Flight Hardware Development for a Space-based Robotic Assembly and Servicing Testbed

by

Bryan Patrick McCarthy

B.S., Astronautical Engineering
United States Air Force Academy (2012)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Signature redacted

Author .....
Department of Aeronautics and Astronautics
May 22, 2014

Signature redacted

Certified by .....
Alvar Saenz-Otero
Director, Space Systems Laboratory
Thesis Supervisor

Signature redacted

Certified by .....
David W. Miller
Professor of Aeronautics and Astronautics
Thesis Supervisor

Signature redacted

Accepted by .....
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Disclaimer: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense, or the United States Government.

Flight Hardware Development for a Space-based Robotic Assembly and Servicing Testbed

by

Bryan Patrick McCarthy

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Robotic assembly and servicing missions in space are becoming increasingly attractive for their potential to expand space capabilities and save money. Future missions may construct large systems on-orbit, service existing space assets, or remove retired satellites from valuable locations in the Geosynchronous orbit, among other things. Due to the high-risk nature of these missions, rigorous test facilities are a necessity. This thesis examines the existing testbeds for robotic assembly and servicing technologies and argues that a new, space-based testbed is necessary. It presents initial ground testing results for applicable control concepts, which also indicate that the dynamic authenticity associated with a six-degree-of-freedom on-orbit testbed is crucial for further development. This thesis then presents the requirements for such a testbed and describes the SPHERES Facility on the International Space Station. The facility, created by members of MIT's Space Systems Laboratory, has many of the desired testbed characteristics and can be easily expanded to meet the requirements through a hardware augmentation known as the Halo. The thesis develops the requirements for the Halo and then steps through the conception, design, and implementation of that hardware, along with the planned operations aboard the International Space Station.

Thesis Supervisor: Alvar Saenz-Otero
Title: Director, Space Systems Laboratory

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics

Acknowledgments

DARPA sponsored a majority of this work under NASA Contract #NNH11CC26C. The United States Air Force Space and Missile Systems Center also sponsored part of this work under NASA Contract #NNH11CC25C. Thank you for your support of this research.

Thank you to the Massachusetts Institute of Technology, the Space Systems Laboratory, and the Space Engineering Academy for the opportunity to gain this valuable education.

I would also like to thank my loved ones and all those who have made this experience so enjoyable, including the Situation Room, the Men of Pine, Park Street Church, the Wolfpack, and the rest of my boys. I thank God for you all.

Contents

List of Figures

List of Tables

List of Acronyms

AMPHIS - Autonomous Multi-Agent Physically Interacting Spacecraft

ARMADAS - Agile Reconfigurable Modules with Autonomous Docking for Assembly and Servicing

CAD - Computer-aided Design

CDIO - Conceive, Design, Implement, Operate

CESSORS - Chinese Exper- imental Space System for On-orbit Robotistic Services

CONOPS - Concept of Operations

DAC - Direct Adaptive Control

DARPA - Defense Advanced Research Projects Agency

DLR - Deutsches Zentrum *für* Luft- und Raumfahrt

DOF - Degrees of Freedom

ETS-VII - Engineering Test Satellite Number 7

FDIR - Fault Detection, Isolation, and Recovery

FREND - Front-End Robotics Enabling Near-Term Demonstration

GEO - Geosynchronous Orbit

GNC - Guidance, Navigation, and Control

HOMER - Holonomic Omni-directional Motion Emulation Robot

HP - Halo Port

HPG - Halo Port Goggles

HRSDM - Hubble Robotic Servicing and De-orbit Mission

HST - Hubble Space Telescope

ISS - International Space Station

LASR - The Land, Air, and Space Robotics Laboratory

LED - Light-Emitting Diode

LEO - Low Earth Orbit

LOS - Line of Sight

MEDUSA - Modular Experimental testbed for the Development of Utilization, Servicing, and Assembly technologies

NASA - National Aeronautics and Space Administration

NRL - Naval Research Laboratory

PCB - Printed Circuit Board

PD - Proportional-Derivative

RGA - Reduced Gravity Aircraft

RINGS - Resonant Inductive Near-Field Generation System

RMS - Robotic Manipulator System

ROKVISS - RObot Komponent Verification on ISS

SLAM - Simultaneous Localization and Mapping

SPHERES - Synchronized Position Hold, Engage, Reorient Experimental Satellites

STEP - Suspended Target Emulation Pendulum

SWARM - Self-assembling Wireless Autonomous Reconfigurable Modules

TECSAS - TEChnology SAtellite for demonstration and verifica- tion of Space systems

USB - Universal Serial Bus

VBN - Vision-Based Navigation

VERTIGO - Visual Estimation for Relative Tracking and Inspection of Generic Objects

Chapter 1

Introduction

1.1 Motivation

In recent years, interest in the concepts of robotic assembly and servicing has grown substantially. As political debates over the prioritization of finances are causing many to reconsider their investments, decision-makers are looking for cheaper options for space assets with even greater capabilities. Robotic assembly and servicing may be the answer.

Robotic assembly generally encompasses ideas such as the construction of large or complex space systems on-orbit. Such systems include large space telescopes, lunar observatories or bases, space stations, or interplanetary vehicles, among others [1]. These systems may be assembled in space for financial reasons or due to the limitations of launch vehicle capabilities (namely, mass and volume).

Space assets, specifically in the Geostationary Orbit (GEO), have significant strategic value for communications, weather, television, science, and military applications, to name a few. With such a large amount of capital invested in the space realm, it is only natural for investors to desire some sort of servicing capability. As these assets break down due to malfunctions, age, or damage from debris or other satellites, robotic servicing satellites could become an invaluable resource. With this in mind, agencies are already starting to research and develop robotic servicing vehicles that may be suitable for the job [2].

More and more space systems are being launched into orbit each year, meaning that real estate in desirable orbits is increasing in value. For example, in the crowded GEO orbit, defunct or retired satellites can occupy precious space that new assets could be using. While many of these satellites were designed to thrust themselves into the graveyard orbit at the end of their mission lives, some have failed to do so because of various malfunctions. Such satellites may require robotic servicers to pull them into the graveyard orbit [3].

Other defunct or retired satellites still have extremely valuable equipment onboard that is no longer being used. Since today it costs approximately $40,000 to $60,000 per kilogram to GEO, it may be more economically feasible to harvest these components than re-develop and re-launch them from the ground. Working components could be re-used, re-purposed, or re-furbished by a robotic servicer [4].

Because missions to conduct robotic assembly or servicing in space cost billions of dollars and have very high risks due to both the nature of proximity operations in space and the general lack of flight heritage, low-risk testbeds on the ground and in space can be of high value to such programs. This thesis describes some of the existing testbeds that can be used specifically to test robotic assembly and servicing technologies and algorithms. It also describes upgrades to a specific on-orbit testbed which will buy down risk for future robotic assembly and servicing missions.

1.2 Literature Review

The Literature Review explores past and future missions related to robotic assembly and servicing, previous research conducted regarding different methods of robotic assembly and servicing, and relevant testbeds both on the ground and in space.

1.2.1 Robotic Assembly and Servicing Missions

In 1993, a small robot performed several tasks on-board a spacecraft for the first time ever. The robot, part of an experiment called ROTEX, flew aboard Space Shuttle Columbia on Space Transportation System 55. The robot performed tasks

such as assembling a simple truss structure, connecting and disconnecting an electrical plug, and grabbing a free-floating object with an on-board gripper. The robot was teleoperated by astronauts on-board the shuttle and was also teleoperated by humans on the ground [5]. This was the beginning of robotic assembly and servicing missions.

The successful construction of both the Russian Mir and the International Space Station (ISS) demonstrated that it was possible to assemble a large structure in space [6]. In preparation for those missions, the National Aeronautics and Space Administration (NASA) developed the capabilities to rendezvous with, dock to, and repair failed communications satellites in the 1980's. They also were able to assemble large trusses. Those competencies were crucial in the development of Mir and the ISS, and they became the foundation for the astronaut servicing of the Hubble Space Telescope (HST) [7].

Multiple HST servicing missions since 1993 have proven that satellite servicing is both feasible and effective. Servicing missions to HST have significantly increased the optical capabilities of the telescope. They also have increased the computational, data handling, power, and pointing capabilities of the telescope by upgrading computers, instruments, and other sensors [8]. These HST servicing missions were also used to design the Hubble Robotic Servicing Vehicle that was planned to service HST for the final time through the Hubble Robotic Servicing and De-orbit Mission (HRSDM), although that mission was cancelled [7]. Nonetheless, extensive work was done in simulation of the capture, berthing, and manipulation phases of that planned mission [9].

The Shuttle Remote Manipulator System, also known as the "Canadarm," was another major milestone in space robotics. The six-degree-of-freedom (6-DOF) robotic manipulator first began working in 1981 and since then has been used in over 50 missions, executing tasks such as the deployment, manipulation, and retrieval of satellites, the aid of astronaut extravehicular activities, and the construction of the ISS and servicing of HST, mentioned earlier. The arm has yet to fail at any mission [10].

The Engineering Test Satellite Number 7 (ETS-VII) mission, launched in 1997,

was used to test a number of space servicing technologies including inspection, handling equipment with robotic arms, maneuvering truss structures, assembling antennas, and the rendezvous and docking of a chaser satellite with a target satellite. The mission was successful and also gave the space robotics community insights into future challenges regarding robotic assembly and servicing technology [10].

The ISS now has a number of its own robotic arms, such as the Space Station Remote Manipulator System, which began functioning in 2001. This arm has a Latching End Effector on either end, so that either end can function as a "shoulder" or a "hand." The arm also has cameras and advanced vision and sensing systems to keep track of its own movements and that of nearby payloads, ensuring collision avoidance and operational safety [10]. Its counterpart is the Special Purpose Dexterous Manipulator, known as "Dextre," which is much smaller and was launched in 2008. Canada's Mobile Servicing System comprises both of these manipulators and the Mobile Base System, which is used to relocate the arms [11]. The Robotic Refueling Mission used Dextre to demonstrate teleoperated manipulation of a module that was not designed to be serviced. The mission was the first time Dextre was used for research, and the robotic manipulators were able to cut and peel back thermal blankets, unscrew caps, adjust valves, and transfer fluid [12].

The Japanese Experiment Module Remote Manipulator System was one of Japan's contributions to the ISS. The 6-DOF main arm is almost 10 meters long, and it can also grapple the Small Fine Arm, which is used for more dexterous missions [11]. The main arm was launched in 2008, and the Small Fine Arm was launched in 2009.

RObot Komponent Verification on ISS (ROKVISS) is a project by DLR, the German space agency, that is meant to test out robotic hardware components and different control strategies. The ROKVISS hardware—a manipulator with two joints—was attached to the Russian Service Module of the ISS in 2005, and since then it has been used with varying levels of autonomy, from teleoperation to full system autonomy. The system has been used to verify the robotic joints and the feasibility of teleoperation [3].

More recently, in 2007, the Defense Advanced Research Projects Agency's (DARPA's)

and Boeing's Orbital Express mission successfully demonstrated the ability for two spacecraft to autonomously rendezvous, dock, and execute tasks such as refueling. DLR has been researching the concept of autonomously rendezvousing, docking, and manipulating a spacecraft that is unprepared for such actions. In that vein, they have been pursuing two missions: Deutsche Orbitale Servicing Mission and Orbital Life Extension Vehicle. The first seeks to capture a non-cooperative tumbling satellite in Low-Earth Orbit and de-orbit the combined system. The second will attach a vehicle to a geostationary satellite that has run out of fuel to aid with stationkeeping, thereby extending its mission life [13].

Another mission called the TEChnology SAtellite for demonstration and verification of Space systems (TECSAS) is also planned to use a manipulator matured under the ROKVISS program in order to fly to an existing satellite, attach to it, manipulate it, be operated via telepresence, operate autonomously, and aid in the de-orbiting of the satellite [3].

The Chinese Shenzhen Space Technology Center is developing the Chinese Experimental Space System for On-orbit Robotistic Services (CESSORS). The CESSORS mission involves a larger, free-flying satellite with a Robotic Manipulator System (RMS) and a separate, smaller, free-flying target satellite. The larger satellite will grasp and manipulate the target satellite [14]. The 6-DOF RMS has a two-fingered gripper, control system, and cameras [15]. The main components of the mission are shown in Figure 1-1.

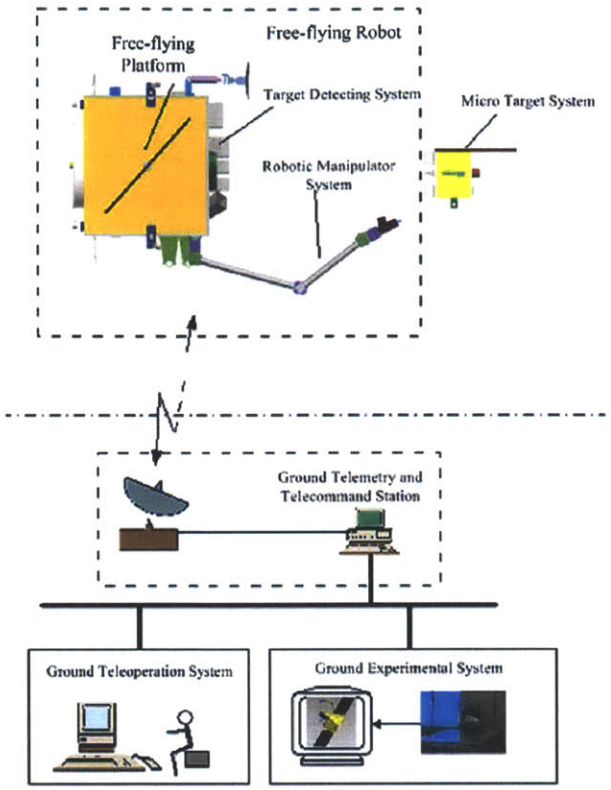


Figure 1-1: Primary Components of the CESSORS Mission [14]

The 7-DOF, 11 meter European Robotics Arm was developed for use with the Russian Segment of the ISS. The arm will be used for maintenance of the ISS as well as handling payloads. It can be operated by crew members either inside or outside the ISS. The arm is scheduled to be launched in 2015 [11].

DARPA's Phoenix mission, with a planned launch in 2017, will use the Naval Research Laboratory's (NRL's) Front-End Robotics Enabling Near-Term Demonstration (FREND) robotic arm to capture and repurpose a retired geostationary satellite. The mission will also determine the feasibility of a new "satlet" concept, which aims to cellularize spacecraft in order to break free of traditional spacecraft morphology [16]. Figure 1-2 shows the Phoenix mission architecture.

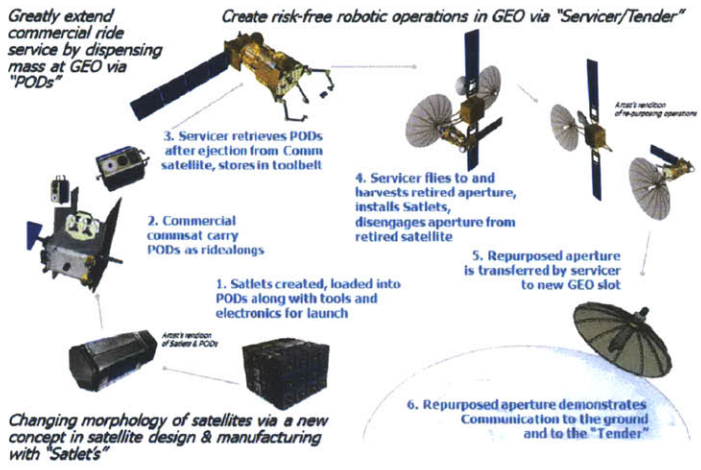


Figure 1-2: Phoenix Mission Architecture [16]

Future missions in the robotic assembly and servicing field may also include refurbishing or repurposing retired satellites, cleaning up space debris, constructing orbiting space hotels and solar power generators, space transportation systems, or resource mining systems [10]. Significant research is already being conducted in many of these areas [17].

1.2.2 Ground-based Robotic Assembly and Servicing Testbeds for On-orbit Operations and Related Research

There are numerous testbeds on the ground that are used to test various robotic assembly and servicing technologies, hardware, and methodologies. This section highlights those testbeds.

The Modular, Reconfigurable, High Energy concept was a joint effort by NASA's Marshall Space Flight Center, Lockheed Martin Advanced Technology Center, and several other subcontractors to mature technology that could be used to assemble groups of solar-powered bus modules with propulsion modules on-orbit, as well as other large space structures and aggregate spacecraft. They developed a ground-based testbed to develop these technologies known as the Controls and Automation Laboratory. The testbed includes three robotic satellite emulators which float on a

flat air-bearing surface made of granite that is 12 by 24 feet across, as shown in Figure 1-3 [18].

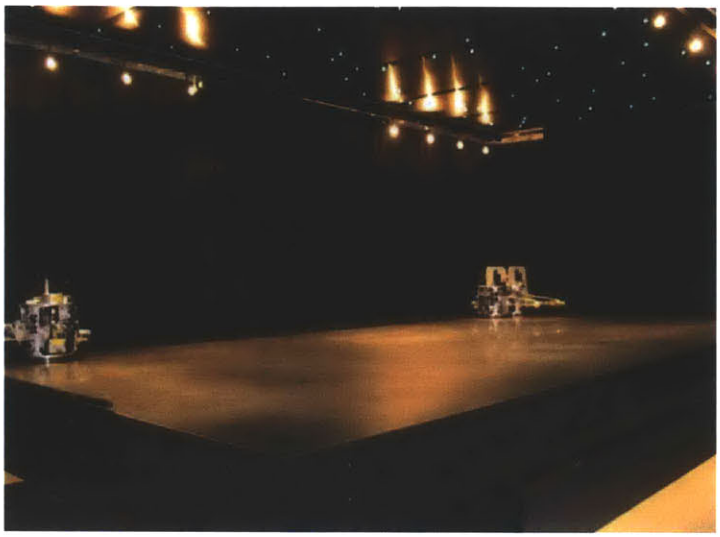


Figure 1-3: Granite Table at the Controls and Automation Laboratory [18]

The robotic vehicles each have 8 thruster valves and a reaction wheel, and can stay afloat for a maximum of 30 minutes. The vehicles have an upward-facing "star tracking" sensor, which looks at a custom starfield of LEDs on the ceiling, and a sideways-facing camera for docking. Additional hardware that has been attached to the vehicles includes docking latches and a deployable boom that can be used to expand the aggregate structure, as pictured in Figure 1-4, although other sensors and actuators could be attached as well [18].

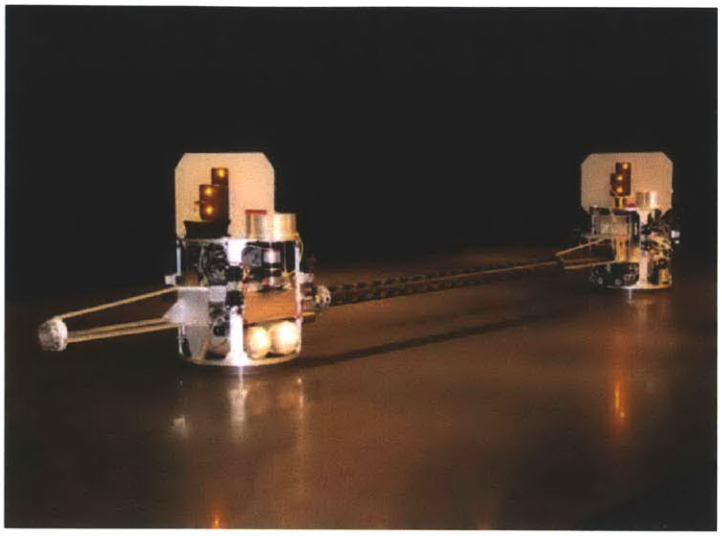


Figure 1-4: Deployable and Retractable Boom Connecting Two Vehicles [18]

The testbed has been used for autonomous path-planning, docking, reconfiguring of the control system, and deployment and retraction of the boom [18].

The Autonomous Docking testbed in the Spacecraft Robotics Laboratory at the Naval Postgraduate School is similar to the one at Lockheed. In this testbed, two spacecraft simulators can float on a 4.9 meter by 4.3 meter epoxy flat floor using air pads. Each of the vehicles include five modular decks that can easily be replaced. The first deck includes the eight-thruster propulsion system and floatation system. The second includes the docking system, the third and fourth include a reaction wheel system and the electronics, and the fifth includes a camera. The vehicles also include inertial measurement units, and are pictured in Figure 1-5 [19].

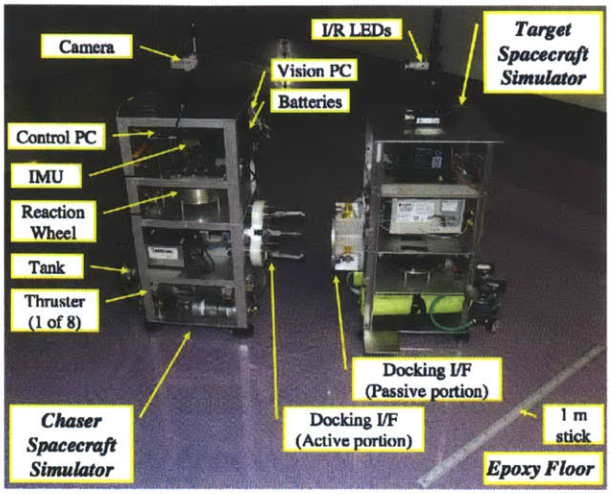


Figure 1-5: Vehicles in the Autonomous Docking Testbed [19]

Through a project called Autonomous Multi-Agent Physically Interacting Spacecraft (AMPHIS), the vehicles were upgraded to be smaller and lighter and were equipped with a new air system that can float the vehicles for up to 75 minutes, rotating thruster nozzles, and a miniature single gimbaled control moment gyroscope, as pictured in Figure 1-6 [19].

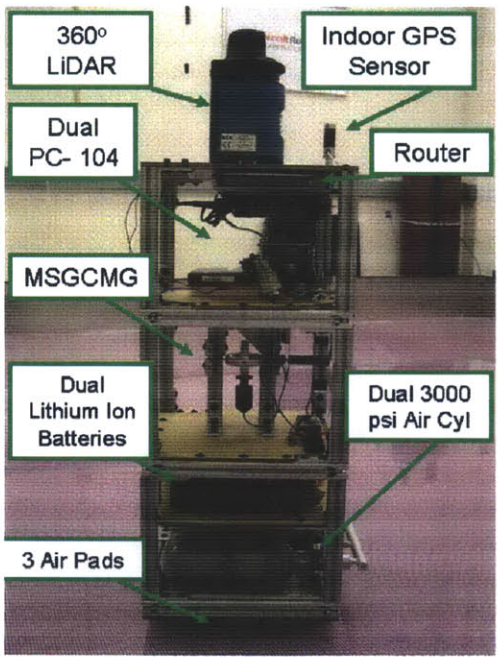


Figure 1-6: Upgraded AMPHIS Vehicle [19]

The MIT Field and Space Robotics Laboratory's Free-Flying Robotics Testbed is also used to test different types of control algorithms, with applications to the robotic assembly of large space structures. The testbed contains several robots with multiple robotic manipulators that float on the Space System Laboratory's Flat Floor, as pictured in Figure 1-7 [20].

Figure 1-7: Two Robots on the Flat Floor with a Flexible Beam [20]

The robots each have their own thrusters, as well as force and torque sensors. One of the robots is pictured in Figure 1-8 [20].

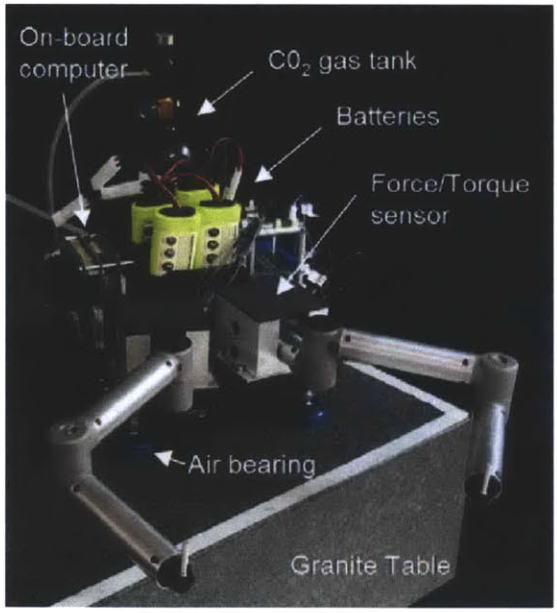


Figure 1-8: A Single Robot with Manipulators [20]

The robots have been used to transport a flexible beam across the Flat Floor, and have already experimentally shown that vibration control with robotic manipulators is much more effective than with thrusters alone. The robots are used for tests involving the manipulation and assembly of larger structures and the capture of other uncontrolled robots [20]. The testbed has also been used to test concepts for the coordination and cooperation of multiple entities [21].

The University of Southern California's Polymorphic Robotics Laboratory also has its own testbed for robotic assembly in space. The testbed consists of an air-hockey table with floating robots and structural components. The robots move around using fan propulsion, and can dock and assemble pieces together [22]. The testbed is pictured in Figure 1-9

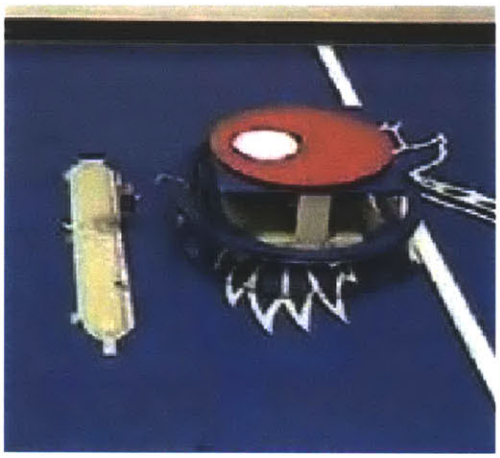


Figure 1-9: A Robot on the Air Hockey Table in the Polymorphic Robotics Laboratory [23]

Researchers at the University of Southern California are also studying free-flying "intelligent fiber/rope" "match-making" FIMER robots, which are tethers with autonomous heads on either end. The heads are capable of moving by themselves and each can reel in and out the tether to assemble components. They are also studying self-reconfigurable robots, and postulate that the future of robotic assembly involves FIMERs and other intelligent and reconfigurable robots [24].

Stanford University also has several robotic assembly and servicing testbeds in their Aerospace Robotics Laboratory. One testbed consists of a 6 foot by 12 foot

granite table, a two-link robotic manipulator arm, and several other objects that float on the table. It is pictured in Figure 1-10 [25].

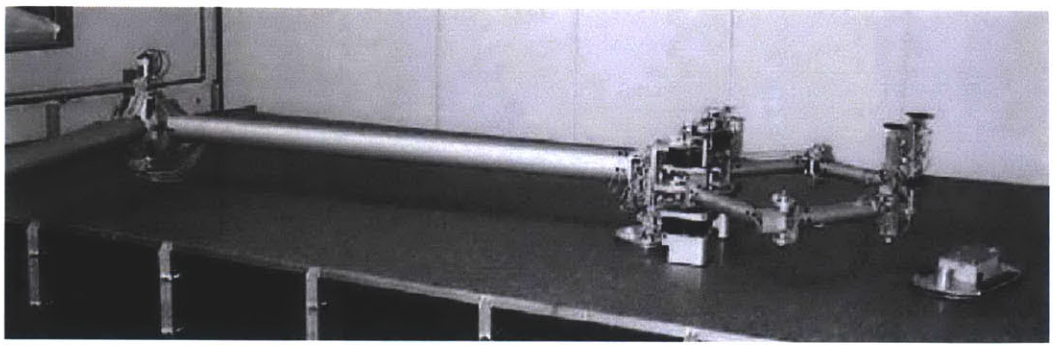


Figure 1-10: Testbed in Stanford's Aerospace Robotics Laboratory [25]

A testbed at the University of Padova has been used to test free-floating robots with extended arms. The testbed includes vehicles that float on a granite table using compressed gas and move using thrusters. One of the vehicles is shown in Figure 1-11 [26].

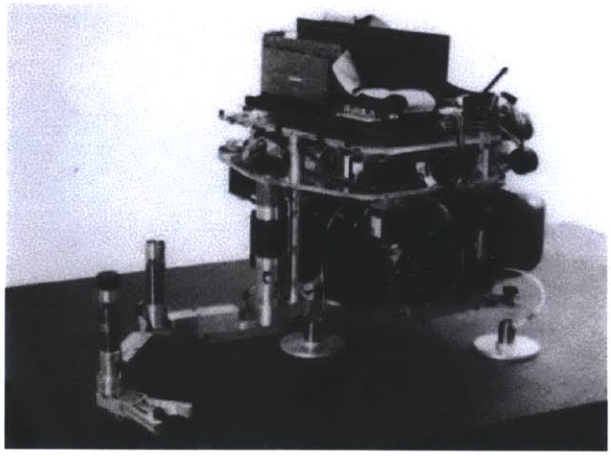


Figure 1-11: Vehicle at the University of Padova [26]

Researchers at the University of Padova have also suspended robots from the ceiling using springs, tested other robots on parabolic flights, and studied the idea of developing "free-flying" underwater robots [26].

The Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) Facility at MIT's Space Systems Laboratory has also been used in the

past to test concepts for the assembly of large space structures. The facility is described in detail in Section 1.3. In 2009, Jacob Katz used the facility to attach a flexible beam and a camera to a satellite floating on a flat floor. Katz used the models of the flexible beam and the camera, which was looking down the length of the beam, to berth a docking port at the end of the beam to one on another satellite. The setup is pictured in Figure 1-12 [27].

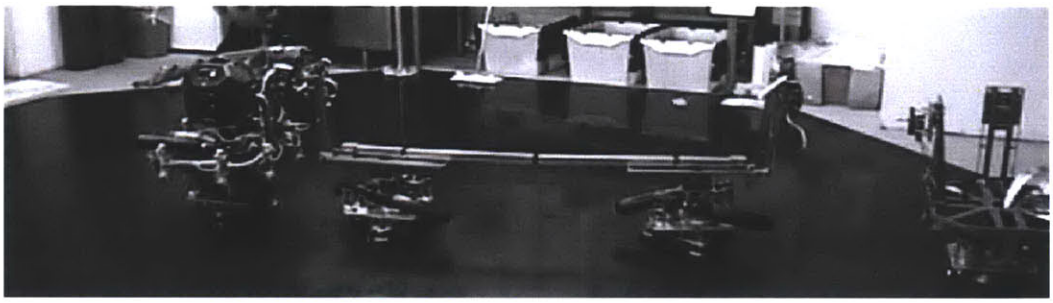


Figure 1-12: SPHERES on the Flat Floor with a Flexible Beam [27]

Further use of the SPHERES Facility for the testing of related concepts is described in detail in Chapter Two.

The Ranger arm is a dexterous manipulator that was developed at the University of Maryland's Space Systems Laboratory in the 1990's, pictured in Figure 1-13 [28].

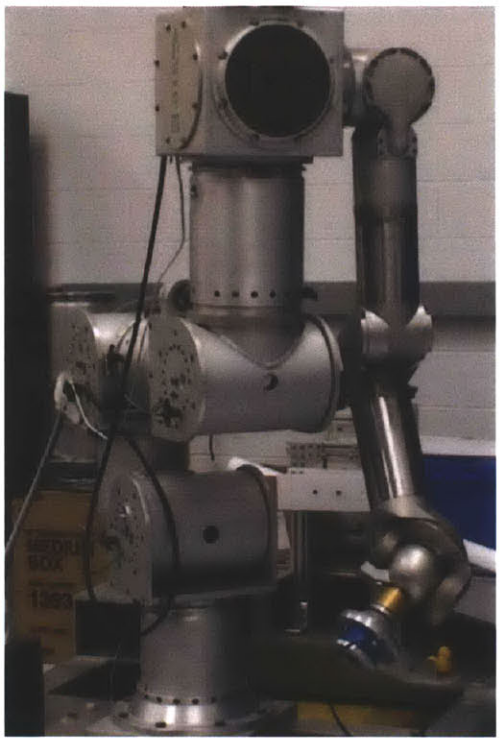


Figure 1-13: The Ranger Arm in the Lab [28]

The arm was intended to be used inside the Space Shuttle to experimentally perform a number of servicing tasks, but it never flew. Nonetheless, it still provides a valuable research tool on the ground for developing robotic servicing and assembly technologies [28].

In 1995, the Ranger Neutral Buoyancy Vehicle was completed at the University of Maryland's Space Systems Laboratory [28]. The vehicle is part of the Neutral Buoyancy Research Facility, another testbed that has been used to mature robotic assembly and servicing technologies. More recently, the facility was used to test technologies such as those that were expected to be used on HRSDM, a mission mentioned in Section 1.2.1. The tank is 15.2 meters in diameter and 7.6 meters deep. For HRSDM testing, the Ranger arm was modified to more closely match the dynamics of the Dextre arm, mentioned in Section 1.2.1, as well as to be fully functional underwater. The testbed is shown in Figure 1-14 [29].

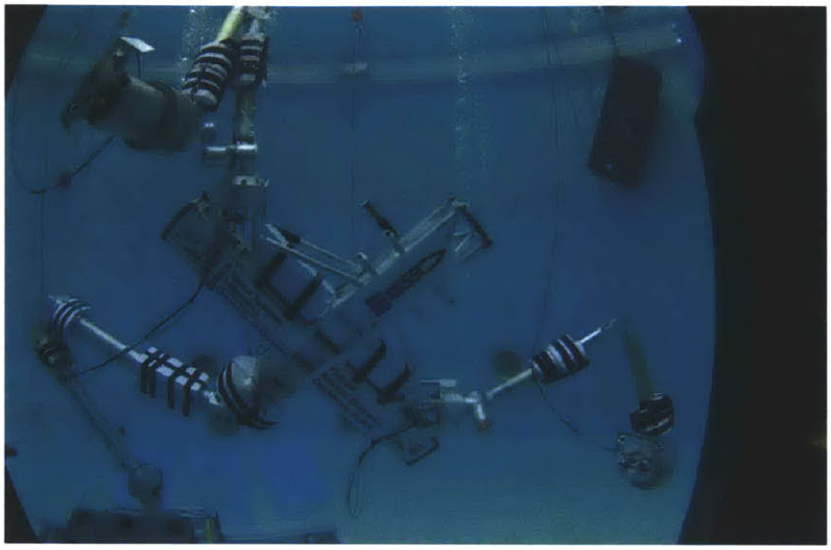


Figure 1-14: The Neutral Buoyancy Research Facility at the University of Maryland [29]

The Dextre arm also has a ground version which is a part of a testbed at NASA Goddard. The arm is identical to the one on the ISS and has a counterbalance to counteract the effects of gravity. The arm also has a force-moment sensor, camera, and a gripper. The arm is pictured in Figure 1-15 [29].

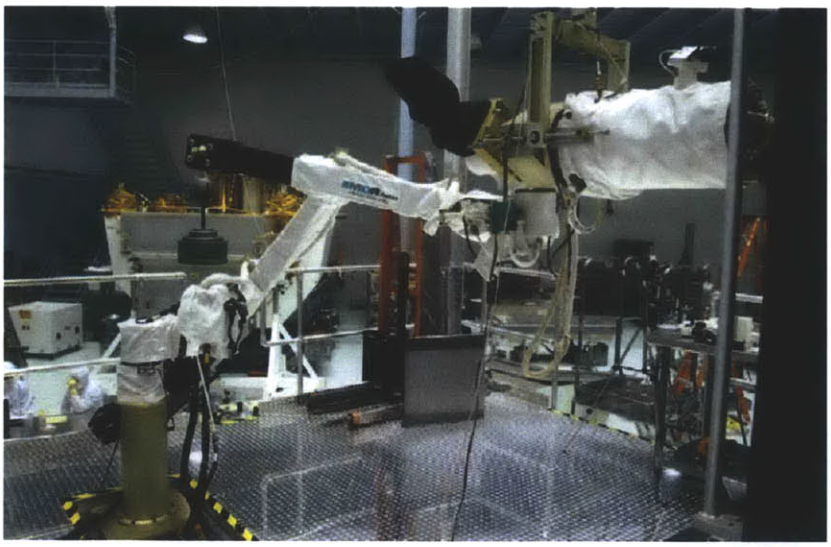


Figure 1-15: The Dextre Ground Testbed at NASA Goddard [29]

This testbed was also successfully used to check out many of the control systems and

operational concerns (such as latency) for the HRSDM [29].

Another testbed is being developed for the CESSORS mission, mentioned in 1.2.1. The testbed will be used to test its RMS, which is too weak to operate in 1-g and thus requires a gravity compensation system. The testbed includes a granite table, an air pad, and support structures. A laser tracking system is the primary sensor on the testbed, which is shown in Figure 1-16 [15].

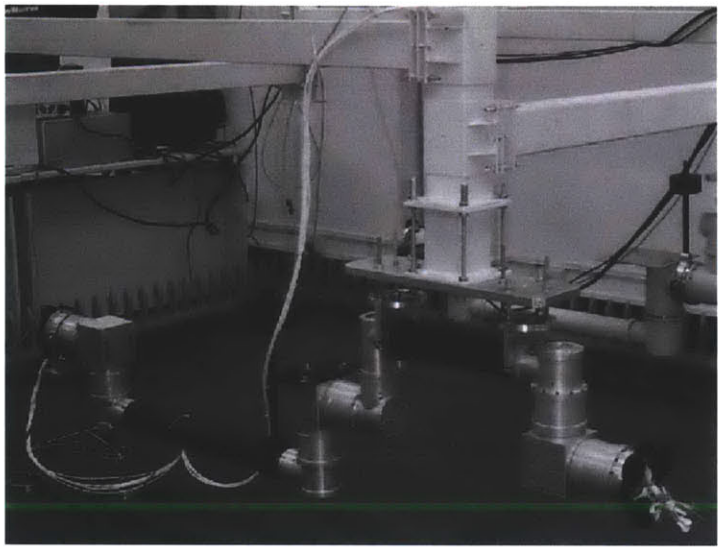


Figure 1-16: The CESSORS RMS Testbed [15]

NASA's Skyworker is a mobile robotic manipulator designed to assemble and service large space structures by walking along them. The robot is capable of carrying and manipulating massive payloads in space, and it uses continuous gait locomotion to move along a structure while smoothly transporting its payloads. The robot has eleven joints and is designed to walk 10 centimeters per second in space. It is pictured in Figure 1-17 [30].

