgravity conditions in three degrees of freedom (x/y/yaw) by mounting one or multiple Astrobees on mobile air bearings to eliminate friction. This facility also replicates a section of the ISS including lighting, handrails, and the Astrobee dock. The MGTF expands on the Granite Lab's capabilities by allowing an Astrobee to maneuver with 6 degrees-of-freedom (DOF). A gantry structure allows linear displacement in x-y-z, and a gimbal performs rotations about those axes. Motion is driven by a physics simulation that responds to simulated thrust command from the robot. The MGTF expands the capabilities to test Astrobee features at a higher fidelity, including path planning, obstacle avoidance, and computer vision-based mobility control algorithms. In addition to carrying out tasks to support ISS operations, Astrobee was designed to support a broad range of "guest science". Guest scientists will use Astrobees to test a variety of mobile payloads and sensors, as well as carry out experiments ranging from human-robot interaction studies to examination of fluid behavior in microgravity. Industry, academia, and government researchers can run their experiments using these free-flying robots. We describe several of the hardware and software payloads currently under development for Astrobee, and discuss the transition from SPHERES to Astrobee in the Zero Robotics program.

I. INTRODUCTION

In this paper we describe a comprehensive view of the Astrobee research platform as an ecosystem geared towards supporting one of its fundamental customers: guest scientists. This ecosystem integrates Astrobee's hardware (ground and flight units), software (on board and simulator), Ames Research Center's (ARC) experimental facilities as well as the International Space Station (ISS) with the guest scientists and their research.

A conceptual interaction between these components is presented in Figure 1 and represents a sequential approach to the development of experiments using the Astrobee research platform. The guest scientist first learns how the Astrobee Robot Software (ARS) [1] works and after becoming familiar with it, starts to develop payload code. The guest scientist then uses the Astrobee Simulator, a component of the Astrobee Robot Software, to evaluate the code. If satisfied with the results, the guest scientist can deploy the code and test on

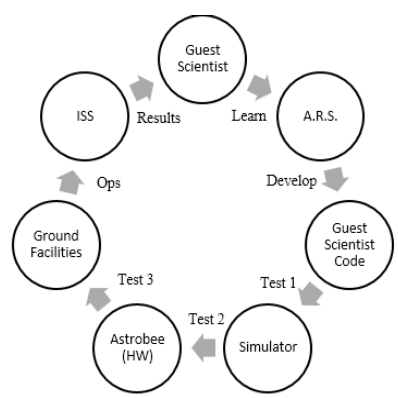


Figure 1. Guest science development life cycle conceptual sequence.

Astrobee robot hardware (HW) at a ground facility. Once the performance is shown to be satisfactory, the operations team uploads the code to the Astrobee robot(s) on board the ISS. Guest scientists have immediate access

Characteristic	Int-Ball	Astrobee	CIMON
Dimensions	15 cm dia sphere	31.75 x 31.75 x 31.75 cm	32 cm dia sphere
Estimated Weight	1 kg	8 kg	5 kg
Estimated Run Time	2 hours	2~4 hours	2 hours
Sensors	2 cameras, 3 ultrasonic range, IMU	IMU, 6 cameras, touchscreen, microphone	RGB, IR, Stereo cameras, ultrasonic sensors, microphones
Propulsion	12 electric fans & 1 reaction wheel	12 thrusters	14 fan outlets
Manipulation	N/A	One 3-DoF perch arm	N/A
Human-Robot Interaction capability	LEDs, HD camera	Signal lights, Touchscreen, perching arm, microphone, speaker, laser pointer, flashlight	LCD screen, speakers, microphone
Dock station	N/A	Yes	N/A
Payload Capable	N/A	Yes (3 hardware bays and software)	Yes (USB port)

Table I: Comparative summary between the three assistive free-flyers and their capabilities.

to their experimental data. In practice, the first steps would be combined: the guest scientist learns the ARS using the Simulator and the examples provided with it, then develop a guest science application, and tests it using the Simulator. The Astrobee Robot Software is the software in charge of Astrobee's autonomous operation; it is open-source and based on the Robot Operating System (ROS). The Astrobee Robot Software contains both the on-board flight software for Astrobee and the Astrobee Simulator (of its hardware and ISS environment). The ARS can be used interchangeably as an Astrobee Simulator or as Astrobee's onboard software, facilitating the development of guest scientist software.

Experimental ground facilities are another component of the Astrobee research ecosystem composed of the Micro-Gravity Test Facility (MGTF) and the Granite Laboratory. These facilities enable guest scientists to evaluate their experiments on the ground prior to launch. They are a fundamental step in validating the results obtained through simulation and determine if the software deployed on to the actual Astrobee performs as expected in a real-world environment.

