Development of a Verification and Validation Framework for Autonomous Soft-Docking of Spacecraft with Uncertain Dynamic Properties

by

Jillian Melanie James

S.B. Aerospace Engineering, Massachusetts Institute of Technology (2010)

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 2016

© Massachusetts Institute of Technology 2016. All rights reserved.
Development of a Verification and Validation Framework for Autonomous Soft-Docking of Spacecraft with Uncertain Dynamic Properties

by

Jillian Melanie James

Submitted to the Department of Aeronautics and Astronautics on May 19, 2016, in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics

Abstract

Although soft-docking in space has been demonstrated in the past, these missions have required detailed information about the target vehicle for success, and often relied on manual control during the final stages. Autonomous docking, however, shows the potential to greatly reduce operation costs while accomplishing complex scenarios. Unfortunately, unknown dynamics and changing parameters stress current attitude control systems for docking applications such as spacecraft servicing, debris capture, and space robotics operations. Adaptation for example may assist with vehicle control under such conditions, however requires careful validation.

Since autonomous soft-docking has limited heritage when there are system uncertainties, risk reduction prior to operation becomes very important for mission success. In this thesis a verification and validation framework was developed for autonomous soft-docking of spacecraft under such uncertainties. The approach combines risk-management techniques, simulation, Monte Carlo analysis, diagnostic tools and experimentation in the micro-gravity environment of the International Space Station (ISS) to create a comprehensive risk-reduction strategy. Development methods are described to provide general guidelines for design of future soft-docking missions.

Additionally, this thesis explores how such verification and validation methods may be used to assess how an adaptive controller can maintain attitude control authority when a spacecraft joins with an object with limited physical parameter information. The goal is to chart a path for controller validation via future spaceflight experimentation. The risk reduction framework and controller analyses and tests are based on working with the Synchronized Position Hold Engage Reorient Experimental Satellite (SPHERES) facility at MIT and on the ISS.

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics
Thesis Supervisor: Alvar Saenz-Otero
Title: Director, Space Systems Laboratory

Disclaimer: This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the United States Air Force.
Acknowledgments

There are so many kind people who have helped me along the way, and I am grateful for every one of them.

I thank my advisor Professor David Miller for his unfailing confidence in my academic abilities, and dedication to ensuring my ultimate success in this program. I would also like to thank Dr. Alvar Saenz-Otero for his guidance and support throughout my research, and Dr. Danilo Roascio for his direction in technical matters. I would also like to acknowledge the influence of Professor Anuradha Annaswamy, Professor Sertac Karaman, Professor Olivier de Weck and Professor Nancy Leveson, as they inspired me through their teachings to apply what I learned about control systems and risk assessment to this thesis.

This work would not have been the same without the contributions of the entire SPHERES team, especially those who have worked with me the last two years — David, Chris, Duncan, Will, Tim, Bruno, Drew, Todd, Jon, Max, Martin, Kevin, Zach, and Antonio. I feel very fortunate to have been a part of such a talented and welcoming team. My thanks goes to Andrew Hilton for taking the time to teach how to use SPHERES, and Duncan Miller for his contributions developing the UDP hardware. I am particularly indebted to David Sternberg and Chris Jewison, and I thank them for their tireless efforts, the numerous technical discussions, and for always inventing time to help me. I would also like to thank Vardaan Gurung for his contributions to building the tools in this thesis, and for making my job as a UROP mentor easy and enjoyable. Additionally, it was Ted Steiner who encouraged me to step out of my comfort zone and present at a conference, and I thank him for the sound technical advice which made its way into this thesis. A special thanks also goes to GA³, and all my officemates and office-neighbors both in the SSL and at MIT LL for both the academic discussions and distractions.

Financial support came from DARPA and the MIT Lincoln Laboratory Lincoln Scholars Program, and I would like to thank them for the generous support that enabled this research. I am especially grateful to Tony Hotz, Dr. William Donnelly,
Dr. Grant Stokes, Lawrence Candell, Craig Perini, Dr. Susan Andrews, Dr. Timothy Hall, Marshall Brenizer, Dr. Mohamed Abouzahra, Dr. Kosti Hennighausen, Dr. Andrew Stimac, Jeffrey Hargreaves, Diane DeCastro, Dr. Robert Legge, Brian Cohen, Brent Parham, Pratik Dave, and Emily Clements for making every effort to help me pursue my masters degree through the Lincoln Scholars Program.

I am especially grateful to all my colleagues at MIT LL, as they have helped me grow both technically and professionally. In particular, Robert Legge, Brian Cohen, Laura Bayley, Deb Blanchard and Jim Sopchak for being such an incredibly supportive team. Thank you does not seem sufficient for Tony Hotz, as he has simply been the best professional mentor I could hope for.

I could not end without thanking my close friends who have helped me along this journey. They have been my comrades, travel companions, comedians, editors, councilors, advocates, and even chefs, supporting me in everything through thick and thin. I am incredibly lucky to have such a wonderful group of friends. A very special thanks goes to Adam, Tina, David, Chris, Antonio, Dianna, Emi, Jason, Eugenio, Ted, Laura, Conor, and Maria who have all directly contributed to the writing of this thesis. I would also like to thank Emi, for helping me win the roommate lottery this past year by being a true friend.

Lastly, I am eternally grateful for my family, for their endless love and support. It was my parents who turned their worlds up-side-down to help me in every way possible growing up and who fueled my passion for rocket science. I thank my mom for her constant encouragement and unwavering faith in me, and my sisters Jennie and Catalina for always being there for me. I could not have done it without them.

Thank you.
Contents

1 Introduction
  1.1 Motivation ........................................ 17
  1.2 Autonomous Soft-Docking Background ................. 20
  1.3 Adaptive Spacecraft Attitude Control ............... 21
  1.4 Verification and Validation ........................ 23
  1.5 SPHERES Experiment Background .................... 24
  1.6 Thesis Objective and Roadmap ....................... 27

2 Verification and Validation Framework Design for Soft-Docking 29
  2.1 Soft-Docking ........................................ 29
    2.1.1 Cooperativeness in Soft-Docking ..................... 30
    2.1.2 Chaser Control Objectives for Soft-Docking .......... 31
  2.2 Risk Reduction Approaches .......................... 32
    2.2.1 Technical-Risk Management ....................... 33
    2.2.2 Risk Assessment Strategies ......................... 33
    2.2.3 Modeling and Experimentation ...................... 35
  2.3 Verification and Validation Framework Design .......... 35
    2.3.1 System Modeling .................................. 37
    2.3.2 Model Diagnostics ................................. 38
    2.3.3 System Characterization through Experiment ............ 39
    2.3.4 Model Scenario Selection .......................... 40
  2.4 Example: SPHERES Docking ............................ 41
    2.4.1 Risk Assessment .................................. 42
4.5 Conclusion ................................................................. 90

5 Adaptive Joint Attitude Control Case Study .......................... 93
  5.1 Controller Development and Test Approach ....................... 94
  5.2 PID Baseline Control .................................................. 96
    5.2.1 Controller .......................................................... 96
    5.2.2 Limitations ......................................................... 96
  5.3 Adaptive Controller Design .......................................... 96
    5.3.1 Adaptive Controller Selection ................................ 96
    5.3.2 Control Laws ..................................................... 98
    5.3.3 Stability Analysis ............................................... 100
    5.3.4 Robustness ........................................................ 101
    5.3.5 Further Improvements ........................................... 101
  5.4 Adaptive Controller Verification and Validation Process ...... 102
    5.4.1 Standalone Simulation Results ................................. 102
    5.4.2 Integrated Simulation Results ................................. 104
    5.4.3 Adaptive Controller Testing on SPHERES Hardware .......... 105
  5.5 Conclusion ............................................................. 107

6 Conclusion ........................................................................ 109
  6.1 Summary ....................................................................... 109
  6.2 Contributions .................................................................. 111
  6.3 Future Work ............................................................... 112

A Appendix: SPHERES Simulations ........................................ 115
  A.1 Spacecraft Dynamics and State ...................................... 115
    A.1.1 Spacecraft Attitude Dynamics ................................ 115
    A.1.2 Spacecraft State Determination ............................... 116
    A.1.3 State-Vector Error Computation .............................. 117
    A.1.4 Spacecraft Parameters .......................................... 118
    A.1.5 Docked Attitude Control Case ................................. 118
List of Figures

