Astrobee Contribution to Collaborative ISS Free-Flyer Robot Paper

Andres Mora Vargas, Astrobee Facilities Team, NASA Ames Research Center [Astrion]
andres.moravargas@nasa.gov

As part of a joint paper with JAXA and ESA that illustrates the work done by NASA's Astrobee, JAXA's Intball, and ESA's CIMON free-flyers on board of the International Space Station (ISS) and to be submitted at the International Space Robotics Conference 2025 (iSparo 2025), the Astrobee team is providing a draft for the portion of the paper that refers to Astrobee development and activities on board the International Space Station (ISS).

Since 2006, NASA's Synchronize, Position, Hold, Engage, Reorient, Experimental Satellites (SPHERES) supported Guest Science research on the International Space Station (ISS) and became one of the most-used ISS payload facilities (Ref 1-3). NASA's three Astrobee free-flying robots (Bumble, Honey, and Queen) became the next generation of free-flying robots on board of the ISS and in 2019 they replaced SPHERES. Astrobee improved the capabilities SPHERES offered to the community by increasing autonomy and therefore reducing the amount of crew time required for experimental setup or regular operation; it does not depend on additional infrastructure on the station to navigate as it has a vision-based navigation system; it has been able to provide Guest Scientists with up to 3 hours of continuous operation and enables them to attach different types of payloads thanks to its three payload bays. The top aft payload bay for instance, may be used by the Astrobee's perching arm (Ref 4) to hold or "perch" on to the handrails around the station to either perform Guest Scientists experiments or serve as an assisting camera during crew operations.

The Astrobees are located at the Japanese Experimental Module (JEM) where the robots can recharge their Lithium-Ion batteries at its docking station (Ref 5-6). Each Astrobee measures an approximate 30x30x30cm and weight approximately 10 kg. It propulses itself through two battery-powered fans and twelve adjustable flow-rate nozzles. It has 6 different cameras (HazCam, Scicam, NavCam, DockCam, PerchCam, and SpeedCam) that enable the robot to navigate inside the ISS (NavCam, SpeedCam, HazCam), perch onto handrails (PerchCam), provide high-definition video for Guest Scientists experiments (SciCam) and autonomously dock and undock (DockCam) from the docking station. It enables human-robot interaction (HRI) research through its front and aft flashlights, laser pointer, LED lights that can be used to generate different lighting patterns, and a touch screen in its front face which displays a looped video of "eyes" conveying the working state of the robot to crew. Astrobee has three main computers, namely the Low-Level Processor (LLP), Mid-Level Processor (MLP), and the High-Level Processor (HLP). The LLP reliably runs high-rate control code in isolation from other software on a Wandboard Dual with 2 cores at 1.0GHz drawing 3W of power. The MLP oversees running most of the Astrobee Software including computer-vision based path planning on an InForce IFC6501 (Qualcomm Snapdragon 805) with 4 cores at 2.4GHz at 10W of power. Finally, the main function of the HLP is to run Guest Science data and managing the SciCam video compression, touch screen, and other less critical devices on a InForce IFC6601 (Qualcomm Snapdragon 820) with 4 cores at 2.2GHz on 10W of power.

The Astrobee Robot Software (ARS) is available open source (Ref 7, 8), and some of its key tasks are to localize throughout the Japanese Experimental Module (JEM) without extra infrastructure, precisely plan and execute motions without collision, provide control and monitoring from the ground with resilience to communication loss, support multiple control modes, including remote teleoperation, autonomous plan execution and on-board control by Guest Science (external researchers) software, autonomously dock for battery recharging and wired communications, autonomously perch on handrails to conserve energy while providing pan/tilt camera functionality, manage Guest Science software, hardware payloads, and user interface components, and simulation and data replay for testing. The LLP and MLP run Ubuntu (currently 20.04) because of its widespread use and the availability of software packages, notably Robot Operating System (ROS) (Ref 9). The HLP, however, runs Android (Nougat 7.1) because it is the only OS supporting some key hardware for Astrobee (the high resolution camera, video encoder and touchscreen). Android allows for the encapsulation of Guest Science software for the HLP as Android Packages (APKs), avoiding custom deployment and management methods.

Astrobees support several types of research including manipulation, computer vision, and human-robot interaction. As of March 2024, the Astrobee Facility has supported 16 unique Guest Scientists in 21 different projects, and it planned to support four additional Guest Scientists and seven additional projects between 2024 and 2025.

