Astrobee Overview

System, Status,

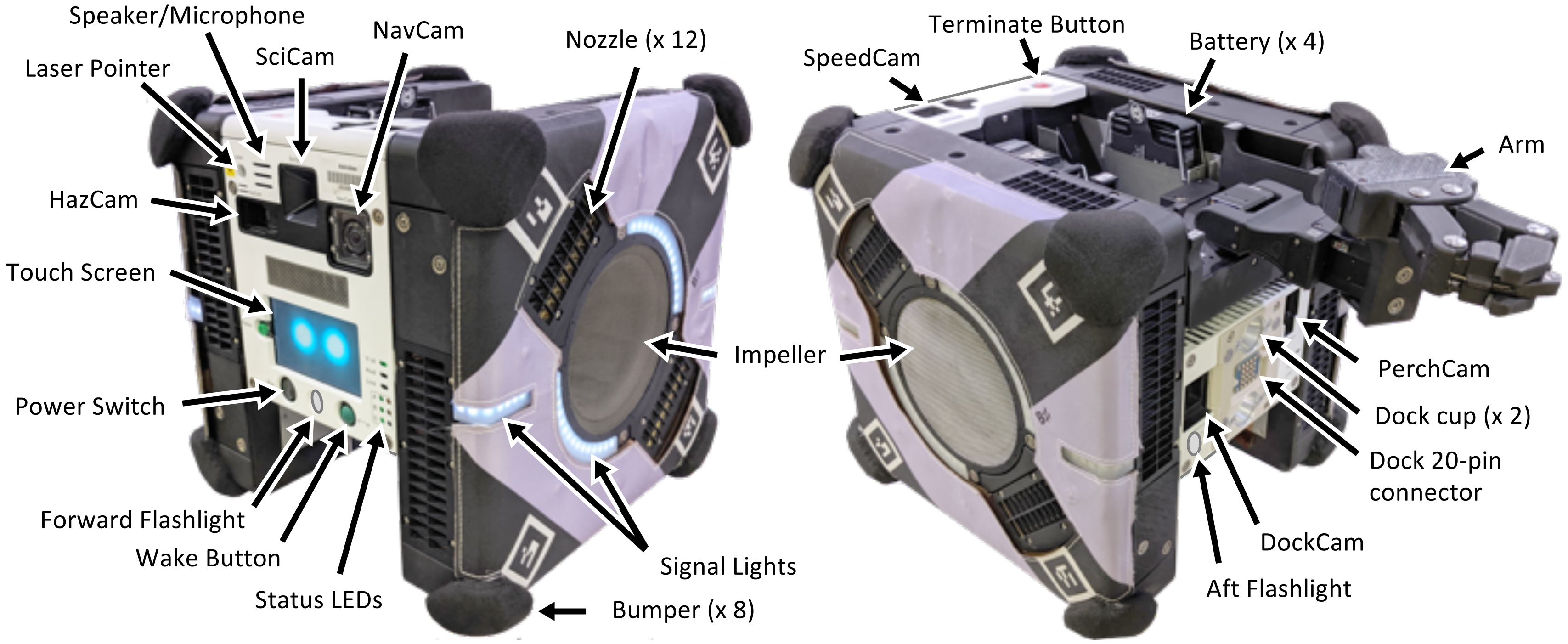
Performance, and specification

November 24th, 2019

Astrobee Objectives

- Provide a microgravity robotic research facility in the ISS US Orbital Segment (USOS), which will replace the existing SPHERES facility
- Provide remotely operated mobile camera views of the ISS USOS to enhance the situation awareness of mission control
- Perform mobile sensor tasks in the ISS USOS

System Description - Hardware

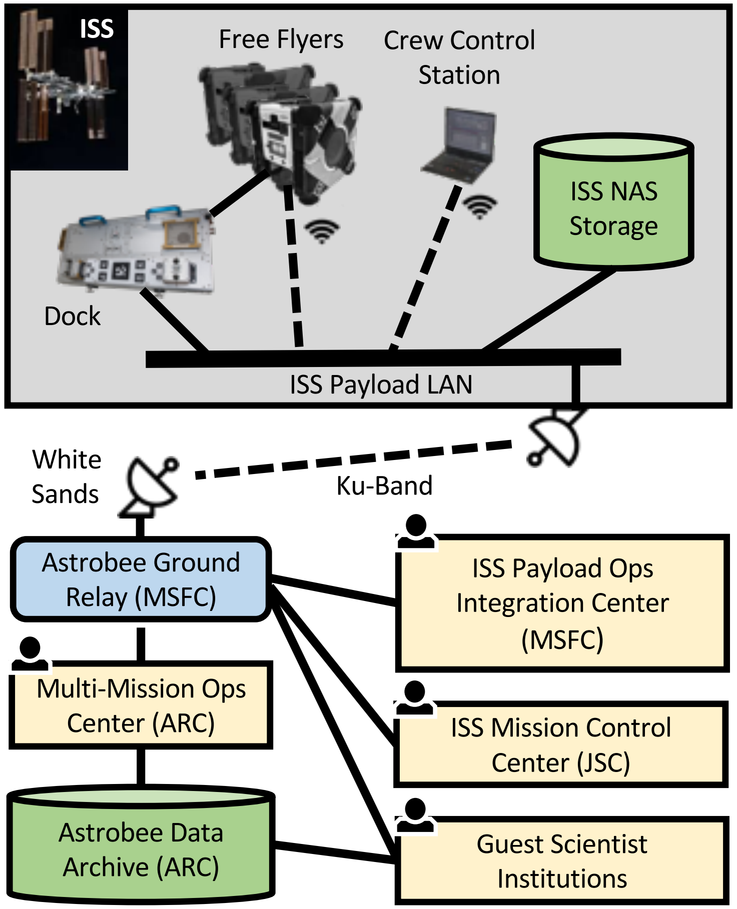


- Free flying robot inside the ISS
- 32 cm wide, ~9.1kg (2 batt., no arm)
- All electric + fan-based propulsion
- Robot arm for “perching”, ~1kg

- Three smartphone computers
- Three payload bays for expansion
- Microphone not currently enabled

System Description - Communications

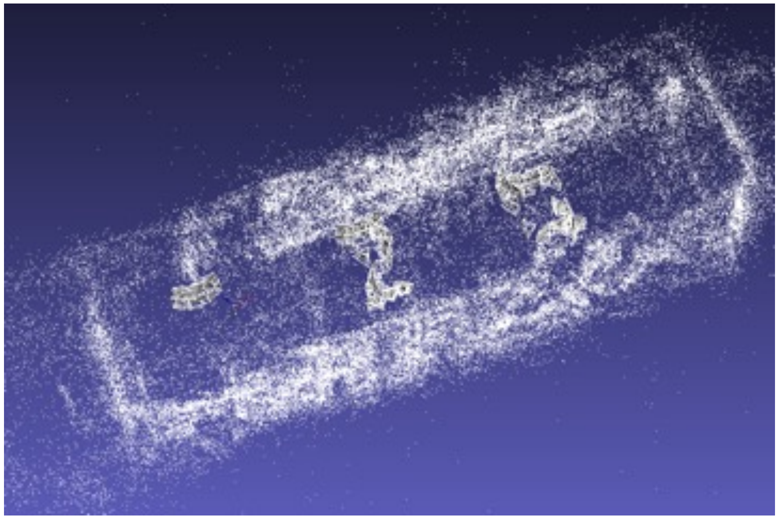
- Communicates through ISS WiFi when flying
- Single telemetry/video stream to ground
- Multiple ground stations can connect through server
- Large file transfers and software updates through Ethernet on the dock


ISS
Free Flyers
Crew Control Station
ISS NAS Storage
Dock
ISS Payload LAN
White Sands
Ku-Band
Astrobee Ground Relay (MSFC)
ISS Payload Ops Integration Center (MSFC)
Multi-Mission Ops Center (ARC)
ISS Mission Control Center (JSC)
Astrobee Data Archive (ARC)
Guest Scientist Institutions

Astrobee communications path

System Description - Navigation

- Vision-based navigation
  - Compares features with on-board a priori map
  - Incorporates inertial measurements
- Fiducials used for autonomous docking
  - Requires approximately 1 cm position accuracy
- Visual odometry
  - Robot can continue to navigate where no map features are recognized

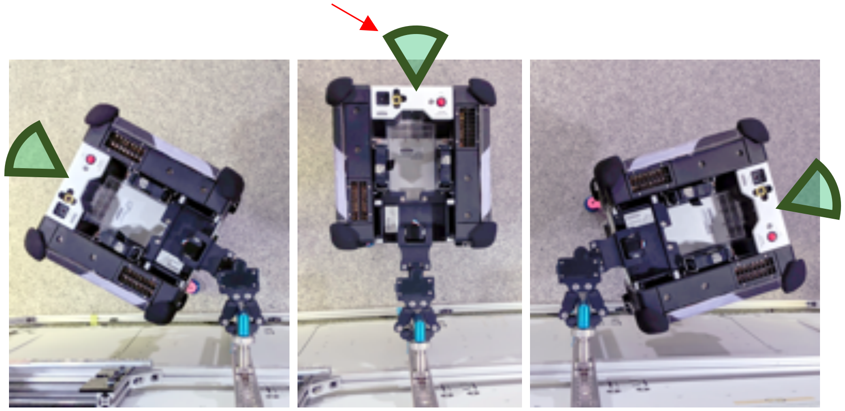


Feature map of the JEM-PM

System Description - Perching Arm

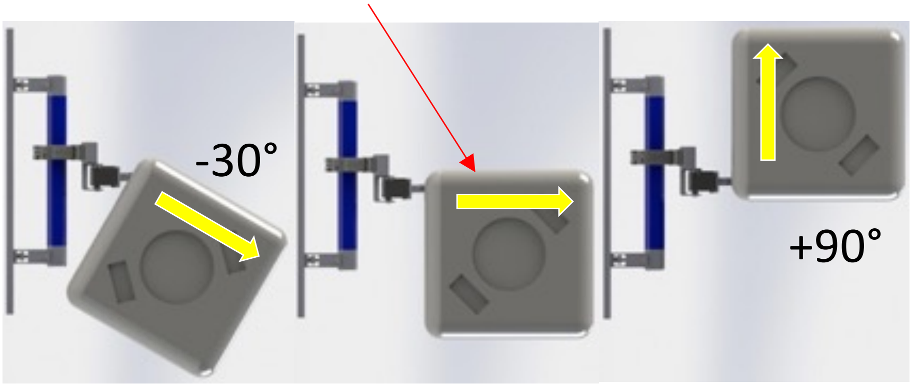
- Designed to grasp handrails
- Stows completely in payload bay
- Acts as a pan-tilt unit while perched
- Flexible and back-drivable
- May be perched manually

Camera View Direction



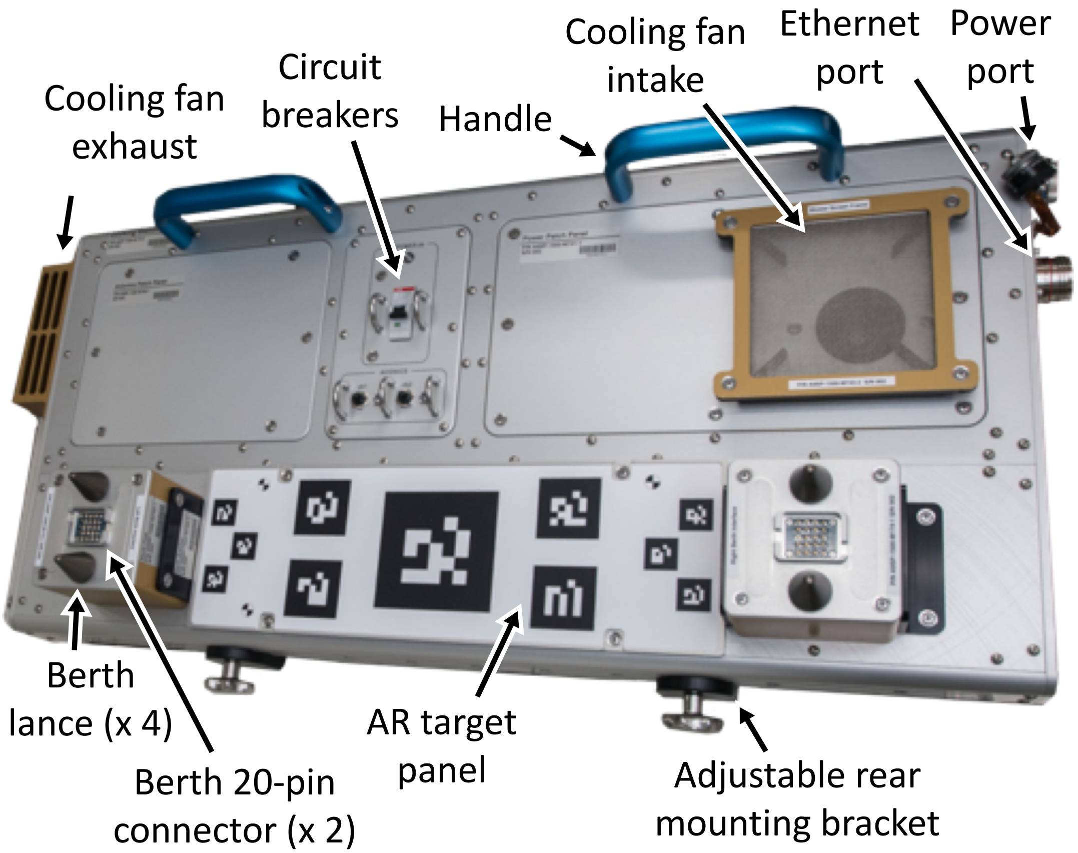
Astrobee Perching Arm pan motion

Camera View Direction



Astrobee Perching Arm tilt motion

System Description - Docking Station

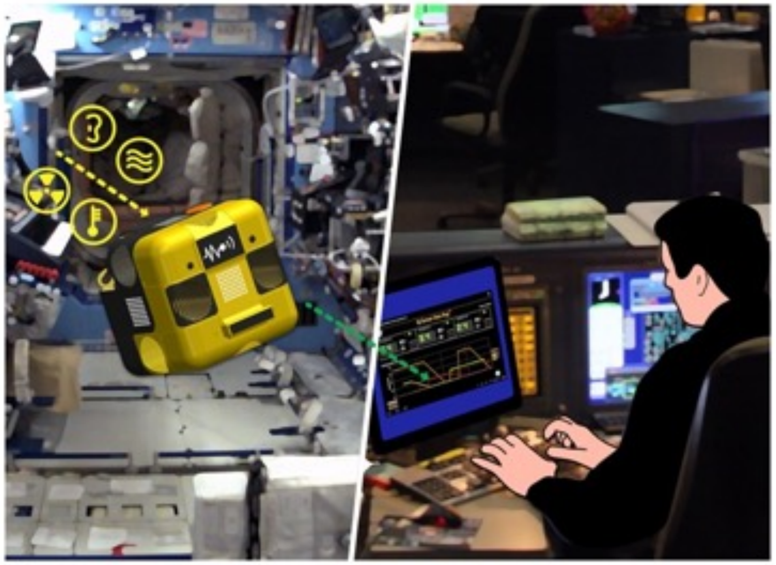


- 85 cm x 38 cm x 28 cm
- Berths for 2 free flyers
- Provides power and Ethernet

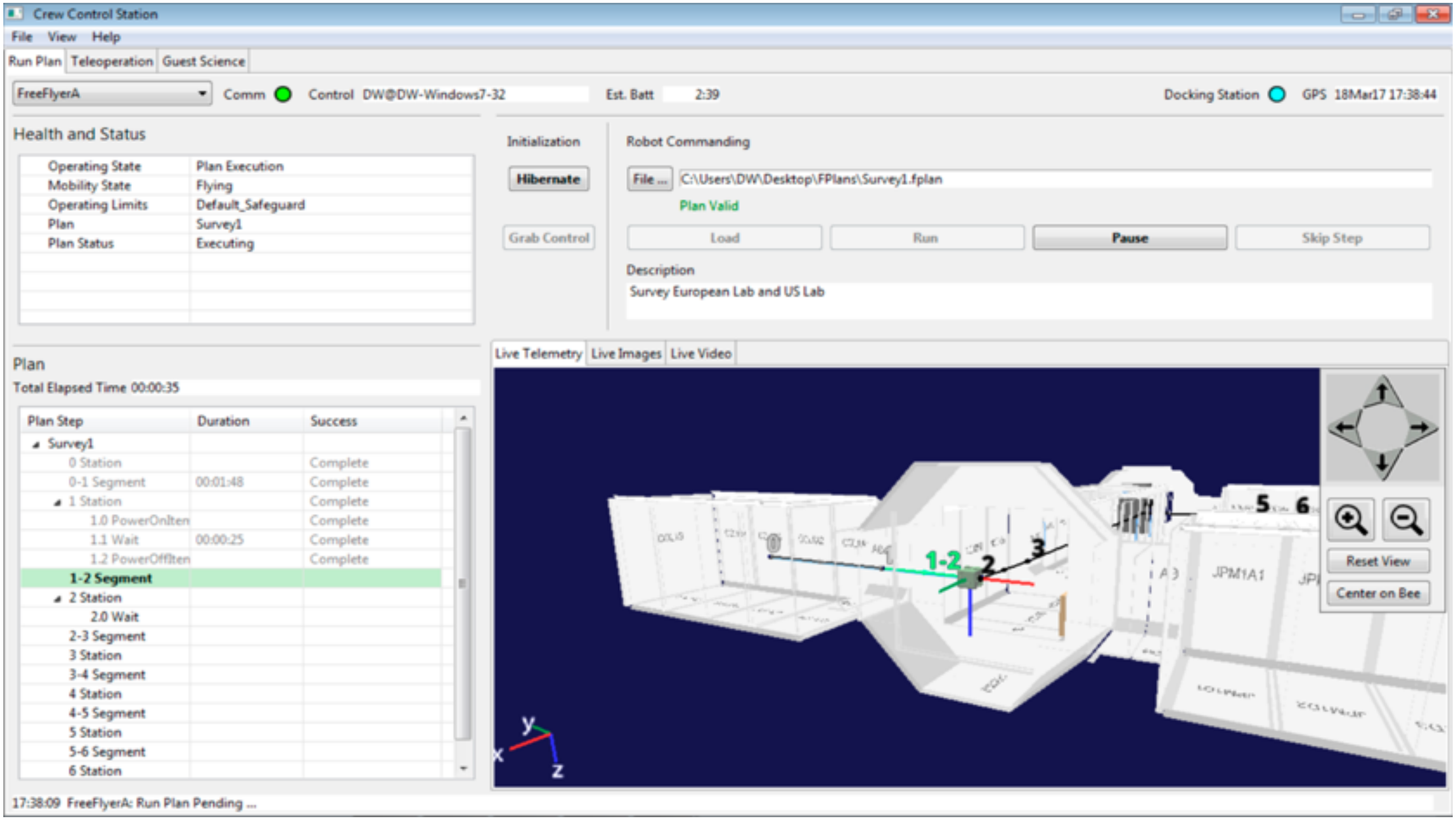
- Fiducials used for visual servoing to autonomously dock
- Magnets provide retention force