1-1 SPHERES satellite, showing locations of thrusters, sensors, and docking port .............................................. 24
1-2 Two SPHERES satellites on the ground-test facility 1-g test environment attempting to dock ............................. 26
1-3 Two SPHERES satellites on the ISS facility 0-g test environment attempting to dock ....................................... 26
1-4 Docking port photographs ..................................................... 27
2-1 Target vehicle capability and cooperation level .................... 30
2-2 Active or passive target modes, where sensing or information transfer, and actuators are required for active translation and pointing ........... 31
2-3 System theoretic risk assessment ........................................... 34
2-4 Modeling, experimentation and analysis aspects of a verification and validation framework ................................. 36
2-5 Soft-Docking system model .................................................... 38
3-1 SPHERES master simulation .................................................. 48
3-2 Block diagram of docking port emulator for imitating function .... 50
3-3 Simulink schematic for docking port emulator .......................... 51
3-4 Satellite visualization for docking ............................................. 52
3-5 Visualization for docking port alignment ................................. 52
3-6 Simulation Framework Information Flow ................................. 54
3-7 Inputs of graphic user interface for input variable selection and distribution ..................................................... 55
3-8 Plots generated by graphic user interface showing distributions

3-9 Example of post-process variable-sensitivity graph

4-1 Hardware testing and simulation in 1-g and 0-g

4-2 Still frame of video plotting of docking port UKF state data (column 1: position, velocity, quaternion, angular rates) and raw data (column 2: position (transformed to center of mass), raw position (rel. to camera), transformed quaternion, and raw quaternion)

4-3 Still frame of video plotting of docking port camera imagery, synchronized with plots in Figure 4-2

4-4 Locations of camera magnets

4-5 Experiment and modeled data supporting essential keepout zone > 6mm (or 4.5mm from the surface) for one magnet

4-6 Yellow "pinch-point" warning label on the left interferes with image processing algorithm

4-7 Target SPHERES bias-corrected raw IMU accelerations before collision during Test 4

4-8 Chaser SPHERES bias-corrected raw IMU accelerations before collision during Test 6a

4-9 Chaser position vs. time for simulated 'truth,' simulated 'actual,' ISS state estimate, and ISS commanded state

4-10 Chaser attitude quaternion for simulated truth, and 'actual,' state estimate of quaternion on ISS, and commanded quaternion during ISS test

4-11 The estimator-computed distance between satellites diverges to an impossible 3.5 meters, even though the SPHERES satellites are rigidly docked throughout the test

4-12 Success histogram for varying SPHERES initial separation distance before GN&C code modifications with only a 56% pass rate for a wide distribution of starting positions
4-13 Histogram from uniform distribution of initial angular rates around Z-axis (tank-axis) showing docking success ......................... 81
4-14 Minimum lance/mag separation distance for runs ......................... 83
4-15 Histogram showing pass/fails for an N-sample run with various drop-
percentages for metrology without any camera data ......................... 84
4-16 Histogram showing pass/fails for an N-sample run with increasing noise level on metrology data ................................. 85
4-17 Histogram showing pass/fails for an N-sample run with various noise levels for relative state estimate of angular rates ......................... 86
4-18 Histogram showing pass/fails for an N-sample run varying thruster force 87

5-1 Decision tree for deciding type of controller ............................ 97
5-2 Adaptive model diagram .................................................. 100
5-3 PID controller response to a step attitude input command for the docked configuration, when gains are tuned for single vehicle............. 103
5-4 Attitude control response to reference step input, showing attitude, rates, actuator commands, gains and error for the adaptive PID controller. ......................................................... 103
5-5 Undocked PID controller attitude response .............................. 104
5-6 Adaptive controller attitude quaternion and reference command through-
out docking scenario. Just after time 110s, the SPHERES dock. At 180s there is a position step command, and at 245s there is an attitude step command. Remember the negative of a quaternion is the same transform, thus q2 is approaching the reference q2 correctly .............. 105
5-7 Hardware setup, with two SPHERES, each with a docking port pointed towards the other, mounted in air-carriages on the low-friction table . 106

B-1 Test 6d chaser position, actual and commanded on ISS, and simulated using ISS initial conditions ................................. 121
B-2 Test 6d chaser quaternion, actual and commanded on ISS, and simu-
lated using ISS initial conditions ............................................. 122
# List of Tables

2.1 Rendezvous and docking sequence ........................................ 32  
2.2 Consequence level of failures ............................................. 36  
2.3 Systems theoretic risk chart for SPHERES satellite Soft-Docking ... 43  
3.1 Monte Carlo input variables .................................................. 57  
3.2 Post processing scripts ....................................................... 60  
4.1 Ground testing ................................................................. 66  
4.2 Stochastic simulation testing for risk assessment ....................... 79  
5.1 Reference command combinations ......................................... 95  
A.1 SPHERES Physical Parameters ............................................ 118
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-g</td>
<td>Micro-gravity</td>
</tr>
<tr>
<td>1-g</td>
<td>Gravity on Earth</td>
</tr>
<tr>
<td>3DoF</td>
<td>Three degree of freedom</td>
</tr>
<tr>
<td>6DoF</td>
<td>Six degree of freedom</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defense Advanced Research Projects Agency</td>
</tr>
<tr>
<td>DART</td>
<td>Demonstration of Autonomous Rendezvous Technology</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>FMEA</td>
<td>Failure Modes Effects Analysis</td>
</tr>
<tr>
<td>FMECA</td>
<td>Failure Modes Effects and Criticality Analysis</td>
</tr>
<tr>
<td>GN&amp;C</td>
<td>Guidance, Navigation and Control</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>IC</td>
<td>Initial Condition</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>ISS</td>
<td>International Space Station</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>MRAC</td>
<td>Model Reference Adaptive Controller</td>
</tr>
<tr>
<td>MRP</td>
<td>Modified Rodriguez Parameter</td>
</tr>
<tr>
<td>NASA</td>
<td>National Air and Space Administration</td>
</tr>
<tr>
<td>PD</td>
<td>Proportional Derivative</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>SPHERES</td>
<td>Synchronized Position Hold Engage Re-orient Experimental Satellites</td>
</tr>
<tr>
<td>STAMP</td>
<td>Systems Theoretic Accident Modeling and Processes</td>
</tr>
<tr>
<td>STPA</td>
<td>STAMP-based Process Analysis</td>
</tr>
<tr>
<td>UDP</td>
<td>Universal Docking Port</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman Filter</td>
</tr>
<tr>
<td>USAF</td>
<td>US Air Force</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>Verification and Validation</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Space mission designs are becoming increasingly complex as advancing technologies make exciting new operational concepts possible. Autonomous soft-docking, is of particular interest because it enables spacecraft reconfiguration, servicing, and debris capture. Such complex missions require a robust verification and validation framework to reduce risks. Many risk-reduction methods currently exist, but applying them in an integrated fashion to autonomous soft-docking under uncertainty to develop a verification and validation framework will help identify obstacles and assist in design of future missions.

1.1 Motivation

Today, there are over four thousand active spacecraft in flight [1]. There are over ten thousand pieces of debris orbiting the Earth, large enough for tracking [1]. In order to reduce the unnecessarily premature degradation of active spacecraft into space debris, autonomous spacecraft servicing has been suggested as a means of fixing satellites without the expense and high risk of manned missions. Another approach, of course, would be to clean up the debris. This might also involve docking in some manner, and, given the high volume of debris, autonomy may be useful. Another evolving mission concept is that of spacecraft assembly and reconfiguration. Imagine the potential of constructing a large space telescope from smaller satellites. Vari-
ous universities including Cornell, UT-Austin and Texas A&M have teamed up with NASA to demonstrate docking technologies with cubesats on programs such as On-Orbit Autonomous Assembly from Nanosatellites (OAAN) and Low Earth Orbiting Navigation Experiment for Spacecraft Testing Autonomous Rendezvous and Docking (LONESTAR) [12], [39].