Guest Scientists may be from academia (Stanford, MIT), industry (KMI, Obruta), or government (NASA, JAXA) backgrounds. Academic Guest Scientists have carried out research in autonomous spacecraft rendezvous and docking (Ref 10), mapping and navigation (Ref 11), detecting changes in the ISS (Ref 12), and semantic mapping and localization (Ref 3). Astrobee has enable academic partners to demonstrate technologies for the first time in space robotics, for instance, the US Naval Research Laboratory's (NRL's) Autonomous Planning In-space Assembly Reinforcement-learning free-flYer (APIARY) team conducted the first ever reinforcement learning control of a free-flyer in space within a 3-month timeline.

Guest Scientists from the industry have successfully demonstrated their technologies leveraging Astrobee capabilities, some examples include Bosch's acoustic maps (ref 15), KMI's on-orbit servicing, assembly, and manufacturing (OSAM) using its large gripper to capture a tumbling target, or Canadian corportation Obruta which builds software for spacecraft rendezvous, proximity operations, and docking (RPOD) to increase the sustainability of the space industry through in-space servicing such as refueling, life extension, or debris removal. Over the course of two years, Obruta worked with NASA Ames personnel to develop experiments using the Astrobee facility to mimic real spacecraft RPOD scenarios, such that Obruta could validate its RPOD software in a representative in-space environment. Over four experiments on the International Space Station, using two Astrobees (Honey and Bumble) Obruta developed progressively-more-complex mission scenarios to test its RPOD software, which resulted in a final experiment on May 27, 2025 where Honey autonomously docked with Bumble using Obruta's AI-based monocular vision-based software along with its custom guidance and control software. Astrobee has made possible such in-space validation that would otherwise not be possible for an early-stage company with a limited budget. The experimental data collected from Astrobee have proven extremely valuable for Obruta in terms of stakeholder trust in the technology, talent acquisition, commercial sales, and fundraising.

With international space agencies partners such as JAXA, Astrobee has been fundamental in its educational outreach role. Since 2019 JAXA and NASA have collaborated to carry out JAXA's Kibo Robotic Programming Challenge (Kibo-RPC, https://jaxa.krpc.jp). This international competition is intended for undergraduate university students, and it has positively impacted in 2024 alone, with over 2760 students across the Asia Pacific region. Kibo-RPC is a programming challenge where students from around the world participate in teams and after passing a series of simulation-based qualifying rounds at their countries, each country selects one team to participate in the final event where students' code controls an Astrobee robot to solve a challenge in real-time on board of the ISS with crew participation.

The following items illustrate some elements of Astrobee's design and development approach and some elements of its operation that did or did not prove to work as expected (Ref 1).

1. Early prototyping became a vital process that provided fundamental experience to develop relevant and verifiable requirements even before the formal review of system requirements review with stakeholders.
2. While initially considering adapting the previously developed Simultaneous Mapping and Localization (SLAM) code, a simpler vision-based navigation approach was adopted. Recognizing the ISS' controlled environment, the team chose to capture imagery data, build maps on the ground, and then upload them for Astrobee's use, resulting in more robust flight software.

3. Modular design was key for the Astrobee project, allowing parallel development and simplified integration. Astrobee achieved modularity through decoupled propulsion, distributed computing, and a modular flight software architecture. However, unforeseen interface issues, like EMI from propulsion and amplified acoustic noise from nozzle mounting, exposed limitations in modularity and led to project delays. Despite challenges, modularity improves maintainability and adaptability, as demonstrated by the ability to isolate and fix issues in individual components.
4. The ISS is a dynamically changing environment and maintaining reliable maps became a task requiring more frequent updates than initially estimated.
5. Until a map has good features over the entire area, maneuvers will always need to point periodically to areas with good features.
6. After several activities dedicated to gather imagery data towards mapping, less reliance on crew assistance was required and accomplished thanks to improvements in the localization upgrade from the initial extended Kalman filter method into the graph-base localizer.
7. Given the unexpected high frequency in ISS activities and the corresponding high amount of data collected (over 2.0 Terabytes up until March 2025), the Astrobee Facilities team had to develop a mechanism to quickly treat it and obtain the required clearance to distribute it back to the user.
8. Users that developed software for the MLP required much more laboratory testing and development time. Guest Scientists such as AstroSee and Astrobatics, who developed software running at the MLP also developed software targeting the LLP.
9. The HLP is indeed a useful and highly used processor. The HLP running Linux would have been easier for user integration and even off-loading the MLP. Users continue to have interest in running Linux on the HLP. MIT and Gecko projects used the HLP to send commands and telemetry.
10. Frequent anomalies during ISS operations drove most users to rely on the Astrobee team for command and control. Special data display requirements led to users needing telemetry directly.
11. Operator training for poor localization situations took longer than the planned one hour of training. The Astrobee facility is running a study to better understand the impact GDS has in the operator's cognitive load during an ISS activity; GDS Helper serves as a comparison baseline in this study.
12. The GDS on a station laptop is not needed when ground commanding is possible.
13. Human interaction elements are unlikely to be incorporated by a user not specifically studying interaction. Only JAXA's Kibo-RPC has used the laser pointer, flashlights, and signal lights as means of communicating the robot's state or intent during their competition. MIT-ZR has also used flashlights for this same purpose.