System Description - Ground Data System

- Astrobee Control Station
  - Sortie planning tool
  - Execution monitoring
    - Live telemetry
    - Image and video streams
    - 3D virtual display
  - Supervisory control (run plans or single commands)
  - Typically used by ground operators
- Crew Control Station runs on an EXPRESS Laptop Computer (ELC)
- Server for archiving and distributing Astrobee data
- Suite of engineering tools to support maintenance and software upgrades



Astrobee Control Station



FreeFlyerA Comm ● Control DW@DW-Windows7-32 Est. Batt 2:39 Docking Station ● GPS 18Mar17 17:38:44

Health and Status

Operating State	Plan Execution
Mobility State	Flying
Operating Limits	Default_Safeguard
Plan	Survey1
Plan Status	Executing

C:\Users\DW\Desktop\FPlans\Survey1.fplan

Plan Valid

Description

Survey European Lab and US Lab

Plan

Total Elapsed Time 00:00:35

Plan Step	Duration	Success
Survey1		
0 Station		Complete
0-1 Segment	00:01:48	Complete
1 Station		Complete
1.0 PowerOnIten		Complete
1.1 Wait	00:00:25	Complete
1.2 PowerOffIten		Complete
1-2 Segment		
2 Station		
2.0 Wait		
2-3 Segment		
3 Station		
3-4 Segment		
4 Station		
4-5 Segment		
5 Station		
5-6 Segment		
6 Station		

17:38:09 FreeFlyerA: Run Plan Pending ...

Basic Conops

- When an Astrobee is idle, it charges in its dock
- Astrobees can execute complex plans with full autonomy and no astronauts present
  - Including undock, traverse multiple modules, return to dock
- However, Astrobees run with ground operator oversight
  - When an anomaly occurs, an Astrobee generally stops and waits for operator intervention
  - It can continue operating during communication outages until it encounters an anomaly
- The operator can always take over and teleoperate
- Astronauts can also be operators, but this is will likely be a rare occurance (minimize crew time)

Hardware Status

- Docking Station
  - Launched on NG-10 – 11/17/18
  - Installed on 2/15/19
- 2 Astrobees (Bumble & Honey) and Dock Spares Kit
  - Launched on NG-11 – 4/17/19
  - Checkout of Bumble began 4/30/19
  - Checkout of Honey around 7/25/19
- 1 Astrobee (Queen) and 3 Perching Arms
  - Launching on SpX-18 – 7/21/19
  - All hardware delivered and verifications closed
- Astrobee Spares Kit and Port Tester
  - Launching on NG-12 – 10/19/19
  - On-dock 8/6/19

Astrobee Commissioning Activities

1. Checkout
   - Crew inspects the robot
   - Functional tests of all hardware
2. Calibration & Mapping
   - Crew collects camera and IMU calibration data
   - Crew collects mapping data of Kibo
3. Localization & Mobility
   - Verify robot localization within Kibo
   - Robot performs increasingly complex motions to test mobility system
4. Checkout & Calibration
   - Combination of 1 & 2 without mapping

Astrobee Commissioning Activities

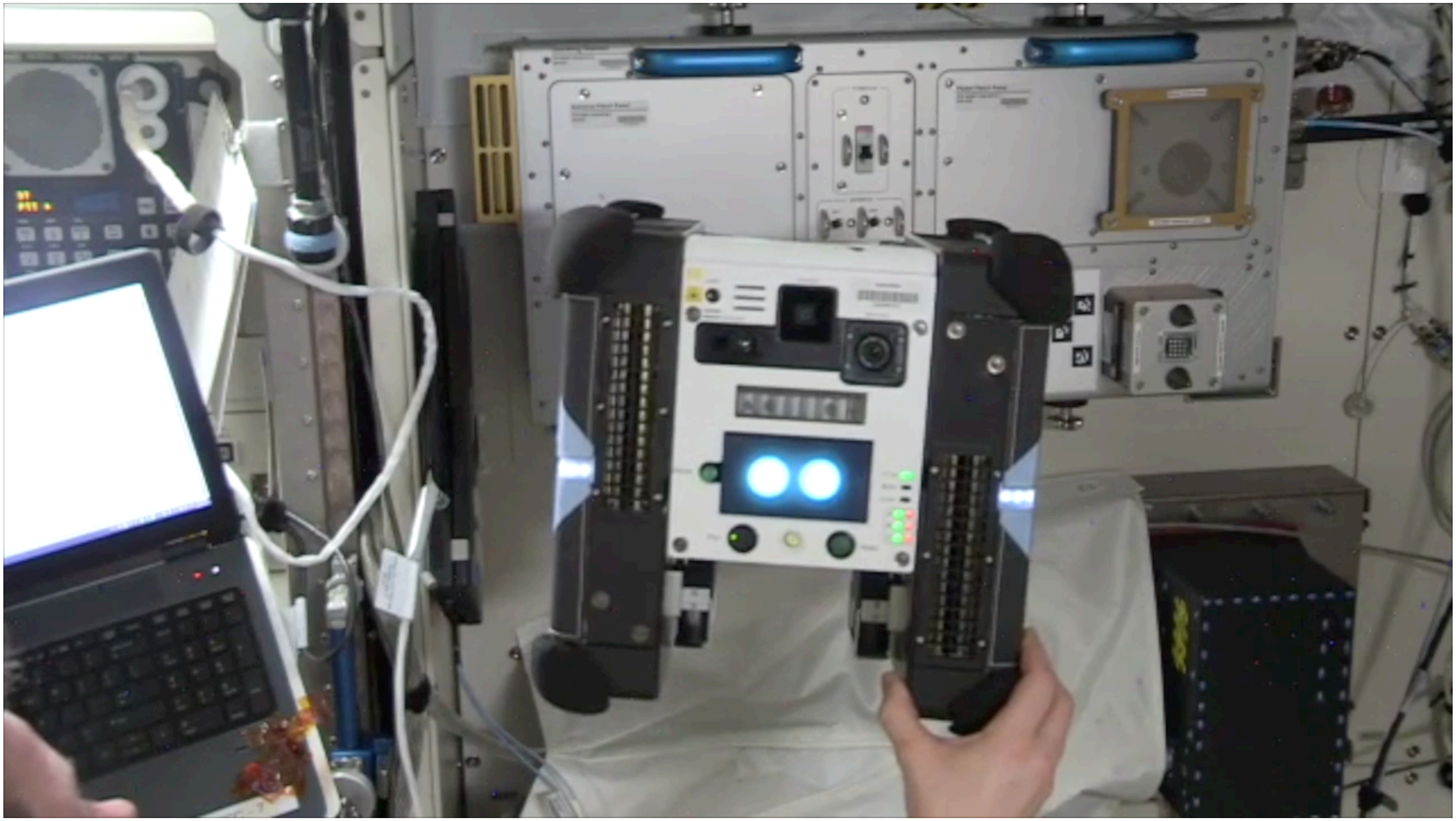
6. Ops Demo
   - Demonstrate an operational mission scenario
7. Payload Installation
   - Crew installs payload (perching arm)
   - Functional tests of payload
8. Payload Demo
   - Operational demonstration of payload (perching arm)
9. Performance Characterization
   - TBD tests to further characterize Astrobee performance
10. SPHERES/Astrobee Hand-off
    - Symbolic passing of the torch from SPHERES to Astrobee
    - Completion of commissioning

Commissioning Schedule

Completed | Scheduled | Desired

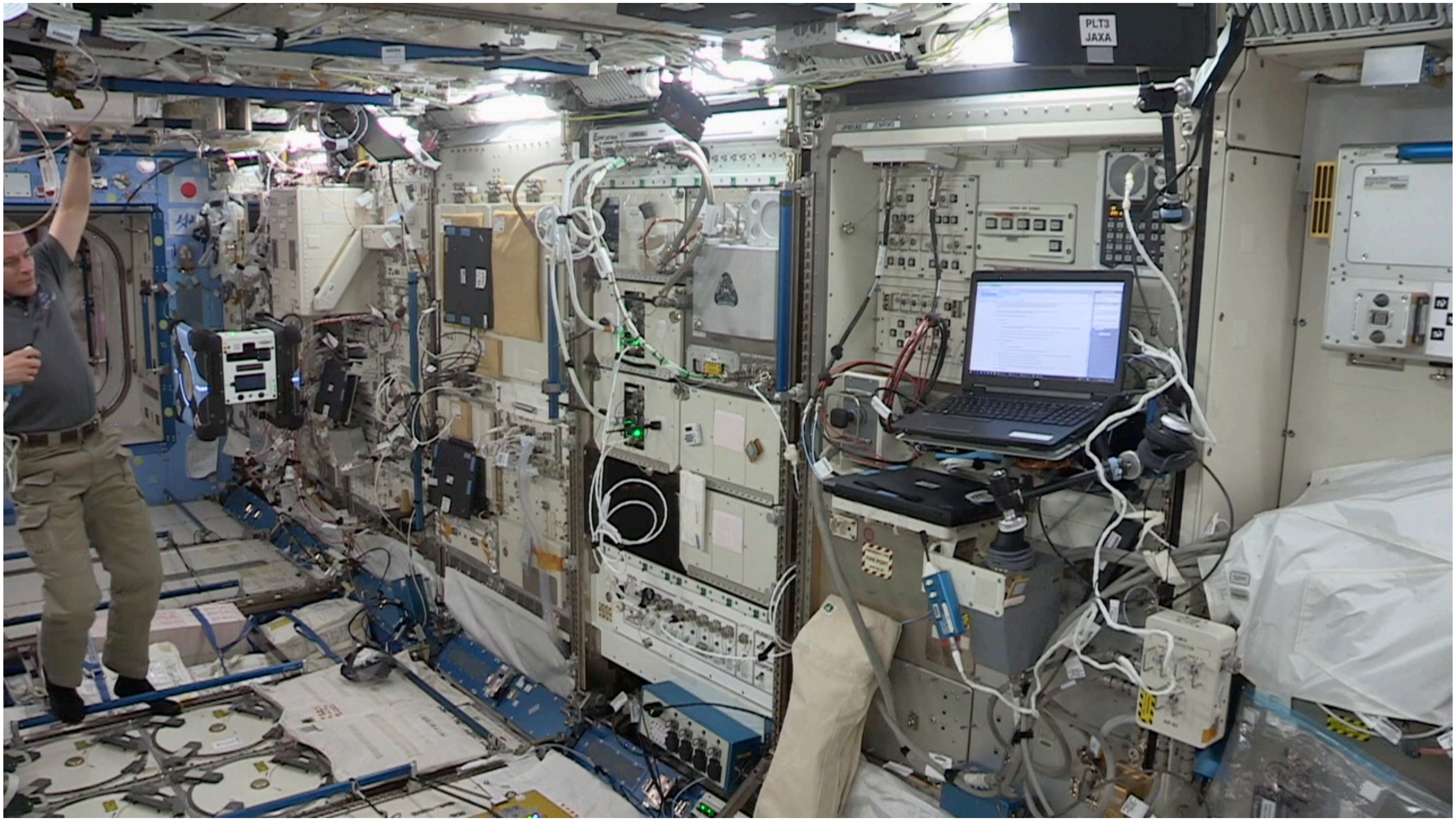
Activity	Robot	Date
Checkout	Bumble	4/30/19
Calibration & Mapping	Bumble	5/13/19
Additional Mapping	Bumble	5/23/19
Localization & Mobility	Bumble	6/14/19
Localization & Mobility	Bumble	7/12/19
Checkout & Calibration	Honey	7/25/19
Localization & Mobility	Bumble	7/30/19
Localization & Mobility	Honey	8/13/19
Checkout & Calibration	Queen	8/20/19
Localization & Mobility	Honey	8/27/19

