Development of an Incremental and Iterative Risk Reduction Facility for Robotic Servicing and Assembly Missions

by

David Charles Sternberg

S.B. Aerospace Engineering (Course 16-1)
Massachusetts Institute of Technology (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

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

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

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

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

Development of an Incremental and Iterative Risk Reduction Facility for Robotic Servicing and Assembly Missions

by

David Charles Sternberg

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

A means for reducing the risk for an on-orbit robotic servicing and assembly mission through the development of a series of testbeds that build successively upon one another is investigated. Robotic Servicing and Assembly (RSA) missions are believed to enable life extension programs for existing spacecraft while also enabling much larger and more complex satellites to be developed through on-orbit construction. Unfortunately, many of the new and innovative technologies required for RSA to be economically and technically feasible are still in their formative development stages. Consequently, such RSA missions are highly risk prone.

This thesis investigates the development of an incremental and iterative testing facility which can be used to reduce these RSA risks by conducting demonstration testing in authentic operational environments while leveraging existing infrastructures to reduce the costs associated with testing. The Defense Advanced Research Project Agency's (DARPA) Phoenix project, a satellite repurposing mission, serves as an example of a full-scale flight mission requiring risk-reduction testing. The thesis presents research that shows how the newly developed testing facility, which expands on the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) facility, can reduce the risk of many technologies required for Phoenix. In particular, testing is discussed and analyzed for the risk reduction of resource aggregation and physical reconfiguration technologies.

This testing is both incremental and iterative in nature as part of two ground test programs and a flight program aboard the International Space Station. The testing progression matures these technologies from base principles tested in the ground environment at the MIT Space Systems Laboratory to the planned implementation aboard the International Space Station prior to the final flight mission. The newly developed testing facility is small in scale as compared to the final RSA flight satellites, so newly developed scaling laws are presented. This process relies on the scaling of testbed results using the combined application of hybrid scaling laws and nondimensional parameters. In doing so, the results from the new testing facility can be applied to the Phoenix mission to raise the probability of mission success.

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

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

Acknowledgments

There are innumerable individuals and groups who have helped make this thesis possible. First and foremost, I am forever indebted to my family and grandparents for their ever-constant support in all of my endeavors. You have shaped my education and career in ways that are just beginning to be comprehended, and I will always love you. You have done so many things for me, and I hope that this thesis is another way for me to show my appreciation. My friends have also stuck with me through both the good and bad times. While many undergraduate peers dispersed after graduation throughout the country, Angelica Ceniceros, Jesus Zuniga, Gina Noh, Zoe Rogers, and Tess Smidt have remained unwavering friends. I am especially grateful for the continued friendship of Pearle Lipinski through my graduate experience thus far at MIT. Of course, I will never forget the other members of the SPHERES team, particularly those of the original AMP team, the Wolf Pack: Chris Jewison and Bryan McCarthy. I eagerly look forward to working with Chris, the other SPHERES team members, and the rest of the SSL in the coming years as a doctoral student. Lastly, I would like to say that I am deeply honored to work with Dr. Alvar Saenz-Otero and Prof. David Miller. Their seemingly limitless knowledge and instructional capacity has paved the way for me to transition from freshman UROP to graduate student on the SPHERES team.

The work performed and detailed in this thesis was financially supported by several organizations, and I am deeply grateful for their support. The National Aeronautics and Space Administration (NASA) and the United States Air Force Space and Missile Systems Center (SMC) supported the Agile Reconfigurable Modules with Autonomous Docking for Assembly and Servicing (ARMADAS) project under NASA Contract #NNH11CC25C. Additionally, Aurora Flight Sciences (AFS), NASA, and the Defense Advanced Research Projects Agency (DARPA) supported the InSPIRE II program though NASA Contract #NNH11CC26C and the Phoenix projects under DARPA-BAA-13-12.

1 Table of Contents

List of Figures

List of Tables

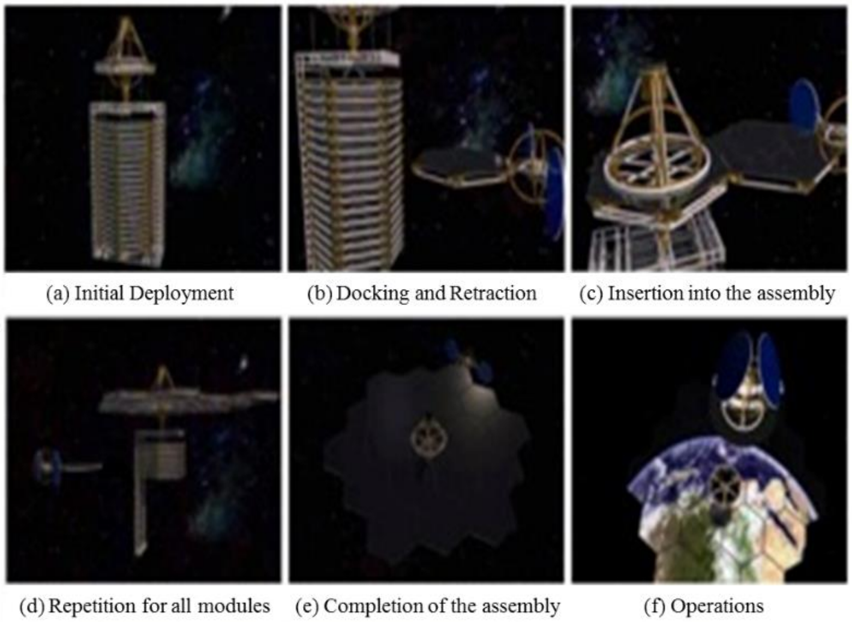
Key Nomenclature

AFS	Aurora Flight Sciences
ARMADAS	Agile Reconfigurable Modules with Autonomous Docking for Assembly and Servicing
ATLAST	Advanced Technology Large-Aperture Space Telescope
CAD	Computer Aided Design
DARPA	Defense Advanced Research Projects Agency
DOF	Degrees of Freedom
ExpV2	Expansion Port Version 2
GEO	Geosynchronous Earth Orbit
GSP	Guest Scientist Program
HL/FFS	Hybrid Length/Force Frequency Strain
ISS	International Space Station
MIT	Massachusetts Institute of Technology
NASA	National Aeronautics and Space Administration
RARC	Resource Aggregated Reconfigurable Control
RINGS	Resonant Inductive Near-Field Generation System
RSA	Robotic Servicing and Assembly
SMC	United States Air Force Space and Missile Systems Center
SPHERES	Synchronized Position Hold Engage and Reorient Experimental Satellites
SSL	Space Systems Laboratory
SWARM	Self-assembling Wireless Autonomous Reconfigurable Modules
TRL	Technology Readiness Level
UART	Universal Asynchronous Receiver/Transmitter
UDP	Universal Docking Port
VERTIGO	Visual Estimation for Relative Tracking and Inspection of Generic Objects

1 Chapter 1 – Introduction

1.1 Motivation: Robotic Servicing and Assembly (RSA) and the DARPA Phoenix Project, Testbed Development, and Scaling

The demand for on-orbit servicing and assembly capabilities has been increasing as space systems grow in size, complexity, and capability scope. There already are several projects which will utilize new robotic servicing and assembly (RSA) technologies. For example, the Optical Testbed and Integration on ISS eXperiment (OpTIIX) program, a joint project between the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory, Goddard Space Flight Center, Johnson Space Flight Center, and the Space Telescope Science Institute, aims to assemble a 1.5m telescope aboard the ISS robotically. This demonstration would require moving beyond simple formation flight control into the realm of resource aggregated reconfiguration. Another telescope project which will rely heavily on RSA technologies is NASA Goddard's Advanced Technology Large-Aperture Space Telescope (ATLAST) program, which plans to develop a highly scalable architecture suite for assembling telescopes much larger than currently feasible by exploiting new RSA technologies and economies of scale. On-orbit assembly of large space telescopes reduces risks and overcomes difficulties associated with launch vehicle constraints, integration and testing cycle times, servicing and maintenance capabilities, and ground testing constraints. Figure 1 shows an example of an on-orbit assembly process which relies on satellite reconfiguration and assembly technologies yet to be fully developed.



(a) Initial Deployment (b) Docking and Retraction (c) Insertion into the assembly

(d) Repetition for all modules (e) Completion of the assembly (f) Operations

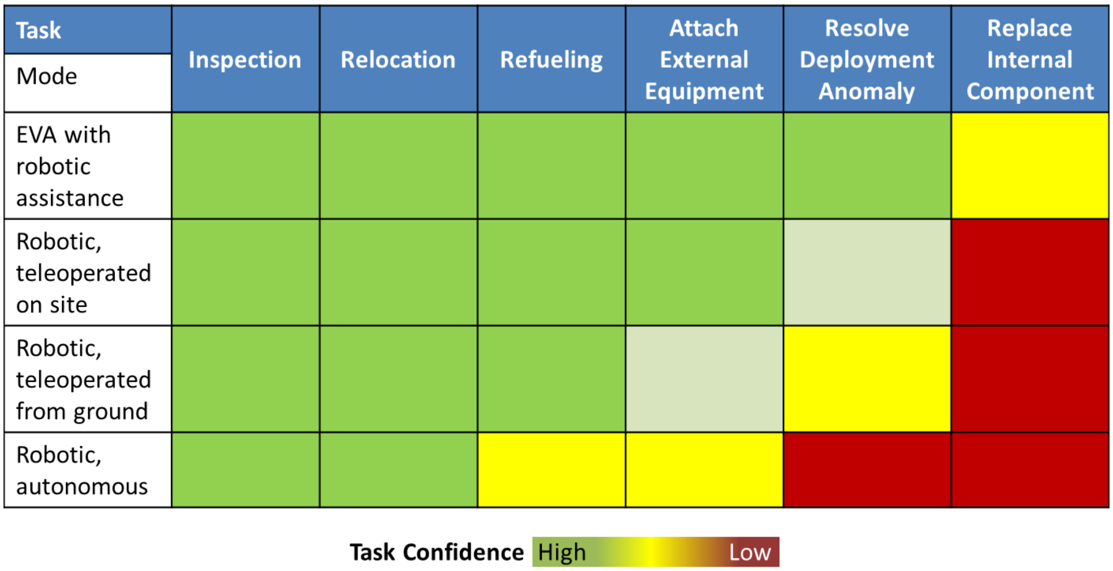
Figure 1: On-Orbit Telescope Assembly Process (Mohan, 2010)

The function of on-orbit servicing can be applied to spacecraft already in orbit today because servicing enables the repair or replacement of components, subsystems, and fuel or cryogenics to extend the operational lifetime of a satellite system. Additionally, the ability to conduct on-orbit servicing can enable the deorbiting or end of life disposal of defunct satellites which cannot be repaired back to a fully operational status or those which are no longer economically viable. Servicing techniques have been explored with humans in Low Earth Orbit, but little has been tested with respect to fully robotic servicing. Further discussions can be found in Guo (2009), Wang (2013), Fredrickson (2003), Reintsema (2012), and in Section 1.3.

The function of on-orbit assembly, however, can primarily be applied to spacecraft yet to be launched, since on-orbit assembly typically entails designing a system to be assembled from the start of a space project. There are many advantages, though, which counter the required added design effort. On-orbit assembly enables the launch of satellites which could not be nominally launched monolithically. Launch vehicle constraints drive most of the structural design requirements for space systems. The ability to assemble multiple modules reduces the direct impact of these launch vehicle constraints, since the spacecraft can be launched in multiple pieces and subsequently assembled on-orbit. Consequently, spacecraft can be developed that would never have been feasible without assembly capabilities. Furthermore, the assembly process and multiple-launch sequence enables the development of staged deployment of system capabilities. Launches after the spacecraft begins operations can provide upgrades or additions to the spacecraft, such as new or improved instruments, actuators, or sensors. The expansion of a spacecraft's functionality over time increases the productivity and return on investment, improving the spacecraft's net worth. The decision to launch these upgrades based on need or on a fixed schedule can be treated as an architectural decision during the spacecraft system design process, thereby increasing the potential versatility of the system as a whole. Further discussions of on-orbit assembly can be found in Barnhart (2009), Guo (2009), Chu (2013), Mohan (2010), and in Section 1.3.

When combined with on-orbit servicing technologies, on-orbit assembly technologies become even more potent in their ability to shape future space system architectures. Combining the two into a complete RSA system enables multiple space system architectures which are much more robust to changing operating environments and performance demands. The life of each satellite can be changed drastically over time, and with the advent of modularity and fractionation, concepts to be discussed in Section 1.3, multiple reconfigurations of a satellite can enable near optimum functionality across a range of performance levels for comparably lower costs.

Unfortunately, robotic, on-orbit servicing and assembly missions entail extensive risk. As a direct result of multiple spacecraft operating in close proximity to one another, multiple elements with independent control laws must operate with very tight constraints on their physical boundaries. Further, it is very difficult to test all possible multi-satellite configurations on the ground with existing testbeds, since there are limitless possibilities of how satellites can be oriented and positioned relative to one another. Since each of these satellites is able to function independently from one another, there is an omnipresent risk of collision and damage which would result in the partial or complete loss of space elements. Moyer and Mauzy (2011) have outlined a high-level description of the various tasks that are required for robotic servicing and assembly and the current confidence in each, shown in Figure 2. The level of confidence is inversely proportional to the level of autonomy for each particular capability. For future robotic servicing and assembly missions, the authors state that it is desired to incorporate as much autonomy as possible into the system to enable new, more complex mission goals. Further, the capability of internal component replacement is crucial for extended missions; the figure shows how no level of autonomy currently provides full confidence in mission success across all mission modes. The RSA testbed developed in this thesis can be used to reduce the risks associated with such tasks in order to enable mission architectures that currently require in-situ human action. The figure therefore shows the need for risk reduction as a whole for complex servicing and assembly on orbit. A new testbed is required to reduce the risks associated with these tasks in order to improve the confidence in robotic capabilities and enable RSA flight missions.



Task Mode	Inspection	Relocation	Refueling	Attach External Equipment	Resolve Deployment Anomaly	Replace Internal Component
EVA with robotic assistance	🟩	🟩	🟩	🟩	🟩	🟨
Robotic, teleoperated on site	🟩	🟩	🟩	🟩	⬜	🟥
Robotic, teleoperated from ground	🟩	🟩	🟩	⬜	🟨	🟥
Robotic, autonomous	🟩	🟩	🟨	🟨	🟥	🟥

Task Confidence
High
Low

Figure 2: Current RSA Task Confidence (Moyer and Mauzy, 2011)

The DARPA Phoenix Program plans on cooperatively harvesting and re-using already existing, retired, non-operating satellite components in geostationary graveyard orbit to demonstrate the capability of creating new space systems for significantly decreased costs. To accomplish this goal, the Phoenix program aims to rely on a new type of spacecraft system called a satlet. According to Barnhart (2012), a satlet is a small spacecraft with only a small fraction of the functionality of a complete satellite. In order to create a new satellite, an aggregate of multiple satlets is required. This key enabling technology is termed cellularization, and is a measure of how functions are distributed across multiple satlet types. An artist’s interpretation of a servicer spacecraft placing a satlet on an aperture in the Geosynchronous Earth Orbit (GEO) graveyard orbit is shown in Figure 3. The aggregation of multiple satlets into a single aggregate with full satellite functionality is a clear demonstration of on-orbit assembly technologies, and the need to repurpose already existing on-orbit hardware is similarly a clear example of the application of on-orbit servicing technologies. The DARPA Phoenix Program is therefore a prime example of a mission still under development which will rely heavily on RSA technologies that are still in their formative development stages. As such, Phoenix is a high risk demonstration of many new technologies, including multi-satellite proximity control, autonomous docking and undocking, and autonomous reconfiguration. Consequently, there exists a need for extensive testing to raise the probability of mission success, necessitating a directly traceable and scalable, low cost testbed.

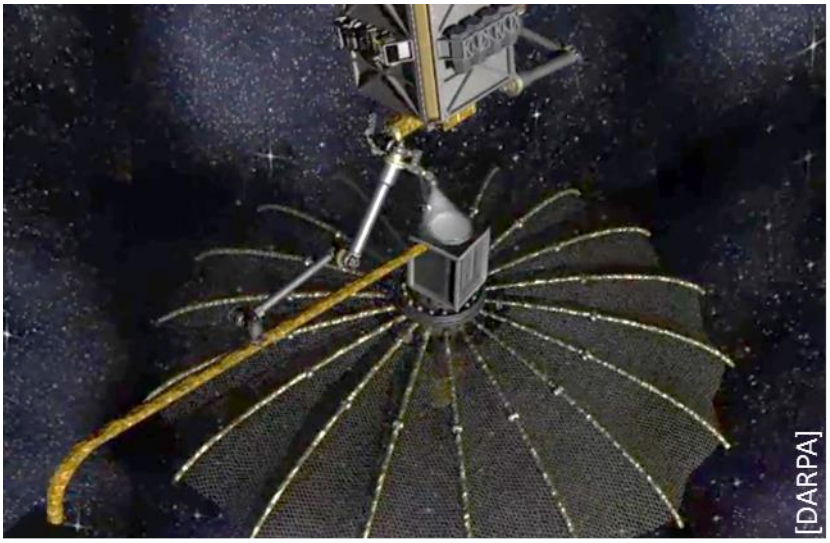


Figure 3: Artist Conception of DARPA Phoenix Mission

To demonstrate all of these elements in concert to reduce risk, there is a requirement for a demonstration in an authentic operational environment for the risk reduction of the various key technologies. All technology risk-reduction processes have inherent challenges, but in order to test a technology fully, it is important to test to failure in an authentic environment in order to understand what makes each technology element brittle. With this understanding, it is possible, then, to make improvements to the technologies. Even the process of integrating all of the technologies into a single demonstration imposes some risk. Consequently, a risk reduction testbed is required prior to the Phoenix mission.

Testing aboard the International Space Station (ISS) can provide significant risk reduction if coupled with ground testing. The ISS provides the authentic operational environment of extended microgravity, the most pertinent aspect to the subelements of the problem associated with RSA operations. Therefore, the combination of ISS testing with initial ground testing with a testbed that is operable in both environments can significantly reduce RSA risks for low cost. The development of such a testbed requires careful requirement definition in order to reduce RSA risks at low costs. This process is further discussed in Chapter 2.

Multiple testbeds already exist for reducing risks associated with on-orbit operations, including the risk reduction of many RSA technologies themselves. Servicing and assembly missions require extensive development of proximity operations algorithms, for example, and air bearings and micro-air-vehicles around the world have conducted fundamental and developmental research for such risk reduction. Further discussion of these existing testbeds is described in Section 1.3 and in Section 3.2.

Conducting a test sequence, as would be created through a design of experiments analysis, requires scaling a final system (typically down in scale for space systems) so that meaningful test data can be obtained on the ground or on-orbit with test articles representative of what will be eventually flown on the final mission. Each system usually has several thousand requirements that must be met in order for the mission to be successful. Of the standard methods for requirement verification (analysis, demonstration, testing, inspection, and similarity/analogy to prior systems), testing is one of the most common and informative. Unfortunately, the bulk of testing occurs near to final hardware delivery for launch: if delays occur early in the development process, testing is usually reduced to stay on time and financial budgets. There is a clear need, therefore, for time and cost efficient testing.

In an effort to make the most of risk-reduction tests, it is best to conduct a sequence of tests which build upon one another so that the full system’s operability is demonstrated incrementally. Doing so provides a clear path through the system of known operation modes and capabilities, helping to pinpoint design or development errors prior to full system integration. The corresponding costs associated with a slow-paced fully incremental testing process, however, prevent a system from ever being completely tested. A properly designed testbed and testing strategy is required for optimal risk reduction. Starting with the design of the testbed itself, the testing process can make use of existing infrastructure for cost reduction purposes, and the test articles can be designed to reduce multiple risks.

For example, it is rarely required to build a structure specifically for vibro-acoustic testing and to build a separate structure specifically for determining the fundamental oscillatory modes of the spacecraft: the same structure could potentially be used in both tests. Similarly, if there are differences between testbeds, significant effort should be placed into ensuring that they are as similar to each other as possible in scale.

The scaling of testbeds has been addressed through engineering modeling, and is discussed further in Section1.3. It is important to note the scale of each testbed, since only through testing with testbed-specific scaling laws can the behaviors be compared across multiple testbeds and be applied to the final, full-scale flight system. The final system is termed full-scale because it consists of the hardware and software that will be launched. Certain components, subassemblies, or subsystems may be too small to test easily; these elements usually are scaled up, increasing their size above that of the final elements. Other elements, however, cannot be tested without decreasing their size, so scaling down to create representative smaller elements is required to enable testing. This scaling process is imperfect, since a scaled system cannot behave identically to the original, non-scaled system. Using scaling laws developed specifically for a given scaled system, however, can solve this and other related issues. Further analysis on this topic is discussed in Sections 1.3 and 2.2, and in Chapter 5.

1.2 Development Process Overview and Rationale

The process for creating an incremental and iterative testbed for risk reduction requires the knowledge of the final flight system and the principal risk areas that require risk reduction. This knowledge comes from the development of the flight project itself. Early in the design phase, such as during the standard NASA Pre-Phase A through Phase A periods, high level concepts are discussed for determining characterization of the flight mission. While feasibility concepts are discussed, little emphasis is traditionally placed on the risk reduction strategies which will inevitably be necessary in later design phases, such as Phase B and Phase C. Consequently, the mission concept definition typically specifies the technologies which eventually will be incorporated into the final mission architecture, but not the means for maturing the technologies to flight readiness.

The technologies engaged in flight missions often span a range of Technology Readiness Levels, or TRLs. These levels range from 1 to 9 as a means of defining the maturity of a given technology or system. Table 1 shows a sample definition for each Technology Readiness Level as defined by Wertz (2011).

Table 1: Technology Readiness Levels (Wertz, 2011)

Technology Readiness Level	Definition
1	Basic principles observed and reported.
2	Technology concept and/or application formulated.
3	Analytical and experimental critical function and/or characteristic proof of concept.
4	Module and/or subsystem validation in a laboratory environment (i.e., software prototype development environment).
5	Module and/or subsystem validation in a relevant environment.
6	Module and/or subsystem validation in a relevant end-to-end environment.
7	System prototype demonstration in an operational, high-fidelity environment.
8	Actual system completed and mission qualified through test and demonstration in an operational environment.
9	Actual system proven through successful mission-proven operational capabilities.

TRLs are useful for correlating the components and subsystems that comprise a full system with system risks. For example, a system comprised of components with lower TRLs on average will generally have higher associated technical risks than a system composed of on average higher TRL components because of the inherent lack of component maturity. The testbed development process is crucial in lowering risks associated with technologies which are low TRL and are still in their formative development stages.

This thesis discusses the methodology for creating a new, incremental, and iterative testbed for reducing the risks associated with RSA technologies with the DARPA Phoenix program as the principal case study. As such, the thesis focuses on four major steps: determine testbed requirements, create or modify an existing testbed, conduct incremental testing, and scale test results. This four step process is used to determine the most appropriate testbed to develop in order to mitigate the most risk for the least associated costs, both in terms of time and money. Dr. Alvar Saenz-Otero discussed in his 2005 thesis a series of testbed design principles. These seven guiding principles provide a basis upon which to devise new testbeds for remote operation. The four step process described here applies these seven principles to the task of RSA technological risk reduction.

1.3 Literature Review and Research Gap Analysis

The development of a new testing facility for the risk reduction of key, formative stage RSA technologies started with a comprehensive literature review to determine the existing research in related fields. Additionally, the literature review was able to demonstrate that a gap exists in the current literature that necessitated the research presented in this thesis in order to develop the risk reduction testbed properly.

The literature review focused on the three principal areas of cellularized and fractionated spacecraft, on-orbit robotic servicing and assembly, and testbed development and scaling laws. These three major research areas have each enjoyed significant research focus over the past several years as missions have become more capable and the concept of RSA missions has gained traction in the astronautical community. Little research, however, has attempted to integrate the fields together, necessitating the research presented in this thesis, which serves as a union of the research areas to enable the development of RSA testbeds that are traceable and scalable to the final flight missions.

One of the most relevant research areas in the field of cellularization and fractionation of spacecraft has been performed by David Barnhart (2012). The author provides an overview of the DARPA Phoenix mission, a project aimed at the repurposing of hardware already in place within the Geosynchronous Earth Orbit (GEO) graveyard orbit. Repurposing hardware requires technologies associated with both servicing and assembly, since the existing on-orbit hardware must be brought to an operational status with new mission goals with new hardware elements. To do so, several spacecraft are required, including a large servicer/tender satellite and a multitude of small satlets. Barnhart describes satlets as small satellites without complete, full satellite capabilities that are created using the process of cellularization. Cellularized satlets each have a small amount of the functionality of a complete satellite because cellularization is a measure of how functions are distributed among satlet types. Effectively, satlets are the embodiment of the physical decomposition of a monolithic satellite into multiple separate modules. Each satlet is considered to be a separate, indivisible, self-sustaining unit, capable of: physically mounting to other satlets or servicing spacecraft, communication with other satlets, command processing, health management, thermal management, and power management. Figure 4 shows an example of how Barnhart envisions this decomposition: the multiple functions of a monolithic satellite can be attained through the aggregation of multiple satlets of multiple satlet types on orbit.

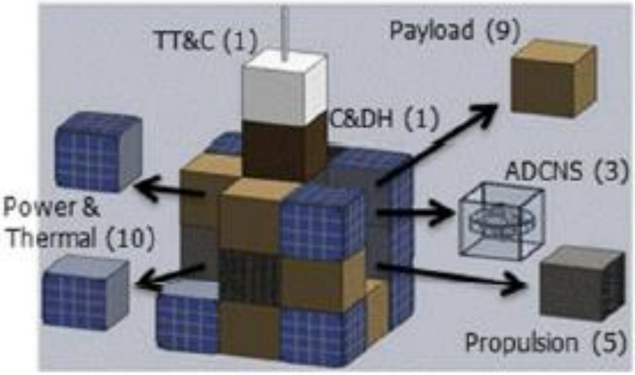


Figure 4: Diagram of Satlet Concept (Barnhart, 2012)

The process of aggregation enables multiple satlets to form a cohesive unit that can operate together to perform a given mission task. This method for satellite construction, therefore, enables the aggregated system: to minimize repeated functionality between modules when attempting to assemble a specific satellite or meet certain performance goals, to provide partial functionality in a servicing scenario when attempting to replace or create certain capabilities on the spacecraft being serviced, and to maintain a clear internal understanding of what capabilities can be created with a given suite of satlets. For example, a servicer/tender satellite might have a certain compliment of satlets that can be used to conduct a servicing mission of one type, but not have the appropriate satlets to perform another. As a result, a different servicer/tender would need to be called upon to complete the mission with a different set of satlets. This variability between which satlets are available to a given servicer/tender increases the architectural design space as to the servicing and assembly missions that may be undertaken.

RSA missions themselves are a crucial extension of current capabilities in order to continue the economic development of the space industry. In the same paper, Barnhart describes how the concept of cellularized satlets can reduce the cost of repurposing missions. Figure 5 shows the projected impact that cellularization will have on the cost of a mission to repurpose an aperture of a given mass from the GEO graveyard orbit. Importantly, though the figure centers on repurposing, Barnhart writes that being able to achieve performance goals with satlets to avoid unnecessary performance overlap between satlets and being able to launch a large number of satlets to perform either on-demand or scheduled servicing to take advantage of economies of scale cost benefits are fundamental beneficial attributes of cellularization that can be extended to other RSA missions beyond Phoenix.

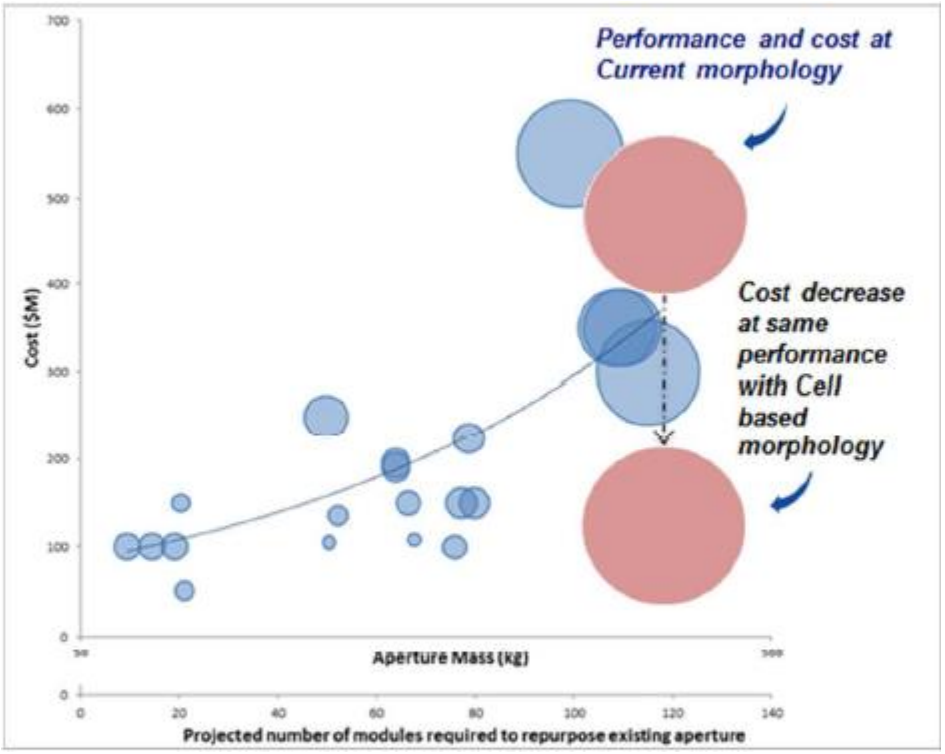


Figure 5: Projected Impact of Cellularized Spacecraft on Mission Cost (Barnhart, 2012)

A related technology concept to cellularization is that of fractionation (Guo, 2009) (DuBos, 2011). As explained in Brown (2006) fractionation is “the decomposition of a spacecraft into modules which interact wirelessly to deliver the capability of the original monolithic system, allowing system flexibility, maintainability, scalability, and reconfigurability”. Therefore, fractionation provides a portfolio of modules based on the physical decoupling of components to reduce system fragility through distribution across multiple modules. Like cellularization, fractionation seeks to reduce mission costs through maintaining contractor diversity and application of learning curves to the manufacturing costs of each module. The DARPA F6 (Future, Fast, Flexible, Fractionated, Free-Flying

Spacecraft) program was the first heavily researched mission to utilize the fractionation of spacecraft. A multitude of papers have been written on the F6 program, including Brown (2008), Eremenko (2011), Brown (2006), and Brown (2009). A more recent fractionated spacecraft program that has been proposed is the Pleiades system in the paper by LoBosco (2008). These papers acknowledge that there are many technologies which require extensive testing and verification prior to implementing a fully fractionated spacecraft system. Such technologies include cluster or formation flight, data transmission, fractionated navigation systems, distributed capabilities and data resources, and the transmission capabilities of power, force, and torque.

An example of a potential on-orbit demonstration sequence that relies on these technologies is shown in Figure 6, as proposed in Eremenko (2011). Unfortunately, there is no testing sequence in place to reduce the risks associated with these demonstrations. The demonstrations have no ground testing or intermediate microgravity testing planned despite the large number of novel, untested technologies. Consequently, the four demonstrations described in the figure are currently only conceptual; additional research and development into these four multi-satellite technology demonstrations is required prior to implementation on-orbit.

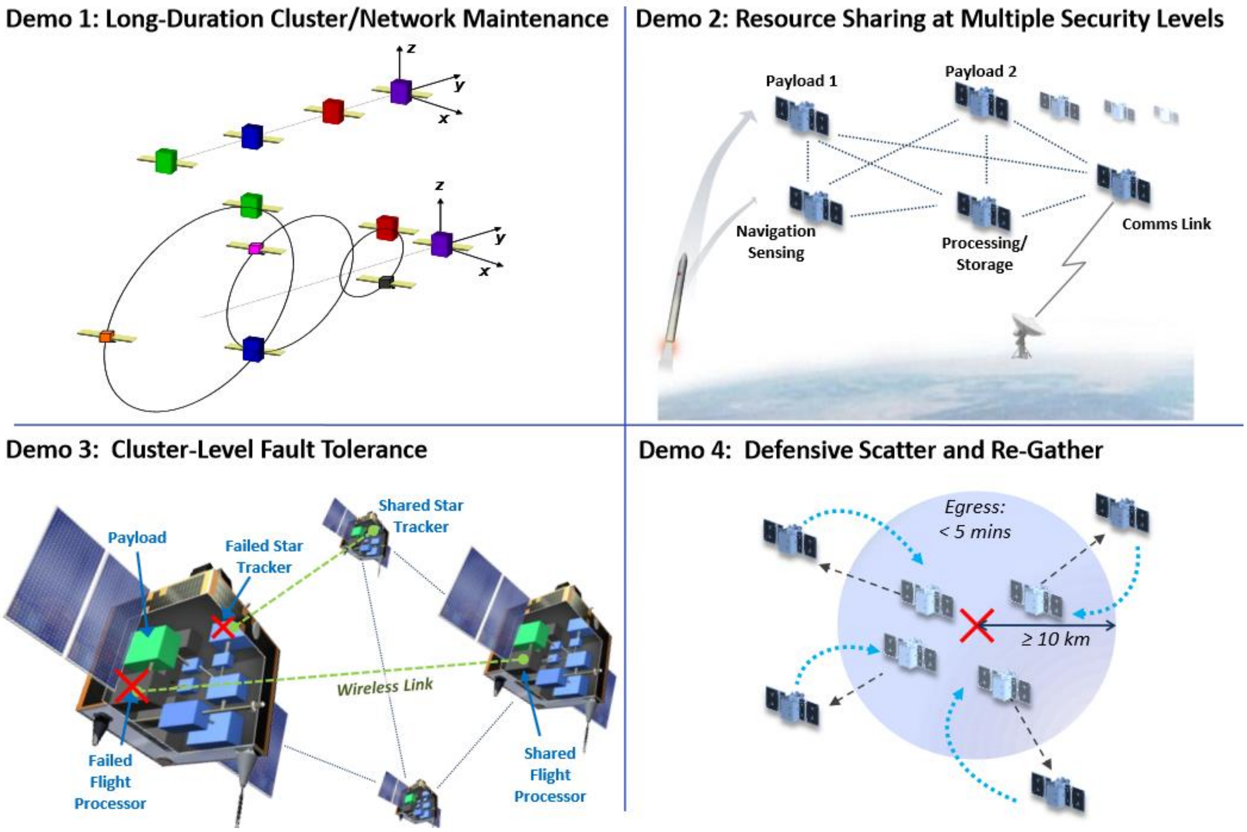


Figure 6: Proposed 2015 Fractionated Spacecraft Demonstrations (Eremenko, 2011)

Horsham (2010) has proposed another type of RSA mission architecture that is focused on what is called a space harbor. In this paper, a space harbor transport facility for a fleet of robotic servicing spacecraft is described. This satellite system includes separate facilities for a sate command, communication, and control system, a parts station, a fuel station or depot, and a fuel/parts replenishment transport vehicle system. Figure 7 shows an artist's image of what the space harbor could look like, including an octagonal truss section for the docking and undocking of all servicing satellites and fuel or replacement component pods. This vision likely will not be implemented in the near future because of the lack of research into many of the required enabling technologies. The authors do not propose any risk mitigation strategies, nor do they propose a detailed mechanism for the on orbit assembly of their space harbor, though they do make use of existing spacecraft structures, exemplified by the use of truss elements from existing space stations. Many of the technologies required for the success of the space harbor will necessitate thorough testing and risk reduction, since the space harbor concept has yet to be demonstrated in either operational or laboratory settings.

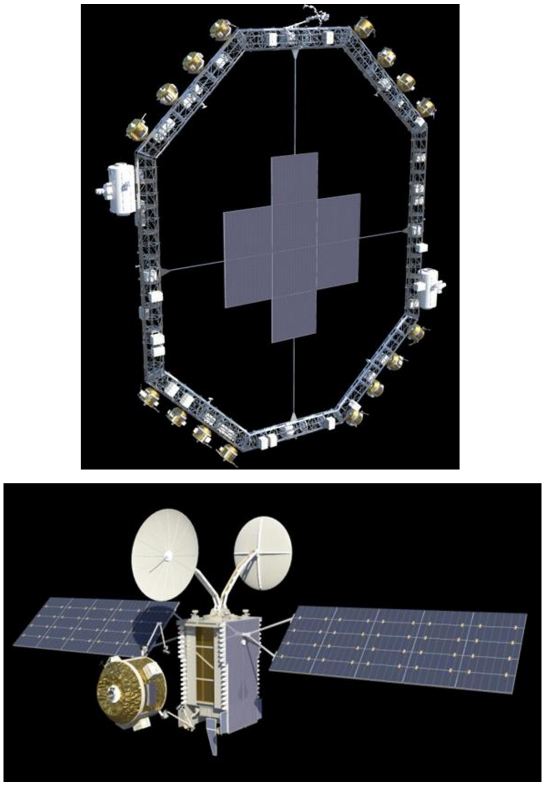


Figure 7: Space Harbor Concept [Top], Corresponding Servicing Concept [Bottom] (Horsham, 2010)

Another possible future spacecraft servicing architecture was proposed by Wang (2013). The authors propose a cluster of five satellites, consisting of a communications satellite, two armed robotic satellites, and two monitoring robotic satellites, to perform in situ servicing missions. These satellites would operate in close proximity to the target satellite or in closed orbits around it. Consequently, this cluster relies on the distribution of capabilities much like a fractionated spacecraft system, though each satellite is fully capable of functioning as an entire satellite on its own. Importantly, however, the authors acknowledge the need for ground testing to be followed by on-orbit demonstrations of incremental capability progression.