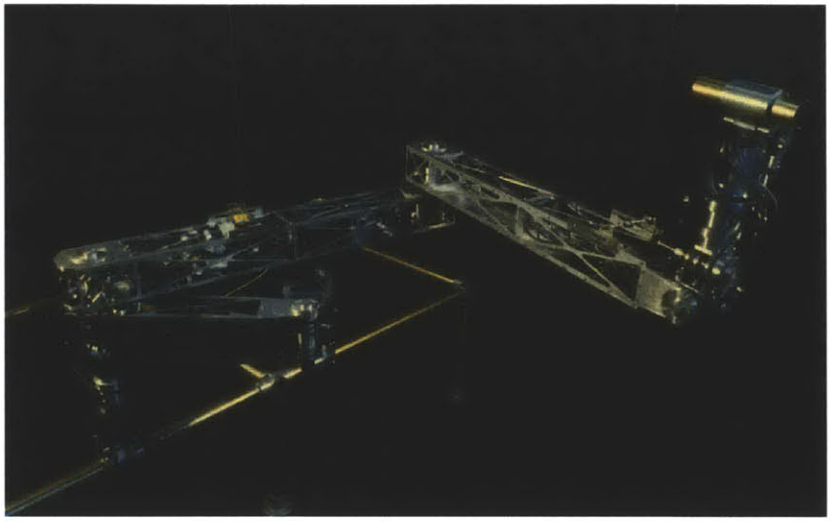


Figure 1-17: The Skyworker Robot [31]

While not in space yet, the Skyworker prototype has demonstrated the ability to walk, turn, and transport payloads on the ground with gravity compensation. This prototype is the first generation of robotic space workers and has also been very useful for determining which technologies need to be further developed [30]. The concept of robots climbing along the structure to assemble components has also been called "intelligent scaffolding" and is being studied by NASA's Space Technology Mission Directorate [32].

Researchers in the Automated Structure Assembly Laboratory at NASA Langley Research Center demonstrated the assembly of an 8 meter planar structure which is made up of 102 truss elements, as pictured in Figure 1-18 [33].

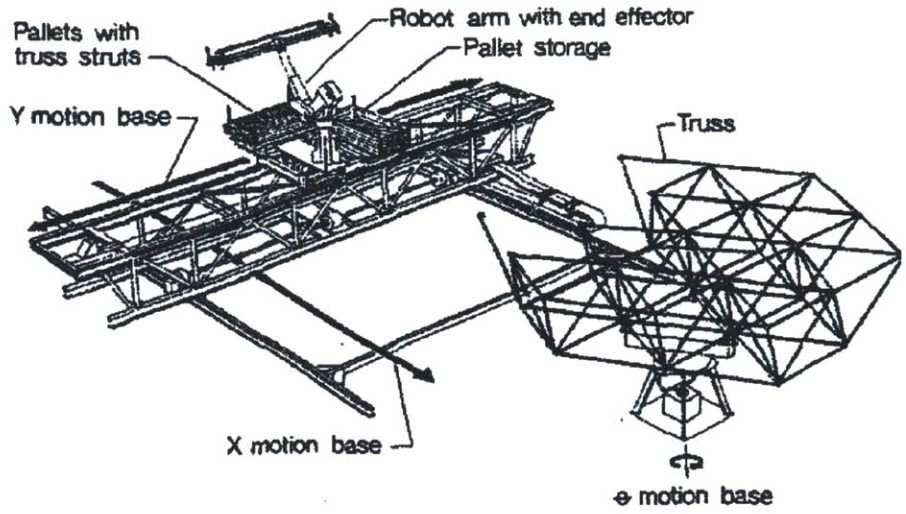


Figure 1-18: Drawing of the Automated Structure Assembly Laboratory [33]

The testbed there includes a robot used to construct the truss assembly. The robot has a 6-DOF industrial robotic arm that can hold up to 30 pounds. The testbed can be used to construct either planar or beam structures autonomously [33]. Research has also been conducted regarding different methods of assembling similar truss structures in space [34] [35].

In the NRL's Proximity Operations Test Facility, the FREND robotic arm, mentioned in Section 1.2.1, demonstrated the ability to successfully grapple a spacecraft that was not intended to be grappled [36]. The test facility enables two full satellites to operate in a 6-DOF environment in close proximity. It can also simulate orbital dynamics and thruster effects during rendezvous and docking. The facility includes an optics testbed that can simulate lighting conditions in space. The optics testbed is used for machine vision algorithms. The facility also includes a granite table that supports a test frame which is used to offset gravity [37]. It is pictured in Figure 1-19.

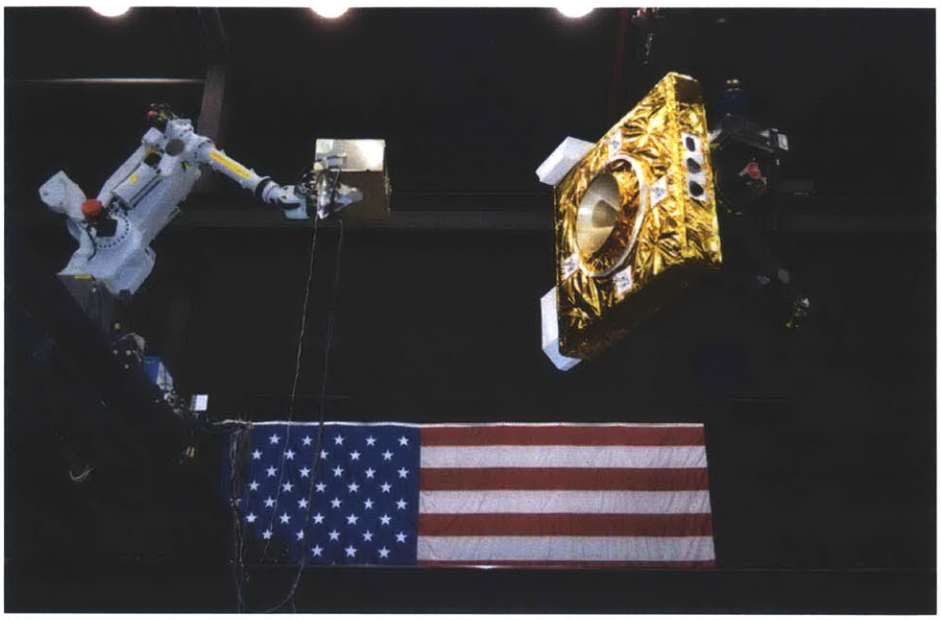


Figure 1-19: The NRL's Proximity Operations Test Facility [38]

The 7-DOF FREND arm was designed to be dexterous, able to position a tool tip within two millimeters and 0.4 degrees of the desired position and angle [36]. It is shown in Figure 1-20.

Figure 1-20: The FREND Robotic Arm [36]

The Land, Air, and Space Robotics Laboratory (LASR) at Texas A&M University also has the capability to test proximity operations and contact dynamics using the Holonomic Omni-directional Motion Emulation Robot (HOMER) and the Suspended Target Emulation Pendulum (STEP). In LASR, laser systems are used to determine position and attitude in inertial space. HOMER is pictured in Figure 1-21 [39].

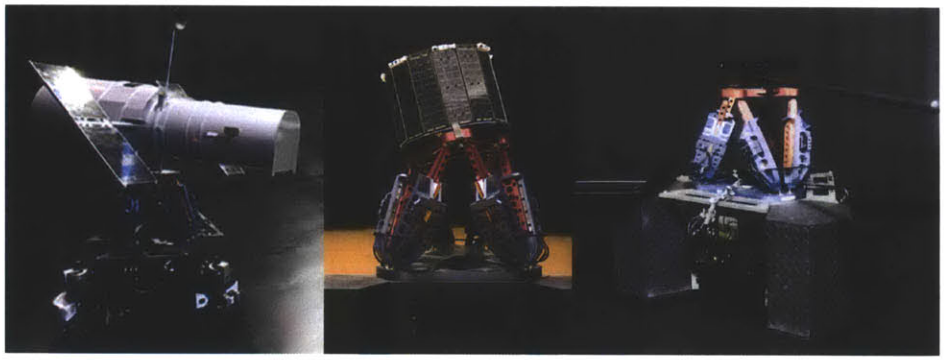


Figure 1-21: HOMER with Experimental Payloads [39]

HOMER simulates 6-DOF motion by translating and rotating on an air-puck system, with the supporting platform providing the remaining degrees of freedom. STEP is a 5-DOF suspended motion simulator, shown in Figure 1-22 [39].

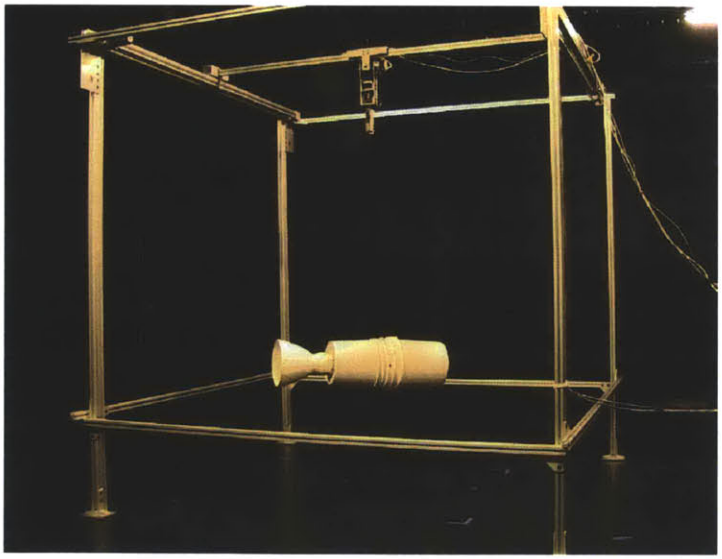


Figure 1-22: The STEP Setup [39]

Together, this equipment has been used to test contact dynamics of two spacecraft

docking [39].

Other air-bearing tables can also be found at the University of Victoria, Tokyo Institute of Technology, Honeywell Space Systems and Virginia Polytechnic Institute and State University, Carnegie-Mellon University and Texas Robotics and Automation Center, Georgia Institute of Technology, Carleton University, and others [26] [40]. There are also other robotic-arm-based and free-fall-based testbeds that were not mentioned in this section [41] [15].

1.2.3 Space-based Robotic Assembly and Servicing Testbeds

So far in history, there have been two primary types of space-based robotic assembly. The first type is one-time experiments which took place either in Low Earth Orbit (LEO) or inside of the Space Shuttle. The second type involves robotic manipulators attached to the ISS. Many of these missions were described in Section 1.2.1.

Experiments such as ROTEX and ETS-VII provided a valuable opportunity to research and develop technologies for robotic assembly and servicing in space. However, they are not testbeds, and thus are not conducive to iterative research.

The robotic manipulators attached to the ISS have generally been used for unsophisticated tasks and assembly of large components [42]. They are well-suited for such tasks, but most of the manipulators are not appropriate for executing the finer tasks that might occur on a robotic assembly or servicing mission. Dextre is one exception, as it was able to execute finer tasks during the Robotic Refueling Mission. NASA's Robonaut is another exception. Robonaut was launched to the ISS in 2011 and is now the first humanoid robot in space. It has been designed to have a similar form factor to astronauts so that it can use the same tools and operate alongside them. It is shown in Figure 1-23 [43].

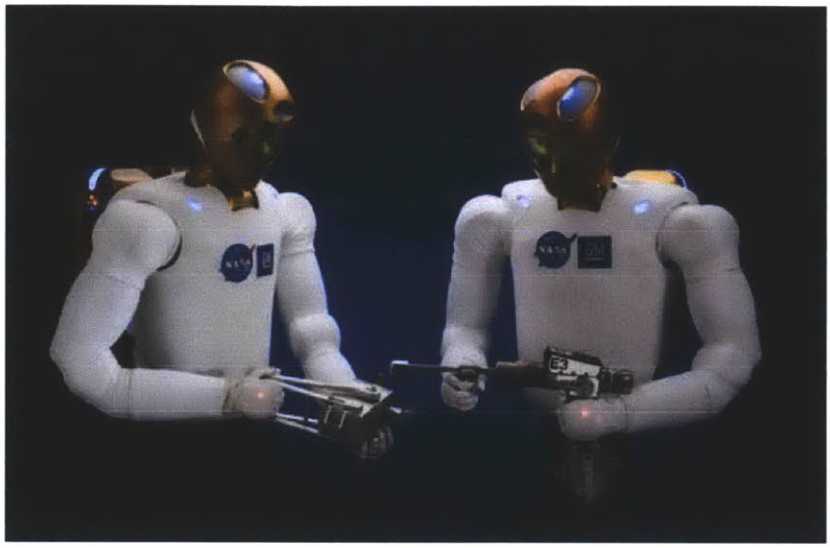


Figure 1-23: NASA's Robonaut [43]

Robonaut will also be upgraded with legs and a battery backpack in the future, and eventually will be operated outside the ISS alongside astronauts. Nonetheless, Robonaut still provides a valuable on-orbit testbed for robotic servicing technologies [43].

1.2.4 Literature Review Conclusions

Developers of the Deutsche Orbitale Servicing Mission and Orbital Life Extension Vehicle argue that simulation does not offer enough of a risk buy-down for a real, complex space mission that has never been attempted. These two missions, which involve rendezvous and docking, place demands on the spacecraft control system that reach new levels [13]. Therefore, rigorous test facilities are absolutely necessary.

While there are certainly many very capable testbeds on the ground, the ability to simulate the dynamics of the space environment by gravity compensation techniques is imperfect, leading to a gap between what can be tested on the ground and what actually occurs in a real mission. Suspension-based testbeds produce unacceptable disturbances on the attitude of suspended robots, and although parabolic flights on a Reduced Gravity Aircraft (RGA) are beneficial for their free-fall environment, they do not allow long enough testing durations to be realistic [26]. Underwater neutral

buoyancy testbeds typically involve robots that are fixed to a wall, rather than free-floating. And testbeds that float robots on a flat table or floor still succumb to the effects of gravity through friction and have limited degrees of freedom.

Thus, there is clear need for a facility that can:

- Support many different methods of on-orbit robotic assembly and servicing, including free-flyers
- Provide a risk-tolerant, dynamically authentic environment
- Support the testing of adaptive GNC technology through modularity and the ability to easily modify control algorithms
- Be cost effective

These requirements are described in detail in Section 3.1.

Some argue that ISS-based test facilities are the next clear step towards successful complex robotic assembly and servicing missions [44]. As the Venn Diagram in Figure 1-24 depicts, ISS-based test facilities, unlike any other current facility, can allow for safe and repeatable operations, can provide a dynamically authentic, 6-DOF testing environment, and can enable long testing durations.

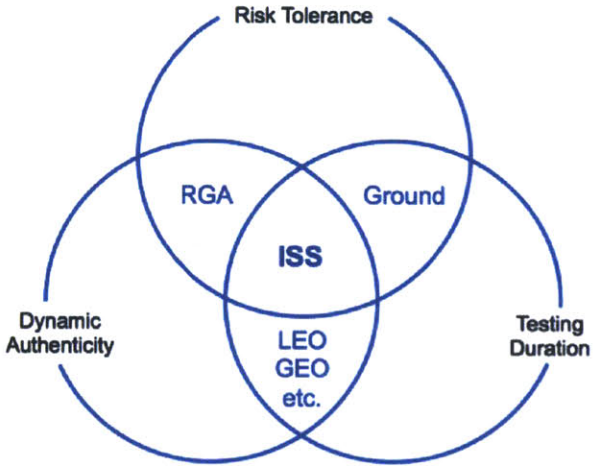


Figure 1-24: Venn Diagram of Important Testbed Characteristics [45]

Thus, ISS-based testbeds seem to be a strong candidate for a solution to this problem. One such testbed is the SPHERES Facility aboard the ISS, which is described in the following section. The development of the SPHERES Facility into a suitable testbed for robotic assembly and servicing is described in detail in Chapter Four.

1.3 SPHERES Facility Overview and Description

The SPHERES were developed by MIT's Space Systems Laboratory and launched to the ISS in 2006 [46]. The SPHERES are used for the development and testing of control algorithms in a low-risk environment, and in the past have been used to test formation flight, collision avoidance, and docking algorithms, among others.

In 2012, the SPHERES Facility on ISS was upgraded to include stereo-vision goggles. The goggles, which were sent to the ISS through the Visual Estimation for Relative Tracking and Inspection of Generic Objects (VERTIGO) program, a part of DARPA's InSPIRE-I contract, have already been used to test Vision-based Navigation (VBN) and Simultaneous Localization and Mapping (SLAM) algorithms.

In 2014, as a part of DARPA's InSPIRE-II program, docking ports are being added to the facility, which will enable the satellites to dock and undock autonomously. The docking ports are expected to launch in the late Fall of 2014.

Also as a part of the InSPIRE-II program, another piece of hardware known as the "Halo" is being sent to the ISS in 2014. The Halo is an interface which allows a SPHERES satellite to have multiple docking ports, goggles, or other cameras attached and running simultaneously.

The VERTIGO Goggles, Docking Ports and Halos will enhance the capabilities of the SPHERES Facility to enable the testing of different methods of robotic assembly and servicing.

1.3.1 SPHERES on the ISS

There are currently three SPHERES satellites aboard the ISS which operate inside a cubic test volume that is approximately two meters on each side. The satellites have

six degrees of freedom: three translational and three rotational. Each satellite has a set of 12 cold-gas thrusters (two per degree of freedom) and a single propellant tank with compressed CO_2. The thrusters are the SPHERES's only method of actuation. The satellites also each have infrared transmitters, which transmit an infrared pulse once per second to five beacons located around the outside of the test volume. Upon seeing the infrared flash, the beacons respond with ultrasound pulses that the satellites can detect with ultrasound sensors. Along with gyroscopes and accelerometers, the satellites use the ultrasound sensors to triangulate their positions and calculate accurate state estimates. A SPHERES satellite is pictured in Figure 1-25.

Figure 1-25: SPHERES Satellite [47]

A labeled SPHERES satellite without its outer shell is shown in Figure 1-26.

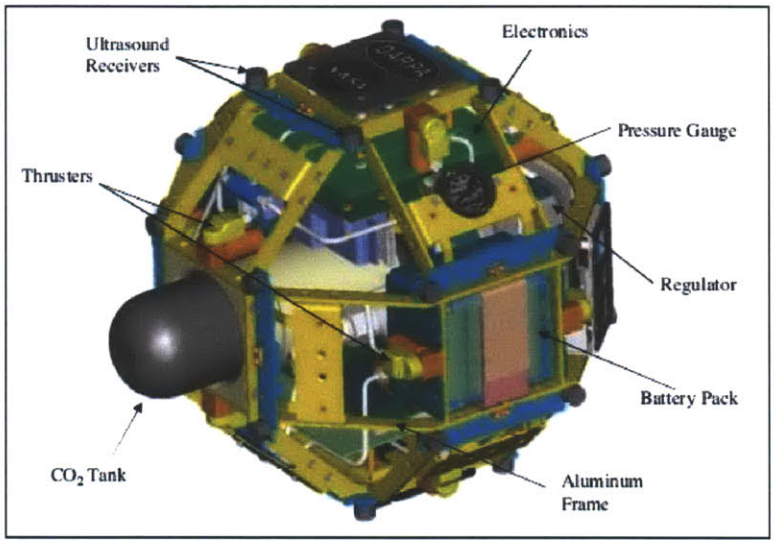


Figure 1-26: Labeled SPHERES Satellite [47]

The purpose of the SPHERES controller is to take in commanded and estimated states and output the forces and torques that should be applied to the system for that time step. The SPHERES satellites use a 13 element state vector (three for position, three for velocity, four for the quaternion, and three for angular rates) and are run on a one Hertz control cycle, meaning the calculation of new forces and torques occurs once per second. The SPHERES testbed was set up to facilitate exploration of various control algorithms, and the controllers are easily modifiable. On the ISS, the SPHERES satellites are in a 6-DOF environment; however, on the ground, the SPHERES satellites operate in a 3-DOF environment: one rotational, two translational. The ground controller controls the SPHERES satellites in these three degrees of freedom only, but it can be expanded to control motion in 6-DOF on the ISS, which will help demonstrate servicing or assembly mission capability.

1.3.2 SPHERES Ground Testing

The Flat Floor Facility is located in MIT's Space Systems Laboratory. The facility is part of the SPHERES ground testbed, which allows for algorithm testing in a 3-DOF environment. The epoxy floor is in the shape of an octagon and is five meters across. The large size allows for the testing of multiple individual components, as well as

large structures. The facility is pictured in Figure 1-27 with several SPHERES and a test structure on top of it.

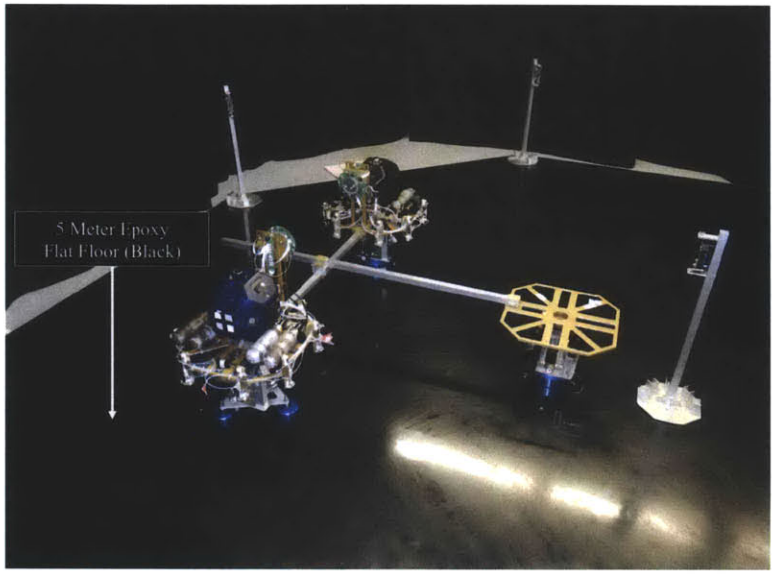


Figure 1-27: The Flat Floor Facility at MIT

Much of the testing documented in this thesis occurred in the Flat Floor Facility.

The Self-assembling Wireless Autonomous Reconfigurable Modules (SWARM) are also key components used alongside the SPHERES satellites on the ground. The SWARM Propulsion Units, pictured in Figure 1-28, each hold a satellite and float on the flat floor using an air puck system, which blows air down on onto the flat floor. The units add an additional 16 thrusters to the capabilities of the SPHERES satellites, enabling them to move much more rapidly across the floor and to overcome friction. The units also have a docking port so that two SWARM units can dock together.

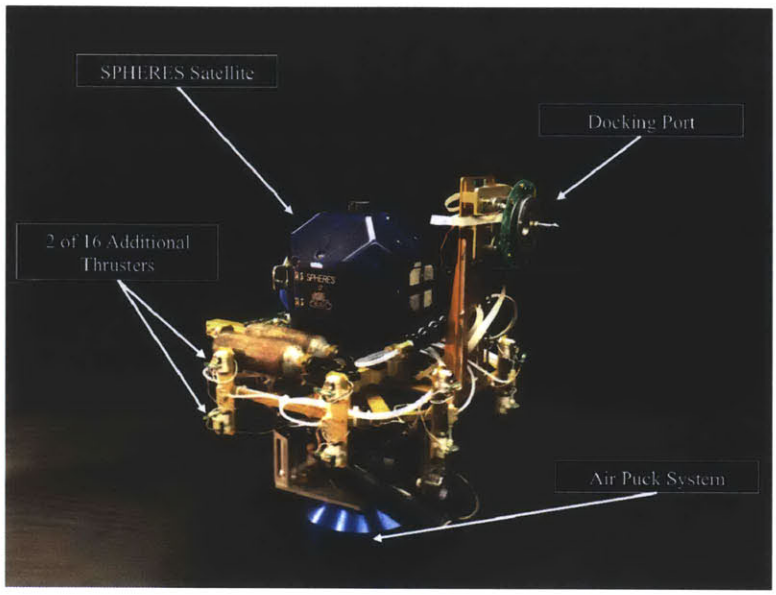


Figure 1-28: SWARM Unit on the Flat Floor

1.4 Thesis Overview

A majority of the work included in this thesis involves the use or development of the SPHERES Facility with robotic assembly and servicing in mind. The remaining four chapters focus on the initial ground testing for several related control concepts, the development of requirements for a suitable space-based testbed that can enable robotic assembly and servicing testing, and the hardware development which expands the SPHERES Facility into such a testbed. The thesis is organized as follows:

- Chapter 2 explores two possible control methods that could be used for robotic assembly and servicing missions, presents the results of initial ground testing of those concepts, and ultimately concludes that fully proving those concepts requires a testbed in space.

- Chapter 3 describes the process of developing requirements for that space-based robotic assembly and servicing testbed, as well as the requirements imposed on upgrades to the SPHERES Facility to make it such a testbed.

- Chapter 4 presents the development of a hardware augmentation to the SPHERES Facility. The new hardware enables the testing of robotic assembly and servicing technologies. This chapter is the main contribution of this thesis and discusses the progression of the hardware from an idea all the way through a functional prototype.
- Chapter 5 highlights the conclusions of the thesis.

Chapter 2

Ground Testing

Control systems play a crucial role in robotic assembly and servicing in space. Whether there is only one moving entity at a time or multiple, maintaining full controllability of a system is a necessity in the space environment. Several concepts have surfaced at the forefront of control system development with regards to robotic assembly, to include reconfigurable control and adaptive control. This chapter explains these two concepts and describes some ground testing that has specifically tested each of them. Ultimately, this chapter concludes that developing a robotic assembly testbed in space is a necessary next step for the evaluation and advancement of these concepts.

2.1 Reconfigurable Control

2.1.1 Hypothesis

Upon docking with another mass, a satellite's mass and inertia properties are guaranteed to change. Thus, that satellite must be able change its controller in order to maintain acceptable control performance. Reconfigurable control proposes that by having two independently controlled systems exchanging mass properties, the aggregated system can modify its controller in real-time to effectively control the system. The concept of Resource Aggregated Reconfigurable Control (RARC) was first developed and tested by MIT's Space Systems Laboratory in the winter of 2012-2013 by

Chris Jewison, David Sternberg, and the author, alongside Professor David Miller as a part of the Agile Reconfigurable Modules with Autonomous Docking for Assembly and Servicing (ARMADAS) project [48]. Portions of their paper are presented below. Resource aggregation is paired with reconfigurable control in the RARC concept so that satellites can also combine their sensor and actuator capabilities properly. The business card model allows satellites to easily and quickly exchange all necessary information upon docking, and that concept was implemented in the experiment described in the following section.

2.1.2 Experiment

The RARC concept was first tested using the SPHERES Facility at MIT, which was described in Section 1.3. Reconfigurable control is accomplished on the SPHERES testbed by modifying the standard SPHERES controller in real time. The standard controller is tuned to effectively control a single SPHERES satellite on a propulsion carriage. However, when two of these modules dock and become a single rigid body, the mass and inertia properties of the system change, necessitating a modification of the controller. This controller reconfiguration is conducted each time a docking or undocking event occurs.

Controller Development

The base controller used for the SPHERES satellites is a simple Proportional-Derivative (PD) controller. The controller receives the commanded and estimated states and differences them to determine the state error. These errors are multiplied by the appropriate gains (K_p and K_d, proportional and derivative respectively) to determine the necessary forces and torques according to the following control law:

$$F_x = K_{p,position} * e_x + K_{d,position} * e_{\dot{x}} \tag{2.1}$$

$$F_y = K_{p,position} * e_y + K_{d,position} * e_{\dot{y}} \tag{2.2}$$

$$T_z = K_{p,attitude} * e_\theta + K_{d,attitude} * e_{\dot{\theta}} \tag{2.3}$$

where F_x and F_y are the controlled forces in the x and y directions, T_z is the controlled torque, e_x and e_y are the errors in x and y position, $e_{\dot{x}}$ and $e_{\dot{y}}$ are the errors in the x and y velocities, and e_θ and e_ω are the errors in the attitude and rate. The proportional and derivative gains are calculated by first modeling the system in state space. The poles of the system are selected based on desired damping ratios and natural frequencies. For the proportional gains, $K_{p,position}$ and $K_{d,position}$, the selected damping ratio is 0.75 and the natural frequency is 0.2 Hertz. For the attitude gains, $K_{p,attitude}$ and $K_{d,attitude}$, the damping ratio is set at 0.75 with a natural frequency of 0.4 Hertz. The poles are placed at these desired locations according to the method of state feedback, which yields the following equations for the gains:

$$K_{p,position} = \omega_n^2 * m \tag{2.4}$$

$$K_{d,position} = 2 * \zeta * \omega_n * m \tag{2.5}$$

$$K_{p,attitude} = \omega_n^2 * I \tag{2.6}$$

$$K_{d,attitude} = 2 * \zeta * \omega_n * I \tag{2.7}$$

It is important to note that while these gains depend on desired damping ratios and natural frequencies, they also depend on the mass and inertia of the system. As the mass and inertia change due to the docking and undocking of modules, the gains change to maintain the desired pole locations. In the SPHERES example, gains are recalculated whenever a docking or undocking maneuver is complete. A challenge arises in reconfiguring controllers when multiple controlled elements are docked. In order to prevent multiple controllers from separately sending possibly conflicting commands, all satellites run the control algorithm, but actuation is performed according to the identification number of each satellite. For example, when two satellites dock, a single satellite is designated as the master satellite, and the other is designated as the subordinate satellite. Both satellites immediately reconfigure their controllers upon

docking and both run the newly reconfigured controller. Consequently, the controller outputs the required thruster firing times for both satellites. Actuation, however, is specified by the identification of each satellite: the master satellite only fires the thrusters that the controller specifies for the master satellite, and the subordinate satellite only fires the thrusters that the controller specifies for the subordinate satellite. This control-actuation arrangement therefore requires both satellites to run the reconfigured controller but only to act upon half of the resulting required actuations. This process is easily expandable for larger satellite systems.

Test Scenario

The test scenario was created to mimic a servicing scenario in space, where a servicing satellite approaches a target satellite and docks to it, they maneuver together (e.g., change orbits), and then they undock. The scenario involves two SWARM units which were initially placed on opposite sides of the Flat Floor described in Section 1.3. The orange SPHERES satellite was designated as the "servicer," and the blue SPHERES satellite was designated as the "target." The setup is pictured in Figure 2-1.

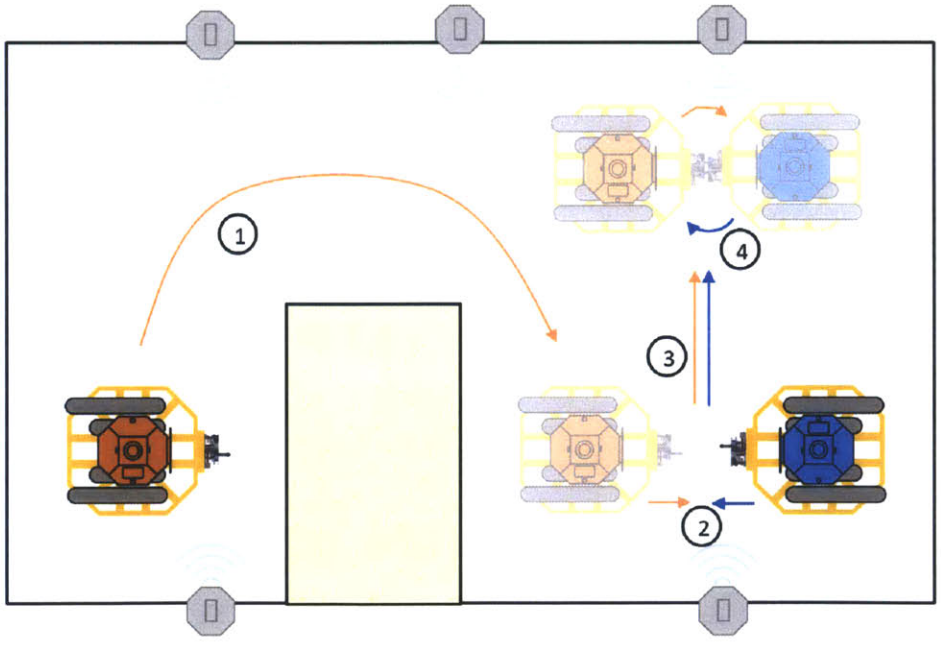


Figure 2-1: Bird's Eye View of the Test Area with 1) Rendezvous, 2) Docking, 3) Translation, and 4) Rotation [48]

Step 1 involved a maneuver where the servicer approached the target satellite using a risk-allocative path-planning algorithm to avoid an obstacle and reach the rendezvous location. In Step 2, the two units docked together: the target simply held its position, and the servicer aligned the docking ports and engaged them. The RARC maneuvers comprise Steps 3 and 4. The satellites first execute a one meter translation maneuver followed by a 180 degree rotation maneuver before undocking.

The satellites each used their own control system for Steps 1 and 2. The controllers were identical and were each designed to give a single SWARM unit the desired performance characteristics. Between Steps 2 and 3, however, the control system reconfigured, and in Steps 3 and 4 the aggregate system used the new controller.

2.1.3 Results

Several key metrics used to determine the success of the reconfigurable controller include rise time, settling time, and percent overshoot for the step input responses in both the translational and rotational cases. For the translational case, Figure 2-2 shows a comparison of the expected step input responses for the system with adapted gains and non-adapted gains.

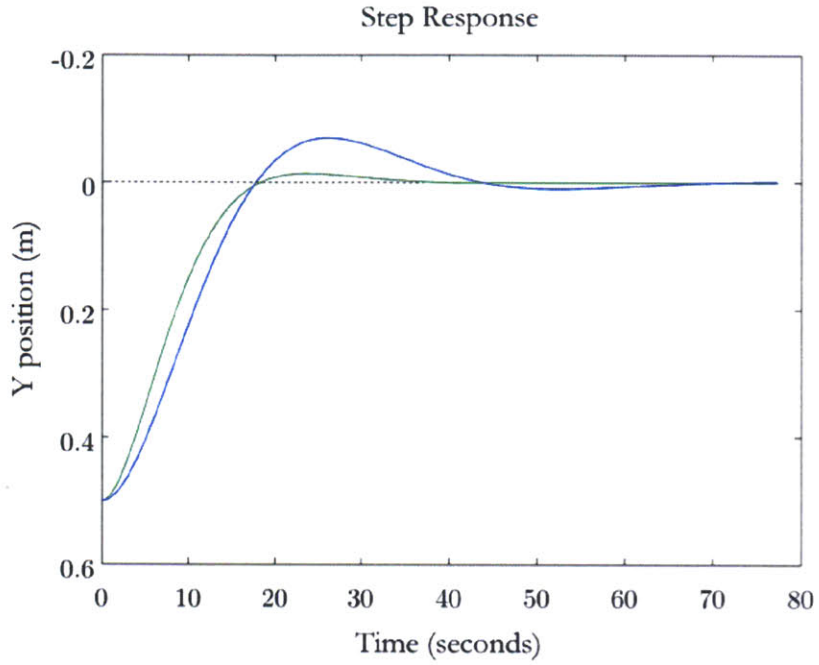


Figure 2-2: Responses to 0.5 meter Step Input for the System with Adapted (green line) and Non-adapted (blue line) Gains [48]

The performance of the system was expected to improve by adapting the gains when the two 17.8 kilogram SWARM units dock. The response with the non-adapted controller has a rise time of 11.7 seconds, a percent overshoot of 14%, and a settling time of 41.1 seconds. With appropriately adapted gains, the response has a rise time of 11.3 seconds, a percent overshoot of 2.8%, and a settling time of 28.9 seconds. In reality, the system did not perform as well as expected in all areas. The response of the aggregate system with adapted gains to a step input of 0.5 meters is shown in Figure 2-3.

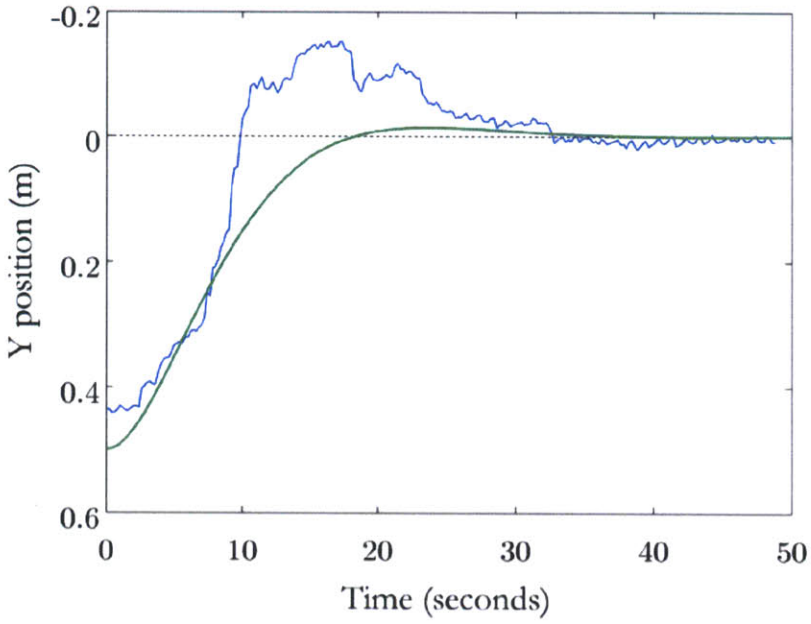


Figure 2-3: Comparison of Expected (green) and Actual (blue) Responses to 0.5 meter Step Input [48]

Figure 2-3 shows a response with a rise time of 5.9 seconds, percent overshoot of 35.1%, and settling time of 41.2 seconds. The system had a very quick rise time, but then overshot significantly and took a while to recover to a steady-state at the commanded position. There may be several reasons for this response. First, the tanks on the air carriages may have been refilled immediately before the test, causing there to be less friction between the pucks and the Flat Floor than standard conditions. The tanks on the SWARM propulsion carriage may have also been refilled immediately before the test, causing the thrusters to exert larger forces upon firing than expected. However, these two causes are not probable, as the overshoot was seen in multiple tests. Another potential source of error is inaccurate sensor measurements due to the setup of the beacons and the location within the test area. Finally, the fact that the rotational maneuver began before the translational response completely settled out likely increased the overshoot and settling time. These issues could be mitigated in the future by rearranging the beacons and allowing more time between maneuvers during the test.

For the rotational case, the expected step responses to inputs of 180 degrees for

the system with adapted and non-adapted gains are shown in Figure 2-4.