Bumble Checkout



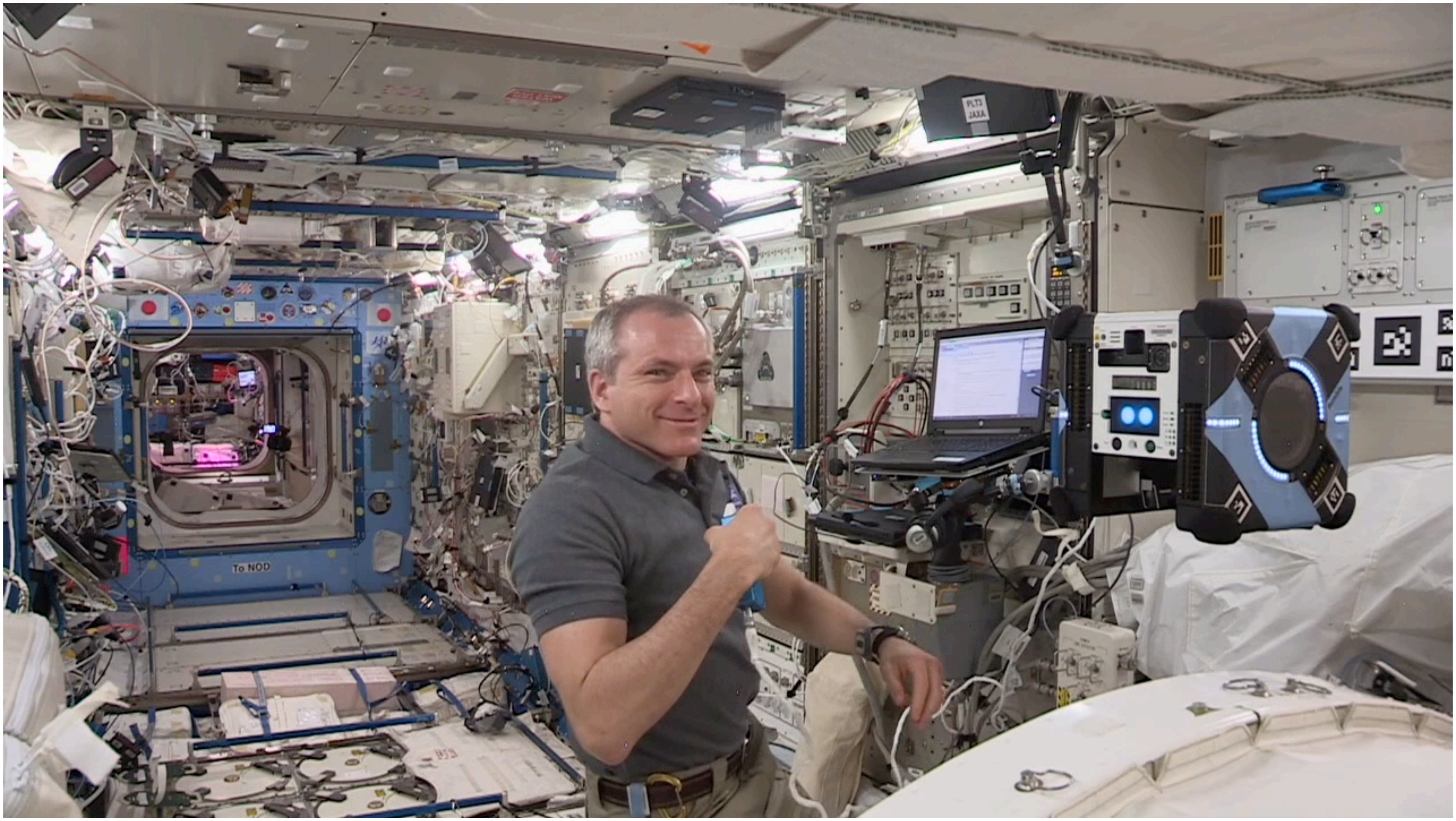
IMU Calibration

1st Flight

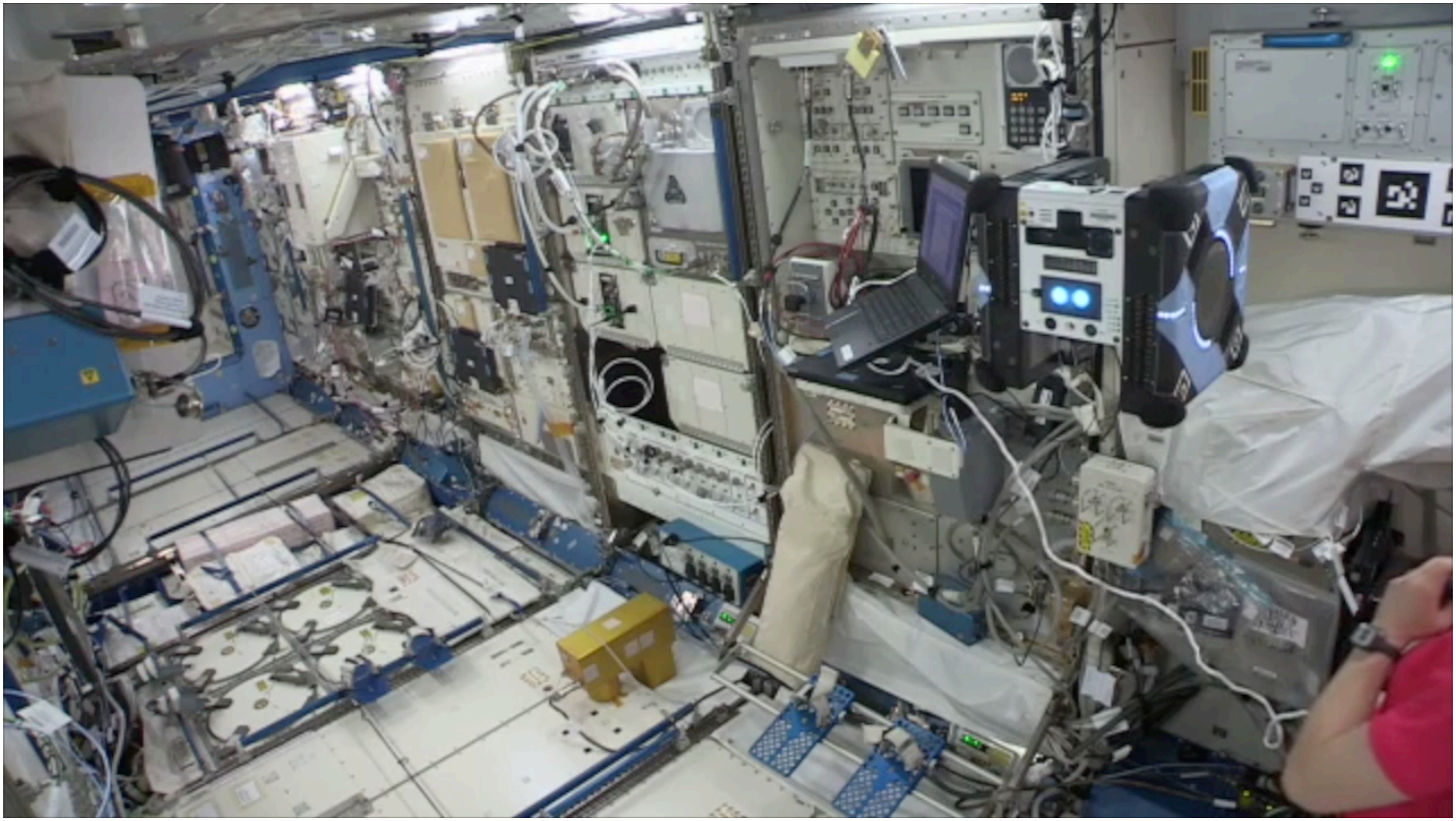


Stopping Ability

1st Autonomous Undocking



1st Autonomous Docking



Near-Term Guest Science Payloads

- NASA JSC’s RFID-Recon (REALM-2)
  - RFID reader allows inventory and searches for logistics reduction
  - Launch on NG-12
- CASIS’s SoundSee (Astrobotic/Bosch)
  - Microphone array that creates a sound map used to monitor health of on-board systems
  - Launch on NG-12
- Gecko Gripper (Stanford)
  - Gecko-inspired end-effector for gripping of smooth surfaces
  - Launch on SpX-18
- Zero Robotics (MIT)
  - Middle school and high school STEM robotics competition
- JAXA Astrobee-IntBall Joint activity, Kibo-RPC
  - Asia-region STEM robotics competition
- Astrobatics (Naval Post-graduate School)
  - Software-only payload investigating propellant-less propulsion using a manipulator

Fun with Robots

Questions?

Astrobee System Overview

Ames Research Center, NASA

Moffett Field, CA, USA

November 25th, 2019

Contents

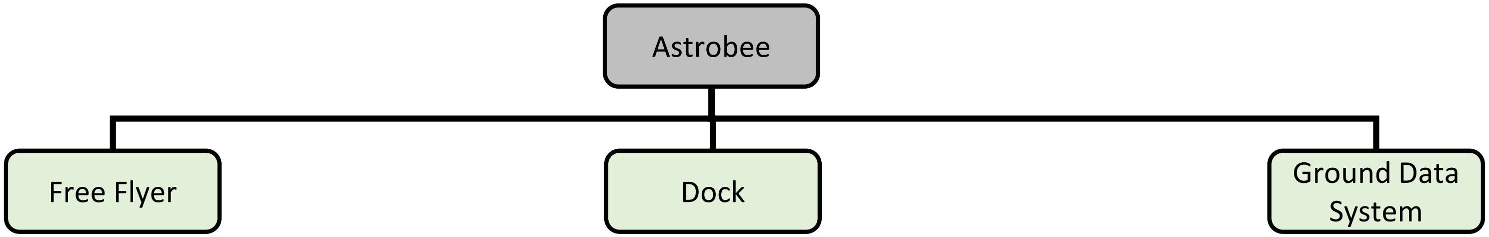
- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

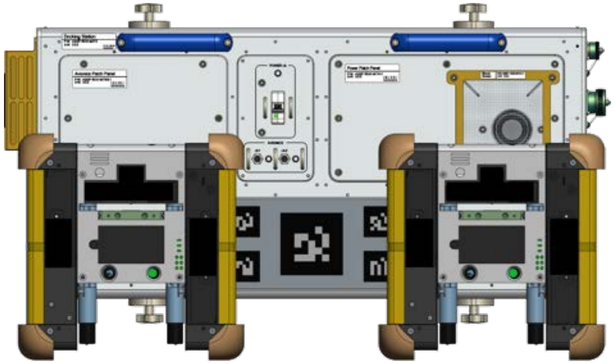
Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

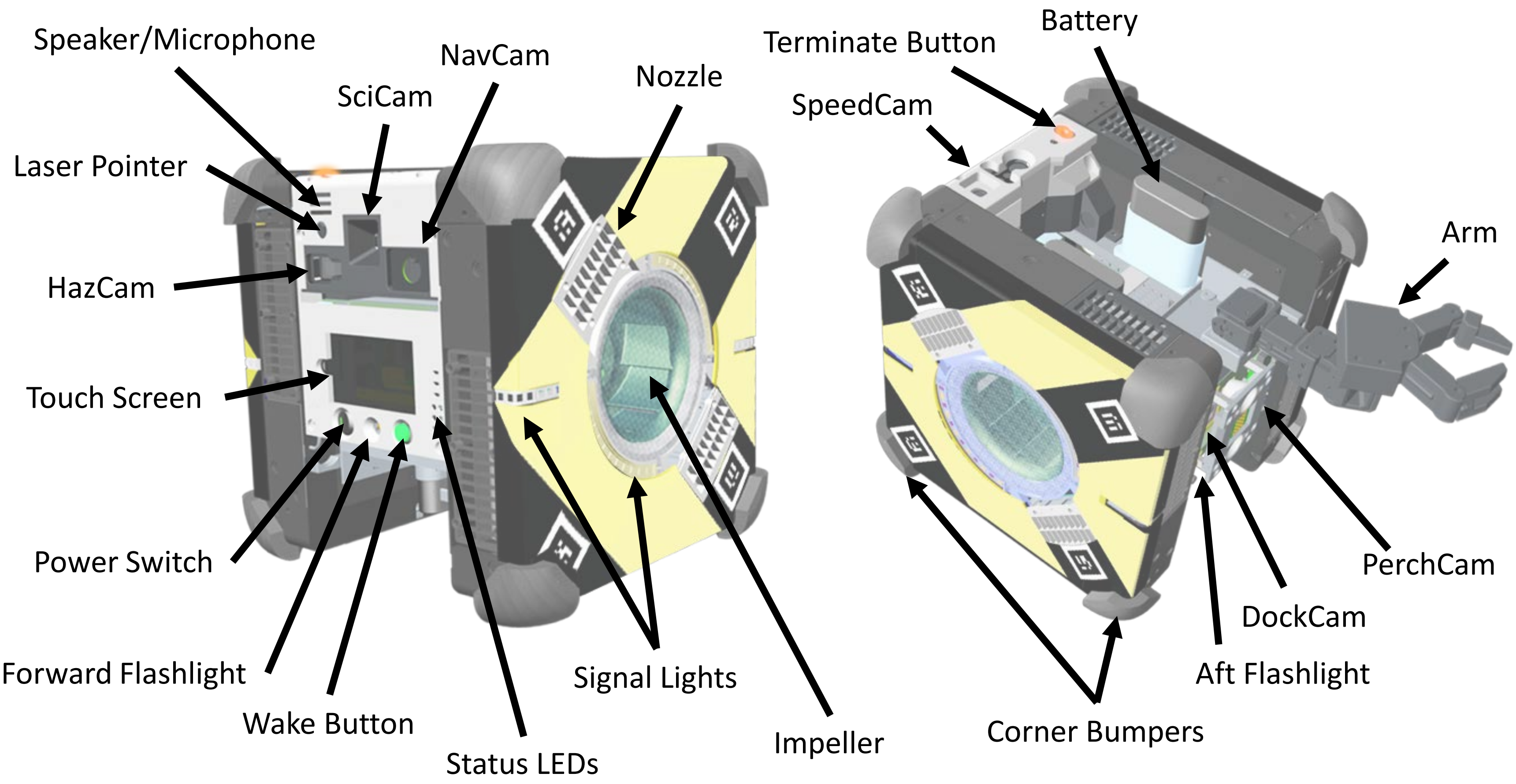
Astrobee System

- Main components





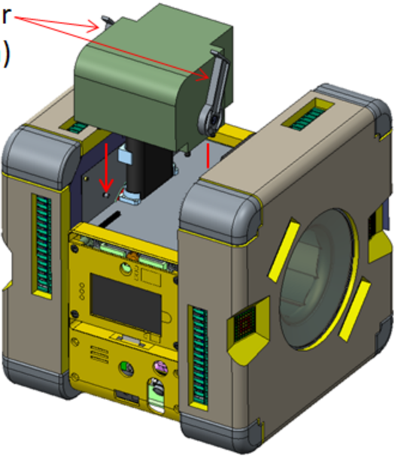
Astrobee System: Free-flyer



Astrobee System: Free-flyer's Hardware Payloads

Quick "No Tool" Payload Attachment

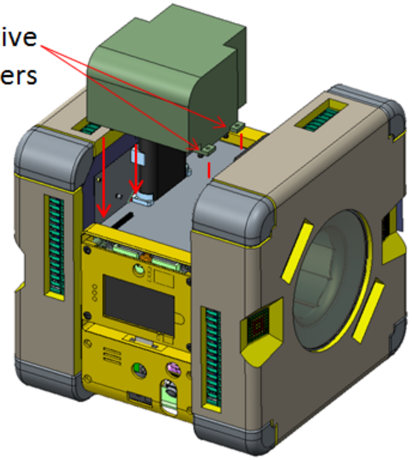
2X Lever (open position)



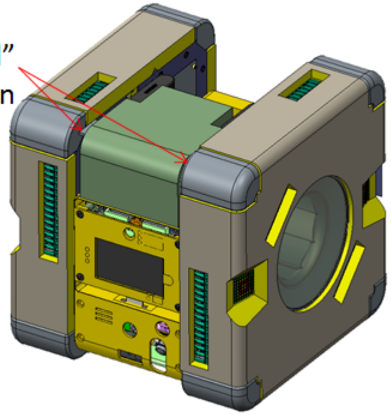
Lever engages and disengages payload connector and provides mechanical attachment