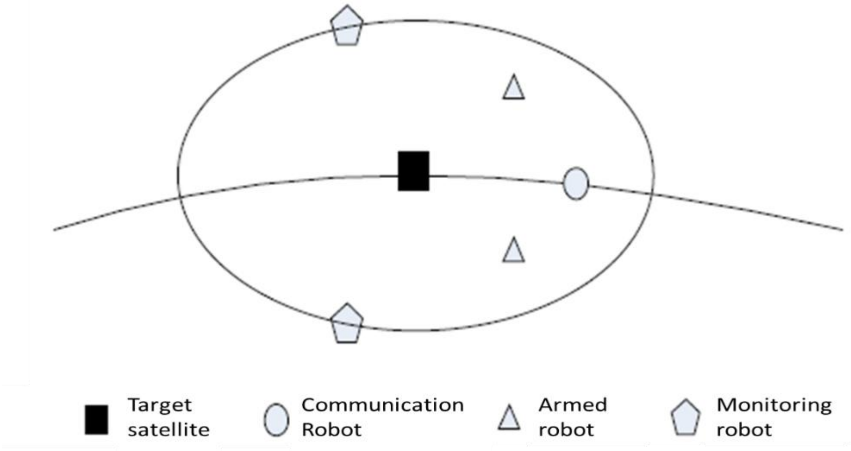


Figure 8: Proposed Micro-Satellite On-Orbit Servicing Platform (Wang, 2013)

Testing has been conducted for technologies related to robotic servicing and assembly. Columbina (1994) focused research on testbed control using the Automation and Robotics Technology Testbed for External Servicing (ARTES) testbed. Using a 6DOF manipulator and 3-camera navigation system, controllers were developed to perform an Orbital Replacement Unit (ORU) change-out scenario and to change the manipulator's compliance dynamically. This controls research, however, was not specific to RSA missions. To maintain the on-orbit servicing platform depicted in the figure, extensive formation flight research and testing is required. Owing to the 6DOF nature of such formation flight, long duration microgravity testing opportunities will likely prove highly valuable for the risk reduction of this mission.

Stroupe (2005) describes how the authors developed two ground systems capable of conducting construction missions. The Robotic Construction Crew and Lemur IIa systems, shown in Figure 9, are, respectively, a two robot ground system for autonomous assembly of structures from large beams and panels, and a construction algorithm testbed for force control for mobility and manipulation and adaptive visual feedback using interchangeable end effectors on six 4DOF limbs. These testbeds, however, are only applicable for captured, ground robotics, and cannot be readily converted to free-flying, action-at-a-distance servicing or assembly satellites. Accordingly, testbeds like these have limited risk reduction potential for on-orbit RSA architectures.

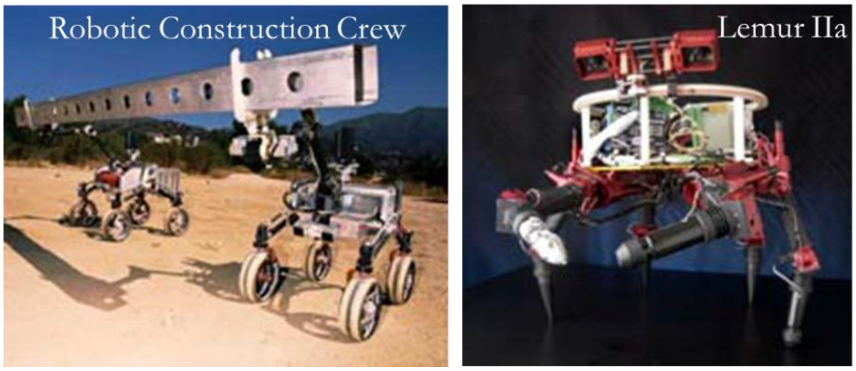


Figure 9: Robotic Construction Crew and Lemur IIa (Stroupe, 2005)

The Fredrickson (2003) paper describes the Mini AERCam, a 10 pound, 7.5 inch diameter free flying nano-satellite aimed at reducing the size of free flyers while maintaining controllability, reliability, and utility as a remote camera platform. The testbed was a technology demonstration unit with the goal of demonstrating the free flyer technologies of relative navigation, stationkeeping, and point-to-point maneuvering. This testbed was successful in its goals, but was unable to be expanded to test additional RSA technologies because of its design. Additionally, its development was focused on a "one step from fight" approach, rather than a multi-step risk reduction sequence of both ground and flight testing, increasing the risk associated with the mission. Additionally, only a single vehicle was created, so testing multi-satellite architectures is much more difficult with the Mini AERCam than with other testbeds which are inherently designed to test multi-satellite configurations. Further discussion of multi-satellite testbeds may be found in Section 3.2.

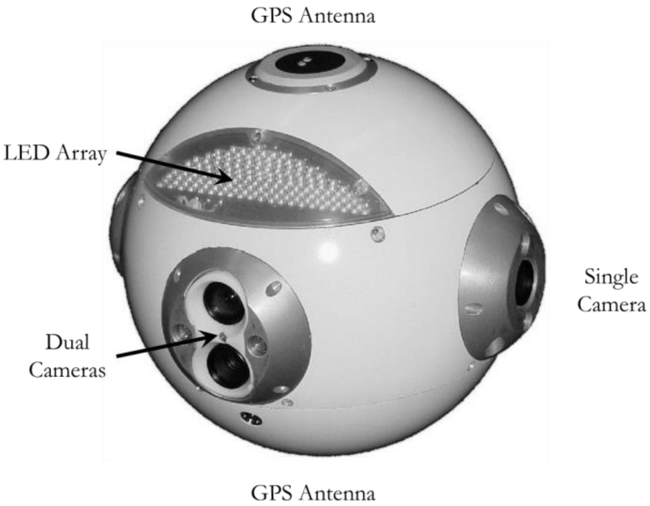


Figure 10: Mini AERCam External View (Fredrickson, 2003)

A ground-only testbed has been described in Sohl, 2005. This paper details the complimentary testbeds: the Formation Control Testbed (FCT) and the Formation Algorithms Simulation Testbed (FAST), shown in Figure 11. FCT is a 6DOF testbed that operates on a flat floor. A flight computer, compressed air thrusters, reaction wheels, gyroscopes, and a star tracker are mounted onto an air carriage which uses compressed air to life the structure off the floor and operate on the nearly frictionless surface provided by the expelled compressed air. This hardware testbed operates in conjunction with the FAST simulation testbed, which enables real time tracking and simulation of what occurs on the hardware. This joint testing process merges the use of simulation and hardware-in-the-loop demonstrations to determine the effect of formation algorithms among the three ground robot systems. Unfortunately, the algorithms cannot be tested using these robots beyond the confines of the flat floor, since the robots cannot move without the assistance of the nearly frictionless surface. Additionally, the robots do not have free flying counterparts for testing in reduced gravity aircraft or microgravity facilities like the International Space Station. Nevertheless, the ability for the simulation to verify the hardware testing is an important testbed capability for ensuring model-data correlation.

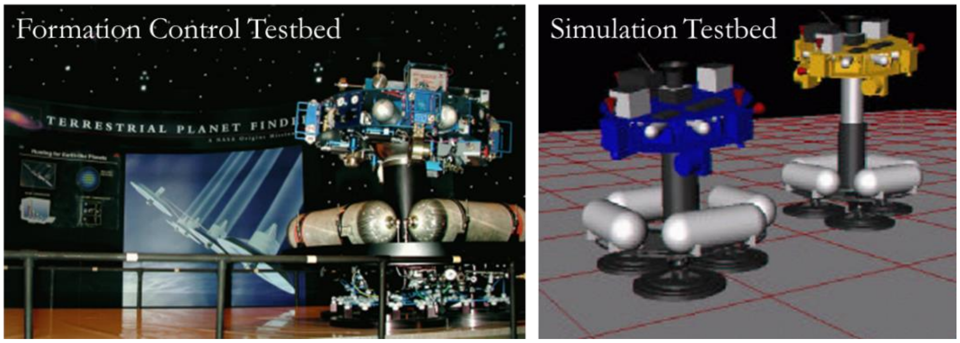


Figure 11: Formation Control Testbed and Formation Algorithms Simulation Testbed (Sohl, 2005)

Barnhart has also conducted research with ground hardware to demonstrate a few initial technologies related to the docking of satlets to already orbiting structures with application traceability to the Phoenix mission (Barnhart, 2009). Using a ground facility akin to that of the MIT Flat Floor, Barnhart was able to conduct demonstrations of satlet to floating beam docking with the test setup shown in Figure 12. As shown in the figure, two satlet-like ground units are able to dock to a floating beam. Importantly, however, the demonstrations by Barnhart do not address a significant number of formative stage technologies, and the algorithms that are developed on his ground-only testbed have no further risk reduction steps planned, including on orbit testing aboard the ISS. Therefore, further research is required to continue reducing the risks associated with his demonstrations by conducting testing on orbit in a relevant space environment. As a result, although Barnhart developed the Phoenix concept, he has not yet created an effective risk reduction process.

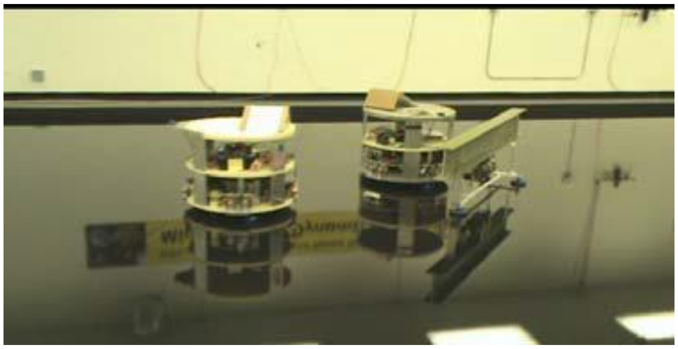


Figure 12: Ground Floating Beam Docking Demonstration (Barnhart, 2009)

The Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) facility combines many of the testbed capabilities discussed thus far. As described in Saenz-Otero (2000), Saenz-Otero (2005), Mohan (2007), and Mohan (2010), the MIT SPHERES facility affords long duration microgravity testing in a risk tolerant environment aboard the International Space Station, as well as ground testing capabilities with identical hardware and software between both environments. Consequently, testing of many RSA technologies can occur with this facility in the relevant operational environment of microgravity after being demonstrated successfully on the ground. This risk reduction pathway was initially developed in Saenz-Otero (2005), where the author described seven testbed design principles:

1. **Iterative Research**
2. Enabling a Field of Study
3. Optimized Utilization
4. Focused Modularity
5. Remote Operation and Usability
6. **Incremental Technology Maturation**
7. Requirements Balance

Here, the first and sixth principles are bolded because they will be further described in Section 2.4. These two design principles played a crucial role in the design and implementation of the risk reduction RSA testbed discussed in this thesis. Further explanation of these two principles can be found in Cockburn (2008) and Larman (2003). The other five, however, while not as immediately prominent in the research, nevertheless played a guiding role as well. For example, the principle of Optimized Utilization can be seen in the discussion of leveraging existing infrastructures where possible and beneficial in Section 2.3. The iterative and incremental research conducted with the SPHERES facility, however, focuses on controls and algorithm development, rather than on specific application to RSA testbeds. Therefore, research is required to develop scaling laws to be able to reduce RSA mission risks.

Scaling laws must be applied both to determine the proper size of a testbed, but also to then scale the results from the testbed back to the flight configuration. Scaling therefore must be incorporated into the developmental process early in order to make the most traceable testbed possible, but also throughout the testing process to ensure that the results that are obtained are capable of reducing the final flight system risks. David (1982) and Schuring (1977) have provided a basis for the creation of scale models. These models are used for sub-scale testing, meaning that the testbeds are a fraction of the size of the final flight system. This reduction in scale is common, since the size of most satellite systems is prohibitively large for full system testing to be cost effective. There are many ways for the testbeds to be created, though there are several common features of each that play a key role in the scale of each testbed.

One of the principal methods for scaling is determining a series of nondimensional parameters, i.e. those parameters which are invariant regardless of the scale of the testbed which they describe. Nondimensional analysis was one of the key research areas for Edgar Buckingham, who described the Principle of Similitude (Buckingham, 1915) and the Buckingham Pi Theorem (Buckingham, 1914). Baker (1991), showed how to apply these concepts to scale models. Kunes (2012) and Sonin (2011) provided valuable insights into various Buckingham Pi numbers, their physical meaning, and their derivations. These fundamental principles of scaling are crucial in comparing results across testbeds of multiple scales, since the nondimensionality of these Buckingham Pi numbers affords the ability to create metrics for determining the extent of traceability from one testbed to another. Further discussion of Buckingham’s work and the scaling of testbed results are in Section 5.1.

The process of determining scaling laws does not need to be rooted in the creation of nondimensional numbers. Dermitzakis, for example, wrote in his 2011 paper about the process for analyzing existing servo motors to generate a scaling law based for the output torque based on the mass of the motor, a pair of performance characteristics frequently used in determining which motor to select for robotics applications. Figure 13 is a plot of how specific physical parameters can be used to develop equations which can show how changing one physical parameter can affect another.

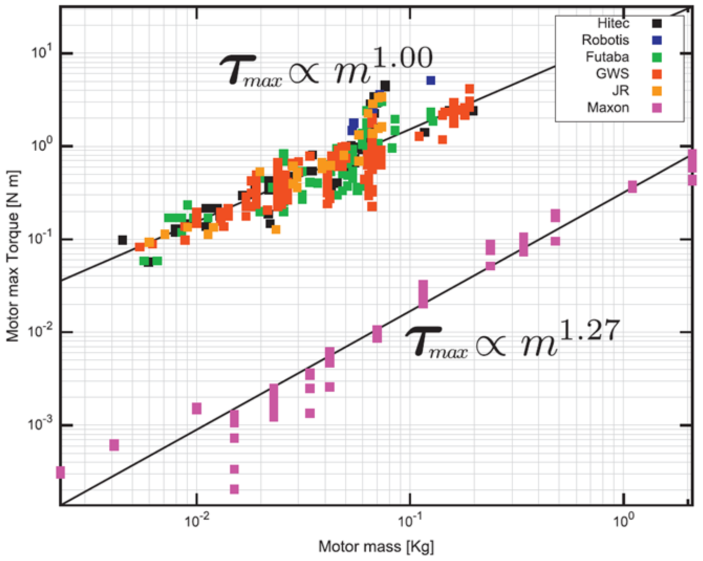


Figure 13: Motor Scaling Law Based on Output Torque (Dermitzakis, 2011)

The creation of a scaled testbed affects the physical parameters of the system in comparison to the full scale flight system. Analytical models can only provide an initial predictive analysis of structural properties, while hardware-in-the-loop testing can provide more insight into the behavior and control-structures interactions that will be seen on orbit. There are several means for ground testing hardware to provide data for full scale risk reduction, including full scale component testing, Multiple Boundary Condition Testing (MBCT) of full scale elements (Wada, 1986), or a hybrid modeling approach. Crawley (1988), Gronet (1989), and Crawley (1990) provide detailed analysis of the hybrid scale dynamic modeling technique. Because physical testing does not require analytic expressions for scaling principal and interaction effects, physical tests can often provide better risk reduction capabilities.

As described in these three hybrid modeling papers, the process of hybrid scaling is effectively an extension of simpler scaling methods. Replica scaling centers on the creation of a single scale factor based on the overall size of the testbed or model which is used at various powers to scale all dimensions, physical properties, and system responses. The top block of Figure 14 shows how a replica scale model compares to a full scale system with the added scaling parameter λ. An extension of the replica modeling technique is the multiple scale modeling technique, where testbeds and models are created with more than one scale factor. An example provided in Gronet (1989) is a model that is built at a scale factor λ_g, but is designed to have the response properties of a model built at scale factor λ.

Multiple scale models therefore afford the ability to have physical models built at certain limiting sized, but with the properties of a model which can better reproduce full scale effects. The middle block of Figure 14 shows how a multiply scaled model compares to a full scale system. Hybrid modeling attempts to combine these positive attributes of both of these methods.

Hybrid scaling results in a model or testbed which has the dynamic properties of a model at scale λ, but with the size governed by the scale factor λ_g. Importantly, hybrid scaling introduces what Gronet calls a strut length distortion factor c, which should be kept near unity to minimize the magnitude of compromises in model fidelity. To do so, Gronet recommends using λ_g values that are smaller, but close to the λ values. Following this recommendation allows the principal dynamics of interest to be preserved through scaling and the use of similarity laws. The bottom block of Figure 14 shows how a hybrid model compares to a full scale system. These methods are ways to create a scaled version of final flight hardware, and can then be used to determine the dynamics of the flight system based on scaled testing results. These methods are not specific to RSA missions, though this thesis will study the application of hybrid modeling in particular to reducing RSA technological risks.

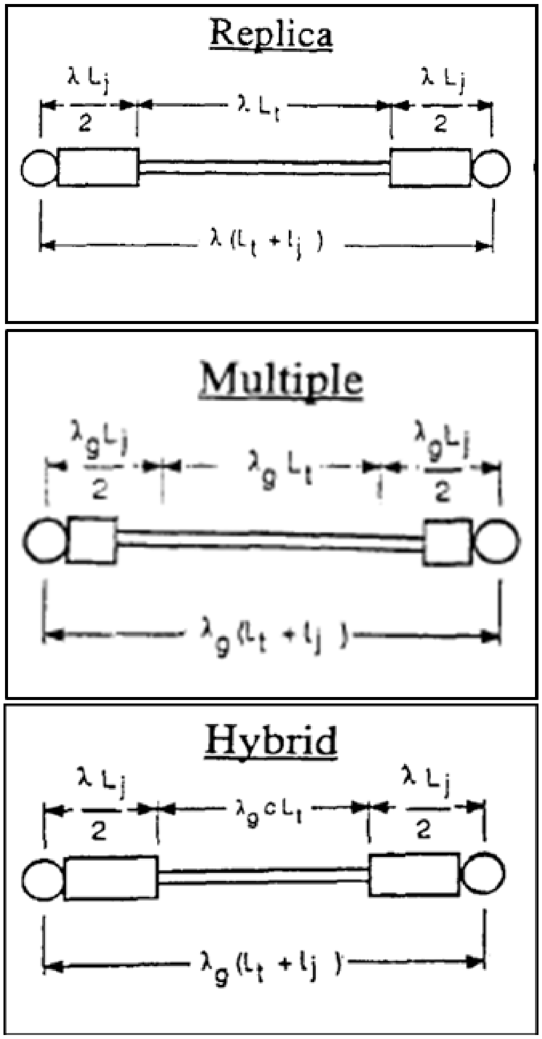


Figure 14: Replica, Multiple, and Hybrid Scaling Geometry (Gronet, 1989)

The Orbital Express mission is one of only a very small number of servicing missions which has been able to validate several enabling RSA technologies through an on-orbit demonstration between multiple satellites. Shoemaker (2003) provides an overview of the mission, which aimed at testing on-orbit refueling and reconfiguration between two satellites. The Orbital Express mission demonstrated in Low Earth Orbit (LEO) the ability for a servicing satellite named Autonomous Space Transport Robotic Operations (ASTRO) to rendezvous with a satellite named Next Generation Satellite and Commodities Spacecraft (NEXTSat) in order to perform local stationkeeping, docking, hydrazine refueling, and the replacement of an Orbital Replacement Unit (ORU). For this mission, the ORU consisted of a battery module for the NEXTSat power system. Figure 15 shows a diagram of the Orbital Express mission concept of operations. The mission therefore was able to demonstrate many of the technologies that are required for RSA missions, but only two satellites were involved in the demonstration. Multi-satellite aggregations like those in Phoenix were not included, nor were the satellites capable of being reconfigured to test new software based on the results from the initial demonstration. Therefore, additional testing with a new space system is required.

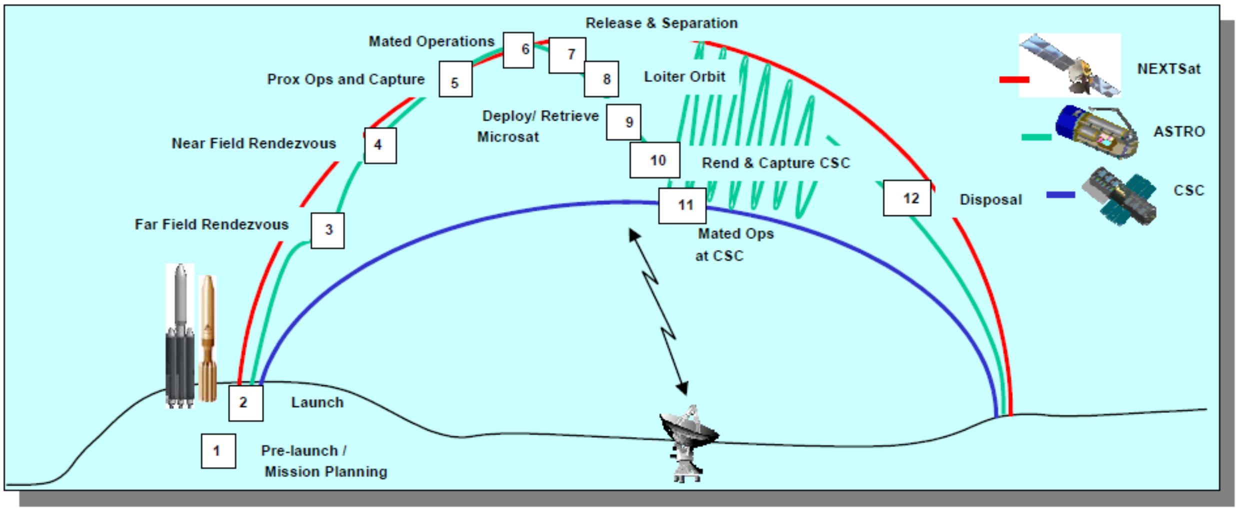


Figure 15: Orbital Express Mission Concept of Operations Diagram (Shoemaker, 2003)

Based on this literature review, it is evident that there is a gap in the current research. There exists a need for a traceable and scalable, low cost testbed capable of operating in an authentic operational environment for the risk reduction of RSA technologies. This testbed will need to be functional both on the ground and in a microgravity environment to enable multiple incremental and iterative testing opportunities and upgradeability overt time. The testbed must therefore be mission flexible while still remaining cost effective. Consequently, this thesis describes the process for creating this new testbed as well as the results of initial testing for the DARPA Phoenix mission.

1.4 Thesis Research Questions

Based on the literature review and research gap presented in Section 1.3, this thesis aims to address the question of:

> How can we reduce the risks associated with formative RSA technologies by creating an integrated on-ground and on-orbit testing facility and test sequence?

This research question addresses the need to reduce the risks that are inherently associated with on-orbit operations of multi-satellite architectures as well as those associated specifically with on-orbit servicing and assembly. As there is no current methodology for conducting such testing, the thesis poses the hypothesis that:

> An incremental, iterative testbed for operation in an authentic environment that leverages existing infrastructure while maintaining traceability to the final RSA project will provide highly valuable risk reduction for formative RSA technologies.

This hypothesis uses the initial research question as a launching point. By directing the hypothesis statement towards the research gap presented earlier in this chapter the hypothesis lays the framework for the research to be conducted and presented in this thesis. Therefore, the hypothesis that the proposed incremental, iterative testbed can reduce RSA risks is evaluated over the course of this work. The research goal, therefore, is:

- **To:** Provide *risk reduction* capabilities for future Robotic Servicing and Assembly (RSA) projects
- **By:** Creating a sequence of *incremental, iterative testbeds*
- **Using:** *Existing infrastructure* in an authentic, yet risk tolerant environment
- **While:** *Maximizing traceability* to an on-orbit RSA program

1.5 Thesis Roadmap

Figure 16 presents a graphical description of the flow of this thesis, summarizing the primary steps involved in both the development and use of a risk-reduction testbed for RSA missions, as well as how the thesis chapters map to the testbed development process. Importantly, the right justified white wording presents the high-level overview of what the left justified black wording describes on a chapter-by-chapter basis.

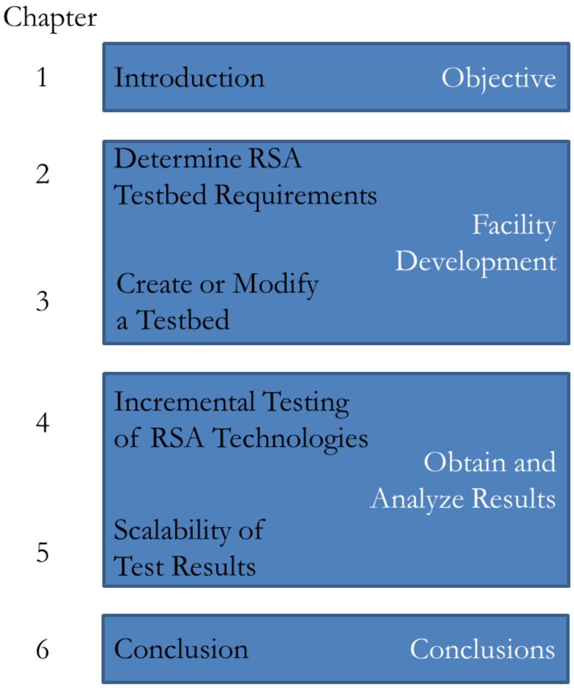

Chapter
1
Introduction
Objective
2
Determine RSA Testbed Requirements
Facility Development
3
Create or Modify a Testbed
4
Incremental Testing of RSA Technologies
Obtain and Analyze Results
5
Scalability of Test Results
6
Conclusion
Conclusions

Figure 16: Thesis Roadmap

Chapter 1 presents the introduction to the thesis, including the motivation, objective, and literature review with an associated gap analysis. This chapter shows the need for the research presented in this thesis and the background required for introducing the newly developed RSA testing facility.

Chapters 2 and 3 together describe how the testing facility was developed, starting with Chapter 2's explanation of the requirements that are levied on such a facility, and ending with Chapter 3's explanation of how to either create or modify a testbed upon which to base the new RSA facility. These two chapters therefore constitute the hardware development stage of the research, where new hardware is designed and manufactured in order to complete risk reduction testing.

Chapters 4 and 5 together describe how to use the facility to both collect and analyze the results that are obtained over the course of testing. Chapter 4 focuses on the need to define a testing sequence clearly that reduces the most risk for the least cost, maximizing the utility and efficiency of all testing, while Chapter 5 focuses on the scaling of the results obtained from the test sequence. The scaling of results is an integral part of the testing process, and is presented here as a means to identify if the testbed can be used as a risk-reduction step based on the testbed properties and how to use the small-scale facility to reduce full-scale risks for the RSA mission.

Chapter 6 presents the conclusions that can be gained from this research by summarizing how the facility is capable of reducing full-scale risks associated with formative stage RSA technologies by creating a new, small-scale testbed and an appropriately defined testing sequence with testbed-specific scalability laws in order to fulfill the objectives of this thesis and show the testbed that is developed confirms the stated hypothesis.

2 Chapter 2 - Determine RSA Testbed Requirements

2.1 Summary of Requirement Definition Process

Wertz (2011) provides a reference for the requirement definition process by stating that critical features of system or subsystem requirements: should be based on the mission objectives and incorporate a logical flow from system to subsystem requirements, should be incorporated into the trade analysis process to determine the final performance figures for the system, should state the function instead of the form of a given element, and should provide specific quantization of each function to be performed. These critical features apply to every requirement definition process, though the application to risk-reduction testbeds for RSA missions naturally entails a degree of specialization to fit the needs of the new testbed. This section is included to provide a basis against which to compare the types of requirements specific to the testbed presented in this thesis. Additionally, the overview provides background information on the entire process and the fundamentals of proper requirements themselves.

The requirement definition process translates stakeholder needs and wants into quantifiable sets of statements called requirements. These requirements must generally be specific to the particular system/subsystem/component for which they are written, quantifiable/measureable, attainable in the given timeframe of the project, relevant to the overall project, and time bound. Requirements should be written in a top-down manner, typically with several levels in a parent-child arrangement. Upper level requirements describe system elements, while lower level requirements often are more numerically detailed to address individual components. These relationships must be maintained throughout the requirements definition process, and each must be verified by a combination of inspection, analysis, testing, demonstration, or comparison to another similar system or component as would best suit each individual requirement.

Requirements should specify the function of a system element, not its form. In order to prevent requirements from directly limiting possible architectural decisions, the particular function or “what needs to be done” should be specified. Should the form, or “how the function is to be carried out”, be specified in the requirement, then large portions of the viable tradespace will be unnecessarily discounted. This factor is critically important when designing new testbeds for risk reduction. Because the flight system will incorporate new technologies which have not previously operated in space, both the system and technology incorporate new designs which must be tested. Should the requirements on testbed development be written in an unintentionally constrictive manner, as would be the case with forms being specified instead of functions, then feasible testbed configurations may not be considered, possibly preventing a low-cost risk reduction step from occurring. Further, because requirements specify the functions of all levels of the system design, the requirements definition process is closely tied to the system cost, schedule, complexity, and associated risks. Requirements are consequently a major driver in system budgets and allocations.

Requirements can fall into several general categories, including functional requirements, performance requirements, and constraints. These three categories describe the "what needs to be done", "how well a certain function needs to be accomplished", and "what cannot be traded", respectively. Importantly, these three categories have several sub-categories, such as those relating to interface and environmental requirements. Regardless of the category, each requirement needs to be verifiable, quantitative, unambiguous, non-conflicting, and without redundancy with other requirements. These categories do not affect the need to complete requirement closure, either. Requirement closure is the need to ensure that if all requirements are satisfied at each requirement subset (all of the children requirements of a given parent requirement), then the parent requirement would also be satisfied. Such requirement closure enables traceability back to where the sub-requirement was derived. This traceability is discussed in the next section in a modified context: traceability of the testbed to the flight system, instead of child to parent requirements.

A formal set of requirements as defined above should be made for each testbed in order to ensure that the testing conducted with each testbed is directly traceable and scalable to the flight hardware and that the results obtained are capable of reducing RSA risks. The next three sections discuss key aspects of any RSA testbed that should be captured in requirements, and detailed requirements for the on-orbit portion of the RSA risk reduction facility can be found in McCarthy's SM Thesis (2014).

2.2 Testbed Traceability

In order for a testbed to be effective at reducing risks for a particular mission, it must maintain traceability to the final flight system. There are several key areas in which traceability must be maintained, including the mission objectives, result scalability, and operational authenticity. The traceability of a system can be defined as the level to which a given testbed is similar to the final mission, though there are many variations of this definition for different applications.

For the RSA technology risk reduction testbed, traceability focuses on ensuring that what is being tested will eventually be used in the flight mission. Testing of applicable hardware and software is crucial for the testbed to be economically and technologically viable. Those elements which most require risk reduction should undergo the most testing, since these elements will be the most likely causes of failure due to the little knowledge about their operational characteristics or unreliable performance. Should these elements not be tested, then high risks would remain in the project, including the possibility of a preventable failure of either the element or the entire spacecraft. It is impossible, however, to test the hardware and software that will be used for flight exhaustively because of the costs associated with fully testing every possible failure scenario of a satellite. Consequently, the most applicable hardware and software should be tested to reduce the risks associated with formative stage technologies.

The testbed developed and described in this thesis focuses on the key enabling technologies for RSA missions and tests the hardware and software that will be required to implement them on RSA missions. These testbeds are also smaller scale versions of full flight systems, so some modifications to the hardware and software are necessary when transitioning to the full scale system. Such scaling does not invalidate the testbed, however, since the results are obtained from test increments which maintain the traceability to the flight hardware and software.

2.3 Leverage Existing Infrastructure

In order to reduce the time and monetary costs associated with the initial development and testing of a new testbed, it is crucial to leverage existing infrastructure where applicable. For example, it is nonsensical to develop a wholly new satellite testbed when one designed for a very similar purpose is available for use and can be modified without excessive difficulty to meet the requirements of the testbed. Chapter 3 delves into more detail about the process for creating or modifying a testbed, but it can be readily seen how the research presented leverages existing hardware and software infrastructure both on the ground and in orbit to create a new risk reduction testbed for RSA missions.

2.4 Incremental, Iterative Testing Opportunities

Another crucial aspect of any testbed is that it must provide the ability to test a hypothesis conclusively through incremental and iterative testing. The extensive testing must be able to either confirm or refute a hypothesis through testing with repeatable test conditions. For the RSA risk reduction testbed discussed in this thesis, the testing increments build upon one another to reduce the risks of associated RSA technologies because only with a carefully selected testing plan can the maturity of these technologies be increased to a flight ready level. Each testing increment will have its own hypothesis and a number of testing iterations. Each iteration represents a complete testing sequence, such that the data collected over the course of a single iteration will be used to determine the success of the testing increment by combining that increment's data with the other data from all of the increment's testing iterations. Section 4.1 delves into a further discussion of the importance of testing increments and iterations.

Each of the testing iterations of a particular increment must have the same operating conditions in order to compare data across tests. Only if all of the initial and operating conditions are the same can the data from one test be able to provide insight as to whether the results support or refute the testing hypothesis. For an RSA mission, these conditions focus on the ability to actuate and sense in specified degrees of freedom and in the relevant space environment. Further discussion of how the specific testbed increments occur in Chapter 4, but it is important to note the final operational environmental conditions.

The space environment subjects hardware to vacuum, thermal, radiation, microgravity, orbital dynamics effects, and field of regard effects. As such, testing should occur in the most relevant of these environmental parameters as possible to reduce the risk of on-orbit failures. Most components can be tested in terrestrial settings for vacuum, thermal, and radiation compliance, and specific testbeds can be developed for simulating orbital dynamics and field of regard constraints during ground testing. Extended microgravity, though, cannot be tested on the ground. The International Space Station (ISS) provides the only long duration microgravity testing environment. As seen in Figure 17, only the ISS is able to provide a risk tolerant, long duration, dynamically authentic operational environment for conducting risk reduction testing for RSA missions. Reduced Gravity Aircraft (RGA) can provide low levels of gravity, but only for short durations during each parabola. Ground testing cannot provide the full motion and microgravity seen on orbit, and orbital operations incur extensive risk, since spacecraft testbeds must themselves be strictly vetted through their own risk reduction campaigns. RSA testing must therefore make the best use of the available testing environments, so the combination of ground and ISS operations can provide the necessary low cost, long duration testing for multiple iterations of a carefully selected sequence of testing increments for RSA technology risk reduction.

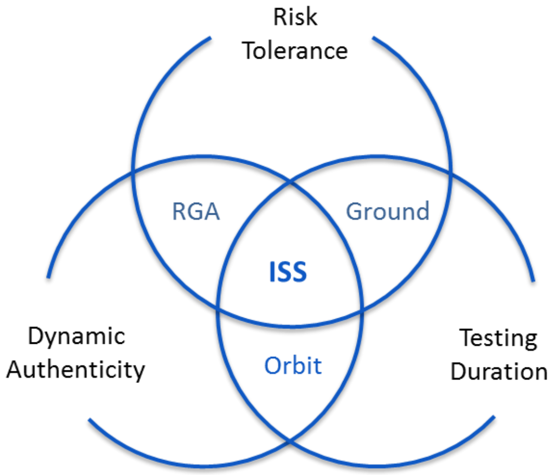


Figure 17: ISS Testing Environmental Conditions (Adapted from Halo System Requirements Document, 2013)

3 Chapter 3 – Create or Modify a Testbed

3.1 Make or Buy Decision

The make or buy decision is fundamental to any project, be it software or hardware focused. The RSA risk reduction testbed, in particular, relies heavily upon both. A hardware infrastructure is required to test various hardware additions and software technologies. As such, a corresponding software infrastructure is required to manage all interfaces between software and hardware elements. The need for a risk reduction testbed with which to test a multitude of RSA technologies drives the make or buy decision to determine the underlying technologies to be tested and the supporting design, implementation, and operational constraints of the testbed.

Each project has its own testbed requirements that lead to the creation of both qualitative and qualitative factors that influence the make or buy decision. Several examples of each are provided by Padillo (1999) and Fine (1996). Qualitative factors include: the level of control over the product to be made or bought, the historical reliability of the company making the product, the historical reliability of the products from that manufacturer, and the subsequent impact of the supplier on the project. These qualitative factors are highly dependent on the project, organization requiring the testbed, and the potential internal and external suppliers. Additionally, the qualitative factors must be assessed at each time a new element is required, since differences in requirements and supplier capabilities can change drastically even over the course of a single project. Quantitative factors can also fluctuate significantly, but they are typically more grounded in the incremental costs required to make or buy the new element. Consequently, quantitative factors often can be computed with more regularity and with more consistency. Examples of quantitative factors include the cost for each to both the supplier and the purchaser, the total production capacity of the supplier, and the costs associated with the creation of any required tooling to manufacture the new element.

There are two principal types of cost that must be considered when conducting a make or buy analysis: sunk and opportunity costs. Sunk costs are the monies which have already been spent and will therefore remain constant regardless of the outcome of the make or buy decision. Opportunity costs, however, represent the monies lost when using a purchased element or funding source in a way other than the optimal way. Therefore, only opportunity costs should be considered when conducting the make or buy decision.

Figure 18 shows a flowchart derived from Padillo (1999) that describes the process for conducting the make or buy decision process. The first step is to define the problem focus, i.e. make a decision as to the scope of the project and what elements need to be created. For the RSA testbed, a hardware and software infrastructure must be developed that will enable the incremental and iterative testing of key RSA enabling technologies that still are in their formative development stages. This problem focus leads to the determination of criteria with which to compare different alternatives, which are defined in the third step. This third step is where the potential suppliers are queried to determine their capabilities. The merits of each alternative are assessed in steps four and five, where a comparison is made between the alternatives to assess the importance of each criteria on the project outcomes successively to create a rank ordering of the various alternatives. Steps six and seven focus on the determination of a ordering scheme with which to determine the optimum decision choice. The determination of which supplier should produce the element is then made in the eighth step, where the actual make or buy decision is made.