Astrobee aims to support several types of research including manipulation, computer vision, and human-robot interaction (HRI). It also works as an educational platform through student competitions such as Massachusetts Institute of Technology's (MIT) Zero Robotics intended for middle and high school students [2]. The Astrobee ecosystem provides all the tools guest scientists need to enable them to easily create software that can be simulated and tested in real-world environments, run experiments on board the ISS and have those results quickly available.

II. RELATED WORK

As a research platform ecosystem, Astrobee integrates multiple shareholders within the context of microgravity robotics research [3], [4], [5]. NASA develops and supports hardware and software that will operate on board the ISS and supports a variety of internal and external research and development programs through which industry and academia can innovate such as Small Business Innovation Research (SBIR)/Small Business Technology Transfer (STTR), Early Career Faculty Program, Early Stage Innovation Program, and Center for the Advancement of Science In Space (CASIS) grants.

Before Astrobee, MIT built SPHERES under Defense Advanced Research Projects Agency (DARPA) funding as a satellite testbed for the development of space technology. In 2006, MIT started the ISS free-flyer Guest Scientist Program that was later transitioned to the NASA Ames Research Center in 2010 to be managed as an ISS National Laboratory Facility. It went on to support dozens of investigations within a variety of fields. It has operated as an educational platform through MIT's Zero Robotics competition, and has been a model other space agencies like the Japan Aerospace eXploration Agency JAXA could replicate [6], [7].

Since 2016, JAXA has developed and deployed their JEM Internal Ball Camera or Int-Ball, an experimental, autonomous, self-propelled, and maneuverable ball camera, in the Japanese Kibō module of the ISS. Int-Ball weighs 1 kg, is 15 cm in diameter, and is propelled by an array of 12 small electric fans mounted on the ball's outer surface [8], [9]. The German Space Agency (DLR), Airbus, and IBM have been collaborating to develop and deploy the Crew Interactive Mobile Companion (CIMON) [10]. CIMON aims to be an AI-based assistant for astronauts on the ISS and is equipped with cameras, ultrasonic sensors. It propels itself using fans. Table I provides a summary of the three assistive free-flyers and their capabilities.

Experimental ground facilities used to evaluate microgravity research can be found around the world [11, 12, 13, 14, 15, 16]. Free-flyer testbeds are also used in institutions in the U.S. (University of Southern California, MIT, Stanford, Georgia Tech, Rensselaer Polytechnic Institute NY, the Naval Research Laboratory, and the Naval Postgraduate School) as well as in different NASA centers (Marshall Space Flight

Characteristic	ARC (MGTF)	ARC (Granite)	Stanford	DLR (TEAMS)	JAXA
Dimension	3 x 1.5 x 1m	2 x 2 x 2 m	5 x 2 x 2 m	Two 4 x 2.5m	2 x 1 x 1m
Materials	Gantry, Gimbal	Granite	Granite	Granite	Granite
Simulated Environment	US lab module at ISS	US lab module at ISS	Non-specific	Non-specific	Non-specific
Ground truth sensors	HTC Vive	HTC Vive	Vicon system	AR Tracking	Optitrack

Table II: Comparison among microgravity ground testing facilities.

Center, Ames Research Centers) and around the world (DLR [17], JAXA, and Tohoku University). The main differences between these facilities are their dimensions, infrastructure materials, hardware and software used to emulate the microgravity environment, and ground truth sensor systems. Ames Research Center's facilities aim to provide detailed simulation of the ISS environment in both the MGTF and the Granite Lab. This is a key factor to help develop robust Astrobee software and to provide to guest scientists experimental environments consistent with those found on the ISS. Table II summarizes a comparison between the MGTF and Granite Lab and other facilities around the world.

III. ASTROBEE

Astrobee includes a variety of components that can be classified in two major categories: on-orbit (free flyers, docking station, and crew control station), and ground (control stations, ground server, DDS server) [18]. As of the writing of this paper, Astrobee is under development. The docking station has been delivered to Johnson Space Center for launch in November 2018. Integration of the Astrobee free flyers is underway, with targeted launches in Spring 2019. The crew and ground control stations, and ground server are ready for operation and the DDS server will be installed in the Huntsville Operations Support Center (HOSC) at Marshall Space Flight Center shortly.