4X Fastener Payload Attachment

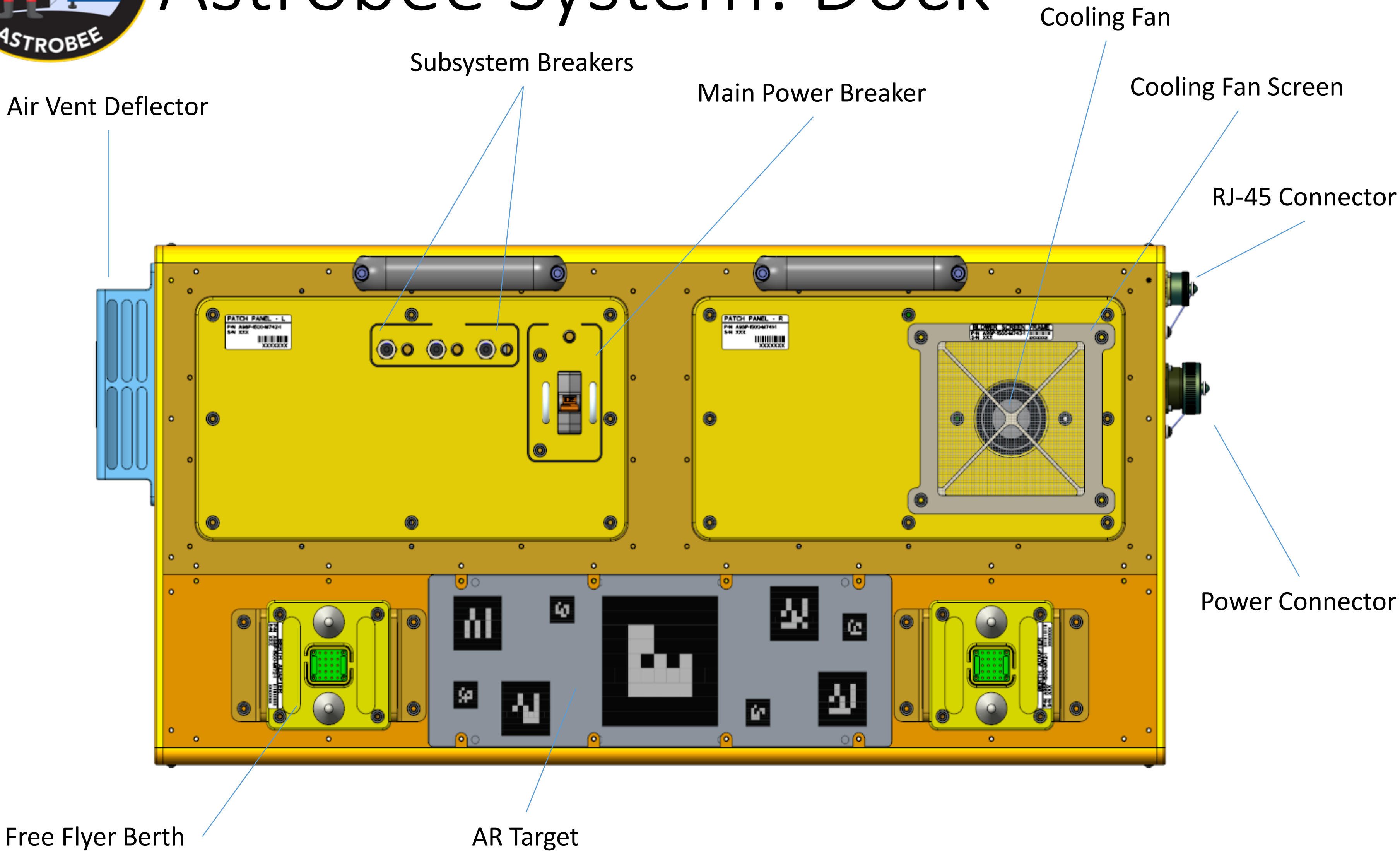
4X Captive Fasteners



Lever in "Locked" position



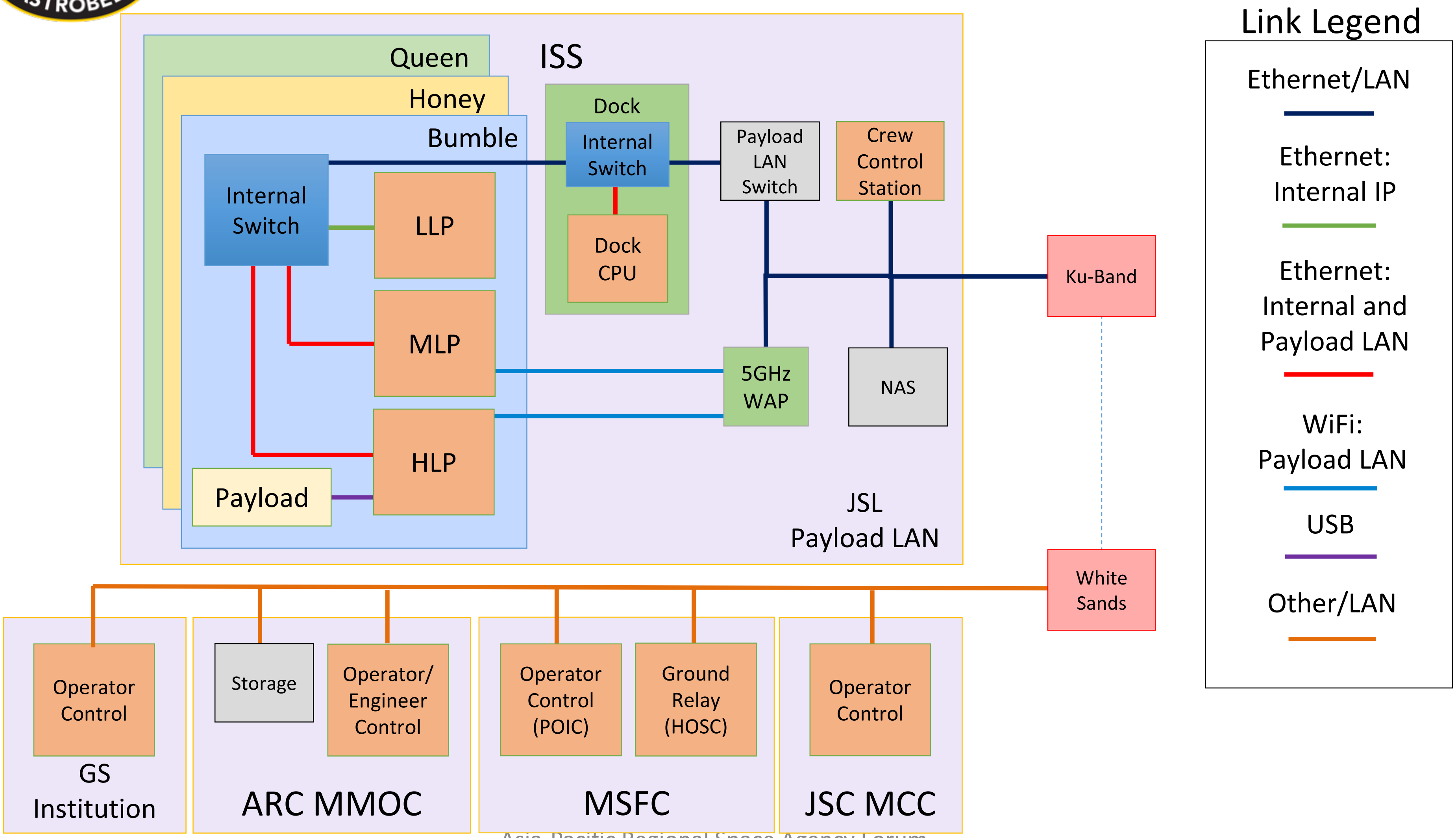
Astrobee System: Dock



Astrobee System: Ground Data System

- Provides control over the Astrobee free-flyer
  - by the astronaut
  - by ground controllers
  - by Guest Scientists (GS)
- Enables data exchange between Ast ground
- GDS interfaces Middle Level Process) Executive
- GDS communicates through Executive with High Level Processor (HLP) where GS Android Apps are deployed

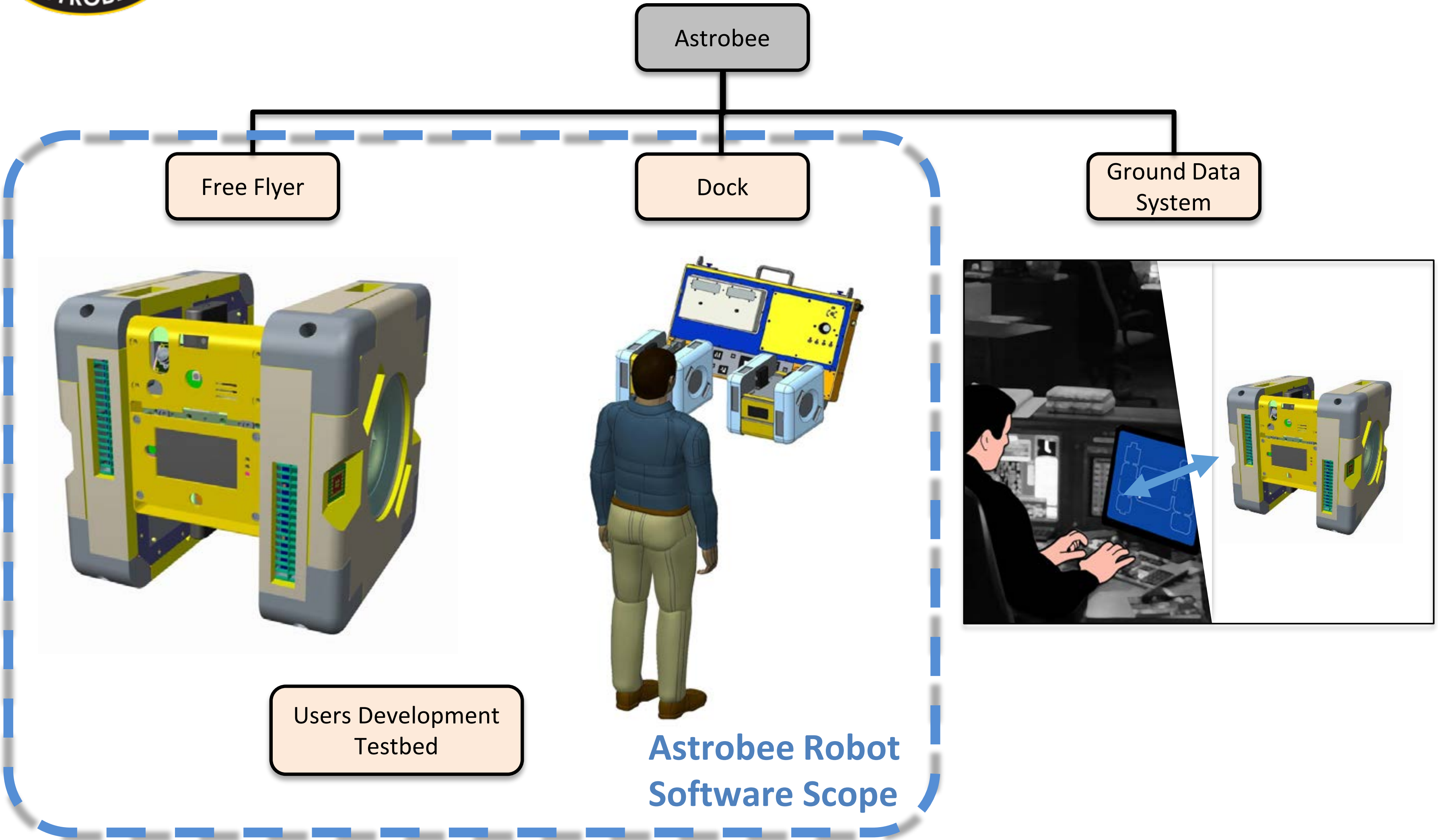
Astrobee System: Data Flow



Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

Astrobee Robot Software



Astrobee

Free Flyer

Dock

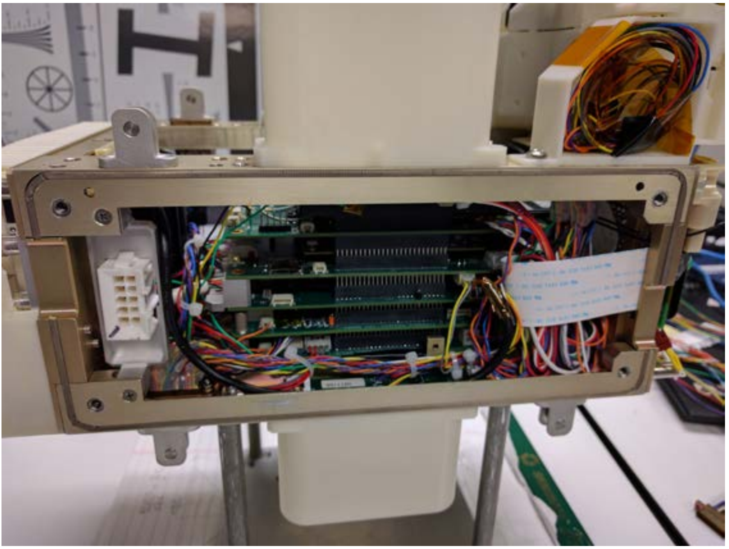
Ground Data System

Users Development Testbed

Astrobee Robot Software Scope

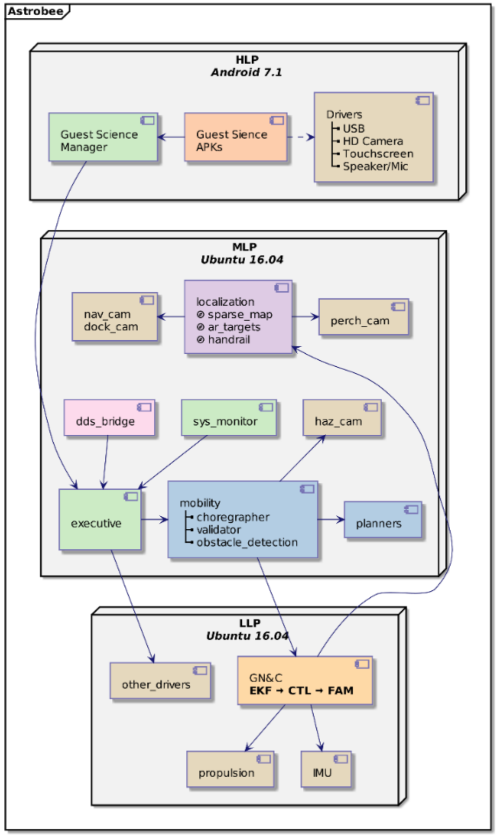
Astrobee Robot Software: Overview

- Astrobee Robot Software (ARS) is deployed on 4 cell phone type processors (Astrobee + Dock) running Linux and Android
- Astrobee contains 7 distinct microprocessors with custom firmware + several microprocessors with Components Off-The-Shelf (COTS) firmware
- Software deliverables includes:
  - Custom firmware(s)
  - Custom tailored Linux kernels
  - Linux and Android Operating Systems
  - ARS dependencies (third-party libraries)
  - **ARS Source Code** ← **Open Source**



Astrobee Robot Software: Overview

Astrobee Robot Software Overview



Astrobee

HLP
Android 7.1

Guest Science Manager

Guest Sience APKs

Drivers
- USB
- HD Camera
- Touchscreen
- Speaker/Mic

MLP
Ubuntu 16.04

nav_cam
dock_cam

localization
- sparse_map
- ar_targets
- handrail

perch_cam

dds_bridge

sys_monitor

haz_cam

executive

mobility
- choregrapher
- validator
- obstacle_detection

planners

LLP
Ubuntu 16.04

other_drivers

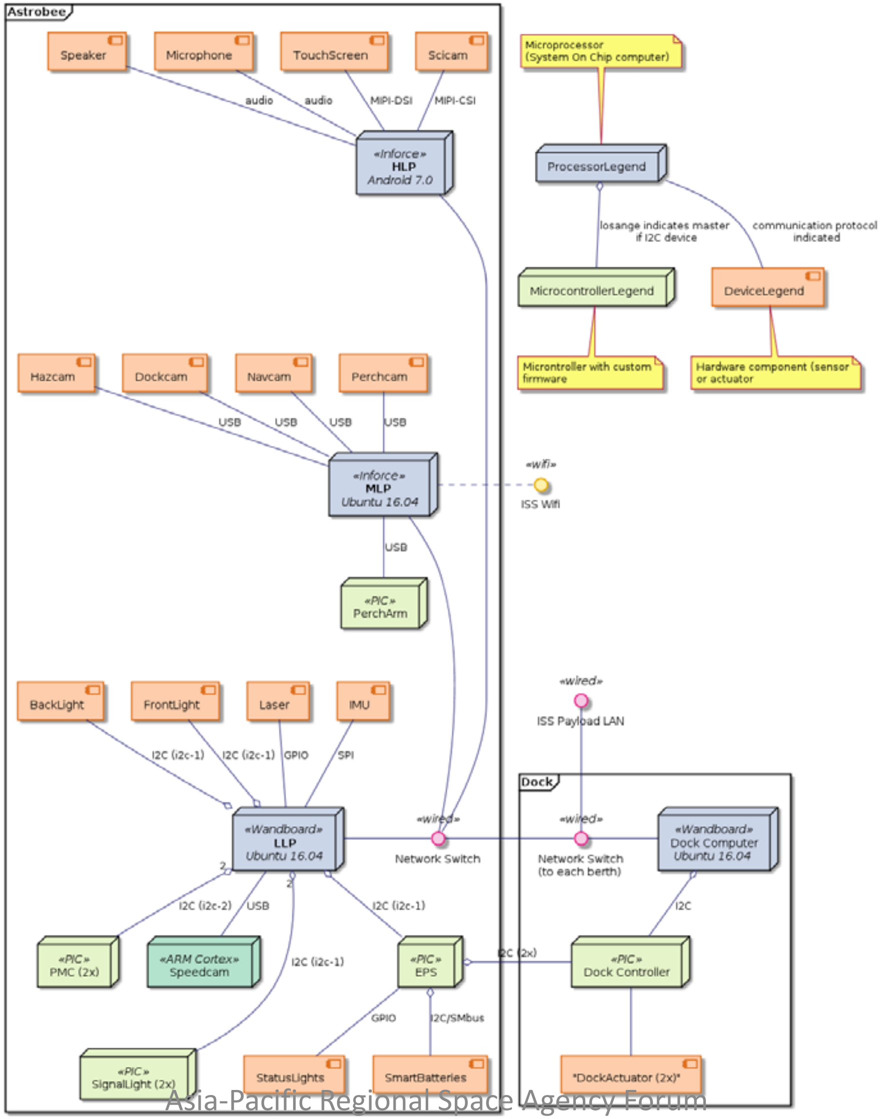
GN&C
EKF → CTL → FAM

propulsion

IMU

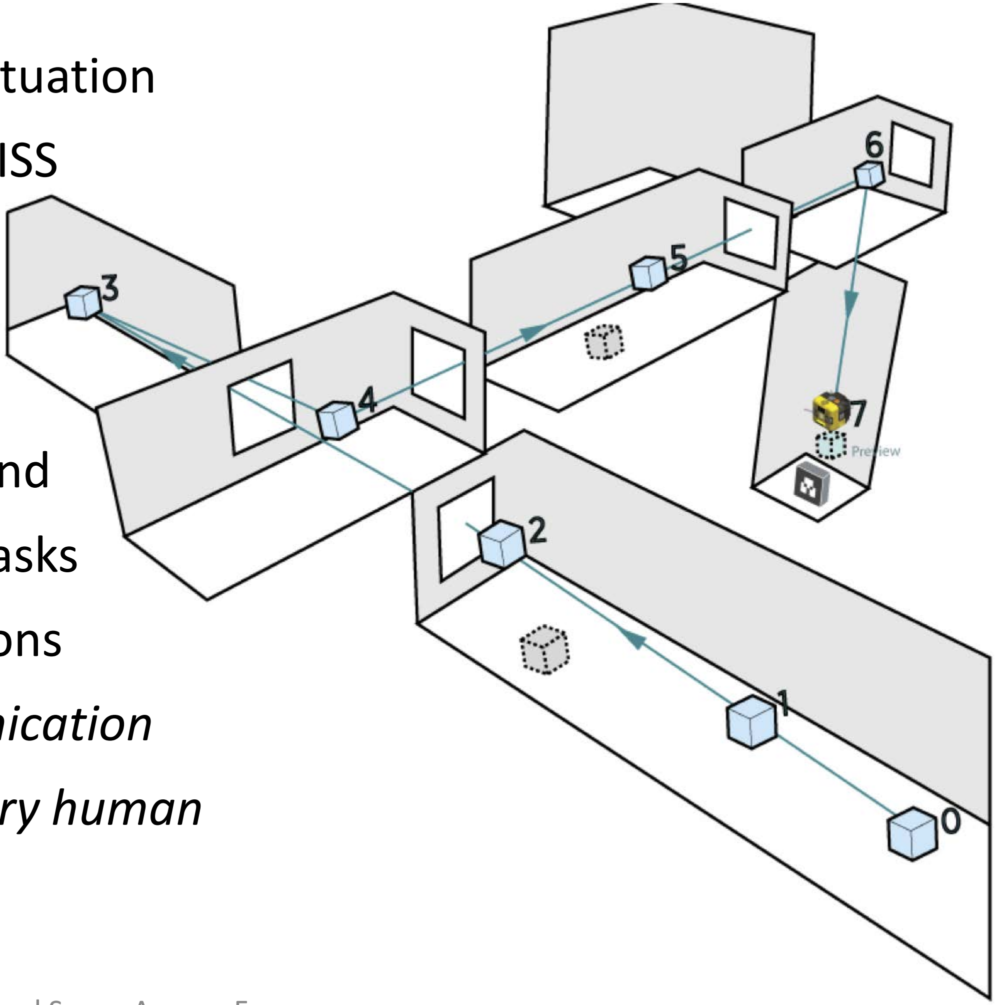
Arrows are representing dependencies, not flow of information