While autonomous spacecraft soft-docking has been identified as a key enabler for these new mission types, it has proven challenging. In 2005 NASA’s Demonstration of Autonomous Rendezvous Technology (DART) mission resulted in premature termination after a collision during approach [6]. The publicly released overview of the mishap report states that "inadequate systems engineering (including a lack of implementation of software requirements, configuration control, validation of math models and testing) was a significant causal factor in the mishap." The report continues on to say "the [Mishap Investigation Board] determined that analyses to identify possible hardware/software faults failed to consider a sufficient set of conditions that could lead to the mishap. For example, the analyses focused on the effects of a complete loss of functionality of the navigation system’s components, but did not address the impact of a degraded functionality of those same components" [6]. Even successful rendezvous and docking missions like Orbital Express have had their issues, as a "sequence of navigation and sensor problems... occurred during one of [Orbital Express'] unmated operations" instigating a thorough review [14]. Learning from these mistakes and forging ahead with new developments would be critical for success of future missions.

Back in 2012, NASA published a road map detailing technological development priorities for the coming years. In that report, docking and capture (TA04 4.6.3), as well as on-board autonomous navigation and maneuvering systems (TA05 5.4.3) were among the high-priority research areas outlined [17]. Since then, NASA’s updated 2015 road map stresses the need to "develop the capability for highly reliable, autonomous rendezvous, proximity operations, and capture/attachment to (cooperative and non-cooperative) free-flying space objects" [38]. Additionally, according to Pavone and authors in their 2014 study on spacecraft autonomy challenges, a "criti-
cal step [for autonomous relative guidance] is establishing a rigorous process for their verification and validation (desirably with flight testing)" [40]. Post-flight analysis of an anomaly on Orbital Express revealed that "pre-flight testing did not reveal the sensor and navigation issues that occurred in flight" since it was deemed "infeasible for the spacecraft sensor suite and navigation software to be tested in a fully integrated, hardware-in-the-loop fashion" [14]. Restrictions on time and budget often limit the rigorousness of verification and validation (V&V) testing.

Given uncertain dynamic properties, the problem of autonomous rendezvous and docking becomes even more difficult for development and V&V. To begin, spacecraft generally require attitude control for accurate pointing to accomplish a number of mission requirements. Traditionally, simple proportional-integral-derivative (PID) controllers (or even just PD controllers) are used since they have proven successful on flight programs [37]. Despite the non-linear dynamics of attitude control, these controllers demonstrate excellent performance for simple applications where the spacecraft physical parameters are known in advance so appropriate gains can be computed on the ground. However, in cases where parameter information is not known in advance, parameters change slowly, flexible structures are involved, spacecraft reconfigurations occur, or in the presence of other un-modeled disturbances, these controllers may benefit from modification. A variety of control approaches exist for coping with these uncertainties, ranging from gain scheduling to sliding mode control, each with pros and cons [19]. Among these methods, adaptation proves promising since it can be integrated with other types of control [19], mathematically proven as Lyapunov stable [25], designed for robustness to small disturbances and model uncertainties [22], and still achieve performance objectives [25], [22].

Current research on using adaptation for spacecraft attitude control shows theoretic potential via mathematical proof of stability and simulation [46], [43], [9], [8]. Additionally, considerations such as stability under disturbances and actuator saturation limits have been studied with success [34], [28]. Adaptation has even been studied as a method for position control in robotic space systems [33].

Few implementations of such adaptive algorithms exist operationally in current
space flight vehicles, however, due to the complexity of applications that would benefit from such a controller. Additionally, adaptation has perceived risk and complexity. Advancements in the validation process for adaptive algorithms are needed before they can be safely implemented on critical systems [27]. Flying a prototype with an adaptive algorithm for attitude control would provide insight as to potential risks and benefits of utilizing such methods on space programs for tasks related to jointed maneuver.

The goal of this thesis is to address the aforementioned motivations and research gaps in the confluence of improving verification and validation processes using flight testing of high-risk control algorithms aimed at rendezvous, docking and post-dock maneuvers for complex missions with uncertain dynamics. This includes creating and applying a V&V framework to a soft-dock application with uncertainty as well as developing a simple adaptive algorithm to validate performance in hardware and eventually in orbit for the particular application of controlling docked systems.

1.2 Autonomous Soft-Docking Background

Given the host of new mission possibilities for spacecraft servicing, reconfiguration, assembly, and debris capture, as well as the benefits of autonomy like reduced ground-dependence, research and development in autonomous docking is well warranted. Reasons for such focus on autonomy include potential benefits of increased safety during proximity operations, and when communication restrictions or delays are involved [36]. Additional motivation stems from the cost savings of autonomous servicing, rather than manned servicing [20]. With all these benefits, it is not a surprise that autonomous docking is not new.

In fact, autonomous soft-docking has been an evolving field in the space arena for decades. A NASA Goddard Space Flight Center report on satellite servicing from 2010 outlined a history of missions relevant to autonomous spacecraft docking [11]. After the successful demonstration of docking aspects of approach and capture on Engineering Test Satellite VII by NASDA (the National Space Development Agency of
Japan), other missions tested additional capabilities. In 2003 and then 2005 (the same year as the aforementioned DART incident) the US Air Force Research Laboratory launched XSS-10 and then XSS-11, successfully performed proximity operations with microsatellites. DARPA’s Orbital Express mission in 2007 showcased autonomous spacecraft docking and servicing between the ASTRO and NextSat satellites specifically designed for this purpose. Currently, the Russian Progress MS vehicle is capable of autonomous rendezvous and docking with the ISS to deliver cargo, as Russian vehicles have incorporated autonomy for decades (with manual override options). Other ISS delivery strategies involve manual berthing using the ISS robotic arm. Most recently cubesats are entering the rendezvous (and eventually docking) field, with the LONESTAR program.

Some of the missions previously mentioned involved cooperation and high-levels of certainty. One of the obstacles of spacecraft servicing, however, would be limited information about or control over the target vehicle. In an demonstration experiment run inside the ISS, a satellite successfully docked with velcro to a tumbling target [36]. Others, like Fezjic and Andrade for example, have further developed this research to focus on the control algorithms for such missions [16], [2]. Continued efforts will undoubtedly involve handling target uncertainty.

1.3 Adaptive Spacecraft Attitude Control

Control after docking with an object remains challenging, since, for example, the spacecraft thrusters are no longer aligned in a known orientation relative to the center of mass, significantly changing the coupling between force and torque commands. Additionally, the inertia of the joint vehicle also changes. Docking contact dynamics are also difficult to model. Traditional controllers demonstrate limited performance in such circumstances, as conditions may deviate too far from nominal. Methods for sharing information between vehicles and devising a composite control strategy have been explored [24], however in the semi-cooperative case where information sharing is limited, these methods cannot be used. Other alternatives, such as gain scheduling
or controller swapping also have drawbacks when it is unclear how, or when, to modify controllers. Augmenting controllers with adaptive features which can handle parameter changes and nonlinear dynamics may allow them to perform even after docking with a little-known object.

Currently, adaptive controllers have little heritage in space, despite having flown successfully on aircraft [28]. While adaptive controls have been around since the 1950’s, robust, stable adaptive systems were only formally developed by the 1980’s [4]. Since interest has grown in space robotics missions, research on adaptation for space applications has become more prevalent in the last two decades.

In their paper on adaptive systems, authors Black, Haghi and Ariyur summarize the history, basic theory, implementation, and current development efforts [4]. They describe how state of the art adaptive controllers can be designed using Lyapunov’s method for stability. For robustness, there are various methods to handle disturbances and avoid adaptation runaway. Inputs with sufficient richness (persistent excitation) also help to ensure fast convergence. Beyond these design tools, various adaptive types exist as well. First, there are adaptive controllers and adaptive observers. Within the control regime, control can be indirect or direct, depending on whether or not the unknown parameters must be exactly identified, or the system simply controlled, respectively. Then, there is an assortment of control types to choose from, like model reference adaptive control (MRAC), adaptive sliding mode control, adaptive PID control, adaptive pole-placement, adaptive sliding mode control, and neural networks to name a few.

In the non-adaptive realm, robustness to large parameter changes or un-modeled dynamics means either designing a controller which can handle a wide range of environments (which is by definition inefficient for most cases), or swapping controllers. For instance, a PID without adaptation will have poor performance in regions far from where it was tuned, but with gain scheduling that region would grow. Knowledge of when to switch gains and to what value, however, is critical and potentially risky. Another control type, sliding mode control, handles parameter uncertainty well and is very robust. However, given its discrete nature, it is prone to chatter.
Certain limitations exist for adaptation as well. A persistently exciting input is required for parameter convergence to the actual value. In some cases, persistent excitation is needed simply for convergence at all, and rate of convergence is not guaranteed. Also, if the input is not persistently exciting, care must be given to stop adaptation runaway by implementing dead-zones or bounds for example. Finally, while adaptation may find a more efficient solution, there is a cost (fuel for example) to searching.