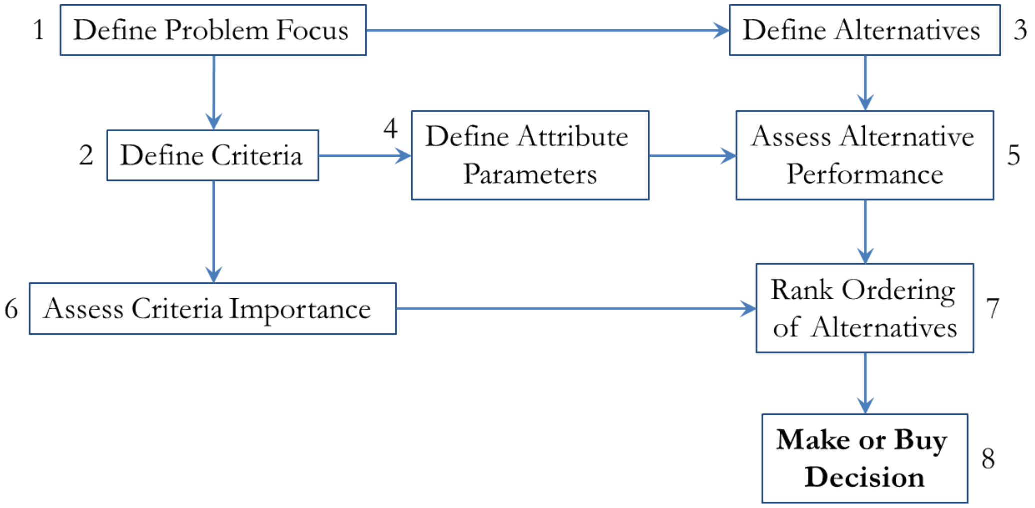

1
Define Problem Focus
Define Alternatives
3
2
Define Criteria
4
Define Attribute Parameters
Assess Alternative Performance
5
6
Assess Criteria Importance
Rank Ordering of Alternatives
7
Make or Buy Decision
8

Figure 18: Make/Buy Decision Flowchart (Padillo, 1999)

There are several risks which also play a role in determining the criteria importance and determining the rank ordering of alternatives. Following the form of the prototypical “iron triangle” of engineering, encompassing performance, cost, and schedule, Figure 19 shows several risk categories as explained by Cardin (2010). All of the peripheral risk categories represent those which funnel into overall programmatic risk. Risks to the program are the most broadly defined, since the program itself must ensure that all performance specifications are met on time and on budget. The RSA testbed, for example, must provide a set level of risk reduction for RSA missions, notably the DARPA Phoenix mission, within the timeframe for the technologies to be sufficiently matured and incorporated into the mission design while operating under tight cost constraints.

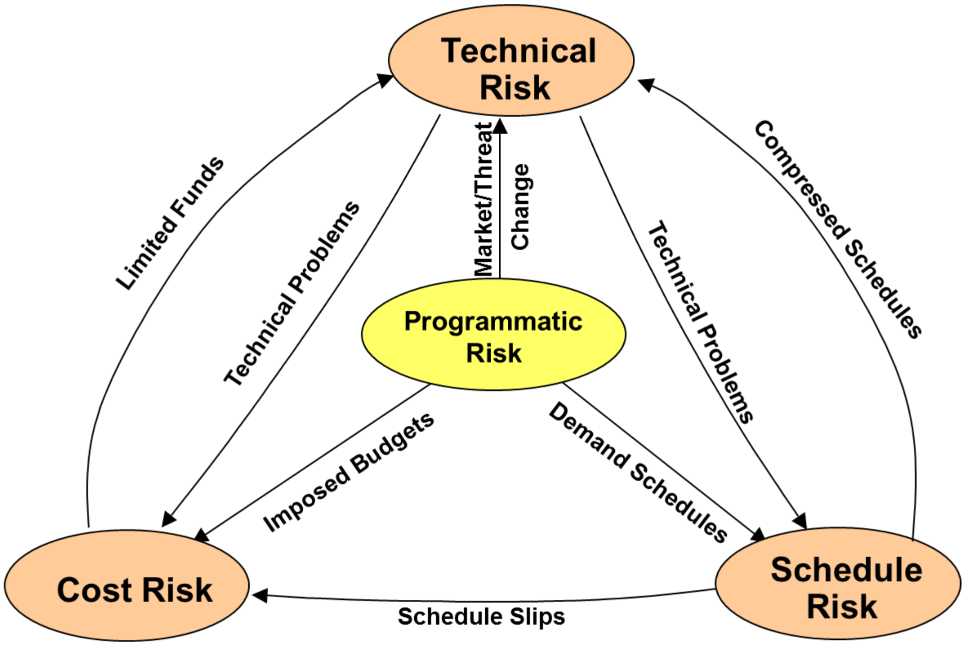


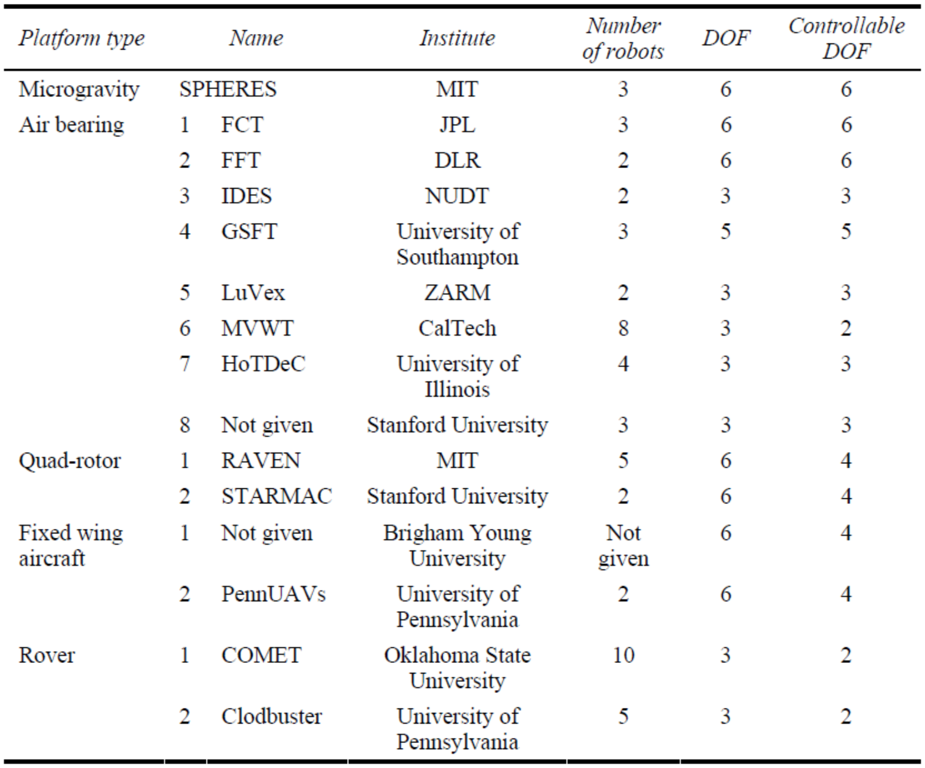
Figure 19: Risk Categories (Cardin, 2010)

For the testbed developed for this thesis, the make or buy decision was made in order to determine which infrastructure elements should be made in-house within the Massachusetts Institute of Technology's Department of Aeronautics and Astronautics, which should be purchased from outside vendors, and which elements should be repurposed from existing testbeds. Repurposing elements from existing testbeds can serve as a hybrid approach: existing equipment from in-house operations can be supplemented with new equipment that must be made or bought. The following sections describe the hardware and software infrastructure that has been selected for use in the new RSA testbed, and Section 2.3 describes the importance of building upon existing infrastructure where technically and economically possible and beneficial. These advantages play a strategic role in conducting the make or buy decision analysis. The RSA testbed was therefore developed to make the most efficient use of the existing infrastructure for maturation of on orbit technologies while reducing costs through careful selection of suppliers and properly scoping the needs of the testbed at each stage of the incremental testing process.

3.2 SPHERES Testbed

In order to develop a new risk reduction facility for testing RSA technologies, there is a need to incorporate multiple robots which can operate with multiple degrees of freedom in environments directly traceable to that which are expected during the RSA missions themselves. A survey of existing testbeds for distributed space systems was conducted and described in Chu (2013). Table 2 shows the results presented in the paper. As can be seen, the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) facility is the only testbed which can operate in microgravity, since three satellites are aboard the International Space Station as part of the US National Laboratory. Additionally, three identical satellites are located at MIT for operation on an air bearing with 3DOF. As a result, SPHERES is the only existing testbed that can conduct testing both on the ground and in a microgravity environment.

Table 2: Existing Testbeds for Distributed Space Systems (Chu, 2013)



Platform type	*Name*		*Institute*	*Number of robots*	*DOF*	*Controllable DOF*
Microgravity	SPHERES		MIT	3	6	6
Air bearing	1	FCT	JPL	3	6	6
	2	FFT	DLR	2	6	6
	3	IDES	NUDT	2	3	3
	4	GSFT	University of Southampton	3	5	5
	5	LuVex	ZARM	2	3	3
	6	MVWT	CalTech	8	3	2
	7	HoTDeC	University of Illinois	4	3	3
	8	Not given	Stanford University	3	3	3
Quad-rotor	1	RAVEN	MIT	5	6	4
	2	STARMAC	Stanford University	2	6	4
Fixed wing aircraft	1	Not given	Brigham Young University	Not given	6	4
	2	PennUAVs	University of Pennsylvania	2	6	4
Rover	1	COMET	Oklahoma State University	10	3	2
	2	Clodbuster	University of Pennsylvania	5	3	2

The SPHERES testbed was originally designed by the Massachusetts Institute of Technology's Space Systems Laboratory as a formation flight testbed, but with the extensibility capabilities for conducting research testing in many future fields of study, including docking and undocking, vision based navigation and mapping, electromagnetic formation flight and wireless power transfer, fuel slosh, and satellite reconfiguration. The SPHERES facility is in two locations: aboard the International Space Station (ISS) and at the Space System Laboratory (MIT SSL) on the ground. These two locations work synergistically together. While testing is conducted in the microgravity environment aboard the space station as a means of obtaining flight data in the relevant space environment, the test plans and code is developed and tested on the ground in 1g in order to demonstrate the effectiveness of the new technology. In doing so, testing aboard the ISS is much more time constrained, since astronaut crew time is much more regulated and less accessible than ground researcher time.

There are three satellites in each location, with each satellite being identical to the others. Each of these satellites represents a complete satellite bus, since each SPHERES satellite has complete attitude determination and control, power, propulsion, structures, avionics, communication, guidance and navigation, and payload systems, seen in Figure 20. Physical properties of each satellite are shown in Table 3. Importantly, the operational environment aboard the ISS neither requires the use of a dedicated thermal subsystem, nor the use of radiation hardened components. The payload system in particular is an expansion port, currently in its second revision to enable more massive and capable expansion port items to be attached to a satellite. This expansion port provides a much desired capability, since a single satellite itself is quite limited on the tasks that it can perform by itself. The ability to attach peripherals to the satellite directly enables future research in a wide array of possible areas. Each facility location also contains a Position and Attitude Determination System (PADS), a set of five ultrasonic beacons, which enables the SPHERES satellites to determine their locations with the predefined test volume. Ultrasound sensors are employed to take advantage of the "shirt sleeve environment" found in both operational locations. Control of the satellites is handled via Graphical User Interface and a single laptop or desktop computer which runs a test project. Each project can contain multiple tests, and each test can contain multiple maneuvers.

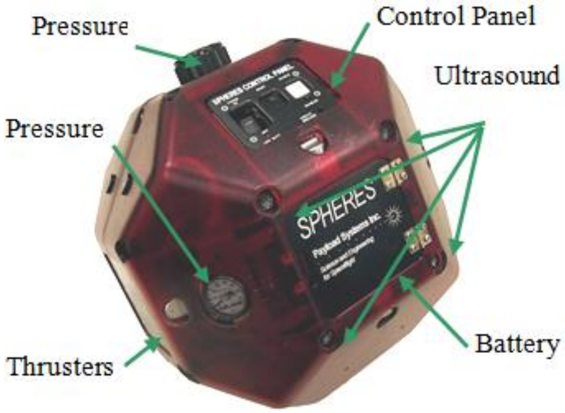


Figure 20: A SPHERES Satellite

Table 3: SPHERES Satellite Physical Properties

Diameter	0.22 m
Mass (w/tank & batteries)	4.3 kg
Max linear acceleration	0.17 m/s^2
Max angular acceleration	3.5 rad/s^2
Power consumption	13 W
Battery lifetime (replaceable)	2 hours

The hardware used to construct a SPHERES satellite is based on the need to create a functioning satellite bus. The SPHERES satellites rely on a SPHERES satellite-to-Laptop (STL) communication channel at 868 MHz for transmission of data and commands to and from the ground station computer, typically a laptop aboard the ISS. Each satellite contains an Inertial Measurement Unit (IMU) consisting of three gyroscopes and three accelerometers. Data from these sensors are combined with the ultrasound metrology system data to compute an estimate of the satellite's location using an Extended Kalman Filter. The ultrasound metrology system itself relies on 24 ultrasound receivers within each satellite to detect ultrasonic chirps from the five transmitting beacons on the periphery of the test volume. Together, these sensors can provide full state estimates for each satellite, totaling 13 state elements, including: x-position, y-position, z-position, x-linear velocity, y-linear velocity, z-linear velocity, quaternion 1, quaternion 2, quaternion 3, scalar quaternion, x-angular velocity, y-angular velocity, and z-angular velocity. A diagram showing the metrology system is shown in Figure 21.

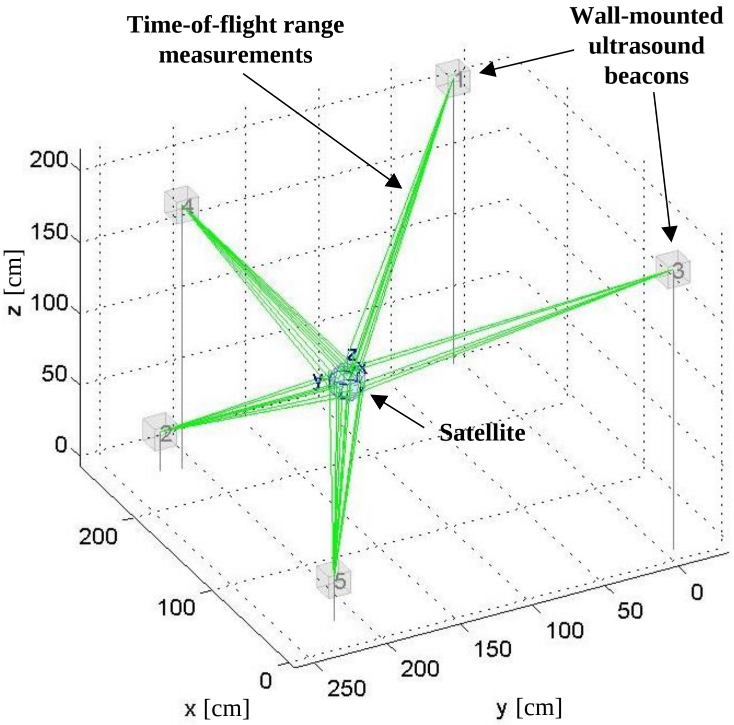


Figure 21: SPHERES Metrology System (Mohan, 2010)

The SPHERES facility was chosen as the basis for the development for the new incremental and iterative testbed for several reasons which are founded in the way SPHERES was itself developed. The SPHERES facility is inherently expandable because of the expansion port located on each satellite's +x face. The Expansion Port Version 2 (ExpV2), shown in Figure 22, provides two Universal Asynchronous Receiver/Transmitter (UART) lines through a standard electromechanical connector that employs captive thumb screws in a keyed position to attach to a peripheral rigidly. Historical use of the ExpV2 has demonstrated the success of communication and power being transferred to the Resonant Inductive Near-Field Generation System (RINGS), Visual Estimation for Relative Tracking and Inspection of Generic Objects (VERTIGO), and Self-assembling Wireless Autonomous Reconfigurable Modules (SWARM) hardware when directly mounted to the external face of the ExpV2. Unfortunately, the expansion port allows only one component to be attached at a given time, so further work, described later in this thesis in Section 3.4, is aimed at providing additional ports which are identical to one another.

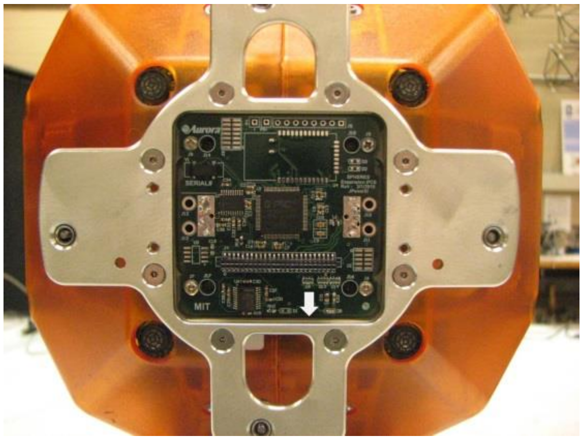


Figure 22: SPHERES Expansion Port V2

Further, the SPHERES Guest Scientist Program, or GSP, provides ready access to the SPHERES control law development tools. Specifically, the GSP enables researchers from any location to code the SPHERES satellites for operational testing at MIT or aboard the ISS. The SPHERES satellites themselves can be operated with standardized functions and control laws, but guest scientists can write customized software as needed. With clearly defined functions, interrupts, and communication channels, it is possible to configure each satellite to highly customized levels of control, command and data handling, or communications architectures. This functional versatility is highly valuable, since it enables multiple fields of study when analyzing varying satellite bus architectures.

Similarly, the three identical satellites at each operational location, at MIT on the ground and aboard the ISS in microgravity, allows for multi-satellite interactions that are both stochastic and deterministic. Multiple satellites can function in concert with one another, provide cooperative targets or assistance, or represent any level of uncooperative behaviors between satellite elements. The use of different GSP codes for each satellite allows them to function independently if desired, but the satellite-to-satellite communication capability enables data sharing and constellation, swarm, or cluster formations.

For these primary reasons, the SPHERES facility was chosen for modification to create the new, incremental, and iterative testbed for risk reduction of RSA technologies. SPHERES directly enables risk reduction for control algorithms as a microgravity analog to a wind tunnel where formative technologies can be fully tested across a range of flight environments to identify flight envelopes and tolerance to off-nominal conditions while maintaining low cost and risk. The ability to expand the SPHERES facility through both hardware and software modifications on the ground and in space provides the basis for the research presented in this thesis. The SPHERES facility, therefore, provides the basis for the new risk reduction testbed.

3.3 SWARM Components

The Self-assembling Wireless Autonomous Reconfigurable Modules, or SWARM, program was aimed at creating a system of hardware elements that would rely on SPHERES satellites for mobility to demonstrate the capabilities of a modular spacecraft for the testing of reconfiguration and self-assembly algorithms. For this testing, new hardware was developed to support the docking of multiple elements. Figure 23 shows a picture of a SPHERES satellite mounted atop a SWARM propulsion unit and a single-puck air carriage for testing on the MIT Flat Floor. The SWARM propulsion unit allows 16 additional thrusters to be actuated and controlled by the SPHERES satellite when operated in conjunction with a modified mixer and a connector to the ExpV2. This connector allows the satellite to communicate with the SWARM avionics board, which distributes commands for thruster firing and docking port actuation through the propulsion unit. The additional thrusters are identical to those within the SPHERES satellite, and provide cold-gas firing times that are equivalent to that of what the satellite is capable. Due to the size constraints of the propulsion unit, however, short CO_2 tanks are required to provide propellant to the additional thrusters. Owing to the placement of the thrusters to accommodate even these smaller tanks, the ExpV2 interface is set well behind the external dimensions of the propulsion unit. Consequently, a support structure mounts the Universal Docking Port (UDP) or any other peripherals that nominally would be connected directly to the ExpV2. The ExpV2 connector cable and SWARM avionics board provide the data handling for these peripherals as well.

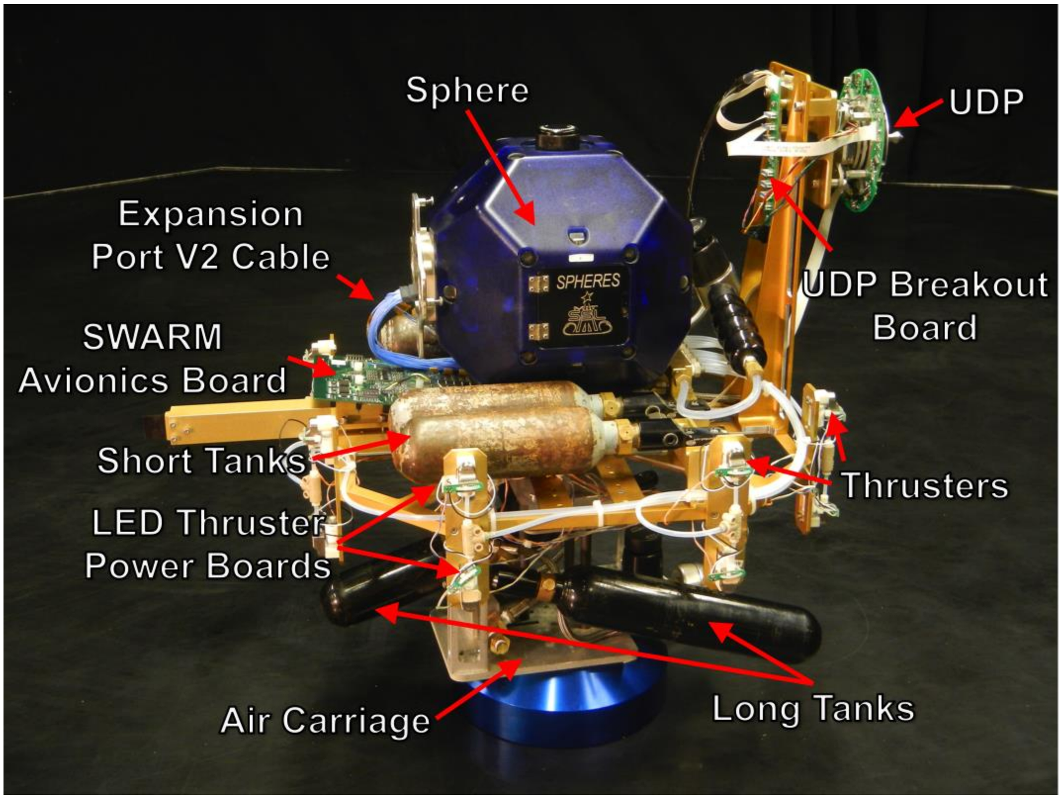


Figure 23: SWARM Hardware (ARMADAS Test Results, 2013)

The MIT Flat Floor is an octagonal 5m diameter poured epoxy surface used to test 3DOF motion of satellites using air carriage systems like that shown above in Figure 23. This air bearing surface enables nearly frictionless motion along the plane of the floor, but there are several drawbacks as identified in Tasker (2008). First, in the plane of the table, rotational dynamics are linear, but the nonlinear and coupling effects seen on orbit cannot be seen or tested except in the rare case of systems with diagonal inertia matrices. Unstable effects may not be seen during ground testing because of the constraint to the 2D Flat Floor surface. Further, coupling will be seen to a much lesser extent between rotational and linear dynamics unless the spacecraft body frame is aligned with that of the center of mass. Lastly, the air bearing surface does exhibit some friction and stick-slip behavior that will not be seen on orbit, especially at the low speeds seen with the SPHERES satellites. The testing of RSA algorithms, which involves larger masses being floated on the Flat Floor, can be hampered by these physical effects on the ground, leading to dynamic properties which are different from those expected on-orbit. Therefore, to avoid situations where improper ground testing of a component or system leads to an on orbit failure, the Flat

Floor must be properly maintained and cleaned, while the air carriages themselves are kept fully pressurized with CO_2 and as clean and scratch-free as possible.

The SWARM hardware provides a ready extension to the SPHERES facility for ground operations. The additional thrusters provide an expansion to the actuation potential of an individual satellite, allowing each satellite to function as if it were a much larger, more powerful satellite. The addition of the propulsion deck itself requires reconfiguration changes of the standard SPHERES mixer, controller gains, and property files, all of which are required for testing RSA technologies. The ability for the docking port to be incorporated simultaneously enables further reconfigurability between satellites, since with the UDP, two satellites can rigidly dock with varying levels of cooperation. Consequently, the SWARM hardware provides a modification to SPHERES which directly enables multiple RSA architectures to be tested, helping to reduce the cost of the testbed. The SWARM hardware, however, is only able to be incorporated for ground testing, since the thrusters are only capable of providing forces and torques in the plane parallel to the ground, and there is no mechanism for rigidly mounting it to the satellite in full 6 degree of freedom environments, as would be seen aboard the ISS. Consequently, the SWARM hardware provides a basis for the development of future systems which can perform similarly, but in the ISS microgravity environment. Such hardware systems have been developed, and are explained further in Section 3.4.

3.4 ISS Flight Hardware

In order to expand the research and continue reducing risk beyond ground testing, flight hardware is needed. Reconfiguration requires that multiple elements be able to maneuver on orbit, dock and undock, and create a number of different configurations with a standardized set of hardware elements.

The three SPHERES satellites that are on orbit aboard the ISS have a single expansion port. This expansion port allows for one satellite to connect to a single peripheral device, such as docking port or a camera system, but it does not allow multiple hardware expansions to be mounted simultaneously. Further, the single expansion port design precludes the possibility for multiple active sites around the satellite. The lack of multiple expansion ports on a satellite severely hampers the reconfiguration possibilities that are required for RSA risk reduction. New hardware is necessary, specifically to provide additional expansion ports and a robust docking mechanism. These new hardware elements are the Halo and the Universal Docking Port, or UDP.

The Halo is a structural expansion to a SPHERES satellite that enables each satellite to interface with up to six external peripherals through rigid mechanical and electrical interfaces called “Halo ports”. The UDP is one of many possible peripherals, with others including the Visual Estimation for Relative Tracking and Inspection of Generic Objects (VERTIGO) Goggles, optical rangefinders, thermographic cameras, LIDAR systems, control moment gyroscopes, or mockups of future microsatellites. These peripherals can be attached or removed from the Halo structure by the astronaut(s) running any given test session aboard the ISS as required to conduct a particular experiment. Further, the use of the VERTIGO Avionics (VA) Stack allows for significantly increased data processing power as compared to the SPHERES Digital Signal Processor (DSP) alone. The VA also provides high speed data lines in the form of Ethernet and USB to each Halo port to enable more data intensive peripherals to be developed and conduct uninterrupted RSA testing. These new capabilities therefore enable RSA testing in the risk-tolerant, dynamically authentic environment aboard the ISS by addressing challenges such as aggregating resources and reconfiguring control systems.

Figure 24 shows a labeled SolidWorks model of the Halo system, and Figure 25 shows a Halo mounted onto the SPHERES satellite with the VERTIGO Avionics Stack in place, representing a flight configuration without any attached peripherals. Figure 26 shows a photo of the functional Halo prototype developed for ground testing.

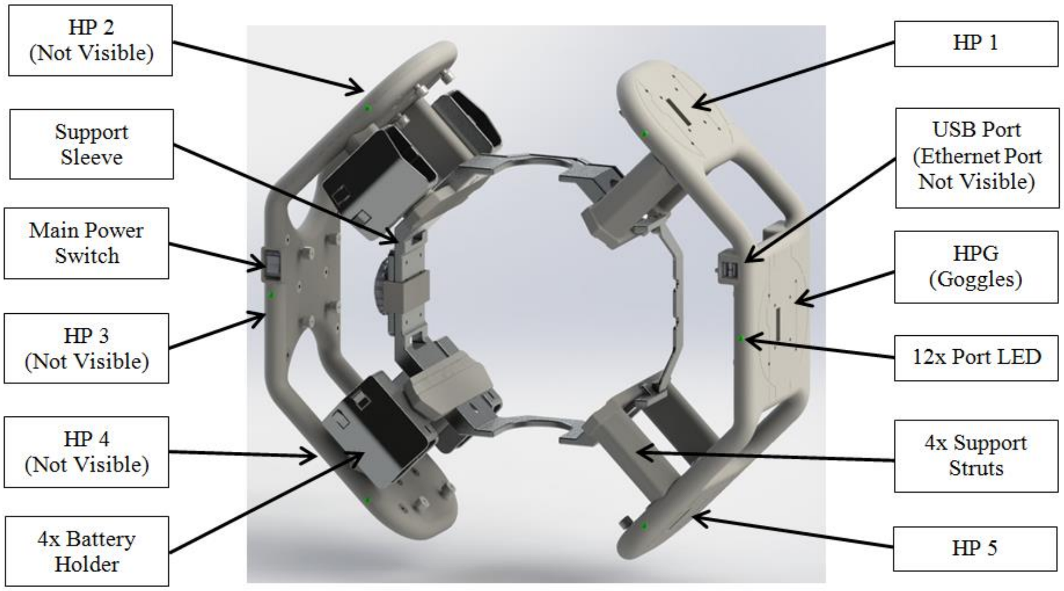


Figure 24: Labeled Halo System

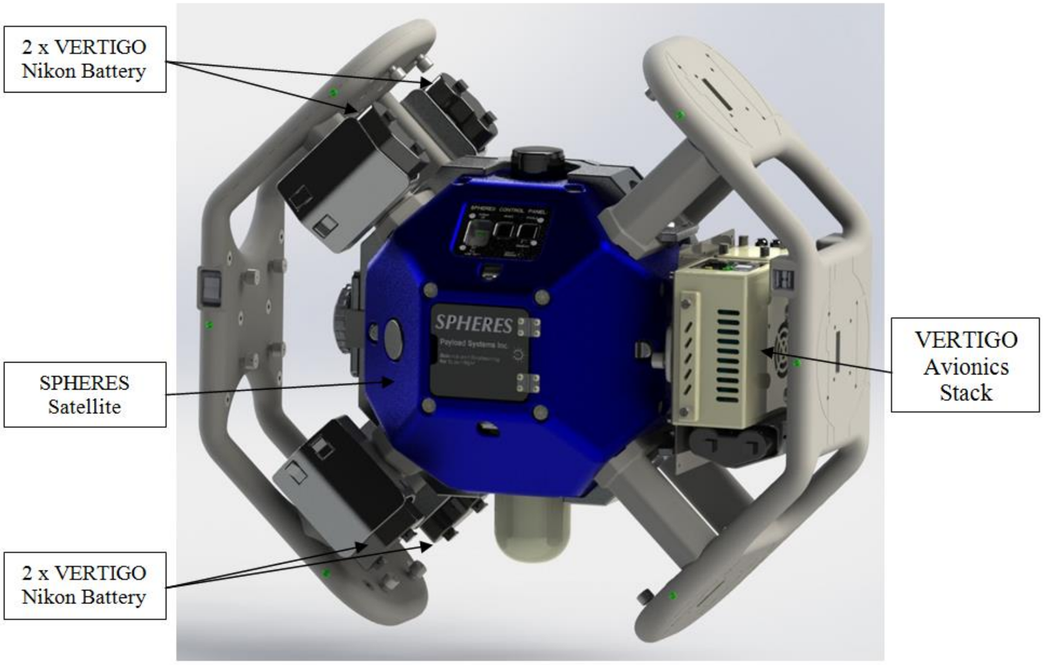


Figure 25: Halo System Mounted on SPHERES/VERTIGO Assembly

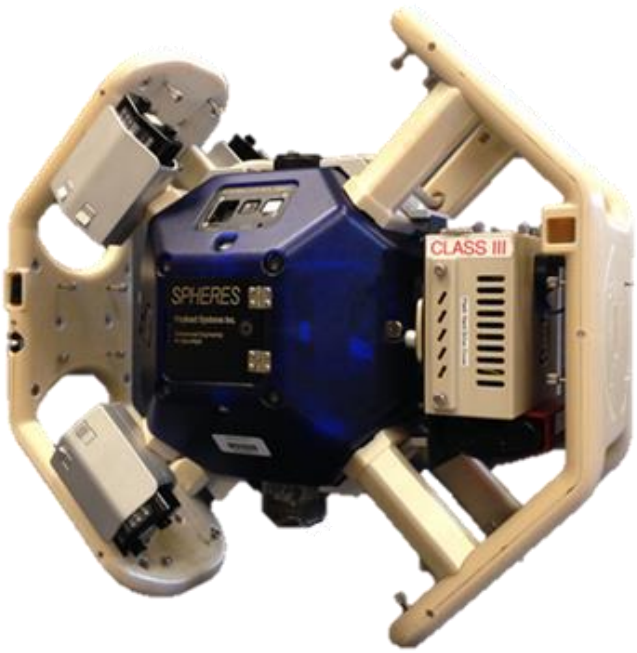


Figure 26: Functional Halo Prototype

The addition of Universal Docking Ports is a critical upgrade to the SPHERES facility aboard the International Space Station. The UDPs add the ability to dock and undock repeatedly, enabling SPHERES to provide a testbed to address many of the challenges of reconfigurable spacecraft. These challenges include performing relative sensing and characterization for docking, adjusting to the new system dynamics of the docked vehicles, and reconfiguring command and control of the new system. The UDPs therefore provide an important capability for future studies of reconfigurable spacecraft for new mission architectures, including on-orbit robotic servicing and assembly.

Figure 27 and Figure 28 show SolidWorks models of the UDP, and Figure 29 shows the functional prototype of the UDP when mounted to the VERTIGO Avionics Stack, the initial testing configuration for ISS operations.

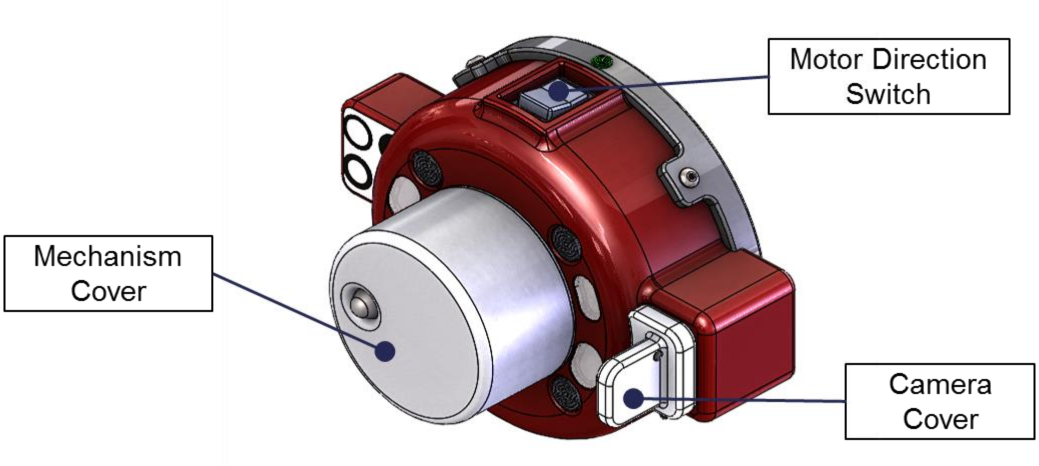


Figure 27: SolidWorks Model of UDP with Protective Pieces Installed

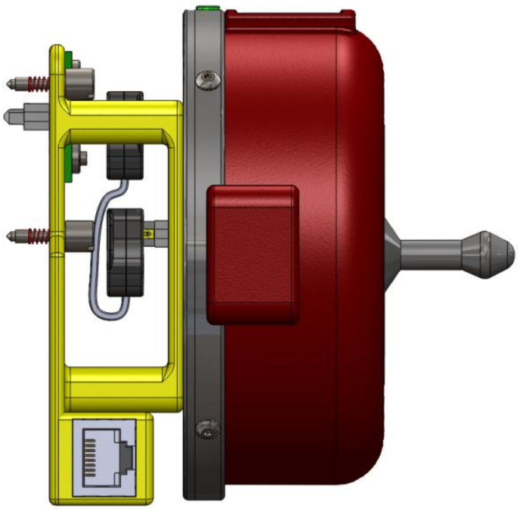


Figure 28: SolidWorks Model of UDP with VERTIGO Avionics Stack Standoff

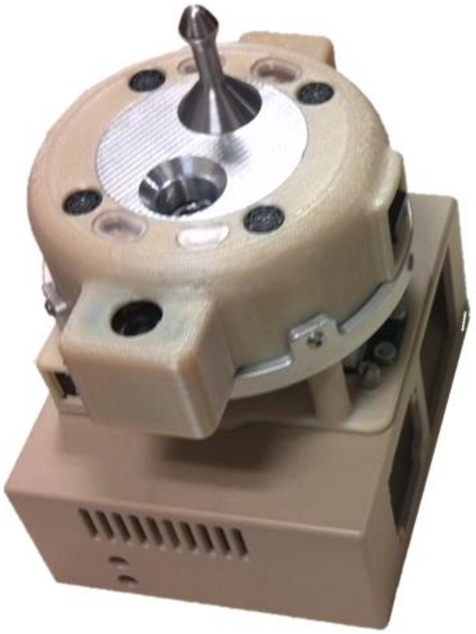


Figure 29: UDP Prototype Mounted to VERTIGO Avionics Stack (Photo Credit: Duncan Miller)

In order to best support the risk reduction of RSA mission technologies, the UDP is designed with both mechanical and sensor functions. These functions, further described in the following paragraphs, enable the SPHERES satellites to perform reconfiguration on orbit while also enabling ground testing to occur with identical docking ports to those aboard the ISS.

The objective of the UDP is to provide a mechanism for rigidly docking and holding two SPHERES satellites or other space elements together. There may be relative motion between these elements prior to rendezvous and docking, so the docking mechanism within the UDP must be able to maintain rigid capture despite forces and torques acting on and across the docking interface. The interface itself must be rigid so that it does not add additional dynamics to the system.

Furthermore, the UDP must provide a capture cone for docking two SPHERES satellites together. This cone allows for slight misalignment in the orientation of two SPHERES satellites approaching docking while still locking into a set position. Correcting for slight misalignment ensures that the intended interface between the satellites is consistently established.

While both SPHERES satellites have global metrology, the UDP must provide direct sensing between the two docking interfaces. This capability, provided by an onboard camera and visual fiducials, allows the SPHERES satellites to assess their relative pose in the approach phase prior to docking. Using relative sensing to supplement global metrology for docking replicates the approach a robotic servicer would take and provides a more realistic testing scenario. The camera need not be used in a docking maneuver.

Additionally, because the UDPs are mounted either directly to the VERTIGO Avionics Box or to the Halo, both of which block four of the Position and Attitude Determination System (PADS) ultrasound sensors, the UDP must replace these blocked sensors to maintain maximum global metrology capability. These metrology sensors are expected to be operated whenever the UDP is powered.