The Astrobee free flyer has two interchangeable propulsion modules: one on the left and one on the right side of the robot. Each module has a plenum which is pressurized by a single centrifugal impeller. The impeller pulls in air through a central intake and feeds six exhaust nozzles around the edges. The modules are arranged so that their impellers rotate in opposite directions at the same speed, cancelling unwanted drag torque and gyroscopic moment disturbances. It also liberates the volume between them to accommodate guest payloads.

Astrobee has three main onboard processors: the Low-Level Processor (LLP), Mid-Level Processor (MLP), and the High-Level Processor (HLP). These processors are Commercial Off-The-Shelf (COTS) devices based on ARM architecture in a System-On-Module (SOM) format. LLP is in charge of the propulsion system, and reliably runs the high-rate control software. MLP software includes the compute-intensive machine vision and path planning algorithms. HLP hosts guest science software as Android PacKages (APKs), and is also in charge of streaming HD video compression and touchscreen interaction. The LLP, MLP, and HLP communicate with each other and with the dock via Ethernet. The Astrobee Robot Software runs on the HLP, MLP, LLP, and Dock's processor. It is modular and distributed; it uses a service-oriented robotic architecture style, and makes use of the Robot Operating System (ROS). The LLP and MLP run Ubuntu Linux 16.04, and the HLP runs Android (Nougat 7.1).

Each Astrobee uses six COTS external cameras: NavCam (mapping and operator awareness), DockCam (docking maneuvers), PerchCam (perching operations), SciCam (HD video), HazCam (obstacle detection), and SpeedCam (redundant velocity estimates). As for manipulation capabilities Astrobee has a compliant, detachable, 3-DOF perching arm [19], [20]. It is located on Astrobee's aft side allowing it to grasp ISS handrails and dwell for extended periods, reducing power consumption and interference with ISS operations. It could also support future manipulation and HRI research.

Astrobee is able to navigate inside the ISS thanks to its on-board mapping and localization algorithms. Mapping is based on a sparse mapping pipeline, which localizes via detection and matching of visual features based on a tree database. A detailed explanation of the map building and localization processes are presented in [21] and [22]. Astrobee's navigation and control tasks are distributed between the MLP and LLP. The MLP hosts the navigation components of the Astrobee Robot Software as ROS nodes handling trajectory generation and modification. The LLP hosts the controller that calculates force/torque commands to be sent to a Force Allocator Module (FAM), which in turn calculates the thruster's servo positions producing PWM outputs for each servo at the propulsion module.

From a high-level perspective, Astrobee has two main communication categories: internal and external. Internal communications are based on the ROS communication framework, e.g. messages (data distribution), services (simple instantaneous requests), and actions (complex longer requests). External communications between Astrobee on the ISS and ground control or other Astrobees make use of NASA's Robot API Delegate (RAPID) and the Data Distribution Service (DDS) middleware. Astrobee communicates commands, telemetry, and video through the ISS data network and the Ku-band satellite downlink. In order to avoid dropouts while transferring large files, Astrobee communicates via Ethernet connection with its dock station. Finally, a Dock Station serves as a recharging and communication station. It has two berths, each providing power and Ethernet connectivity to one Astrobee. When docking, an Astrobee autonomously approaches its berth using visual servoing relative to fiducials mounted to the dock. A system of conical lances (on the berth) and cups (on the robot) guides the final mating, accommodating up to ~1 cm of alignment error. The berth connector has 20 spring-loaded pogo pins that contact matching pads on the robot side. Compliance in the pins accommodates any remaining alignment error.

IV. SIMULATOR

Guest scientists can run experiments with Astrobee by writing Android applications and deploying them to the HLP. Initial evaluation will be possible through the Astrobee Simulator, which simulates Astrobee operating inside the ISS. In essence, it is a set of plugins for Gazebo, an open-source robot simulator that mimics the behavior and features of the actual hardware acting as replacements for the true hardware. It also contains some Gazebo-based world representations that emulate the environment where Astrobee will perform its activities, both in the ISS and the ground facilities.

Figure 2 shows that while the FFSW uses drivers to interact with actual hardware, when it comes to simulation, these drivers are replaced with Gazebo plugins that interact with an open-source Gazebo dynamics engine. The rest of the FFSW remains the same, including the *hardware messages* used in the intra-communication through the ROS network as well as the FFSW with all its different internal packages. The Astrobee Simulator is meant to run headless, i.e. a simulation execution will not show any graphical interface. It makes sense especially if an instance of the FFSW running in a server in the cloud for testing purposes is desired, such as for future MIT's Zero Robotics Competition games. However, it can also run simultaneously with several graphical interfaces. Gazebo enables a high-fidelity rendering of the simulated world and basic interaction through its client interface. The FFSW also provides several graphical interfaces for debugging, commanding and telemetry reading purposes.