One of the greatest concerns for space applications, however, is risk itself, which must be mitigated through validation and test. Simulation and ground testing can only go so far, since dynamics are different in free-fall. Since adaptive attitude control has potential to benefit various space applications, there is value in putting research into validation and test to pave the way for testing in a relevant environment.

1.4 Verification and Validation

Numerous approaches to V&V have been studied across various disciplines. For spacecraft, NASA’s Systems Engineering Handbook provides guidelines for everything from risk reduction to hardware-in-the-loop testing. The purpose of such oversight is to verify that what was designed meets the specifications, and validate that the system behaves as intended. Continuous risk assessment, cycling through identification, analysis, planning, tracking and control, serves to monitor and reduce risks as they arise from the design phase through validation [35]. Additional methods, which have been used in other applications besides autonomous docking, include systems theoretic accident modeling and processes (STAMP) and failure-model effects analysis (FMEA) for risk management [29], [7]. Beyond these methods, in order to reduce the uncertainty that contributes to risk, research has gone into handling uncertainty and designing experiments to effectively reduce uncertainty [48], [47].

Despite progress, challenges still exist in the areas of verification and validation that reduce risks specifically for spacecraft docking [40]. While a high level approach to automated rendezvous and docking is thoroughly described in Fehse’s book, from
dynamics equations and control algorithms to V&V, implementation of such an approach remains difficult due to intricacies in implementation [15]. To overcome these difficulties, several hardware-in-the-loop ground-test setups designed to replicate the complexities of docking have been developed for research, such as one at the Spacecraft Robotics Laboratory of the Naval Postgraduate School [41]. Another test system at the Space Systems Laboratory at MIT (described in more detail in the next section), not only allows 3-degree of freedom (3DoF) soft-dock testing on the ground, but also in space [32], [45].

A contribution to the field by this thesis is reducing risk by developing and testing an integrated V&V framework by combining many of these studied methods for autonomous soft-docking under uncertainty. Further applying these methods in orbit allows improvement and validation of the process.

1.5 SPHERES Experiment Background

System validation implies ensuring the system meets intended goals in a relevant environment. Therefore, hardware testing in space would help with algorithm validation. A facility designed for this purpose by the MIT Space Systems Laboratory called SPHERES, for Synchronized Position Hold Engage Reorient Experimental Satellites, has been operating on the International Space Station since 2006 [42]. The facility consists of ground-testing laboratories (see Figure 1-2) and the ISS test setup (Figure 1-3), including three small satellites and supporting equipment. The SPHERES satellites aboard the ISS have sensors, actuators, and on-board computing capability for navigating in the 6 degrees of freedom of space (shown in Figure 1-1).
A current SPHERES project aims to reduce the risk of autonomous docking in space by designing and testing control systems on a test platform inside the ISS microgravity environment [46], [32], [36]. In the summer of 2015, SPHERES docking ports arrived on the ISS for testing. A hardware checkout test was performed at the end of September 2015 that demonstrated functionality of the docking port mechanism. An additional docking test was conducted in March of 2016.

As SPHERES satellites move around, they consume fuel and thus their mass decreases. Additionally, once docked, the inertia matrix changes drastically. Finally, sensing and docking components (such as cameras, docking ports, and in the future, a robotic arm) may be added to the satellites that also change the mass, location of the center of mass, and inertia of the vehicle.

Each SPHERES satellite comes equipped with a suite of sensors (including a three-axis inertial measurement unit and ultra-sound receivers) that provide accelerations, angular rates, inertial position, and the inertial to body frame attitude transformation. Sensor and model confidence information goes through a Kalman filter to create a state estimate. The state estimate (position, velocity, inertial attitude, and angular rates) is fed back to the controller, which commands the pressurized gas thrusters that maneuver the vehicle. There are no reaction wheels aboard, thus thrusters are responsible for attitude control as well. The controller operates given a reference command program running on-board the satellites. For testing, a detailed MATLAB/Simulink/C-code simulation exists which integrates the actual C-code that runs on the SPHERES with a MATLAB/Simulink dynamics simulator. This high-fidelity simulation includes sensor and actuator noise models, detailed physical parameters, timing delays, and environment non-linear dynamics.
The docking port subsystem is comprised of an additional avionics box which interfaces with the SPHERES core computer, as well as the docking port itself. The avionics is responsible for computing the relative state between the satellite and any target vehicle in the field of view of the docking port camera. It uses a Kalman filter to process the incoming data, and then transforms it into the vehicle coordinate frame. The port itself consists of a camera on one side, and fiducial markings on the other (such that if two satellites were aligned for docking the camera would stare directly at the markings on the opposing satellite). When the markings are in clear, focused view of the camera for the tracking algorithm to locate the markings, the vehicle is said to have ‘target-lock.’ Once sufficient, consistent data has been collected after target-lock, that meets certain estimation criteria, the Kalman filter produces a state result which is sent to the controller. The docking port hardware also has a lance and
hole with a photo-sensor-triggered locking mechanism, such that there is only one correct orientation for docking. Figure 1-4 shows photographs of the docking port from different angles. In the photographs, one can see the lance and hole, the four round ultrasound sensors (symmetric around the front face), the fiducial markings (right) and the camera lens (left). Figure 1-4a gives a clear view of the manual dock button and the lens cover attached to a yellow string.

Using these experimental satellites with docking ports, developers can acquire experimental data to validate control algorithms and processes.

### 1.6 Thesis Objective and Roadmap

The objective of this thesis is to figure out areas of concern for soft-docking missions, develop a systematic approach to identifying them and reducing risk, create tools
for such methods, test out tools in a relevant environment, use methods to assess feasibility of adaptive methods for joint maneuvers and work on validation of applying adaptive controls in orbit. The goal is to enable future on-orbit soft-docking missions through improved verification and validation processes using simulation and experiment.

Chapter 1 introduces the topic of autonomous soft-docking, and its importance
Chapter 2 develops approach to risk assessment and V&V for docking applications
Chapter 3 describes implementation of modeling for V&V
Chapter 4 describes simulation, testing, and results
Chapter 5 describes an adaptive attitude-control case study for jointed maneuver
Chapter 6 summarizes work, contributions, and describes potential extensions

In more detail, Chapter 1 has described how docking and post-dock control are of interest for robotic assembly and spacecraft servicing missions, among others, giving motivation for research into V&V [46], [32], [38]. Chapter 2 follows with development of a risk reduction framework specifically for soft-docking using an integrated approach. In Chapter 3, the development process of building the models and tools for that approach is described. The approach is applied to SPHERES docking to demonstrate software and hardware implementation. Next, in Chapter 4 the results found by cycling back and forth between 3 DoF ground testing, pure simulation, and ISS testing, explain the lessons which contributed to improvements. Chapter 5 describes the design and implementation of an adaptive attitude-control method for using on-board docked SPHERES satellites in the future as a test-case to assess how well such algorithms might work for docking applications, as well as to develop an approach toward validating such algorithms in space. Finally, Chapter 6 ends by synthesizing lessons learned, summarizing contributions, and motivating future work in soft-docking V&V.
Chapter 2

Verification and Validation

Framework Design for Soft-Docking

Numerous independent risk reduction techniques exist which could apply to controller development for soft-docking. These include both failure-based and systems-based approaches, as well as experimental design for key parameter identification. Here, the design process for developing an integrated risk-reduction framework for control-system testing is examined.

Existing methods are introduced in the context of soft-docking, an integrated method is described, and then applied to SPHERES satellites for soft-docking. The methods here serve as structure for the testing and analysis in this thesis.

Before delving into risk reduction, it is helpful to define the application itself to discern the types of requirements that must be met.

2.1 Soft-Docking

Soft-docking (as opposed to colliding) may happen when the relative velocity and angular rates between two vehicles are low enough to allow a mechanical connection to join them without any structural damage. A similar concept, termed ‘berthing,’ means that the vehicles get close together, and another maneuverable device (such as a robotic arm) positions the vehicles for the final link [15]. In order to reach a
stage where this can happen, the chaser vehicle must gain enough information about the target to approach, either through its own sensors, or from communication with the target. Therefore, the target spacecraft’s cooperativeness and technical capability levies requirements on the chaser for soft docking.