Figure 30 shows two UDPs attached to the Halo. As can be seen, the UDPs can be placed on any of the six Halo ports, enabling extensive reconfiguration for satellite assembly and servicing mission scenarios. This reconfiguration is limited, however, owing to the required orientations of the UDP when mounted to the Halo or directly to the VERTIGO Avionics Stack. In order to minimize the mass and inertia increase created by mounting the UDP far from the center of mass of the SPHERES satellite, the UDPs maintain a fixed docking orientation with respect to the SPHERES satellite’s body axes. Future peripherals can be developed which would enable docking at various angles, such as with gimbals or other rotary sections.

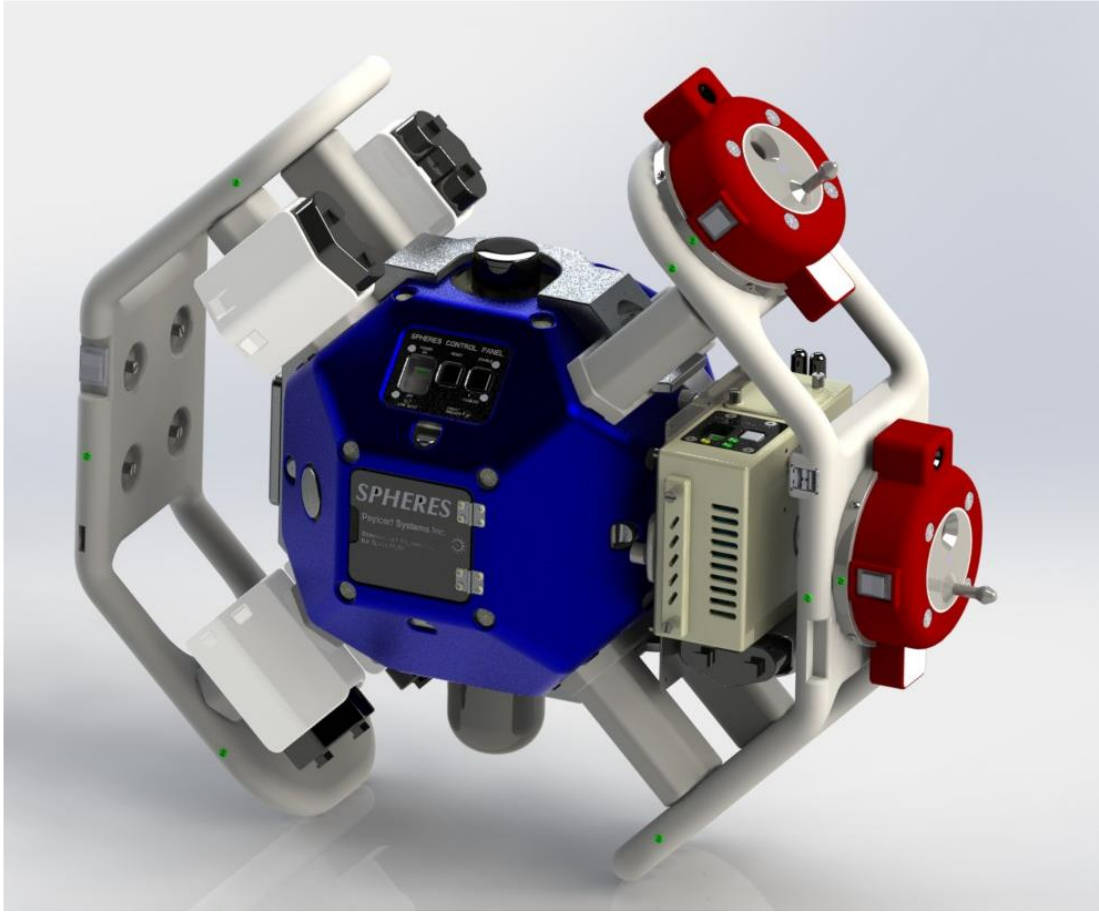


Figure 30: Two UDPs Mounted to Halo

A total of three Halo structures and six UDPs will be launched. With this up-mass, it is possible for each of the three satellites on orbit to be fitted with a Halo and two docking ports. These UDPs can be arranged as necessary by the astronauts according to the test descriptions, allowing multi-satellite rendezvous and reconfiguration architectures to be tested. This flight hardware extension to the SPHERES satellites therefore provides the needed expansion of the SPHERES facility on orbit to reduce the risks associated with close proximity operations and satellite-to-satellite interactions via docking and undocking. The ability to repeatedly dock and undock affords RSA risk reduction aboard the ISS, which enables testing in the long duration microgravity environment that only the ISS can provide.

3.5 Summary of Final RSA Testbed Design

The final testbed design incorporates the elements of the SPHERES satellites, SWARM Propulsion Units, Halo structures, and Universal Docking Ports. This set of hardware, along with the associated software, control laws, firmware, and operational procedures, provides a strong baseline with which to conduct testing for the risk reduction of RSA missions.

The SPHERES facility provides a well-established series of satellites and a code base which affords ready access to both ground and ISS testing environments, but also to an easily modifiable and expandable code core. The basic functionality of the SPHERES satellites can be changed according to the particular testing which must be performed through software, reducing the time and monetary costs associated with testbed development. Further, the ability to test on satellites on the ground which are identical to those aboard the space station provides immediate feedback for making low cost improvements or fixes to flight code and hardware, further reducing the risk of RSA missions: ground testing with identical equipment to that which is on orbit reduces the risks associated with a risk reduction flight test.

The hardware that is used to test on the ground is designed to provide similar functions as that which is designed for flight operations. The SWARM Propulsion Unit allows the SPHERES satellites to dock and undock with one another, a first step towards multi-satellite reconfiguration. This hardware, however, is limited to ground operations on the MIT Flat Floor; it cannot be used aboard the ISS. The additional actuation capability created by the 16 thrusters on the unit helps the satellites perform controller reconfiguration and initial physical satellite reconfiguration on the ground. New hardware is required to further this technology maturation beyond the Flat Floor.

Additional reconfiguration capability is afforded with the Halo structure. Its six Halo ports provide a method for rigidly securing multiple peripherals to each SPHERES satellite, whether on the ground or aboard the ISS. The development of the Halo follows on the work conducted through ground testing, since the Halo is designed to be fully operable aboard the ISS, but with only marginally reduced functionality during ground testing. For example, the limitations imposed by 1g dynamics and the need to operate in a 2D realm limits the possible clocking angles between two satellites as they maneuver around each other. Further, the lack of the vertical testing direction prevents a large number of possible close proximity maneuvers which could be tested aboard the ISS. These limitations are expected, so the Halo was designed to maximize the ground-feasible multi-satellite reconfigurability architectures that it could enable. Flight testing can then perform verification and validation of these ground-tested maneuvers before conducting testing of architectures or maneuvers which cannot be fully replicated on the ground. The ISS is therefore a crucial testing environment that is required to reduce the risks associated with many new RSA technologies for low costs while maintaining traceability to the final flight architectures of RSA missions.

The UDP provides a universal method for providing a rigid connection capable of transferring loads from one satellite to another when mounted to either the VERTIGO Avionics Box or a Halo Port. The ground UDP, a prototype for the flight docking ports shown above, has provided extensive testing and feedback for the design of this flight UDP design, thereby allowing design iteration for optimization of features against several cost drivers, namely mass, system center of mass, and fine alignment. Further, the decision to incorporate a camera for relative pose sensing instead of relying on the prior rendition of an active metrology ring greatly expands the traceability of the UDP to current and future docking systems.

Key design properties for both of the flight hardware elements, the Halo and the UDP, are listed in Table 4 and Table 5. These properties show how the addition of mass and inertia to the SPHERES satellites will limit the effectiveness of the thrusters when operating on-orbit. These limitations illustrate the importance of the additional actuation capability provided by the sixteen thrusters on the SWARM Propulsion Unit, which must also support the UDP ground prototype. These properties show the flight values of both the Halo and UDP, so the controllers used on the SPHERES satellites are tuned to function with these updated values. Additionally, it should be noted that these properties have changed extensively over the design process for both elements, highlighting the iterative and incremental nature of design. The iterative, incremental testing process is further discussed in Chapter 4, and further details on the design of the flight Halo hardware can be found in McCarthy (2014).

Table 4: Principal Halo Properties

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

Table 5: Principal UDP Properties

UDP Property	Value
Dimensions (UDP)	15.5 cm x 11.5 cm x 8cm height
Dimensions (UDP Standoff)	10 cm x 10 cm x 5 cm
Protruding Pin Length	3.2 cm
Entrance Hole Diameter	1.4 cm
Unit Mass (UDP)	0.47 kg
Unit Mass (UDP Standoff)	0.04 kg
Maximum Docking Range	~10 cm
Time for Final Capture	2-5 seconds
Time to Lock	<2 seconds
Camera Mass	12g
Camera Resolution	2592 x 1944
Camera FPS	30
Camera Shutter	Rolling

4 Chapter 4 – Incremental Testing of RSA Technologies

4.1 Introduction to Testing Process

Two concepts significantly influencing a meaningful risk reduction testing process are those of testing increments and testing iterations. A testing increment is a step forward in a testing process or sequence that builds upon the work conducted in earlier testing, either from prior renditions of a given test increment or from prior increments themselves. Iteration, however, can be defined as repeatedly conducting the same test in repeatable test conditions to obtain sufficient data to either confirm or refute the hypothesis for a given test iteration. As a result, a single increment should incorporate multiple iterations in order to ensure statistically relevant results, regardless of the technology that is being tested.

The testing process detailed in this chapter focuses on creating a sequence of incremental tests, each conducted with several iterations. Multiple increments are described due to the extensive testing required for reducing risks associated with RSA missions. Consequently, the testing process is designed to rely on the new testbed described in Chapter 3 and to make the most efficient use of the hardware and software available to utilize the existing infrastructure. Not only can the existing infrastructure of the hardware and software elements described in Chapter 3 be used to reduce time and monetary costs associated with testing each increment, but the infrastructure from one increment to another can be reused if appropriately designed. As will be seen, the same fundamental setup can be reused with only slight modifications to the test articles or the testing environment. A primary goal of the testing, therefore, is to ensure that costs are reduced by limiting the number of testbed changes from one increment to another.

The criteria for determining when an increment is complete are more subtle than the criteria for determining when a particular iteration is complete. Iteration is typically deemed to be complete at the conclusion of a test, with the test results indicating whether or not the test objectives were met and if the test was able to meet its success criteria. The test success criteria are determined prior to the start of each test increment, and form the basis of comparison against which data can be measured. This success criteria-data correlation is the determining factor for when an iteration may be deemed complete: once sufficient data is obtained to allow the success criteria to be scrutinized, an iteration has completed. An increment, however, is composed of a sequence of iterations. Consequently, iteration criteria are related to the test criteria, though they typically are broader in scope. Each iteration may be deemed complete when the cumulative testing results across all iterations demonstrate that the test objectives were met. The increment may be deemed unsuccessful, however, if the collective data warrants revision of the increment objectives, test setup, or success criteria. Additionally, an increment should be devised to make best use of the prior work, but also to provide a starting point for the next increment. This incremental and iterative process has been applied to the task of reducing on-orbit RSA technologies in the subsequent sections of this thesis.

It is highly important to plan the sequence of tests appropriately prior to beginning testing of the first increment. The process for determining the test objectives for each increment is broad by necessity: each program will require its own set of unique testing increments. Generally, a fewer number of testing increments will lead to reduced testbed costs, since fewer testbeds need to be developed. In order to obtain the same amount of risk reduction data, however, an increased number of tests may be required to compensate. These additional tests may then increase the total testing duration of the test campaign. Therefore, a proper balance must be struck.

For the RSA testbed described in this thesis, a sequence of testing increments was developed to mature the testbed and the corresponding capabilities of the RSA technologies being tested progressively. This sequence is described in Figure 31.

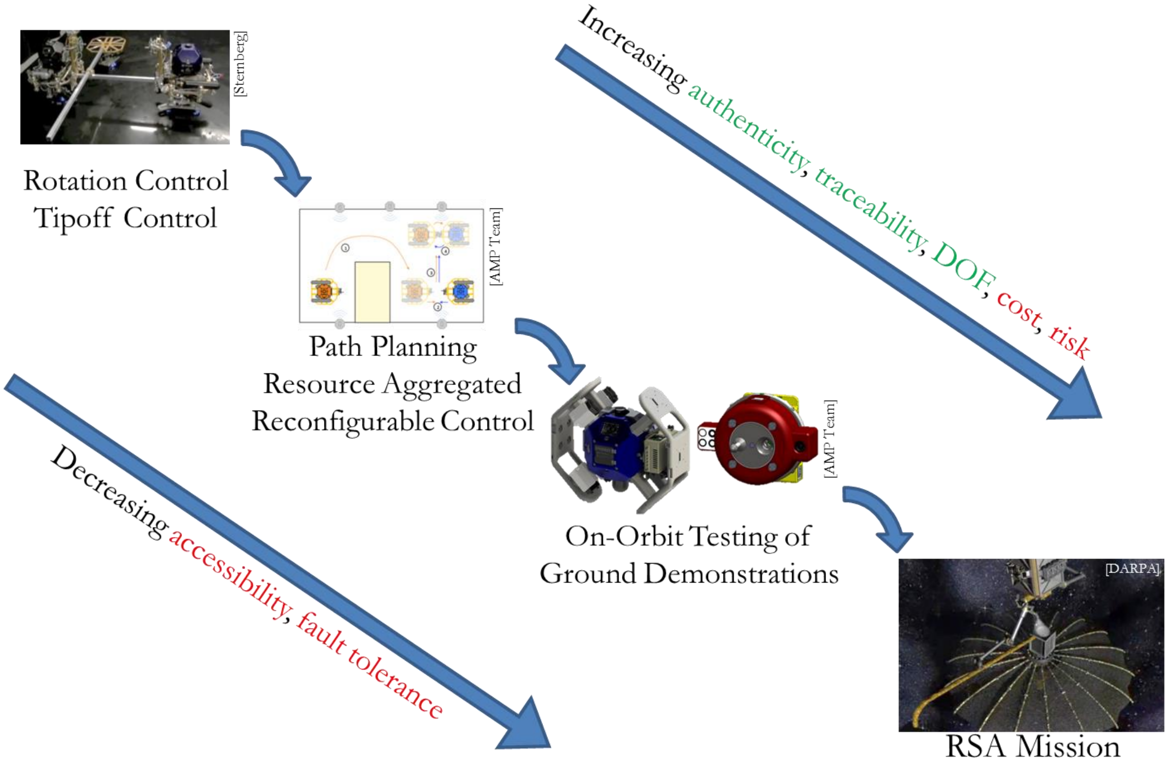


Figure 31: RSA Testbed Testing Increments

As can be seen in the figure, there are three testing increments prior to the implementation of the various technologies in the final RSA flight mission. The sequence methodically transitions from ground to flight testing, with the third increment conducting the first on-orbit testing. The sequence shows that test increments build upon one another so that the information gained from one increment can be used as the basis for the next. Further details about these increments are discussed at length in the following sections of this chapter. It is important to select these increments carefully, because with each successive testbed, there are both benefits and detriments. As testing transitions from ground to flight operations, the authenticity, traceability, and number of degrees of freedom which can be tested increases. The cost of the testbed and the risks associated with testing, however, also increase. Simultaneously, moving away from a controlled ground testing environment decreases the accessibility to the particular testbed and the ability of the testbed to cope with various failures. These multiple considerations play a key role in determining the number and scope of each testing increment. The number of iterations required for each increment then is to be determined based on the goals of each increment. For this RSA testbed, the number of iterations varied widely, for example, from conducting a few dozen tests to several hundred. Information from each is used to inform the level of risk which is reduced during a particular increment.

This sequence was designed specifically for the Phoenix case study, as indicated by the Phoenix conceptual drawing as the final increment in Figure 31. The testing is designed to reduce risks associated with the Phoenix mission, though the technologies tested in these increments are applicable to many RSA missions beyond Phoenix. The testing conducted in Section 4.2 was performed in conjunction with Christopher Jewison and Bryan McCarthy, and the testing in Section 4.3 was performed in conjunction with Christopher Jewison and Bryan McCarthy with support from Daniel Strawser and Cheng Fang. The design of on orbit operations in Section 4.4 was conducted in conjunction with Christopher Jewison, Bryan McCarthy, and Duncan Miller. Additional descriptions of the concept of operations and results for the ground testing can be found in Jewison's SM Thesis (2014). Jewison more specifically addresses thruster selection algorithms and the results from implementing these algorithms on the ground testbeds. McCarthy's SM Thesis (2014) more specifically addresses the ground testing control and performance results as a step towards the development of the new flight hardware to expand the SPHERES on-orbit facility.

4.2 Test Unknown Dynamics in 1g

The first increment of testing is directed towards unknown dynamics in 1g. This increment is aimed at demonstrating the ability to develop a controller that can govern the rotation of a spacecraft system on-orbit and eventually be incorporated in a multitude of RSA architectures. Consequently, this increment also provides the first test results for the RSA testbed. In order to reduce the risks of RSA testbeds, the formative stage technology of controlling rotation rates and attitude must be tackled first, since it is one of the most basic capabilities that an orbiting system must be able to perform. The SPHERES facility provides the basis for this ground-only testing, but it is expanded with the SWARM Propulsion Unit to enable reconfiguration and to provide additional thrusting capacity. This hardware arrangement provides traceability to the final RSA mission architectures since each SPHERES satellite is able to function as either a single monolithic satellite module or as a cluster of Phoenix-like satlets. Because each SPHERES satellite can function with its own communication, attitude determination and control, structures, avionics, propulsion, and power subsystems, it is feasible to instantiate various levels of functionality with the hardware through small software changes. Furthermore, the ability to specify through software alone particular master/slave relationships between SPHERES satellites allows for future RSA command and data handling (C&DH) architectures to be simulated.

This testing increment was broken into two phases, each aligned with the goal of demonstrating the ability for multiple spacecraft to control cooperatively the attitude of satellite aggregates that are rigidly docked together. The ability to control an entire system of docked spacecraft on the ground is the first step in an extensive sequence of incremental, iterative tests to reduce the risks associated with RSA missions. In this testing increment, all of the satellites are rigidly docked together to create an unknown but fixed set of physical dynamics properties. This simplification provides a basis for comparison against future, more complex testing both in 1g and in microgravity. The 1g Flat Floor facility was used as the testing environment in order to allow rapid testing iterations with very low risk to the hardware as software changes are made. Throughout this testing, very little risk was invoked, despite providing valuable testing results that reduce on-orbit RSA risks. The first phase focused on the demonstration of rotation about the servicing or assembly system's center of gravity, while the second phase focused on the demonstration of tipoff control. Much of the information, including figures, for this section was derived from the test report written primarily by David Sternberg for AFS (Phoenix Testing Summary, 2013) on work conducted as a collaboration between David Sternberg, Bryan McCarthy, and Christopher Jewison.

4.2.1 Phase I Testing: Rotation Control

The goal of the first phase of this first testing increment was to demonstrate that multiple satlet clusters or small satellites could provide the attitude control of a repurposed satellite, satellite under service, or one being assembled by initiating and completing a rotation about the center of mass of the satellite system. A master/slave relationship was used between two SPHERES satellites to mimic a proposed C&DH structure, and each SPHERES satellite functioned as a satellite or satlet cluster capable of providing actuation, sensing, and C&DH capabilities.

Owing to the large expected size of RSA systems, it was necessary to scale down the demonstration to fit within the confines of the largest ground environment, the MIT Flat Floor facility. This Flat Floor is a smooth, hardened epoxy, octagonal testing environment that permits the SPHERES satellites to move nearly without friction in 2D motion. Additionally, each satellite communicated both with each other and with a ground station computer that runs the testing at the Flat Floor, and an ultrasonic metrology system is in place for providing the satellites with state updates throughout each test in reference to a user-defined coordinate system.

This phase of testing joined two SPHERES satellites along with a representative test mass through a rigid mounting system, shown in Figure 32. As can be seen, the two satellites are mounted aboard propulsion units to provide additional actuation capacity in the form of 16 additional thrusters. This SWARM Propulsion Unit is mounted atop an air puck that allows compressed CO_2 to provide the nearly frictionless surface upon which to float for 1g testing. A proof mass consisting of the base aluminum frame of another SWARM Propulsion Unit is attached to another CO_2 carriage. This proof mass represents a piece of space hardware that does not have any actuation or sensing capabilities of its own, and must therefore rely on other spacecraft for orbit maintenance and general operations.

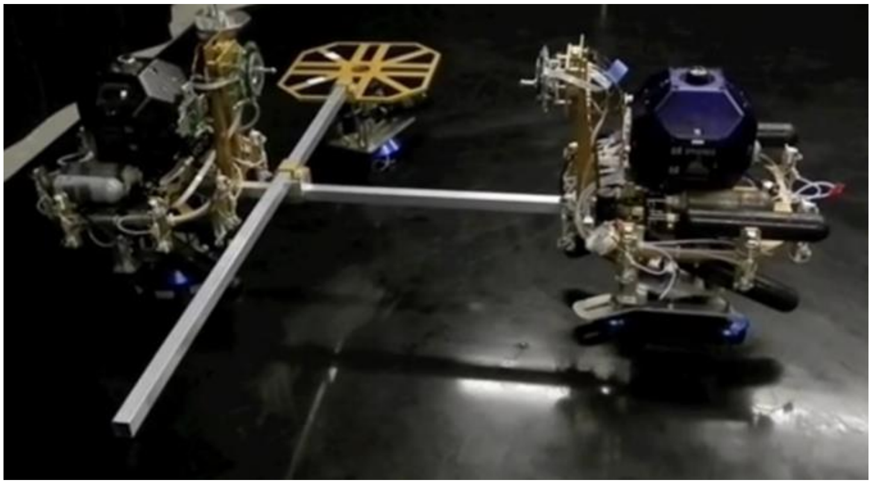


Figure 32: First Testing Increment Hardware Setup (Phoenix Testing Summary, 2013) (Photo Credit: David Sternberg)

A rigid aluminum beam assembly connects the carriages together. Importantly, the central joint between the orthogonal aluminum beams is adjustable, allowing the center of mass, and therefore the inertia properties of the system, to be adjusted as needed for a given test prior to test start. With this setup, the location of the proof mass relative to the two SPHERES satellites can be adjusted along the three foot long main beam between the satellites and along the four foot beam that connects the proof mass to the main beam.

The full Flat Floor setup can be seen in the schematic shown in Figure 33. The proof mass, *M*, is located at a user-defined point, as both lengths *A* and *B* are adjustable. As shown in this figure, the two aluminum beams provide a nearly perfectly rigid connection between the three elements to retain the center of mass of the system in a nearly constant location relative to each of the principal masses, i.e., the two satellites and the proof mass. The ground station computer is responsible for receiving the state information from each satellite, representing human supervision from a ground station. The firing times for each satellite to actuate its thrusters were calculated aboard each satellite. In this manner, a possible C&DH structure and satlet aggregation structure are incorporated into the demonstration for increased traceability towards eventual RSA missions.

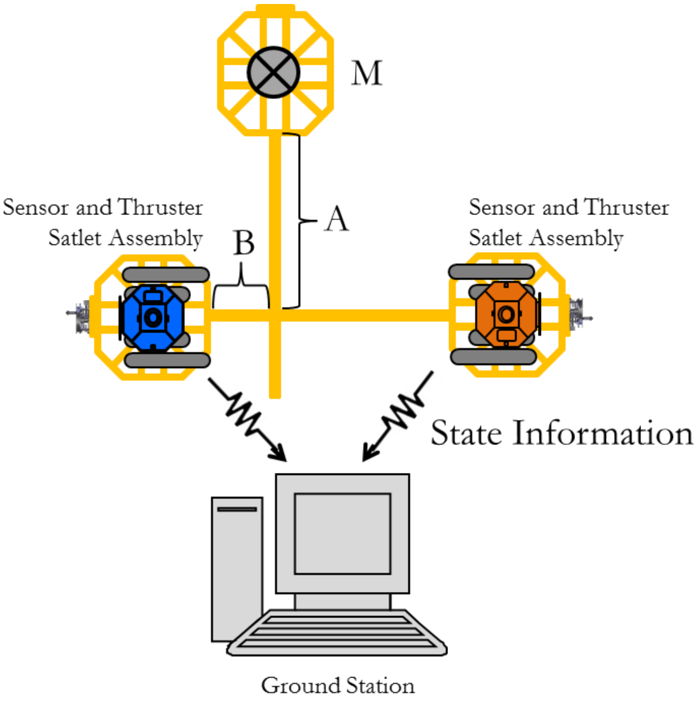


Figure 33: First Testing Increment Schematic (Phoenix Testing Summary, 2013)

The software written for this phase enables the demonstration of rotation control about the system’s center of mass. A simple open loop controller was developed to calculate the firing times for each of the thrusters; a total of 12 thrusters innate to each SPHERES satellite are controlled in concert with the 16 thrusters aboard each SWARM Propulsion Unit for a total of 28 thrusters per satellite controller. An open loop controller was selected to allow the satellites to rotate easily about the system center of mass, since equal and opposite firing times can be commanded to attain the desired rotation owing to the relative positions of the two SPHERES satellites as seen in Figure 32 and Figure 33. Additionally, it is possible to use this thruster firing scheme regardless of the position of the proof mass. Two different proof mass configurations were tested. In the symmetric case, the proof mass was fixed midway between the two SPHERES satellites and 40 inches from the main beam connecting the satellites. In the asymmetric case, the proof mass was fixed 20 inches from the main beam and 15 inches from the SWARM Propulsion Unit of the primary satellite. Since there are no constraints on the time duration in which to perform the rotations, the firing times were kept constant across all testing configurations and iterations.

Figure 34 shows the angular displacement vs. time for a typical iteration when testing the asymmetric case. Unfortunately, although many of the SPHERES data files for the symmetric case iterations were either broken or invalid, similar results were observed in video recordings of the tests. As can be seen in the figure, the open loop controller to command thruster firings on both satellites to both initiate and stop a rotation in order to attain a desired angular displacement of 90deg from an initial position-hold maneuver. A position-hold was also commanded after the rotation completed. At this point of the test, the standard position-hold algorithm was able to maintain the attitude of the combined system in an orientation orthogonal to the initial orientation with little overshoot, which itself is due to the open loop controller not accounting for errors that arise over the course of a particular command and control sequence. Based on this data, it can be said that this first phase's testing was able to rotate the system from an initial orientation about its center of mass to a new attitude and, once attained, maintain that attitude. With this knowledge it is possible to proceed to the next phase of testing by building on this capability. This increment demonstrated basic rotation control; the next step is to regain control of a system spinning at an initially unknown rate.

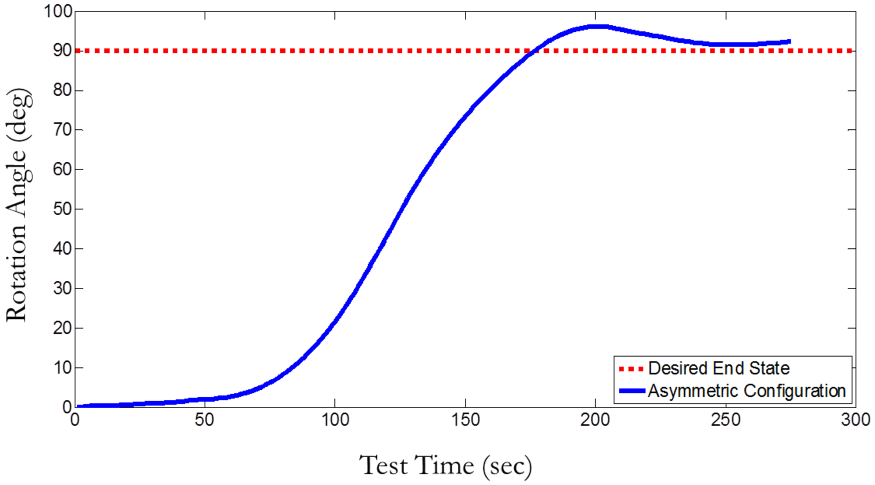


Figure 34: Angular Displacement vs. Time (Asymmetric Case) for Phase I Testing (Phoenix Testing Summary, 2013)

4.2.2 Phase II Testing: Tipoff Control in 1g

The second phase of this increment demonstrates an advancement of the control testing that occurred during the first phase. With the ability to both initiate, control, and complete a rotation to a set attitude having been demonstrated in Phase I, Phase II is aimed at demonstrating the ability to halt motion induced by a tipoff offset at an initially unknown rate. Though tipoff scenarios are not expected to be common during RSA missions, the ability to regain control after a tipoff is crucial to ensuring mission success, since tipoffs can induce large rotational and translational disturbances into the system. For example, a robotic arm from a servicer or assembler spacecraft could accidently hit the spacecraft being serviced or assembled prior to completion of the RSA process, leaving the number, orientation, or operability of actuation and sensor capabilities uncertain. As a direct result of this risk, the assembly process is expected to be conducted slowly and methodically so that a tipoff event would only upset systems with an educated estimate of system performance capabilities.

To reduce the risks associated with tipoff events even further, Phase II testing incorporates the same hardware and software configuration from Phase I testing in order to maintain traceability to future RSA missions and to utilize heritage configurations through the use of existing infrastructure. Therefore, Figure 33 shows the same configuration of two SPHERES satellites connected via an adjustable position, rigid aluminum beam assembly to a proof mass air puck with commands being handled through the Flat Floor's ground station computer. Additionally, the same two configurations were used: a symmetric and an asymmetric configuration with the same lengths between primary mass elements. In order to create initial tipoff rates that are unknown to the controller, no assumptions were made about the initial rotation rates during code writing, leaving the controller to stop the tipoff rotation with the knowledge only of the thruster actuation capabilities of the SPHERES satellite plus SWARM Propulsion Unit system.

A new controller developed for Phase II testing to accommodate the lack of initial condition knowledge was needed, so direct adaptive control methodologies were incorporated into the controller. With direct adaptive control, unknown parameters do not significantly degrade the performance of the system, leaving the system to be controllable without the necessity to perform a full system identification process prior to beginning the tipoff control maneuver. Additionally, the adaptive nature of the controller provides robust performance to changes in the system configuration. Consequently, the direct adaptive controller relies on the computation of gains based on deviations from a given reference model of system dynamics, rather than system mass and inertia properties. This direct adaptive controller therefore affords the future RSA system the ability to react immediately to tipoff events without having to identify the particular physical parameters of the system or the rotation rate of the system as a result of tipoff.

Additionally, RSA missions may not have the luxury of precise state information at the location of either servicing or assembly. Since the SPHERES satellites are able to determine their orientation using an Extended Kalman Filter (EKF) with data sources from both an Inertial Measurement Unit (IMU) and ultrasound metrology system, it is possible to ignore data from the metrology system and only use gyroscopic data for determining rotation rates. This limitation more accurately represents the scenario where position data is not available to RSA spacecraft on orbit. By relying solely on SPHERES satellite gyroscopes, however, the testbed does not have translational knowledge during testing. As a result, translation in the plane of the Flat Floor is left uncontrolled during this phase of testing.

Figure 35 and Figure 36 show the simulated time response and tracking errors for the direct adaptive controller that performs computations with both the standard SPHERES satellite mass of 4.16kg and with twice that mass, 8.32kg. The robustness of the controller is shown by comparing the two plots: the error between the actual performance and the reference model follows the same trend, and the magnitude of the error does not appreciably change based on the physical properties of the RSA system. It is important to note that the reference model was not changed across these two scenarios, so the actual performance of the controller is a produce purely of the change in the system mass variable. These figures therefore show that risks associated with tipoff and unintended input forces and torques to a non-fully-functional system can be reduced by incorporating direct adaptive control methodologies when developing controllers for RSA system. The direct adaptive controller developed here in simulation demonstrates in this increment of testing across a multitude of iterations that controllability of an RSA system can be maintained even when the mass and inertia properties vary widely, such as when multiple elements are being assembled to produce new space systems on orbit.

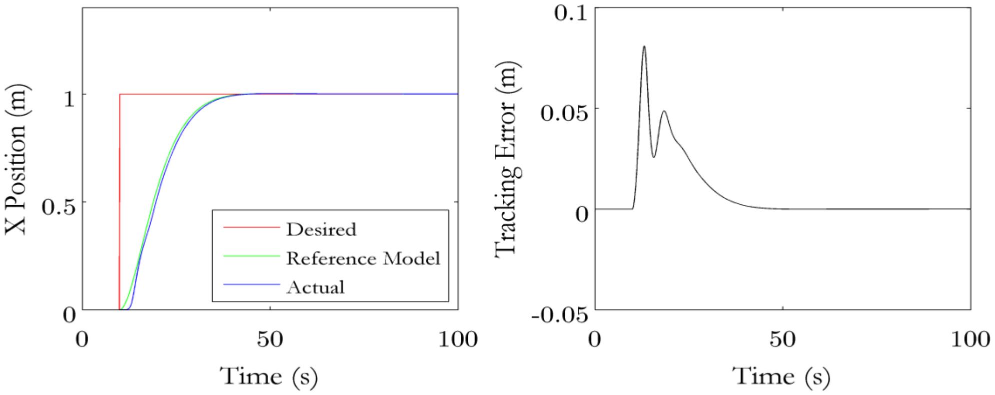


Figure 35: Time Response and Tracking Error to Step Input for Direct Adaptive Controller (Phoenix Testing Summary, 2013)

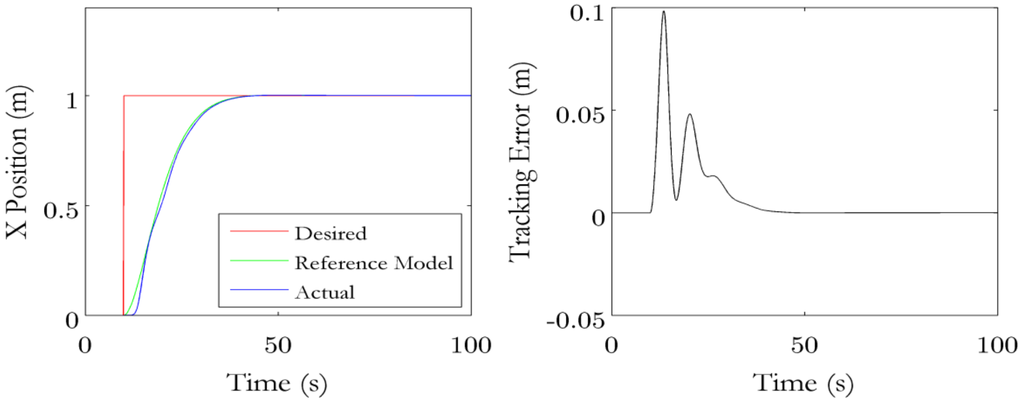
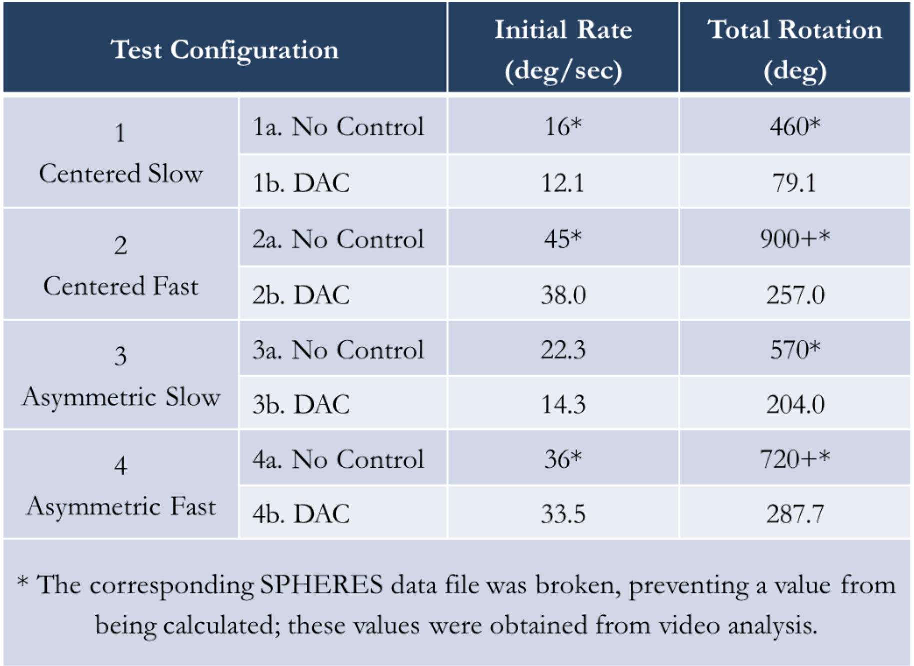


Figure 36: Time Response and Tracking Error to Step Input for Direct Adaptive Controller with Doubled System Mass, but Controller has No A Priori Mass Change Knowledge (Phoenix Testing Summary, 2013)

A series of four testing configurations were demonstrated on the MIT Flat Floor during this second phase of testing, thereby utilizing the iterative and incremental nature of the testing facility. In order to demonstrate the robustness of the direct adaptive control algorithm, both the center of mass and the initial rotation rate were adjusted. Therefore, utilizing two center of mass locations (symmetric and asymmetric) and two categories of initial rotation rates (fast and slow), a total of four possible combinations span the tipoff control tradespace. The symmetric and asymmetric cases are located in the same configurations as in Phase I, and the initial tipoff rate was created by spinning the entire system by hand at the fast and slow rates. As a result, the tipoff rate was approximated through hand rotations, ensuring that no assumptions could have been implemented into the controller as to what these initial rates could be. All inconsistencies in either the center of mass location or initial tipoff rate that are introduced into the system therefore provide further evidence as to the direct adaptive controller's robustness. For example, as will be seen, the variations in initial rate caused by inconsistencies in hand spinning the system did not result in uncontrollability of the system.