The simulator does not need additional hardware since all the code running in the LLP and MLP is launched in the computer hosting the simulation. An exception is the HLP: the simulator does not consider this last processor. All code addressed to it will have to run inside a separate Android emulator or a developer Android board connected to the main simulator. In spite of emulating the hardware with high fidelity, the simulator still has some limitations. Although it does simulate Astrobee's cameras, it does not simulate any vision-based localization. However, this approach provides major performance gains. Also, when running

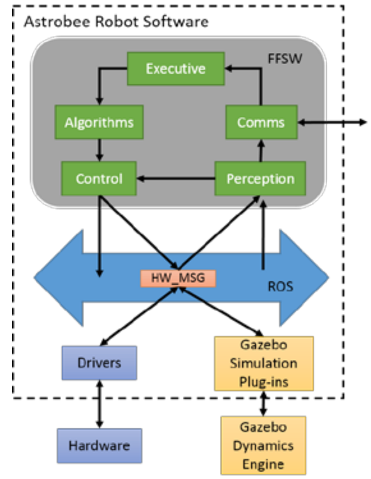


Figure 2. Astrobee Robot Software relationship with hardware and simulation.

over real time, DockCam and NavCam are disabled because of the large overhead required to simulate their images.

V. GROUND FACILITIES

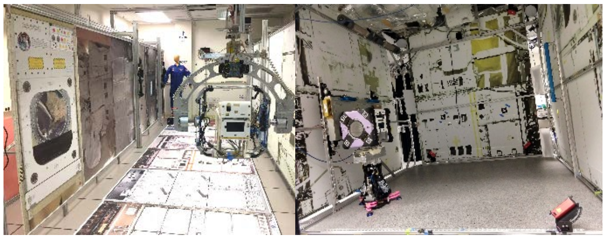
Ames Research Center hosts two different facilities: the Granite Laboratory and Micro-Gravity Test Facility (MGTF) as shown in Figure 3. Both facilities provide different experimental capabilities and can run experiments that may complement each other.

V.I Granite Laboratory

The Granite laboratory consists of a 3m x 3m granite table complying with ISO 10012, ANSI/NCSL Z540-1, ISO/IEC 17025, and has been certified with a surface accuracy of 0.0004 inches. It is used to mimic microgravity conditions in three axes (x/y/yaw) by mounting one or multiple Astrobees on mobile air bearings bases to eliminate friction. These bases use CO2 tanks providing an experimental continuous time of up to 15 minutes. The Granite lab also hosts a working certified Astrobee Dock station as well as walls with ISS pictures that help increase the fidelity of the simulated ISS environment. Further, this facility has lighting control such as the intensity, colorimetry, and shape of the lighting source enabling a multitude of experiments such as mapping an environment with changing light conditions [21]. The Granite lab also incorporates in its walls the same handrails present at the ISS and incorporates ground truth localization equipment currently based on a customized HTC Vive Sense system with with sub-millimetric precision when its tracker is static [23].

V.II Micro-Gravity Test Facility

The Micro-Gravity Test Facility (MGTF) is a laboratory at NASA Ames Research Center hosting a



(a) (b)

Figure 3. (a) Micro-gravity Test Facility, (b) Granite Laboratory.

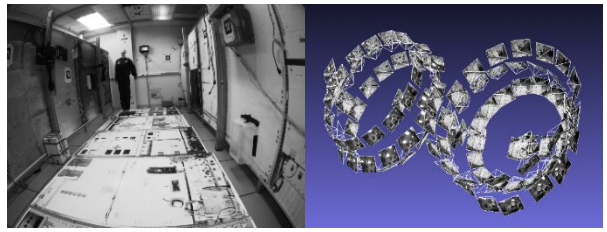
gantry and gimbal structure that provides Astrobee with 6 DoF movement capabilities. First built to support the Personal Satellite Assistant (PSA) [3] project in 2003, the MGTF has a gantry that achieves linear displacement in x-y-z, and a gimbal performs rotations about those axes. Motion is driven by a physics controller that responds to simulated thrust commands from Astrobee. The MGTF expands the capabilities to test at a higher fidelity the free-flyer's features from path planning and obstacle avoidance to computer vision based mobility control algorithms.