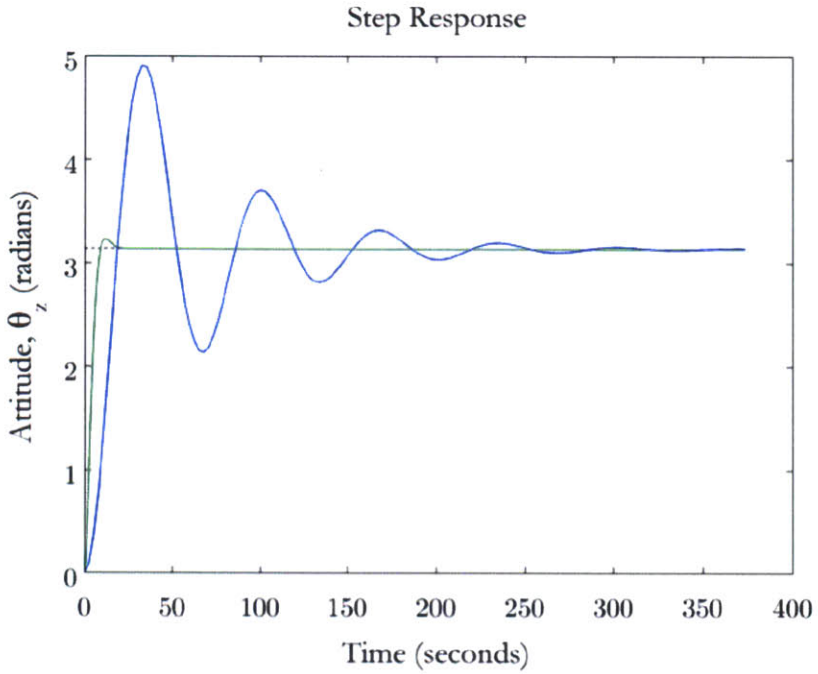


Figure 2-4: Responses to 180-degree Step Input for the System with Adapted (green) and Non-adapted (blue) Gains [48]

Figure 2-4 highlights the significant performance improvements expected due to adapting the gains upon docking. Because each SWARM unit has a mass of 17.8 kilograms, inertia of 0.16 kilogram-meters^2, and a distance to the UDP from the center of mass of 0.265 meters, the inertia increases to 2.82 kilogram-meters^2, making adaptive gains critical to the successful performance of the system. The response with the non-adapted controller has a rise time of 12.5 seconds, a percent overshoot of 56.4%, and a settling time of 211 seconds. In contrast, the system with appropriately adjusted gains has a rise time of 5.7 seconds, a percent overshoot of 2.9%, and a settling time of 14.4 seconds. When implemented on the hardware, the aggregate system with adapted gains produced the response to a step input of 180 degrees shown in Figure 2-5.

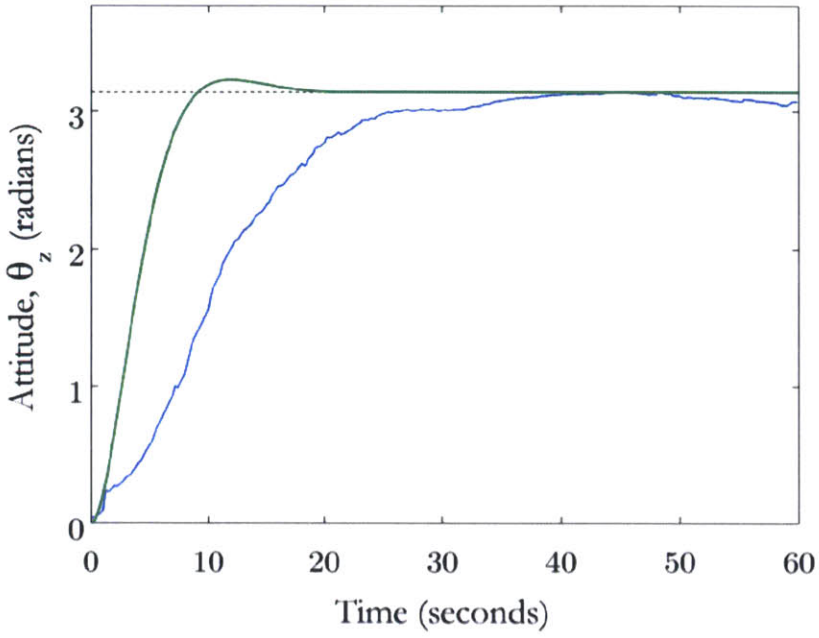


Figure 2-5: Comparison of Expected (green) and Actual (blue) Responses to 180-degree Step Input [48]

The system had a response with a rise time of 16.75 seconds, a percent overshoot of zero percent, and a settling time of 35.5 seconds. Friction on the surface of the Flat Floor facility is likely the cause of the response being slightly slower than expected and marginally underdamped. However, the response is certainly desirable. The performance without adapted gains is expected to be much worse, although that scenario was not tested during this experiment. The plots in Figure 2-6 show the reliability of these results over multiple tests.

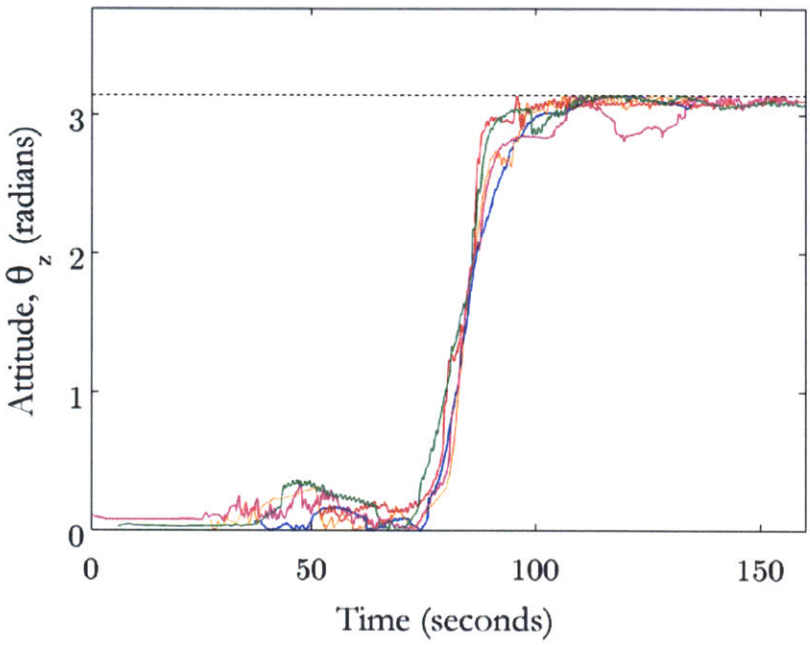


Figure 2-6: Multiple Responses to 180-degree Step Input [48]

2.1.4 Analysis

These results show that there is strong potential for the methods of reconfigurable control to be effective when docking two modules together which can each self-identify its mass properties and communicate with the other. However, further testing is certainly needed to prove the concept. The results lead one to believe that with more testing and tuning of the controllers, the ground results can be vastly improved.

2.2 Direct Adaptive Control

2.2.1 Hypothesis

Direct adaptive control is different from reconfigurable control because controller gains are continuously being updated based on control errors, rather than being modified based on changing mass properties. A direct adaptive control law is theoretically robust to mass and inertia property changes in a system—even unknown or unexpected ones. Similar to the way that a human can pick up objects of different masses and

tune the amount of force exerted on the object to pick it up in a controlled manner, direct adaptive control changes the gains of the system in real-time to meet performance requirements. Thus, unknown parameters do not significantly degrade the performance of the system. In the Spring of 2013, MIT's Space Systems Laboratory decided to apply these concepts to a system of distributed satellites that were rigidly attached to one another, proposing that the satellites could successfully rate-dampen the system without knowing the system's mass properties, inertia properties, or initial tip-off rate ahead of time.

2.2.2 Experiment

The team conducted an experiment to test this hypothesis using the SPHERES satellites, SWARM units, and Flat Floor facility. Chris Jewison, David Sternberg, and the author created and executed the experiment. Sections 2.2.2 and 2.2.2 describe the development of the controller used in the testing and the test scenario and setup.

Controller Development

In direct adaptive control, the forces and torques commanded can be calculated according to the same control law as that of reconfigurable control, given in Equations 2.1, 2.2, and 2.3 of Section 2.1.2. However, there are two main differences: input shaping and time-varying gains. Input shaping involves feeding the commanded position or attitude into a reference model before differencing it with the state, as shown in Figure 2-7.

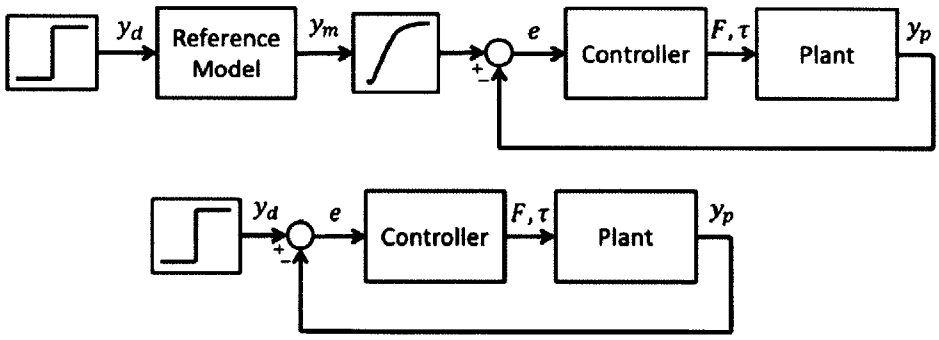


Figure 2-7: Control Loop with Input Shaping (top) vs. Standard Control Loop (bottom) [48]

The reference model is designed to produce an output with the desired properties, such as settling time and overshoot. This method yields much smaller error terms and is less likely to saturate the actuators because the controller tracks the smooth reference output rather than the sharp step input.

The gains used in direct adaptive control for translational motion, shown in Equations 2.8 and 2.9, also vary with time as a function of the error, unlike those of reconfigurable control given in Equations 2.4 and 2.5, which only vary based on mass [49] [48].

$$K_{p,position} = K_1 * e_{position} + K_2 * e_{position} \tag{2.8}$$

$$K_{d,position} = K_3 * e_{velocity} + K_4 * e_{velocity} \tag{2.9}$$

where the proportional parts of the control gains are given by the following equations:

$$K_1 = e^T_{position} * \Gamma_1 \tag{2.10}$$

$$K_3 = e^T_{velocity} * \Gamma_3 \tag{2.11}$$

and the integral parts of the control gains are given by the following equations:

$$\dot{K}_2 = e^T_{position} * \Gamma_2 - \sigma_a * K_2 \tag{2.12}$$

$$\dot{K}_4 = e^T_{velocity} * \Gamma_4 - \sigma_b * K_4 \tag{2.13}$$

In Equations 2.12 and 2.13, the σ terms are used to prevent the integrated control gains from diverging, and the Γ terms are tuning parameters[49]. Note that these equations allow for the gains to be adjusted according to the error in tracking the reference output regardless of the mass of the object being moved, making this controller more robust to mass uncertainties.

Initial simulation results show similar responses for the PD controller and the direct adaptive controller for a given mass, as seen in Figures 2-8 and 2-9. However, the tracking error is significantly smaller for the direct adaptive controller case.

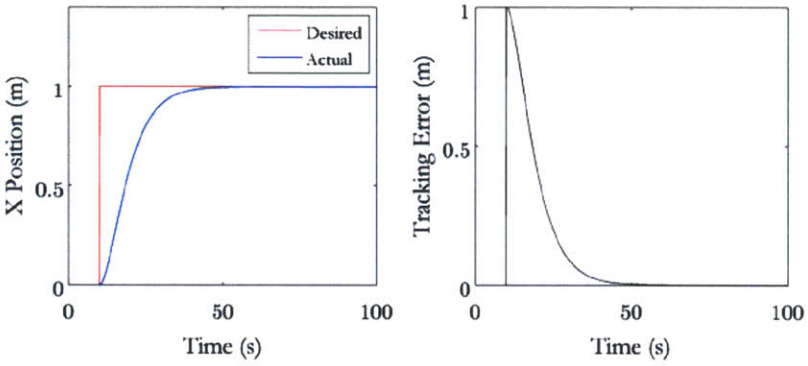


Figure 2-8: Time-response and Tracking Error to Step Input for the PD Controller [48]

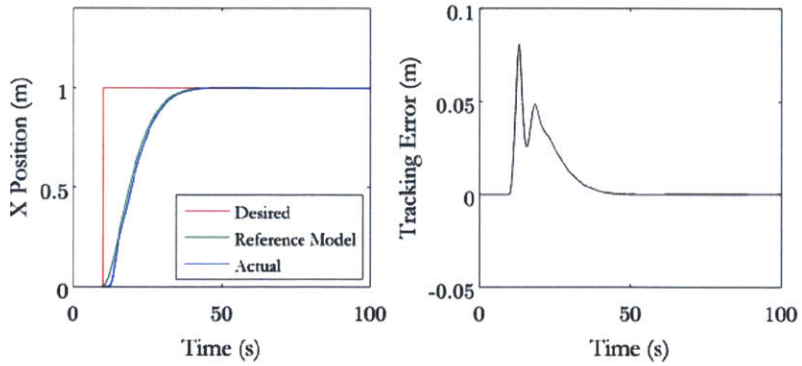


Figure 2-9: Time-response and Tracking Error to Step Input for the Direct Adaptive Controller [48]

When the mass of the system is doubled, as in the case of docking, the performance of the two controllers differs significantly. Figure 2-10 shows the overshoot present in

the response with the PD controller, as well as the consistently high tracking error.

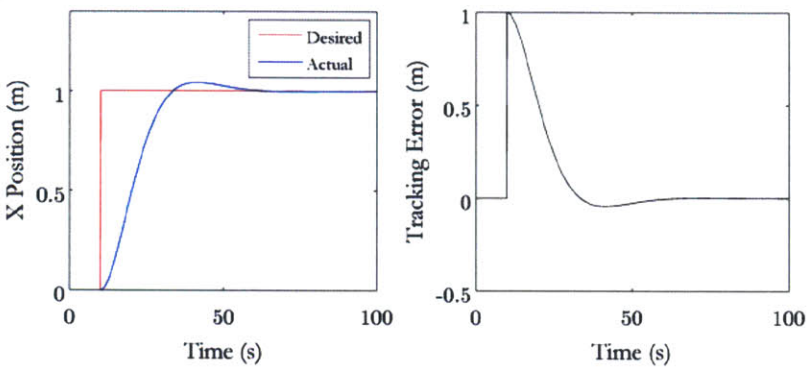


Figure 2-10: Time-response and Tracking Error to Step Input for the PD Controller in the Two-satellite Case [48]

With the system mass doubled, the response of the system with the direct adaptive controller maintains performance, as shown in Figure 2-11.

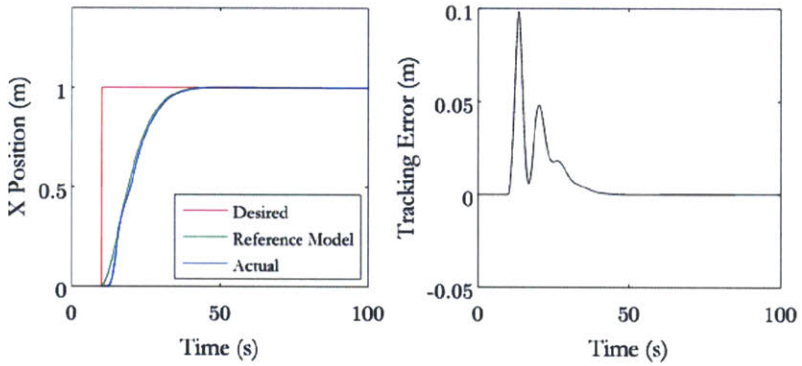


Figure 2-11: Time-response and Tracking Error to Step Input for the Direct Adaptive Controller [48]

The plot in Figure 2-11 is key: it shows how the performance is not significantly degraded and the tracking error remains small. Thus, the direct adaptive controller is robust to unknown mass changes and is less prone to saturating the actuators. While these simulation results only apply to translational motion, similar results were expected for the rotational motion case.

For the rotational motion case, the control law given in Equation 2.14 was used

[49].

$$T_z = K_x * x_m + K_e * e_x \tag{2.14}$$

where x_m is the reference model state and e_x is the error between the state of the plant and the reference model. The gains are given in Equations 2.15 and 2.16.

$$K_x = K_{x,P} + K_{x,I} \tag{2.15}$$

$$K_e = K_{e,P} + K_{e,I} \tag{2.16}$$

where the proportional terms are given by the following equations:

$$K_{x,P} = e_x * x_m * \Gamma_5 \tag{2.17}$$

$$K_{e,P} = e_x^2 * \Gamma_6 \tag{2.18}$$

and the integral terms are given by the following equations:

$$K_{x,I} = e_x * x_m * \Gamma_7 \tag{2.19}$$

$$K_{e,I} = e_x^2 * \Gamma_8 - \sigma_c * K_{e,I} \tag{2.20}$$

Test Scenario

The test scenario was set up to mimic the concept behind DARPA's Phoenix project, described in Section 1.2.1, in which a servicing satellite would rendezvous with a retired communications satellite in the GEO graveyard orbit, chop off its antenna, and attach "satlets" to the antenna in order to control it and communicate with the ground. These satlets each contain a single function (e.g., propulsion) and combine with one another to attain equivalent functionality to a full satellite. In the scenario, two thruster satlets are attached to a communications antenna that has an initial, unknown rotation rate. The satlets must detect the rotation rate and, working together, use their thrusters to dampen out the rotation. The satlets have no knowledge of the

initial rotation rate before the test. They also have no knowledge of the system mass or inertia properties. For the purposes of this test, two SWARM units represented the satlets and were attached to each other by a rigid bar, as shown in Figure 2-12. A second rigid bar connected a proof mass to the system.

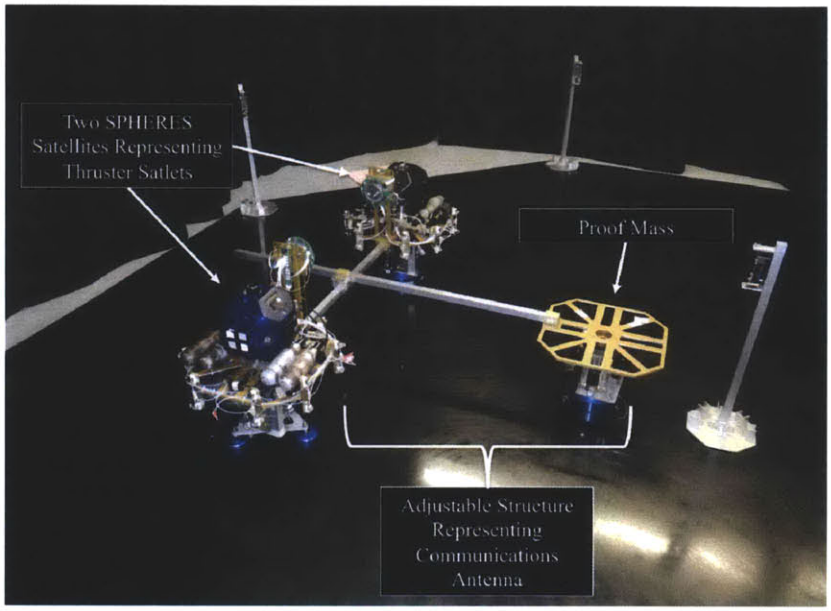
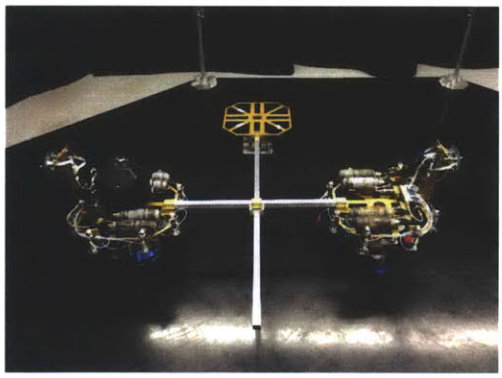
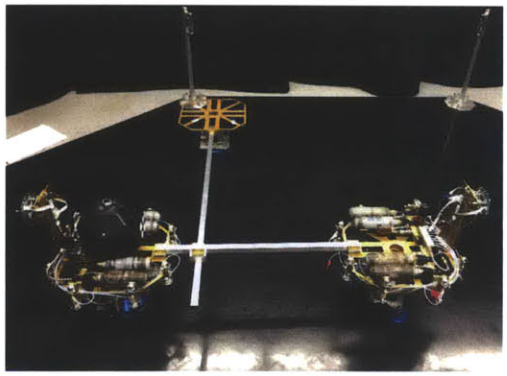


Figure 2-12: Adaptive Control Test Setup in the MIT Flat Floor Facility

The second rigid bar could be shifted to change the inertia properties of the system, as shown in Figure 2-13



(a) Centered Configuration



(b) Asymmetric Configuration

Figure 2-13: Modifiable Adaptive Control Test Structure in Two Configurations

Note that the SPHERES were only using gyroscope measurements for attitude

determination and control. Thus, any translation of the system was uncontrolled.

There were four different test configurations. The first was called "Centered Slow," which means that the test structure was in the centered configuration, as pictured in Figure 2-13a, and the initial tipoff rate was slow. For all of the tipoff rates, one of the team members simply spun the entire assembly as appropriate. It did not matter that these spin-ups were not exact, because the satellites were measuring their initial spin rate using gyroscopes. Thus, for "slow" tipoff rates, the team member spun the assembly at a rate that varied between 10 to 20 degrees per second. The second test configuration was called "Centered Fast." In this configuration, the test structure was also in the centered configuration, but the initial tipoff rate was anywhere between 30 and 45 degrees per second. The third configuration was called "Asymmetric Slow," meaning the test structure was in the asymmetric configuration, as shown in Figure 2-13b. And finally the fourth configuration was called "Asymmetric Fast."

For each of the four test configurations, tests were conducted both with and without the direct adaptive controller running, and the total rotation of the system was measured. This was done in order that the rotation with the controller running could be compared to the rotation without the controller running, demonstrating that friction on the Flat Floor was not the only reason for the dampening.

2.2.3 Results

The results from the experiment are shown in Table 2.1 below. These results are taken from the MIT Project Phoenix Testing Summary [50].

Table 2.1: Initial Rates and Total Rotations for Each Test Configuration [50]

Test Configuration		Initial Rate (deg/sec)	Total Rotation (deg)
1 Centered Slow	1a. No Control	16*	460*
	1b. DAC	12.1	79.1
2 Centered Fast	2a. No Control	45*	900+*
	2b. DAC	38.0	257.0
3 Asymmetric Slow	3a. No Control	22.3	570*
	3b. DAC	14.3	204.0
4 Asymmetric Fast	4a. No Control	36*	720+*
	4b. DAC	33.5	287.7

* The corresponding SPHERES data file was broken, preventing a value from being calculated; these values were obtained from video analysis.

By comparing the total rotation in the "No Control" cases (i.e., 1a, 2a, 3a, and 4a), with the total rotation in the Direct Adaptive Control (DAC) cases (i.e., 1b, 2b, 3b, and 4b), one can see that the uncontrolled system rotated significantly further than the controlled system in every configuration, suggesting that friction did not play a substantial role in damping the control systems. Note that for 2a and 4a, the system had to be stopped before colliding with the side of the Flat Floor facility, which is why the "+" symbol is written. This table also shows that while the asymmetric cases did perform slightly worse than the centered cases, the control system was able to successfully adapt to the different inertia properties. As expected, the system rate-damped the slow cases more quickly than the fast cases.

Using the same naming convention found in Table 2.1, Figures 2-14, 2-15, 2-16, and 2-17 show the z-axis rotation rate for the four direct adaptive control test configurations. In each, the initial angular rate corresponds to the initial rate listed in Table 2.1, and the SPHERES data and reference model used in the controller are plotted. As can be seen in Figure 2-14, the system was able to track the reference model well with the direct adaptive controller. Error between the plant and the

reference model is due to the aggressive nature of the model, which aimed to damp the system within 25 seconds regardless of the initial tipoff rate or symmetry of the system. Slowing down the reference model (increasing the time constant) would allow the system to track it more closely.

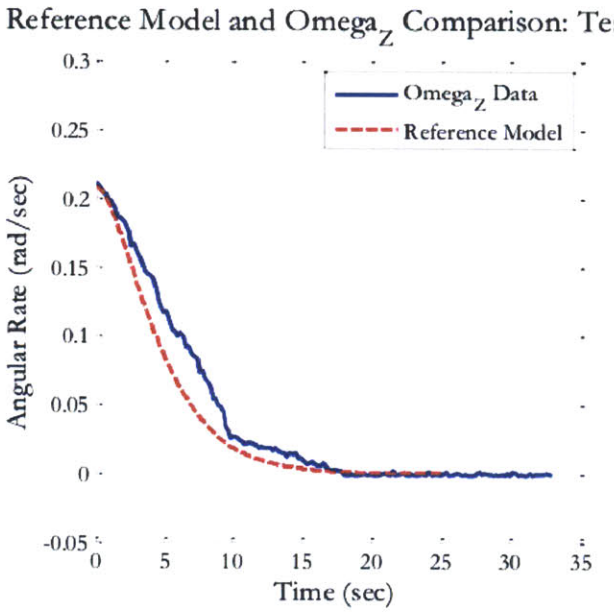


Figure 2-14: Reference Model and Z-Axis Rotation Rate: Test 1b [50]

Figure 2-15 shows the control system working well even in the asymmetric configuration, although the error between how the system rotated and the reference model is larger, which was expected.

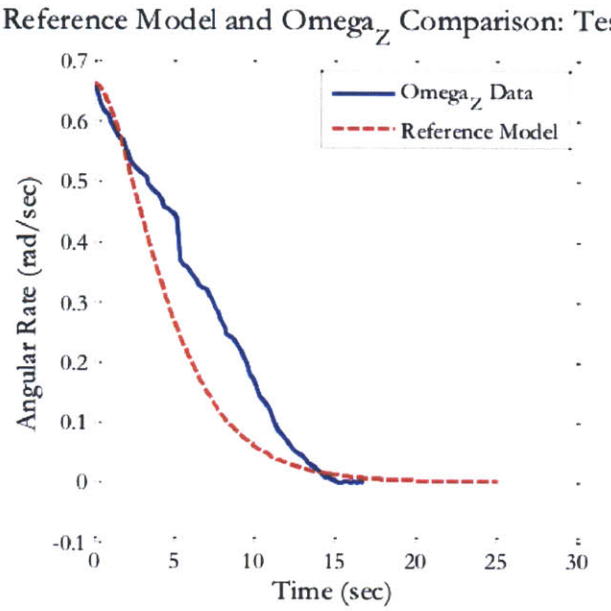


Figure 2-15: Reference Model and Z-Axis Rotation Rate: Test 2b [50]

Figures 2-16 and 2-17 show that, even for the higher rotation rate cases, the controller was able to dampen out the system's spin. Future work on the control system includes adjusting the reference model and further tuning the direct adaptive control parameters.

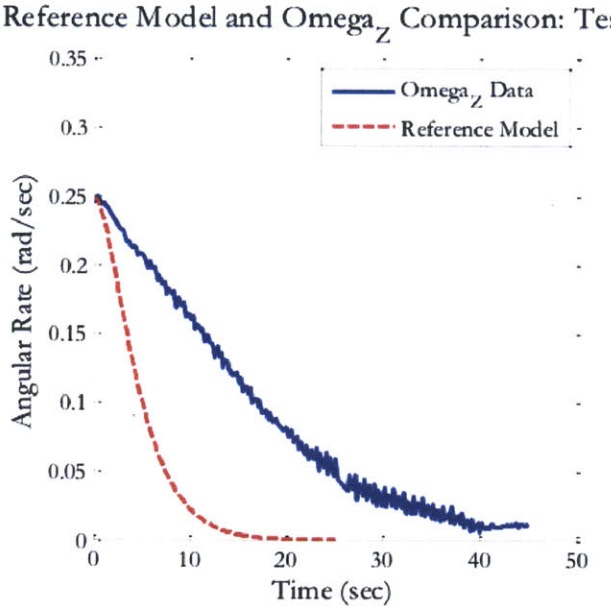


Figure 2-16: Reference Model and Z-Axis Rotation Rate: Test 3b [50]

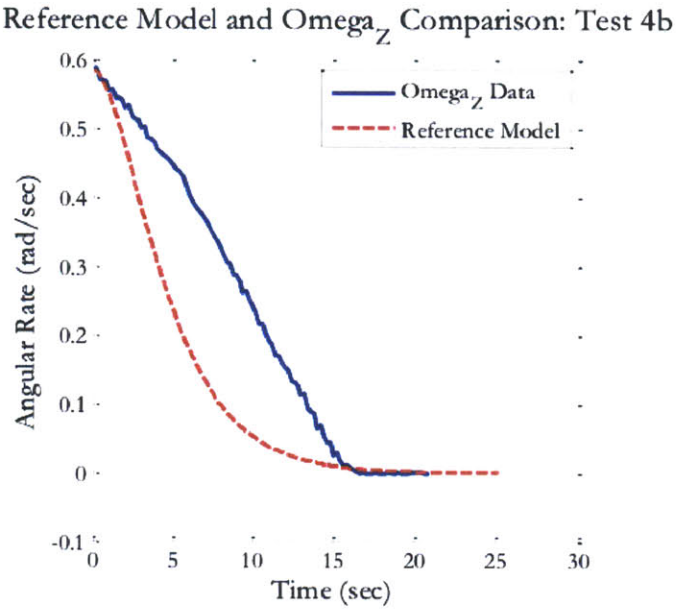


Figure 2-17: Reference Model and Z-Axis Rotation Rate: Test 4b [50]

2.2.4 Analysis

These results do indicate that direct adaptive control systems have the ability to control a system of unknown mass, inertia properties, and initial rate. However, much more testing needs to be done in order to prove the concept. In addition, although the data suggest that friction did not completely nullify the experiment, it still was a factor. The amount of air in each of the CO_2 tanks likely affected the results. Repeated ground testing would lend credibility to the data. Also, tests where the system's initial rate is zero and it is given a commanded rotation would be valuable, because friction may play a smaller role in the results than it did in the rate damping tests (since friction is a damping force).

2.3 Conclusions from Ground Testing

With regards to the SPHERES Facility, there is no escaping the errors induced in the test by friction and differing CO_2 tank levels. Also, due to the nature of a 3-DOF testbed on the ground, only one axis of attitude control can be tested. The dynamics

involved when performing full 3-axis attitude control are much more complex, but can only be tested in an environment where there are 3 degrees of rotational freedom. All of these limitations of ground testbeds serve to increase the demand for a space-based testbed, where testing can occur in true micro-gravity, with fewer sources of error, and in a 6-DOF, dynamically authentic environment.

Chapter 3

Development of Requirements

Chapters 1 and 2 concluded that a space-based testbed is a necessity for the advancement of robotic assembly and servicing technologies. Members of the MIT Space Systems Laboratory, alongside NASA, are seeking to develop such a testbed through DARPA's InSPIRE-II program, described in Section 1.3. This chapter lays out the objective statement for the InSPIRE-II program as a whole and describes how that objective statement flows down into testbed requirements, system-level requirements, and subsystem-level requirements. Chris Jewison and the author developed the Halo System Requirements document, which is paraphrased below [45]. The requirements were developed alongside Chris Jewison with inputs from the the MEDUSA team in Professor David Miller's 16.851 class. Team members included Chris Jewison, Dustin Hayhurst, Pete Davison, Brandon Karlow, Jared Rize, and the author.

3.1 Testbed Requirements

The objective statement for the InSPIRE-II program is to:

Develop a cost-effective facility for maturing adaptive guidance, navigation, and control (GNC) technology in support of on-orbit, robotic satellite assembly and servicing in a risk-tolerant, dynamically authentic environment

This objective statement can be broken into four separate sub-objectives:

- Facility shall support robotic satellite assembly and servicing
- Facility shall provide a risk-tolerant, dynamically-authentic environment
- Facility shall support testing of adaptive GNC technology
- Facility shall be cost effective

Each of these sub-objectives is described in more detail in the following subsections.

3.1.1 Sub-objective #1 and Related Requirements

The first sub-objective is as follows:

1.0 [Obj] Facility shall support robotic satellite assembly and servicing
Facility shall enable testing of as many mission architectures as possible, which span the following design dimensions:
1.1 Utilizing both integrated and external servicing/assembly functions
1.2 Based on both distributed and centralized servincing/assembly functions
1.3 Involving both fully captured and proximity operation concepts

These requirements are based on a framework for understanding the various methods of robotic assembly and servicing in space. The framework was developed at MIT's Space Systems Laboratory, and it divides the concepts into eight different architectures for robotic assembly and servicing. Professor David Miller, Chris Jewison, David Sternberg, and the author were on the team that created the architectures and drew up the figures.

The different methods of robotic assembly and servicing can be thought of as spanning three orthogonal dimensions. Those dimensions are depicted in Figure 3-1. Although each axis is a spectrum, this analysis considers only the extremes at either end of each axis, yielding eight distinct architectures.

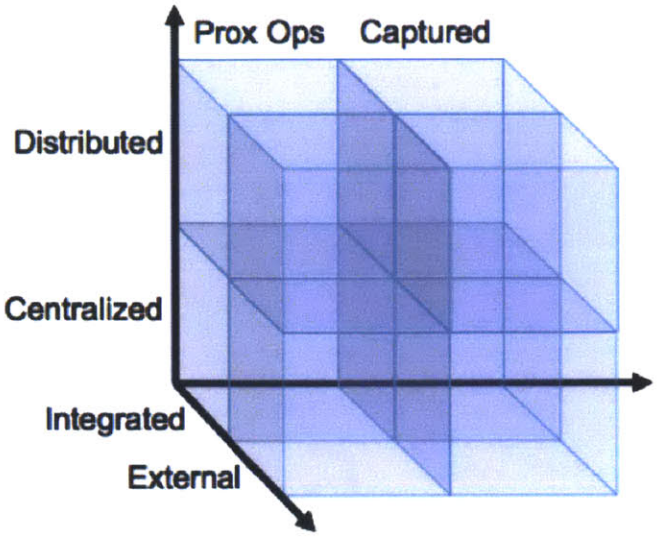


Figure 3-1: Categories of On-orbit Robotic Assembly and Servicing Architectures [45]

The first axis designates a distinction between "Integrated" and "External." In an Integrated architecture, all of the robotic servicing and assembly capability is entirely contained within the system. In other words, all of the functionality necessary for a spacecraft to assemble or service itself is native to the spacecraft and remains with the spacecraft for its entire lifetime. External architectures, on the other hand, are architectures in which an external vehicle brings all of the assembly or servicing functionality to a given spacecraft for a robotic assembly or servicing mission. Such a vehicle also departs the spacecraft after executing its mission and may move on to assemble or service another spacecraft.

The second axis designates a distinction between "Centralized" and "Distributed" architectures. Centralized architectures are those in which there is only a single servicing or assembly entity. All of the functionality for assembling and serving is contained within this entity. Distributed is just the opposite—multiple entities may be servicing or assembling a spacecraft a the same time in such an architecture.

The final axis designates a distinction between "Proximity Operations" and "Fully Captured Operations." In an architecture involving Proximity Operations, each of the servicing or assembly entities is independently controlled and maneuvered in close

proximity to the spacecraft being assembled or serviced. This may be done through electromagnetic formation flight, thrusters, or some other type of propulsion system. Fully Captured Operations are those in which all entities are physically attached to each other at all times during the process of assembly or servicing—there is never a free-flying entity throughout the operation. These entities may use a method of locomotion such as robotic arms to move around the spacecraft.

Thus, there are eight architectures in the tradespace. These architectures are laid out in Table 3.1 and given nicknames to make them easier to picture.

Table 3.1: Eight On-orbit Robotic Assembly and Servicing Architectures

Architecture Number	1	2	3	4	5	6	7	8
Description	Integrated Distributed Prox Ops	Integrated Distributed Captured	Integrated Centralized Prox Ops	Integrated Centralized Captured	External Distributed Prox Ops	External Distributed Captured	External Centralized Prox Ops	External Centralized Captured
Nickname	"Beehive"	"Integrated Construction Crew"	"Integrated Tug"	"Spider"	"Carrier"	"External Construction Crew"	"External Tug"	"Shuttle"

The first architecture is called the "Beehive," drawing to mind a picture of bees flying around and constructing a beehive. This architecture is integrated, meaning all of the servicing and assembly functionality is inherent in the spacecraft itself, like a beehive. It is distributed, meaning there are multiple entities servicing or assembling rather than a single actor. And it involves proximity operations, meaning at some point, the entities are physically separated from one another and independently controlled. A sample Concept of Operations (CONOPS) is shown in Figure 3-2.

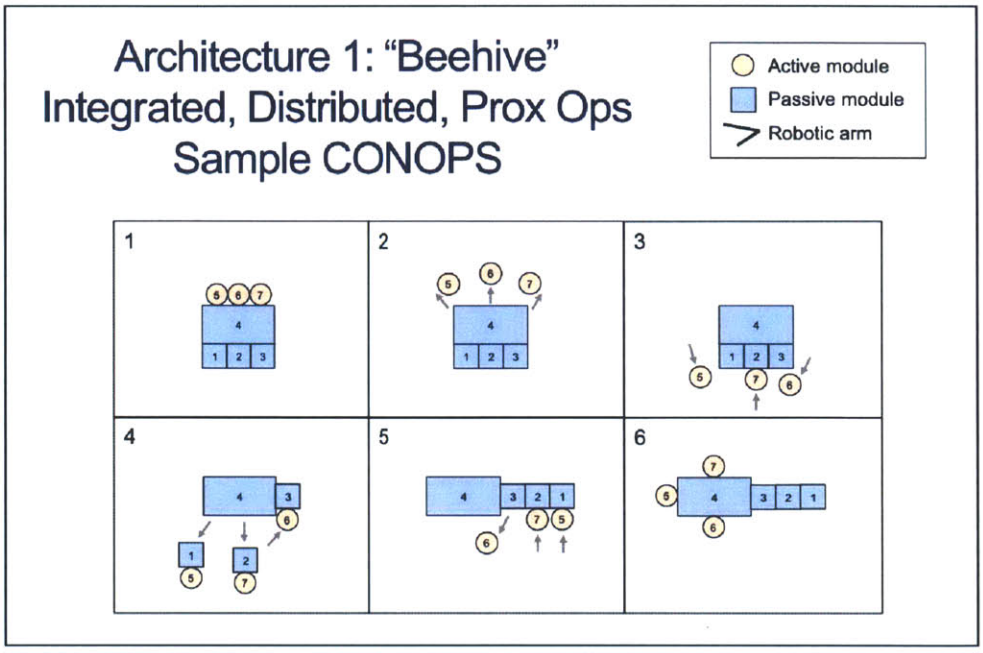


Figure 3-2: On-orbit Robotic Assembly Architecture 1

In all of the figures in this section, there are six panels depicting the CONOPS. The orange circles are active modules, meaning they actively maneuver during the assembly or servicing process. The blue squares are passive modules, meaning they simply hold their position during the entire process or get manipulated by an active module. Dark black lines signify robotic arms, although none are pictured in this particular CONOPS.