Although the Flat Floor environment is nearly frictionless, friction still plays an important role in ground dynamics. In order to show that the rate dampening with the direct adaptive controller was a result of controlled thruster firings and not merely a result of surface friction, each configuration was tested twice: once without controller input where the system was free to settle solely because of friction effects and once where the direct adaptive control algorithm was free to function normally through commanding thruster firings. The ability to test these configurations rapidly shows the iterative nature of the test facility. The difference between these two cases is shown in Table 6. Since the total rotation is much greater in the No Control cases than in the Direct Adaptive Control (DAC) cases, it can be said that friction was not the principal mechanism by which the initial tipoff rate was reduced to naught. Unfortunately, owing to broken SPHERES data files, some of the exact values in each column could be not obtained from SPHERES telemetry. For these values, approximations are shown which are derived from analysis of video footage from each test. Regardless, the trend that the total rotation angle without the use of thrusters and the controller is significantly higher than with the use of thruster firings. Additionally, there were a few tests as indicated by the +, such as Centered Fast No Control, where the total rotation that was achieved by the system was in excess of the listed value. This outcome is a result of needing to prevent the system from impacting the edge of the Flat Floor. In these cases, which were only evident in the No Control cases, the system was still rotating at the time the test was prematurely ended to prevent damage to the system. Had the test not been ended early and the system been allowed to continue rotating, an even greater total rotation angle would have been recorded. Since these were No Control tests, it can be said that the friction on the Flat Floor was not the dominant factor in reducing the system's rotation rate, and that the direct adaptive controller was the primary mechanism by which the system's rotation slowed to a halt after the tipoff.

Table 6: Initial Rates and Total Rotations during Phase II Testing (Phoenix Testing Summary, 2013)



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.

Using the same naming convention found in Table 6, Figure 37, Figure 28, Figure 39, and Figure 40 show the z-axis rotation rate for the four direct adaptive controller test configurations tested in Phase II. In each of these figures, the initial angular rotation rate corresponds to the initial rate listed in Table 6, and the SPHERES data and reference model used to create the controller are plotted. As can be seen in Figure 37, the system was able to track the reference model well with the direct adaptive controller. The error between the plant and reference model is due to the aggressive nature of the model, which was created to damp the system to no angular rate within 25 seconds, regardless of whether the system was in the symmetric or asymmetric configuration, or whether the system was spun at a fast or a slow initial rate. Slowing down the reference model by increasing the time constant would have allowed the system to track the reference model more closely. Figure 39 shows the control system working well in the asymmetric configuration, though with the expectedly larger error between the data and reference model.

Figure 38 and Figure 40 show that the controller was able to dampen the system's spin at even high initial rotation rates, though the disparity between the actual and reference model is increasingly large. Future work on the control system therefore could include adjusting the reference model and further tuning the direct adaptive control parameters to improve the correlation between the data to the reference mode. Since this test was just a single increment, it would be possible to conduct further testing iterations with another increment with an improved controller.

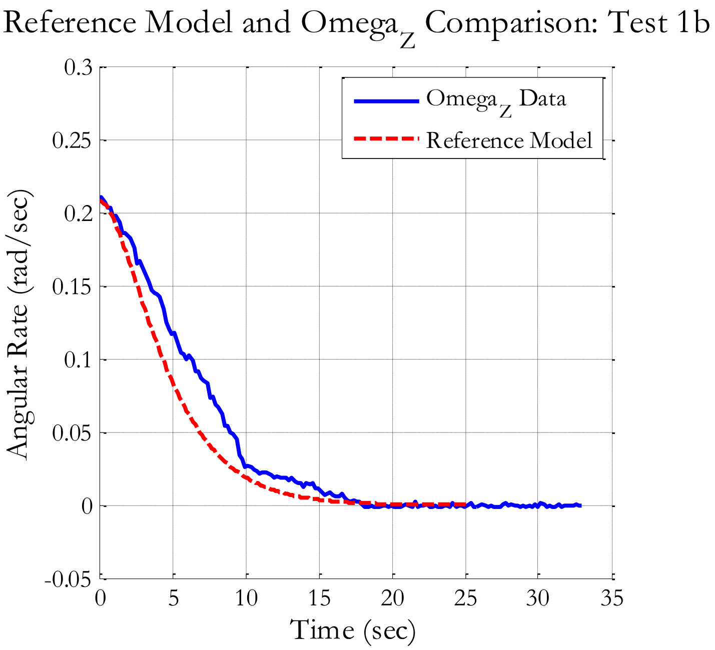


Figure 37: Reference Model and Z-Axis Rotation Rate: Test 1b (Phoenix Testing Summary, 2013)

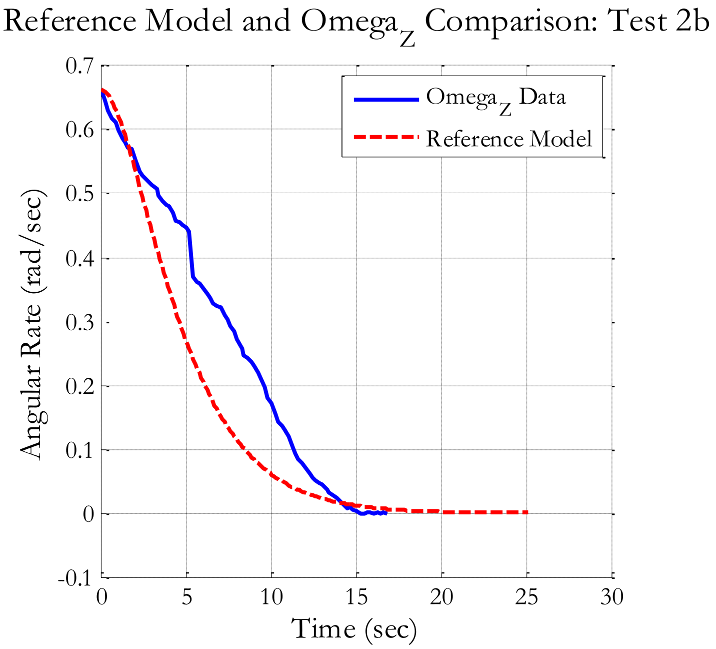


Figure 38: Reference Model and Z-Axis Rotation Rate: Test 2b (Phoenix Testing Summary, 2013)

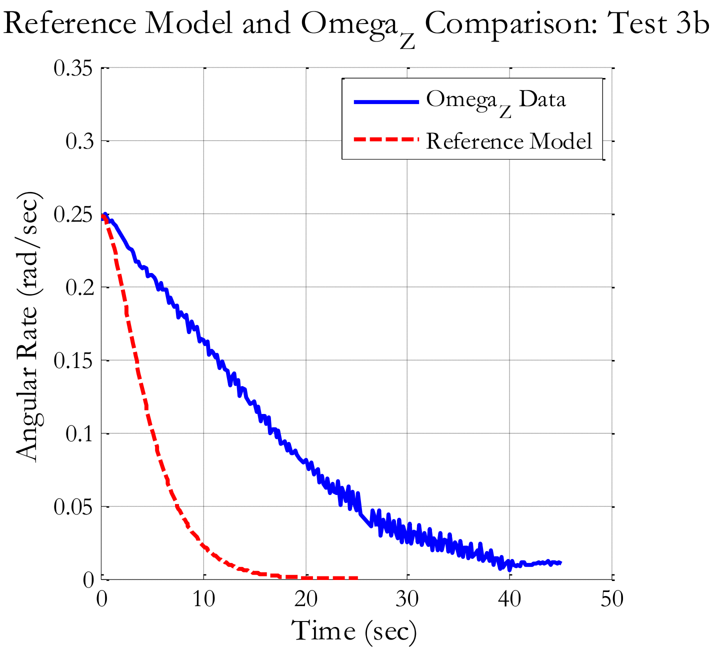


Figure 39: Reference Model and Z-Axis Rotation Rate: Test 3b (Phoenix Testing Summary, 2013)

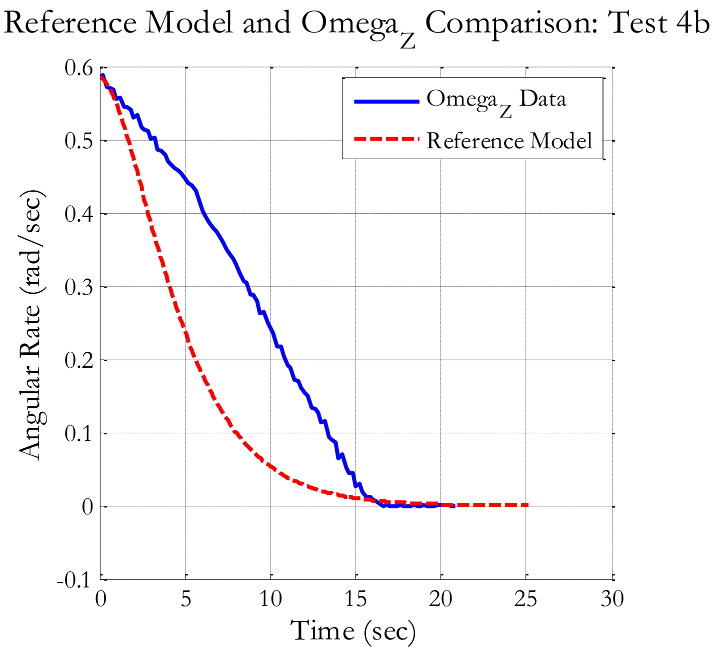


Figure 40: Reference Model and Z-Axis Rotation Rate: Test 4b (Phoenix Testing Summary, 2013)

4.2.3 Unknown Dynamics in 1g Conclusion

The increment which demonstrated the testing of unknown, yet fixed, dynamics in 1g environments consisted of two successive phases, each incrementally contributing to the risk reduction of RSA missions. Phase I demonstrated the ability for clusters of satlets or satellites to initiate and complete rotations of a satellite under servicing or assembly required to maintain a particular attitude, and Phase II demonstrated the ability for that satellite to regain control over its attitude after a tipoff event. Additionally, the direct adaptive control algorithm developed for the second phase was able to control the entire assembly's thruster suite to eliminate the tipoff rotation and maintain the resultant attitude successfully.

Further testing can be conducted to expand the scope of these tests in a sequence of further increments, each with multiple iterations to reduce the risks to RSA missions even further. For example, flexible elements can be incorporated into the currently rigid beams connecting the primary mass elements, or a full system identification process can be appended to the end of the Phase II testing to determine the physical properties of the system, including mass, inertia, and thruster availability and performance. It can be concluded that this initial testing increment provided the groundwork for both further ground and flight testing prior to the start of full RSA mission implementation on orbit. The results gathered in this increment can be used moving forward to the next increment as required to make the most use of the facility in the most effective manner as was described in Sections 2.3 and 2.4.

4.3 Test Path Planning and Changing Dynamics in 1g

Another key capability for RSA missions is the capability to cope with changing physical parameters of an orbiting spacecraft system. As space systems grow in size and complexity, formation flight was initially used as a mechanism for enabling system growth, but RSA missions will require a more sophisticated approach in order for such missions to be carried out in a versatile and robust manner. Research into Resource Aggregated Reconfigurable Control, or RARC, is needed to enable RSA missions. In order to mature RARC technologies, which are primarily software based, risk reduction testing must be completed both on the ground and in a representative microgravity environment such as the ISS. Only through rigorous developmental testing in characteristic research environments can RARC technologies be matured to a level at which RSA missions can safely carried out. As will be discussed, the use of the SPHERES satellites in the newly developed RSA risk reduction testbed allows for the advantages of RARC to be acquired.

Further, autonomous path planning is required of RSA satellites in order to determine a safe path from an initial rendezvous location to the final docking position. Safety is paramount in this approach phase, since the relative velocities must be carefully controlled to prevent damage to either satellite involved. With independent controllers, the servicing/assembler satellite must be sure not to endanger itself or the target satellite, which may not be in a known, safe physical configuration. Through this increment of testing, RARC and autonomous path planning will be demonstrated to enable robotic servicing and assembly of spacecraft of much greater size, complexity, and mission performance or scope than the present standard of monolithic satellites. Information for this section, including many figures, was taken from the paper by Jewison (2014) and the corresponding ARMADAS Test Report (2014), which represent collaborative work between Chris Jewison, Bryan McCarthy, David Sternberg, Cheng Fang, and Daniel Strawser.

4.3.1 Resource Aggregation

The aggregation of resources across multiple spacecraft or satellite modules through docking or manipulation has the potential to enable much larger and capable satellites or to enable satellites to regain control of a previously uncontrollable system. Resource aggregation applies to both sensors and actuators, including any sensors for position, attitude determination, and control, and any actuators for satellite positioning, maneuvering, or internal reconfiguration. When two or more modules dock, all resources become shared between them, providing the entire system with a much broader set of tools with which to perform the mission. RARC is the set of control and physical and aggregation algorithms that are required to govern these expanded systems. For example, if one spacecraft has a failure in a sensor or actuator, then another spacecraft can dock with it and use its capabilities to control the aggregate assembly through the use of RARC algorithms.

In order for such a scenario to be feasible, a standard communication schema must be developed. This communication must be common to all RARC-capable satellites in order to share the pertinent information between all spacecraft. Because mass and inertia properties in particular play a crucial role in the effect of actuation, each module to be assembled or serviced must know the mass, center of mass, and inertia properties of all other modules. Additionally, all actuator and sensor performance figures and overall topology must be shared in order to apply forces and torques properly and to estimate the state of the aggregate correctly and accurately. These properties form the most basic set that must be transferred between satellites, though others may also be communicated, including flexibility modes or the particular types of sensors and actuators as discussed in Jewison (2014).

All of these properties must be transferred between every satellite so that a cohesive model of the system can be computed by the satellites. As described in Jewison (2014), the communication schema can therefore be taken to be a business card which can be sent to all aggregating spacecraft. When a new cohesive model is developed, a new business card is created for the aggregate, which can in turn be shared with other satellites. Figure 41 shows an example of what information can be transferred in a prototypical business card. Figure 42 shows an example of how two satellites (modeled as SPHERES satellites in this example) can share business cards to create a new card that is representative of the newly formed docked system.

Business Card

$$\begin{bmatrix} m & \vec{r}_{cg} & \mathbf{I} \\ \mathbf{R}_{act} & \mathbf{D}_{act} & \vec{F}_{act} \\ \mathbf{R}_{sens} & \mathbf{D}_{sens} & \vec{T}_{sens} \\ \mathbf{R}_{\text{interface}} & \mathbf{D}_{\text{interface}} & \vec{\alpha}_{status} \end{bmatrix}$$

Row 1	Mass (scalar) Center of mass (1x3 vector) Inertia (3x3 matrix)
Row 2	Actuator locations, directions (Nx3 matrices) and force magnitudes (Nx1 vector)
Row 3	Sensor locations, direction (Nx3 matrices) and types (Nx1 vector)
Row 4	Docking point locations, directions (Nx3 matrices) and statuses (Nx1 vector)

Figure 41: Example RARC Business Card Contents (ARMADAS Test Results, 2013)

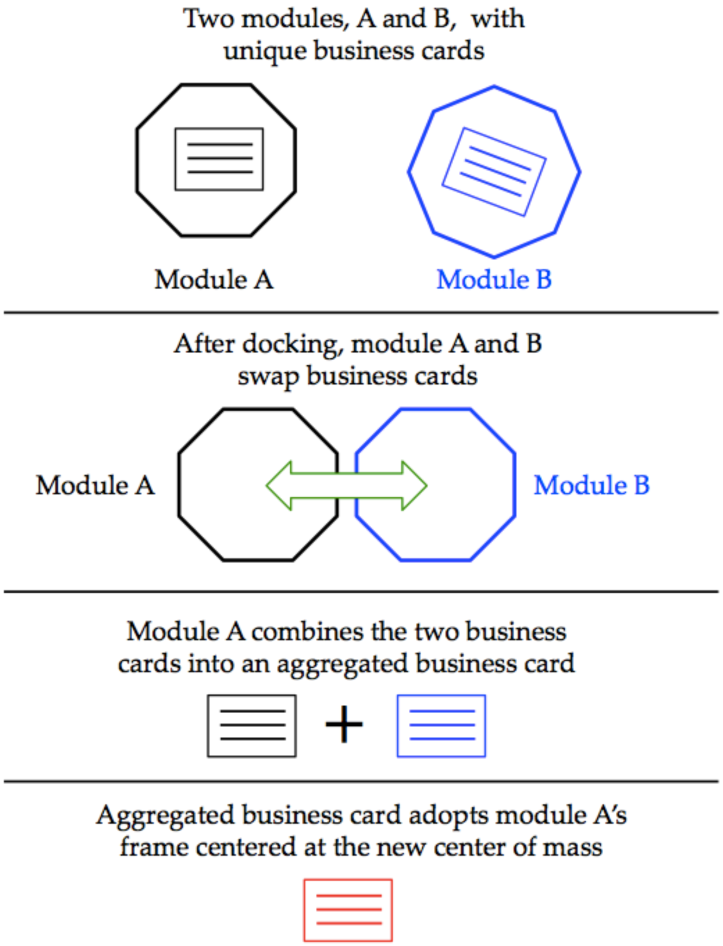

Two modules, A and B, with unique business cards
Module A
Module B
After docking, module A and B swap business cards
Module A
Module B
Module A combines the two business cards into an aggregated business card
+
Aggregated business card adopts module A's frame centered at the new center of mass

Figure 42: Diagram of RARC Business Card Communication (ARMADAS Test Results, 2013)

4.3.2 Reconfigurable Control

The concept of reconfigurable control is based on the requirement that a controller for spacecraft conducting either servicing or assembly will need to be able to maintain control of itself and any docked modules or satellites in real time by changing the controller dynamically. For this increment, the reconfigurable controller was developed for the SPHERES satellites by creating a controller that changes its gains each time a docking or undocking event occurs. It is at these phases of the increment scenario described below that the physical system parameters change. On orbit, however, the mass and inertia properties may change in real time without docking or undocking, as would be the case during a refueling mission or during manipulation steps with a robotic arm.

The reconfigurable controller for this increment need only be capable of controlling the SPHERES satellites in three degrees of freedom due to operation on the Flat Floor, though the controller can easily be expanded to provide full six degree of freedom control aboard the ISS. Under this microgravity and full 6DOF environmental condition set, a full RSA mission scenario could be demonstrated. The controller was developed as a Position-Derivative (PD) controller, which takes as inputs the commanded state and the estimated state from the standard SPHERES estimator.

The error between the commanded and estimated states is multiplied by the PD gains to determine the appropriate forces and torques to apply to the system according to the following standard SPHERES control law:

$$F_x = k_{p,position} e_x + k_{d,position} e_x$$

$$F_y = k_{p,position} e_y + k_{d,position} e_y$$

$$T_z = k_{p,attitude} e_\theta + k_{d,attitude} e_\omega$$

In these equations, 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. These three equations incorporate four gain constants which are based on the physical properties of the system. For the position gains, $k_{p,position}$ and $k_{d,position}$, the selected damping ratio is 0.75 and the natural frequency is 0.2 Hz, but for the attitude gains, $k_{p,attitude}$ and $k_{d,attitude}$, the damping ratio is set at 0.75 with a natural frequency at 0.4 Hz. The equations for each of these gains are shown below:

$$k_{p,position} = \omega_n^2 m$$

$$k_{d,position} = 2\zeta\omega_n m$$

$$k_{p,attitude} = \omega_n^2 I$$

$$k_{d,attitude} = 2\zeta\omega_n I$$

In these equations, ω_n is the natural frequency of the system, m is the system mass, I is the system inertia, and ζ is the damping ratio. As a result, changes in these physical parameters, most notably the mass and inertia, will have an effect on the magnitude of these four gains. These gains are computed in real time autonomously on the RSA risk reduction testbed for a single satellite.

In order to prevent multiple controllers from sending possibly conflicting command information to the system's actuators, only one satellite is given control over the aggregate. For the testing increment described below, one satellite is the master satellite and the other functions as the slave. Both, however, compute all necessary thruster commands for both satellites, but each satellite only actuates based on the role that it plays. This process, while computationally expensive, provides a robust means of checking if all commands are correct, since two independent computers can check their computations.

4.3.3 Autonomous Path Planning

In order for two satellites to dock, they first must arrive at a pre-designated docking point. The process by which these two satellites are assigned a full rendezvous path can encompass various stages, from far rendezvous approach (starting upwards of 100km away from the target) through the near rendezvous approach of a few kilometers, and down to a few meters away during the final docking approach. Orbital dynamics play a major role when the separation distances between the two satellites are large, but play only a small role during final approach. It is during this phase that the two satellites are also in the most danger, since the close proximity raises the risk of collision between the space elements. Certain obstacles may be in place between the docking spacecraft as well, coming in the form of other spacecraft, satellite appendages, or unexpected orbital debris. A path planner is therefore required that can safely bring the spacecraft together so that the rendezvous can occur to enable the servicing or assembly sequence.

p-Sulu, or probabilistic Sulu, is an algorithm developed by Ono [Ono, 2008] to enable fully autonomous path planning around predetermined obstacles. With a given a-priori set of obstacles and target locations, the p-Sulu algorithm computes a set of waypoints for the SPHERES satellites to follow with a guarantee on the probability of success. In order for this set of waypoints to be computed, a model-based executive is built into the p-Sulu algorithm that takes as input the temporally flexible state and to compute a near-optimal control sequence. The p-Sulu algorithm has been shown to be robust to disturbances, to be capable of planning both discrete and continuous actions, and to be able to operate within set, user-defined risk bounds [Ono, 2008] [Ono, 2010] [Ono, 2012] [Ono, 2013].

The p-Sulu algorithm provides a unique perspective into the concept of risk reduction for RSA missions. p-Sulu is a risk allocative algorithm, meaning that it is able to distribute a set amount of user-defined risk across the entire traverse path from initial to final rendezvous location. This user-defined level of risk is representative of the non-zero probability of failure do to stochastic system response. For example, while a command to fire a thruster may be sent, there is a non-zero probability that the thruster not fire, or that another, unintended thruster may fire instead. As a result, the p-Sulu algorithm takes a set probability of failure and allocates the risk adaptively to determine an optimal series of waypoints for the vehicle (regardless of type) to follow. For example, cutting a corner requires a larger risk allocation than taking a wide berth. If a user provides a certain level of acceptable risk in the form of a maximum allowable probability of failure, then the p-Sulu algorithm will compute a series of waypoints that spreads that risk percentage over the entire trajectory in a goal-optimizing (e.g., minimum required time or fuel spent) manner.

In order for the SPHERES satellites to receive the waypoints generated by the p-Sulu algorithm, which are computed on a separate computer and sent over the standard SPHERES 868Mhz communication channel, the SPHERES satellites must first send their state to the algorithm. p-Sulu then uses this current state estimate to compute in a receding horizon fashion a series of ten risk allocative waypoints to send to the satellites. While a satellite is executing one set of waypoints, p-Sulu plans the next set. Since this process repeats prior to the completion of the 10 point trajectory, the satellite will never be able to reach the last several waypoints on a given trajectory; the next update will provide the new set of 10 waypoints to follow. The waypoint sequence generated at each update will take the satellite closer to the target location, since the last waypoint is always the target location. Risk is allocated for each trajectory to account for the non-zero probability of a satellite not performing exactly in accordance with the model-based executive within the p-Sulu algorithm.

In order for the communication channel to provide the waypoints and state information to and from the satellites and to enable the testing described in the next section, a customized communication pathway needed to be developed. Figure 43 shows how the communication architecture allows the satellite to communicate with the p-Sulu path planner. First, the satellite determines its state based on the ultrasonic metrology system. The state estimate passes through the standard communication channel to the ground station computer on the Flat Floor, which converts the incoming data to a MATLAB-readable format for sending via TCP/IP to the Linux machine running the p-

Sulu planner. The resultant waypoints are then sent back to the satellite with the reverse process. Once received, the waypoints are used by the SPHERES reconfigurable controller to actuate as necessary to reach its objective.

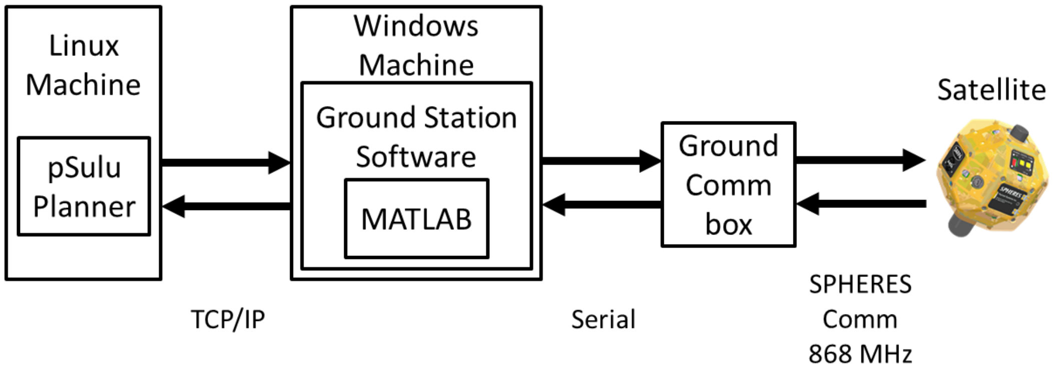


Figure 43: p-Sulu to SPHERES Satellite Communication Architecture (ARMADAS Test Results, 2013)

4.3.4 RARC and Path Planning Increment Description

The testing of both RARC and autonomous path planning using the p-Sulu algorithm can be tested in the same increment. Importantly, the testing in this increment can be conducted by testing either of the two elements separately or together in an integrated manner. In order to make the best use of limited time and money available for testing, efficient use of the test infrastructure is necessary. While individual technology maturation requires less resources, the more intricate or integrated tests where multiple technologies are tested together provide much greater traceability to the final project. For this reason, both RARC and p-Sulu path planning are tested together, but in easily separable test sections so that this single testing increment can both mature each technology individually, but also affording end-to-end testing in a flight scenario.

This testing again occurred on the MIT Flat Floor because of its large testing area upon which multi-satellite dynamics can be tested. An obstacle course was set up on the Flat Floor so that an assembler/servicing spacecraft (primary) needed to avoid an obstacle while maneuvering to a designated rendezvous location near the other (secondary) satellite. The course was created so that a collision avoidance maneuver by the primary satellite was mandatory in order to reach the rendezvous location successfully. Once the primary satellite navigated around the obstacle using the p-Sulu planner, RARC testing could begin by docking the two satellites together and demonstrating the newly reconfigured controller. Consequently, both the p-Sulu path planner and the RARC algorithms could be tested in the same end-to-end test, but they also could be run as self-sufficient independent tests from the same testing code. Figure 44 shows a top-down diagram of the Flat Floor testing configuration, with the primary and secondary satellites, obstacle, and demonstration maneuvers indicated.

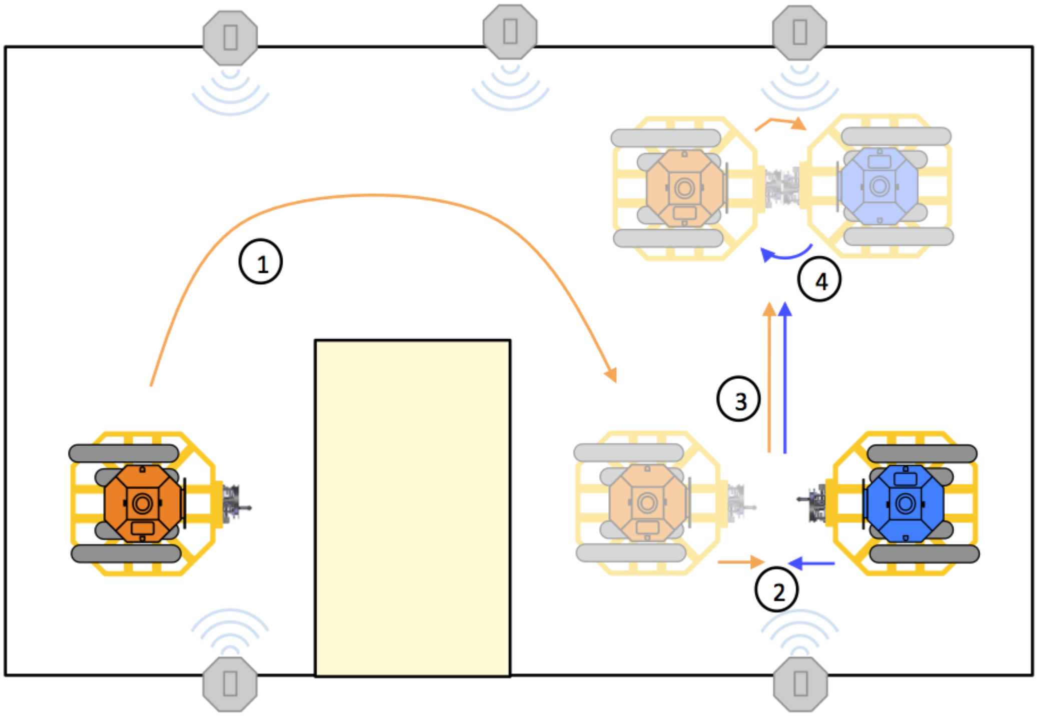


Figure 44: Top-down view of test area where (1) is the p-Sulu path-planning phase (2) is the docking phase (3) is a RARC translation and (4) is a RARC rotation (ARMADAS Test Results, 2013)

As indicated in Figure 44, there are several phases that occur in this testing increment. First, the servicer/assembler satellite follows waypoints that are generated by the p-Sulu path planner based on the satellite's estimate of its initial position at the start of the test. Though the p-Sulu algorithm can be used in a receding horizon manner, the testing of this phase was primarily concerned with the successful instantiation of the algorithm with the SPHERES satellites. The ability of the primary satellite to communicate with the path planner and execute the necessary maneuvers to reach the rendezvous location on the other side of the obstacle was a test success metric. Future work with the testbed could demonstrate the inclusion of the receding horizon capabilities of the planner. With the fixed horizon approach that was implemented as part of this increment, the servicer/assembler satellite demonstrates the ability to fully autonomously determine its initial state, determine the waypoints needed to maneuver around an obstacle, and maneuver to the final location without impacting the other satellite or the obstacle, while also staying within the designated testing area. In order to dock, the primary satellite undergoes a state convergence series of software gates, each more constrained than the last. In doing so, the satellite is able to reduce the initial docking error present whenever two satellites attempt to dock.

In order to initiate docking, both satellites must be in the correct positions as determined by passing through the successive state convergence gates. Once both satellites are ready to dock, the primary satellite makes the final approach by firing its thrusters gently. Upon contact of the two UDPs, the lances of each UDP block the photosensors in the opposing UDP, automatically initiating the motor actuation sequence of locking the lances with the counter rotating cams within each UDP. A rigid dock is accomplished at the completion of the docking. Now that a new satellite system has been created, the two satellites are able to function as a single, larger satellite with aggregated sensors and actuators. Therefore, the controllers are reconfigured with the RARC algorithm with the new system information, consisting of the mass and inertia properties of the combined system.

The RARC algorithm computes the new gains for the controller, so a physical maneuver is required to demonstrate the new system capabilities. The two satellites can perform any maneuver in the 3DOF space afforded by the Flat Floor; however, because any motion can be decomposed into a translation and a rotation, each of these elements are demonstrated separately. It is a simple extension to these demonstrations to maneuver with a combination of translation and rotation using the RARC controller. The two satellites first translate, and then both rotate 180^{o} around the aggregate system center of mass. These maneuvers use all thrusters available to the system, and the state estimate is computed using the sensors from both satellites. Consequently, the aggregation of sensors and actuators, along with the reconfiguration of the controller, demonstrates the RARC algorithms on the Flat Floor with clear traceability to flight operations. Once the rotation is completed, the two satellites undock from one another and reconfigure their controllers back to their original gain values, since the aggregate system parameters no longer apply. Therefore, the test ends with the satellites able to function independently once again. The full sequence of this increment shows an example of necessary steps for an RSA mission:

two satellites docking in orbit, then maneuvering to a new position or orientation, and concluding with the servicer/assembler satellite then moving away to conduct another mission. Though much simpler than what will occur on orbit in terms of the risk and duration of the mission, the elements demonstrated in this increment are crucial for the success of RSA missions, since every one of these missions will involve these path planning for rendezvous and joint motility capabilities.

4.3.5 RARC and Path Planning Increment Results

In order to implement the p-Sulu algorithm on the SPHERES platform, a series of simulation studies were performed using a stochastic simulator to model the real system by including random sensor errors and unmodeled system dynamics. For this increment, the full receding horizon capability of p-Sulu was simulated, with ten waypoints being generated with each state estimate every three time steps. Additionally, the risk bound for p-Sulu was set to 10%, indicating that the probability of obstacle collision is set to 1 in 10 attempts. Though improved risk tolerance can be achieved with lower risk bounds, such as 1% or lower, these low risk boundaries reduce system performance, since with decreasing risk bounds, increasingly conservative (and therefore costly) paths are computed.

Figure 45 shows the result of these simulations: a series of state trajectories as would be computed by the SPHERES estimator of the primary satellite as it traversed the Flat Floor from its initial position at (-1m, 0.5m) to the rendezvous location at (0.5m, 0.5m). Importantly, this figure shows that while every satellite started and ended at the same locations, there exists variability in the paths because of the simulated errors in sensing and actuation performance with uncertainty in dynamic effects. Despite this modeled uncertainty, the p-Sulu algorithm was able to provide paths that successfully brought the simulated satellite to the rendezvous location more than 90% of the trials, as expected. The figure does not show unsuccessful trials.

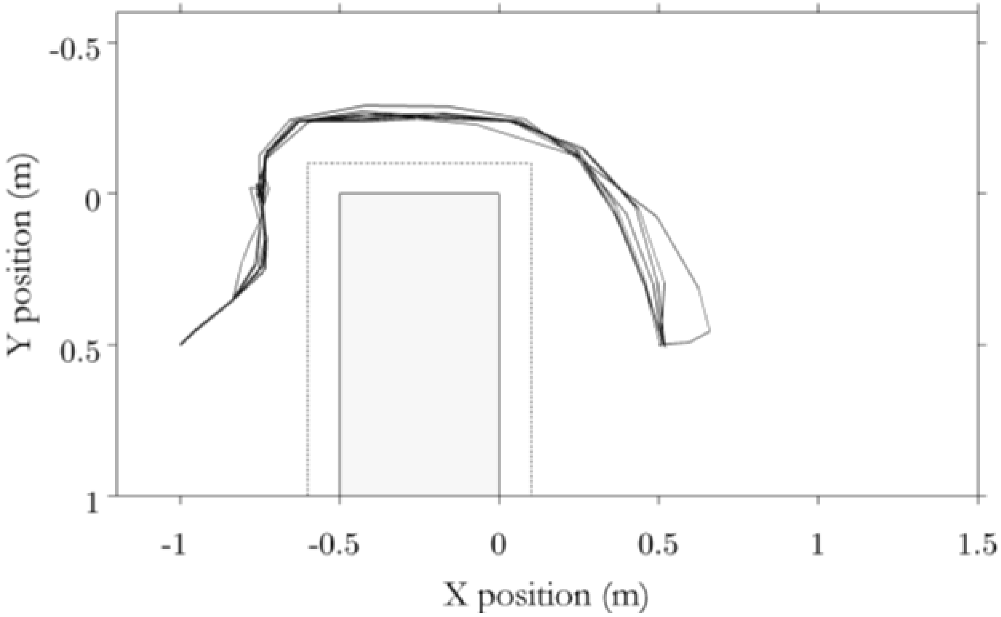


Figure 45: Simulated SPHERES Satellite State Data over Multiple p-Sulu Trials (ARMADAS Test Results, 2013)

Figure 46 shows an example of the comparison between the computed p-Sulu waypoints (dots) and the path that was taken by the satellite (solid line). As can be seen, the primary SPHERES satellite was able to track the waypoints successfully within several centimeters. Although the waypoints were able to prevent a collision with the obstacle, further improvement could have been realized through increasing the density of the waypoints and by providing waypoint updates in a manner more closely timed with the arrival of the satellite at the prior waypoint. Doing so would provide the next waypoint when the satellite is ready to begin the next maneuver instead of commanding a maneuver prior to the satellite being in the expected initial position for the maneuver. These modifications can be implemented in future testing or on orbit for the final RSA testbed, since the basic functionality of the algorithms has been tested in this risk-reduction increment.

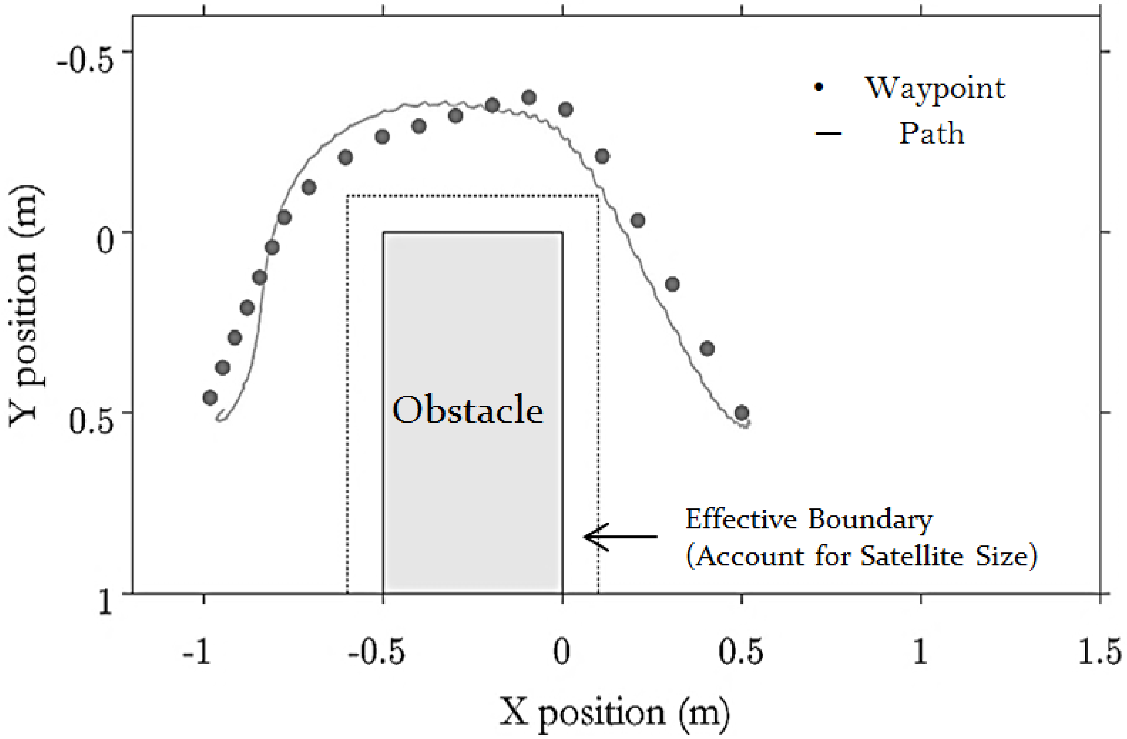


Figure 46: Example SPHERES Satellite Data and p-Sulu Waypoints (ARMADAS Test Results, 2013)

Figure 47 shows the results of several tests on the Flat Floor where the primary satellite maneuvers around the obstacle from an initial position around (-1m, 0.5m) to (0.5m, 0.5m). The primary satellite begins the trajectory at an initial position that has some variation based on the positioning of the satellite on the Flat Floor prior to the start of a particular testing iteration, and as a result, the figure shows this variability. Combined with differences in the surface of the Flat Floor and other 1g effects, the p-Sulu algorithm and the resultant motion of the satellite are not identical from iteration to iteration. There exist a family of curves that maneuver the satellite around the obstacle, but no two trajectories are exactly alike, demonstrating the stochastic nature of the algorithm.