2.1.1 Cooperativeness in Soft-Docking

Much study and research has gone into cooperativeness in soft-docking, as evidenced from discussions in Nolet’s thesis [36]. The level of spacecraft cooperativeness and capability maps out a space of evasive, passive and collaborative target-chaser maneuvers, as shown below in Figure 2-1. It is assumed the chaser is fully capable.

![Figure 2-1: Target vehicle capability and cooperation level](image)

Three key aspects of capability are the ability to sense, actuate, and communicate. Sensors and filtering on-board the target vehicle can provide information about the vehicle state (attitude, angular rates, position and velocity), in relative (to the chaser), local, or absolute terms. If the target vehicle has fuel and a working actuation system, it may translate, rotate or both to assist with soft docking. Finally, the target vehicle may transfer information about its state to the chaser, receive state information about the chaser, and even respond to commands from the chaser. When both the chaser and the target spacecraft have all three capabilities, they may be fully collaborative.

For example, if the target communicates all sensed state information with the chaser vehicle and additionally has full attitude and position control, it is said to be fully collaborative and cooperative (far right column). Even limited target-sensing may be collaborative, as the target may help in pointing towards the chaser using its actuators. A piece of space debris with no fuel, sensing information, or ability to
communicate would be classified as passive and uncooperative. Finally, evasive vehicles would use their sensing capabilities and actuators to actively avoid docking. For soft-docking, this thesis examines both collaborative and passive regimes, neglecting evasive maneuvers.

**Target Motion**

For the target to collaboratively point towards the chaser, it requires relative state information. The information may be obtained from the chaser, or by filtering data from the sensors (like an inertial measurement unit or vision system) on-board the target. Additionally, the target would require a functional actuation system, like reaction wheels or thrusters for attitude control. Without working actuators or relative state information, the target may be considered passive, maintaining a fixed orientation and position, or tumbling. Both active and passive target motions are shown in Figure 2-2.

![Figure 2-2: Active or passive target modes, where sensing or information transfer, and actuators are required for active translation and pointing](image)

**2.1.2 Chaser Control Objectives for Soft-Docking**

Successful autonomous soft-docking can be broken into the far-field approach, near-field sensing and motion, docking, and finally post-dock operations [15, p. 9]. Each
phase has success criteria as well as control-related objectives, depending whether the
approach is cooperative or not, outlined in Table 2.1.

Table 2.1: Rendezvous and docking sequence

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Cooperative Objectives</th>
<th>Uncooperative Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far-Field</td>
<td>Approach close enough to enable near-field sensors</td>
<td>Approach using information from ground and the target about target location</td>
<td>Approach using information from ground only</td>
</tr>
<tr>
<td>Near-Field</td>
<td>Sufficient knowledge of relative state to position vehicle for docking</td>
<td>Approach using chaser and target sensors for relative state knowledge</td>
<td>Approach using chaser on-board sensors only</td>
</tr>
<tr>
<td>Docking</td>
<td>Successful capture by mechanism</td>
<td>Dock using chaser and target mechanism</td>
<td>Chaser docks to target</td>
</tr>
<tr>
<td>Post-Dock</td>
<td>Stable position and attitude joint control</td>
<td>Chaser and target coordinate actuation commands</td>
<td>Chaser handles all actuation</td>
</tr>
</tbody>
</table>

Turning these objectives into requirements, by adding metrics and tests, helps converge on a design. Ensuring that these requirements are met is the process of verification.

### 2.2 Risk Reduction Approaches

Once some requirements have been outlined for soft-docking, an approach for algorithm development and technical-risk reduction can be developed. Various risk management techniques already exist, ranging from failure-based assessment (FMECA - Failure Mode Effects and Criticality Analysis) to systems approaches (STAMP - Systems-Theoretic Accident Modeling and Processes) [7] [30]. Careful design of experiments and system modeling can reduce uncertainty on high-impact parameters (Sobol) and thus reduce risk [48][47].
2.2.1 Technical-Risk Management

The idea behind risk management and reduction is to identify hazards, likelihoods and consequences to create solutions that minimize harmful outcomes [35]. A simple risk management practice is to simply identify possible failures, and diagram them by estimating likelihood and consequence. Likelihood is given a value of 1-5, where 5 is extremely likely, and consequence a value of 1-5, where 4 is mission critical and 5 is detrimental even beyond the mission [35].

Once risks are brought to the attention of the team, measures to better quantify and reduce or eliminate risks can be taken. The process of identifying, quantifying, and reducing risks continues throughout the program.

While this high-level approach to technical-risk management brings risks to the foreground for discussion, more detailed risk analysis in combination with experimentation can help address the underlying causes.

For this thesis, a detailed risk reduction plan can be outlined as follows:

- Identification of influential variables that affect risk
- Determination of ranges and distributions of these variables
- Analysis of system sensitivity to identified variables
- Experimentation or redesign to reduce uncertainty or influence of critical variables

Beyond developing mitigation processes, improving observation of problems early in design also reduces risk. All aspects from identification to mitigation, to tracking are considered for risk reduction.

2.2.2 Risk Assessment Strategies

Many risk assessment strategies exist across various fields and agencies. For creating a comprehensive approach for this thesis, two popular assessment types were studied, integrated, and applied to the soft-docking problem.
The first method, called Failure Mode Effects and Criticality Analysis, FMECA, is a risk assessment process outlined by the Department of Defense [7]. The process involves listing and tracking potential failures, determining their consequence, and providing a mitigation strategy. The addition of a criticality metric allows developers to compare importance of proper functioning of various components.

While FMECA provides a framework for tracking risks associated with particular failures, sometimes problems occur because failures went unnoticed or mitigation control strategies were not built into the system. Additionally, emergent failures may arise at the system level due to interactions between subsystems. Software ‘failures’ are particularly difficult to track, since software generally does exactly what it was programmed to do, and therefore does not fail. These aspects are not captured well using the FMECA approach.

A systems control theory approach to risk reduction, however, entails identifying risk factors, and ensuring that there are methods to observe, mitigate, and control them [29]. Figure 2-3 shows the cycle of risk mitigation using this approach.

![Figure 2-3: System theoretic risk assessment](image)

Much like control systems, the concepts of observability and controllability are applied to risk factors to ensure they are mitigated. If, for example, there is no way to know that spacecraft attitude pointing is poor, then there is no way to control this risk. However, if attitude data is captured, then poor pointing may be observed, and mitigation techniques applied. Leveson and team outline a process by which designers can apply this risk assessment technique to their own applications [30]. In Section 2.4.1 this process is applied to an example soft-docking mission using SPHERES satellites.
2.2.3 Modeling and Experimentation

Fueling any risk assessment is the system itself which is the source of all risk factors and potential failures. Developers must determine what the failures and risk factors are before any assessment is possible. Literature, historical evidence, experiment and modeling all provide insight as to where things might go awry. Another approach is to start with the requirements and think about what happens when these are not met.

Analysis may also come from simulation and test. Two typical approaches are vertex testing (checking bounds) and Monte Carlo (stochastic) testing [48]. Stochastic testing is easier in simulation than in hardware (given time constraints), thus information gained from simulation drives which hardware tests are selected. Chapters 3 and 4 demonstrate how the approach described in the remainder of this chapter of simulation and experiment may be used for reducing risk in soft-dock applications.

2.3 Verification and Validation Framework Design

Verification and validation (V&V) refer to the process of ensuring the design meets requirements, and also that it acts as intended in a relevant environment [35]. Since spacecraft docking occurs in a free-fall environment, there are particular barriers to testing in a relevant environment. Simulation of the environment then plays a greater role in system validation. Fehse explains the basics of verification and validation process for automated spacecraft rendezvous and docking in Chapter 10 of his book [15]. The key elements are the development of a detailed system model and assessment tools for checking performance and identifying risks. Experimentation helps in ensuring the accuracy of the model. Finally, assessment of resulting runs provides more information for further investigation. These three areas which correspond to sections in this chapter and contribute to development of a sound verification and validation framework are outlined in Table 2.3.
Figure 2-4: Modeling, experimentation and analysis aspects of a verification and validation framework

Spanning this entire framework is risk management. As information is gained about the system through such experiment and simulation, risks can be estimated for tracking and mitigation purposes. For the case of soft-docking, some risks and associated consequence levels are outlined in Table 2.2 below:

Table 2.2: Consequence level of failures

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data storage failure</td>
<td>Communication failure</td>
<td>Sensor failure</td>
<td>Actuator failure</td>
<td>Miss</td>
</tr>
</tbody>
</table>

A systems-level risk management approach begins with brainstorming potential risk factors, and how they might be identified and mitigated. For autonomous soft-docking, some factors which were compiled from experience, and various resources, are listed below. [15][26][36].