In Figure 3-2, the active modules start out as part of the system. They then separate in panel two and fly around the passive modules using proximity operations. In panels three and four, they attach to the smaller passive modules and remove them from the central module. In panels five and six they use the smaller modules to assemble a structure on the side of the main module before reattaching themselves to the spacecraft until further servicing is required. Note that this is a simple CONOPS used to depict this specific architecture, and a much more complex CONOPS could be conceived using the same architecture.

The second architecture is called the "Integrated Construction Crew," which makes one think of a construction crew which lives on-site and climbs around the

structure to build it. The architecture is integrated and distributed, like the first, but this time is captured, meaning there are never any free-flying entities; all entities are attached to the rest of the system at all times. Figure 3-3 shows a reference CONOPS for this architecture.

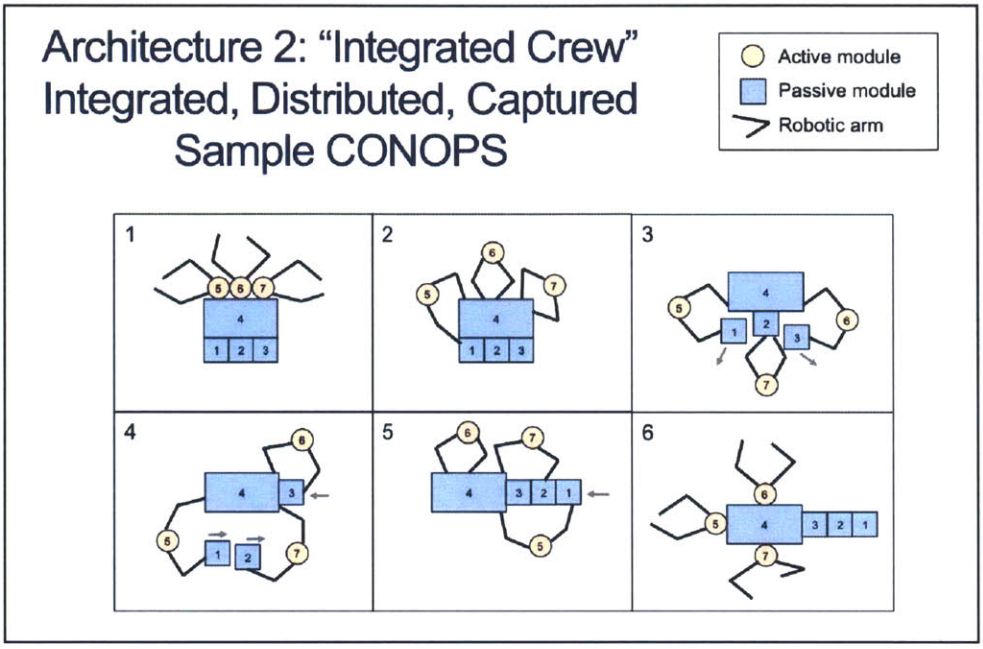


Figure 3-3: On-orbit Robotic Assembly Architecture 2

In panel one, multiple members of the robotic construction crew begin integrated with the system. When it is time to begin assembling, the entities climb around the outside of the main module, shown in panel two. In panels three and four, they detach the smaller passive modules and begin assembling them into a structure before returning to their stowed locations in panel six. An example of a mission using this architecture from Section 1.2.2 is the Skyworker robot.

The third architecture is known as the "Integrated Tug," meaning that a "tug-boat" type of assembler and servicer is integrated into the system. However, there is only one tug, and thus the architecture is centralized, as the single tug is the only entity responsible for all assembly and servicing actions. The architecture also involves proximity operations, and a CONOPS is given in Figure 3-4.

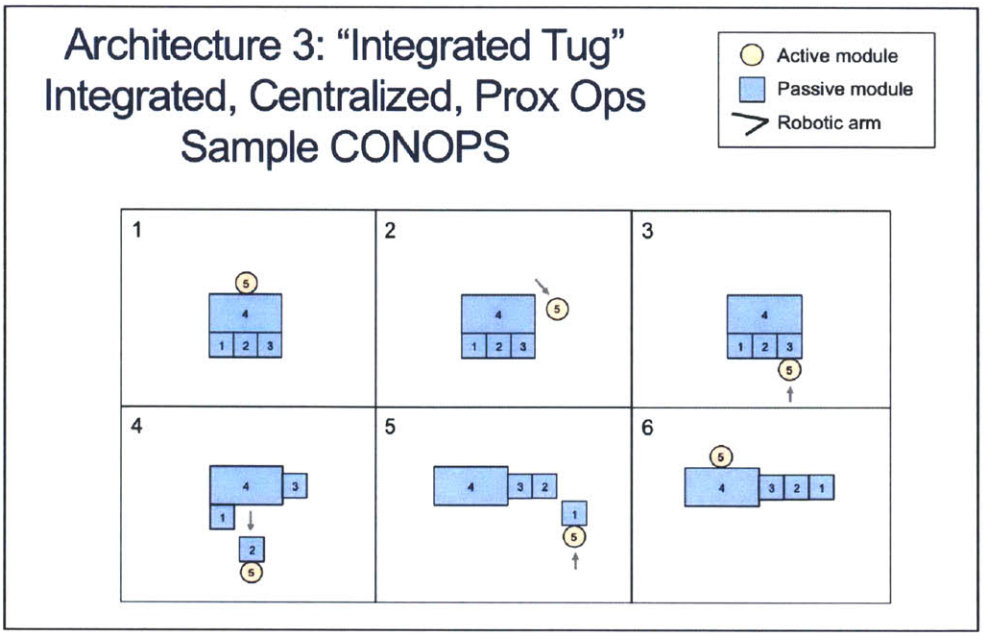


Figure 3-4: On-orbit Robotic Assembly Architecture 3

In Figure 3-4, the tug flies independently around the rest of the system grabbing one smaller passive module at a time and placing it into its new position in the overall structure before returning to its stowed location to await future tasks. Another project with a CONOPS that used this architecture was ARMADAS, which was described in Section 2.1.

The fourth architecture, called "Spider," brings to mind the image of a spider crawling around its own web and assembling and servicing it. This architecture is very similar to the previous one, having an integrated and centralized servicer and assembler, with the exception that this architecture is captured. The CONOPS is shown in Figure 3-5.

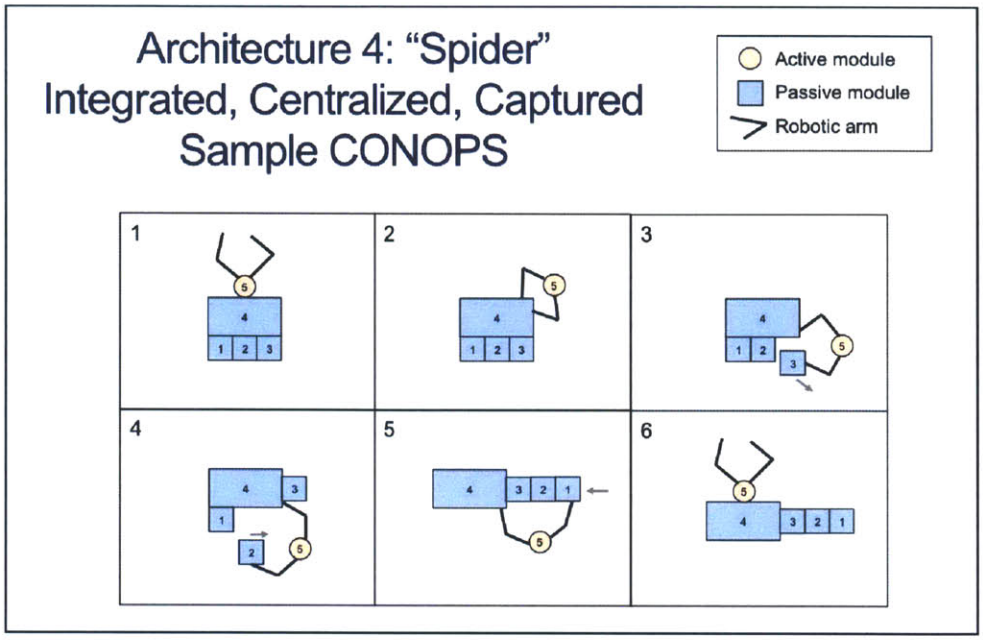


Figure 3-5: On-orbit Robotic Assembly Architecture 4

Here the single servicer/assembler crawls around the system rearranging passive modules into a new structure. A real-world example of a mission using this architecture is the Space Station Remote Manipulator System with its ability to climb around the outside of the ISS structure, assembling it and servicing it.

In the fifth architecture, known as "Carrier" like a naval carrier ship, an external assembly and servicing system joins a separate spacecraft to assemble or service it, and may even bring components that will add to the original system, such as extra fuel, stronger reaction wheels, more solar panels, or more structure. Architecture five is also distributed and involves proximity operations. It is depicted in Figure 3-6.

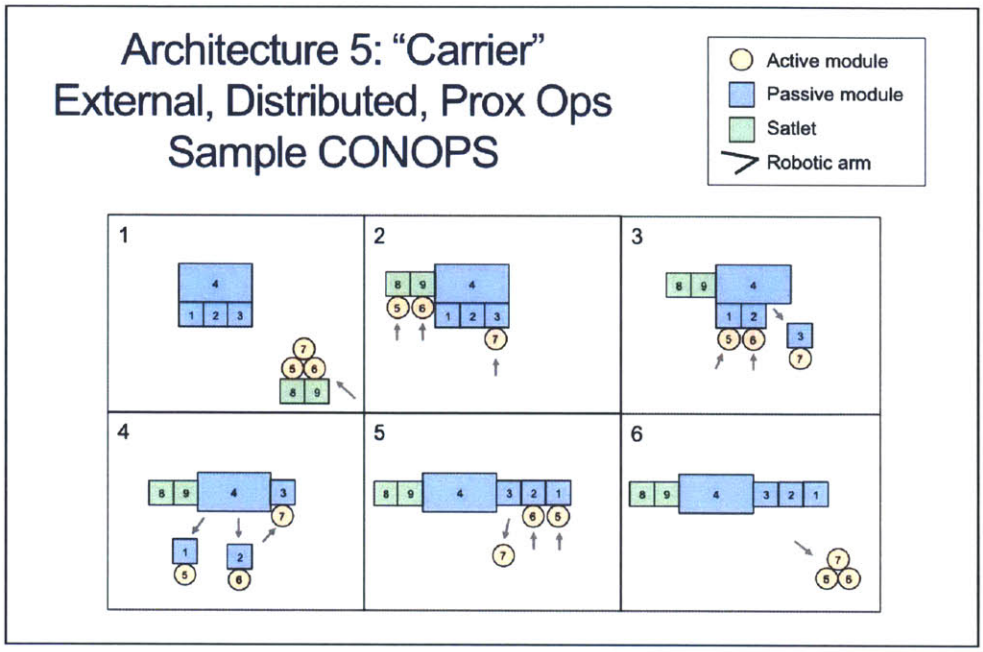


Figure 3-6: On-orbit Robotic Assembly Architecture 5

In Figure 3-6 and in all subsequent CONOPS, the green squares represent satlets—items that the external servicer brings that become a permanent part of the original system. In panel one, the external assembler/servicer must rendezvous with the target satellite. In panel two, it drops off the new satlets that it is bringing before going to transport passive modules to a different place on the spacecraft in panels three, four, and five. Finally, in panel six, the external servicer leaves the satellite and moves on, possibly to service or assemble another spacecraft.

Figure 3-7 shows the sixth architecture, known as "External Construction Crew." Similar to the Internal Construction Crew, the assembly and servicing capabilities are distributed and all entities are captured during the entire process. However, the "construction crew" is external, and must rendezvous with the target spacecraft to begin the assembly and servicing process.

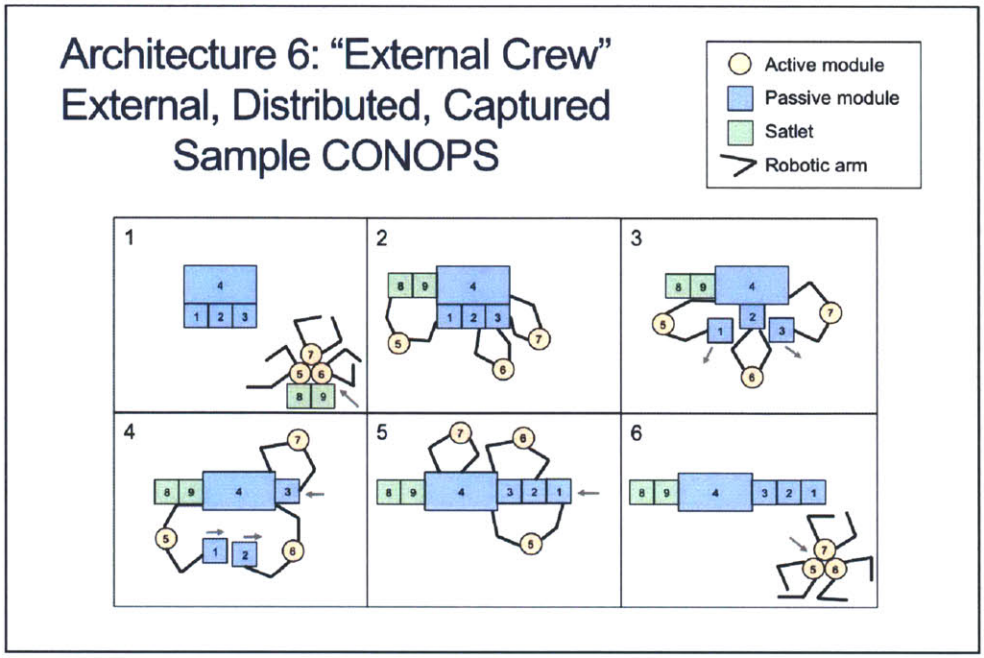


Figure 3-7: On-orbit Robotic Assembly Architecture 6

Similar to Figure 3-6, Figure 3-7 shows an external assembler/servicer bringing satlets to a target satellite, attaching them to the spacecraft, reconfiguring the spacecraft, and departing to assemble or service another satellite.

The seventh architecture is called "External Tug." This architecture has a single external assembling and servicing entity that performs tug-like maneuvers to aid a target satellite, similar to architecture 3. A sample CONOPS is given in Figure 3-8.

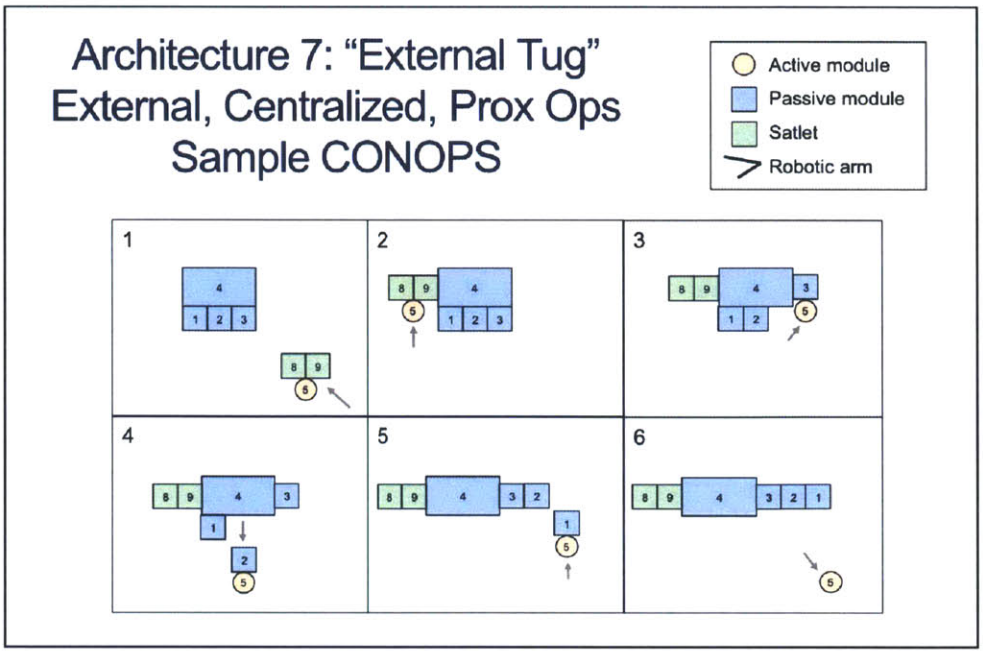


Figure 3-8: On-orbit Robotic Assembly Architecture 7

This figure depicts an external servicer bringing satlets to a target satellite, attaching them, transporting passive modules to a different location on the target satellite, and departing.

The eighth and final architecture is called "Shuttle" because famous missions that used this architecture are the Hubble Servicing Missions, described in 1.2.1, which involved the Space Shuttle and its robotic manipulator. This architecture utilizes an external, centralized servicer and all operations are captured. A CONOPS is shown in Figure 3-9.

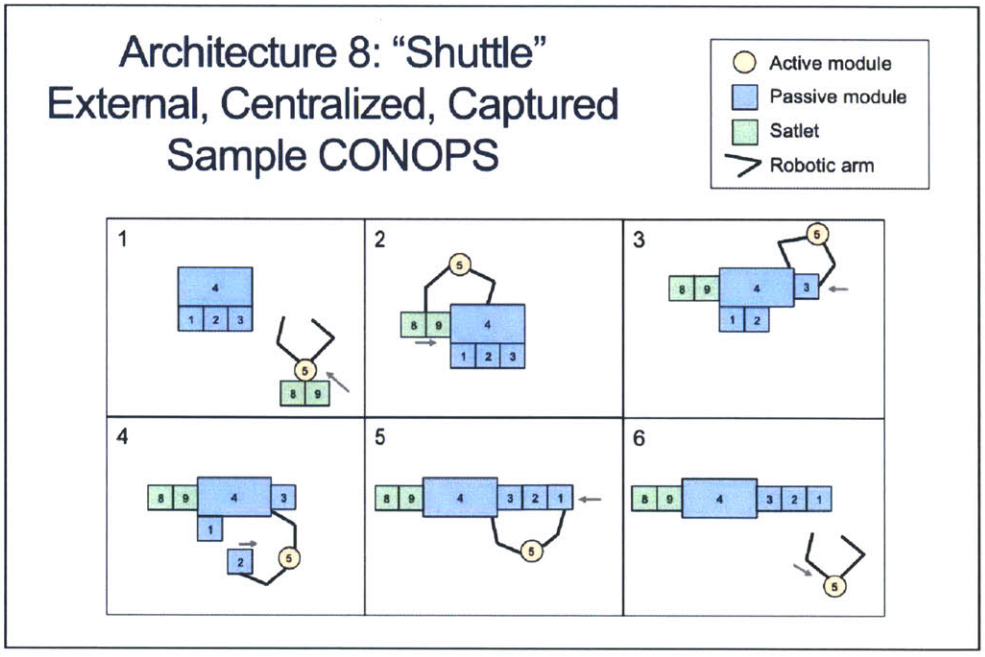


Figure 3-9: On-orbit Robotic Assembly Architecture 8

This final CONOPS shows a single external servicing and assembly entity that attaches to the target satellite, deposits several satlets, manipulates multiple passive modules, and departs for another mission. The Phoenix mission, described in Section 1.2.1, gives another CONOPS that falls within this architecture.

The objective of the testbed is to enable testing of as many of these architectures as possible.

3.1.2 Sub-objective #2 and Related Requirements

The second sub-objective statement is as follows:

2.0 [Obj] Facility shall provide a risk-tolerant, dynamically authentic environment

2.1 Individual experimental failures shall not present a safety hazard according to defined NASA/ISS regulations or damage the facility in any way

2.2 Facility shall enable iterative research with repeatable test conditions

2.3 Facility shall provide resources to diagnose the cause of a technology failure

2.4 Facility shall enable vehicle dynamics representative of close-proximity on-orbit robotic assembly and servicing missions

In order to be considered risk-tolerant, a facility must meet a number of requirements. One of them is that the facility is safe—operations within the facility should not endanger any operator of the facility or the facility itself. NASA has a set of safety regulations for equipment aboard the ISS that all components of the facility must adhere to. The facility should also enable iterative research and provide adequate resources to an experimenter for data collection. This means that an experimenter must be able to repeat tests with similar initial conditions and achieve similar results. An experimenter must also be able to collect data for each test and diagnose any issues before developing a new experiment. Such a setup is highly conducive to advancing research as it allows scientists to iteratively build upon successes and correct failures. On the ISS, astronauts have the ability to simply reset a test that is going wrong, making such a testing environment highly risk-tolerant.

There are six common areas that typically define an authentic space environment: radiation, thermal, vacuum, orbital dynamics, field of regard, and microgravity. The first three conditions can be tested on the ground with relatively high fidelity, and there are numerous testbeds for that sort of development. Orbital dynamics, such as rendezvous, is a field which has been studied and tested through a number of space missions already, and thus is not the most pressing area for research. And while field of regard—a characteristic that refers to the high-contrast lighting situation in space—is another challenging aspect of the space environment to replicate, testbeds on the ground such as those at the NRL and Lockheed Martin, described in Section 1.2.2, are spending money to advance technology in those areas. The highest-risk aspect of robotic assembly and servicing missions in space is microgravity, especially including proximity operations. Microgravity may also be the most difficult aspect of the space environment to simulate on the ground. Therefore, the testbed must be a microgravity facility.

3.1.3 Sub-objective #3 and Related Requirements

The third sub-objective statement is as follows:

3.0 [Obj] Facility shall support testing of adaptive GNC technology

All of the following shall be modifiable on-orbit:

3.1 Algorithms

3.2 Models

3.3 Available sensors, actuators, and plant components

3.4 CONOPS

A truly versatile and valuable testbed also enables the modification of algorithms, models, available sensors, available actuators, available plant components, and CONOPS throughout the lifetime of the test facility. A facility's user gains the ability to test a broad spectrum of control schemes and methodologies when each of the parts of the control loop are modifiable. Such a system allows a scientist to test every area of robotic assembly and servicing, from reconfiguring controllers to gaining or losing resources such as sensors or thrusters. Experimenters can also devise a number of concepts for robotic assembly and servicing and test each of them on the same testbed.

3.1.4 Sub-objective #4 and Related Requirements

The final sub-objective statement is as follows:

4.0 [Obj] Facility shall be cost-effective

4.1 Facility shall make maximum use of existing hardware, software, and procedures

4.2 Facility shall be extensible

4.3 Facility shall be multi-functional

The driving force behind the development of testbeds in the first place is the cost savings. Because real-world space missions are so expensive, investors want

assurance that the mission will be a success. They choose to invest in testing in order to lower risk so that the mission will have a lower chance of failure, which would be the least-effective financially. Thus, testbeds must be cost-effective. One of the primary means for making a facility cost-effective is utilizing an existing facility, if an appropriate one exists. Testbeds that are extensible also add value, as they can be further developed down the road to increase capabilities at a low cost, which may help research and development capabilities keep ahead of the pace of real-world missions. Multifunctional facilities are also cost-effective because they enable experimenters to test a variety of science objectives.

3.1.5 Resulting System Form

The SPHERES Facility aboard the ISS, which was described in Section 1.3, is an ideal candidate for such a testbed. The facility already allows for control algorithms, models, and CONOPS to be easily modified on-orbit. Additionally, to a limited extent, sensors and actuators can already be added to or removed from the system. Specifically, the SPHERES can be equipped with stereo-vision cameras or electromagnetic coils for sensing or actuating. By augmenting the SPHERES Facility with new hardware, it is possible to enable each SPHERES satellite to interface with multiple sensors and actuators simultaneously. The facility is already operating smoothly on the ISS, and expanding it is a cost-effective way to meet the requirements for a robotic assembly and servicing testbed.

3.2 Motivation for Halo

The SPHERES satellites have already been supplemented by the VERTIGO vision system, mentioned in Section 1.3, and have a limited ability to dock using velcro pads. However, in order for the SPHERES to meet the requirements set forth in Section 3.1, they must be augmented with further capabilities. The satellites each need to be able to dock and undock with a high probability of success, they need to be able to use one or multiple vision systems, and they need to be able to support some sort of

robotic arms. Most importantly, they need to be able to have all of these capabilities simultaneously. However, each satellite currently only has a single expansion port—known as Expansion Port Version 2.0—that can support one peripheral device at a time. A peripheral device is anything that can be attached to a SPHERES satellite. The Expansion Port Version 2.0 is pictured in Figure 3-10.

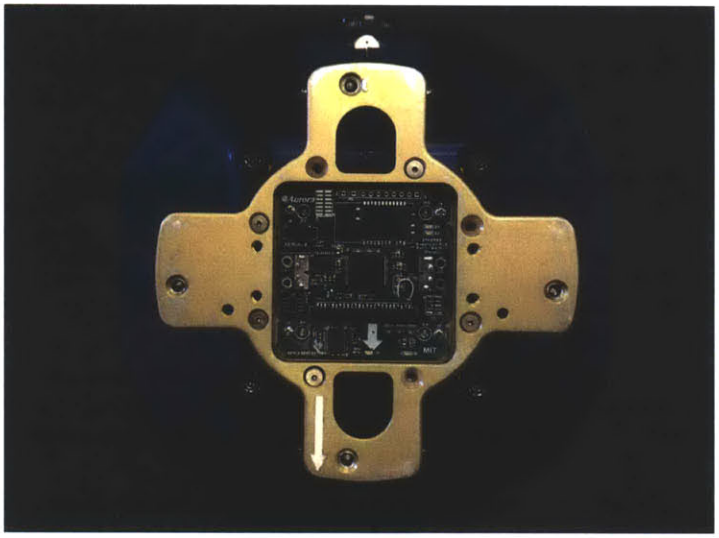


Figure 3-10: Expansion Port Version 2.0 on a SPHERES Satellite

Therefore, the satellites each need some sort of adapter that allows them to interface with multiple peripherals at the same time. That adapter is known as the Halo.

3.3 Halo Requirements

This section presents all of the requirements for the Halo system. These requirements stem from the testbed requirements in Section 3.1 and the motivation for the Halo in Section 3.2.

3.3.1 System Level Requirements

Table 3.2 lists all of the system-level requirements imposed on the Halo. Each requirement is given a label of the form "SYS-#" for reference purposes. The requirements

are not listed in any particular order. Requirements at every level are derived from an objective statement or a higher-level requirement that is listed in the "Parent" column. All of the system-level requirements are derived from the objective statements discussed in Section 3.1. The table also lists the verification method for each requirement: either analysis (A), inspection (I), testing (T), or demonstration (D).

Table 3.2: Halo System Level Requirements [45]

ID	Statement	Parent	Verification
SYS-1	The Halo and its expansions shall operate safely within the ISS as defined by existing NASA Safety Regulations and ISS Operations	2.1	T
SYS-2	Individual SPHERES satellites shall enable the utilization of multiple peripherals simultaneously	1.1, 1.2, 1.3	D
SYS-3	SPHERES satellites shall provide a mechanical and electrical interface that enables the existing peripherals' full capabilities	4.1, 4.2	T
SYS-4	A rigid structural interface shall exist between the SPHERES satellites and attached peripherals	2.4	I
SYS-5	The structural interface shall provide additional mounting locations to further extend hardware without utilizing an electrical interface	4.2	I
SYS-6	SPHERES satellites with the Halo and its expansions shall be fully controllable and observable	2.3, 2.4, 4.1	A
SYS-7	SPHERES satellites with the Halo and its expansions shall enable testing durations that are representative of on-orbit robotic assembly missions	2.4	A
SYS-8	The time to assemble the Halo onto SPHERES shall permit system setup, experiment execution, and storage within the allotted test time of an ISS SPHERES test session	4.1	D

Note that the Halo must comply with all of NASA's safety and integration requirements for flying aboard the ISS, per SYS-1. Derived requirements and the specifics of the Halo safety analysis are not discussed in this document.

All subsystem requirements shown in the following sections have been derived from the Halo system requirements described in Table 3.2.

3.3.2 Structures Subsystem Requirements

Table 3.3 lists the requirements for the structures subsystem (STR) within the Halo.

Table 3.3: Structures Subsystem Requirements [45]

ID	Statement	Parent	Verification
STR-1	The Halo structure shall provide multiple ports which allow for rigid mechanical and electrical mounting of existing SPHERES peripherals	SYS-2, SYS-4	T
STR-2	The Halo structure shall position Halo ports such that their surface-normal vectors point in at least two dimensions	SYS-2, SYS-4	I
STR-3	The Halo structure shall position all Halo ports such that there is no mechanical interference when both installing and using multiple peripherals simultaneously	SYS-2, SYS-4	I
STR-4	The Halo structure and any hard-mounted component shall permit all SPHERES thrusters to fire without plume impingement greater than that which would cause the system to become uncontrollable	SYS-6, SYS-4	D
STR-5	The Halo structure shall permit the SPHERES satellite to determine its position and attitude using the pre-existing metrology system (IR and US)	SYS-6, SYS-4	T
STR-6	The Halo structure shall allow removal and installation of all (i.e. SPHERES, Halo, and VERTIGO Avionics) batteries without disassembling the Halo	SYS-7, SYS-4	D
STR-7	The Halo structure shall allow removal and installation of the SPHERES CO_2 propellant tank without disassembling the Halo	SYS-7, SYS-4	D
STR-8	The Halo structure shall allow adjustment of the SPHERES CO_2 propellant tank regulator without disassembling the Halo	SYS-7, SYS-4	D
STR-9	The Halo structure shall provide the operator with a visible line of sight to the CO_2 propellant gauge	SYS-7, SYS-4	I
STR-10	The Halo structure shall allow the operator to execute all intended functions with the SPHERES user interface controls (i.e. buttons and switches)	SYS-7, SYS-4	D

More detailed rationales for each of the requirements follow.

Rationales and Descriptions:

STR-1 through STR-3

The first three structural requirements flow from SYS-2 and SYS-4, which state that multiple peripherals must be mounted to the SPHERES satellites simultaneously and rigidly. The ports to which peripherals are attached are known as "Halo Ports." Each Halo Port must provide both a mechanical and electrical interface for peripherals. These ports must be oriented in such a way as to allow multiple peripherals to be attached at the same time without contacting each other. Such a requirement rules out designs in which multiple ports are very close together or facing each other. The

ports also must not all be aligned in a single dimension, because that would only allow for one type of straight-line test configuration when the satellites are docked. Rather, designing the ports' surface-normal vectors to point in more than one dimension increases the number of possible test configurations and allows for more interesting robotic assembly and servicing testing.

STR-4 through STR-10

All of the remaining structural requirements are based on SPHERES satellite operations. Astronauts operating a Halo-equipped SPHERES satellite must have access to certain parts of the satellite without any obstruction by the Halo. These parts include all of items listed in Table 3.4, known as the "SPHERES Keep Out Zones."

Table 3.4: SPHERES Keep Out Zone Requirements [45]

Category	Req. #	Item	"Keep Out" Zone
Consumables	STR-6.1	Battery doors	70 mm radius from hinge
Consumables	STR-7.1	CO_2 Tank	6 in. diameter, 8 in. distance
Consumables	STR-8.1	Pressure regulation knob	6 in. diameter
Operations	STR-10.1	SPHERES Control Panel	Clear
Operations	STR-9.1	Pressure gauge	LOS
Performance	STR-4.1	Thrusters	18 deg. cone
Performance	STR-5.1	Ultrasound (US)	90 deg. cone
Performance	STR-5.2	Infrared (IR)	120 deg. cone

The three categories listed in Table 3.4 are described below, along with further descriptions of what it means to meet each requirement:

Consumables: consumables include batteries and CO_2 tanks that must be changed during a test session; for efficiency purposes, astronauts must be able to change these items without having to remove the Halo or any peripherals

Operations: areas such as the SPHERES control panel and pressure gauge must be visible and accessible to the astronauts at all times during a test session

Performance: the performance of the system is degraded if the Halo blocks the thrusters, ultrasound receivers, or infrared transmitters

- *Thrusters:* (STR-4.1) actuation performance is degraded if the Halo impinges upon the thruster plumes

- *Ultrasound:* (STR-5.1) when the system is in any position/orientation in the test volume, at least 4 of the 5 global positioning beacons located around the test volume must each have LOS to at least 3 of 4 US receivers on at least 1 face of the SPHERES

- *Infrared:* (STR-5.2) at least the 4 beacons defined in STR-5.1 must have LOS to at least one IR transmitter while the system is at any position/orientation in the test volume

NOTE: The Halo or any peripherals must provide a functional replacement for any violated keep out zone.

Figure 3-11 shows the approximate thruster plumes for the 12 thrusters on the SPHERES satellite. The plumes extend several meters beyond the thruster nozzles, and the grey sphere in the figure is meant to show the plume radius at an arbitrary distance. The best-case scenario is zero plume impingement. However, slight impingement is acceptable and probably necessary. Impingement that simply slows down the acceleration of the satellites is tolerable to a point, and that limit is based on the length of time needed to conduct an interesting robotic assembly or servicing test. Note that the center of the thruster plumes contains the airflow with the highest velocity, and should be avoided if at all possible.

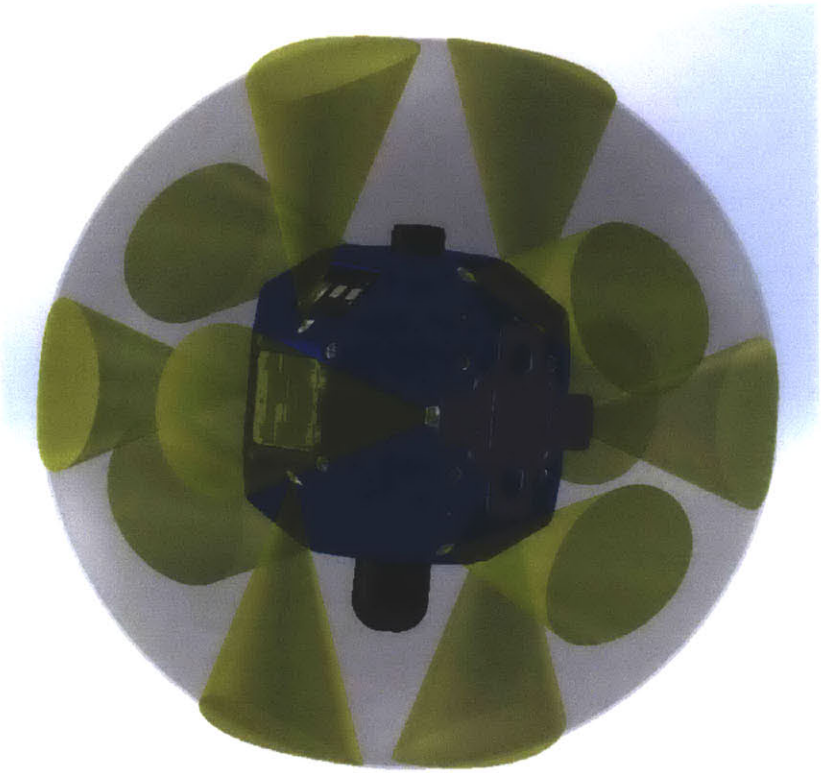


Figure 3-11: SPHERES Thruster Plume Keep Out Zones [45]

The keep out zones shown in Figure 3-12 and Figure 3-13 are fields of view. There are four ultrasound receivers on each face of the satellites, and those receivers can detect pulses within direct line-of-sight up to four meters away. There are also two infrared transceivers on each face of the satellites. The transceivers can transmit an infrared flash to any line-of-sight location within their field of view. The structure can block portions of these cones as long as STR-5.1 and STR-5.2 are met. Sensors must be replaced for keep out zones that are completely violated.

Figure 3-12: SPHERES Ultrasound Receiver Keep Out Zones [45]

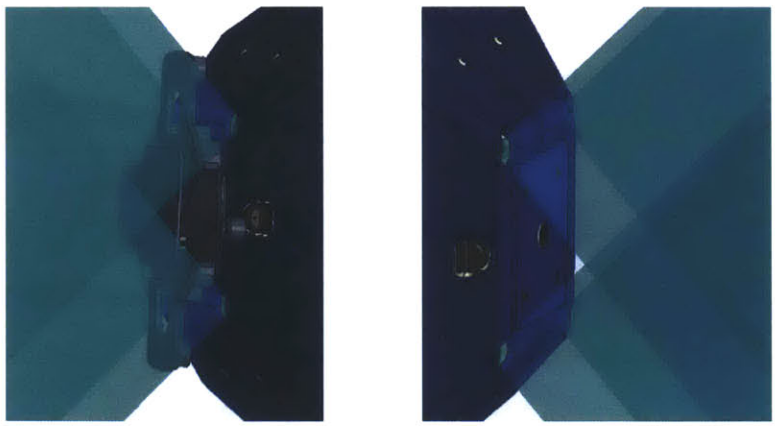


Figure 3-13: SPHERES Infrared Transceiver Keep Out Zones [45]

The crew must also be able to insert and remove CO_2 tanks from the bottom of the satellite (a process which requires screwing the tank in or out), adjust the regulator knob on the top of the satellite, and fully open the battery doors in order to install or remove a battery pack. Those keep out zones are shown in Figure 3-14. These areas should not be blocked by the Halo or any peripheral.

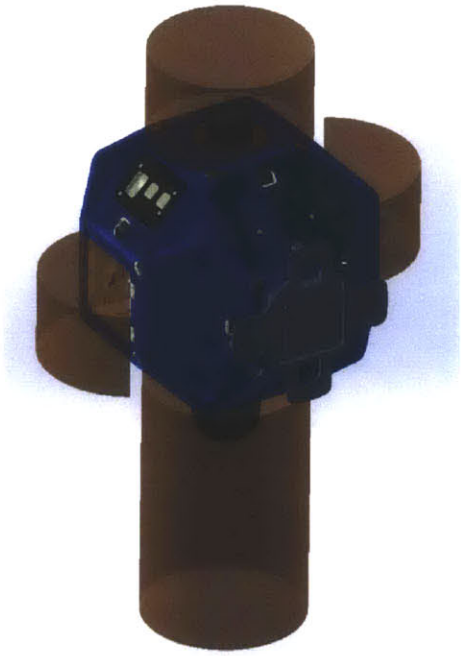


Figure 3-14: SPHERES Battery Door, CO_2 Tank and CO_2 Regulator Knob Keep Out Zones [45]

3.3.3 Data Handling Subsystem Requirements

Table 3.5 lists the Halo's data handling subsystem (DHS) requirements.