Astrobee Robot Software: Overview



Astrobee Robot Software: Features

- Manage Astrobee sensing and actuation
- Localize and Navigate within the ISS
- Perform autonomous docking
  (+ return to dock)
- Perform autonomous perching
- Support teleoperation from ground
- Support plan based automated tasks
- Support "Guest Science" operations
- *Support multi Astrobees communication*
- *Support hardware for multisensory human interaction*



Astrobee Robot Software: Components

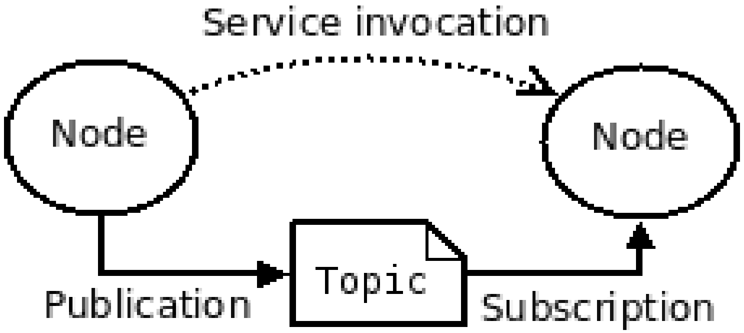
- Communication Framework (ROS + DDS based)
- Localization
    - Marker less Flying
    - Docking
    - Perching
- Offline mapping for localization
- Pose Estimation + Propulsion Control (GNC)
- Executive
    - Mode Management
    - Sequencer (Plan Execution)
- Mobility
    - Coordinates required subsystem
    - Generates and validates trajectories
    - Performs collision detection
- Fault Management
- Guest Science
- User Interfaces Support
- Simulator
- Platform Management and development tools

Number or ROS nodes for a simulation:	~36
Number of ROS nodes on Astrobee: (not counting HLP)	~48

```
---------------------------------------------------
Language           files   comment    code
---------------------------------------------------
C++                  263     11078    45023
Lua and config        76      2179    14380
C/C++ Header         158      4232     8209
Python                58      1798     6706
CMake                150      3631     5624
IDL MSG
SRV ACTION           148         0     5139
XML                  163       309     4271
Java                  70      1764     3822
Bourne (Again) Shell  83       853     2522
YAML                   4        29      294
make                   9        75      238
---------------------------------------------------
SUM:                1182     25948    96228
---------------------------------------------------
```

Astrobee Robot Software: ROS

- ARS makes extensive use of the open-source Robot Operating System (ROS):
  - Communication framework linking all "nodes" running on the target platform
  - Try to maximize the re-use of existing ROS messages benefit from existing ROS packages
  - Use ROS introspections tools to rapid debugging
  - Use ROS facilities to record/replay/analyze data
  - Use some ROS/Gazebo components for the simulator



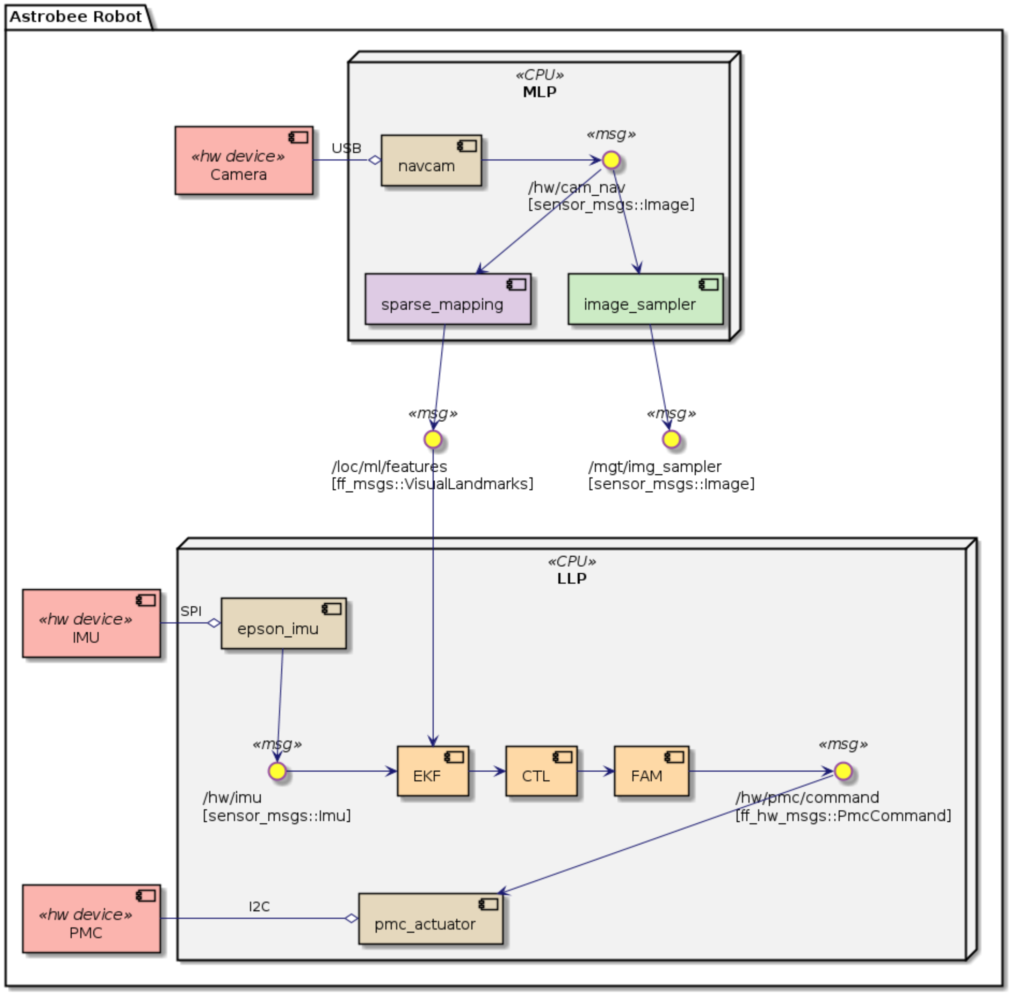
Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

Simulator

- ARS can run on:
  - actual hardware on Astrobee (camera, IMU, PMC)
  - simulated robot hardware on a PC
  - both simultaneously
- Simulation is possible by enabling/disabling ARS elements
- Simulator may be thought as a “software-in-the-loop” simulation of the free-flyer’s hardware drivers.
- Hardware drivers are simulated via Gazebo Plug-ins

Astrobee Robot Software deployed on Physical Robot

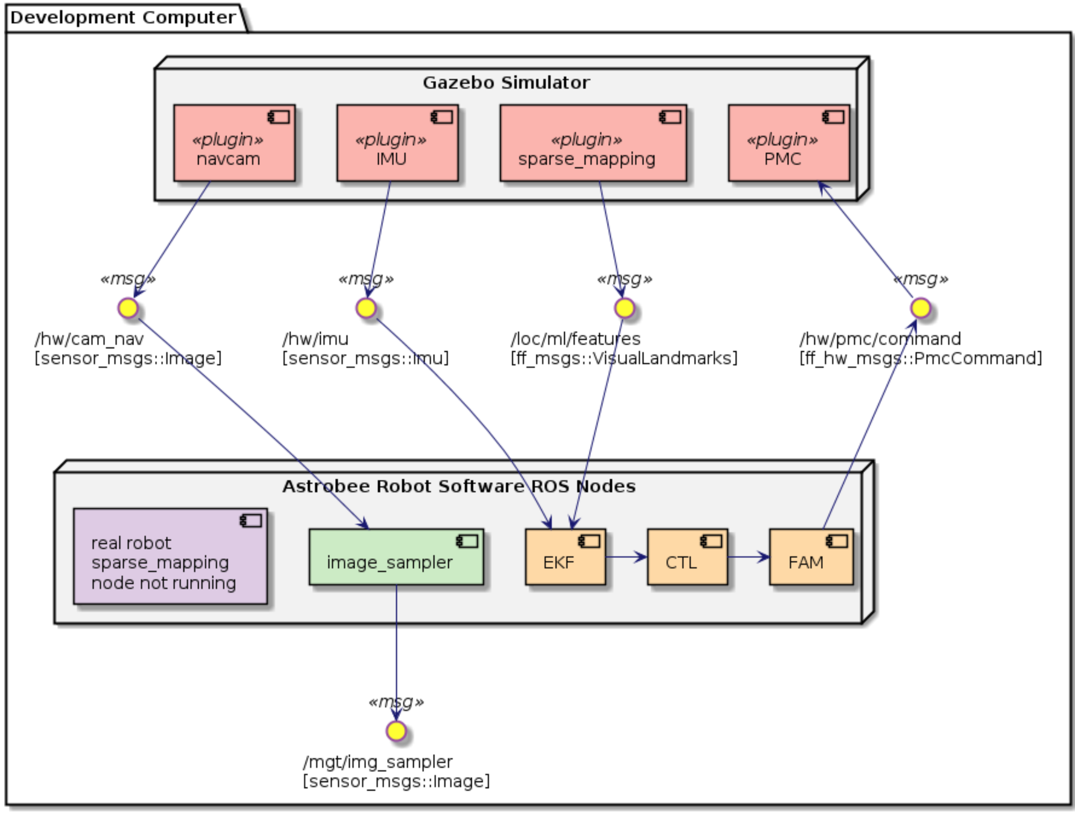


PMC: Propulsion Module Controller
CTL: GN&C Control
FAM: GN&C Force Allocation Module

Simulator

- ARS can run on:
  - actual hardware on Astrobee (camera, IMU, PMC)
  - simulated robot hardware on a PC
  - both simultaneously
- Simulation is possible by enabling/disabling ARS elements
- Simulator may be thought as a “software-in-the-loop” simulation of the free-flyer’s hardware drivers.
- Hardware drivers are simulated via Gazebo Plug-ins

Astrobee Robot Software testing with Simulator

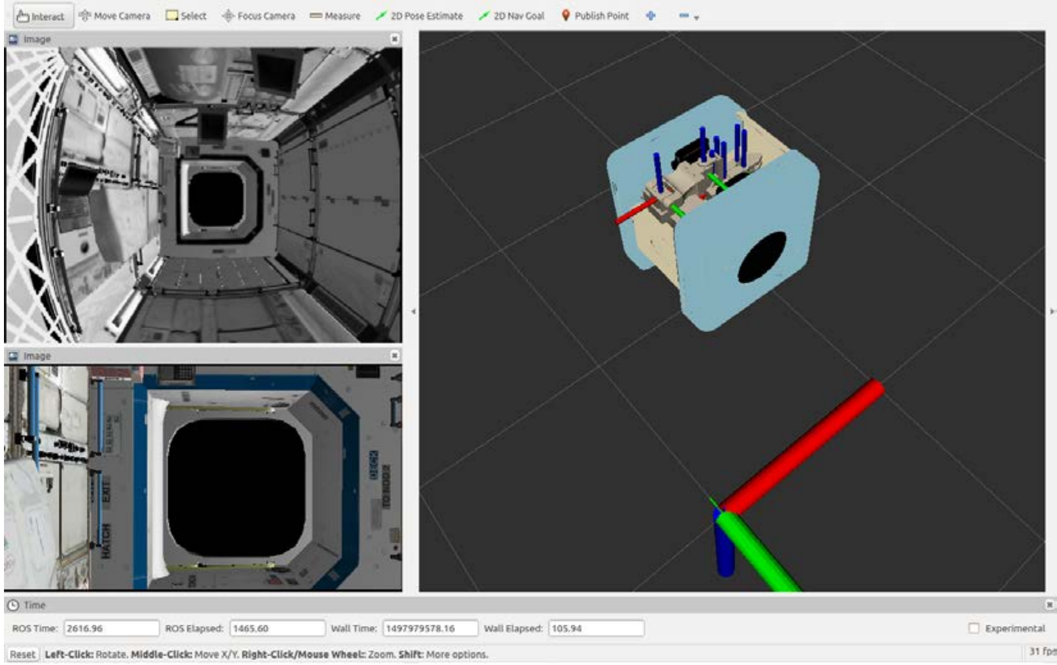


Development Computer
Gazebo Simulator
«plugin» navcam
«plugin» IMU
«plugin» sparse_mapping
«plugin» PMC
«msg»
/hw/cam_nav [sensor_msgs::Image]
«msg»
/hw/imu [sensor_msgs::Imu]
«msg»
/loc/ml/features [ff_msgs::VisualLandmarks]
«msg»
/hw/pmc/command [ff_hw_msgs::PmcCommand]
Astrobee Robot Software ROS Nodes
real robot sparse_mapping node not running
image_sampler
EKF
CTL
FAM
«msg»
/mgt/img_sampler [sensor_msgs::Image]

PMC: Propulsion Module Controller
CTL: GN&C Control
FAM: GN&C Force Allocation Module

Simulator

- Contains custom propulsion system and some localization sensors
- Gazebo-based dynamics model, imagers, arm, lights and ISS model
- Can run all nodes on desktop or some nodes on target development board



ROS Time: 2616.96 ROS Elapsed: 1465.60 Wall Time: 1497979578.16 Wall Elapsed: 105.94 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

Gazebo Rviz (ROS visualization) used to debug ARS

Contents

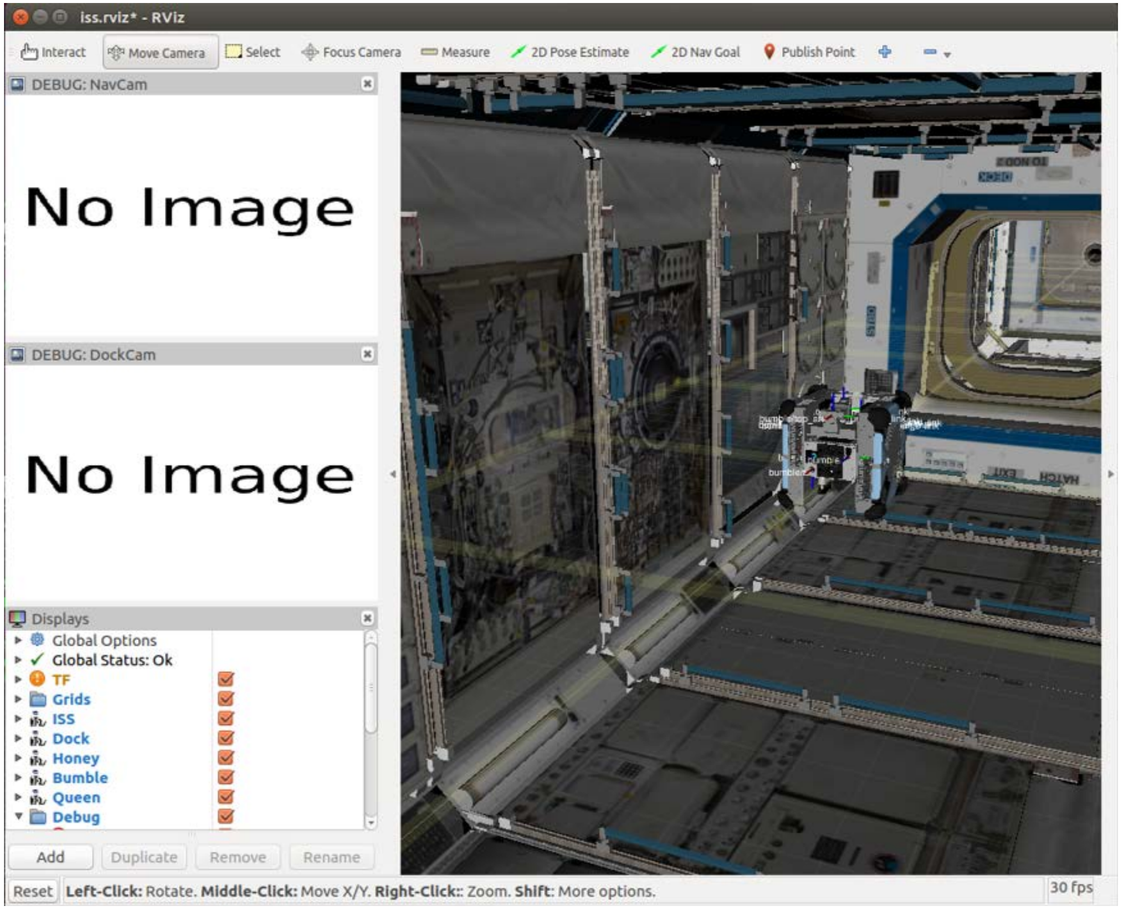
- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

Astrobee Robot Software: Visualizers

- ARS supports 3 main visualizers: Rviz, Gviz, and Sviz
- Simulation can run headless (without any visualizer) or with any combination of them
- Rviz serves as an internal debugging tool from the robot’s perspective
- Gviz describes several data streams (e.g. EKF, PMC, etc.) to determine attitude/localization
- Sviz shows a 3D view of the ISS, the Astrobee free-flyer, its arm, and obstacles

Astrobee Robot Software: Visualizers

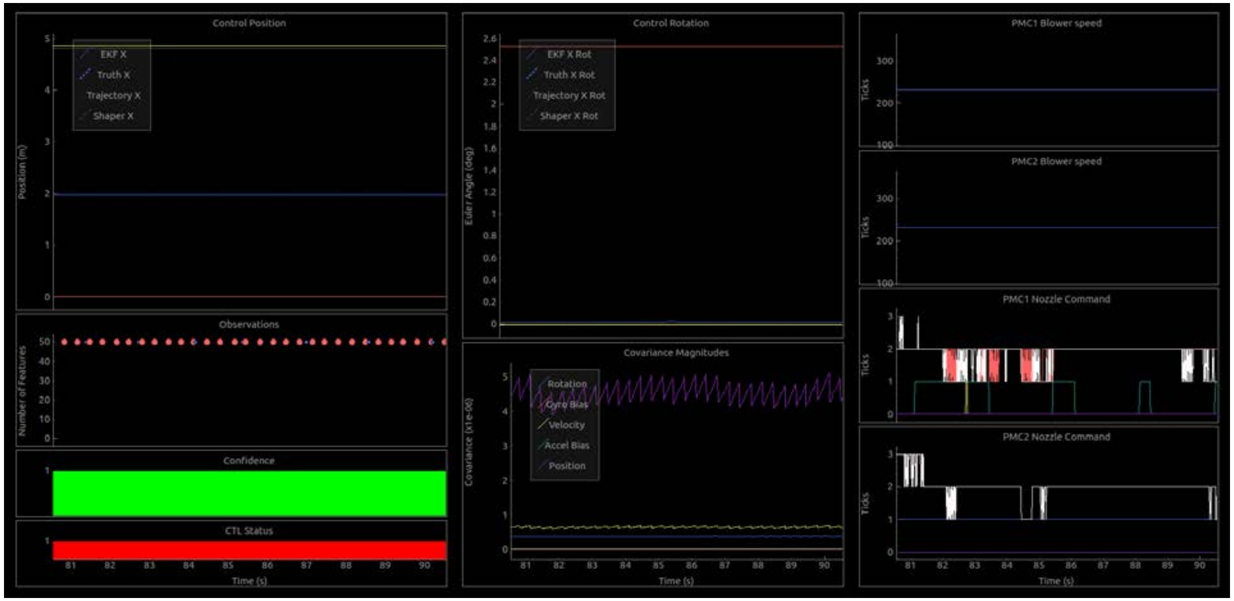
- Rviz



Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click::** Zoom. **Shift:** More options.