**General Risk Factors and Contributors:**

- Missing data (no saved data)
- Bad/missing sensor info
- Timing delays (delayed sensor info)
- Unmodeled disturbances

36
• Avionics processing or interface issues

• Invalid models (bad assumptions)

• Incorrect parameters

• Actuator issues (control authority, timing, changing thrust)

• System changes in time

• System requirements change in time

• Test plan does not test objectives, or obscures results

• Test plan not implemented correctly

• Limitations on duration/complexity/number/types of tests

• Emergent problems with communication system, propulsion system and power system

The remainder of this section shows how modeling, diagnostic tools and experimentation can identify and mitigate such risks. Section 2.4.1 includes a walk-through of how to systematically go through factors to reduce risk.

2.3.1 System Modeling

For in depth V&V, modeling helps test the system and to identify and quantify risks and risk factors. A general schematic for modeling soft-docking is shown below in Figure 2-5. It includes both the models of the vehicles themselves, as well as the environment and supporting diagnostic tools which go with them.
Additionally, when developing the model it is important to build in the capability to arbitrarily ‘break’ components and introduce noise to sensors and actuators. While during initial design phases these may be ignored, as the design progresses one can ensure robustness by adding disturbances to the system and checking response. Section 3.1 describes generic modeling in more detail, while Section 3.2 shows a real-system application using the SPHERES facility.

2.3.2 Model Diagnostics

An equally important aspect of the model, beyond matching the physical attributes of the system, is the diagnostic portion. Development goes faster if problems are easy to spot. Spending time creating visualization tools and data processing tools which
will make problems obvious will reduce risks over the course of the design process. These diagnostics should aid in observing risk factors.

Below is a list of various diagnostic tools that will assist in development and test:

- **Visualization Tools**
  - Vehicle position and orientation
  - Docking alignment and error

- **Data Logging & Post Processing**
  - Time
  - Chaser and target state vectors (position, velocity, attitude, angular rates)
  - Control commands
  - Control error
  - Fuel consumption
  - Dock state and mechanism state
  - Vehicle separation and relative orientation

### 2.3.3 System Characterization through Experiment

Models are only as accurate as the parameters which go into them, thus experiments provide the parameters necessary for simulation. Below is a list of experiments which may be useful to gain information about the system:

- **System Characterization**
  - Mass identification
  - Inertia identification
  - Volume identification

- **Actuator Characterization**
- Timing
- Thrust profile(s)
- Thruster pointing directions
- Multiple thruster effects
- Torque-device response profile
- Fuel capacity

- Sensor Characterization
  - Sensor hand-off timing
  - Sensor(s) noise and bounds
  - Sensor output frequency

- Docking Mechanism Characterization
  - Timing
  - Acceptable contact forces
  - Acceptable position and angle offsets

Care must be taken to not only identify the mean values for various parameters, but also the distribution, uncertainty, and bounds on various values. Information gained will make the simulation more accurate, and allow simulated testing of the operational space.

2.3.4 Model Scenario Selection

Once basic system characterization is achieved through experimentation and reflected in the model, numerous simulations and experiments may be run to reduce risk. Simulations will guide where problems might occur, and where more experimentation is needed. A thorough description of using stochastic simulation to reduce risk may be found in Chapter 3.3.
Developers must balance resources, risk, and value when designing experiments. As a thought exercise, questions for experiment design are listed below:

*Experiment Design Questions*

- What tests are necessary?
- How should tests be sequenced (or parallelized)?
- How invasive or risky are tests?
- How much would the results impact the design?
- Is the test ‘exploratory’ or ‘acceptance?’
- How effective or complete is the test?
- Does the test need to happen just once, or whenever there is a change?
- What resources (time, money, personnel, equipment, etc.) are needed for the test?

Next, the key variables and parameters must be determined to map out the tradespace of potential experiments.

### 2.4 Example: SPHERES Docking

In order to illustrate the risk reduction process for spacecraft soft docking, the SPHERES facility was used (refer to Section 1.5 for background). A SPHERES project aimed at docking SPHERES satellites equipped with docking ports applied some of the processes described earlier in this chapter to reduce risk. An introduction to the risk assessment process is outlined here, while Chapter 3 and Chapter 4 offer more detail on the modeling, testing and results.
2.4.1 Risk Assessment

For SPHERES satellite docking, risks range from high level risks (danger to astronauts) to experiment-level risks (colliding during a docking attempt). For the purpose of this research, only experiment-level risks were considered, as the SPHERES facility has long been tested for safety. Additionally, component failures are considered beyond the scope of this analysis, as they also reflect a problem with the experiment facility and not the rendezvous and dock algorithms.

One benefit of the SPHERES facility is the ability to perform a whole range of uncooperative and cooperative missions. Satellites may work independently or communicate information. This allows developers to fully control the cooperativeness of the test.

In Section 2.2.2, the concept of systems theoretic risk analysis was introduced. Here, it is applied to a satellite docking application. Risk factors were determined, along with methods of observation, mitigation strategies, and control methods. For example, poor pointing was identified as a risk which could lead to failure to dock. Potential contributors to poor pointing range from poor sensing, inadequate control authority, to coding errors. To determine if pointing is poor however, there must be diagnostics in place which allow observation of this problem. Therefore visualization and graphing tools must be created that clearly demonstrate the pointing accuracy of the chaser and target vehicles. Next, if the pointing is shown to be inadequate through these diagnostic tools, mitigation approaches must be developed. One might be changing the pointing algorithm. Finally, there must be a way of implementing the changes, like uploading new code. Such cycles were thought out for a variety of factors as shown in Table 2.3.

Cycling through these steps reduces risk. Sometimes, steps are missing, and this is when the approach is most helpful. The tools and strategies in the highlighted cells in Table 2.3 were only developed as a result of gaps in this chart. For example, mitigation strategies for magnetic interference had to be thought up once it was realized there might be a problem through experimentation. Observation tools were also missing
Table 2.3: Systems theoretic risk chart for SPHERES satellite Soft-Docking

<table>
<thead>
<tr>
<th>Risk Factors</th>
<th>Observation Method</th>
<th>Mitigation Strategies</th>
<th>Control Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimator handover</td>
<td>target-lock graphic; post-operation graphing</td>
<td>modify algorithms to smooth hand-over</td>
<td>upload new code</td>
</tr>
<tr>
<td>magnetic interference</td>
<td>graph of distance between magnet and lance</td>
<td>command away from magnet; remove magnet</td>
<td>modify control commands</td>
</tr>
<tr>
<td>weak thruster</td>
<td>accelerometer data; check hardware</td>
<td>change thruster mixing matrix</td>
<td>upload new mixing matrix</td>
</tr>
<tr>
<td>poor beacon-metrology</td>
<td>poor tracking; recorded metrology signals from ultra-sound beacons</td>
<td>improve filtering; reduce environmental noise</td>
<td>new filter; turn off lights etc. that create IR noise</td>
</tr>
<tr>
<td>divergent/bad state estimation</td>
<td>recorded state comparison between global sat1 and sat2, and docking port cameras</td>
<td>modify estimator</td>
<td>upload new estimator</td>
</tr>
<tr>
<td>incorrect inertia or mass estimate</td>
<td>poor controller performance; mass ID test</td>
<td>modify stored inertia and/or mass estimate</td>
<td>upload new inertia or mass parameters</td>
</tr>
<tr>
<td>un-tuned gains</td>
<td>data recording of commanded and actual location and orientation</td>
<td>change gain</td>
<td>upload new gains</td>
</tr>
<tr>
<td>contact dynamics issues</td>
<td>simulation limits entrance velocity</td>
<td>reduce approach velocity; improve orientation</td>
<td>command slower approach and upload new code</td>
</tr>
<tr>
<td>poor pointing</td>
<td>visualization of pointing for docking</td>
<td>modify gains; change pointing strategy;</td>
<td>upload new gains and/or code</td>
</tr>
</tbody>
</table>

for easily identifying pointing problems, thus tools were developed. Another concern was how incorrect gains and inaccurate mass/inertia or thruster force information are
all coupled and it may be difficult to differentiate between them. Awareness of these weakness makes the need for tool development more obvious.