Table 3.5: Data Handling Subsystem Requirements [45]

ID	Statement	Parent	Verification
DHS-1	The Halo DHS shall provide at least one port with two dedicated USB protocol data lines and VERTIGO 50-pin connector	SYS-3	T
DHS-2	The Halo DHS shall provide all ports at least one Ethernet protocol data line and at least one USB protocol data line	SYS-3	T
DHS-3	The Halo DHS shall replace up to four blocked US/IR sensors	SYS-3, SYS-6	T
DHS-4	The Halo DHS shall allow peripherals on all Halo ports to be operated simultaneously	SYS-2	T
DHS-5	All data requiring hard real-time action shall be routed to the SPHERES satellite	SYS-3, SYS-6	T

More detailed rationales for each of the requirements follow.

Rationales and Descriptions:

DHS-1: The Halo must be able to support the VERTIGO Optics Mount in at least one port. This demands that one port has two dedicated USB lines and can interface with the Optics Mount 50-pin Samtec connector. The Optics Mount is necessary for many of the CONOPS.

DHS-2: In order for the Halo to be extensibility to future peripherals, each Halo port must provide standard Ethernet and USB data protocols. These are common protocols and many peripherals will likely require them.

DHS-3: The Halo must replace any blocked ultrasound or infrared sensors. Because there are only four ultrasound and infrared lines coming out of the SPHERES Expansion Port Version 2.0, only up to four ultrasound and infrared sensors must be replaced.

DHS-4: The Halo must support operations in which multiple peripherals are operating and communicating at the same time. This is a requirement based on the CONOPS. Without this capability, the functionality of the testbed is significantly

decreased.

DHS-5: Hard-real-time data, such as the information to and from the ultrasound and infrared sensors, must be routed to the SPHERES satellite because it is the only hard-real-time system in the Halo design. All other data should be routed to the much more capable VERTIGO Avionics Stack computer.

3.3.4 Electrical Power Subsystem Requirements

Table 3.6 lists the Halo's Electrical Power Subsystem (EPS) requirements.

Table 3.6: Electrical Power Subsystem Requirements [45]

ID	Statement	Parent	Verification
EPS-1	The EPS shall deliver the same voltages to all Halo ports as the VERTIGO Avionics Stack provides to the VERTIGO Optics mount	SYS-2, SYS-3	T
EPS-2	The EPS shall allow the system to run uninterrupted tests for durations that are representative of on-orbit robotic assembly missions	SYS-7	A
EPS-3	The EPS shall provide sufficient current to support any combination of peripherals simultaneously	SYS-2	T
EPS-4	The EPS shall protect SPHERES and any attached peripherals from power surges	SYS-1	T

More detailed rationales for each of the requirements follow.

Rationales and Descriptions:

EPS-1: The Halo must be able to support the VERTIGO Optics Mount by supplying the necessary voltages and by being electrically compatible. The docking port also requires the same voltages. A short study of future peripherals has shown that they will also demand the same voltages. Thus, the Halo should supply both 11.1V and 5V at every port.

EPS-2: The Halo EPS should provide enough battery life to execute the CONOPS associated with each of the eight architectures described in Section 3.1.1. The specific capacity requirement is based on the number of maneuvers and the operating peripherals for those mission scenarios.

EPS-3: The Halo must support operations in which multiple peripherals are powered on at the same time. For example, the Docking Ports may need to be used at the same time as the Optics Mount.

EPS-4: The EPS should protect the SPHERES and VERTIGO Avionics Stack from any power surges that may result from a peripheral being short circuited or plugged in incorrectly. The Halo should also protect itself from damage in that case.

3.4 Requirements Wrap-up

Chapter 3 laid out the requirements for the development of a space-based robotic assembly and servicing testbed, described the resulting hardware augmentation that the SPHERES Facility requires, and enumerated the specific requirements for that piece of hardware, known as the Halo. The next steps involve developing a suitable design for the Halo that can interface with the existing SPHERES Facility and enhance its capabilities. This process is described in depth in Chapter 4.

Chapter 4

Conception, Design, Implementation, and Planned Operations

Based on the requirements laid out for the Halo in Chapter 3, Chapter 4 explains the CDIO process used to develop the Halo. CDIO is a framework used for educating students by teaching them to Conceive, Design, Implement, and Operate real-world systems [51]. This chapter steps through each of the elements of CDIO.

The Halo was originally conceived and designed by the MEDUSA team in Professor David Miller's 16.851 class. Much of the work from that class has been expanded upon to fully flesh out the concept and design of the Halo, which are presented in this chapter.

Other large contributors to the Halo concept include Professor Miller and Dr. Alvar Saenz-Otero. Dr. Saenz-Otero, along with John Merk and Roedolph Opperman from Aurora Flight Sciences, have also been large contributors to the design and implementation of the Halo, which are presented in this chapter. The author worked closely with Chris Jewison throughout much of the process.

4.1 Halo Concepts and Trades

The first step of the CDIO process is "Conceive." Given the requirements for the Halo system discussed in Section 3.3, the MEDUSA team in Professor Miller's 16.851 class generated multiple concepts that could meet the requirements and also created a system for trading between them in order to select the best concepts. Dustin Hayhurst and Brandon Karlow were key players in developing the system. Chris Jewison was responsible for the structure subsystem, Pete Davison was responsible for the data subsystem, and the author was responsible for the power subsystem. This section describes the concepts that the team selected.

4.1.1 Structures Subsystem Concept Selection

In order to create a concept for the structure that met the requirements laid out in Section 3.3.2, several concepts were developed with the aim of spanning the tradespace of options. Each of the concepts had a different shape, mounting location, or number of ports. Concepts that did not meet the requirements set forth for the Halo structure were thrown out. Using a decision matrix, the concepts were each scored according to metrics such as mass and structural rigidity. Dominated concepts were then eliminated, and there were three remaining concepts, pictured in Figure 4-1 [52].

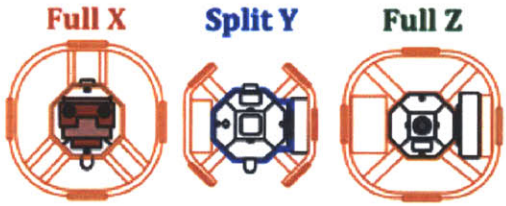


Figure 4-1: Full-X, Split-Y, and Full-Z Concepts [52]

The decision matrix for the three remaining concepts is shown in Figure 4.1.

Table 4.1: Structure Decision Matrix [52]

Scoring Matrix for HALO Structure				
Parameter	Weighting of Parameter	Alternatives		
		Full Ring X	Split Ring Off Y	Full Ring Z
Assembly Time	10	10.00	8.00	6.00
Number HALO Ports	10	9.60	8.40	7.60
Structural Rigidity	6	6.50	7.00	9.00
Inertia	6	2.63	5.89	1.94
Mass	6	1.93	3.67	1.00
Cost	5	6.00	8.33	6.33
Rel. Access to Batteries	5	5.00	10.00	5.00
Rel. Access to Tank/Regulator	5	6.00	10.00	10.00
Rel. Access to UI	5	9.00	10.00	10.00
# Ind. Comps.	5	8.40	8.50	8.70
Workspace	4	8.23	6.60	4.11
Internal Space	4	3.60	6.00	9.00
Non-orthogonal Ports	4	10.00	6.00	10.00
Pinch/snag Points	1	7.00	5.00	7.00
Un-weighted Sum		93.88	103.39	95.69
Weighted Score		**6.956**	**7.591**	**6.675**

Although some of the concepts were easily scored objectively for certain metrics, such as mass, there was a level of subjectivity in assigning weightings and other scores. In order to combat the inherent subjectivity of this decision process, each score for a given metric was given a 2σ uncertainty, and the weighting for each metric was given a 2σ uncertainty. Scorings that were more objective or quantifiable received smaller uncertainty values, and more subjective scores received larger uncertainty values. This method yielded the results shown in Figure 4-2 [52].

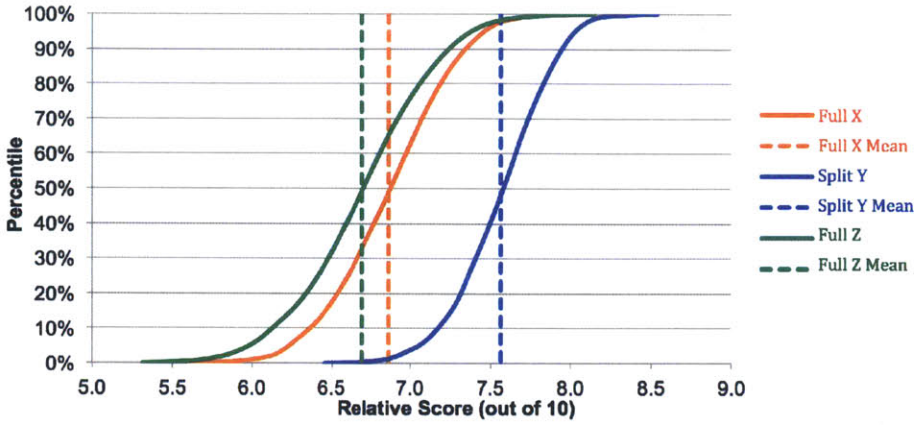
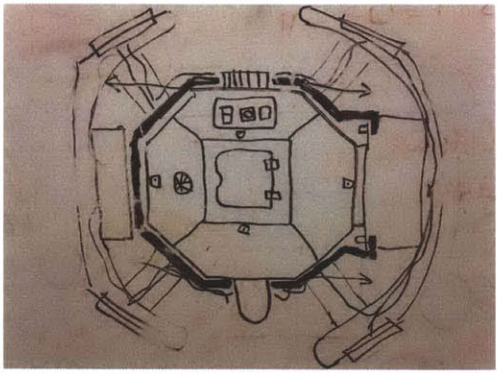


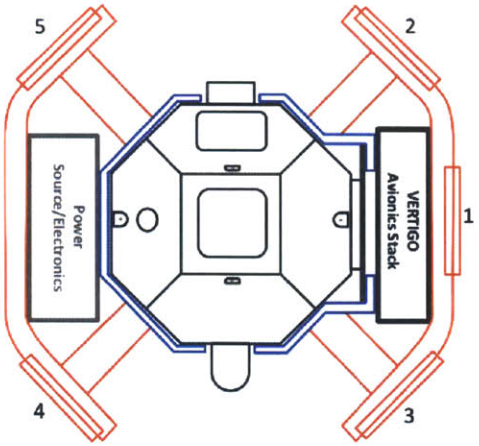
Figure 4-2: Cumulative Distribution Functions for the Three Remaining Structural Concepts [52]

The dotted lines represent the mean scores, and the scores over 2000 Monte Carlo iterations are plotted to give the curves. Steeper slopes represent higher confidence than shallower slopes. This method allowed the team to discover the range of scores that a given concept might receive and the confidence in those scores. A sensitivity analysis was also conducted. Based on these results, the Split-Y concept was selected [52].

The Split-Y concept scored well due to the easy accessibility it provides to the SPHERES satellite and VERTIGO Avionics Stack, the lack of thruster, ultrasound sensor, and infrared transceiver impingement, and the smaller radius, giving it a lower mass and inertia. The first ever drawing of the Split-Y design, credited to Chris Jewison, is shown in Figure 4-3a, and the team's design at the end of the course is shown in Figure 4-3b [52].



(a) Original Drawing



(b) MEDUSA Team Final Concept

Figure 4-3: Conceptual Design of the Split-Y Halo Structure [52]

In Figure 4-3b, the Halo structure is shown in orange, the mounting sleeve is shown in blue, the VERTIGO Avionics Stack is shown in black, and the originally planned location for the power subsystem is shown in grey.

4.1.2 Data Handling Subsystem Concept Selection

The two primary questions the team asked when coming up with a concept for the data handling subsystem in the Halo were as follows:

- Where does the Halo data pathway connect to existing elements in the system?
- How does the Halo communicate at that connection?

Figure 4-4 lays out the three choices that were available. The dashed vertical lines represent the locations in which a Halo could be connected, and the data protocols at each interface are listed [52].

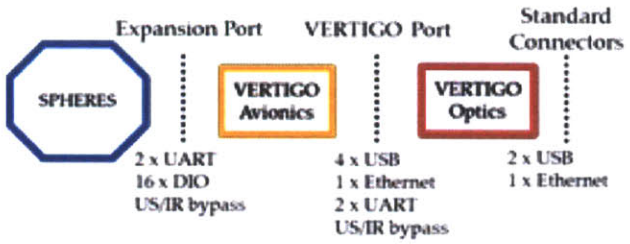


Figure 4-4: Three Possible Halo Data Handling Subsystem Interfaces [52]

Based on these options and considering the requirements, the team came up with four possible concepts:

- Expansion Port Version 2.0/UART (SPHERES-centric UART)
- VERTIGO Avionics Stack Interface/Ethernet (VERTIGO Avionics Stack-centric + Ethernet)
- VERTIGO Avionics Stack Interface/UART (VERTIGO Avionics Stack-centric + UART)
- Standard Connector/Ethernet (VERTIGO Goggles-centric + Ethernet)

Each of these concepts were scored in a decision matrix according to metics such as the processing power each option would provide, and the results are shown in Figure 4-5 [52].

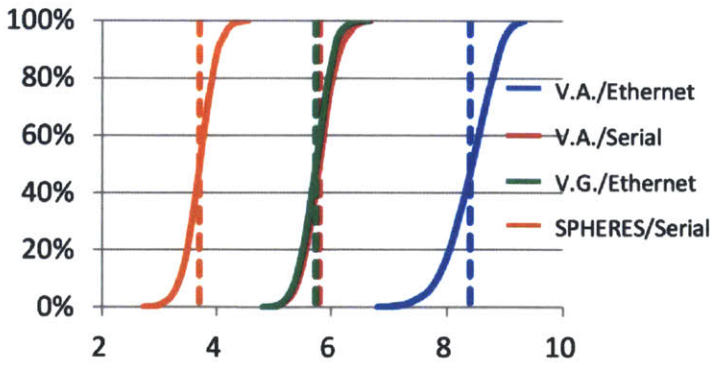


Figure 4-5: Cumulative Distribution Functions for the Four Data Handling Concepts [52]

Based on these results, the VERTIGO Avionics Stack-centric Ethernet-based concept was selected [52].

4.1.3 Electrical Power Subsystem Concept Selection

When coming up with concepts for the power subsystem, it was important to understand the components that already existed and the constraints they imposed. First, the current SPHERES power subsystem consists of two packs of eight AA batteries that power the entire bus and produce a regulated power supply to the outside world of 3.3V, 5V, 15V, and -15V, at 0.5A. Additionally, the VERTIGO Avionics Stack also has its own power supply—a rechargeable Nikon camera battery—which powers the stack as well as the VERTIGO Optics Mount. The team estimated that the Halo power subsystem had to have a capacity of at least 10Wh to complete a full CONOPS [52].

The first trade study looked at how to support the demanding peripherals without limiting test duration. The three options considered include powering the peripherals through an independent power supply, through the SPHERES power supply, or through the VERTIGO power supply. The SPHERES power subsystem was designed to power only the Docking Ports, and thus cannot support the power and voltage constraints that other peripherals necessitate. Similarly, the VERTIGO Avionics Stack power supply can only support itself and the VERTIGO Optics Mount for 70 minutes, so it is already the weakest link. Thus the Halo power subsystem had to have an independent power supply [52].

The next step involved selecting the optimal battery to power the Halo system. The batteries used in SPHERES, VERTIGO, and RINGS all have flight and program heritage, and thus were the three options researched for the second power subsystem trade study. The SPHERES use two packs of 8 AA batteries, VERTIGO uses a Nikon battery, and RINGS uses a DeWalt battery. Battery specifications are shown in Table 4.2. Because of the constraints that the peripherals impose on the system, both the Nikon and AA batteries were also considered as a set of four batteries in parallel, to increase the capacity. This quadrupled their available current and capacities, but

also quadrupled their masses and volumes [52].

Table 4.2: Flight-Certified Battery Specifications [52]

Battery	DeWalt	Nikon	4 Nikon	8 AA	4x 8 AA
V	18	11.1	11.1	~12	~12
Wh	36	28	112	~25	~100
mAh	2000	2500	10000	2000	8000
kg	0.69	0.162	0.648	0.226	0.904
cm^3	~700	115	460	108	432
Max A	~4	~1.9	~7.6	~1.35	~2.70
Wh/kg	52	173	86	111	55

These concepts were put into the power subsystem decision matrix. The key parameters used to evaluate the concepts were duration of testing, mass, volume, and complexity. Using the scoring method mentioned earlier in this section, the best battery turned out to be VERTIGO's Nikon rechargeable battery, arranged with four in parallel. This concept is the best because it can provide a high enough voltage and current for the peripherals, it has enough capacity to support an intense CONOPS, and it is still smaller and lighter than a single DeWalt battery. Its size integrates easily into the Halo structure, and its mass minimizes impact on Halo mass and inertia [52].

4.2 Halo Hardware Design

The Halo design flows directly from the selected concepts in Section 4.1. This section transitions from the "conceiving" portion of the CDIO process to the "designing" portion, in which the selected concepts are refined and solidified into a design that can be built. This section explores the design of the components and the decisions made along the way to the final design.

4.2.1 System Overview

The Halo expands the current capabilities of the SPHERES on the ISS. It does so by enabling each satellite to interface with six external objects simultaneously through rigid mechanical and electrical interfaces called "Halo ports." These external objects, called "peripherals," include the following: docking ports, used for testing docking algorithms and reconfiguring satellite geometry; sensors, such as the VERTIGO Goggles; and actuators, such as control moment gyroscopes. These peripherals can be added and removed from the Halo by the astronaut as necessary for each test. The Halo has the processing power of the VERTIGO Avionics Stack and provides Ethernet and USB to each Halo port for any future peripherals. The Halo is able to provide power to all attached peripheral devices for a duration that allows uninterrupted, interesting robotic assembly testing. The new capabilities enable testing of on-orbit robotic assembly and servicing in a risk-tolerant, dynamically authentic environment by addressing challenges such as aggregating resources and reconfiguring control systems. The fully labeled Halo system is shown in Figures 4-6 and 4-7. HP stands for Halo Port, and HPG stands for Halo Port Goggles, to indicate the primary port to which the VERTIGO Optics Mount is attached.

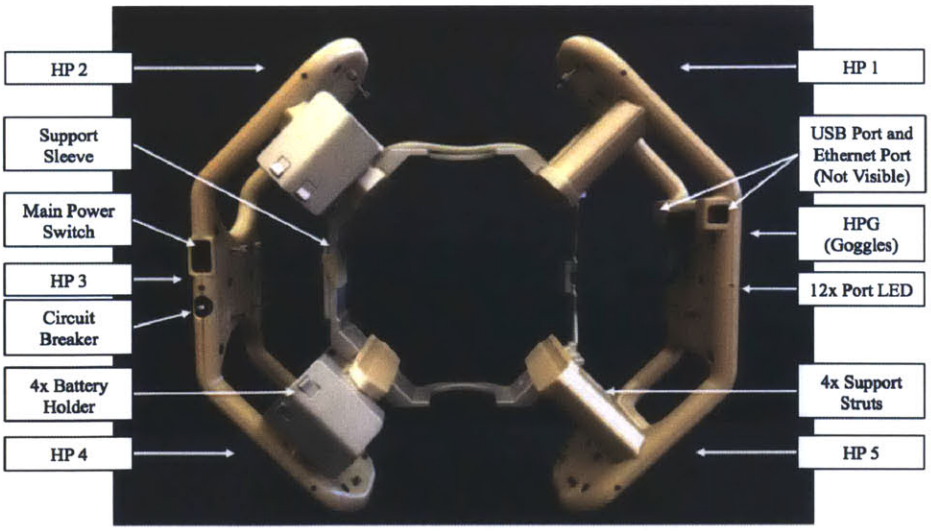


Figure 4-6: Labeled Halo System

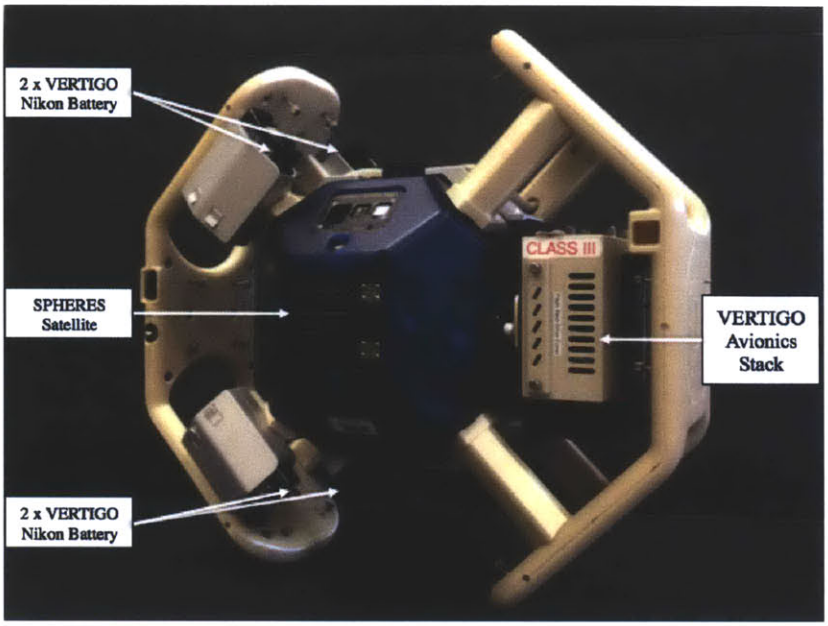


Figure 4-7: Labeled Halo System Mounted on SPHERES/VERTIGO Assembly

4.2.2 Halo Mechanical Design

In order to mount the Halo, a SPHERES satellite must first have the VERTIGO Avionics Stack attached to the Expansion Port Version 2.0. The Halo slides around the satellite/stack assembly, such that the Halo interfaces with the VERTIGO Avionics Stack 50-pin connector. The Halo includes six identical 50-pin connectors at six rigid HPs around the SPHERES. The structure supports six Printed Circuit Boards (PCBs): the Halo Motherboard at HPG, the Power board at HP3, and four HP boards at HPs 1, 2, 4, and 5. A key factor in the structural design was the avoidance of the SPHERES keep out zones, as described in Section 3.3.2. HP1 and HP5 are angled 45 degrees from the surface of HPG so that multiple peripherals can be attached simultaneously without impinging upon one another. This configuration also allows two Halo-equipped SPHERES to dock together without their peripherals contacting one another, as shown in Figure 4-8. Likewise, HP2 and HP4 are angled 45 degrees from the surface of HP3.

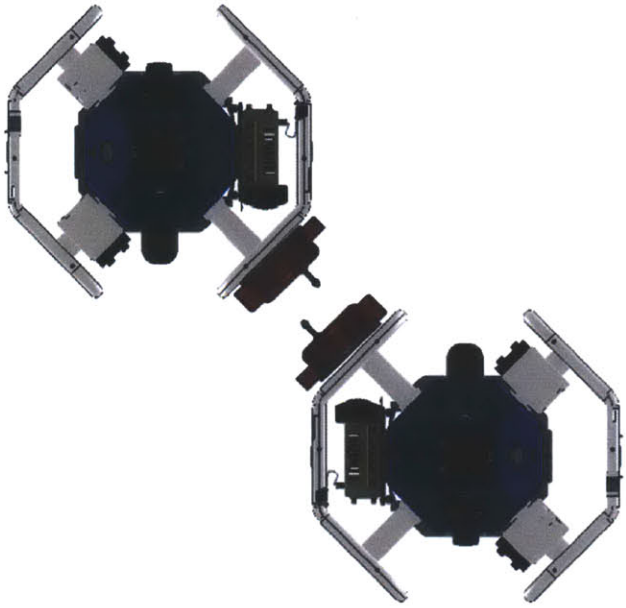


Figure 4-8: Two Halo-Equipped SPHERES in a Docking Configuration

Images of the Halo Prototype are shown below, and an exploded view of the Halo is shown in Figure 4-9. This figure also contains all of the part names for each component in the Halo.

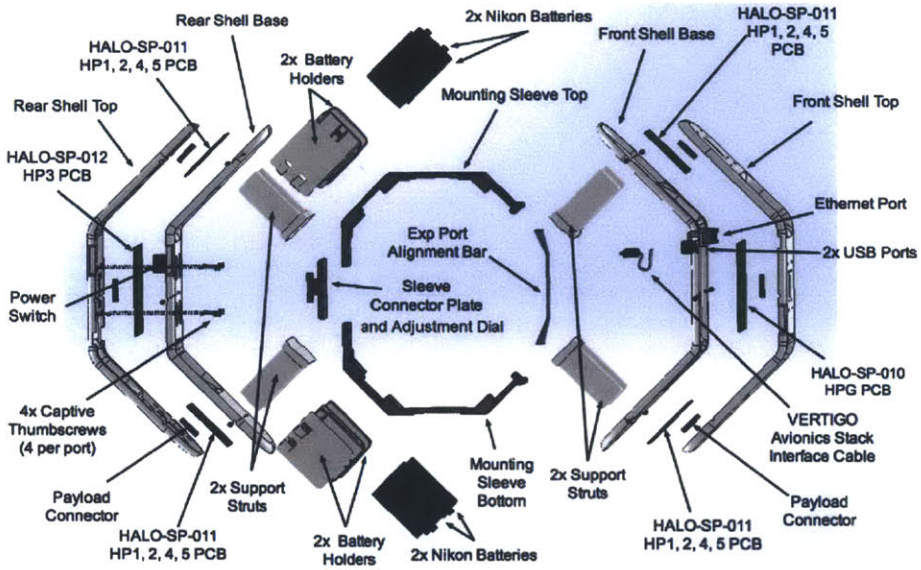


Figure 4-9: Halo Exploded View

Halo Size and Mass

The Halo has the following properties:

Table 4.3: Halo Size and Mass

Dimensions of Manifested Halo Hardware				**Mass [kg] (without Batteries)**
X-dim [cm]	**Y-dim [cm]**	**Z-dim [cm]**	**Panel Thickness [cm]**	
43.4	17.9	40.9	1.905	4.08
Dimensions of Halo/SPHERES/VERTIGO Assembly				**Mass [kg] (Includes SPHERES, VERTIGO Avionics Stack and all Batteries)**
X-dim [cm]	**Y-dim [cm]**	**Z-dim [cm]**	**Panel Thickness [cm]**	
43.4	21.6	40.9	1.905	9.89

These dimensions are depicted in Figure 4-10.

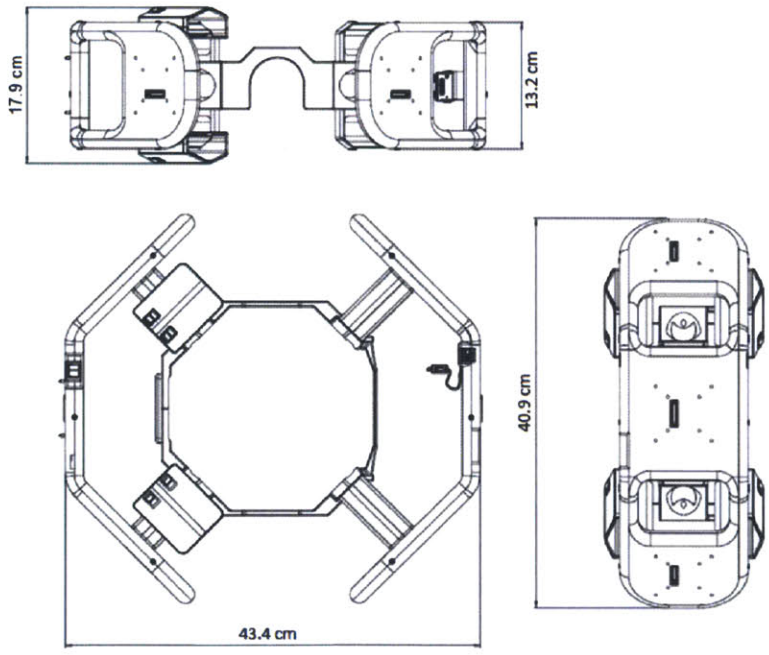


Figure 4-10: Halo Outer Dimensions

Halo Material Selection

Table 4.4 contains the material selection and rationale for each of the parts of the Halo.

Table 4.4: Halo Material Selection

Part	Material	Rationale
Sleeve and battery holders	6061-T6 Aluminum	Strong, yet cheap and lightweight – ideal for clamping the SPHERE (as used on RINGS)
Struts and port housings	3D Printed Ultem	Recently Flight Approved, can be designed to have adequate strength in key locations, lightweight, easy to manufacture

Halo Mounting Assembly

The Halo mounting assembly is highlighted in Figure 4-11.

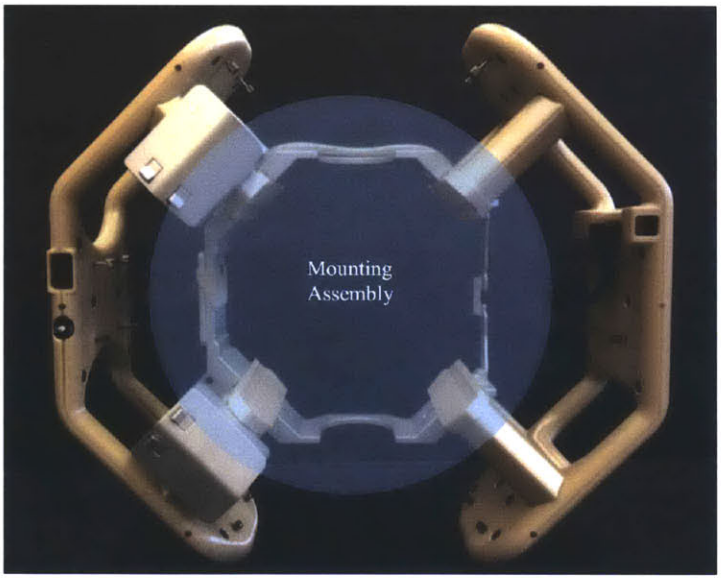


Figure 4-11: Halo Mounting Assembly

The Halo is mounted onto a SPHERES satellite using a press-fit design, similar to the Resonant Inductive Near-field Generation System (RINGS), which utilizes a C-clamp and four struts. The Halo uses a sleeve that wraps completely around the satellite, rather than the C-clamp. The sleeve includes an adjustable dial that is used to tighten and loosen the sleeve as necessary to fit a given SPHERES shell. The dial

is against the -X face of the satellite. Two of the four struts have been modified to incorporate four battery holders, which provide easy access for astronauts to install and remove batteries. For more information on the batteries and power subsystem, see Section 4.2.3. The mounting assembly is shown in Figure 4-12.

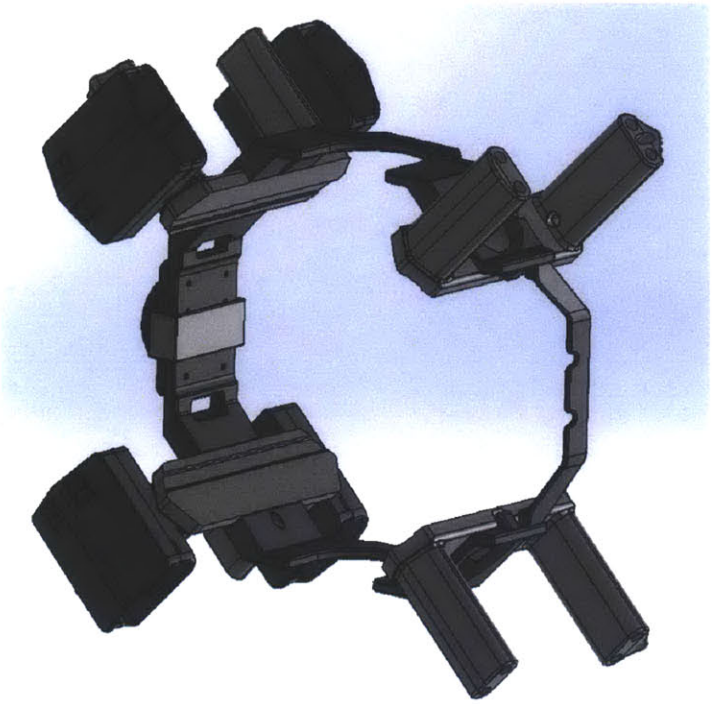


Figure 4-12: Sleeve and Struts for Mounting on SPHERES

This design avoids the SPHERES keep out zones and provides a rigid mount to the satellite. The sleeve is also be used to properly align the Halo on the satellite. A Teflon-insulated cable between the front and back halves of the Halo is routed along the struts and sleeve in a recessed channel. The channel in the prototype sleeve is pictured in Figure 4-13 with the cable laying next to it.

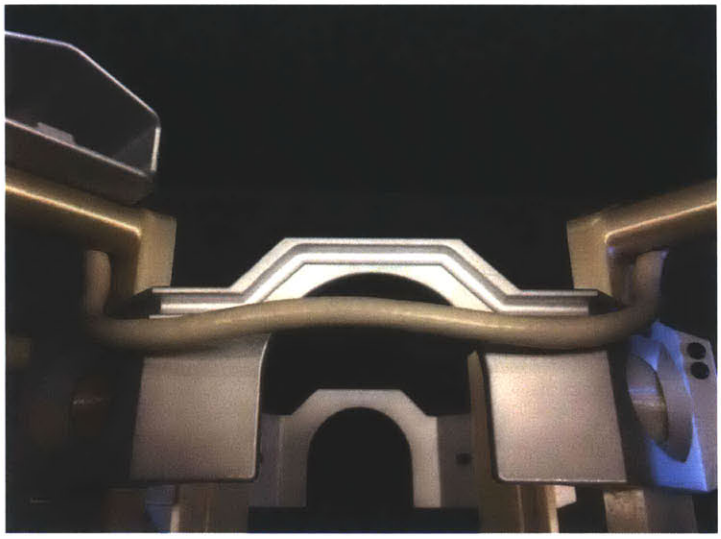


Figure 4-13: Sleeve Channel for Wire Routing

Halo Expansion Port Side

The Halo Expansion Port side is highlighted in Figure 4-14.

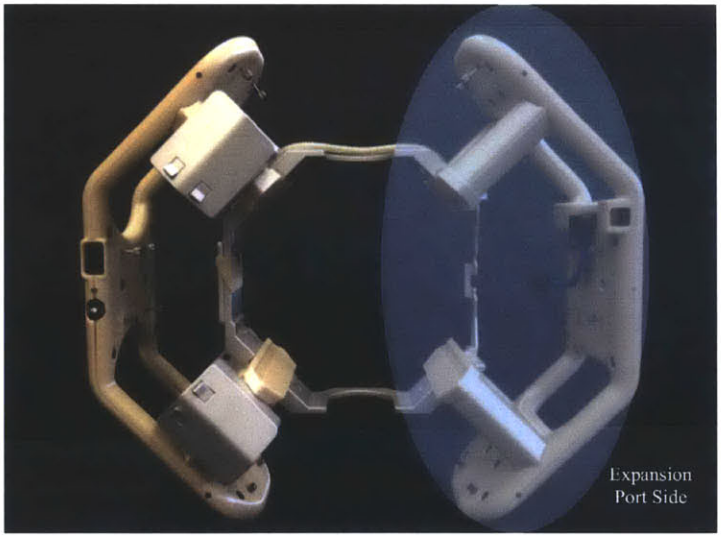


Figure 4-14: Halo Expansion Port Side

The Expansion Port side of the Halo attaches to the VERTIGO Avionics Stack through a connector between the VERTIGO Avionics Stack external interface and a connector on the back of HPG, and it also mounts onto the front two struts of the Halo mounting assembly. The inner shell of the Expansion Port side provides a

male 50-pin Samtec connector to interface with VERTIGO Avionics Stack, and this connector has an identical form-factor to the VERTIGO Optics Mount. It houses the Halo Motherboard as well as two other PCBs for the two other HPs on the Expansion Port side of the Halo while avoiding SPHERES keep out zones. It is pictured in Figure 4-15.

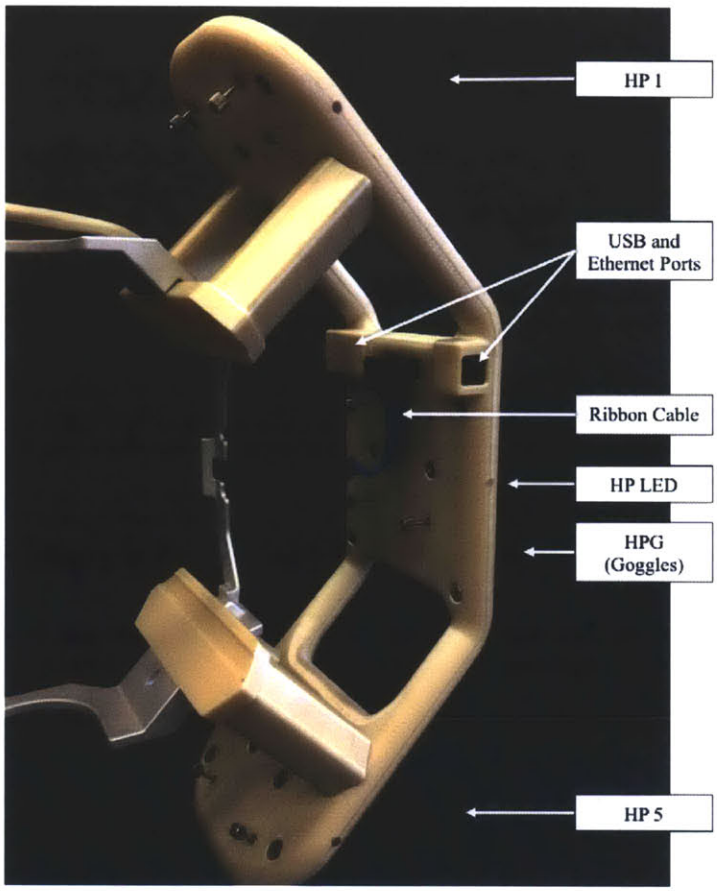


Figure 4-15: Expansion Port Side of the Halo

The Expansion Port side of the Halo houses the Motherboard immediately adjacent to the VERTIGO Avionics Stack, and the Motherboard contains all of the circuits for HPG as well. The Motherboard also includes three external connectors: two USB connectors and one Ethernet connector (RJ45). These connectors are externally accessible on the side of the Halo.