Other features can be seen in the figure. One of the trajectories (shown in red) enters the boundary box indicated by the dashed line. In doing so, this trajectory uses all the risk allotment from the p-Sulu algorithm and as a result the trajectory ends prior to the satellite reaching the rendezvous location. This iteration is therefore a failed rendezvous. The others, however, are successful. The variation in the paths shows how risk is allocated along the trajectories. Some paths have closer approaches to the first corner, but others have closest approaches near the second corner of the obstacle. Other paths, though, maintain a relatively constant distance from the obstacle. This variation is because of the different initial positions and the uncertainties in the model. Further, it is possible to see the variation of ending locations that are distributed near the designated rendezvous location. This inconsistency in this positioning is one of the principal reasons why the series of successively constrained state-based gates were implemented. These results therefore show that the p-Sulu algorithm was able to provide a series of waypoints for the primary satellite to reach its rendezvous location with a set user-defined risk allocation.

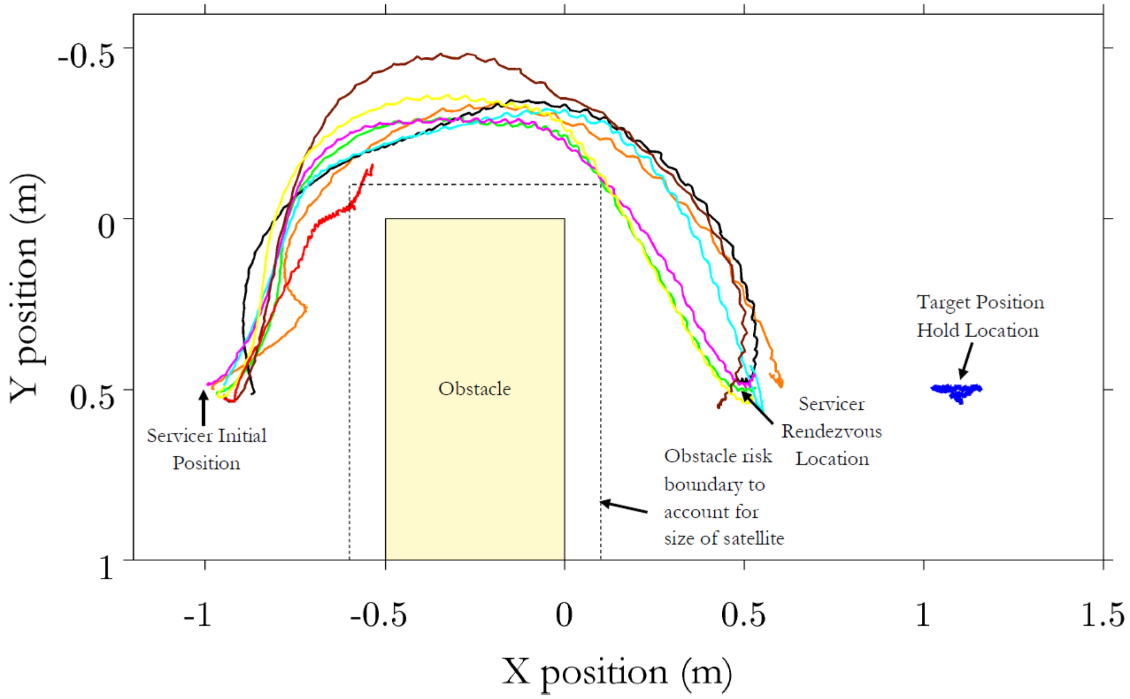


Figure 47: Multiple primary SPHERES Satellite Trajectories Following Fixed Horizon p-Sulu Waypoints (ARMADAS Test Results, 2013)

Docking of one satellite to another was achieved by using the ground Universal Docking Ports, or UDPs. In order for the UDPs to engage and create the rigid dock between the two satellites, the satellites first had to reach the rendezvous location and pass through a series of gates which narrowed the error band on their position and attitude relative to one another. The docking maneuver could then occur, since the satellites both needed to be past the most stringent gate, which corresponded to a 1cm error tolerance between the satellites. With this tolerance, the relative metrology capability of the docking ports was not necessary. Future testing in later increments could utilize this feature and show traceability to different variants of RSA missions. For example, this testing focuses on the traceability to missions where a satellite would need to manipulate a non-cooperative object or any object not outfitted with a UDP. Since most missions will not be conducted in the near future with hardware that is specifically made to provide docking capabilities, the greatest traceability can be achieved with the infrastructure available by testing without the relative metrology system.

Once the two satellites were sufficiently close to one another, with approximately a centimeter gap between the two, the electromagnets within each UDP were activated to reduce the bounce-back that occurs upon the nearly elastic collision between the satellites upon first contact. With the electromagnets holding the two satellites in a soft dock configuration, the motor within each UDP then was able to close counter-rotating cams around the lances to create a rigid dock. This process begins with the lances are sufficiently far into the opposing UDP that it trips the photosensor, an action which leads to the satellite activating the motor. Importantly, due to the design of the UDP, power need not be applied continuously to maintain the rigid dock, making the UDP energy efficient and a valuable asset for future on orbit RSA missions. Undocking could then occur when the satellite commanded the motor to operate in the reverse direction and the electromagnet to generate a magnetic field to repel the other UDP.

Importantly, however, the constraints imposed by the Flat Floor limit the direct traceability to full 6DOF motion aboard the ISS or during on orbit RSA missions. The 3DOF ground environment eliminates errors in relative clocking angles between the UDPs. Further, there are many additional possible rendezvous position and attitude errors that could complicate the docking process on orbit. Though this increment reduces many risks associated with path planning and RARC, on orbit testing is in part necessitated by these ground-induced testing limitations.

The reconfiguration of the controller occurs immediately following the docking event. Each satellite computes the new center of mass, inertia properties, the aggregated thrusting mixing matrix, body blockage constraints for thruster effectiveness, and new controller gains based on the business card properties from Figure 42. All of these calculations are done autonomously and in real time.

The demonstration of this newly reconfigured controller was completed by a translation 0.5 meters followed by a 180 degree rotation. It was determined from the test data that the resource aggregation and controller reconfiguration was successful because no body-blocked thrusters were activated, while the non-body-blocked thrusters fired as required to maneuver the satellite system efficiently. Quantitatively, there are several test metrics that can be analyzed to determine the success of RARC in this testing increment. These metrics include the reconfigured controller’s rise and settling time, as well as the percent overshoot in both rotational and translational configurations. Figure 48 shows a comparison between the expected and actual responses for an example translation of 0.5m.

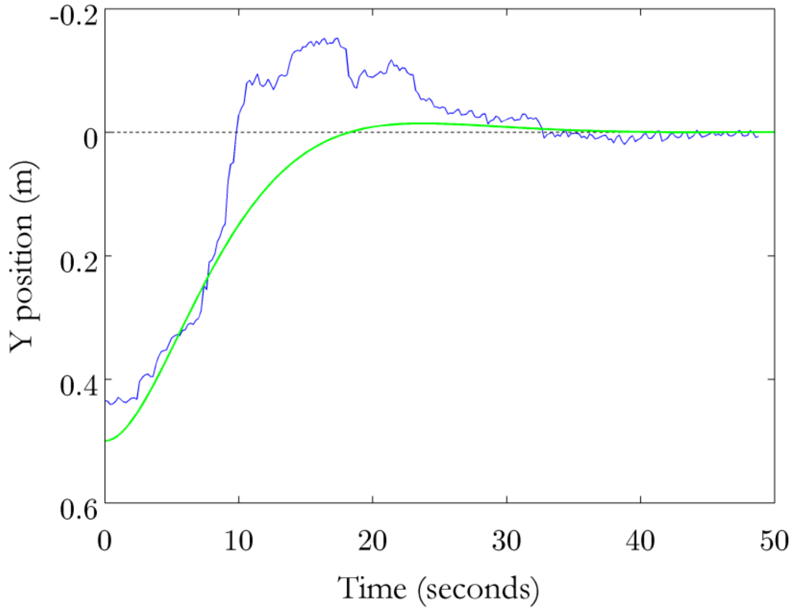


Figure 48: Comparison of Actual (Blue) vs. Expected (Green) Response Curves to 0.5m Step Input (ARMADAS Test Results, 2013)

As can be seen in the figure, the system exhibited a response time of 5.9 seconds, a 35.1% overshoot, and a settling time of 41.2 seconds. Unfortunately, the rapid rise time yielded a large overshoot, and the system also took a while to return to a steady state condition. These results are the consequence of several factors. First, the friction on the Flat Floor is highly dependent on the pressure of the air carriages used to float the satellites: should they have been over-pressured or recently refilled with carbon dioxide, then the amount of friction would have been lower and the aggregate more able to float freely. Additionally, the carbon dioxide tanks containing propellant may also have been refilled shortly before the test occurred, resulting in a larger than expected actuation force. These two potential causes, however, are not likely the principal effects, since the excessive overshoot was seen repeatedly. A more likely error is the improper spacing of metrology beacons.

Figure 49 shows an example comparison of the actual and expected response curves to the 180 degree step input used to rotate the satellite system in the rotational demonstration of RARC effectiveness. As can be seen, the system exhibited a rise time of 16.75 seconds, a 0% maximum overshoot, and a 35.5 second settling time. It is believed that friction on the Flat Floor is the most likely cause for the slower than expected system response. Regardless, such a result is operationally desirable, since the system would be more capable of identifying errors in the attitude of the system before they grew to any dangerous levels. The curves in Figure 50 show the reliability of the RARC algorithm for rotational control over multiple testing iterations.

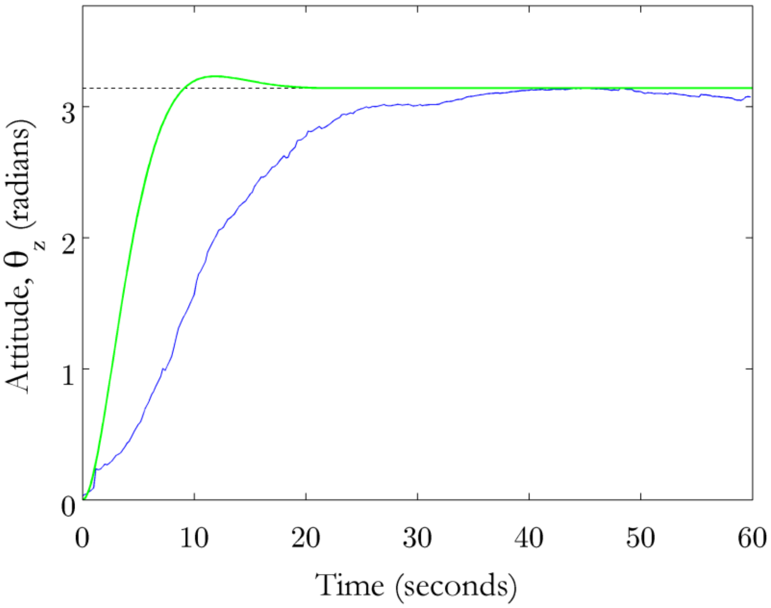


Figure 49: Comparison of Actual (Blue) vs. Expected (Green) Response Curves to 180 Degree Step Input (ARMADAS Test Results, 2013)

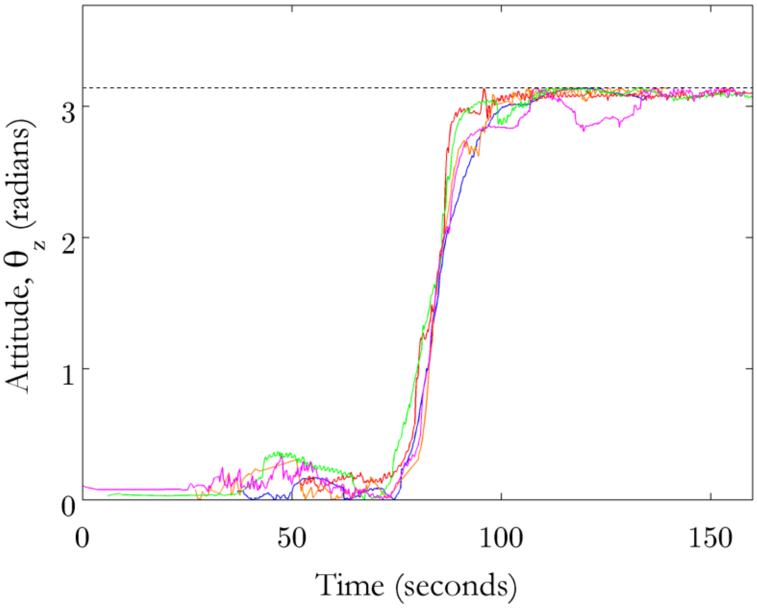


Figure 50: Actual Angular Response Curves over Multiple Testing Iterations (ARMADAS Test Results, 2013)

4.3.6 Path Planning and Changing Dynamics in 1g Conclusion

This increment has demonstrated the successful implementation of a fully autonomous path planner to determine a risk optimum path around an obstacle to reach a target satellite. Additionally, it has demonstrated the successful implementation of RARC algorithms to aggregate the sensors and actuators from both satellite systems as well as to reconfigure the controllers used for maintaining attitude and positional control of the aggregate system. Further, the increment tested the docking and undocking capabilities of a servicer/assembler satellite with a target satellite and the ability for that servicer/assembler to perform both translations and rotations, thereby showing traceability towards the payload manipulation that will be seen during RSA missions. Therefore, this testing increment has through several iterations reduced the risks associated with rendezvous path planning and RARC for RSA missions by raising the technological readiness for flight of the tested algorithms. To do so, traceability to the final flight system was maintained while leveraging the existing infrastructure efficiently to conduct multiple iterations with 1g, hardware-in-the-loop, autonomous testing with clear implications towards future testing increments on the ISS that will make use of the microgravity environment to test in all 6DOF.

4.4 Test Multi-Vehicle Dynamics in 0g

The most directly traceable testing in terms of reducing risks in an operational environment occurs when testing in the microgravity environment. As described previously, the International Space Station provides the optimum combination of risk-tolerant, long duration microgravity. Testing aboard the ISS relies on the availability of the astronaut crew time, a highly limited resource aside from the complexities of developing a testing payload that can meet the necessary ISS safety requirements. The SPHERES facility is already aboard the ISS and has been used successfully for approximately 60 test sessions. Because the newly developed RSA risk reduction testing facility relies heavily on this existing infrastructure, the new hardware and corresponding software as discussed in Section 3.4 was able to enjoy a simplified development process. The use of components and developmental processes with extensive flight heritage is another way to leverage an existing infrastructure to reduce time and monetary costs associated with testbed development.

The change from testing in the ground environment to testing aboard the ISS introduces several fundamental effects to the testing scenarios that can be demonstrated. In this testing increment, not only are the dynamics seen in the ground testing possible, but those in all six degrees of freedom can be seen independently and interacting with one another. For example, the motion constraints imposed by operating on the Flat Floor are removed aboard the ISS, allowing clocking angles to affect docking performance and repeatability. Therefore, in order for this increment to be successful, the testing must not only perform new, innovative risk reduction steps, but also verify that the testbed is still an accurate scaled analog for the flight system by demonstrating similar behavior to the ground testbeds. Additionally, the ability to operate in microgravity enables many new architectures to be tested by fully utilizing all three SPHERES satellites aboard the ISS. Many new multi-satellite configurations can be tested to demonstrate the technologies that have been tested on the ground. The increment must conduct 0g testing that validates the 1g, 3DOF testing results and expands to include multi-satellite operations. In doing so, the incremental testing process can validate the prior increments before beginning new test increments that make specific use of the microgravity environment aboard the ISS.

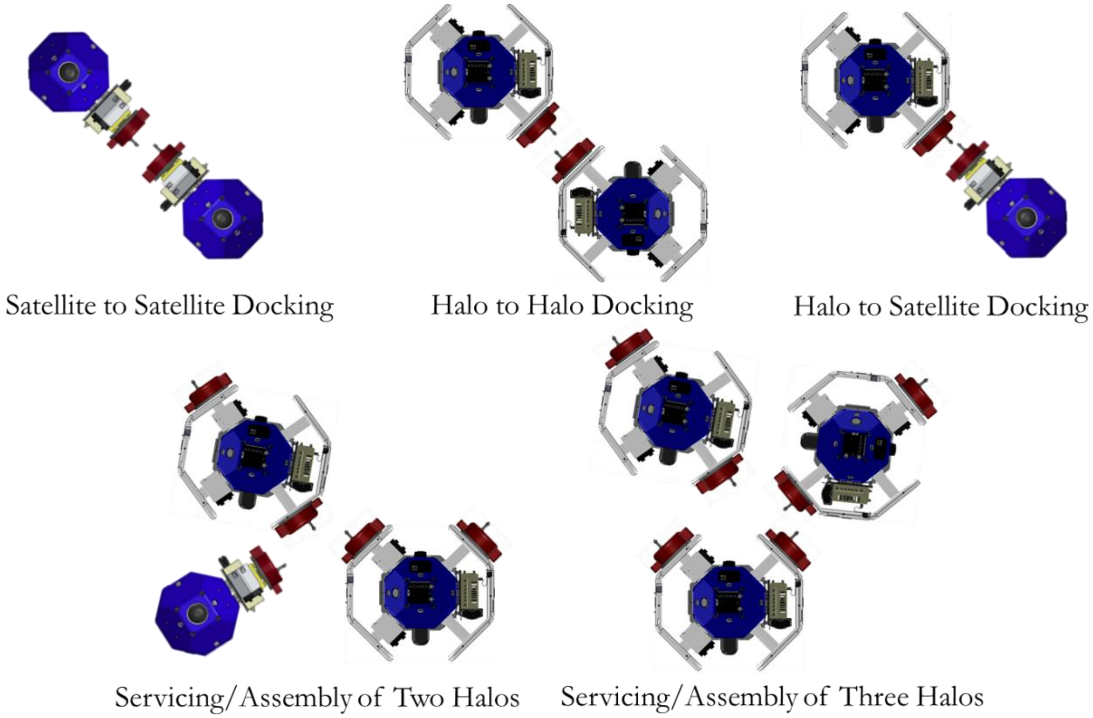
The hardware described in Section 3.4 augments the SPHERES satellites with new multi-satellite and expansion capabilities. The Halo affords the new testing facility multiple rigid electrical and mechanical interfaces to support multiple peripherals simultaneously to enable RSA testing within the constraints imposed by the ISS test volume. The UDP allows for multiple objects to dock and undock autonomously, enabling physical reconfiguration of space systems. These new hardware elements allow for the SPHERES satellites to reconfigure themselves by interacting with one another. Different tests can be created, then, to make the most use of this reconfigurability. For example, interactions with potentially uncooperative targets can be studied, and new sensing and inertial control regimes can be analyzed.

Multiple satellites can interact with one another in all 6DOF aboard the ISS, highlighting the reconfigurability gains afforded by the microgravity environment. The testing that occurs as part of this increment can include a variety of metrics against which to compare the different algorithms used for RSA missions. Control system performance metrics include trajectory tracking error, stability margins, controller convergence properties, the accuracy of any system models or estimators, the controller robustness, resource consumption, total control downtime, the performance drop while transitioning from one controller or state to another, and the fidelity of the system model required to reconfigure a controller correctly. Additionally, there are several spacecraft specific metrics which can also be used for comparison between RSA technologies, including the amount of propellant used, the total additional mass that constitutes the mass overhead penalty for incorporating reconfigurability and modularity into the design, the required sensors and actuators to make reconfiguration feasible, the communication bandwidth necessary to coordinate proximity operations, docking, and controller reconfiguration amongst multi-satellite architectures, and the total added computational capacity and storage memory required to perform controller reconfiguration. At the assembly level, there are also several performance metrics, including the particular assembly sequence that is required given a set of individual modules, the total assembly time (starting with the launch of the first module to be assembled and ending with the assembled system becoming operational), the total fuel used to complete the assembly process, and the TRL of the various hardware and software elements of the architecture.

The evaluation of these various metrics serves to map the architectures to the overall servicing/assembly scenario by observing the effects of configuration changes on overall performance. Certain objectives can be optimized or used as a part of goal programming in this testing increment. These objectives can be implemented as an individual test objective or as a particular configuration that must be achieved to demonstrate the successful implementation of a particular software element or algorithm. Several examples include the state of each module to be assembled or serviced, the state of the servicer or assembler, the total time required to complete the RSA demonstration mission, the total propellant or other consumable usage required to complete the RSA mission, the trajectory tracking error of each spacecraft, and the computational and memory costs of the RSA mission. It is important to note, therefore, that these test objectives are directly correlated to the various metrics by which the different architectures are judged.

The ISS environment combined with the Halo and UDP elements enables the testing of a wide array of risk reduction RSA architectures. The number of satellites, Halos, and UDPs that are used in a particular test determine the possible reconfiguration states. Therefore, a carefully selected incremental sequence of tests has been selected for testing aboard the ISS that will maximize the risk reduction of RSA algorithms while reducing the required level of crew time to conduct the tests. Figure 51 shows five physical architectures, ranging from simple satellite to satellite docking to a full servicing or assembly scenario including three Halo and UDP-equipped SPHERES satellites.

With increasing numbers of each element, the complexity of the test increases. In particular, the ability to incorporate several UDPs onto a single SPHERES satellite enables multiple spacecraft to be docked to it at any given time. Consequently, this satellite can function as a space harbor, docking platform, or servicing/assembling satellite. The testbed elements are agnostic as to their functionality, since it is only through software that specific roles are assigned to the hardware. The SPHERES facility provides a robust mechanism for enabling multi-satellite architectures to incorporate various levels of cooperability as well; the SPHERES hardware is agnostic to the communications architecture implemented in a given test as well. Just as certain roles with a given level of cooperation in the prior two testing increments, the ISS testbed can also test flight-like scenarios by assigning roles and tasks to each satellite. For example, one satellite may be left entirely uncommunicative while another could bring a third satellite for a repair mission. Another example could set all three satellites initially free floating in random locations and task them with conducting self-assembly. Therefore, the Computer Aided Design (CAD) renditions shown for each tradespace architecture in Figure 51 should be taken as one set of potential hardware-only architectures, with each representing an entire family of RSA scenarios.


Satellite to Satellite Docking

Halo to Halo Docking

Halo to Satellite Docking

Servicing/Assembly of Two Halos

Servicing/Assembly of Three Halos

Figure 51: Tradespace of On-Orbit RSA Testing Demonstrations

The iterative testing of each scenario allows the risks associated with each enabling technology to be reduced. This increment is innately full of iterative, incremental testing, since the scenarios will be tested in an order corresponding to increasing complexity and risk. In order to reduce the risk associated with on orbit effects, small risk reduction gains must be realized through small increments with multiple iterations. These tests have been developed as of the time of this writing at a high level; the launch of the new Halo and UDP hardware is scheduled for late 2014, so the testing of this increment will not begin until several weeks thereafter. It is planned to execute this testing sequence on orbit as a means of demonstrating highly likely RSA mission scenarios, architectures, and technologies to reduce the risks associated with multi-satellite interactions in close proximity to one another. Table 7 lists the description and rationale for several possible incremental tests to be performed aboard the ISS based on the architectures shown graphically in the tradespace of Figure 51.

Table 7: Possible On-Orbit Tests with the RSA Risk Reduction Facility

On-Orbit Test Description	Testing Rationale
Two satellite test, each with a UDP, for proximity operation and docking/undocking control	Preliminary controller and algorithm development for docking and undocking
Three satellite test, each with a UDP, for RSA and satellite tug algorithms	Preliminary controller and algorithm development for space tug maneuvering
Two satellite test, one with a Halo and a UDP and the other with a UDP, for docking and maneuvering satellites with different mass and inertia properties	Develop reconfigurable controllers to allow tugs to maneuver satellites of different masses
Three satellite test, one with a UDP and the others each with two UDPs on a Halo, for a tug-based servicing/assembly mission	Develop tug technologies associated with assembly of multiple free-flying satellite modules
Three satellite test, each with two UDPs attached to a Halo, for the aggregation and reconfiguration of a series of orbiting satellite modules	Develop technologies associated with the on-orbit assembly and reconfiguration of multi-satellite architectures
Note: These tests can be carried out with varying levels of autonomy and inter-satellite cooperation. Each iteration should incrementally build on prior results to reduce the risks associated with the particular flight mission. For example, since Phoenix is a highly autonomous mission with a mix of cooperative and uncooperative targets, the tests as outlined in the left column should be tailored to be traceable to Phoenix.	

The development of the ISS RSA testbed is crucial to such risk reduction. It is likely that the results of the ISS testing will directly affect the technology selection of the final RSA project, since this increment is the last of the risk reduction sequence. Consequently, many iterations of each test are required to provide mission assurance and demonstrate the feasibility of the RSA program. While the new risk reduction testbed described in this thesis is a scaled version of an eventual RSA mission system, the results of the aforementioned testing can be used to reduce the full scale risks of the flight system because of care taken in the development of the testbed. A simulation of the full scale behavior using the technologies demonstrated on this testbed will subsequently be performed as a means of verification that the scaling of the results yields expected behavioral and performance results. Chapter 5 discusses the scaling of test results and how the small scale testbed for ground and ISS testing already discussed can provide results that will reduce full scale, on-orbit risks.

5 Chapter 5 – Scalability of Test Results

5.1 Review of Buckingham Pi Theorem and Dimensional Homogeneity

The Buckingham Pi Theorem provides a mechanism for creating a set of non-unique nondimensional parameters based on the principal terms and units of a given phenomenon when written as a mathematical expression (Buckingham, 1914). Particularly, for an equation which has v variables and d dimensions, there exists a set number of possible independent nondimensional parameters given by the equation:

$$p = v - d$$

For a particular equation, there can be many sets of p nondimensional parameters, since the creation of each set of p is determined by the desires of the engineer. Additionally, there is no implied valuation of these parameters; some may be physically relevant and meaningful to a particular problem, but others may be of little use in determining system significant properties. Therefore, it is crucial when using the Buckingham Pi Theorem to begin with an initial estimate of what parameters will be useful in further analysis.

The Buckingham Pi Theorem is rooted in the concept of dimensional homogeneity, originally termed the Great Principle of Similitude by Isaac Newton, which is exemplified by Rayleigh's Method of Dimensional Analysis. Rayleigh's Method focuses on the particular dimensions of the physical constants and variables of an equation, where the dimensions are the base units which describe each variable. According to Sonin (2001), in 1931, Bridgman's Principle of Absolute Significance of Relative Magnitude was created to describe how different factors would affect the dimensions of various units. According to Bridgman's Principle, a number Q obtained by inserting the numerical values of base quantities into a formula is a physical quantity if the ratio of any two samples of it remains constant when base unit sizes are changed. Effectively, this Principle stipulates that the fundamental base units against which measurements are compared must be fixed in order to compare different samples or measurements. Consequently, Bridgman's Principle laid the foundations for the power law form used in dimensional homogeneity, which will be seen in the Pi-number derivation later in this section. In order for an equation to be dimensionally homogenous, both sides of the equation must have the same units. This fundamental property of homogenous equations enables significant analysis to occur relating to the physical interpretation of the equation. If both sides of the equation are equal both in value and dimension, then it is possible to create meaningful derived expressions. Further, dimensional homogeneity enables the creation of Buckingham Pi numbers through the Rayleigh Method.

Using the Rayleigh Method to compute Pi-parameters begins with the determination of which independent variables are included in a given equation. Such a listing must not include variables which are can be rewritten as a combination of the other variables, since doing so would affect the number of *v* variables in the Buckingham Pi Theorem. As an example, it is possible to take the following list of variables as being independently included in an equation:

$$\mathrm{X, P, m, J,}\ \omega, \mathrm{v, a,}\ \alpha, \theta, \tau$$

These variables represent a given state, applied force, system mass, moment of inertia matrix, angular velocity vector, velocity vector, acceleration vector, and applied torque vector, respectively. This set of variables has been selected as an example because it spans a wide array of possible physical parameters which are most common in many physical systems.

It is important to note that each of these variables has an associated set of units. The applied force P, for example, has units of Newtons, which can be written in base units as $\mathrm{kg}\cdot\mathrm{m}/s^2$. By observing the base units for all of the variables listed above, it can be seen that all of the variables can be expressed in terms of the base units for time, length, mass, and angle. Therefore, multiplying a certain set of different powers of the variables together must be able to produce a nondimensional result, as shown by:

$$T^{a_0}X^{a_1}P^{a_2}m^{a_3}J^{a_4}\omega^{a_5}v^{a_6}a^{a_7}\alpha^{a_8}\theta^{a_9}\tau^{a_{10}} \Bumpeq T^0L^0M^0\theta^0$$

This equation shows how each of the variables, including the base variables for time, length, mass, and angle, can be combined with powers $a_i$to produce a nondimensional result where the base units of time, length, mass, and angle are raised to the 0th power. In this equation, the $\Bumpeq$ symbol can therefore be taken to mean “dimensionally equivalent to”. In order to create a set of nondimensional parameters, the values for the eleven a_i variables must be determined: to do so, first the dimensional equivalents are substituted into the prior equation:

$$T^{a_0}L^{a_1}\left(\frac{ML}{T^2}\right)^{a_2} M^{a_3}(ML^2)^{a_4}\left(\frac{\theta}{T}\right)^{a_5}\left(\frac{L}{T}\right)^{a_6}\left(\frac{L}{T^2}\right)^{a_7}\left(\frac{\theta}{T^2}\right)^{a_8}\theta^{a_9}\left(\frac{ML^2}{T^2}\right)^{a_{10}} \Bumpeq T^0L^0M^0\theta^0$$

Next, the exponents can be matched for the four base units, yielding on equation for each unit:

$$T^{a_0-2a_2-a_5-a_6-2a_7-2a_8-2a_{10}} \Bumpeq T^0$$
$$L^{a_1+a_2+2a_4+a_6+a_7+2a_{10}} \Bumpeq L^0$$
$$M^{a_2+a_3+a_4+a_{10}} \Bumpeq M^0$$
$$\theta^{a_5+a_8+a_9} \Bumpeq \theta^0$$

These four can be rewritten as:

$$a_0 - 2a_2 - a_5 - a_6 - 2a_7 - 2a_8 - 2a_{10} = 0$$
$$a_1 + a_2 + 2a_4 + a_6 + a_7 + 2a_{10} = 0$$
$$a_2 + a_3 + a_4 + a_{10} = 0$$
$$a_5 + a_8 + a_9 = 0$$

This set of four equations cannot provide a unique solution, since there are eleven unknowns and only four equations. This step therefore is why the Buckingham Pi Theorem cannot provide a unique set of nondimensional parameters for a given equation, and why it is necessary to begin the analysis with an introductory understanding of the problem at hand. In order to solve this set of equations, the set must be solved in terms of seven of the unknowns. These seven will therefore be seen in the resultant Pi-numbers. If the set were to be solved in terms of $a_0, a_1, a_3, a_4, a_5, a_6, a_9$, then it is possible to write the original equation in terms of only these six variables. Therefore, the equation can be written as:

$$\left(T\sqrt{P}\sqrt{a}\frac{1}{\sqrt{\tau}}\right)^{a_0}\left(\frac{XP}{\tau}\right)^{a_1}\left(\frac{Ma}{P}\right)^{a_3}\left(\frac{PJa}{\tau^2}\right)^{a_4}\left(\frac{\omega\sqrt{P}\sqrt{a}}{\alpha\sqrt{\tau}}\right)^{a_5}\left(\frac{v\sqrt{P}}{\sqrt{a}\sqrt{\tau}}\right)^{a_6}\left(\frac{Pa\theta}{\alpha\tau}\right)^{a_9} \Bumpeq T^0L^0M^0\theta^0$$

This equation now contains seven terms on the left hand side, the product of which is dimensionally equivalent to the right hand side for a set of powers a_i which are not needed to be known. The expressions in each parenthesis are the Buckingham Pi numbers; there are seven because of solving in terms of seven exponents earlier in the derivation. Taking the above equation, it is possible to write the following definitions for the seven Pi-numbers:

$$T\sqrt{P}\sqrt{a}\frac{1}{\sqrt{\tau}} = \pi_1$$
$$\frac{XP}{\tau} = \pi_2$$

$$\frac{Ma}{P} = \pi_3$$

$$\frac{PJa}{\tau^2} = \pi_4$$

$$\frac{\omega\sqrt{P}\sqrt{a}}{\alpha\sqrt{\tau}} = \pi_5$$

$$\frac{v\sqrt{P}}{\sqrt{a}\sqrt{\tau}} = \pi_6$$

$$\frac{Pa\theta}{\alpha\tau} = \pi_7$$

Again, this set of seven Pi-numbers is not unique to this problem, since any combination of nondimensional parameters will still yield a non-dimensional parameter. For example, by combining π_2 and π_3, we obtain $\pi_2\pi_3 = \frac{Xma}{\tau}$, which is a restatement of the definition of torque. Similarly, $\frac{\pi_1}{\pi_5} = \frac{T\alpha}{\omega}$, which is a restatement of the definition of angular velocity, and $\frac{\pi_1}{\pi_6} = \frac{Ta}{v}$, which is a restatement of the definition of linear velocity. It is possible, though, to create meaningful relationships between linear and angular effects, such as $\frac{\pi_5}{\pi_6} = \frac{a\omega}{\alpha v}$ or $\frac{\pi_2}{\pi_7} = \frac{x\alpha}{\theta a}$. Additionally, it is possible to create relationships between applied loads and system responses, such as $\frac{\pi_4}{\pi_7} = \frac{J\alpha}{\tau\theta}$ or $\frac{\pi_2}{\pi_4} = \frac{X\tau}{Ja}$. As exemplified by these examples, there are innumerable ways to combine multiple Pi-numbers together to obtain new nondimensional factors, though they may not all be useful for later analysis.

5.2 Scalability Factors

The hybrid scaling laws discussed by Gronet (1989) and Crawley (1999) provide a means to scale the physical parameters that define a given system based on geometric bases. These scaling laws, written in terms of the two scaling parameters λ and λ_g, can be used to determine the full scale effects as determined by small scale testing conducted with the new testbed developed in this thesis. Since these scaling parameters are developed based on geometric bases and the testbed was developed to maintain traceability and scalability to the final flight system, the testbed to full scale comparison can be made across several measures.

In order to determine the full scale behaviors, the RSA risk reduction testbed must be able to record several measurements while operating either on the ground or aboard the ISS. For example, the frequency of a flexible mode cannot be seen easily without high-speed IMU data collected from the SPHERES satellites. For tests where storage memory is limited, as might be the case for an extended duration test, certain information may be gathered from a scaling analysis, such as using scaling to determine how long a full scale maneuver might require to complete. The IMU data will also be able to show how linear and angular velocities and accelerations scale. Importantly, however, a distinction must be made when comparing across testbeds with differing numbers or types of actuators.

There are several parameters that do not require active testing to be computed or identified. These include the moment of inertia matrices and total mass. The forces and torques generated by actuators may also fall into this category, since the reactions to the application of actuators such as the SPHERES thrusters may be known a priori based on the mass and inertia properties that can be calculated statically and the manufacturer or unit testing of the actuators. It is important, however, to take careful notice of the relative number and relative positioning of thrusters between testbeds, since these variations may make a scaled testbed behave quite differently than what is geometrically expected based on the hybrid scale modeling approach. Additionally, the moment of inertia can be computed statically, but care must be taken to ensure that axis definitions and hardware orientations are maintained wherever possible. Otherwise, rotation matrices must be included in any subsequent analysis, such as when developing a control law or defining rendezvous orientations. Should hardware be mounted in different orientations, then the testbed will not have as much direct traceability to the final flight hardware, which can lead to less reliable results when scaling to the full scale system. The level of risk reduction will correspondingly be less than if all hardware orientation and axis systems were identical to those found in the flight system.

5.3 Hybrid Scaling of Testbed Parameters

In order to use the hybrid scale model parameters to develop scaling laws that can be used to provide full scale RSA risk reduction from scaled test results, the physical properties that describe the testbed must be scaled by terms formed of the scaling parameters. Table 8 shows the hybrid model equivalents for many common physical parameters by expressing the parameters in terms of their base units and substituting the corresponding hybrid scale parameters as appropriate. Several additional parameters have been included in this table beyond an initial set constructed by Gronet and Crawley in their respective papers.

Table 8: Hybrid Model Equivalents for Common Physical Parameters

Symbol	Physical Meaning	Hybrid Model Equivalent
F	force	λ^2
ρ	density	1
m	mass	$\lambda^4 {\lambda_g}^{-1}$
l	length	λ_g
v	linear velocity	$\lambda^{-1}\lambda_g$
ω	angular velocity	λ^{-1}
a	linear acceleration	$\lambda^{-2}\lambda_g$
α	angular acceleration	λ^{-2}
f	frequency	λ^{-1}
M	moment	$\lambda^2\lambda_g$
I	moment of inertia	$\lambda^4\lambda_g$
KE	kinetic energy	$\lambda^2\lambda_g$

It is important to note the power order of each of these parameters. For example, the moment of inertia scales with a combined power of X^5, but a length only scales with X^1, where X is the multiplication of both scaling parameters. By adding the powers of the scaling parameters together, it is possible to determine how sensitive each physical parameter is to scaling: the higher the sum, the more sensitive the parameter. This sensitivity is important when designing testbeds, since a particular requirement on direct traceability to a full scale flight system may limit the degree of scaling should λ and $\lambda_g$be largely different from one another. In those cases, a small difference in the scaling parameters can result in a much more pronounced effect on the scaling value for a particular physical parameter. Should this effect be seen, the testbed may not be deemed a valid testbed for risk reduction of full scale technologies, since the testbed would behave in a manner that is sufficiently different from the flight system to prevent a valid comparison from being made between the testbed and the flight system.