2.4.2 Verification and Validation Framework

A detailed framework for SPHERES satellite docking verification and validation is described in various sections throughout the thesis. The risk management strategy outlined in Table 2.3 serves as the foundation for later tool development and analysis. In Section 3.2 development of system models and visualization tools is discussed to allow for observation of issues. Then in Section 3.3 stochastic simulation methods are described to allow for risk assessment. Chapter 4 shows the implementation and results.

2.5 Summary

Approaches to developing a solid verification and validation framework specifically for soft-docking were described in this chapter, along with a hardware-based application example. The remainder of this thesis serves to apply both the techniques, and the identified risk factors and observation types outlined here to improve V&V for soft docking spacecraft. While all risks will not be discovered at the onset of mission design, iterating risk assessment, simulation and experiment will help in identifying and mitigating them as soon as possible.
Chapter 3

Development of Verification and Validation Tools for Soft-Dock Control Systems

In order to test a soft-dock controller, it is useful to develop a model that emulates the spacecraft in its environment. The simulation must determine the states of both the chaser and target by modeling the system dynamics, control algorithms, docking mechanism, sensors and actuators. Additionally, development of assessment tools allows one to measure and visualize performance. This chapter shows how a generic SPHERES satellite simulation was enhanced for docking, and further improved by incorporating the verification and validation framework from Chapter 2 to develop tools needed for V&V.

The process is explained as follows. First, using the framework design from the previous chapter, critical elements for simulation are described for a generic soft-dock system. Next, these elements were simulated for SPHERES docking. Augmenting the simulation, a method was created for stochastically selecting scenarios for running numerous cases. Finally, post-processing tools were developed for data assessment.
3.1 Generic System Model Development for Soft-Docking

Back in Section 2.3.1, the concept of system modeling was introduced and motivated. Now the system component models may be developed. In Figure 2-5 several elements were shown, including the dynamics control loop model, the satellite emulator, scenario generation, and V&V tools. These aspects have been modeled in prior research and on flight programs, however an overview of the process in context of risk-reduction is shown here [15][16][36][35].

When beginning the system modeling process, a decision must be made as to the software package or programming language used for the model. An advantage to high-level programming (graphical languages for example) is the ease in which a non-programmer may understand the workings of the model. Code-generation may even remove some burdens of programming. On the other hand, graphical languages tend to require more disk space and overhead than simpler languages, which may be prohibitive on a flight system. Users also have less control over the details. Care should be taken to explain both how to use a model by carefully defining the interface, as well as how the model works with its inherent assumptions. In this manner engineers can easily return to the code to make modifications later if assumptions are found invalid, or higher fidelity is needed.

Once the software is chosen, the development process may begin. When developing a soft-dock model for controller testing, the basic dynamics constitute the core of the model. Appendix A.1 walks through some relevant equations and definitions for reference. Specific to docking, the model must have the capability to transition back and forth between the un-docked and docked cases, keeping in mind the change in location of the center of mass, as well as total mass and inertia. A decision regarding the fidelity of the modeling of contact dynamics must also be made, and would be dependent on the quality of available testing data and the acceptable risk levels for the mission.

Information about the truth environment and any interactions with a ground
station must also enter the model. State information in the form of GPS data, two-
line element sets or ranging information from the ground on a contact schedule may
be included in the model. Relevant celestial objects that influence certain sensors,
like the Sun, Earth or Moon, might also be modeled.

For the spacecraft models, the objective is to emulate function, and "test like you
fly" as much as possible. This means using the actual flight GN&C if possible in the
model. To support the controller, sensor and actuator emulators which are vehicle-
specific must be developed for the chaser, and in the case of a cooperative vehicle, the
target as well. For the purpose of performance measurement, consumables (like fuel
or power) may also be tracked as necessary for a mission. Additionally, the docking
port mechanism must be emulated, along with any associated sensing or locking.

Throughout, careful attention to timing considerations and the accuracy of inter-
faces should be considered. Information arriving at the wrong time is a big risk as
it may be worse than no information at all. To test timing and function, veriﬁcation
tools should be created during the modeling process to check the validity of the sim-
ulation. Backing the simulation with inputs from experiment and matching results
also helps to validate the model.

After creating a system model that is backed by experimental data, and can be
modiﬁed to emulate 0-g dynamics, one can create a wrapper to run the simulation
numerous times while adjusting variables. This allows statistical testing for conﬁdence
building by identifying areas of risk. These types of tools may also serve to diagnose
problems found in testing and operations.

3.2 SPHERES Docking Simulation

In order to support rapid flight-worthy algorithm development, a simulation for
SPHERES was coded and validated through careful experimentation with hardware
on the ground, and data from on-orbit testing [45]. The MATLAB Simulink-based
SPHERES simulation allows multi-satellite scenario testing in both 3DoF and 6DoF,
using flight code. The simulation allows users to modify SPHERES initial conditions,
add on payloads, and run numerous successive tests to evaluate performance and predict the behavior of the satellites.

The simulation framework consists of initialization, visualization, and analysis tools, combined with an environment model containing models for three SPHERES satellites. These models include a dynamics model, a metrology beacon emulator, SPHERES flight code modules, and emulators for sensors, actuators, and any payload. Emulators need not operate the same way as in hardware, but must accurately reflect the interface, timing, noise, and disturbances the system may see in flight. Figure 3.2 shows the top-level Simulink block diagram of the simulation. From left to right, payload emulators (for the docking ports), satellite GN&C code, plant dynamics, and sensor models make up the core of the simulation. Termination, verification, and external beacon metrology emulation may be seen from top to bottom on the right side of the model.

![SPHERES Docking Simulation](image)

Figure 3-1: SPHERES master simulation

The base model was developed and validated previously by a succession of SPHERES team members over a number of years [42][36][32][46]. Modifications were necessary to enable simulated SPHERES satellite docking, however, including changes to the dynamics propagator and the addition of docking port emulators. The capability to detect docking and enable ‘joint-dynamics’ mode was a team effort on the program [45].
As a result of the dynamics modeling work, satellites may dock and undock while preserving momentum. Contact/impact forces are greatly simplified in the current simulation, in that the docking process model assumes an instantaneous and perfectly inelastic collision. Future work will involve measuring the deviation from this approximation on board the ISS and incorporating adjustment factors in the simulation.

To date, the simulation has been improved through a series of model-data correlation steps in order to match the model parameters with those of the physical satellites. Noise power levels for the docking port simulator were estimated based on processed results from experimental imagery taken on the ground at a set of measured distances and orientations. Thus noise contributions from image quality, image processing, and computation error are all bundled together. Data obtained from the first SPHERES test sessions on the ISS were used to tune the simulation to include, for example, the representative levels of metrology noise, thruster non-uniformity, sensor inaccuracies, and timing delays. The SPHERES simulation relies on these validated models to perform the maneuvers necessary to demonstrate new science, such as docking. The physical properties of the docking ports have been included in the simulation after being determined through computer modeling and physical experimentation. The measured accuracy and precision of the relative state vector computed directly from real docking port camera imagery on the ground has also been integrated into the simulation, as described in Section 4.1. Operation on-board the ISS will provide data feedback for adjusting these model parameters, as well as those for other new hardware, thus further refining and validating the simulation.

**Docking Port Emulator Development**

With the addition of hardware docking ports to SPHERES satellites, the simulation required the development of a docking port emulator. The emulator need only imitate actual hardware function (described in 1.5), rather than replicating the function itself. The data interface inputs and outputs (and timing), however, should be identical. Figure 3-2 outlines the high-level model requirements, like matching the interface, checking docking constraints, computing the relative state, detecting the
lance, and triggering the docking mechanism.

Throughout development, it is important to ensure that simulated noise levels are the same magnitude as in hardware, and timing delays are modeled, to make the model as realistic as possible without making it too complex. Overly complex simulations often take a long time to run and are also difficult to develop and maintain. In this case, the trade was made such that the Kalman filtering and image processing algorithms were not replicated — only their output was matched. Creating the imagery to allow such algorithms to run would have been very time-consuming, and potentially inaccurate given that during initial development there was no actual imagery from space.