Because the outer (+X) face of the VERTIGO Avionics Stack has venting slits for hot air that is blown out of the VERTIGO Avionics Stack, there is a gap in between

the VERTIGO Avionics Stack and the Halo for venting. Although there are no major electrical components that emit heat, the Halo also has venting.

Each HP has two light-emitting diodes (LEDs) so that astronauts can easily determine whether or not any of the HPs are powered.

The Halo connector on the back of HPG mates to the VERTIGO Avionics Stack using a ribbon cable, similar to RINGS. This COTS connector simply clicks into place.

Halo Back Side

The back side of the Halo is highlighted in Figure 4-16.

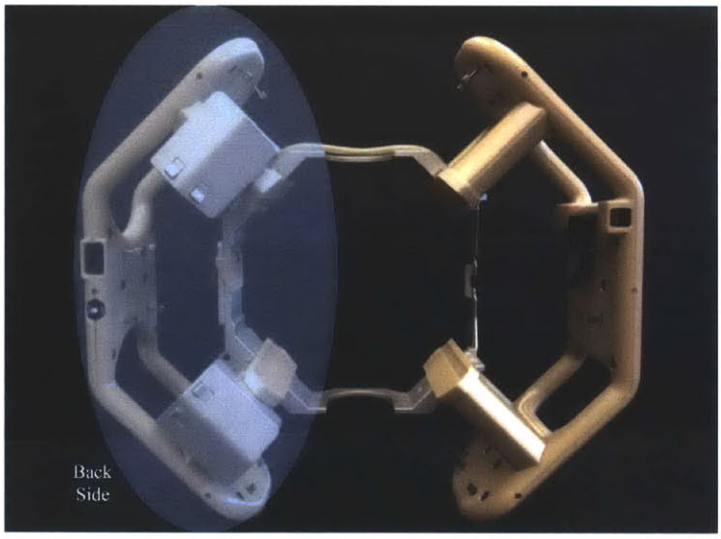


Figure 4-16: Halo Back Side

The back side of the Halo mounts to the back two struts in the Halo mounting assembly. It houses the Halo Power board as well as two other boards for the two other HPs on the back side. It also avoids the SPHERES keep out zones and is pictured in Figure 4-17.

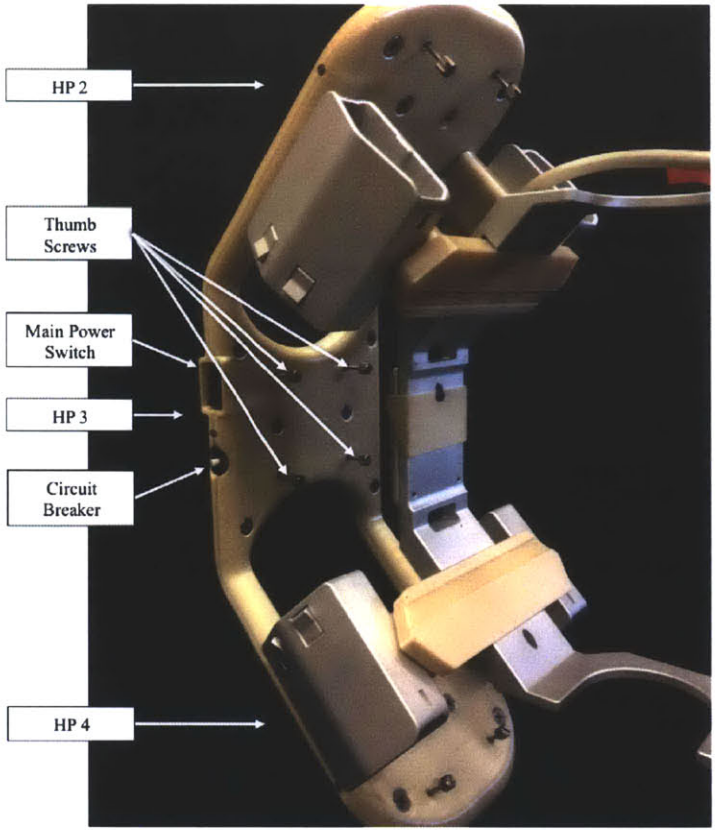


Figure 4-17: Back Side of the Halo

Because the back side of the Halo contains the Power board immediately adjacent to HP3, the Power board contains all of the circuits required for HP3 as well. The back side of the Halo also has the shrouded Halo master power switch and the main circuit breaker located on the side of HP3. As with the Expansion Port side, each HP has two LEDs.

Halo Ports

Each HP provides 4 male thumbscrews protruding outwards from the Halo on a flat face for flush mounting. The HPG face is 12 centimeters by 12 centimeters, HPs 1, 2, 4, and 5 are 12 centimeters by 9 centimeters, and HP3 is 12 centimeters by 7 centimeters. Similar to the SPHERES's Expansion Port Version 2.0, the Halo ports provide male screws so that peripherals simply need to provide female mounting holes rather than their own male screws. This reduces the necessary size and mass of pe-

ripherals. The screws are arranged in a square configuration with 5.715 centimeters on a side, which is different from the Expansion Port Version 2.0 (Expansion Port Version 2.0 has a diamond configuration, but is electrically incompatible), but interfaces well with the SPHERES Docking Port. In addition, the 50-pin connector is located 1.91262 centimeters down from the top of the square screw configuration—rather than centered—so that astronauts can easily recognize the proper mounting orientation. An image of the HPs is provided in Figure 4-18.

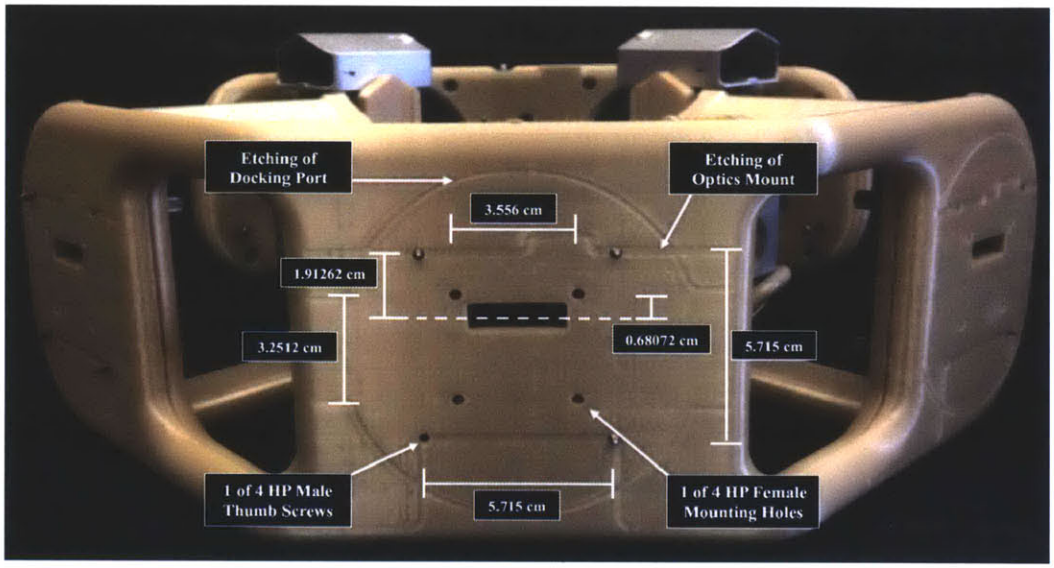


Figure 4-18: Halo Port Footprint

The HPs provide a mechanical connection that is identical to the VERTIGO Avionics Stack, including the female 50-pin Samtec connector (ERF8-RA Series; part number: ERF8-025-01-S-D-RA-TR) and the four female mounting holes. The holes and connector are in an identical configuration to those on the external face of the VERTIGO Avionics Stack. These holes are provided so that the VERTIGO Optics Mount, which has its own male screws and is designed to mount to female threaded holes, can mount on any HP. The electrical connectors are compatible.

Future peripherals may either provide four female mounting holes to align with the four male screws on the HPs or four male screws to align with the four female mounting holes on the HPs. Future peripherals should provide a male connector to

mate with the female connector on the HPs.

The male thumbscrews on the HPs are #4-40. When the male thumbscrews on the HPs are fully pushed in (extending as far away from the SPHERES and protruding as far into the peripheral as possible), they reach approximately 0.635 centimeters beyond the HP face, in order to be compatible with the Docking Port. Nominally, the male thumbscrews are fully recessed due to their springs. The rear grip screw section is 0.8382 centimeters in diameter.

The Halo port faces also each have an etching outlining both the Optics Mount and the Docking Port, which aids the astronauts in quickly attaching peripherals.

Halo Assembly Sequence

The Halo assembly sequence consists of following four steps, each illustrated in the accompanying images:

- Step 1) Mount the VERTIGO Avionics Stack onto the SPHERES Expansion Port Version 2.0

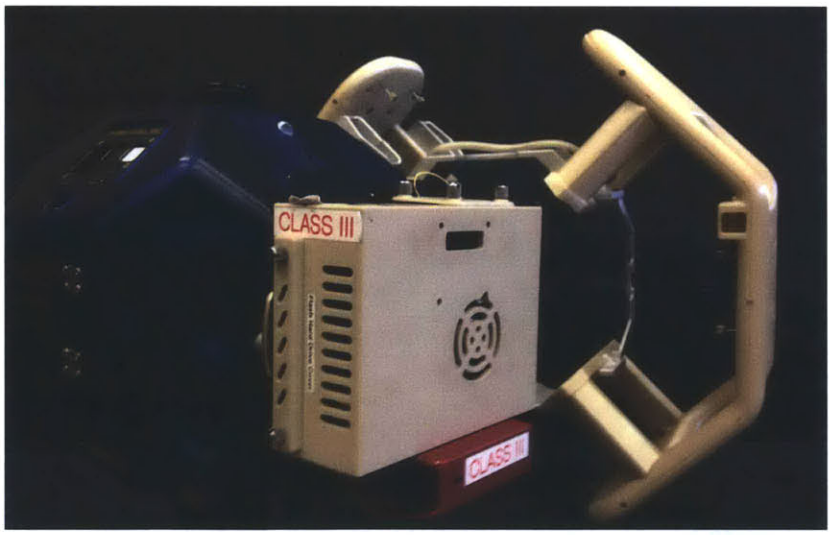


Figure 4-19: Step 1 of the Halo Assembly Sequence

- Step 2) Slide the SPHERES with VERTIGO Avionics Stack into the sleeve of the Halo

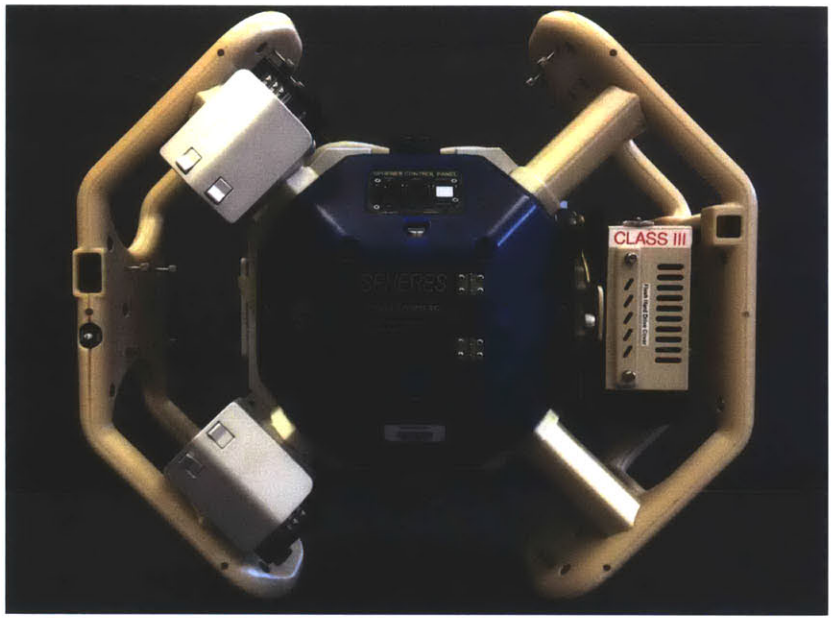


Figure 4-20: Step 2 of the Halo Assembly Sequence

- Step 3) Turn the adjustment dial on the rear of the Halo sleeve in a clockwise rotation to clamp down on the SPHERES satellite, ensuring a tight fit; to release, pull the knob outward

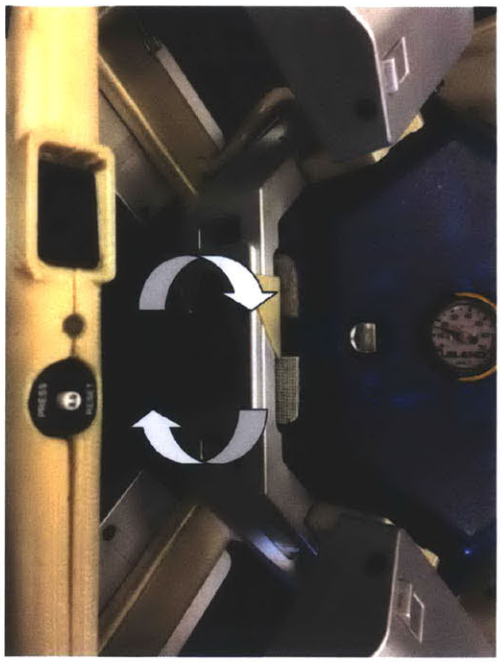


Figure 4-21: Step 3 of the Halo Assembly Sequence

- Step 4) Connect the tethered connector on the back side of HPG with the connector on the VERTIGO Avionics Stack so that the latching mechanism engages

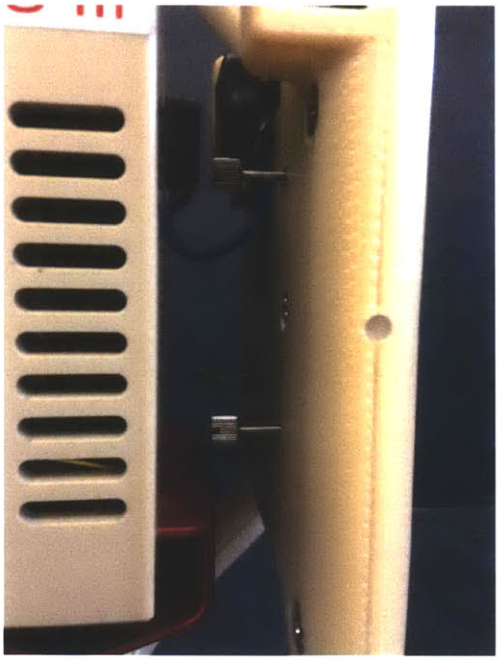


Figure 4-22: Step 4 of the Halo Assembly Sequence

SPHERES Keep Out Zones vs. Halo

Figure 4-23 demonstrates how the Halo completely avoids the battery doors, regulator knob, and tank.

Figure 4-23: SPHERES Battery and CO_2 Tank Keep Out Zones vs. Halo

The Halo also does not impinge on any of the thruster plumes in the Y-axis or Z-axis. As shown in Figure 4-24, the Halo does not impinge either of the thrusters on the -X face of the SPHERES.

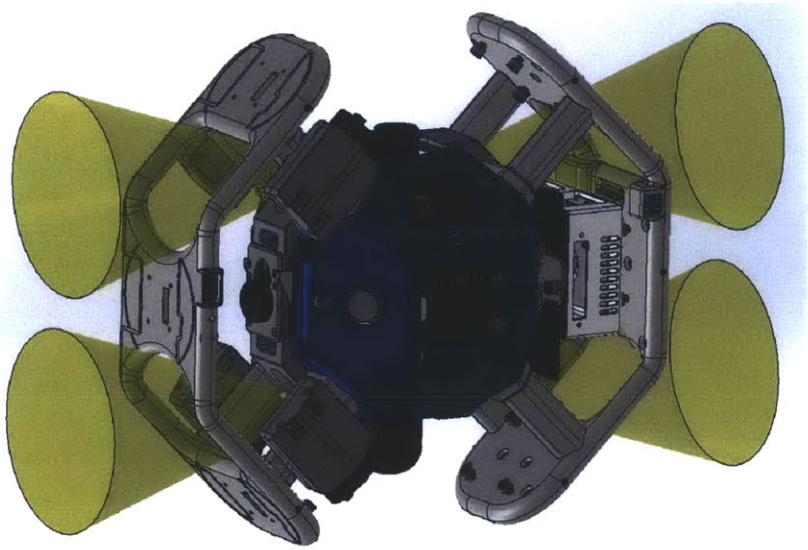
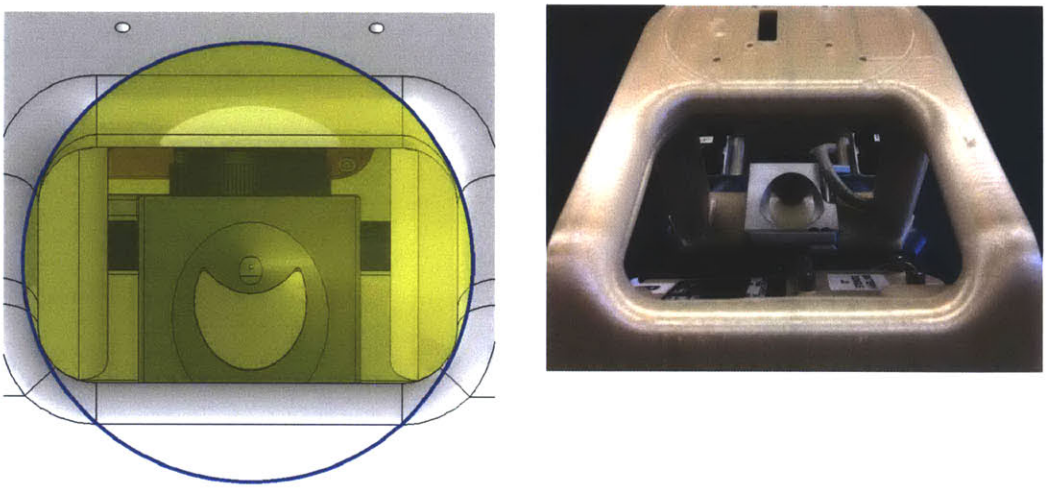


Figure 4-24: SPHERES Thruster Plume Keep Out Zones vs. Halo

The only thruster impingement by the Halo is on the two thrusters on the +X face of the SPHERES satellite. Figure 4-25 shows a head-on view of the thruster on the satellite's +X, +Z face.



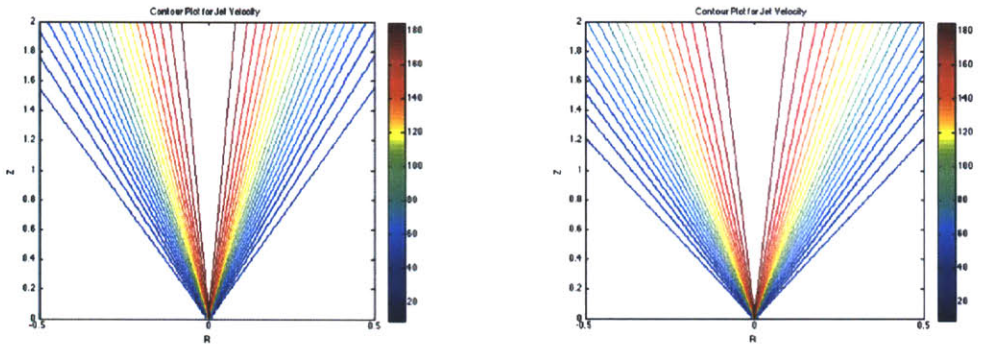
(a) CAD (b) Photograph

Figure 4-25: Head-on View of +X, +Z SPHERES Thruster with Halo Impingement

Note that the VERTIGO Optics Mount does not block anything that would not have been blocked by the Halo anyway. Thus, only the blockage by the Halo needs to be calculated. The amount of Halo impingement was quantified using the following equation for the SPHERES thruster velocity profile:

$$u(x,r) = u_{max} * e^{-29.5*r^2/x^2} \tag{4.1}$$

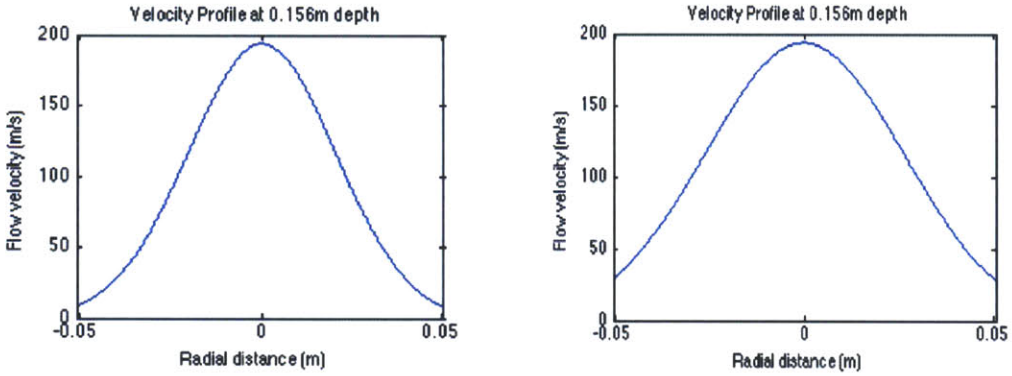
where $u(x,r)$ is the plume velocity at radius r and depth x and u_{max} is the maximum plume velocity [53]. The thruster plumes on the SPHERES are an 18-degree cone (9 degrees off the center line), per the Halo System Requirements Document. Thus, Equation 4.1 gives the velocity profile for the 18-degree cone in Figure 4-26a and for a 23-degree cone (5 degrees of margin) in Figure 4-26b.



(a) 18-degree Cone (b) 23-degree Cone

Figure 4-26: Thruster Plume Velocity Profiles

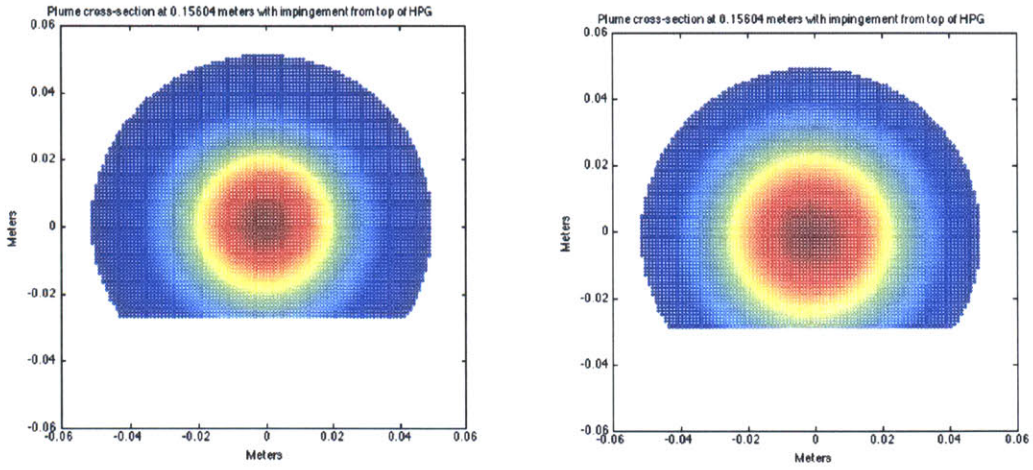
To aid in the visualization of the plume velocities, Figure 4-27a and Figure 4-27b show the thruster velocities at a distance of 0.156 meters as a function of radius.



(a) 18-degree Cone (b) 23-degree Cone

Figure 4-27: Velocity Profile at 0.156 meters

The percentage of plume blockage was calculated based on these thruster plume velocity profiles. Figure 4-28a and Figure 4-28b show plots of the plume blockage, with higher plume velocities in red and lower plume velocities in violet.



(a) 18-degree Cone

(b) 23-degree Cone

Figure 4-28: Plume Cross-section at 0.156 meters with Impingement from Top of HPG

Based on the velocities for the 18-degree cone, 7.07% of the plume is impinged. For the 23-degree cone, 10.28% of the plume is impinged. Table 4.5 shows the results after running the calculations for several different configurations.

Table 4.5: Thruster Effectiveness with Halo Impingement for Various Configurations

Configuration	Nominal plume	+5 deg plume
Thruster plain	100.00%	100.00%
Thruster w/ Halo	92.93%	89.72%
Thruster w/ Halo + UDP on HPG	91.73%	88.76%
Thruster w/ Halo + Optics on HPG	85.14%	82.29%
Thruster w/ Halo + Optics on HP1	82.55%	75.93%
Thruster w/ Halo + Optics on HPG and HP1	74.76%	68.50%

This table demonstrates that the Halo with any existing peripherals causes acceptable thruster performance degradation.

4.2.3 Halo Electrical Design

The Halo electrical design includes 3 types of Printed Circuit Boards (PCBs): the Motherboard, Power board, and HP boards. The single Halo Motherboard supports the 50-pin connector to interface with the VERTIGO Avionics Stack, contains two USB hubs and an Ethernet switch, and routes wires to the proper HPs. It also contains the circuitry necessary for HPG, including the 50-pin connector. The single Halo Power board parallelizes the four Halo batteries, creates 5V and Ground lines from Batt+ and Batt-, and routes power lines to the proper HPs. In addition, the Power board contains the circuitry for HP3, including the 50-pin connector. The four HP boards take data lines and power lines from the Halo Motherboard and Power board and route them to the proper pins on the 50-pin connector they each provide to interface with peripherals. The Halo PCB locations are shown in Figure 4-29.

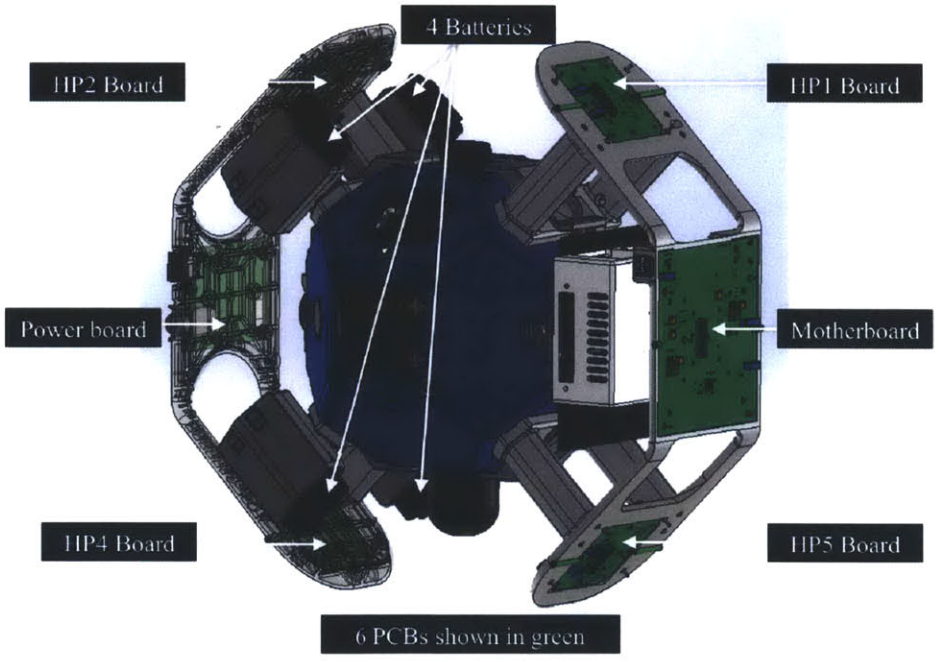


Figure 4-29: Halo PCB Locations

The block diagram in Figure 4-30 shows how the PCBs in the system interface with each other.

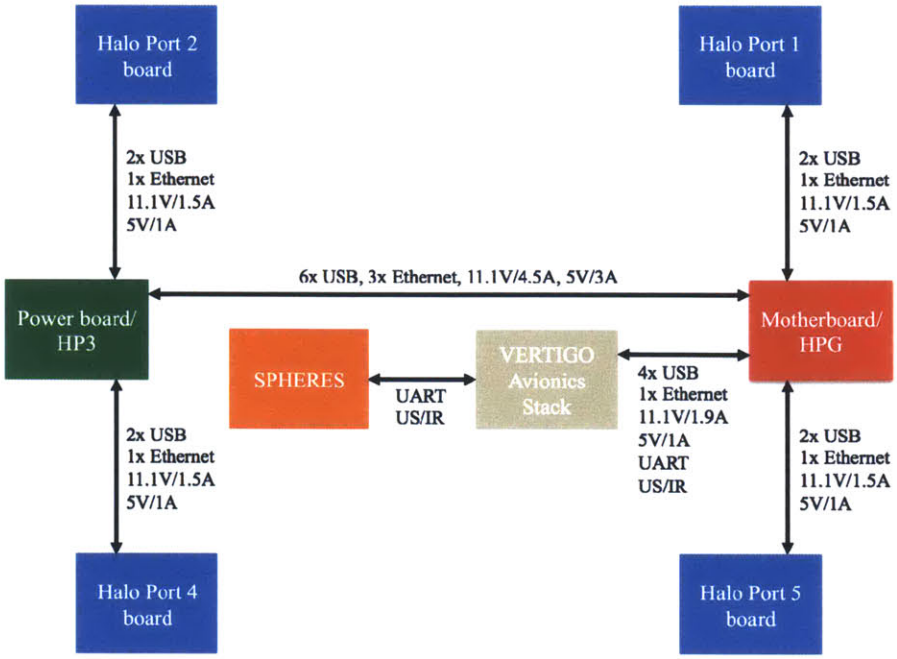


Figure 4-30: Halo PCB Block Diagram

Halo Motherboard

The block diagram of the Halo Motherboard is shown in Figure 4-31.

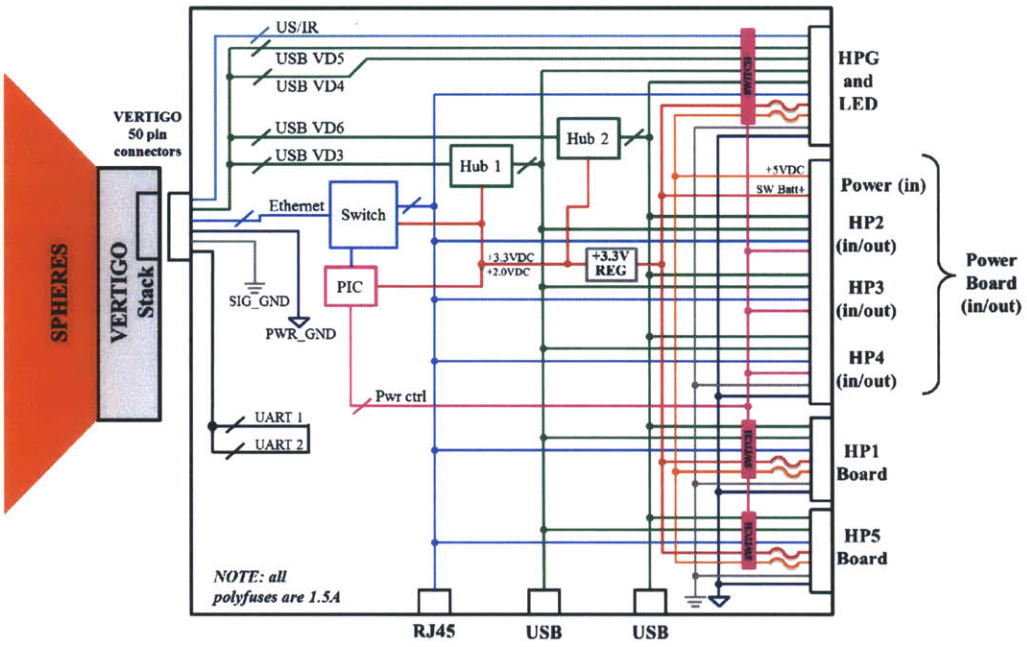


Figure 4-31: Halo Motherboard Block Diagram

The Motherboard provides the male 50-pin connector to mate with the female connector on the VERTIGO Avionics Stack. Coming out of the VERTIGO Avionics Stack are the following lines:

- 4 USB 2.0 data lines
- 1 Ethernet data line
- 1 UART data line
- 4 Ultrasound receive data lines
- 2 Infrared receive data lines
- 1 Infrared transmit line
- Batt+ power line
- Batt- power line
- 5V power line

- Ground line

More detail about the VERTIGO Avionics Stack external interface can be found in the VERTIGO Goggles ICD.

The Motherboard has voltage converters in order to supply 3.3V and 2.0V lines to the Ethernet switch, USB hubs, and PIC processor on the board. The PIC is used to send signals along power control lines to each of the ports so that ports can be powered on and off through software. There are three switches on the Motherboard and three switches on the Power board. As of the time of this writing, the PIC is Microchip Technology's PIC16F648A, which was also used inside the Docking Port.

The Motherboard has all of the circuits for HPG, due to their proximity. The Motherboard provides the LEDs for HPG.

USB

The Motherboard routes the first two of the USB lines directly to HPG. Thus, HPG has two dedicated USB lines because, in the nominal configuration, the VERTIGO Optics Mount is plugged into HPG. All other peripherals can share the other two USB lines.

The other two USB lines are each routed to a USB hub. The hubs each have eight nodes: one to the VERTIGO Avionics Stack (master), one for each of the six HPs (slaves), and one to either an external connector on the side of the Halo (slave) or the PIC (slave). One of the hubs is routed to the external connector and one is routed to the PIC. Thus, HPG has four USB lines, and HPs 1-5 each have two USB lines. Providing four USB lines to HPG leaves flexibility for a future peripheral requiring high bandwidth to utilize all four lines. Note that the two lines routed to HPs 1-5 are on the same pins as the two direct lines to HPG, which allows the VERTIGO Optics Mount, or any peripheral requiring two USB lines (such as the Docking Port), to receive the two USB lines on the same two pins no matter which HP they are attached to.

The USB hub integrated circuit (IC) is the Microchip USB2517i, which is a USB 2.0 Hi-Speed 7-Port Hub Controller (there are two in the design). Upstream ports

are 2 VERTIGO Avionics USBs, and available downstream ports are 12 Halo Ports (2 USB channels per port) and 2 External USB Connectors.

Ethernet

The Motherboard routes the single Ethernet line to an Ethernet switch, which has nine nodes: one for each of the six HPs, one to an external connector (RJ45), one to the VERTIGO Avionics Stack, and one to the PIC processor. A switch is being used rather than a hub because it allows all nodes to talk at the same time and to send and receive at the same time. The Ethernet coming out of the VERTIGO Avionics Stack is 1 Gbps.

At the time of this writing, the nine-port Ethernet switch IC is the Micrel KS8999 9-Port Ethernet Switch, which features Integral Physical Layer Transceivers (PHY) and Media Access Control units (MAC). It can operate as a standalone 8-port-switch or provide more customized operation via a PIC processor. This device supports 10/100 Mbps Ethernet, and may be upgraded to a 1 Gbps switch.

UART

The two UART lines are looped back so that UART1_RS232_TX is tied together with UART2_RS232_RX and UART2_RS232_RX goes to UART2_RS232_TX. This loop-back is identical to what the VERTIGO Optics Mount does when plugged directly into the VERTIGO Avionics Stack. The loop back simplifies the design but does not allow peripherals to use UART in the future, so some flexibility is lost. However, the Optics Mount and Docking Port do not need the UART lines. The Optics Mount does not use them and the Docking Port has a USB to Serial converter so that it simply needs the USB lines. As of April 2014, the team adding control moment gyroscopes to SPHERES is planning to use USB, and the team adding an optical range finder and a thermographic camera is planning to use Ethernet.

Ultrasound/Infrared

There are four ultrasound receive lines, two infrared receive lines, and one infrared transmit line coming out of the VERTIGO Avionics Stack that can be used to replace

the blocked ultrasound and infrared receivers and infrared transmitters on the +X face of the SPHERES. Because there are only four ultrasound receive and two infrared receive lines, only four ultrasound receivers and two infrared receivers can be replaced at any time. Since the +X Face of the SPHERES always needs them replaced, the lines are fed straight though to HPG.

Halo Power Board

The block diagram of the Halo Power board is shown in Figure 4-32.

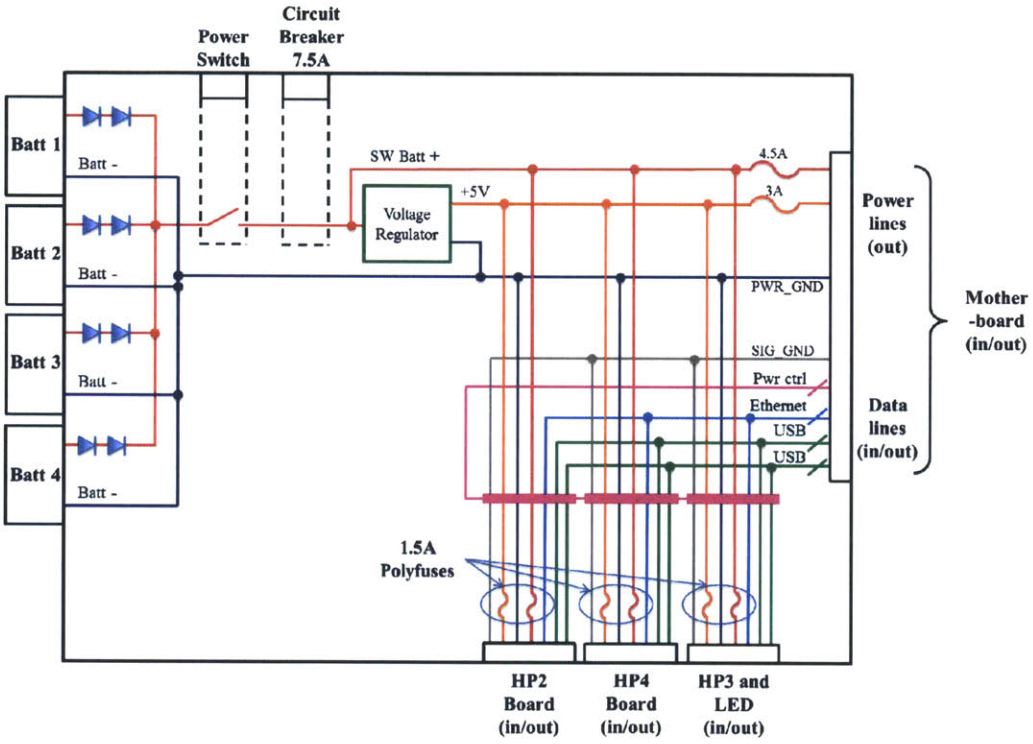


Figure 4-32: Halo Power Board Block Diagram

The Power board provides the following lines to be routed to each of the HPs:

- Batt+ power line
- Batt- power line
- 5V power line

- Ground line

These power lines are identical to the lines coming out of the VERTIGO Avionics Stack, so that any of the peripherals can be plugged into any port.