Another scaling aspect that is made evident in this table is the combined scaling parameter power of similar physical parameters. For example, the scaling factor for linear velocity goes with X^0 while linear acceleration goes with X^{-1}, analogous to taking the derivative of velocity and applying the power rule. A similar effect is seen with the angular velocity at X^{-1} and angular acceleration at X^{-2}.

Additionally, it is possible to identify how the physical parameters relate to one another by observing how the scaling factors can be viewed as expressions corresponding to the equations that govern the physical parameters. For example, because angular velocity scales with X^{-1} and a length scales as X^1, we can observe that the scale factor for linear velocity, which has the equation of $V = r\omega$, multiplying the scale factors for the length (r) and the angular velocity (ω) yields the scale factor of X^0 for the linear velocity (V). These types of relationships are an indicator that the physical parameters are fundamentally based on a small set of base units, specifically those listed in the analysis in Section 5.1. More rigorously, the scale factors can be applied to these physical equations in the place of the physical parameters: in this example, we can see how $\lambda_g \cdot \lambda^{-1} = \lambda^{-1}\lambda_g$, which is the scaling factor for linear velocity. Again, this result is obtainable because of the physical basis for the hybrid model scaling factors in the physical, geometric regime. It is useful when conducting this type of analysis to perform these simple checks to ensure proper scaling factor formulation.

In Figure 52, a hybrid scaling schema is shown that shows how a flight RSA satlet can be scaled to the size of a pair of SPHERES satellites. As shown, two satlets are joined together by a docked interface or connector beam that, while rigid, still exhibits flexible modes to be controlled by the onboard controllers. Similarly, the SPHERES satellites are docked together by a docked interface, such as a pair of SPHERES Docking Ports, which also are rigid while still exhibiting flexible modes. The physical dimensions are shown in terms of the length scales and hybrid scaling parameters shown in Figure 14, though an assumption is made that the RSA satlets are larger than the SPHERES satellites in this figure. For the final RSA mission, however, no constraints on the comparative scales are required, since the scalability analysis presented in this chapter is valid across both up- and down-scaling from the flight satellites to the testbed.

The hybrid scaling approach that uses the scaled dimensions shown in Figure 52 is taken to be the Hybrid Length/Force Frequency Strain (HL/FFS) model, because both linear and angular deflections as characterized by mode shapes are preserved across scales. In this scaling model, the magnitude of the linear deflections are given by λ_g, and the kinetic energy and potential energy terms are given by $\lambda^2\lambda_g$, which are slightly different from their respective values in the replica model. Consequently, the rotational deformations are correctly scaled using the HL/FFS scaling model, which most accurately describes the types of deformations expected to be seen across the docked interface: rotation of the two satellites will present themselves with larger displacements than linear translation between the satellites owing to the inherent flexibility within the docked system.

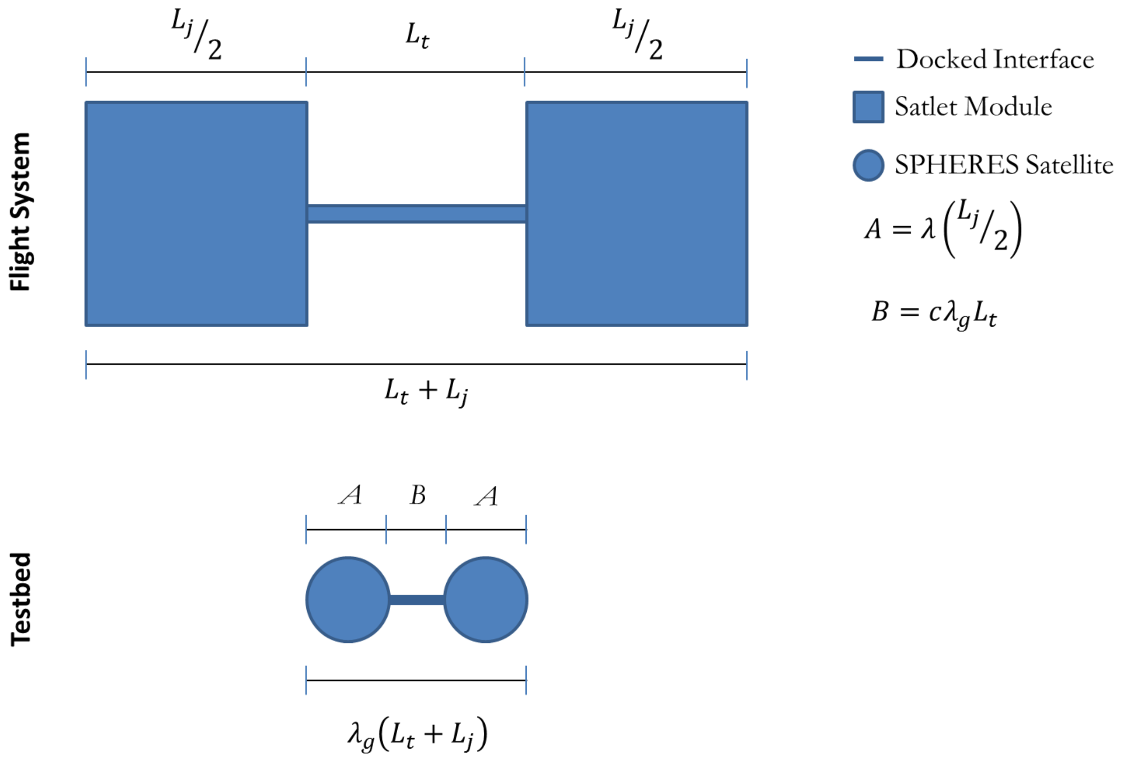


Figure 52: Scaling of RSA Satlet to SPHERES Satellite Using Hybrid Scaling Parameters

5.4 Combination Process for Nondimensional Hybrid Modeling

The need to determine full-scale effects from small-scale testbeds is based on the need to reduce full-scale, on-orbit risks using small-scale ground and flight experiments. There are several challenges associated with scaling, including that scaling up or down cannot exactly replicate full-scale dynamics, and that it is not easy to compare results across differently scaled testbeds. Solutions to these issues come in the form of using hybrid modeling to determine effects at different scales and using nondimensional analysis to compare results directly across multiple testbeds by creating common parameters for comparison. The combination of these two methods therefore provides a mechanism for overcoming the challenges associated with testing and reducing risks at multiple scale levels.

The process for combining the two techniques of hybrid modeling and nondimensional analysis focuses on the development of hybrid equivalents for nondimensional parameters. While nondimensional analysis remains bounded to physical parameters themselves, hybrid modeling creates hybrid model equivalents for each physical term. The combination of the two is therefore a hybrid scaling model equivalent nondimensional Buckingham Pi number. Combining the two methods provides a mechanism for solving the challenges associated with scaled testbeds.

As an example of how the nondimensional hybrid modeling approach can be used in practice with scaled testbeds, it is possible to analyze the ground testbed that was used for RARC and path planning testing and compare it against a proposed flight configuration involving two docked SPHERES satellites, each with a Halo and a docking port. For this scaling process, the scaled dimensions shown in Figure 52 are used. Each SPHERES satellite is 21.3cm in diameter, and each RARC testbed floating element, consisting of a SPHERES satellite, SWARM Propulsion Unit, SPHERES Docking Port, and air carriage, has a width of approximately 98cm. If the two docked Halos are taken as the final flight configuration, then we can say that it is full scale with its own values of the hybrid modeling scale parameters, $\lambda = \lambda_g = 1$, and compare the ground testbed to it. If we take a value of 2cm to be the flexible docking port region that connects two SWARM Propulsion Units together, then it is possible to write that:

$$c\lambda_g L_t = 2cm$$

$$\frac{\lambda L_j}{2} = 48cm$$

$$\lambda_g\left(L_t + L_j\right) = 98cm$$

These three expressions can be visualized in Figure 53:

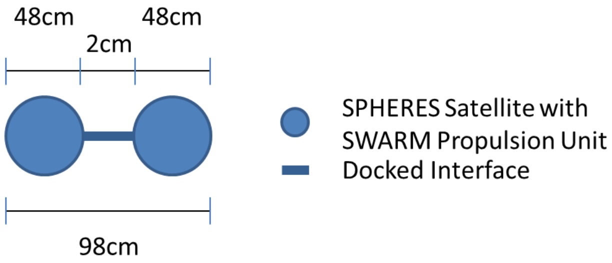


Figure 53: SPHERES Satellite on SWARM Propulsion Unit Scaling Schematic

For the Halos, it is possible to use the same 2cm value for the flexible region because the docking ports are used both on the Halo and the SWARM Propulsion Unit. Each Halo is 41cm in diameter. Consequently, the total docked width of the flight system is 88.8cm so that $L_j = 86.8cm$ and $L_t = 2cm$, as shown below in Figure 54:

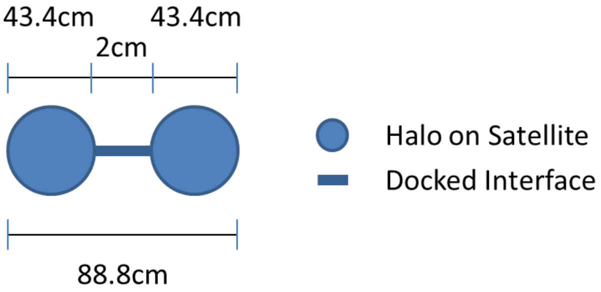


Figure 54: Docked Halo Scaling Schematic

Now it is possible to return to the original equations written for the SWARM Propulsion Unit to determine the values of the scaling factors:

$$\frac{\lambda L_j}{2} = 48cm$$

$$\lambda_g \left(L_t + L_j\right) = 98cm$$

Based on these two equations, we can determine the hybrid modeling scale parameters are: $\lambda = 1.106$ and $\lambda_g = 1.104$ by solving with the values of L_j and L_t for the total docked Halo system.

Using these two hybrid scaling parameter values, it is possible to determine the scaling factors for the several common physical properties listed in Table 8 for the docked Halo system, as shown in Table 9.

Table 9: Expected Scaling Parameters for Docked Halo System

Symbol	Physical Meaning	Hybrid Model Equivalent	Scaling Value
F	force	λ^2	1.223
ρ	density	1	1
m	mass	$\lambda^4 {\lambda_g}^{-1}$	1.355
l	length	λ_g	1.104
v	linear velocity	$\lambda^{-1}\lambda_g$	0.998
ω	angular velocity	λ^{-1}	0.906
a	linear acceleration	$\lambda^{-2}\lambda_g$	0.903
α	angular acceleration	λ^{-2}	0.818
f	frequency	λ^{-1}	0.904
M	moment	$\lambda^2\lambda_g$	1.350
I	moment of inertia	$\lambda^4\lambda_g$	1.652
KE	kinetic energy	$\lambda^2\lambda_g$	1.350

As an example of how these scaling values can be used in practice when the Halo design has yet to be finalized, it is possible to compare the mass as would be expected based on these scaling values with that of the current SolidWorks model. The docked SWARM Propulsion Unit configuration has a total mass of approximately 35.6kg, including the two SPHERES satellites, tanks, air carriages, and SWARM Propulsion Unit. Therefore, the mass of the docked Halo configuration on air carriages would be approximately $\frac{35.6kg}{1.355}$ = 26.3kg, or 13.15kg per docking element. Because an air carriage with tanks is 5.54kg, a Halo mounted onto a SPHERES satellite with the direct scaling provided by the above analysis should have a mass of 7.61kg. Consequently, this approach yields a Halo mass of 3.45kg, accounting for a 4.16kg SPHERES satellite. This mass value is approximately 76% of the SolidWorks estimate. This difference between the scaled and SolidWorks estimates shows the importance of designing testbeds to be as traceable as possible both across testbeds and to the final flight system, but also that while this scaling approach can provide initial estimates for future systems, detailed analysis of the flight hardware is still necessary for determining on-orbit performance.

5.5 Scaling of Existing and New Pi-Numbers

It is possible to scale existing Pi-numbers and newly created Pi-numbers using a sequence of steps which create hybrid model equivalents of the physical parameters which comprise the Pi-number. This process, outlined below, can be applied to any Pi-number, but it can also be applied to dimensional parameters which are incorporated into a nondimensional Pi-number. The Pi-numbers themselves introduce the common element for comparison across many scales.

1. Write out the expression for an existing nondimensional parameter

$$Ne = \frac{F}{\rho l^2 v^2}$$

2. Write down all directly available hybrid model equivalents used to form the selected nondimensional parameter

$$\rho \rightarrow 1$$

$$v \rightarrow \lambda^{-1}\lambda_g$$

$$l \rightarrow \lambda_g$$

3. Write out the units of the variables in the nondimensional parameter for which there are no directly known hybrid model equivalents (note: Table 8 provides most of the hybrid model equivalents for the physical quantity-based variables, including the force term derived here)

$$F = [N] = \left[\frac{kg \cdot m}{s^2}\right]$$

4. Write out the hybrid model equivalents for each parameter base unit

$$kg \rightarrow mass \rightarrow \lambda^4{\lambda_g}^{-1}$$

$$m \rightarrow length \rightarrow \lambda_g$$

$$\frac{1}{s} \rightarrow frequency \rightarrow \lambda^{-1}$$

5. Combine all λ, λ_gfactors to create the new nondimensional parameter based on the hybrid modeling scaling factors

$$Ne = \frac{\lambda^2}{1 \cdot \lambda_g^2 \cdot \left(\lambda^{-1}\lambda_g\right)^2}$$

6. Reduce to the form $\langle parameter \rangle = \lambda^a \lambda_g^b$ for constants a and b

$$Ne = \frac{\lambda^4}{{\lambda_g}^4}, a = 4, b = -4$$

With this sequence of steps, the Newton Number has been converted from an expression in terms of physical quantities into an expression of the hybrid model scaling factors. In order to enable the scaling of physical effects, then, this new formulation must be analyzed.

Figure 55 shows a contour plot of the Newton Number in its hybrid equivalent form. As can be seen, there is a very rapid change in Newton Number value near the x- and y- axes as a result of the dependence on the fourth power of each of the hybrid scaling factors. This strong dependence implies that even small changes in the scale of the small-scale testbed can have significant impacts on the relevance of a given testbed for reducing full-scale risks. Here, relevance is based on the similarity between the Newton Number of the full-scale system and the Newton Number of the small-scale testbed. Much like the Reynolds Number in fluid dynamics, comparisons can be made between different scales only if the Reynolds Number is similar. In this analysis, the Newton Number must be similar should the physical effects be comparable.

A full-scale system will have a Newton Number equal to 1, since both the hybrid scaling factors $\lambda_g$and λ are equal to 1, indicating that the full-scale system has not been scaled. The full-scale system is plotted as the green circle with a black perimeter in Figure 55. If a small-scale testbed were to be developed with both $\lambda_g$and λ equal to 0.6, indicating that the testbed was scaled to the level 0.6 of full-scale, then it, too, would have a Newton Number of 1. Such a testbed is plotted as the green circle in Figure 55. Since it has the same Newton Number as the full-scale system, it can therefore be said that the dynamical behaviors seen in the testbed related to the velocity response of the testbed as a result of an outside force will accurately approximate the same behaviors of the full-scale system. Should the Newton Number be different across scales, then the behavior of one scale cannot be said to represent the full-scale behavior accurately. In

Figure 55, the yellow circles represent Newton Numbers which are approximately 50% different from the full-scale value of 1. Testbeds corresponding to these values would have similar behaviors, but they would not be fully representative of the full-scale system. Testbeds with Newton Numbers even further from full-scale, such as those represented by the red circles, cannot be said to provide relevant behaviors to the full-scale system: their dynamics are too different across scales for full-scale risks to be reduced with the scaled testbeds. Effectively, the results from these testbeds are unable to be applied to the particular full-scale system.

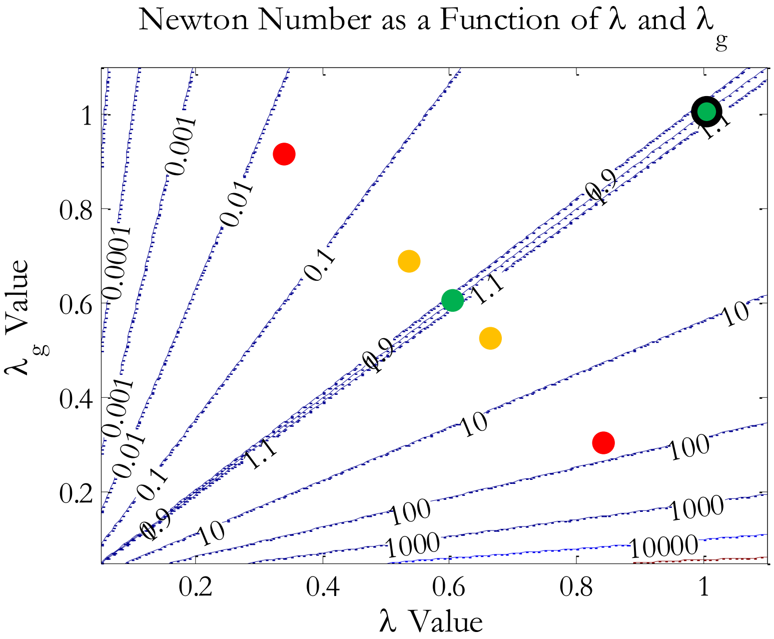


Figure 55: Newton Number as a Function of λ and λ_g

The Newton Number is not the only nondimensional parameter that can be scaled. There are thousands of possible nondimensional numbers that have been named, and Table 10 lists several common examples from multiple disciples. The table lists applicable fields of application as well as the laws and definitions which are used in deriving the parameters. Often, the parameters, including the Newton Number, are formulations of common equations expressed in a non-standard way in order to incorporate multiple elements of the system. As can be seen, the field of fluid dynamics has a considerable number of common nondimensional parameters which relate to the effects caused by certain flow properties across multiple scales.

Table 10: Common Nondimensional Parameters from Multiple Disciplines (Schuring, 1977)

Name	Symbol	Expression	Fields of Application	Constituting Laws/Definitions
Euler	Eu	$2p/\rho v^2$	Friction of flow in conduits	Newton's 2nd law
Newton	Ne	$F/\rho l^2 v^2$	Phenomena involving inertial forces	Newton's 2nd law
Cauchy	Ca	$\rho v^2/E$	Vibration of elastic systems	Hooke's Law of isotropic solids and Newton's 2nd law
Strouhal	St	$fl/v,$ $f = \sqrt{\frac{k}{m}}$	Elastic vibrations	Modified Ca: all elastic properties are lumped into *k*, all masses are lumped into *m*
Drag Coefficient	Cd	$2F_d/A\rho v^2$	Resistance of bodies moving through a fluid	Newton's 2nd law
Eckert	Ec	$v^2/c_p\Delta T$	Forced heat convection	Law of heat capacity, definition of kinetic energy
Bond	Bo	$\rho g l^2/\sigma$	Sloshing of liquid under low gravity; liquid atomization	Potential energy (law of gravitation) and Surface energy of liquids
Weber	We	$\rho l v^2/\sigma$	Surface tension effects	Surface energy of liquids, definition of kinetic energy
Froude	Fr	$v/\sqrt{gl}$	Motions of accelerated masses in gravitational field, surface waves	Newton's 2nd law and corresponding law of gravity
Reynolds	Re	$\rho v l/\mu$	Turbulent flow	Shear stress of Newtonian fluid and Newton's 2nd law to describe inertial to viscous forces

The field of structural dynamics has its own set of nondimensional parameters, as shown in Table 11. Here, the same columns are included as in Table 10. Each of these nondimensional parameters can be scaled using the same process as described above in order to see how the effect of scaling will influence system behaviors. Further, these named nondimensional numbers can be merged with others to form new nondimensional parameters in the same way that Pi-numbers can be merged to create new such numbers.

Table 11: Common Nondimensional Parameters from Dynamics (Kunes, 2012)

Name	Symbol	Expression	Fields of Application	Constituting Laws/Definitions
Ekman Number	Ek	${}^{w}/_{\omega L^2}$	Inertial to centrifugal forces ratio during rotation with moving fluid	Angular velocity, angular frequency, characteristic length
Gravitational to Centrifugal Acceleration Ratio	Ngc	${}^{g}/_{\omega^2 R}$	Dynamic force relations, for example, in rotary systems	Gravitational acceleration, angular frequency, radius
Homochronous Number	Ho	${}^{\omega\tau}/_{L}$	Serves to choose the timescale for similar processes	Motion velocity, transfer time to distance L, distance
Structural Utilization Number	Nsu	${}^{gL\rho}/_{E}$	Expresses the ratio of the weight of a structural part to its rigidity	Gravitational acceleration, characteristic length, density, modulus of elasticity
Strouhal Number	Sn	${}^{fL}/_{w}, {}^{w}/_{nL}$	Relates the ratio of local force, caused by a non-stationary process, to the inertia force. Represents the universal dynamics similarity criterion of non-stationary processes in systems which are similar geometrically and kinematically.	Frequency, characteristic length, velocity, rotational frequency
Force Numbers	Nf	${}^{F\tau^2}/_{mL}$, ${}^{F}/_{qL}$, ${}^{FL^2}/_{GJ}$	Forces applied to beams	Dynamic force in damped mass movement, static force in longitudinal beam load, static force with static torsional stress

5.6 Control Law Scaling

The interaction between different elements on-orbit will be based on how the independent control laws interact with one another. One of the principal risks associated with multiple free-flyers is that each is operated with its own control law in close proximity to one another. It, therefore, is crucial that the control laws are finely tuned so that errors are minimized while also enabling maneuver capabilities on rapid timescales. As elements dock, undock, or reconfigure, the mass and inertia properties of the system change. The dynamics of the newly reconfigured system state are dependent on these physical system properties, so the system controller must be able to adapt and take advantage of these new dynamics.

Control laws can be developed based on the physical scaling effects associated with the system's physical properties, such as the mass and inertia, as they change over time. By using the hybrid scaling laws discussed in Section 5.3, it is possible to rewrite the various control law elements using the hybrid scaling parameters. These elements include control gains, the actuator controllability, sensor observability, and the frequency and magnitude response due to either control overshoot or contact between orbiting spacecraft. These control elements are important indicators of system performance on-orbit, and must be scaled appropriately in each testbed according to the scale of the testbed. Risk reduction is afforded by implementing not the exact flight controller, but one that is appropriately scaled to that of the risk reduction testbed. With a scaled controller, the system plant need not be scaled. Consequently, the plant dynamics can be left unscaled and the process and system actuators unchanged. Only onboard software need be modified between testbeds.

Figure 56 shows an example of a standard control loop. Here, $y_d$is the desired response, y_p is the produced state, e is the error, and F and τ are the required force and torque, respectively, to meet the desired output as output from the controller. The sensors read the output state y_p in order to send the information back to the controller, and the actuators perform the control actions based on what is computed by the software. It is from this diagram that the control elements which can be scaled are derived.

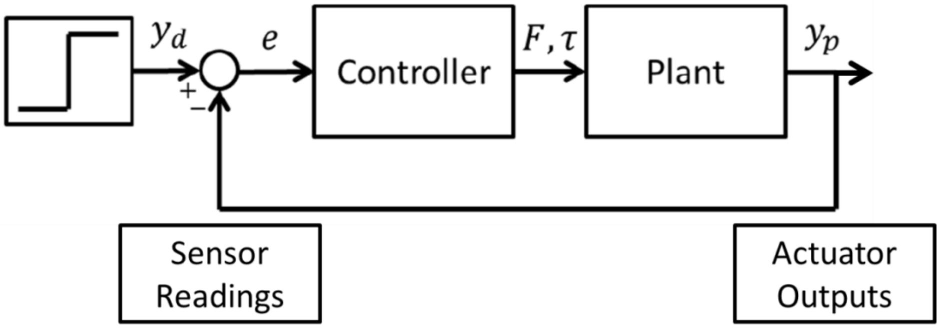


Figure 56: Example Basic Control Loop

As described in Jewison, et al., (2014) and in Section 4.3, the principal controller implemented on the SPHERES satellites is a Proportional-Derivative (PD) controller. The controller receives the state estimate from the extended Kalman filter based on the sensor readings and the commanded state target from the user code. By differencing the state target and the state estimate, the PD controller is able to determine the state error and compute the necessary forces and torques for the thrusters to actuate as described in Figure 56. The PD control law is given below for control on the MIT Flat Floor, where motion is only possible in the *x*-*y* plane and rotation can only occur about the *z*-axis:

$$F_x = k_{p,position} e_x + k_{d,position} e_x$$

$$F_y = k_{p,position} e_y + k_{d,position} e_y$$

$$T_z = k_{p,attitude} e_\theta + k_{d,attitude} e_\omega$$

In these equations, 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. A corresponding control law can be written for motion in any other space, including the full 6DOF ISS environment. These three equations incorporate four gain constants which are based on the physical properties of the system; of all the terms in the control law, it is these three which must be scaled based on the specific testbeds. The proportional and derivative gains are calculated by modeling the spacecraft system in state space. The poles of the system are selected based on desired damping ratios and natural frequencies. For reference, the following values were selected: for the position gains, $k_{p,position}$ and $k_{d,position}$, the selected damping ratio is 0.75 and the natural frequency is 0.2 Hz, but for the attitude gains, $k_{p,attitude}$ and $k_{d,attitude}$, the damping ratio is set at 0.75 with a natural frequency at 0.4 Hz. The poles are placed at these desired locations according to the method of state feedback. The equations for each of these gains are shown below:

$$k_{p,position} = \omega_n^2 m$$

$$k_{d,position} = 2\zeta\omega_n m$$

$$k_{p,attitude} = \omega_n^2 I$$

$$k_{d,attitude} = 2\zeta\omega_n I$$

In these equations, ω_n is the natural frequency of the system, *m* is the system mass, *I* is the system inertia, and ζ is the damping ratio. As can be seen in the above equations, as the mass and inertia change due to the docking and undocking of spacecraft, the gains must correspondingly change to maintain the desired control pole locations. The challenge arises, then, to reconfigure the controller based on the scale of a particular testbed in relation to the full-scale spacecraft, or when reconfiguration occurs. It is possible to use the hybrid modeling parameters listed in Table 8 to create the hybrid model equivalents for these control gains, but a new hybrid model equivalent is required for the damping ratio.

As described in Crawley, et al, (1990), there are two primary sources for passive damping: material and interface/structural damping, where the latter provides the majority of observed damping. The authors of this paper note that the scaling of strain energy is a mechanism for determining how the damping ratio scales based on a nonlinear dissipation loss factor, given as:

$$\eta = \frac{\Delta U}{2\pi U}$$

In this equation, U is the strain energy in the system at a given point, and ΔU is the amount of strain energy that is left after some energy is lost to damping. Importantly, if this loss factor is invariant to scaling, then the scale of a particular testbed does not impact the structural damping due to joints. Consequently, the authors show the following table to describe how these two strain energy terms scale (listed as lower bounds):

Table 12: Scaling of Strain Energy (Crawley, 1990)

Strain Energy	Replica Model	L/FFD Model	L/FFS Model
$\boldsymbol{U}$	λ^3	λ^3	$\lambda^2\lambda_g$
$\boldsymbol{\Delta U_{pin,geom}}$	λ^3	$\lambda^4\lambda_g^{-1}$	λ^3
$\boldsymbol{\Delta U_{pin,grav}}$	λ^4	$\lambda^5\lambda_g^{-1}$	$\lambda^5\lambda_g^{-1}$
$\boldsymbol{\Delta U_{axial,geom}}$	λ^3	λ^3	λ^3
$\boldsymbol{\Delta U_{axial,grav}}$	λ^4	λ^4	$\lambda^5\lambda_g^{-1}$
$\boldsymbol{\Delta U_{deadband}}$	λ^3	λ^3	$\lambda^4\lambda_g^{-1}$

In this table, the gravity-based strain energies show how gravity-induced-dissipation scales, while the geometry-based strain energies show how dissipation due to all other sources of load scales. The deadband strain energy shows how energy is lost to impacting of an axial joint. Based off of these scaling values, it is possible to determine how the nonlinear dissipation loss factor scales for each of the scaling models. For the replica model, all of the geometric effects scale as λ^3, including U, making η independent of scale. Similarly, the L/FFD axial, geometric and the L/FFS pin and axial geometric strain energies have η independent of scale. Because the loss factor scaling is akin to the scaling of the critical damping ratio, it is possible to take the scaling of η to be the scaling of ζ. Because the modeling of the testbeds in this thesis is taken as HL/FFS, the damping ratio ζ

will be taken to be 1, indicating that it is independent of scale, because $\eta = 1$ for both axial and pin geometric dissipation.

It is now possible to create scaling laws for the four control gains as derived from the method of state feedback. The two position and two attitude gains, one each for the proportional and derivative terms, are listed below:

$$k_{p,position} = \lambda^2 \lambda_g^{-1}$$

$$k_{d,position} = 2\lambda^3 \lambda_g^{-1}$$

$$k_{p,attitude} = \lambda^2 \lambda_g$$

$$k_{d,attitude} = 2\lambda^3 \lambda_g$$

These PD gain scaling factors can be used to modify the final flight gains as required for testing. The resulting controller will exhibit performance properties specific to the gains. The transient response to unit step inputs provides several performance properties which can be used to compare the effectiveness of different controllers. Of particular importance are the rise time, settling time, time of maximum amplitude (peak time), and the peak overshoot. Figure 57 shows these transient response properties graphically for a response to a unit step input.

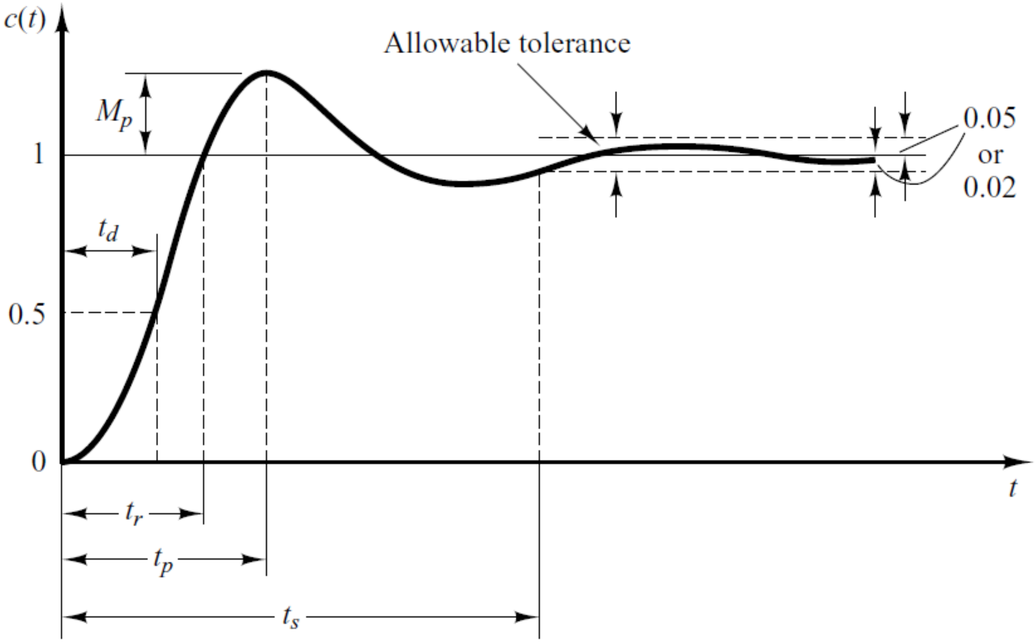


Figure 57: Transient Response Properties (Ogata, 2010)

Each of these transient response properties has a corresponding equation, as given below:

Table 13: Transient Response Properties (Ogata, 2010)

Property	Expression	Definition
$\boldsymbol{\omega_d}$	$\omega_n\sqrt{1-\zeta^2}$	Damped frequency
$\boldsymbol{t_p}$	$\frac{\pi}{\omega_d}$	Time of peak
$\boldsymbol{M_p}$	$e^{-\pi\zeta/\sqrt{1-\zeta^2}}$	Peak overshoot
$\boldsymbol{t_r}$	$\frac{1}{\omega_n\sqrt{1-\zeta^2}}\left(\pi-\tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)\right)$	Rise time (from 0% to 100% of an underdamped second order system)
$\boldsymbol{t_s}$	$\frac{-\ln(Tolerance)}{\zeta\omega_n}$	Settling time

It is possible now to create the scaling factors for each of these properties using the scaling laws developed for the natural frequency and critical damping ratio. These scaling factors are listed below:

$$\omega_d = \lambda^{-1}$$

$$t_p = \lambda$$

$$M_p = 1$$

$$t_r = \lambda$$

$$t_s = \lambda$$

With these scaling factors for the fundamental controller properties, it is possible to determine how the scaled testbed results can be used to obtain full scale system dynamics. An important aspect of these scaling factors is that they are all based solely on λ, with no dependence on the scaling parameter λ_g. Because these controller properties are based on the time responses to forces and impulses from actuators, this dependence on the scaling factor for system frequency agrees with the physical intuition of the relationship between frequency and time. Additionally, since peak overshoot is a percentage value, it also agrees with the physical intuition that it should have a scaling factor value of 1.

It is possible to use the Matlab step function to determine the step response of both a scaled and an unscaled system. An analysis was performed to compare the effect of the scaling of the natural frequency and damping ratio on several of the factors listed above:

time of peak, peak overshoot, and the rise and settling times. The damped natural frequency was not analyzed using the step response function in Matlab, since it is a parameter describing the system itself, not the step response. For this analysis, the values for both λ and λ_g are set to 0.6, the same value highlighted in Figure 55 by the black circle as an example. Additionally, the standard SPHERES satellite controller values were used for the natural frequency and damping ratio (0.2Hz and 0.75, respectively, for position control, or 0.4Hz and 0.75, respectively, for attitude control).

Figure 58 shows the step response for the position controller, when given both scaled and unscaled inputs for the system mass, natural frequency, and damping ratio, whereas Figure 59 shows the step response for the attitude controller when given the same inputs. As shown in both of these figures, the scaling has the effect of hastening the rise time, for example, because of the decrease in scaled system mass.

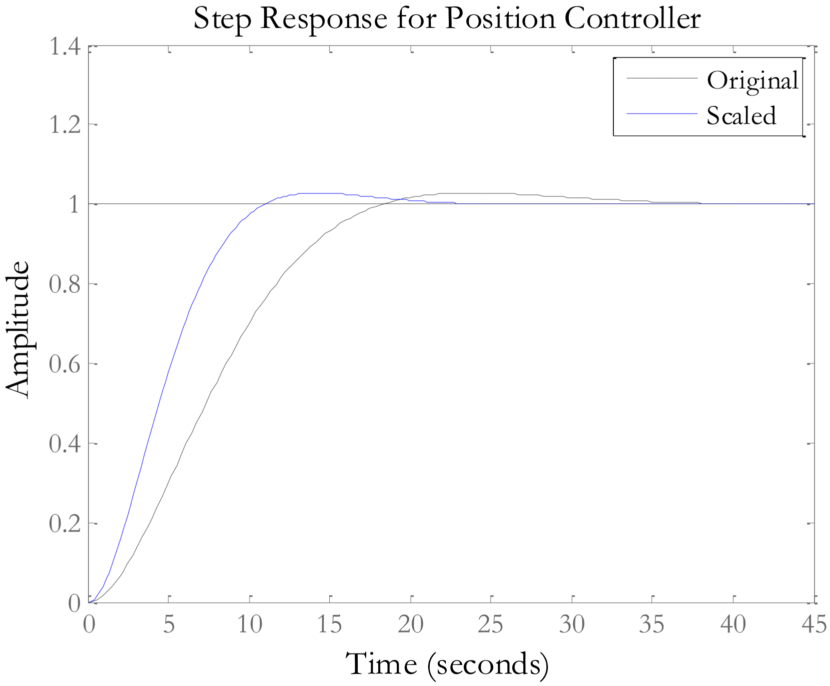


Figure 58: Step Response for Position Control with Original and Scaled Inputs

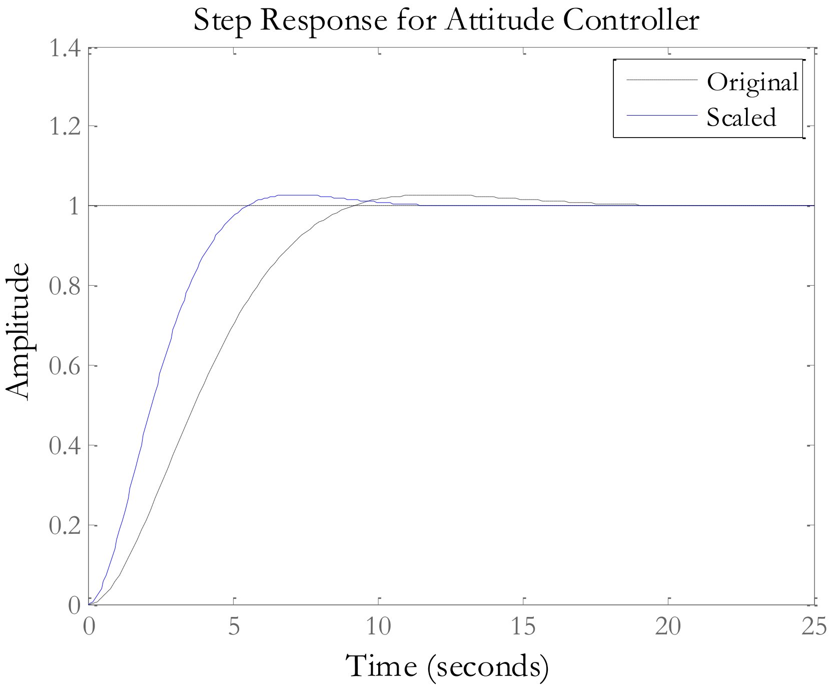


Figure 59: Step Response for Attitude Control with Original and Scaled Inputs

Table 14: Position and Attitude Control Step Response Scaling Results

Parameter	Ratio of Scaled to Original Values: Position Control	Ratio of Scaled to Original Values: Attitude Control	Value of Scaling Factor
t_p	0.600000000000000	0.600000000000178	0.6
M_p	1.000000000000011	1.000000000000000	1
t_r	0.599999999999999	0.600000000000006	0.6
t_p	0.600000000000355	0.600000000000178	0.6

Table 14 shows a comparison between the position and attitude controller properties when subjected to a unit step input. These outputs were obtained using the `format long` command in Matlab to show the machine precision error on the ratios of scaled to unscaled parameter values. Since the difference between the expected value, given as the value of the scaling factor, and the observed ratio for each of these parameters is at the level of machine precision, it may be said that the approach for scaling the control law response parameters can provide predictable results.