V.III Facilities Complementary Usage

The Granite lab and the MGTF are complementary to each other as they provide different capabilities and impose different limitations on Astrobee. The Granite lab is a smaller environment and restricts Astrobee to movement on x, y, and yaw. These characteristics limit its mobility to a smaller area but improves research focused on fine-tuned movement control, grasping and perching of handrails, docking maneuver validation, and mapping and localization experiments.

An important distinction between the Granite lab and the MGTF is how Astrobee moves. At the Granite lab, Astrobee is capable of propelling itself using its propulsion module. Additionally, the base it is attached to can be modified such that Astrobee has a fixed rotation around a pitch/roll angle while it moves on x, y, or yaw directions. At the MGTF however, Astrobee is mounted inside the MGTF's gimbal structure without its propulsion module: the MGTF then follows the commands of the free-flyer.

The MGTF represents a larger volume of the ISS US Laboratory allowing experiments to have a longer range of motion, to reach Astrobee's maximum accelerations and velocities, and to be only time-limited by the free-flyer's batteries charge. Currently, an experimental limitation the MGTF has due to hardware safety is that no perch arm experiments can be carried out in it. As Astrobee relies in its motion on the MGTF's gantry and gimbal system, an error in the execution of the free-flyer commanded motion during a perching or docking maneuver can severely damage the arm or the mockup.



(a) (b)

Figure 4. (a) MGTF picture taken with Astrobee's NavCam. (b) 3D visualization of a MGTF map based on NavCam images.

Both the Granite lab and the MGTF can run experiments from guest science payloads using Astrobee's payload bay but only the Granite lab can support those requiring manipulation similar to the free-flyer's perch arm.

Examples of applications using these facilities include HRI experiments [24] [25], environment mapping using Astrobee's onboard cameras, and Astrobee localization within that environment. Figure 4 (a) shows the MGTF viewed from the Astrobee's NavCam and Figure 4 (b) shows a 3D visualization of a MGTF map generated using Astrobee's NavCam.

VI OPERATIONS

Before the Astrobee platform on the ISS is fully operational to perform science, a series of functional checkouts called Commissioning Activities, must first be done. After commissioning is complete in summer 2019, Guest Science can begin; the first major step to performing science on the ISS is to become an official payload on the ISS Integrated Payload List (IPL). There are several ways to integrate guest science on the Astrobee platform; the ISS Technology Demonstration Office (TDO) and CASIS are two main avenues. Guest scientists are encouraged to contact the Astrobee team early to explore possible science, and to better understand the integration process. Once an official payload is on the IPL, the Astrobee Ops and Engineering Teams can work more closely with the Guest Science Team; every payload will have unique requirements and needs. The Astrobee Ops Team will also work with the assigned Payload Integration Manager (PIM) from Johnson Space Center (JSC) to coordinate the integration process of the Guest Science.

Flight operations begin on the ground and require a series of planning products (flight and ground procedures) that the Guest Science Team will develop along with the Astrobee Ops Team and PIM. Depending on the science, other ground products may be necessary such as Operations Readiness Tests (ORT) and Engineering Readiness Tests (ERT). Crew training products may also be required in the form of OBT (On Board Training) videos [26]. After all ground products are in place, and flight products have passed approval, the guest scientist work begins on the ISS. The Astrobee Ops Team will work with Marshall Space Flight Center (MSFC) to plan and schedule operations during which the science will be conducted. Depending on the science being performed, the Guest Science Team may have live data feeds, or will receive data post ops in a private data pipeline, or from a secure data transfer from the Astrobee Ops Team. The guest scientists may also have the capability to participate in the live operations of their science.

VII. GUEST SCIENTISTS

The Astrobee research platform revolves around guest scientists and its guest science is built on the successful SPHERES Research Facility. Guest scientists will pursue their own funding to develop their experiments, and receive in-kind support from the new Astrobee Facility e.g. integration guidance, ground testing facilities and the ISS Program by way of ISS payload sponsorship, e.g. launch services.

VII.I Guest Science Program

The Astrobee Facility, like the SPHERES Facility before it, is in charge of the Guest Science Program. One aspect of the SPHERES/Astrobee Facility is the SPHERES/Astrobee Working Group (SAWG) quarterly meeting. It provides an opportunity for information sharing across the SPHERES/Astrobee user community. This includes facility assessments on available resources, scheduling, and an open forum for developing new joint user opportunities. As Astrobee is about to replace SPHERES aboard the ISS in 2019, Astrobee Facility has been working on the transition between both research platforms and the new Guest Science Program. A guest scientist can become part of this program following the Guest Science Lifecycle (GSL).