The four batteries in the Halo battery assembly are parallelized on the Power board so that the Batt+ and Batt- lines have a difference in electric potential of approximately 11.1V, unregulated, which is sent to each HP. This voltage is also fed into a voltage regulator to produce the regulated 5V and Ground lines, which are also sent to each HP. The 11.1V lines have a current limit of 1.5A, and the 5V lines have a current limit of 1A.

Power control lines, which are routed to the Power board from the Motherboard PIC, allow the power to each HP to be controlled through software. There is also a master power switch on the Batt+ lines coming off the batteries. There is a circuit breaker in series with the power switch that is rated for 7.5A. This number is slightly smaller than the maximum current that can be drawn from the four batteries, which is 1.9A x 4 batteries = 7.6A.

The Halo power subsystem—rather than the VERTIGO Avionics Stack battery—powers HPG in order to increase testing duration.

The Power board contains all of the circuits for HP3, because of their proximity. HP3 has two LEDs which are nominally green, but which turn orange if the battery voltage drops below acceptable levels. The LED color change serves as the low battery indicator.

Halo Port Boards

The block diagram of the HP boards is shown in Figure 4-33.

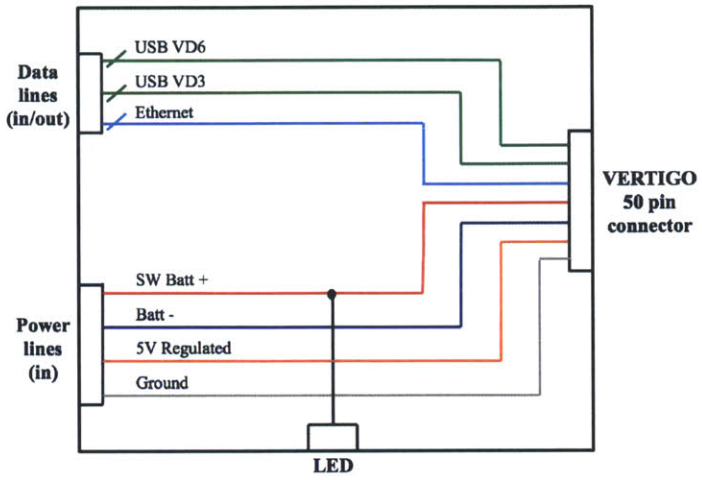


Figure 4-33: Halo Port Board Block Diagram

The HP boards combine data lines and power lines from the Motherboard and Power board into one external HP interface to peripherals. The boards support 50-pin connectors to interface with peripherals; these connectors are mechanically identical to the external connector on the VERTIGO Avionics Stack. They are also electrically identical to the connector on the VERTIGO Avionics Stack, with the exception that the UART, ultrasound, and infrared pins are unused. The HPs each have two LEDs.

Halo Battery

The selected battery for the Halo power subsystem is the Nikon 16650 Lithium Ion Battery EN-EL4a, which is currently used to power the VERTIGO Avionics Stack and is rechargeable on station. The battery is pictured in Figure 4-34.

Figure 4-34: Rechargeable Nikon Lithium Ion Battery [54]

These batteries are already on the ISS, and their small size allows them to integrate well into the Halo structure. The batteries provide sufficient current and capacity to support an intensive Concept of Operations. The Halo power subsystem has four of these batteries in parallel, giving the values shown in Table 4.6.

Table 4.6: Nikon Battery Specifications

	Voltage (V)	Capacity		Mass (kg)	Volume (cm^3)	Max Current (A)	Specific Capacity (Wh/kg)	Capacity Density (Wh/cm^3)
		(Wh)	(Ah)					
Single Battery	11.1	28	2.5	0.162	115	1.9	172.84	0.24
Power Subsystem Total	11.1	112	10	0.648	460	7.6	172.84	0.24

Halo Electrical Interfaces

All Halo interfaces are mechanically and electrically identical to the VERTIGO Avionics Stack interface. The only changes are that there are empty pins in place of the UART lines (boxed in red), and HPs 1-5 also have empty pins in place of two of the USB lines (boxed in green), as shown in Figure 33, which depicts the VERTIGO Avionics Stack 50-pin connector schematic.

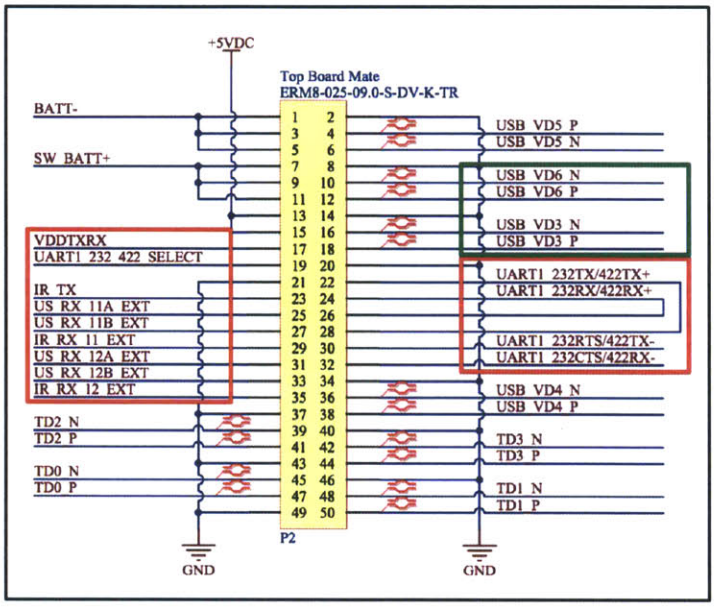


Figure 4-35: VERTIGO Avionics Stack External Interface

4.3 Halo Prototype Hardware Design

Due to scheduling pressures, the Halo Prototype was built with a slightly simplified electrical design, while still maintaining the primary functionality. For the prototype, there is no connection to HP 2 from the Power board, and the cable from the Motherboard to the Power board only has 2x USB and 1x Ethernet. The block diagram for the Halo Prototype is shown in Figure 4-36.

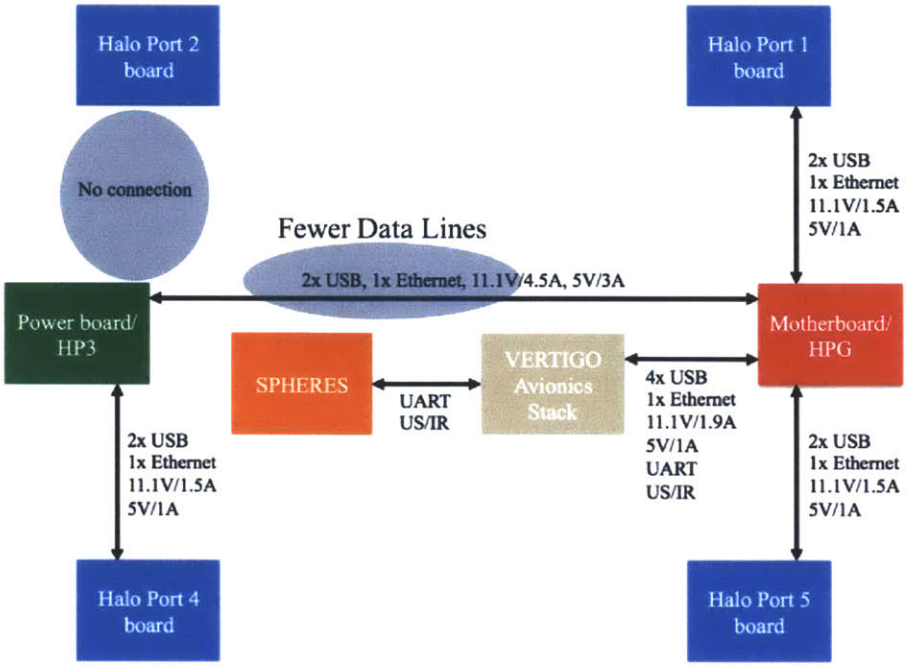


Figure 4-36: Halo Prototype PCB Block Diagram

The modified block diagram for the Prototype Motherboard is shown in Figure 4-37.

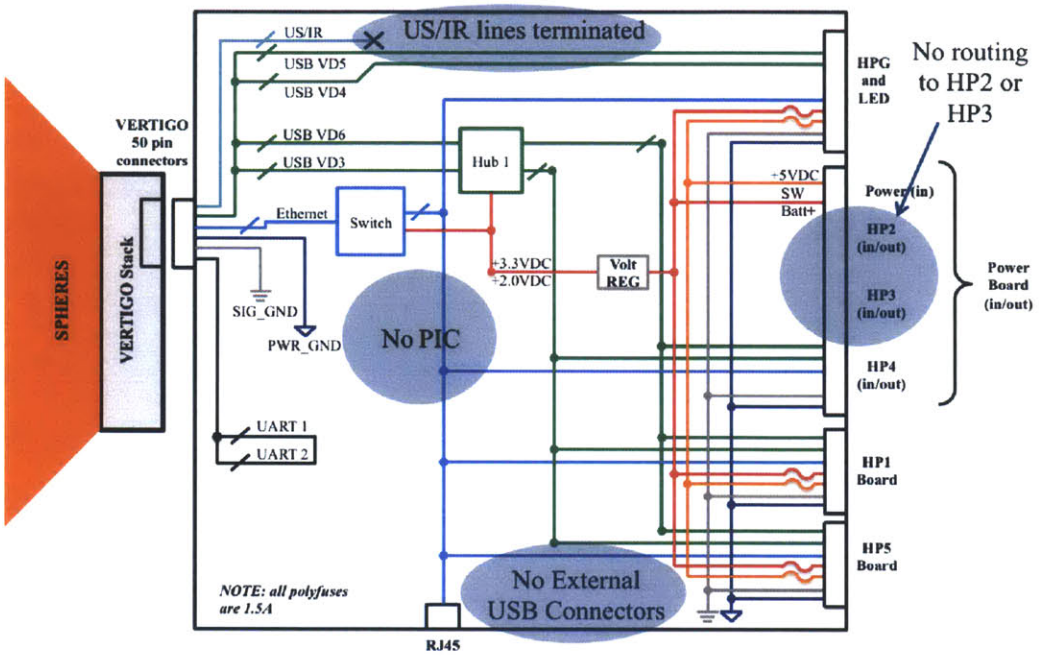


Figure 4-37: Halo Prototype Motherboard Block Diagram

For the prototype, the Motherboard has only a single USB hub and no external connectors. The Motherboard does not have a PIC processor or power control lines. The ultrasound and infrared lines are terminated rather than routed to HPG. Only 2 USB lines are routed to HPG. The Ethernet switch supports 10/100 Mbps Ethernet rather than 1 Gbps.

Figure 4-38 shows the modified block diagram for the Prototype Power board.

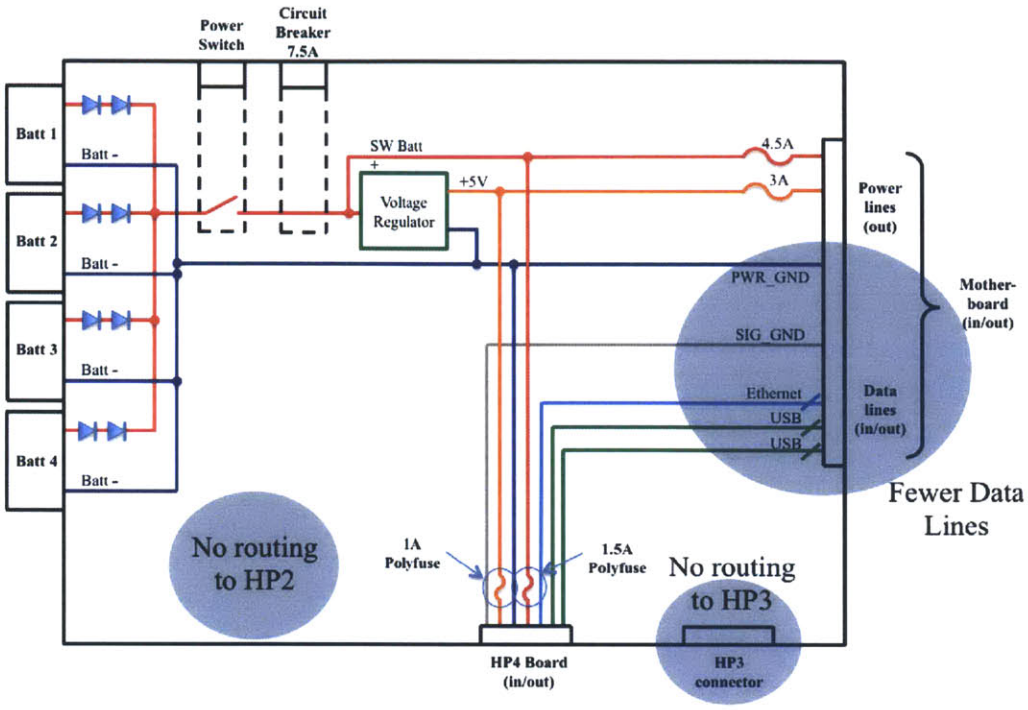


Figure 4-38: Halo Prototype Power Board Block Diagram

The Prototype Power board does not route lines to HP2 or HP3.

The Prototype Halo port boards route 10/100 Ethernet (4 pins) to the external connector rather than 1 Gbps Ethernet (8 pins).

All Prototype Halo port electrical interfaces have 10/100 Ethernet rather than 1 Gbps Ethernet.

4.4 Halo Software Design

The Halo software is centered around the VERTIGO Avionics Stack, which has a Via Pico-ITX P830 processor. This processor provides 4GB of RAM, 1 Gbps Ethernet, and USB 2.0, among other features. Two 64GB flash drives are connected to it. The computer runs a Linux operating system with a Lucid Lynx Ubuntu distribution [55]. This computer has already been used to execute machine vision algorithms on the ISS, and as such, a significant code base already exists.

The VERTIGO software was written to operate a specific set of stereo-vision cam-

eras, so part of the task of writing the Halo software included modifying the existing code to work with other peripherals. The VERTIGO code, which was written in C++, was broken down into two main parts: GogglesCore and TestProjects. GogglesCore is the portion of the code written by MIT that is responsible for operating the cameras, capturing images, storing data, communicating with the SPHERES, and several other primary functions. TestProjects is the portion of the code where a guest scientist writes his or her own machine vision algorithms, determines how to prioritize data storage, and decides how to act on that data, among other things. The guest scientist is exposed to several interface functions through which he or she can interact with GogglesCore. Thus, new Core-type software had to be created to match Halo's objectives. This software is called "HaloCore." Likewise, new TestProjects had to be created to utilize the functions in HaloCore.

Two of the main tasks in the development of HaloCore were to create new and generalized versions of the camera-specific classes in GogglesCore and to simplify the process for creating new classes, allowing for peripherals to be added to the Halo software in a more modular way. Thus far in the software development process, the generic GogglesCore functions have been moved to HaloCore, stereo-camera-specific functions were moved to a newly created OpticsMountCore, and single-camera-specific functions were created for a new DockingPortCore (because the SPHERES Docking Port only has a single camera). Much of this work was done alongside Chris Jewison, who also created some of the following diagrams. Figure 4-39 shows the breakdown of the functional responsibilities of each of the software segments.

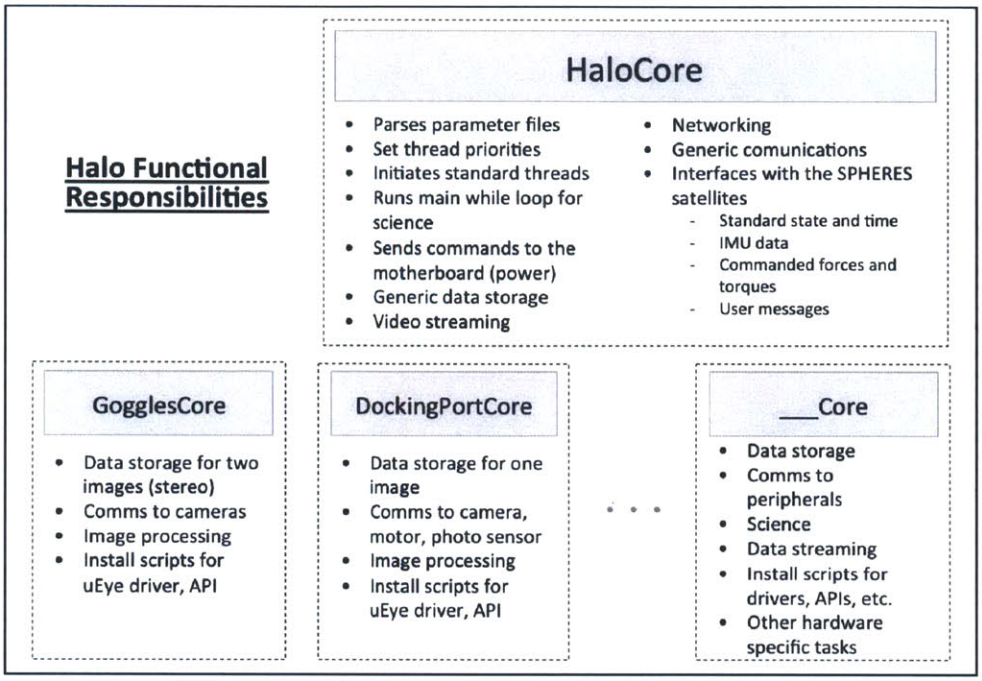


Figure 4-39: Breakdown of Functional Responsibilities within the Halo Software

Note that the "____Core" block in the lower right-hand corner is intended to signify code that will be developed as future peripherals are built and added to the testbed.

Using a similar format, Figure 4-40 shows how the functional responsibilities laid out in Figure 4-39 are divided into different code segments.

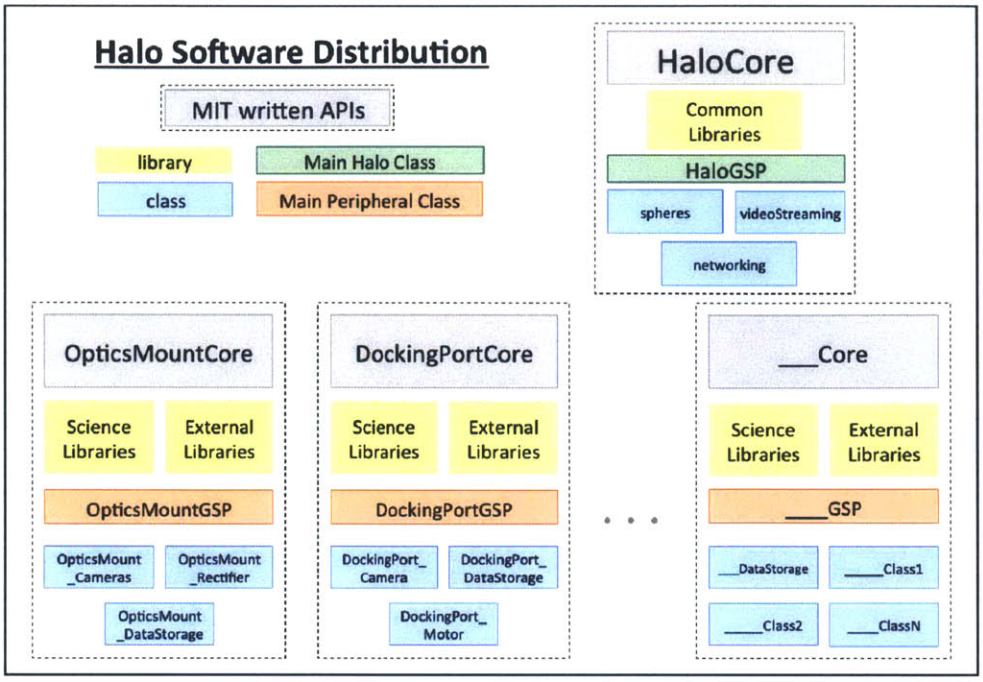


Figure 4-40: Distribution of Files within the Halo Software

As Figure 4-40 shows, the primary class within HaloCore is HaloGSP. Similarly, the primary class in OpticsMountCore is OpticsMountGSP, and the primary class in DockingPortCore is DockingPortGSP. These main classes contain all of the functions that a guest scientist calls within his or her TestProject code. For example, OpticsMountCore has functions that initialize two cameras, and DockingPortCore has functions that initialize one camera. Figure 4-41 shows the inheritance hierarchy within the Halo software.

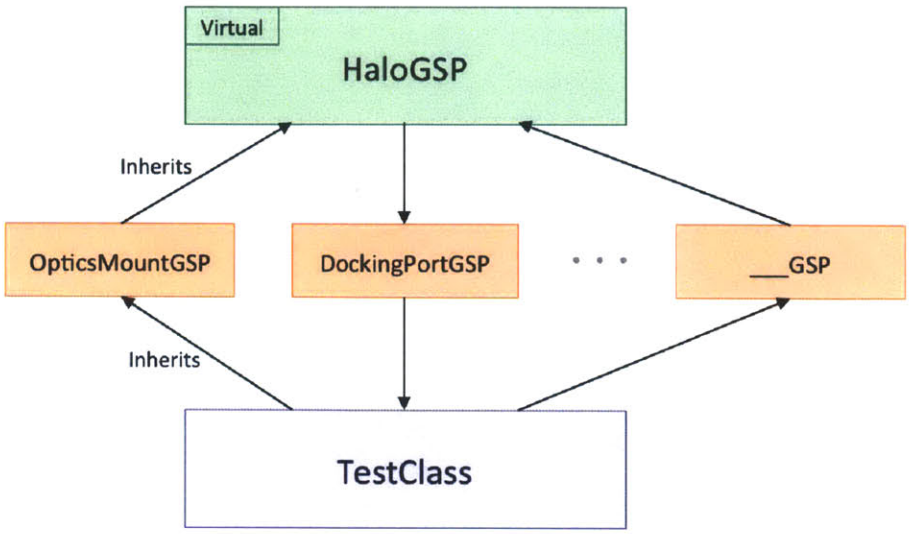


Figure 4-41: Halo Software Inheritance Hierarchy

With this hierarchy, a guest scientist creating a TestProject has the freedom to inherit the appropriate classes for whatever peripherals he or she is using, and the ability to omit classes pertaining to peripherals he or she is not using. So, for example, a guest scientist writing a test that involves only Docking Ports and no Optics Mounts inherits DockingPortGSP and not OpticsMountGSP. Additionally, regardless of which peripheral class gets inherited, the HaloGSP class is always inherited, allowing the guest scientist to use or override those generic functions.

Figure 4-42 lays out the flow of the software beginning with the "main()" function that gets called at the start of the execution of the C++ code. After the "test" object gets created, which inherits the appropriate GSP classes, the "runMain()" function is called inside HaloGSP. This function parses the parameter file and initializes the various threads. Next, the function enters a while loop which calls "GSrunMain()"—a function in which the guest scientist does all of his or her data processing, image processing, and so on. This function is called over and over until the test is ended and everything is shut down.

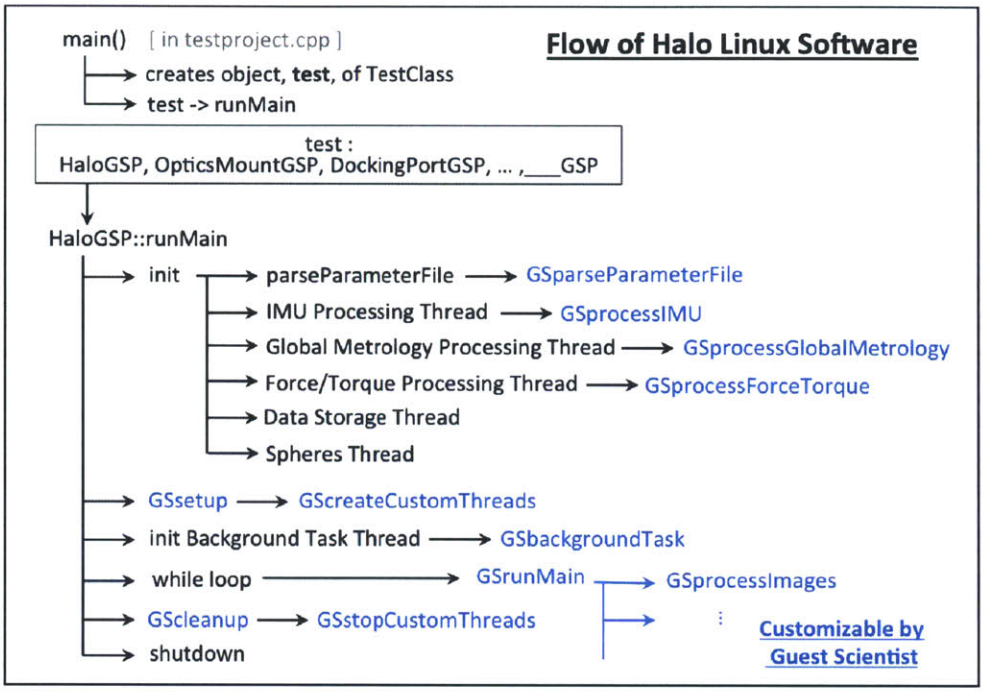


Figure 4-42: Flow of the Halo Software

Future work for the Halo software includes further restructuring of HaloCore to enable code for new peripherals to be inserted more easily. The process for creating new classes still needs to be simplified and modularized. Data storage prioritization is another issue that has not yet been dealt with. With multiple cameras attached, prioritization will certainly be an important issue to solve.

4.5 Halo Prototype Implementation and Testing

The third step in the CDIO process, following conception and design, is "Implement." Once the Halo design was solidified, the team moved into the phase of implementing the design by creating a Halo Prototype. In the implementation phase, the team at MIT worked very closely with a subcontractor called Aurora Flight Sciences to order parts and build the prototype. As noted in Section 4.3, the prototype was intended to be mechanically identical to the flight units but electrically simplified.

There are two primary ways in which the flight units will be different from the

prototype. The first is upgrading the electronics to match the initial design before the prototype simplifications. Secondly, through the process of building and testing the prototype, many lessons were learned. These lessons will be applied to all future Halo units built.

One issue that became immediately apparent when assembling the Halo was the routing of cables. The connectors on the PCBs were all arranged facing in the right directions, but they were not in optimal locations for easily routing the cables between them without having to make sharp turns. Thankfully, that will be a relatively easy change for the flight units, as it will only require changing the layouts.

The thumb screws that were selected also turned out to not be a good choice. They were not long enough to screw all the way into the Docking Port and firmly attach it to the Halo, even without the springs. The thumb screws will be extended.

The amount of room between the VERTIGO Avionics Stack and the back of HPG felt much smaller in real life than it looked in the CAD images. That led to another problem: the cable that contacted the back of the Motherboard to the VERTIGO Avionics Stack was very difficult to get in. After realizing this was an issue, the team decided that one end of the cable could be permanently attached to the motherboard through a board-to-cable connector. This will help in several ways. First, the cable will be unable to be completely separated from the Halo, and thus will not get lost. Secondly, there will be no need for the large connector head that was used to plug the cable into the Motherboard. Instead, the cable will enter through a slot on the back side of HPG, opening up even more room for the astronaut to get in there and attach the other end of the cable.

The team also realized that the prototype has no indication of how much power is in each of the batteries, which could make operations difficult. One solution is to make the LEDs on the sides of HP3 turn orange when the batteries fall below a certain voltage. The orange LEDs provide an indication to the astronauts change all four of the batteries.

The clamping system, which is used to tighten the Halo onto the SPHERES as described in Section 4.2.2, did not function as expected. Although it did tighten the

Halo sleeve, no one on the team was able to torque it hard enough to tighten the Halo to acceptable levels. With two people, it was possible to crank the system one extra click and secure the Halo onto the satellite, although it was still difficult. A proposed solution is to look into larger and more heavy-duty versions of the same clamping system. If that does not work, a more simple latching mechanism will likely be used.

The recessed channel shown in Section 4.2.2 also turned out to be too small for the cable that runs from the expansion port side of the Halo to the back side. In future versions, small walls will be added to protrude upwards from both sides of the channel so that the cable will fit. The channel will then be covered so that the cable is not exposed.

Another small change includes increasing the diameter of the holes in the sleeve that are intended to make the sleeve not block the thruster nozzles, as pictured in Figure 4-25 in Section 4.2.2. The holes in the prototype cannot tolerate much misalignment before completely impinging the thrusters.

Finally, countersunk screws will be added to the plastic struts so that they can be rigidly attached to the sleeve in a non-permanent manner (as opposed to glue).

After the prototype, five more Halos will be built that will all be flight-like. The best three will fly, and the other two, along with the prototype, will become a part of the ground testing facility.

4.6 Testbed Operations

The fourth and final step in the CDIO process is "Operate." Although no operations have been conducted at the time of this writing, this section outlines the operations plan for the Halo aboard the ISS.

4.6.1 Sample CONOPS

Figure 4-43 shows a summary of the robotic assembly architectures described in Section 3.1.1. One of the goals of the InSPIRE-II program is to test as many of these architectures as possible.

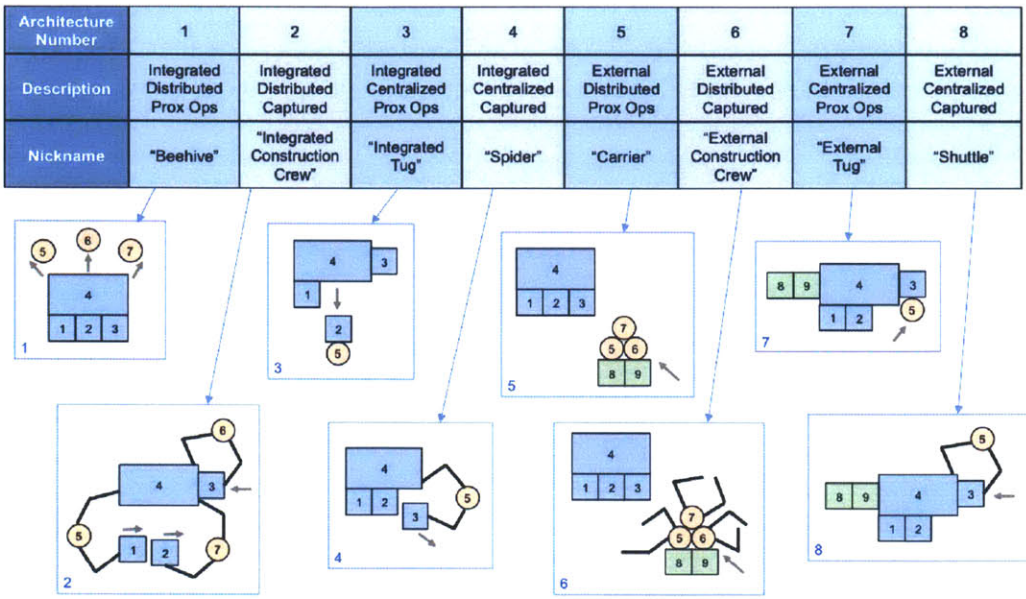


Figure 4-43: Summary of Robotic Assembly Architectures

As more hardware gets added to the SPHERES Facility on the ISS, more of these architectures are able to be tested. Figure 4-44 describes the different architectures that can be tested for each new piece of hardware the SPHERES Facility gains. Listed in the figure for each stage are the SPHERES satellites, SPHERES Docking Ports, and Halos, which have already been mentioned in this paper. In addition, some type of robotic arms and some type of proof masses, modules, or satlets also need to become part of the facility.

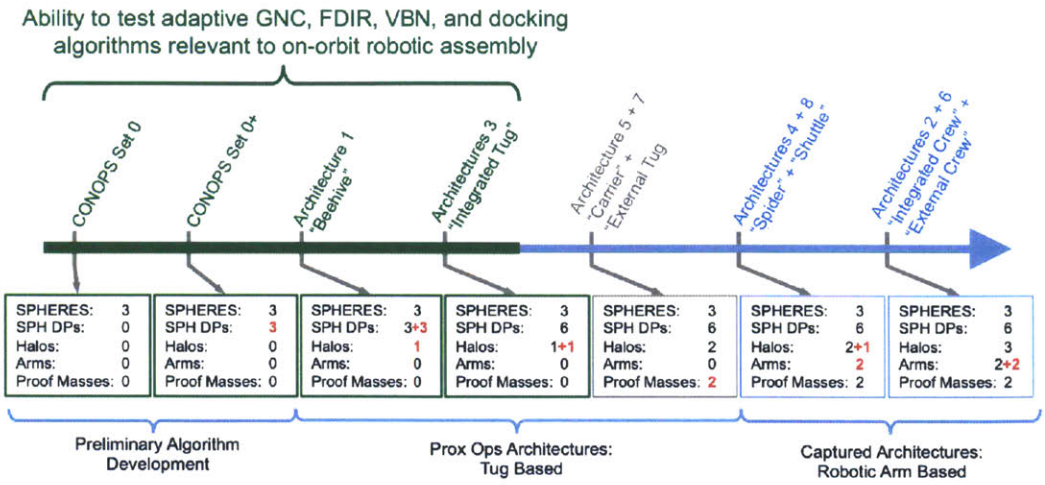


Figure 4-44: SPHERES Facility Progression on the ISS

Figure 4-44 also lists which architectures can be tested at teach stage of the facility progression. A sample CONOPS for each stage follows.

Stage One begins the facility progression with the current capabilities of the SPHERES Facility. CONOPS 0, pictured in Figure 4-45, is an example of a CONOPS that can already be conducted on the ISS without Halos or Docking Ports. It requires three SPHERES satellites. In this CONOPS, satellite #1 has VERTIGO Goggles and does not actuate itself. This could represent a satellite that has expended all of its fuel. Satellite #2 does have actuation ability, and it docks with satellite #1 using velcro. It then maneuvers satellite #1 to face satellite #3, which is spinning, in order to map and characterize it with the goggles. Satellite #3 could represent the earth or a communications satellite, for instance.

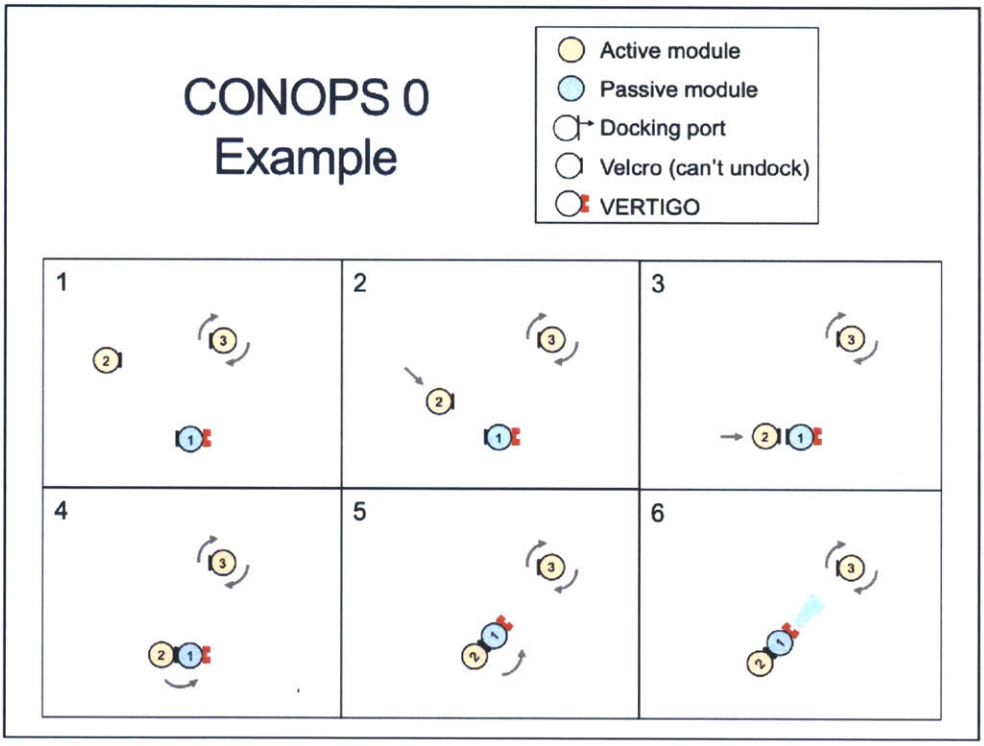


Figure 4-45: Sample CONOPS 0

Figure 4-45 shows CONOPS 0+, which can be conducted with the addition of three Docking Ports—Stage Two of the facility progression. In this CONOPS, satellite #1 docks to satellite #2, maneuvers it to attach to satellite #3 using velcro, undocks, and maneuvers around the assembly to attach at a different location (satellite #3's docking port). This CONOPS could represent an assembler satellite repositioning modules.