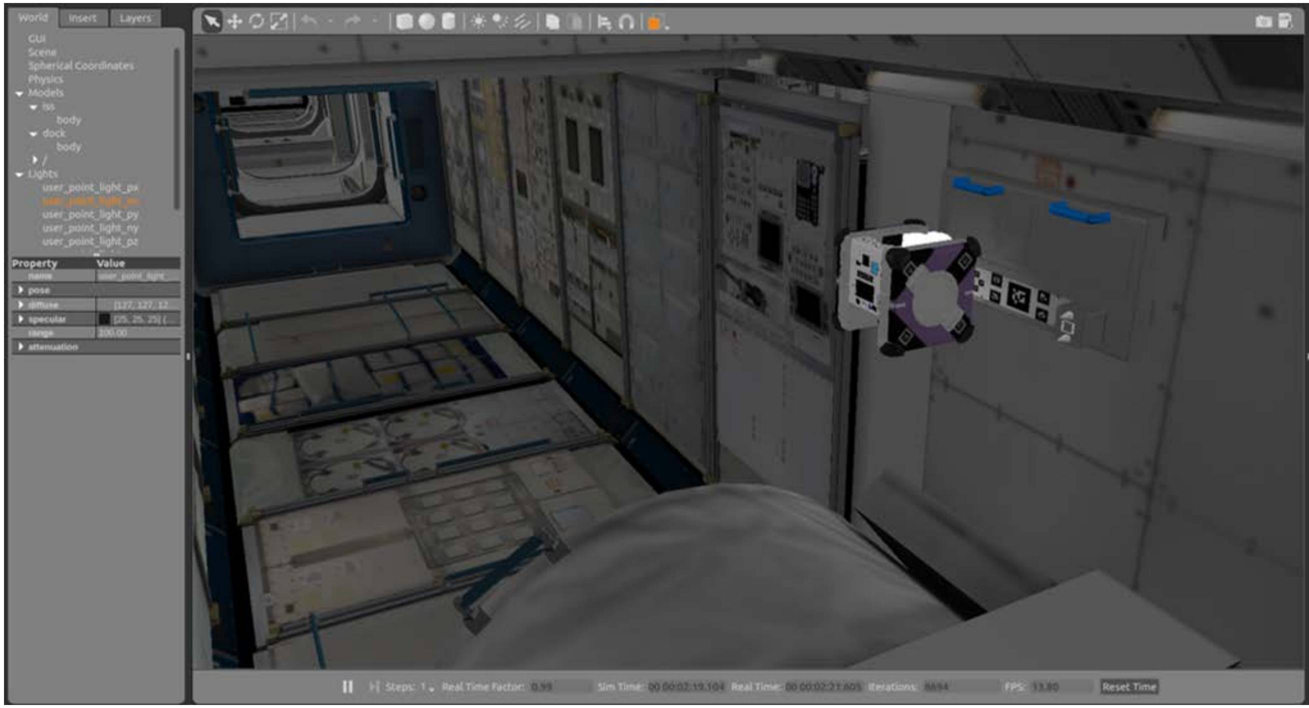
Astrobee Robot Software: Visualizers

- Gviz



Astrobee Robot Software: Visualizers

- Sviz



Property	Value
name	user_point_light_
pose	
diffuse	[127, 127, 12
specular	[25, 25, 25] (
range	100.00
attenuation	

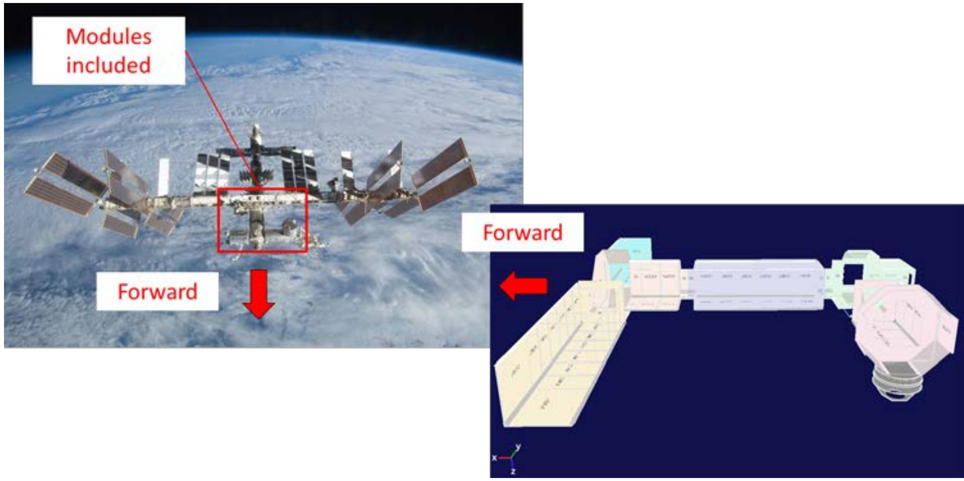
Steps: 1 Real Time Factor: 0.99 Sim Time: 00 00:02:19.104 Real Time: 00 00:02:21.605 Iterations: 8694 FPS: 13.80

Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

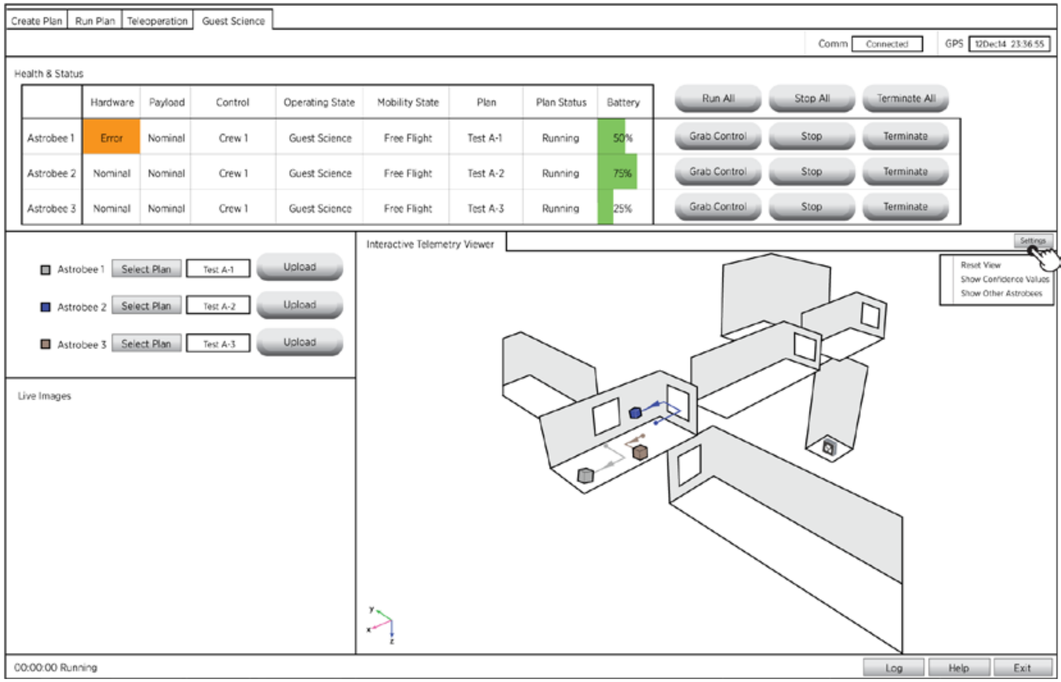
Ground Data System (GDS)

- Ground communication uses DDS middleware to command Astrobee and receive telemetry
- Data Distribution Service (DDS): standard enabling secure real-time information exchange, modular application development and rapid integration (www.rti.com/products/dds-standard)



Ground Data System (GDS)

- Commands:
  - Commands are defined using XP-JSON schema
  - ARS defined a “ROS Command” mirroring the DDS command structure
  - Onboard Astrobee Guest Science or Ground Applications **share** the **same command dictionary** with either DDS or ROS transport
- Telemetry:
  - Internal uses ROS Messages (using ROS messages when possible)
  - External uses DDS Messages (subset only)



Comm: Connected

GPS: 12Dec14 23:36:55

Health & Status

	Hardware	Payload	Control	Operating State	Mobility State	Plan	Plan Status	Battery
Astrobee 1	Error	Nominal	Crew 1	Guest Science	Free Flight	Test A-1	Running	50%
Astrobee 2	Nominal	Nominal	Crew 1	Guest Science	Free Flight	Test A-2	Running	75%
Astrobee 3	Nominal	Nominal	Crew 1	Guest Science	Free Flight	Test A-3	Running	25%

Astrobee 1	Select Plan	Test A-1
Astrobee 2	Select Plan	Test A-2
Astrobee 3	Select Plan	Test A-3

00:00:00 Running

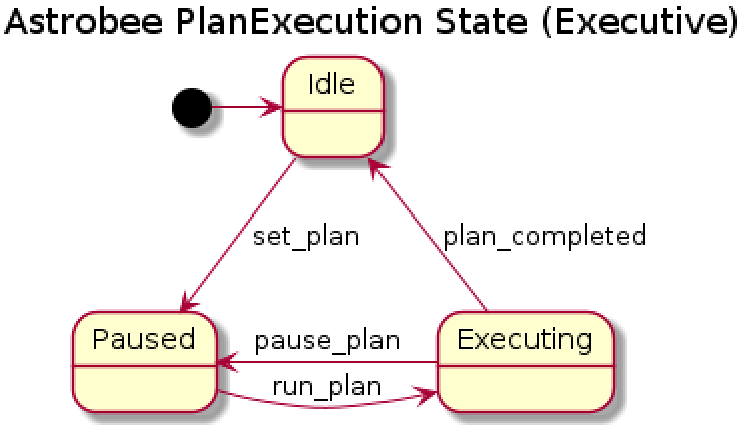
Simplified image for illustrative purposes

Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

Executive

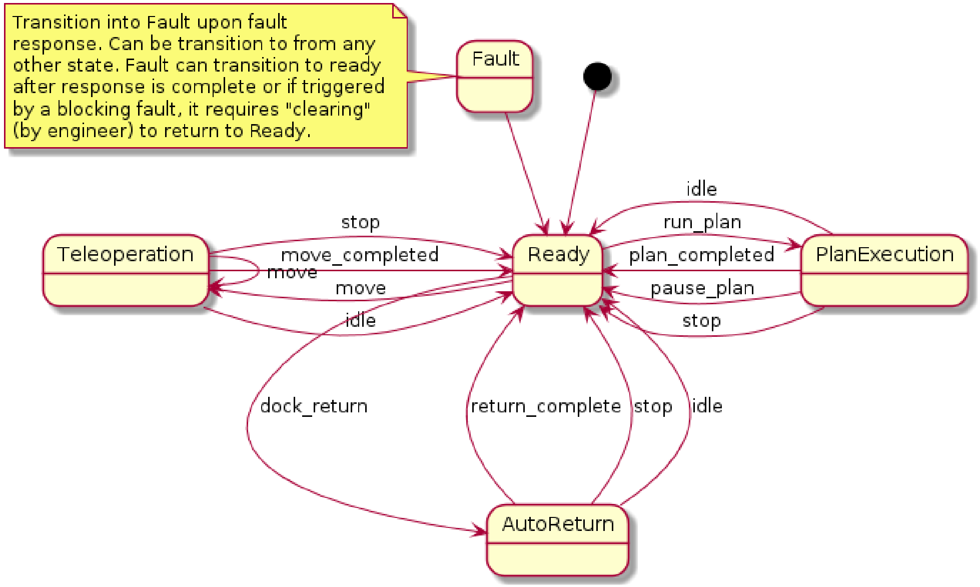
- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)



Executive

- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)

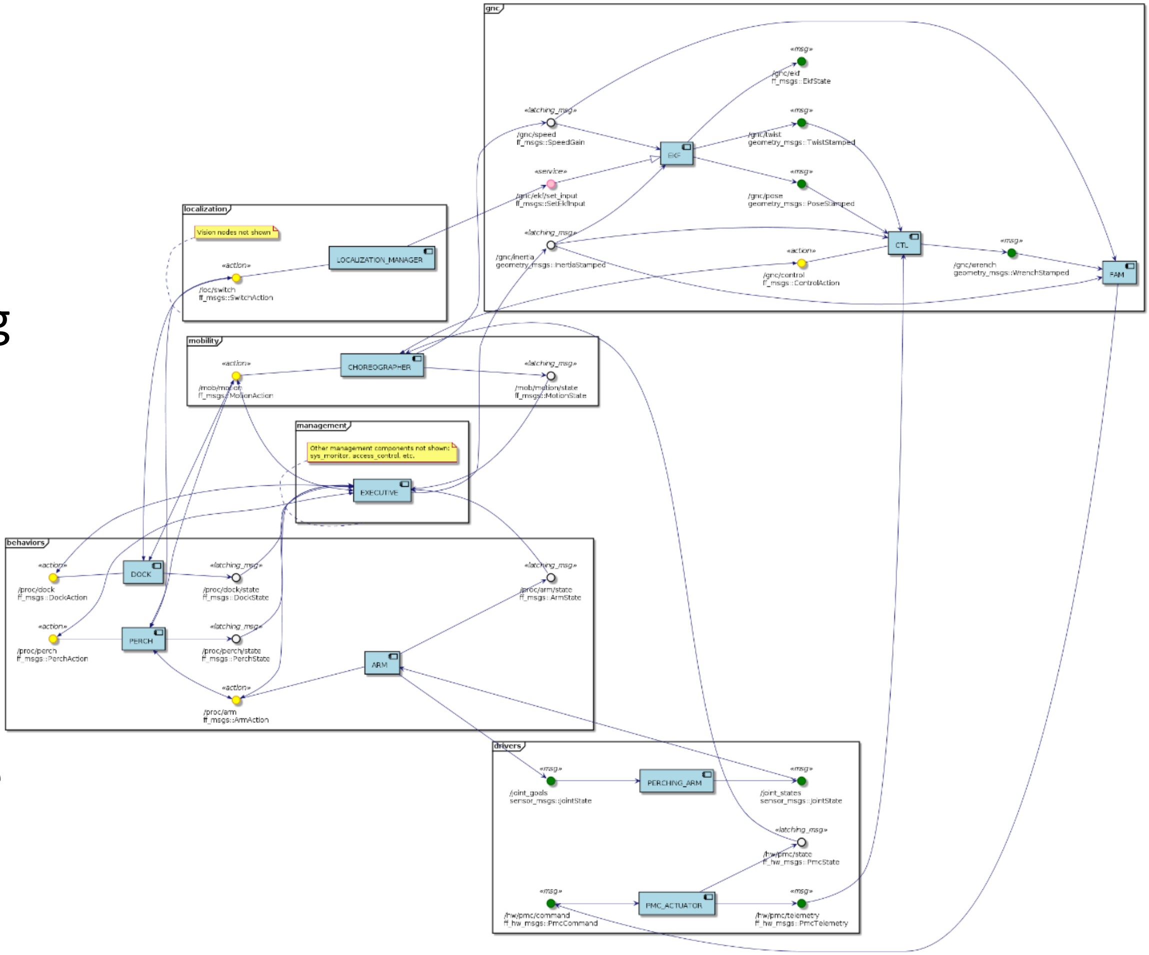
Astrobee Operating State (Executive)



Transition into Fault upon fault response. Can be transition to from any other state. Fault can transition to ready after response is complete or if triggered by a blocking fault, it requires "clearing" (by engineer) to return to Ready.