References:

1. T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, E. Smith, et al., “Astrobee: A new platform for free-flying robotics on the
2. International Space Station,” in Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, 2016.
3. S. Mohan, A. Saenz-Otero, S. Nolet, D. W. Miller, and S. Sell, “SPHERES flight operations testing and execution,” Acta Astronaut., vol. 65, no. 7-8, pp. 1121–1132, 2009.
4. I. W. Park, T. Smith, and J. F. Love, “Thermal design of Astrobee perching arm,” in IEEE/SICE Int. Symp. System Integration (SII), 2019, pp. 444–449.
5. Smith, T., Wheeler, D.W., Alexandro, O., Barlow, J., Battazzo, S., Benavides, J., Blair, J. Bualat, M., Carlino, R., Coltin, B., Cortez, J., Daley, E., Dille, M., Feller, J. Flückiger, L., Fong, T., Fusco, J., Garcia Ruiz, R., Goetz, R., Gogna, R., Hamilton, K., Hanson, R., Kanis, S., Kashani, A., Katterhagen, A., Kim, H.., Kim, Y., Koss, B., Lee, D.., Dongmeng, L., Love, J. F., Lum, J.Q., Ma, R. J., Mai, N, McIntyre, M., McLachlan, B., Mora Vargas, A., Moratto, Z., Morr, D., Morse, T., Nakamura, R., Park, I., Paulson, G., Priscal, C., Provencher, C., Pacis Rius, E., Rogg, A., Sanchez, H., Sharif, K., Smith, E., Snyder, C., Soloway, D., Symington, A., Talavera, R. O., Tardy, A., To, V., Torres, R. J., Wu, S., Yoo, J., Zuniga, A.: “Astrobee: Free-Flying Robots for the International Space Station”, IEEE Transactions on Field Robotics, Special issue “Space Robotics”, 2024. Pending final review.
6. A. Mora Vargas, J. Benavides, J. Barlow, H. Orosco, S. Doi, R. G. Ruiz, R. Carlino, J. Cortez, A. Katterhagen, S. Kanis, B. Coltin, R. Soussan, and K. Hamilton, “Astrobee’s multi-year activities at the international space station’s Japanese experimental module,” in Proc. Int. Aeronautical Congress (IAC), 2022.
7. K. Albee, M. Ekal, and C. Oestreich, “A brief guide to astrobee’s flight software,” 2020.
8. Astrobee robot software,” https://github.com/nasa/astrobee, 2025, accessed: 2025-06-04.
9. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in Proc. Int. Conf. Robotics and Automation (ICRA) Workshop on Open Source Software, 2009.
10. K. Albee, C. Oestreich, C. Specht, A. Teran Espinoza, J. Todd, I. Hokaj, R. Lampariello, and R. Linares, “A robust observation, planning, and control pipeline for autonomous rendezvous with tumbling targets,” Frontiers in Robotics and AI, p. 234, 2021.
11. M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, “HTC Vive: Analysis and accuracy improvement,” in Proc. Int. Conf. on Intelligent Robots and Systems (IROS), 2018.

12. H. Dinkel, J. Di, J. Santos, K. Albee, P. V. Borges, M. Moreira, R. Soussan, O. Alexandrov, B. Coltin, and T. Smith, “Astrobeecd: Change detection in microgravity with free-flying robots,” Acta Astronautica, 2024.