The GSL starts by the guest scientist contacting the Astrobee Facility and communicating the guest scientist's interest in having an Astrobee payload or running an experiment with it. After this initial contact is established there are 4 additional phases: Strategic, Tactical, Operations, and Post-Operations. The Strategic phase defines high-level who, what, where, and how the guest scientist' research will be done. The Tactical phase includes technical planning, development, and evaluations. During the Operations phase, the guest scientist's science using the hardware and/or software developed during the Tactical phase is run on the ground (at the MGTF or at the Granite laboratory) or on the ISS. The guest scientist receives the experimental data and reports from the Operations phase during the Post-Operations phase.

Funding and the timeline for an Astrobee payload varies greatly depending on complexity. In order to operate on the ISS, the guest scientist should be sponsored as an official ISS payload. There are various sponsorship routes. If a guest scientist is government funded (e.g. NASA, National Science Foundation (NSF), DARPA), then he/she works with the ISS Program's Technology Development Office (TDO) for getting listed on the official Integrated Payload List (IPL). Foreign researchers are welcomed and require an international agreement between the guest scientist's country's space agency and NASA. Commercial guest scientists can apply for ISS support through the non-profit Center for Advancement of Science in Space (CASIS) organization. Typical timelines to have a payload approved to be launched are in the order of several months: from 6 months to up to 2 years. A summary of this scheduling with approximate time includes: Requirements development (2 months), Design (6 months), Integration and hardware delivery (8 months), Simulation and software (4 months), Science plan (4 months).

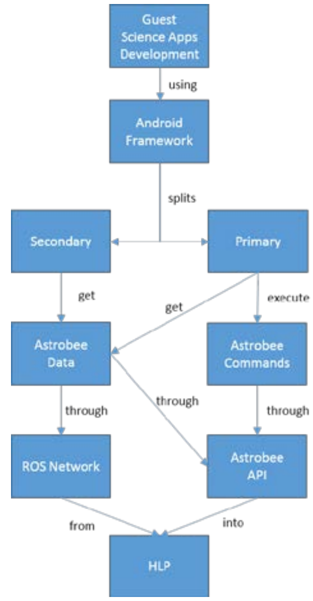


Figure 5. Guest Science application development: primary vs secondary application types.

VII.II Application Development

Guest scientists are required to follow a series of guidelines (hardware and software) to be able to receive approval to interface with Astrobee and use the ground facilities or the ISS. From a software standpoint, the guest scientist will interface with Astrobee via the HLP by creating an APK. These software packages are called Guest Science (GS) Applications and they utilize several libraries and follow a set of guidelines in order to command Astrobee and get data from it. Figure 5 shows the categorization and restrictions pertinent to GS applications.

GS applications should execute in background mode only. That is, they should not show any graphical display but for testing purposes. This makes sense when

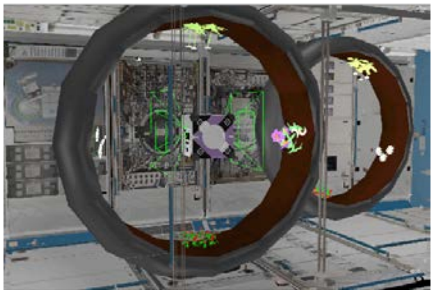


Figure 6. View of the MIT Zero Robotics proof-of-concept game using Astrobee Robot Software Simulator.

considering that astronauts will mostly not be looking at the screen in Astrobee during GS execution. GS apps are designed to take commands and retrieve data only through the Astrobee Ground Data System (a software interface for astronauts and ground controllers known as GDS), not through the touchscreen on Astrobee. There are two types of GS applications: primary and secondary.

Primary GS apps are allowed to command and get data from the robot. Since GS applications are running in the HLP and in connection to the other processors, they can access all data being generated by the rest of the FFSW components through the ROS network. They can also get a predefined set of data by executing calls to the Astrobee API. These applications are ideal for executing trajectories and other active experiments. To command the robot, primary applications must utilize a predefined set of functions available in the Astrobee API.

Secondary GS applications are allowed to get data from the robot only. They are not allowed to command Astrobee in any way. Secondary applications are also running inside the ROS network, therefore, they can access all data generated by the rest of the FFSW components the same way primary applications do. They may use the Astrobee API only to get data, never to execute commands. They may even leave aside the API and make use of ROS communications only. These applications are ideal for listener and monitoring software. They can also be used to get data generated by other applications running simultaneously.

VII.III MIT Zero Robotics Competition

MIT has been the guest scientist using this research platform most frequently through their Zero Robotics (ZR) competition. ZR is a competition aiming at inspiring middle and high school students to solve programming challenges using SPHERES in order to win a game and starting from 2020, using Astrobee.