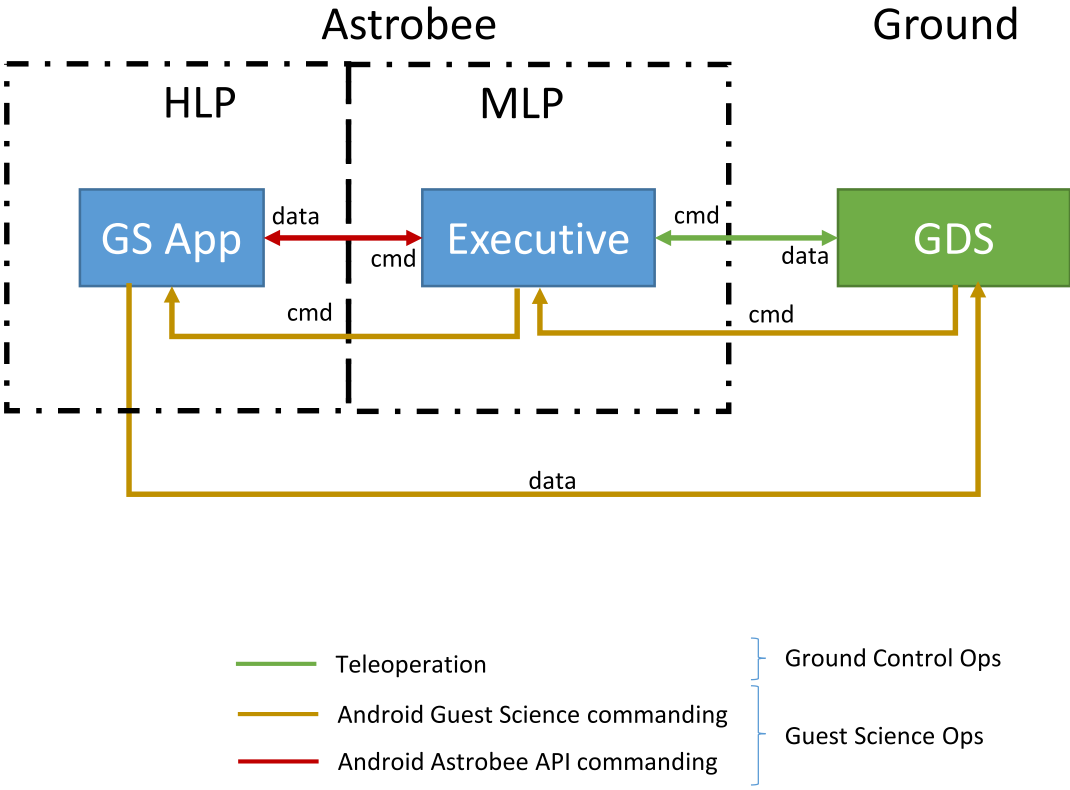
Executive

- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)



Executive

- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)



Astrobee

Ground

HLP

MLP

GS App

Executive

GDS

data

cmd

cmd

data

cmd

cmd

data

🟩 Teleoperation

🟨 Android Guest Science commanding

🟥 Android Astrobee API commanding

Ground Control Ops

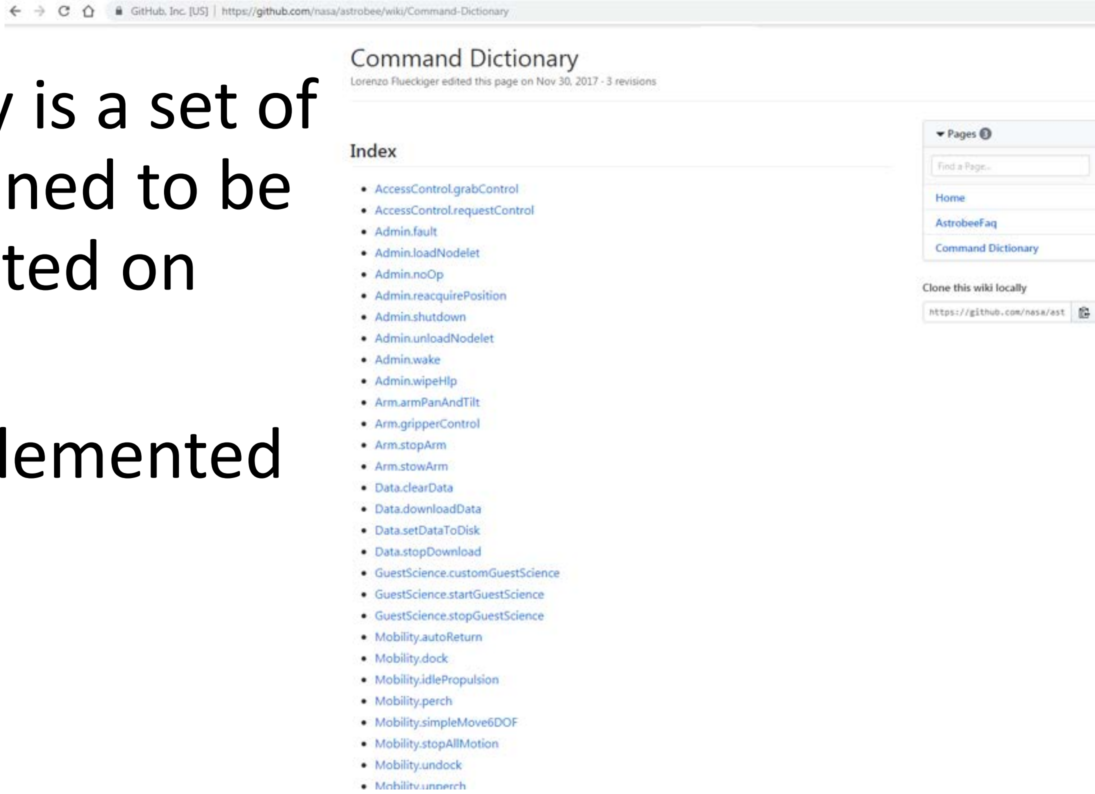
Guest Science Ops

Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

Command Dictionary

- Command Dictionary is a set of functions either planned to be or already implemented on Astrobee.
- Its functions are implemented in the Executive


GitHub, Inc. [US] | https://github.com/nasa/astrobee/wiki/Command-Dictionary
Pages 3
Find a Page...
Home
AstrobeeFaq
Command Dictionary
Clone this wiki locally
https://github.com/nasa/ast

Command Dictionary

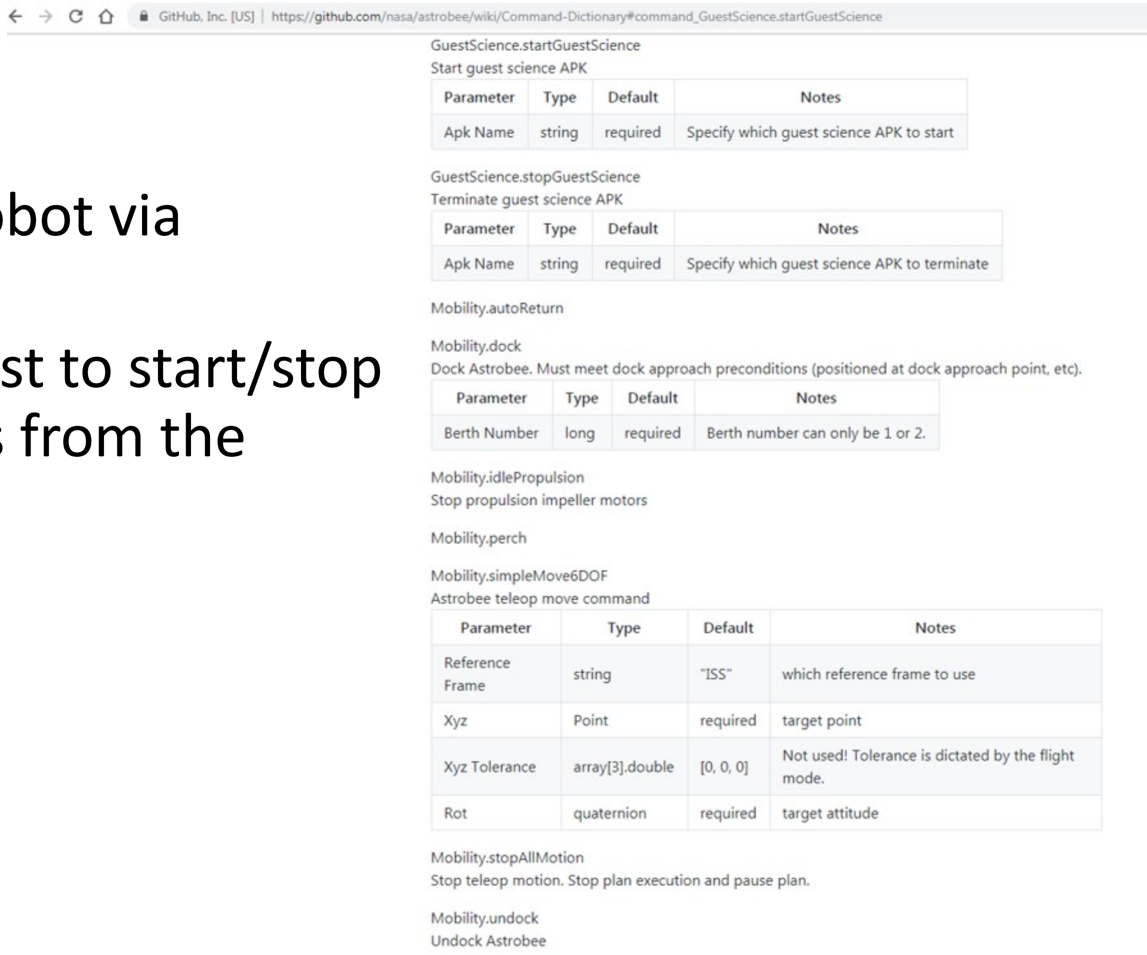
Lorenzo Flueckiger edited this page on Nov 30, 2017 · 3 revisions

Index

- AccessControl.grabControl
- AccessControl.requestControl
- Admin.fault
- Admin.loadNodelet
- Admin.noOp
- Admin.reacquirePosition
- Admin.shutdown
- Admin.unloadNodelet
- Admin.wake
- Admin.wipeHlp
- Arm.armPanAndTilt
- Arm.gripperControl
- Arm.stopArm
- Arm.stowArm
- Data.clearData
- Data.downloadData
- Data.setDataToDisk
- Data.stopDownload
- GuestScience.customGuestScience
- GuestScience.startGuestScience
- GuestScience.stopGuestScience
- Mobility.autoReturn
- Mobility.dock
- Mobility.idlePropulsion
- Mobility.perch
- Mobility.simpleMove6DOF
- Mobility.stopAllMotion
- Mobility.undock
- Mobility.unperch

Command Dictionary

- These functions
  - help control the robot via commands
  - allow guest scientist to start/stop their Android apps from the ground


GitHub, Inc. [US]
https://github.com/nasa/astrobee/wiki/Command-Dictionary#command_GuestScience.startGuestScience

GuestScience.startGuestScience
Start guest science APK

Parameter	Type	Default	Notes
Apk Name	string	required	Specify which guest science APK to start

GuestScience.stopGuestScience
Terminate guest science APK

Parameter	Type	Default	Notes
Apk Name	string	required	Specify which guest science APK to terminate

Mobility.autoReturn

Mobility.dock
Dock Astrobee. Must meet dock approach preconditions (positioned at dock approach point, etc).

Parameter	Type	Default	Notes
Berth Number	long	required	Berth number can only be 1 or 2.

Mobility.idlePropulsion
Stop propulsion impeller motors

Mobility.perch

Mobility.simpleMove6DOF
Astrobee teleop move command

Parameter	Type	Default	Notes
Reference Frame	string	"ISS"	which reference frame to use
Xyz	Point	required	target point
Xyz Tolerance	array[3].double	[0, 0, 0]	Not used! Tolerance is dictated by the flight mode.
Rot	quaternion	required	target attitude

Mobility.stopAllMotion
Stop teleop motion. Stop plan execution and pause plan.

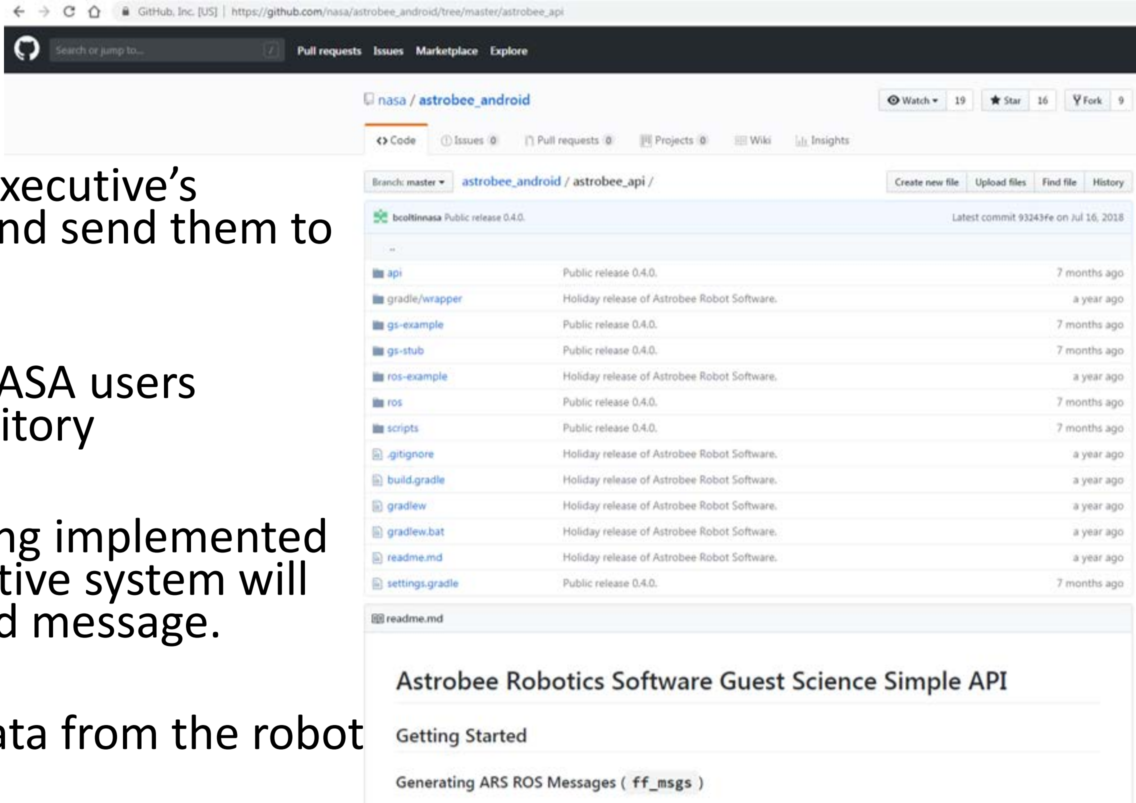
Mobility.undock
Undock Astrobee

Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

Libraries

- Astrobee API Library
  - Java API/library wraps Executive's Dictionary commands and send them to the robot.
  - Found at GitHub Non-NASA users Astrobee Android repository
  - If a command is not being implemented yet, the Astrobee Executive system will return a failed command message.
  - Provide means to get data from the robot
  - Utilized by the Guest Science Code (Android App) at the HLP



bcoltinnasa Public release 0.4.0. Latest commit 93243fe on Jul 16, 2018

..