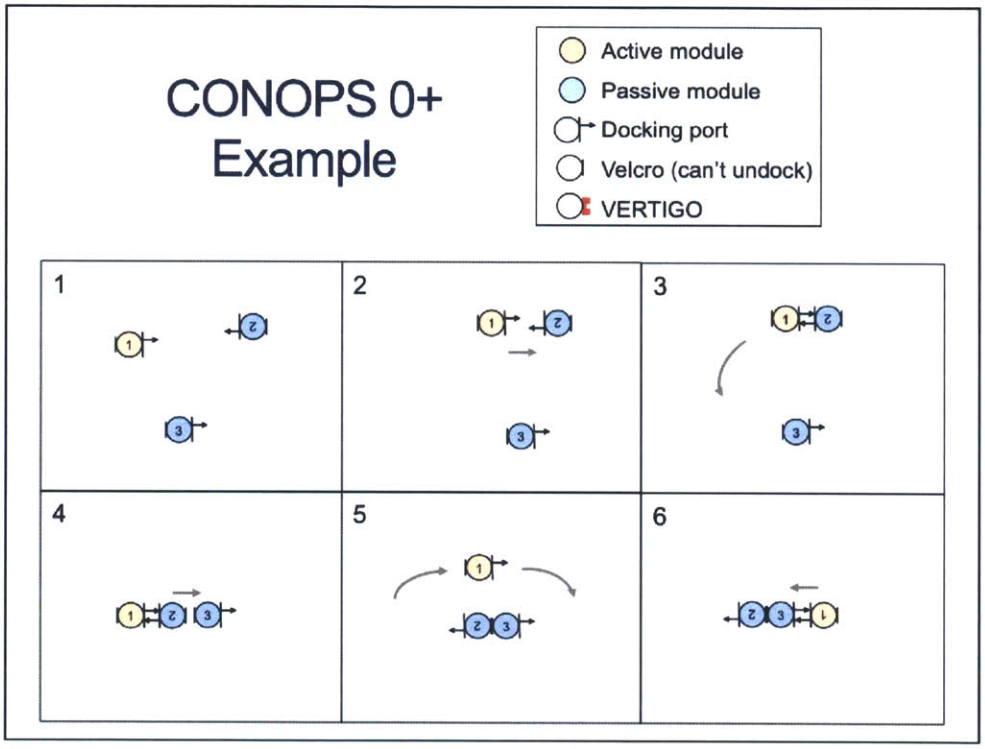


Figure 4-46: Sample CONOPS 0+

In Stage Three of the facility progression, Architecture 1, known as "Beehive," can be tested. This stage requires three satellites, six Docking Ports, and one Halo. A sample CONOPS is shown in Figure 4-47. Two satellites (#2 and #3) are attached to a passive satellite with a Halo (#1). They both undock from the Halo, use proximity operations to move to different locations on the passive module, and re-dock. This CONOPS could represent some sort of servicing mission, where two servicing vehicles need to detach from the main spacecraft and fly to its back side to service it, for example.

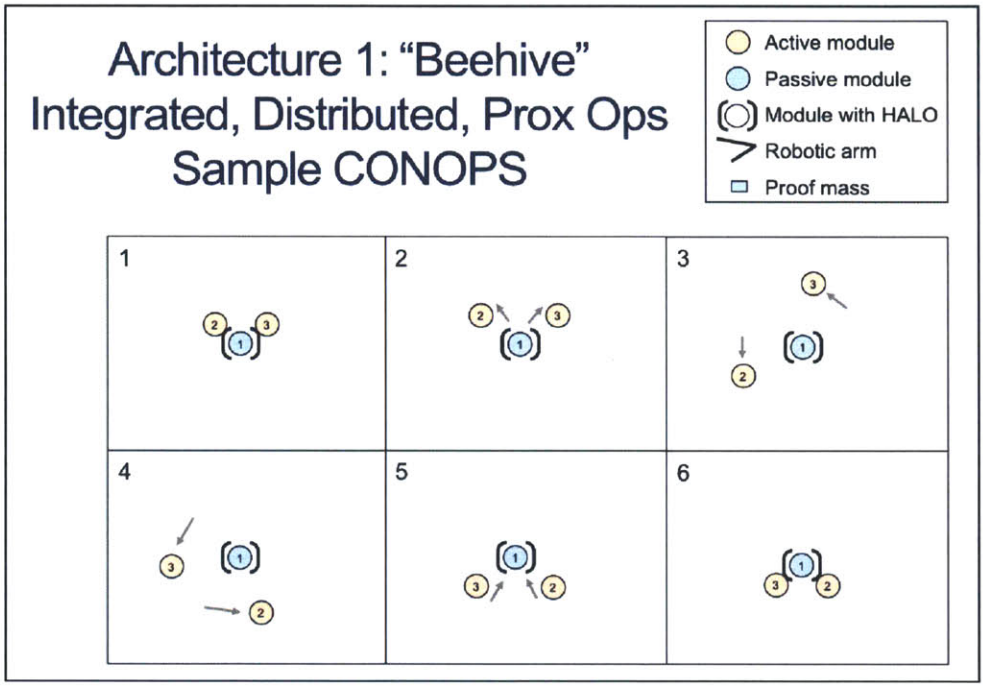


Figure 4-47: Sample Architecture 1 CONOPS

Stage Four of the facility progression requires a second Halo be brought into the mix and can enable testing of Architecture 3, "Integrated Tug." A sample CONOPS is shown in Figure 4-48, in which a single active module (satellite #3) acting as a tug relocates a passive module (satellite #2) with respect to another passive module (satellite #1) before returning to its stowed location. This CONOPS could represent a spacecraft with an integrated servicer that uses proximity operations to fly around, inspect, and manipulate the spacecraft.

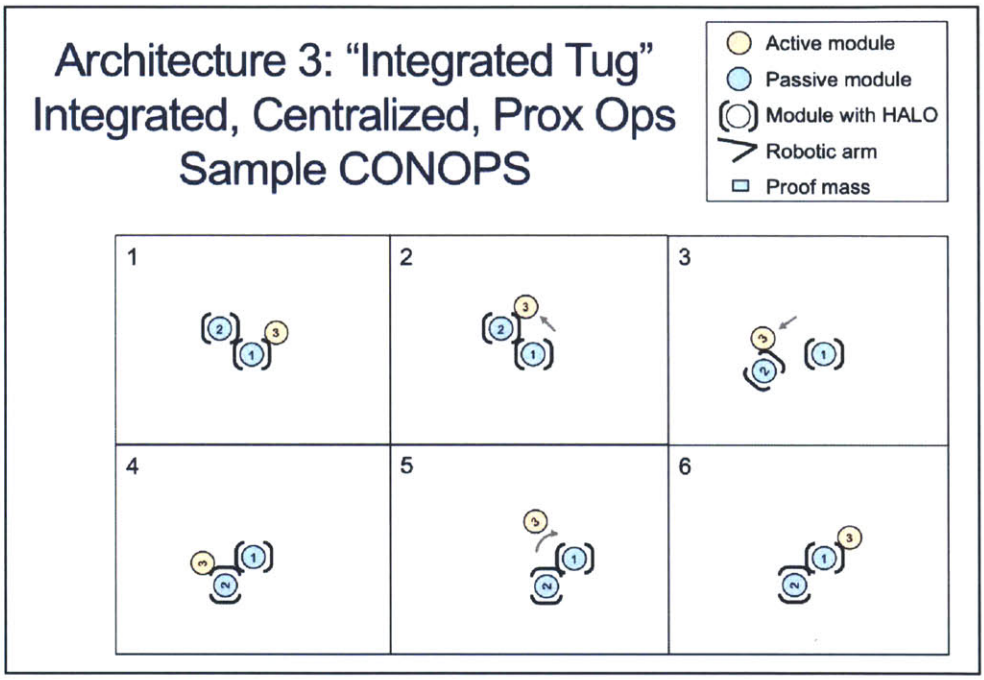


Figure 4-48: Sample Architecture 3 CONOPS

Stage Five requires the addition of two proof masses to the facility. The proof masses serve as representative modules to be manipulated by the servicer or assembler satellite. Figure 4-49 shows a CONOPS that is very similar to the one shown in Figure 4-48. The difference is that the servicer/assembler satellite (#3) is external to the rest of the system. Satellite #3 brings a module with it when it comes on the servicing mission, and after manipulating the system, departs to go execute another mission.

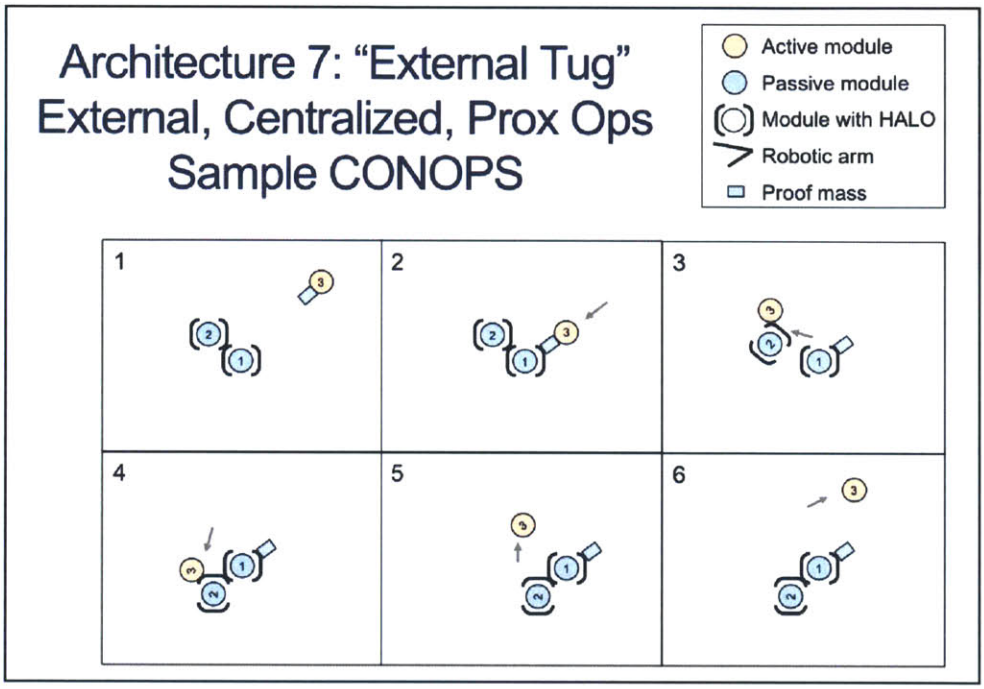


Figure 4-49: Sample Architecture 7 CONOPS

The sample CONOPS for Architecture 5, shown in Figure 4-50, depicts a servicer satellite with distributed servicing capability approaching a target satellite, repositioning several of its modules, and departing.

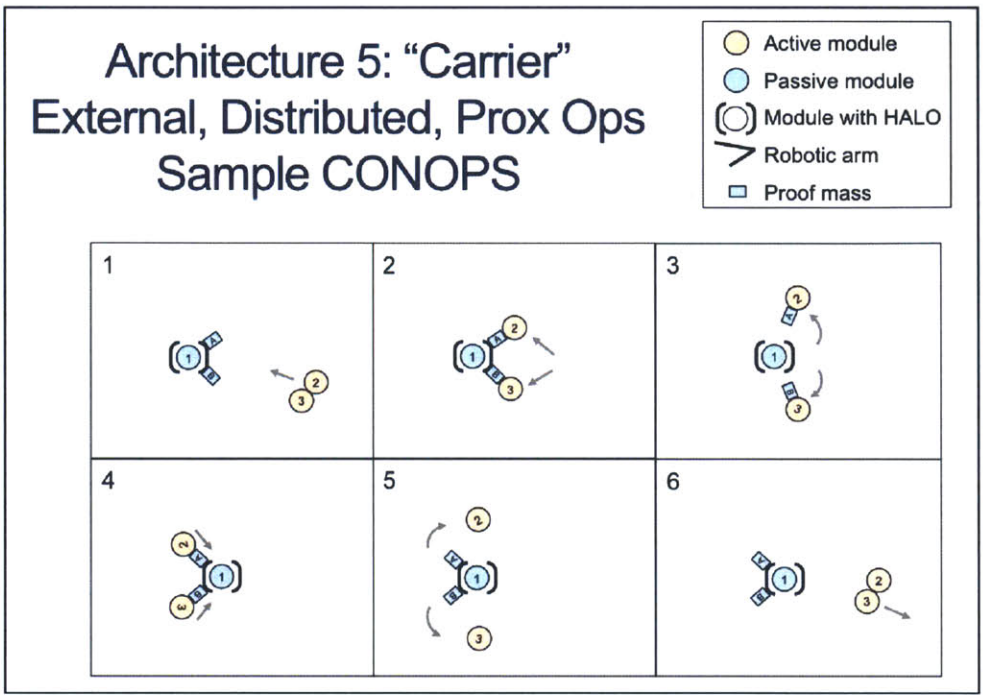


Figure 4-50: Sample Architecture 5 CONOPS

In Stage Six of the facility progression, a third Halo and robotic arms are necessary, and testing of Architecture 4, "Spider," and Architecture 8, "Shuttle," become possible. Figure 4-51 depicts a sample CONOPS for Architecture 4. In it, a servicer (satellite #3) that is initially integrated with the rest of the spacecraft crawls around the outside of the passive modules, moving satellite #2 into a different location on satellite #1's Halo, and then reattaching itself to its initial stowed location.

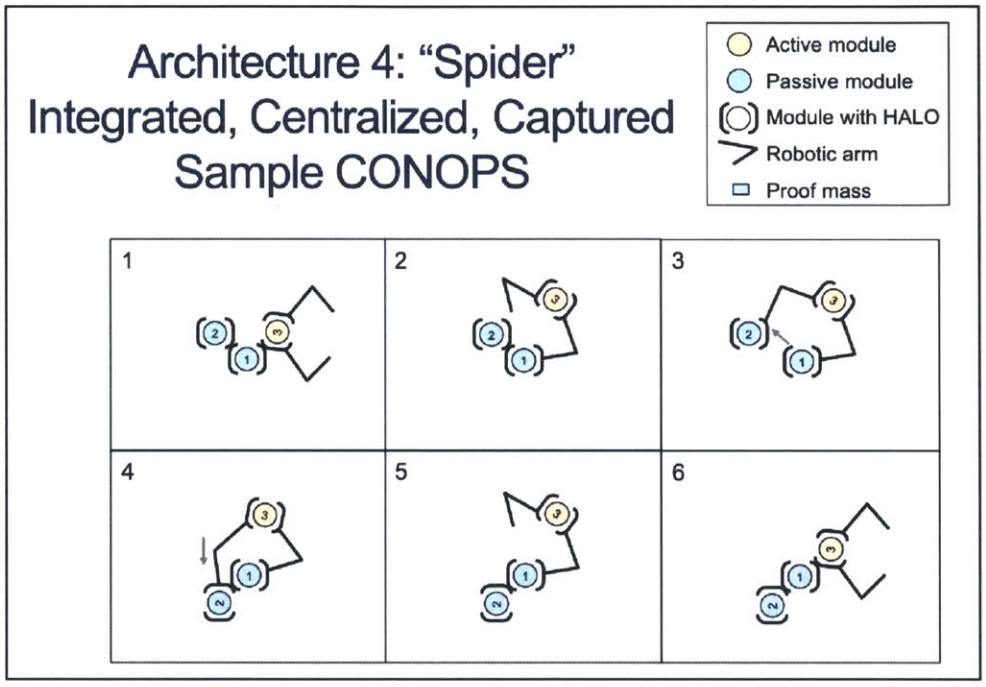


Figure 4-51: Sample Architecture 4 CONOPS

Figure 4-52 shows a CONOPS for Architecture 8 that is very similar to that of Figure 4-51. The primary difference is that the servicer satellite (#3) is external to the spacecraft, and it brings a module with it to attach to the spacecraft before manipulating other parts and departing.

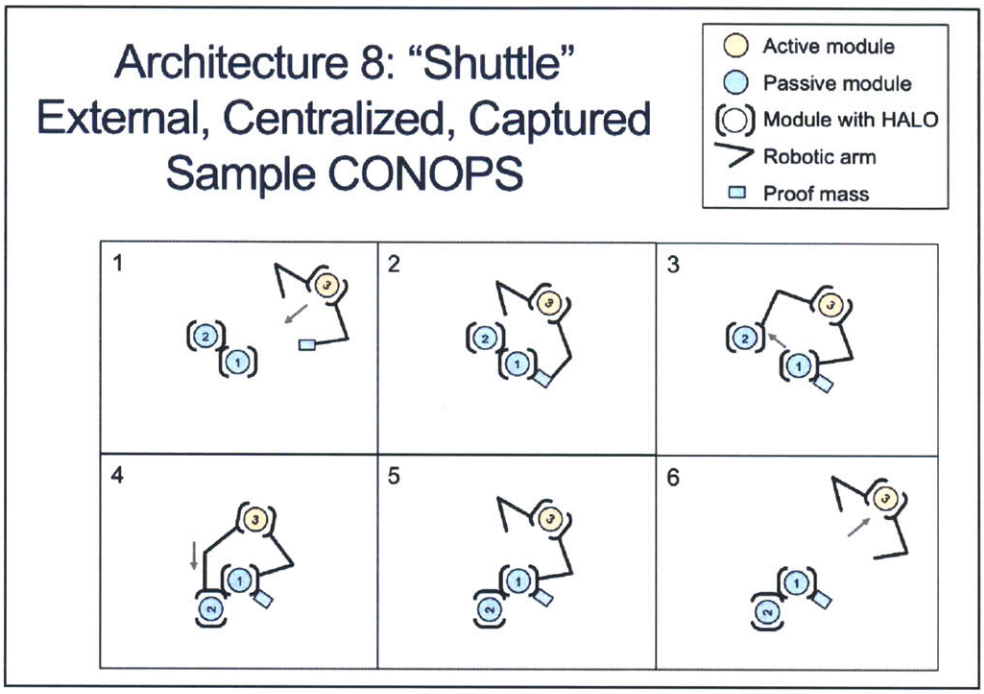


Figure 4-52: Sample Architecture 8 CONOPS

The final stage of the facility progression outlined in this section is Stage Seven, in which two more robotic arms are added. This final stage enables the testing of Architectures 2 and 6, thus increasing the facility's capability to test all architectures in the robotic assembly and servicing tradespace. Architecture 2, known as the "Integrated Crew," involves two servicer satellites integrated into the system (satellites #2 and #3). They each detach from the passive main spacecraft (represented by satellite #1), climb around the outside, and reattach on the opposite side, demonstrating the ability for captured, distributed servicers to work together successfully.

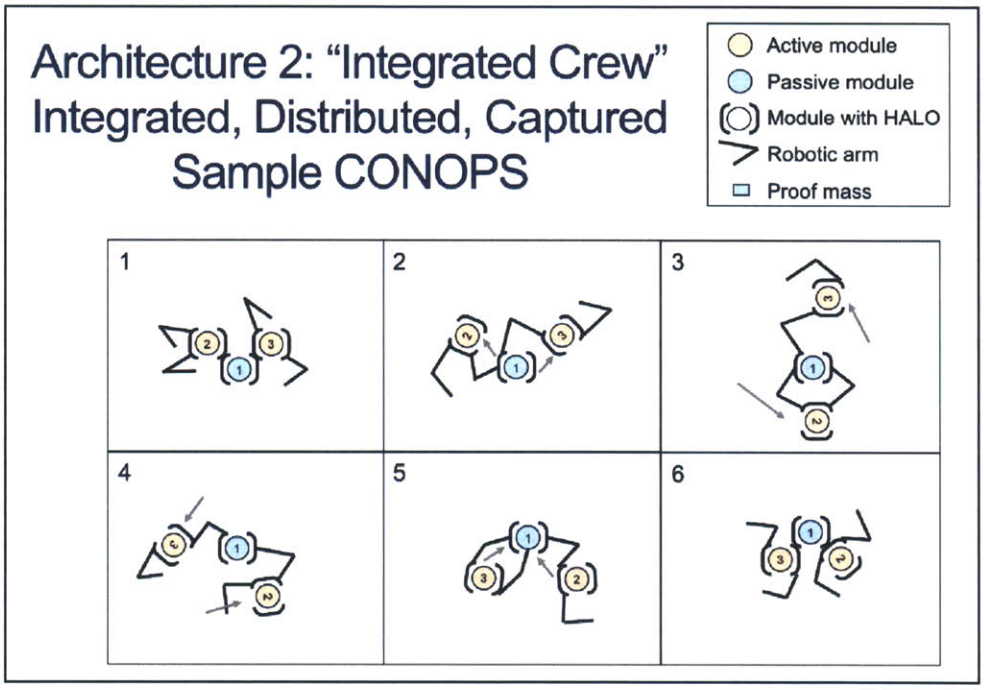


Figure 4-53: Sample Architecture 2 CONOPS

A sample CONOPS for Architecture 6, which is known as the "External Crew," is shown in Figure 4-54. In this CONOPS, an external servicing satellite (comprised of satellites #2 and #3) attaches to a target spacecraft, repositions several of its modules simultaneously, and then departs for a new mission.

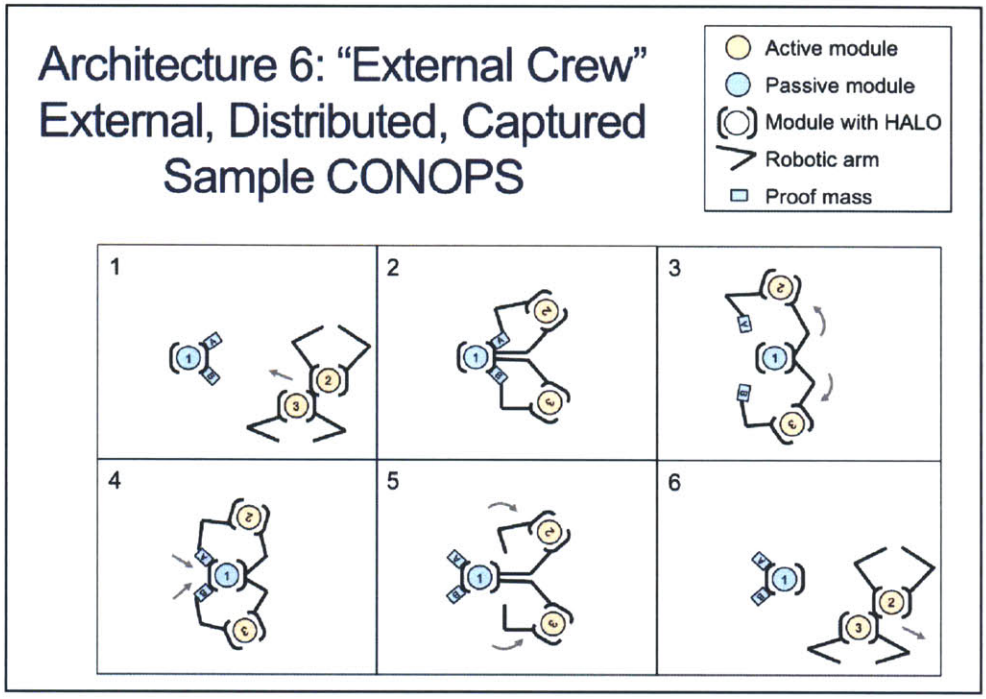


Figure 4-54: Sample Architecture 6 CONOPS

4.6.2 Test Sessions

At the time of this writing, there are four test sessions dedicated specifically to operations involving the Halo units abroad the ISS. They are known as "SPHERES Halo Checkout," "SPHERES Halo Science 1," "SPHERES Halo Science 2," and "SPHERES Halo Science 3."

Halo Test Session #1: SPHERES Halo Checkout

Estimated Time: 2-4 weeks after unstow

Description: The primary purpose of this session will be for hardware checkout and crew familiarization. In this session, the crew will turn on the Halo and test each of the Halo ports with both Docking Ports and Optics Mounts. Tests will begin

the process of Halo inertia characterization and execute simple position and attitude maneuvers. At the end of the session, the most basic science test will be conducted: two satellites (one of them Halo-equipped) will move individually, dock together, reconfigure their control systems, move together, and undock.

Halo Test Session #2: SPHERES Halo Science 1

Estimated Time: 4-8 weeks after Checkout

Description: The purpose of the first Halo science session is to begin testing different science algorithms incrementally. First, it will likely be necessary to tune the controllers, since it will only be the second time a SPHERES satellite has flown with a Halo attached to it. Initial science tests will likely include a test of the RARC algorithms described in Section 2.1, further testing with the Docking Ports and Optics Mounts, and incrementally testing the Robotic Assembly Architectures described earlier in this section, beginning with the first ("Beehive").

Halo Test Session #3: SPHERES Halo Science 2

Estimated Time: 6-10 weeks after Science 1

Description: The second Halo science session will be used to advance the algorithms. In an incremental fashion, the more complex Robotic Assembly Architectures 3 and 7 ("Integrated Tug" and "External Tug") are planned to be tested.

Halo Test Session #4: SPHERES Halo Science 3

Estimated Time: 6-10 weeks after Science 2

Description: The third Halo science session will be another opportunity for incremental advancements in algorithms. Testing will also likely expand to include CONOPS with resource aggregation, reconfigurable control, vision-based navigation and mapping, and robotic assembly of modules.

Test Session Timelines

Halo Test Sessions are expected to take a total of six hours of crew time. Halo Test Session setup time will progress according to the schedule in Table 4.7.

Table 4.7: Halo Test Session Setup Timeline

Step	Source	Astronaut Time
1.0 Charge 10 Batteries	Existing ISS Camera Charging Procedures	15 min
2.0 Upload Code and Disk Image to ISS	VERTIGO Procedures	0 min (Behind the scenes)
3.0 Check Flash Disk Consistency	VERTIGO Procedures	40 min
4.0 SPHERES Work Area Setup	SPHERES Proc: 1.001	45 min
5.0 VA Stack and Halo Attachment	VERTIGO Procedures and New Halo Procedures	30 min
6.0 Load SPHERES and VA Stack programs	SPHERES and VERTIGO Procedures	20 min

The total time it will take the crew to complete Halo setup is approximately 2.5 hours.

Halo Test Session experiment time will go according to the schedule laid out in Table 4.8.

Table 4.8: Halo Test Session Experiment Timeline

Step	Source	Astronaut Time
1.0 Power on SPHERES, VA Stack, and Halo	SPHERES Procedures, VERTIGO Procedures, New Halo Procedures	30 min
2.0 Quick Checkout SPHERES	Embedded as Test	10 min
3.0 Quick Checkout Halo	New Halo Test	20 min
4.0 Run Tests	2.001: 2 & 3	50 min
5.0 Change SPHERES Consumables (C02 Tanks and SPHERES Batteries)	2.002	20 min
6.0 Change VA Stack and Halo Consumables (Batteries)	VERTIGO Procedures and New Halo Procedures	25 min
7.0 Download Data from VA Stack to Laptop	VERTIGO Procedures	35 min
8.0 Shutdown experiment	SPHERES Procedures, VERTIGO Procedures, New Halo Procedures	20 min

The total time it will take the crew to complete Halo experimentation is approximately 3.5 hours.

4.7 Halo Development Wrap-up

This chapter stepped through the conception, detailed design, and implementation of the Halo system. It also described the expected operations involving the Halo aboard

the ISS. Future Halo development work includes the following:

- Implement all of the changes based on the lessons learned from the Halo prototype
- Finalize the Halo flight design and build five flight-like units
- Fully test the functionality of the flight units and select three to go to the ISS
- Continue developing the HaloCore software and write the tests for the ISS test sessions
- Execute the four test sessions

These actions are expected to be fulfilled within the remaining time and budget of the InSPIRE-II contract.

Chapter 5

Conclusions

This thesis began by motivating the need for the development of robotic assembly and servicing technologies. The literature review provided a summary of past, present, and future missions related to robotic assembly and servicing, and it laid out the different types of ground-based and space-based testbeds that have been and will be used to buy-down risk for such missions. Due to the importance of testing for such expensive, complex, and risky missions, and due to the limitations of ground-based testbeds and the general lack of testbeds in space, the literature review concluded by arguing for a new space-based testbed that could test a variety of important and related technologies in a low-risk and cost-effective manner. The SPHERES Facility aboard the ISS was then proposed as an ideal candidate for such a testbed.

The thesis then described several of the important control concepts for robotic assembly and servicing in space (i.e., reconfigurable control and adaptive control), and it described two experiments that were conducted using the SPHERES Ground Facility. The ground testing ultimately showed that there is promise for the proposed control methods, yet the limitations of the ground testbed leave one desiring the ability to test those concepts and others in a dynamically authentic environment. There is a significant amount of work remaining in the development and advancement of these control methods, and a space-based testbed would provide the necessary capabilities to conduct such research.

The thesis described the process of developing appropriate requirements for a

space-based robotic assembly and servicing testbed, beginning with the testbed objectives and drawing out important and necessary features. It was determined that by upgrading the SPHERES Facility on the ISS, a testbed could be developed which met the requirements. Upgrades included adding hardware that would allow each satellite to interface with multiple devices and multiple other satellites simultaneously, and requirements for that hardware then flowed from the testbed objectives and requirements.

Another significant contribution of this thesis was the conception, design, and implementation of that piece of hardware, as well as the proposal of future operations on the ISS that will utilize it. This thesis covered the mechanical, electrical, and software developments that were necessary to make the hardware into a tool that will successfully increase the capabilities of the SPHERES Facility in the necessary ways. Ultimately, the contributions of this thesis will serve to advance algorithms and buy down risk for important space-based robotic assembly and servicing missions in the future.

Future work with regard to satellite robotic assembly and servicing includes developing the necessary hardware components and software algorithms required to execute intricate tasks and complex maneuvers in space. Software development must include the determination of optimal robotic assembly and servicing architectures for a variety of missions. Such developments can either be done through a series of expensive missions or through iterative and incremental testing using an appropriate platform. These advancements will lower the risk of failure for future missions, increase investor confidence in such missions, and improve the robustness of related technologies, consequently opening the door to important space capabilities.

Bibliography

[1] Steven B. Skaar and Carl F. Ruoff. *Teleoperation and Robotics in Space.* AIAA, January 1994.

[2] G. Visentin and D. L. Brown. Robotics for geostationary satellite servicing. *Robotics and Autonomous Systems*, 23(2):45–51, March 1998.

[3] K. Landzettel, C. Preusche, A. Albu-Schaffer, D. Reintsema, D. Rebele, and G. Hirzinger. Robotic on-orbit servicing - DLR's experience and perspective. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4587–4594, October 2006.

[4] David Barnhart and Brook Sullivan. Economics of repurposing in situ retired spacecraft components. In *AIAA SPACE 2012 Conference & Exposition.* American Institute of Aeronautics and Astronautics, 2012.

[5] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. ROTEX-the first remotely controlled robot in space. In *IEEE International Conference on Robotics and Automation, 1994. Proceedings*, pages 2604–2611 vol.3, May 1994.

[6] Douglas Zimpfer, Peter Kachmar, and Seamus Tuohy. Autonomous Rendezvous, Capture and In-Space Assembly: Past, Present and Future. In *1st Space Exploration Conference: Continuing the Voyage of Discovery.* American Institute of Aeronautics and Astronautics, 2005.

[7] Charles Lillie. On-orbit servicing for future space observatories. In *Space 2005.* American Institute of Aeronautics and Astronautics, 2005.

[8] Dan King. Saving hubble. In *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law.* American Institute of Aeronautics and Astronautics, 2004.

[9] Jiegao Wang, R. Mukherji, M. Ficocelli, A. Ogilvie, M. Liu, and C. Rice. Modeling and simulation of robotic system for servicing hubble space telescope. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1026–1031, October 2006.

[10] Dan King. SPACE SERVICING: PAST, PRESENT AND FUTURE. In *6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001*, St-Hubert, Quebec, Canada, June 2001.

[11] Laryssa Patten, Lindsay Evans, Layi Oshinowo, Marius Ochisor, Nara Kazuharu, Aris Lodewijk, and Ed Tabarah. International space station robotics: A comparative study of ERA, JEMRMS and MSS. ESTEC, Noordwijk, The Netherlands, November 2002.

[12] Robotic refueling mission. National Aeronautics and Space Administration Website.

[13] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber. On-orbit servicing missions: Challenges and solutions for spacecraft operations. Huntsville, Alabama, April 2010.

[14] Bin Liang, Cheng Li, Lijun Xue, and Wenyi Qiang. A Chinese small intelligent space robotic system for on-orbit servicing. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4602–4607, October 2006.

[15] Lijun Xue, Wenyi Qiang, Bin Liang, and Cheng Li. A robotic testbed for positioning and attitude accuracy test of space manipulator. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5085–5090, October 2006.

[16] David Barnhart, Lisa Hill, Margaret Turnbull, and Peter Will. Changing satellite morphology through cellularization. Pasadena, California, September 2012.

[17] W. Whittaker, P. Staritz, R. Ambrose, B. Kennedy, S. Fredrickson, J. Parrish, and C. Urmson. Robotic assembly of space solar-power facilities. *Journal of Aerospace Engineering*, 14(2):59–64, 2001.

[18] Edward LeMaster, David Schaechter, and Connie Carrington. Experimental demonstration of technologies for autonomous on-orbit robotic assembly. San Jose, California, September 2006.

[19] Marcello Romano and Jason Hall. A test bed for proximity navigation and control of spacecraft for on-orbit assembly and reconfiguration. San Jose, California, September 2006.

[20] Peggy Boning, Masahiro Ono, Tatsuro Nohara, and Steven Dubowsky. An experimental study of the control of space robot teams assembling large flexible space structures.

[21] C. Toglia, D. Kettler, F. Kennedy, and S. Dubowsky. A study of cooperative control of self-assembling robots in space with experimental validation. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pages 3031–3036, May 2009.

[22] J. Everist, K. Mogharei, H. Suri, N. Ranasinghe, B. Khoshnevis, P. Will, and Wei-Min Shen. A system for in-space assembly. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings*, volume 3, pages 2356–2361 vol.3, September 2004.

[23] Philip Ball. Puckish robots pull together, May 2004.

[24] Wei-Min Shen, P. Will, and B. Khoshnevis. Self-assembly in space via self-reconfigurable robots. In *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03*, volume 2, pages 2516–2521 vol.2, September 2003.

[25] Jana Schwartz, Mason Peck, and Christopher Hall. Historical review of air-bearing spacecraft simulators. *Journal of Guidance, Control, and Dynamics*, 26(4), August 2003.

[26] Carlo Menon, S. Busolo, S. Cocuzza, A. Aboudan, A. Bulgarelli, C. Bettanini, M. Marchesi, and F. Angrilli. Issues and solutions for testing free-flying robots. *Acta Astronautica*, 60(12):957–965, June 2007.

[27] Jacob Katz. *Estimation and Control of Flexible Space Structures for Autonomous On-Orbit Assembly.* Masters, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[28] Enrico Sabelli, David Akin, and Craig Carignan. Selecting impedance parameters for the ranger 8-DOF dexterous space manipulator. Rohnert Park, California, May 2007.

[29] Jill McGuire and Brian Roberts. Hubble robotic servicing and de-orbit mission: Risk reduction and mitigation. Long Beach, California, September 2007.

[30] P.J. Staritz, S. Skaff, C. Urmson, and W. Whittaker. Skyworker: A robot for assembly, inspection and maintenance of large scale orbital facilities. In *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA*, volume 4, pages 4180–4185 vol.4, 2001.

[31] Skyworker. National Aeronautics and Space Administration Website - The Robotics Institute.

[32] Erik Komendera. NASA - autonomous assembly of structures in space. Space Technology Mission Directorate, STMD, Space Technology Research Grants.

[33] W. Doggett. Robotic assembly of truss structures for space systems and future research plans. In *IEEE Aerospace Conference Proceedings, 2002*, volume 7, pages 7–3589–7–3598 vol.7, 2002.

[34] Lennon Rodgers. *Concepts and Technology Development for the Autonomous Assembly and Reconfiguration of Modular Space Systems.* Masters, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[35] Swati Mohan. *Reconfiguration Methods for On-orbit Servicing, Assembly, and Operations with Application to Space Telescopes.* Masters, Massachusetts Institute of Technology, Cambridge, MA, May 2007.

[36] Thomas Debus and Sean Dougherty. Overview and performance of the front-end robotics enabling near-term demonstration (FREND) robotic arm. Seattle, Washington, April 2009.

[37] B.E. Kelm, J.A. Angielski, S.T. Butcher, N.G. Creamer, K.A. Harris, C.G. Henshaw, J.A. Lennon, W.E. Purdy, F.A. Tasker, W.S. Vincent, and B.P. Whalen. FREND: pushing the envelope of space robotics, 2008.

[38] Satellite servicing capabilities office. National Aeronautics and Space Administration Website.

[39] Austin Probe and John Junkins. Robotic simulation experiments demonstrating docking proximity operations and contact dynamics. National Harbor, Maryland, January 2014.

[40] Steve Ulrich. Spacecraft robotics and control laboratory. Carleton University Website.

[41] G. Colombina, F. Didot, G. Magnani, and A. Rusconi. Automation and robotics technology testbed for external servicing. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94*, volume 3, pages 1954–1963 vol.3, September 1994.

[42] L. Pedersen, D. Kortenkamp, D. Wettergreen, and I. Nourbakhsh. A survey of space robotics. 2003.

[43] M.A. Diftler, J.S. Mehling, M.E. Abdallah, N.A. Radford, L.B. Bridgwater, A. M. Sanders, R. S. Askew, D.M. Linn, J.D. Yamokoski, F.A. Permenter, B.K. Hargrave, R. Piatt, R.T. Savely, and R.O. Ambrose. Robonaut 2 - the first humanoid robot in space. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2178–2183, May 2011.

[44] Harley Thronson, David Akin, John Grunsfeld, and Daniel Lester. The evolution and promise of robotic in-space servicing. Pasadena, California, September 2009.

[45] Christopher Jewison. Halo system requirements document. MIT Internal Report, October 2013.

[46] Alvar Saenz-Otero. *The SPHERES Satellite Formation Flight Testbed: Design and Initial Control.* Masters, Massachusetts Institute of Technology, Cambridge, MA, August 2000.

[47] ISS: SPHERES - eoPortal directory - satellite missions. Sharing Earth Observation Resources Website, 2002.

[48] Christopher M. Jewison, Bryan McCarthy, David C. Sternberg, Daniel Strawser, and Cheng Fang. Resource aggregated reconfigurable control and risk-allocative path planning for on-orbit servicing and assembly of satellites. In *AIAA Guidance, Navigation, and Control Conference.* American Institute of Aeronautics and Astronautics, National Harbor, Maryland, January 2014.

[49] Steve Ulrich, Dustin Hayhurst, Alvar Saenz-Otero, and David Miller. Simple adaptive control for spacecraft proximity operations. In *AIAA Guidance, Navigation, and Control Conference.* American Institute of Aeronautics and Astronautics, National Harbor, Maryland, January 2014.

[50] David Sternberg and Bryan McCarthy. MIT project phoenix testing summary. MIT Internal Report, August 2013.

[51] Worldwide CDIO initiative. Conceive Design Implement Operate Website.

[52] Peter Davison, Dustin Hayhurst, Christopher Jewison, Bryan McCarthy, Brandon Karlow, and Jared Rize. Project MEDUSA final paper. Technical report, Massachusetts Institute of Technology, Cambridge, MA, December 2012.

[53] Benoit Cushman-Roisin. Chapter 9: Turbulent jets. In *Environmental Fluid Mechanics.* John Wiley & Sons, Inc., New York, March 2014.

[54] Nikon battery pack EN-EL4a for d3 Series/MB-D10. Hitchee Website.

[55] Brent Tweddle. *Computer Vision-Based Localization and Mapping of an Unknown, Uncooperative and Spinning Target for Spacecraft Proximity Operations.* PhD, Massachusetts Institute of Technology, Cambridge, MA, September 2013.