In preparation for the transition from SPHERES to Astrobee, MIT is working to support this change through the development of a new game using the Astrobee's Simulator. An overview of the game process is as follows: 1) students submit their code that is interpreted with the production of an output file; 2) the output file is wrapped into a customized file containing game logic; 3) the customized file is passed on to a simulator that returns results; and 4) the simulation is reset and visualization is returned to the students.

A proof-of-concept game called Astrobotany has been developed to satisfy this process based on a customized version of the Astrobee Simulator. Figure 6 represents the visualization of the game where Astrobee's objective is to navigate large rotating rings with plants on their inner surfaces meant to simulate gravity. Astrobee can collect and deliver pollen to these plants and scores points based on proper pollination. One key challenge of this transition for MIT is to use a version of the Astrobee Simulator that can be deployed in the cloud, scale to thousands of students, and incorporate student submitted code following the guidelines presented in Section VII.II.

VII.IV Guest Scientists Payloads

Astrobee, as a robotic free flying research and technology development platform, has garnered a lot of interest [27]. At the time of the November 2017 (SAWG) quarterly meeting (https://www.nasa.gov/Astrobee), over 40 unique groups spanning industry, academia, and government had registered interest in utilizing Astrobee. Currently, six groups are actively working towards ISS activities.

The first to operate will be a NASA Radio Frequency IDentification-Enabled Autonomous Logistics Management (REALM) project called RFID Recon, where Astrobee will be equipped with an RFID reader and antennas to conduct autonomous inventories and searches. MIT's CASIS sponsored Zero Robotics project was described earlier. Another CASIS sponsored Astrobotic/BOSCH partnership is working on SoundSee, a Deep Audio Analytics (DAA) payload for autonomous acoustic monitoring on the ISS. The Naval Postgraduate School is working on Astrobatics, a software-only payload for demonstrating spacecraft robotic hopping on the ISS. Stanford is working on a Gecko Perching Gripper, a Gecko-inspired Adhesive gripper for automated logistics in space. NASA and JAXA are working together on a joint Astrobee/Int-Ball student programing competition for the Asia-Pacific Region.

VIII CONCLUSIONS AND FUTURE WORK

We have presented the components of the Astrobee research platform, the interaction between them, and how guest scientists are involved in that ecosystem. The current status of the Astrobee robot, its software including its simulator as well as the micro-gravity simulating research facilities at Ames Research Center are described to provide an overview of how guest scientists develop hardware and software payloads and the processes required to use these facilities.

Current efforts are geared towards the successful commissioning of the Astrobee platform on the ISS in

2019, the integration of Astrobee's first users, and continued sustaining support of the overall Astrobee facility. Future work may include a multi-robot framework supporting JAXA's Int-Ball, ESA's CIMON, and NASA's Astrobee to enable guest scientists to expand the range of experiments that can be carried out. Of particular importance, is improving the capability and performance of Intra Vehicular Activity (IVA) robots to perform payload operations and spacecraft caretaking in preparation for human exploration beyond Earth orbit.

IX REFERENCES

[1] L. Flückiger, K. Browne, B. Coltin, J. Fusco, T. Morse and A. Symington, "Astrobee Robot Software: Enabling Mobile Autonomy on the ISS," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Madrid, Spain, 2018.

[2] S. Nag, J. G. Katz and A. Saenz-Otero, "Collaborative Gaming and Competition for CS-STEM Education using SPHERES Zero Robotics," *Acta Astronautica,* vol. 83, no. February-March, pp. 145-174, 2013.

[3] G. A. Dorais and et al., "The Personal Satellite Assistant: An internal spacecraft autonomous mobile monitor,"," in *Proceedings IEEE Aerospace*, 2003 .

[4] W. Baker, Z. Kingston, M. Moll, J. Badger and L. E. Kavraki, "Robonaut 2 and you: Specifying and executing complex operations," in *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO),* Austin, TX, USA, 2017.

[5] S. Frederickson and et. al, "Mini AERCam: Development of a free-flying nanosatellite inspection Robot," in *Proceedings SPIE 5088 Space Sys. Tech. Oper.*, 2003.

[6] A. Saenz-Otero, A. Chen and et. al., "SPHERES: Development of an ISS Laboratory for Formation Flight and Docking Research," in *2002 IEEE Aerospace Conference*, 2002.