There are limitations to this analysis, however, stemming from the assumption that the plant is defined solely by the damping ratio, natural frequency, and total mass of the system. The above results do not take into account actuator geometry or performance, as would be likely when comparing testbeds of different scales from each other or from the flight hardware. In this analysis, the transfer function representing the controller was taken to be the following:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2},$$

where $H(s)$ is the transfer function, s is the Laplace variable, and ω_n and ζ are the natural frequency and damping ratio, respectively. As can be seen in this equation, actuator effects do not play a role. Additional research into this area is required to improve the scaling laws and incorporate additional configuration parameters that describe the satellite.

5.7 SPHERES Simulation Implementation

In order to demonstrate that the control scaling presented in Section 5.6 can be applied to an actual satellite system, it is possible to implement the scaling of the SPHERES PD control law in the SPHERES simulation. This Matlab-based simulation relies on C/C++ based libraries that enable it to run using identical code to that which runs on the satellites themselves. Up to three satellites can be modeled in the simulation at once, each with full 6DOF motion capabilities, which enables the simulation to test ISS code efficiently on the ground. Additionally, the simulation operates at a 1ms time step to match the SPHERES clock and fastest rate at which thruster commands can be sent. To simulate at as high fidelity as possible, the simulation incorporates noise on sensor measurements, thruster performance, and metrology receiver measurements (Katz, 2013). The dynamics models of the satellite, sensors, and thrusters have been calibrated using flight data obtained during 2006 test sessions aboard the ISS.

The SPHERES simulation relies on mass, inertia, and control gains to model each satellite. These variables are user-modifiable, allowing multiple control schemes and hardware configurations to be simulated. Because of this high fidelity and versatility, it is possible to use the simulation to test the control scaling laws through the modification of these parameters. As described in Chapter 4, it is possible to use a single SPHERES satellite to represent a Phoenix satlet. Because such satlets are of a comparative size to that of a SPHERES satellite, little scaling is required to determine satlet performance characteristics. For larger satellites, such as the Phoenix servicer/tender satellite, larger scaling values are required to determine on-orbit performance.

In order to demonstrate the effect of large changes of satellite scale on step response characteristics, a total of six simulations were run. These simulations scaled the SPHERES satellite model from its actual size to that of a satellite ten times as large. The hybrid scaling model was used to determine the scaling values for the mass, inertia, and the four PD control gains with $\lambda = \{1,2,4,6,8,10\}$. The simulation set therefore represents a series of six Phoenix servicer/tender satellites that are dimensionally larger than a SPHERES satellite by a ratio equal to these λ values. Step response characteristics can be determined by simulating a satellite moving one meter from an initial rest position to another rest position. This simulation therefore relies on the model of the SPHERES satellite and the scaled controller to perform the unit step translation.

Figure 60 shows the position of the simulated SPHERES satellite with its control gains and physical parameters scaled by six values of the hybrid modeling parameter λ. The $\lambda = 1$ curve shows what a full scale servicer/tender's response would be if it were to be the same size as a SPHERES satellite, while the $\lambda = 10$ curve shows the response if the servicer/tender were ten times the size of a SPHERES satellite. As can be seen, increasing the size of the servicer/tender has a marked effect on the unit step translation response. The rise time, settling time, and time of peak are all affected by the change in scale, while the overshoot does not show the same dependence on scale.

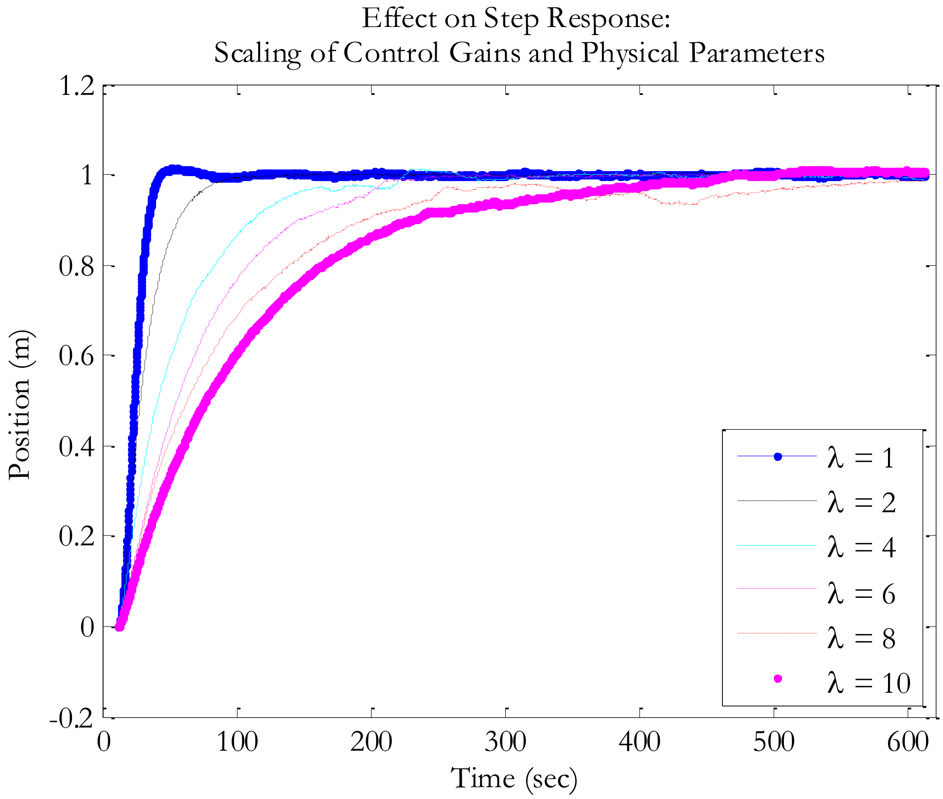


Figure 60: Effect on Step Response: Scaling of Control Gains and Physical Parameters in the SPHERES Simulation

Figure 61 shows the scaled controller step response parameters that were obtained through analysis of the step response curves plotted in Figure 60. The rise time, settling time, and time of peak are plotted against the left y-axis with units in seconds, while the percent maximum overshoot is plotted against the right y-axis with units in percent. The rise time follows the expected scaling response of an increase corresponding directly to the scale λ of the satellite. The maximum overshoot also follows the expected scaling response in that it is nearly independent of scale: all of the computed maximum overshoots are within 1.5% of the maximum overshoot of a standard, unscaled SPHERES satellite. The settling time and time of peak, however, do not directly follow the expected increase with λ. Though both of these curves show a net increase with scaled satellite size, the increase is not monotonic. Instead, there are deviations away from the expected increase that can be explained by noting the method by which these two parameters are computed. When determining the settling time from experimental data, noise sources that are introduced in the simulation to increase the fidelity of the model can lead to the step response to settle at the target value at a time later than expected. Similarly, the noise can lead to peak values occurring later in a simulation or experiment than expected. For example, a low probability event that introduces a large amount of noise may occur when the satellite has nearly completed a maneuver that causes it momentarily to deviate from its target position so that a new peak value is recorded. Such a deviation may be from airflow effects when operating aboard the ISS or from measurement errors of the satellite state. As a result, the time of peak and settling time may be affected by noise in the simulation or experiment, yielding scaling results that are not directly in accordance with the analytically derived scaling factors.

A method for showing the magnitude of the discrepancy between the expected scaling factor and the simulation results is to analyze the trend line for each of these curves. Since the scaling factor relationship is linear, a linear fit trend line can be determined for each step response parameter.

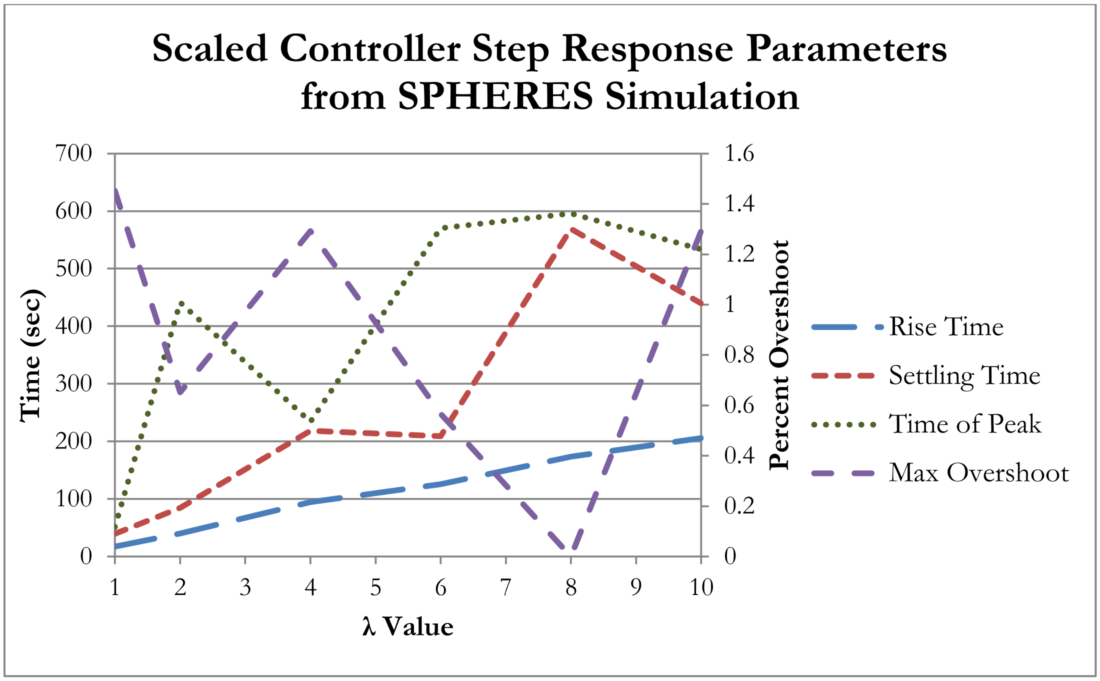


Figure 61: Scaled Controller Step Response Parameters from SPHERES Simulation

A related method is to analyze the ratio of each parameter at a given scale to that of an unscaled SPHERES satellite. Since the scaling factor relationship is linear with an expected slope equal to λ, a linear fit trend line can be computed for each step response parameter ratio and its slope analyzed to determine the experimental scale factor. Figure 62 shows the ratio of each step response parameter when normalized by the unscaled ($\lambda = 1$) parameter value. The same non-monotonic nature is seen in this figure, though it is possible to observe the linearity of the scaling in the normalized plot directly.

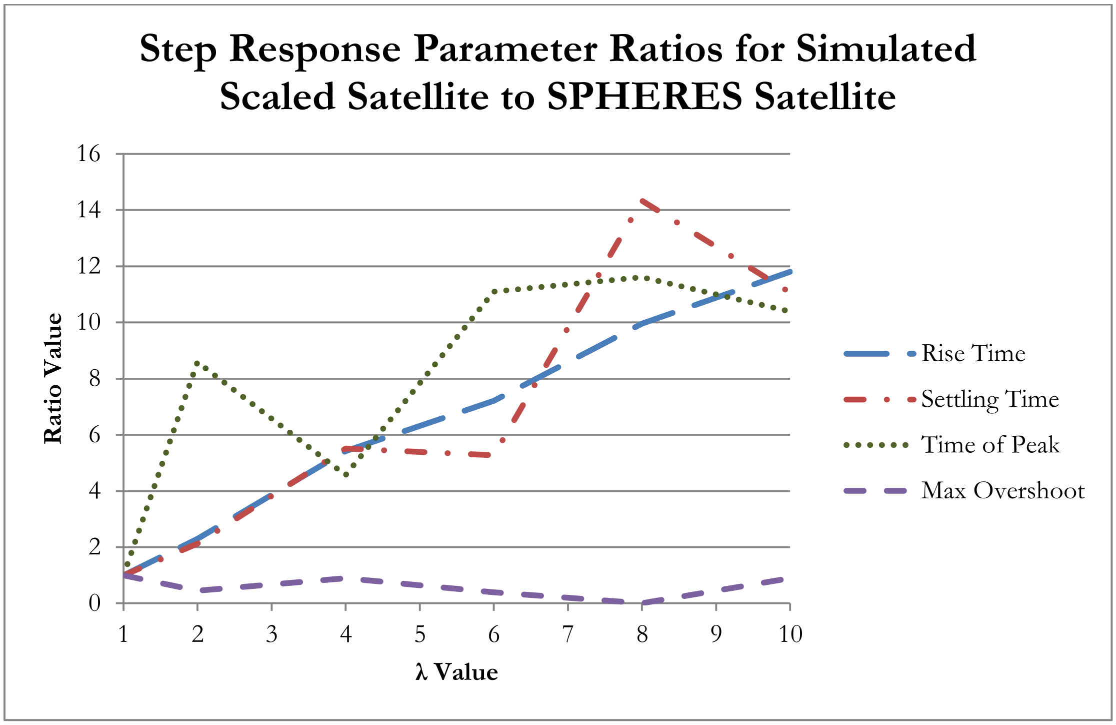


Figure 62: Step Response Parameter Ratios for Simulated Scaled Satellite to SPHERES Satellite

Table 15 shows the trend line equation and coefficient of determination (R^2) value for each parameter when normalized against a standard SPHERES satellite. These trend lines indicate how scale change affects the step response parameters. Based on the analytical derivation for the scale factor for the rise time, settling time, and time of peak, it is expected that each of these times would scale with λ, so the expected trend line slope for these normalized parameters is 1. Additionally, it is expected that the y-intercept of the equation would be 0. The results of the simulations show that the ratio in each of these three cases does scale approximately with λ, since the slopes are approximately equal to 1. The y-intercepts for the rise time and the settling time are also much closer to 0 than the time of peak, which also has a much poorer fit as indicated by a lower R^2 value. It is believed that the modeled noise as described above prevents the system from achieving the precision shown in Table 14 that would indicate direct scaling with λ. Additionally, the maximum overshoot, which was analytically determined to be independent of scale, is also shown to be affected by the noise in the simulation, since the y-intercept is further from 1 than the slope is from 0. It is believed that the low goodness of fit is a consequence of the points all occurring between 0 and 1 in the normalized data set and the simulation data indicating that the maximum overshoot never exceeded 1.46% at any scale.

Table 15: Normalized Step Response Parameter Linear Fits

Parameter	Linear Trend Line Equation	R^2 Value
Rise Time	$y = 1.0278x + 0.0397$	0.9937
Settling Time	$y = 1.3338x + 0.3424$	0.8077
Time of Peak	$y = 0.9143x + 3.1494$	0.569
Maximum Overshoot	$y = -0.0329x + 0.7728$	0.0871

Based on the SPHERES simulation results, the scaling factors developed for the PD control gains have been shown to result in step response characteristics that follow their respective scaling factors. The linear trend line analysis on the normalized parameter values supports the analytical scaling factor analysis. Therefore, the hybrid modeling approach for scaling control systems has been demonstrated with a high fidelity model of a satellite that can be tested on the ground and aboard the International Space Station. Further work is required to scale the control system in the presence of scaled actuators and sensors, which have not been incorporated into the analysis at the time of this thesis. With these scaling factors, full scale behaviors can be determined based on small scale experimental and simulated testing with the RSA risk reduction facility. This section provides validation for the scaling approach, but future work must be performed to validate the scaling laws themselves across a wide range of scaling factors.

6 Chapter 6 - Conclusion

6.1 Summary of Thesis Results

This thesis presents new research into the area of risk reduction of Robotic Servicing and Assembly missions through the development of a new, incremental, and iterative testbed. A literature review revealed a gap in the existing research, leading to the need to address the need for an integrated, scalable, on-ground and on-orbit testing facility that enables incremental, iterative testing while maintaining traceability to the final RSA project. This thesis therefore has provided risk reduction capabilities for future RSA projects by creating a sequence of such testbeds that uses the existing infrastructure to conduct testing in an authentic, yet risk tolerant environment.

The research in this thesis supports the technologies being developed for RSA missions like the DARPA Phoenix mission by providing a risk-tolerant and extensible series of testbeds for reducing the risks associated with their implementation on multi-satellite space architectures. This testbed sequence provides risk reduction for these technologies for their initial application but also for further refinement over additional testing increments and iterations. For example, the technologies that have been described in this thesis, including rotation control, tipoff control, resource aggregated reconfigurable control, and autonomous path planning, can be tested on-orbit in the near future with the Halo, UDP, and SPHERES satellites prior to their first demonstration on a full RSA mission, but further development of these technologies can occur by testing improved algorithms from the broader community based on the results of the first demonstration.

Chapter 1 presents the motivation for the thesis, including a discussion of the existing literature and the current research gap. It laid the framework for the research conducted as part of this thesis. Chapter 2 discusses the importance of properly creating testbed requirements and enumerated the three key RSA risk reduction testbed requirements of maintaining testbed traceability to the final project, leveraging existing infrastructure where possible, and providing opportunities for incremental, iterative testing. Chapter 3 focuses on the development of the facility with which to conduct this traceable and scalable, incremental and iterative testing both on the ground and aboard the International Space Station. Chapter 4 presents the testing that has occurred on the ground to test rotation control, tipoff control, autonomous risk allocative path planning, and resource aggregated reconfigurable control. It also presents the forthcoming testing that will occur on the ISS to both verify the ground testing, but to expand the testing of RSA technologies to those which require more than two satellites in a full 6DOF microgravity environment. This chapter shows how the incremental and iterative testing process can be used to reduce the risks associated with RSA missions through a carefully planned sequence of testing that is designed to build successively. Chapter 5 discusses how to scale the results obtained from testing on a testbed to reduce the risks associated with full scale testing on-orbit. By using the Buckingham Pi theorem, dimensional homogeneity, and hybrid scaling laws, this chapter shows a mechanism for scaling physical properties of a testbed and how control laws can be scaled as a means of identifying how future systems will behave on-orbit.

The infrastructure developed for this thesis and the risk reduction facility provides a platform for testing high risk and high payoff technologies. The research presented in this thesis creates a new paradigm for testing emerging technologies that will benefit multiple RSA programs that either cannot be tested on a flight demonstration or are inherently too risky to be incorporated into a mission prior to extensive prior testing. Additionally, the facility is living in that new technologies can be tested using the complete testing sequence both on the ground and aboard the ISS that stem from initial RSA testbeds. Therefore, the ability to integrate and test new technologies beyond the initial set presented in this thesis enables many future RSA mission technologies to be tested, providing incremental, demonstrable results for many current and future RSA mission stakeholders.

6.2 Limitations of Research

The research presented in this thesis has been conducted to reduce the risk of RSA formative technologies. There are several limitations inherent with the testbed development process and the ability to implement the process for future space missions. For example, the presented framework has been shown to be applicable for reducing only RSA mission technological risks. Though the process can readily be expanded to encompass a much more broad range of mission scenarios, further research will be required to demonstrate the application of this methodology to non-RSA mission architectures.

Another limitation is that the scalability analysis presented in this thesis requires a full-scale design that has been matured to at least a preliminary phase. Because the scalability laws are generated solely from geometric considerations, the scaling approach requires that a full-scale system be in place against which a small-scale system can be compared. While it is possible to begin with a small-scale system and determine for which full-scale systems the small-scale tests would be applicable, this reverse approach introduces significant risk itself because of the uncertainty regarding whether or not the final flight design would be as expected when developing the small-scale testbed. Additionally, because only geometric considerations are used to compute the hybrid scaling laws, the approach described here us currently unable to address the scaling of non-physical parameters. Importantly, the need to scale effects which are not dependent on geometry will drive future research into new models of how small-scale testbeds can reduce full-scale risks.

Currently, the ISS is the best long-duration, risk-tolerant, microgravity environment that is available to researchers. As such, the constraints imposed by the ISS create strict limitations as to what hardware and software can be launched, operated, or studied in orbit. Other limitations exist as well, such as the limited crew time available for conducting experimentation, the limited data upload and download capabilities of the ISS, the inaccessibility of the testbed once launched, and the difficulty for testing thermal, radiation, or vacuum environment testing aboard the ISS without significant changes from the ground hardware. Each of these other environments can be tested on the ground, but it is much more difficult to test them aboard the ISS without a large expenditure of time and money to develop support equipment that is rated for ISS operations.

Another area that is a limitation of this research is the inability to determine when a risk has been sufficiently reduced. Over the course of testing across multiple testing increments, a scaled technology is tested to reduce full-scale risks. Currently, however, a mechanism by which to determine when a full-scale risk has been reduced an acceptable amount for a technology to be determined flight ready does not exist. While technology readiness levels provide a means for identifying if a particular technology is ready for flight, there isn't a corresponding level set for how the appropriately scaled technologies can be deemed flight ready. Future research is required to analyze risk quantization for scaled systems.

These limitations can be summarized by a series of disadvantages of the RSA testbed described in this thesis. Currently, the application of the risk reduction process has been tailored solely to RSA missions, and the scalability framework requires a full-scale system already in place for the best flight traceability of testing results. Additionally, the use of the

SPHERES testbed aboard the ISS provides significant advantages over other testbeds, especially those developed specifically for such risk reduction testing, but the ISS levies a heavy penalty in the form of design and operational constraints of any testbed that is to be sent to orbit. Further, the operation aboard the ISS does not expose any of the aforementioned hardware to the true space environment.

6.3 Research Contributions

Despite the limitations presented in Section 6.2, there are a significant number of advantages of the research discussed in this thesis. First, the testbed that has been developed is based on a new, low cost process for reducing RSA technological risks. The testbed has been shown to be effective at reducing several key, enabling technologies that will be used on many missions even beyond those pertaining to RSA. In creating this testbed, the traceability has been maintained directly at all levels with great care being taken to design the incremental testing process properly. The use of the SPHERES facility greatly enhances the capabilities of the testbed, since by leveraging the existing infrastructure, many ground and on-orbit procedures, software, algorithms, and significant portions of the necessary hardware are identical. The use of SPHERES satellites also directly enables the testing of various levels of autonomy, spanning the full range of the Sheridan Levels of Autonomy. Additionally, the testing and scaling process enables multiple iterations of each testing increment to confirm or refute a hypothesis, increasing the result confidence from each increment. These advantages all play a crucial role in providing the necessary risk reduction that will enable RSA mission architectures to be realized and provide meaningful services to both existing and future mission concepts that will further mankind's reach into space.

6.4 Future Work

Testing will always be a critical component of the systems engineering process. The RSA risk reduction testbeds developed as part of this thesis have been developed in support of RSA missions in general and of the DARPA Phoenix mission in particular. Future work to expand the applicability to additional RSA missions as they are proposed will demonstrate the versatility of the SPHERES satellite-based facility on the ground and on orbit as a means of reducing many technological risks. Additionally, since many technologies fundamental to RSA missions are also a part of non-RSA missions, additional research can be performed to reduce risks associated with many other types of missions. For example, the testing conducted as part of the Phoenix-derived increments focused on high levels of satellite autonomy, whereas future missions may be much more dependent on human interaction either from the ground or nearby astronauts. New testing with this facility will be required to reduce risks associated with these new concepts of operation. Additional versatility can be provided to the testbeds as well through the incorporation of new actuators and sensors to afford increased direct traceability to new flight missions. The Halo and UDP additions to the on-orbit facility provide an initial suite of functionality, though new hardware upgrades to the facility will enable further testing for RSA and other mission architectures.

One of the most effective ways to maintain traceability and directly reduce risks associated with RSA missions is to test with actual or scaled modules that will be used in the assembly or servicing process. The SPHERES satellites, enabled by their expansion ports, can assume many assembly and servicing roles. Figure 63 shows an assembly scenario of a modular telescope as the planned demonstration as part of the Assembly of a Large Modular Optical Space Telescope (ALMOST) project. As described in Miller (2008), ALMOST's main objective is to mature the development of in-space robotic deployment and assembly of optical telescopes. Though the ALMOST project also plans to use the SPHERES facility and the Universal Docking Ports, the testbed developed as part of this thesis is capable of providing significant risk reduction for ALMOST. The incremental and iterative development process can mature the guidance, navigation, and control algorithms, but also the proximity operations to enable docking and assembly of the telescope. Initial testing can occur both on the ground and on orbit without the need for new hardware, such as the telescope array elements shown in Figure 63. By testing with scaled mirror segments, highly valuable data can be obtained to determine and reduce the areas of risk for similar, full scale flight missions. Once these first increments have provided a technological foundation, new hardware can be launched to augment the facility and provide further technological risk reduction for a flight mission.

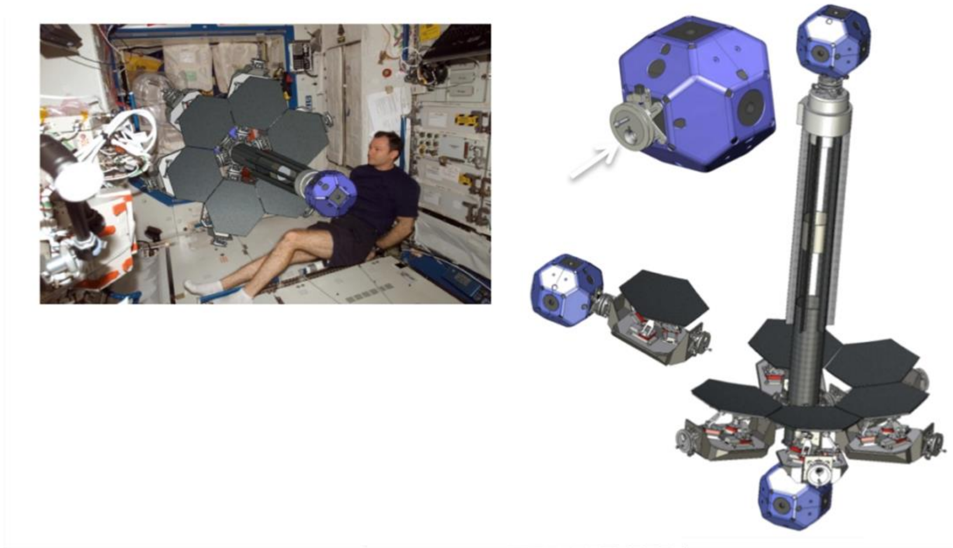


Figure 63: SPHERES Demonstration of Space Telescope Assembly Aboard ISS (Miller, 2008)

While the facility developed for this thesis has shown the ability to reduce the risks associated with RSA missions, there still remains much research potential because of the careful design of the hardware and software infrastructure. The hypothesis which was initially posed has been confirmed through testing, and many new hypotheses may also be tested with this facility. The research thus far with SPHERES, Halo, and the UDPs has reduced several RSA risks, but much more research is expected in the coming years to build incrementally and iteratively on this foundation.

7 Works Cited

1. Baker, W., Westine, P., Dodge, F., *Similarity Methods in Engineering Dynamics: Theory and Practice of Scale Modeling*, Elsevier Science Publishers, New York, 1991.

2. Barnhart, et al., “Changing Satellite Morphology through Cellularization,” *AIAA Space Conference 2012,* AIAA 2012-5262, Pasadena, CA (2012).

3. Barnhart, et al., “Development and Operation of a Micro-satellite Dynamic Test Facility for Distributed Flight Operations,” *AIAA Space Conference 2009,* AIAA 2009-6443, Pasadena, CA (2009).

4. Brown, O. & Eremenko, P., “Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft,” AIAA-2008-7869, *AIAA Space 2008,* SanDiego, CA (2008).

5. Brown, O. & Eremenko, P., “The Value Proposition for Fractionated Space Architectures,” AIAA-2006-7506, *AIAA Space 2006,* San Jose, CA (2006).

6. Brown, O., Eremenko, P., and Callopy, P., ‘‘Value-Centric Design Methodologies for Fractionated Spacecraft: Progress Summary from Phase 1 of the DARPA System F6 Program1,” *AIAA Space Conference 2009,* AIAA 2009-6540, Pasadena, CA (2009).

7. Buckingham, E., “On Physically Similar Systems; Illustrations of the Use of Dimensional Equations,” Phys. Rev. Journal, Vol. 4, No. 4, 1914, pp. 345-376.

8. Buckingham, E., “The Principle of Similitude”, Nature, Vol. 96, No. 2405, 1915, pp. 396-397.

9. Cardin, M., “Flexibility in Multidisciplinary Design: Theory and Experimental Validation”, 16.888 Lecture Note, MIT OpenCourseWare, 2010, < http://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec21.pdf>

10. Chu, J., Guo, J. and Gill, E. “A survey of autonomous cooperation of modules’ cluster operations for fractionated spacecraft”, Int. J. Space Science and Engineering, Vol. 1, No. 1, 2013, pp.3–19.

11. Cockburn, A., “Using Both Incremental and Iterative Development”, *Software Engineering and Technology,* 2008, <http://www.crosstalkonline.org/storage/issue-archives/2008/200805/200805-Cockburn.pdf>

12. Colombina, G.; Didot, F.; Magnani, G.; Rusconi, A., "Automation and robotics technology testbed for external servicing," *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference*, Vol. 3, pp.1954-1963, (1994)

13. Crawley, E., Sigler, J., and van Schoor, M., “Hybrid Scaled Structural Dynamic Models and Their Use in Damping Prediction,” *AIAA Journal of Guidance,* Vol. 13, No. 6, 1990, pp. 1023-1032.

14. Crawley, E., Sigler, J, and van Schoor, M., “Prediction and Measurement of Damping in Hybrid Scaled Space Structure Models, SSL Report 7-88, MIT Space Systems Laboratory, July 1988.

15. David, F., and Nolle, H., *Experimental Modelling in Engineering,* Butterworths, London, 1982.

16. Dermitzakis , K., Carbajal, J., Marden, J., “Scaling Laws in Robotics,” Procedia Computer Science, Vol. 7, 2011, pp. 250-252.

17. DuBos, G., and Saleh, J., “Comparative cost and utility analysis of monolith and fractionated spacecraft using failure and replacement Markov models,” Acta Astronautica, Vol. 68, 2011, pp. 172-184.

18. Eremenko, P., “System F6 Technology Package”, DARPA Proposers’ Day Briefing, November 22, 2011, <http://upload.wikimedia.org/wikipedia/commons/f/fd/System_F6_Proposers_Day_ProgrPr_Overview.pdf>

19. Fine, C., and Whitney, D., “Is the Make-Buy Decision Process a Core Competence?”, *International Motor Vehicle Program,* 1996., <http://dspace.mit.edu/bitstream/handle/1721.1/1626/Make_Buy.pdf?sequence=2>

20. Fredrickson, S., et al., “Mini AERCam: Development of a Free Flying Nanosatellite Inspection Robot,” SPIE, Vol. 5088, 2003, pp. 97-111.

21. Gronet, M., Crawley, E., and Allen, B., “Design, analysis, and testing of a hybrid scale structural dynamic model of a Space Station,” *AIAA Structures, Structural Dynamics and Materials Conference*, 30th, Mobile, AL, Apr. 3-5, 1989, Technical Papers. Part 4 (A89-30651 12-39). Washington, DC (1989).

22. Grubin, C., “Docking Dynamics for Rigid-Body Spacecraft,” AIAA Journal, Vol. 2, No. 1, 1964, pp. 5-12.

23. Guo, J., Maessen, D.C., and Gill, E., “Fractionated Spacecraft: The New Sprout in Distributed Space Systems”, *60th International Astronautical Congress,* IAC-09-D1.1.4, Daejeon, South Korea (2009).

24. Jewison, C., “Halo System Requirements Document”, *Internal Report,* Submitted to NASA, October 21, 2013.

25. Jewison, Christopher. “Reconfigurable Thruster Selection Algorithms for Aggregative Spacecraft Systems”. SM Thesis, MIT, 2014.

26. Jewison, C., McCarthy, B., and Sternberg, D., “MIT Project Phoenix Testing Summary”, *Final Report,* Submitted to Aurora Flight Sciences, August 23, 2013.

27. Jewison, C., McCarthy, B., Sternberg, D., Fang, C., and Strawser, D., “ARMADAS Test Results”, *Final Report,* Submitted to NASA HQ and SMC, May 1, 2013.

28. Jewison, C., McCarthy, B., Sternberg, D., Fang, C., and Strawser, D., “Resource Aggregated Reconfigurable Control and Risk-Allocative Path Planning for On-orbit Assembly and Servicing of Satellites”, *AIAA SciTech 2014,* AIAA-2014-1289, National Harbor, MD (2014).

29. Katz, Jacob. “Achieving Broad Access to Satellite Control Research with Zero Robotics”. Ph.D. Thesis, MIT, 2013.

30. Kunes, J., *Dimensionless Physical Quantities in Science and Engineering,* Elsiver, London, 2012.

31. Horsham, G., Schmidt, G., and Gilland, J., “Establishing a Robotic, LEO-to-GEO Satellite Servicing Infrastructure as an Economic Foundation for Exploration,” *AIAA Space 2010,* AIAA–2010–8897, Anaheim, CA (2010).

32. Larman, C., and Basili, V., “Iterative and Incremental Development: A Brief History”, *IEEE Computer Science Society,* 2003, <http://www.craiglarman.com/wiki/downloads/misc/history-of-iterative-larman-and-basili-ieee-computer.pdf>

33. LoBosco, D., et al., “The Pleiades fractionated space system architecture and the future of national security space,” *AIAA Space Conference 2008,* San Diego, CA (2008).

34. McCarthy, Bryan. “Development of a Space-based Robotic Assembly and Servicing Testbed”. SM Thesis, MIT, 2014.

35. Miller, D., et al., “Assembly of a Large Modular Optical Space Telescope (ALMOST)”, SPIE Space Telescopes and Instrumentation, Vol. 7010, 2008.

36. Mohan, Swati. “Reconfiguration Methods for On-orbit Servicing, Assembly, and Operations with Application to Space Telescopes”. SM Thesis, MIT, 2007.

37. Mohan, Swati. “Quantitative Selection and Design of Model Generation Architectures for On-Orbit Autonomous Assembly”. Ph.D. Thesis, MIT, 2010.

38. Moyer, D., Mauzy, S., “Manned Geosynchronous Earth Orbit (GEO) Servicing Study 2011 Final Report”, NASA Report Number NASA SP 2012-589, Summer 2011.

39. Ogata, K., *Modern Control Engineering, 5th Edition.* Prentice Hall. Upper Saddle River, NJ, 2010.

40. Ono, M., Graybill, W., & Williams, B., "Risk-sensitive Plan Execution for Connected Sustainable Home," *BuildSys '12 Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings,* New York, NY, 2012, pp. 45-52.

41. Ono, M., & Williams, B., "Iterative Risk Allocation: A New Approach to Robust Model Predictive Control with a joint chance constraint," 47th IEEE Conference on Decision and Control, Cancun, Mexico, 2008, pp. 3427-3432.

42. Ono, M., & Williams, B, "Market-Based Risk Allocation for Multi-Agent Systems," *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems,* Toronto, Canada, Vol. 1, 2010, pp. 1423-1424.

43. Ono, M., Williams, B., & Blackmore, L., "Probabilistic Planning for Continuous Dynamic Systems under Bounded Risk," Journal of Artificial Intelligence Research, Vol. 46, 2013, pp. 511-577.

44. Padillo, J., and Diaby, M., "A multiple-criteria decision methodology for the make-or-buy problem", International Journal of Production Research, Vol. 37, No. 14, p. 3203-3229, 1999.

45. Reintsema, D., et al., "DEOS – The German Robotics Approach to Secure and De-Orbit Malfunctioned Satellites from Low-Earth Orbit," *IAC Conference 2012*, IAC-12,D1,1,7,x15613, 2012.

46. Saenz-Otero, Alvar. "Design Principles for the Development of Space Technology Maturation Laboratories Aboard the International Space Station". Ph.D. Thesis, MIT, 2005.

47. Saenz-Otero, Alvar. "The SPHERES Satellite Formation Flight Testbed: Design and Initial Control". SM Thesis, MIT, 2000.

48. Schuring, D., *Scale Models in Engineering: Fundamentals and Applications,* Pergamon Press Inc., New York, 1977.

49. Shoemaker, J., Wright, M., and Sivapiragasam, S., "Orbital Express Space Operations Architecture Program," *17th Annual AIAA/USU Conference on Small Satellites,* SSC03-IV-2, Logan, UT (2003).

50. Sohl, G., Udomkesmalee, s., and Kellogg, J., "Distributed simulation for formation flying applications", *AIAA Modeling and Simulation Technologies Conference 2005,* San Francisco, CA (2005).

51. Sonin, A., “The Physical Basis of Dimensional Analysis”, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2011, <http://web.mit.edu/2.25/www/pdf/DA_unified.pdf>

52. Stroupe, T. Huntsberger, B. Kennedy, H. Aghazarian, E. Baumgartner, A. Ganino, M. Garrett, A. Okon, M. Robinson, and J. Townsend, "Heterogeneous Robotic Systems for Assembly and Servicing," Proceedings of ISAIRAS 2005.

53. Tasker, F., and Henshaw, C.H., “Managing Contact Dynamics for Orbital Robotic Servicing Missions,” *AIAA Space Conference 2008*, AIAA 2008-7908, San Diego, CA (2008).

54. Wada, B., Kuo, C., and Glaser, J., “Multiple Boundary Condition Tests (MBCT) for verification of large space structures,” *AIAA Structures, Structural Dynamics and Materials Conference*, A86-38801 18-39, New York, New York (1986).

55. Wang, X., et al., “A Novel On-Orbit Servicing Technology Based on Micro-Satellite Platform,” *64th International Astronautical Congress*, IAC-13,B4,7B,1,x19997, Beijing, China (2013).

56. Wertz, J., Everett, D., and Puschell, J., *Space Mission Engineering: The New SMAD*, Microcosm Press. Hawthorne, CA, 2011.