api	Public release 0.4.0.	7 months ago
gradle/wrapper	Holiday release of Astrobee Robot Software.	a year ago
gs-example	Public release 0.4.0.	7 months ago
gs-stub	Public release 0.4.0.	7 months ago
ros-example	Holiday release of Astrobee Robot Software.	a year ago
ros	Public release 0.4.0.	7 months ago
scripts	Public release 0.4.0.	7 months ago
.gitignore	Holiday release of Astrobee Robot Software.	a year ago
build.gradle	Holiday release of Astrobee Robot Software.	a year ago
gradlew	Holiday release of Astrobee Robot Software.	a year ago
gradlew.bat	Holiday release of Astrobee Robot Software.	a year ago
readme.md	Holiday release of Astrobee Robot Software.	a year ago
settings.gradle	Public release 0.4.0.	7 months ago

readme.md

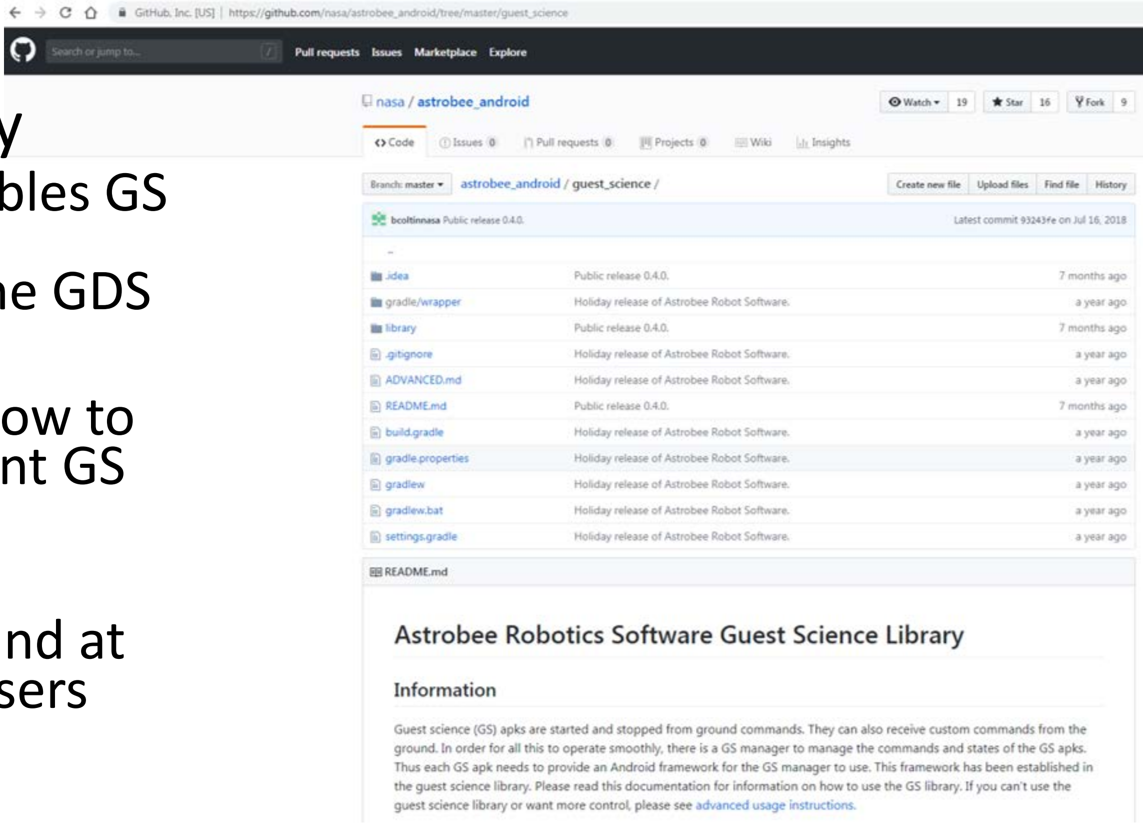
Astrobee Robotics Software Guest Science Simple API

Getting Started

Generating ARS ROS Messages (`ff_msgs`)

Libraries

- Guest Science Library
  - Java library that enables GS applications to be commanded from the GDS
  - Sets a standard on how to design and implement GS apps
  - Examples can be found at GitHub Non-NASA users repository
  - Utilized by the Guest Science Service (Android App) at the HLP



GitHub, Inc. [US] | https://github.com/nasa/astrobee_android/tree/master/guest_science

nasa / astrobee_android

astrobee_android / guest_science /

bcoltinnasa Public release 0.4.0. Latest commit 93243fe on Jul 16, 2018

..

.idea	Public release 0.4.0.	7 months ago
gradle/wrapper	Holiday release of Astrobee Robot Software.	a year ago
library	Public release 0.4.0.	7 months ago
.gitignore	Holiday release of Astrobee Robot Software.	a year ago
ADVANCED.md	Holiday release of Astrobee Robot Software.	a year ago
README.md	Public release 0.4.0.	7 months ago
build.gradle	Holiday release of Astrobee Robot Software.	a year ago
gradle.properties	Holiday release of Astrobee Robot Software.	a year ago
gradlew	Holiday release of Astrobee Robot Software.	a year ago
gradlew.bat	Holiday release of Astrobee Robot Software.	a year ago
settings.gradle	Holiday release of Astrobee Robot Software.	a year ago

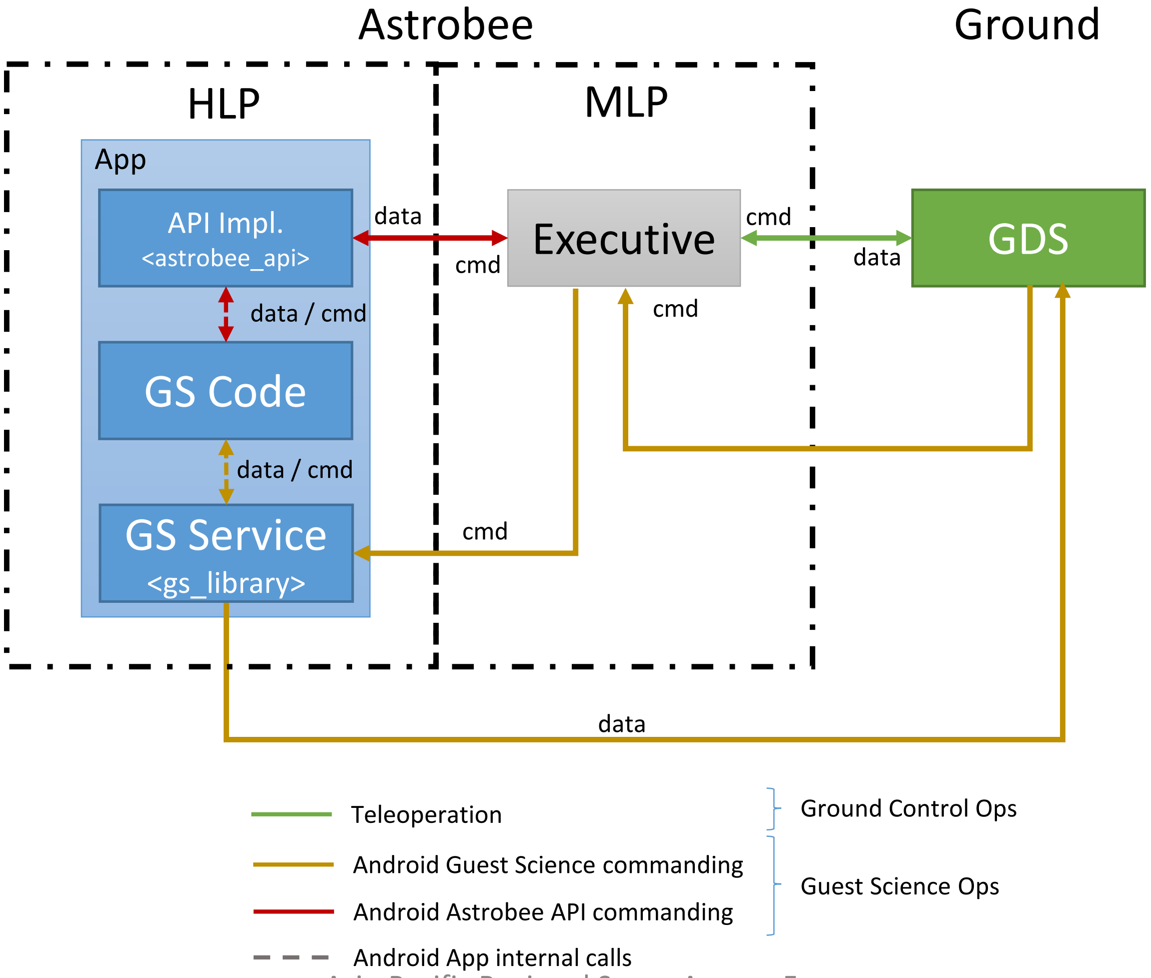
README.md

Astrobee Robotics Software Guest Science Library

Information

Guest science (GS) apks are started and stopped from ground commands. They can also receive custom commands from the ground. In order for all this to operate smoothly, there is a GS manager to manage the commands and states of the GS apks. Thus each GS apk needs to provide an Android framework for the GS manager to use. This framework has been established in the guest science library. Please read this documentation for information on how to use the GS library. If you can't use the guest science library or want more control, please see advanced usage instructions.

Libraries


Astrobee
Ground
HLP
MLP
App
API Impl.
<astrobee_api>
data
cmd
Executive
cmd
data
GDS
data / cmd
GS Code
cmd
data / cmd
GS Service
<gs_library>
cmd
data
Teleoperation
Android Guest Science commanding
Android Astrobee API commanding
Android App internal calls
Ground Control Ops
Guest Science Ops

Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications

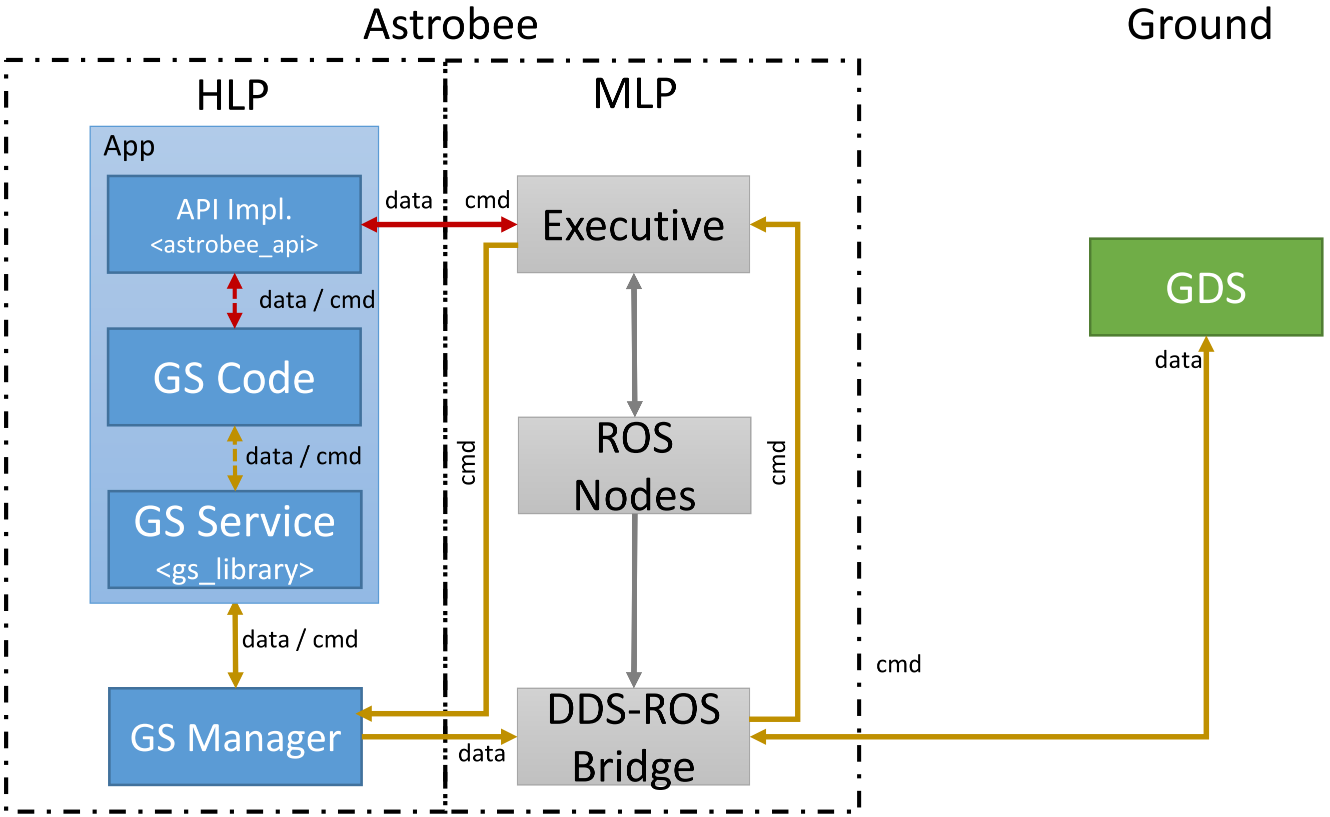
Guest Science Applications

- Guest Science Applications
  - Two types:
    - Primary (commands robot, gets data, internal computation, payload communication)
    - Secondary (gets data, internal computation, payload communication)
  - Standard components:
    - GS API implementation (astrobee_api)
    - GS Service implementation (gs_science_lib)
  - Remote capabilities from GDS:
    - Starting/Stopping
    - Custom commanding and data feedback

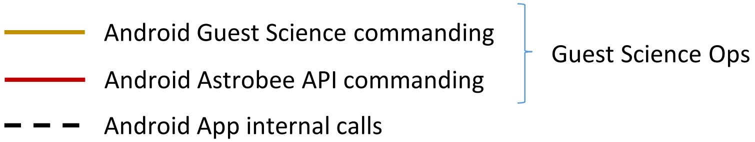
Guest Science Applications

- Guest Science Manager
  - GS Manager is an Android App of its own (Android Service)
  - Manage the commands and states of the GS Apps
  - Each GS App needs to provide an Android framework for the GS Manager to use
  - In charge of interfacing GS Android Apps and the GDS through the Executive system and the DDS-ROS Bridge

Guest Science Applications

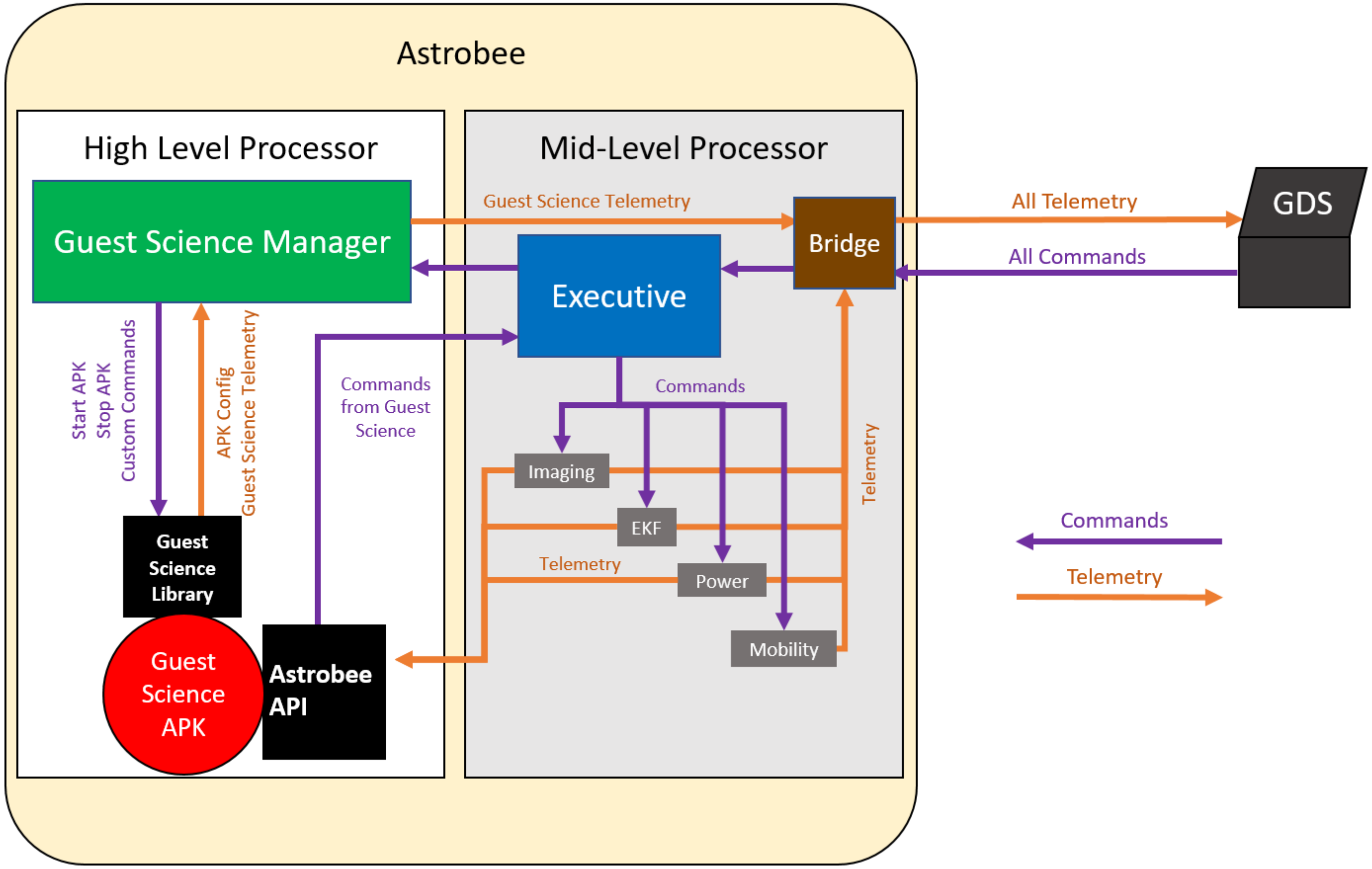


*Teleoperation is not represented in this data/cmd flow diagram


Android Guest Science commanding
Android Astrobee API commanding
Android App internal calls
Guest Science Ops

Guest Science Applications

Astrobee Guest Science Architecture


Astrobee
High Level Processor
Mid-Level Processor
Guest Science Manager
Guest Science Telemetry
Bridge
All Telemetry
GDS
All Commands
Executive
Start APK
Stop APK
Custom Commands
APK Config
Guest Science Telemetry
Commands from Guest Science
Commands
Imaging
EKF
Power
Mobility
Telemetry
Telemetry
Guest Science Library
Guest Science APK
Astrobee API
Commands
Telemetry