[7] A. Saenz-Otero, Design Principles for the Development of Space Technology Maturation Laboratories Aboard the International Space Station, Massachusetts: MIT, 2005.

[8] N. Tanishima, S. Mitani, S. Shigeto, Y. Matsumoto, Y. Arai, M. Goto and S. Suzuki, "Effective and Accurate Method for Ground and On-Orbit Verification of Control Systems for Free-Flying Robot With Low Thrust Force," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Madrid, Spain, 2018.

[9] S. Mitani, M. Goto, R. Konomura, Y. Shoji, K. Hagiwara, S. Shigeto and N. Tanishima, "Crew-Supportive Autonomous Mobile Camera Robot on ISS/JEM," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Madrid, Spain, 2018.

[10] C. Karrasch, V. Schmid, T. Eisenberg, J.-I. Buchheim, M. Biniok, G. Demoulin and C. Lombardi, "Project CIMON - the intelligent astronaut assistant Crew Interactive MObile companioN," Joint CSA/ESA/JAXA/NASA ISS Increment 57/58 Science Symposium, 2018.

[11] S. L. Liu, Z. H. Jiang, H. Li and Q. Huang, "A low-cost microgravity simulating system for motion control study of space robot," in *IEEE International Conference on Robotics and Biomimetics (ROBIO),* Shenzhen, China, 2013.

[12] F. Jin, "PD control of coordinated manipulation for multi-free-flying space robot," Ifost, Ulaanbaatar, 2013.

[13] H. H. King, T. Low, K. Hufford and T. Broderick, "Acceleration compensation for vehicle based telesurgery on earth or in space," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Nice, France, 2008.

[14] Y. Yuguchi, K. Nagaoka and K. Yoshida, "Verification of gait control based on reaction null-space for ground-gripping robot in microgravity," in *IEEE International Conference on Robotics and Automation (ICRA),* Stockholm, Sweden, 2016.

[15] R. MacPherson, B. Hockman, A. Bylard, M. A. Estrada, M. R. Cutkosky and M. Pavone, "Trajectory Optimization for Dynamic Grasping in Space using Adhesive Grippers," in *Results of the 11th International Conference, Field and Service Robots (FSR)*, 2018.

[16] P. Roque and R. Ventura, "Space CoBot: Modular design of an holonomic aerial robot for indoor microgravity environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Daejeon, Korea, 2016.

[17] M. Schlotterer and S. Theil, "Testbed for on-orbit servicing and formation flying dynamics emulation," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, Canada, 2010.

[18] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, E. Smith and Astrobee Team, "A New Platform For Free-Flying Robotics On The International Space Station," in *13th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS),* Beijing, China, 2016.

[19] I. W. Park, T. Smith, H. Sanchez, S. W. Wong, P. Piacenza and M. Ciocarlie, "Developing a 3-DOF Compliant Perching Arm for a Free-Flying Robot on the International Space Station," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM),* Munich, Germany, 2017.

[20] D. H. Lee, B. Coltin, T. Morse, I. W. Park, L. Flückiger and T. Smith, "Handrail Detection and Pose Estimation for a Free-Flying Robot on the International Space Station," *International Journal of Advanced Robotic Systems,* vol. 15, no. 1, 2018.

[21] P. Kim, B. Coltin, O. Alexandrov and H. J. Kim, "Robot Visual Localization in Changing Lighting Conditions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[22] B. Coltin, J. Fusco, Z. Moratto, O. Alexandrov and R. Nakamura, "Localization from visual landmarks on a free-flying robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Daejeon, Korea, 2016.

[23] M. Borges, A. Symington, B. Coltin, T. Smith and R. Ventura, "HTC Vive: Analysis and Accuracy Improvement," in *To appear in the proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS 2018).*, Madrid, Spain, 2018.

[24] P. A. Lasota, T. Fong and J. A. Shah, "A Survey of Methods for Safe Human-Robot Interaction," *Foundations and Trends in Robotics,* vol. 5, no. 4, pp. 261-349, 2017.

[25] D. Szafir, B. Mutlu and T. Fong, "Communication of intent in assistive free flyers," in *ACM/IEEE international conference on Human-robot interaction (HRI)*, 2014.

[26] M. Bualat, T. Smith and T. W. Fong, "Astrobee: A New Tool for ISS Operations," in *International Conference on Space Operations*, Marseille, France, 2018.

[27] J. Barlow, J. Benavides, C. Provencher, M. Bualat, M. Smith and A. Mora Vargas, "Astrobee Payloads," in *International Space Station Research and Development Conference*, Washington, DC, USA, 2017.