Once the functional requirements were outlined, a basic docking port emulator was designed and later refined. MATLAB Simulink was chosen for the design, as it was compatible with the existing SPHERES simulation. The SPHERES docking ports were emulated by payload modules which provide a realistic, noisy estimate of the relative state vector to any sphere within the computed target-lock angle and range. Essentially the simulated ‘truth’ states of the satellites were used to compute the relative states and check for satisfaction of soft-dock conditions. In hardware, ‘target lock’ only happens when the docking port camera can focus properly on the fiducial markings located on the docking port of the target, thus the model reflects this requirement. Once target lock is achieved, the relative state vector information can then be used by the SPHERES flight code for control. When two satellites are close enough to dock, and if they are oriented such that the lances align with the holes and then approach at low velocity, a timer trips on the virtual docking mechanism,
allowing the satellites to dock. The dynamics model then treats the satellites jointly as one rigid body, until an undock command is received from each of the SPHERES.

The Simulink model shown in Figure 3-3 shows how the truth states, polling and undock commands are used to create a ‘state of health’ output for the docking port model. Inputs and outputs are highlighted in yellow, while key computations for the relative state output, photo-sensor and mechanism status occur inside the blue blocks.

Modules within the model serve to add appropriate levels of noise to the signal in lieu of adding a true filter. The text notes that were built into the model are not shown in the figure, however they help programmers understand the reasoning and assumptions inherent to the model.

![Figure 3-3: Simulink schematic for docking port emulator](image)

As the emulator was developed, it was tested as a unit in an independent testing simulation. Once it passed validity requirements, it was integrated into the master simulation and tested as a system for correctness. On integration, it was soon realized that better visualization tools were needed to verify functionality. The capability of visualizing docking and un-docking, as well as docking port alignment during ap-
proach, was not yet developed. Once the importance of such tools was perceived, they were designed. Figure 3-4 shows a snapshot of the SPHERES animation during a docking maneuver, used by researchers to visualize the expected satellite behavior prior to hardware testing. One can see the avionics stack and docking ports that were added to the pre-existing simulation-time visualization tool.

![Figure 3-4: Satellite visualization for docking](image)

Another simulation run-time feature was added, which was the docking port alignment tracing window (see Figure 3-5). The window animates the projected position of the chaser lance and hole onto the target vehicle’s docking port. It also provides information on the relative separation and offset angle dynamically throughout a simulation. Data post processing provides additional detail for analysis.

![Figure 3-5: Visualization for docking port alignment](image)

All of these verification and validation tools had to themselves be validated, which required additional simulating. Additionally, certain assumptions are inherent to the
model, as time restrictions prevent one from modeling every last detail.

Assumptions and Limitations

As with any model, simplifications were necessary to create a working docking simulation. For the SPHERES docking model in particular assumptions include:

- No external forces and torques (i.e. no accounting for magnetism or airflow including plume impingement)
- No contact dynamics
- No emulated imagery (UKF estimation "approximated" in code)
- Limited fidelity in docking port insertion model
- Instantaneous completely-inelastic docking

Since simplifications may become important later, these were documented (in code and written files) to allow team members to re-evaluate later in the design. It may turn out that an assumption is overly simplistic, and may need to be added into the simulation at a later point. For example, magnetic effects were not initially modeled, however once the satellites docked magnetically during checkout (see 4.2), additional modeling and post-processing tools were added to examine the validity of runs given major inaccuracies within a 2cm radius of the magnets.

The reasoning behind some of these simplifications was both difficulty in implementation and limited information. For contact dynamics, it was decided to begin with a simple model and progress with a higher fidelity version as information was gained in flight.

In order to further reduce risk, a method must be developed to robustly test the system within the confines of various assumptions. By doing so, variables which the system is most sensitive to, or have high uncertainty, will reveal themselves. Identifying them, and then classifying the level of control the user has over them (or their certainty) is a good initial step towards reducing risk.
3.3 Stochastic Simulation Test Framework for SPHERES

For determining the variables with the most impact, running Monte Carlo analyses can provide insight. A GUI was created allowing human input of both variable selection, and probability distribution. After initial conditions and variables were set for each run, these simulations could be run in parallel on multiple machines (even stopped in the middle) and the data stored. Post-processing scripts were developed (with human input) to assess correlation between variables and success or performance. Knowledge gained from post-processing was used for determining what future tests (or simulations) to run, and to make code modifications. Figure 3-6 shows the cycle of how flight code is run based on statistics chosen by the user in the Monte Carlo GUI, simulations are run, data is post-processed, and results then drive modifications.

Figure 3-6: Simulation Framework Information Flow
Ideally, lessons learned from simulation feedback into code improvements, thus reducing risk and making ISS testing more valuable. By ensuring that the risk factors identified in 2.4 are included as variables, one can test how they affect the system. Allowing users access to a number of variables, many not already identified as critical, will also let the simulation find potential problems that were not expected. The biggest risks are often unknown unknowns, and incorrect "knowns." By running simulations that sample the trade space beyond what is expected, potential risks may be identified.

User Input to Simulation

A user interface (shown in Figure 3-7) was developed to allow one to select the probability distributions of variables important for soft-docking. The interface was designed for both flexibility in choosing relevant distributions, and to help users visualize the outcome of their selections. Invalid selections are highlighted with warnings immediately to aide users. Plots are generated (see Figure 3-8) to show what was generated by the user inputs before the simulations are run.

Figure 3-7: Inputs of graphic user interface for input variable selection and distribution
By allowing the user instant feedback as to the validity and distribution of input selections, the user saves time by avoiding running unintended datasets. Transparency is critical, since users depend on the tool to behave as intended to trust results. For example, during the design of the tool there were a number of glitches in the programming of the distributions that only became apparent once graphing was implemented.

**Simulation Variable Selection**

Various configuration options were selected as variables available for tuning using the Monte Carlo test setup. Essentially, variables which might change from test to test (like initial conditions) were first choices, and then parameters which might have a great influence on the system (like sensor noise) were also selected. Table 3.1 shows what the tuneable variables are, associated options, and an explanation of why the variable is important. Additionally, if "out-of-bounds" inputs are created,
error-checking immediately informs the users of the incorrect entry. For example, if the user enters a negative separation distance, it will be rejected.

Table 3.1: Monte Carlo input variables

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Options</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position, Velocity, Quaternion, and Rate</td>
<td>Constant, Uniform, Normal</td>
<td>Select probability distribution of initial state for each satellite</td>
</tr>
<tr>
<td>Distributions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Condition</td>
<td>Constant, Min/Max or Mean/Std</td>
<td>Select constant, range or Gaussian parameters for selected distribution</td>
</tr>
<tr>
<td>Dev.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Configuration</td>
<td>Test number, success codes,</td>
<td>Pick what codes are considered successful for post-processing; fixed</td>
</tr>
<tr>
<td></td>
<td>random seeds option</td>
<td>seeds are for reproducing results</td>
</tr>
<tr>
<td>Min Separation</td>
<td>Positive number</td>
<td>Select minimum separation between satellites for error checking to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ensure initial conditions do not make them start inside each other</td>
</tr>
<tr>
<td>UDP Noise</td>
<td>Check/uncheck</td>
<td>Toggle on and off the noise on the virtual docking port relative</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state output for testing the ideal case</td>
</tr>
<tr>
<td>Start Docked</td>
<td>Check/uncheck</td>
<td>Some tests must start docked</td>
</tr>
<tr>
<td>Configuration Parameters</td>
<td>UDP, dynamics, and metrology;</td>
<td>Allows designation of noise characteristics for individual</td>
</tr>
<tr>
<td></td>
<td>name, value, std. dev. min,</td>
<td>satellites</td>
</tr>
<tr>
<td></td>
<td>max and select satellites</td>
<td></td>
</tr>
</tbody>
</table>

Looking back at the risk assessment from Chapter 2, it is important to make sure that the various risk factors identified are included as variables for simulation testing.

Another important feature is the number of tests generated. In order to guide the user in the selection, the Chebyshev inequality is used to bound the uncertainty on the mean pass-rate. A bound on the number of required runs \( n \) to estimate the mean pass rate within an error \( \varepsilon \) to a confidence level of probability \( 1 - P(|B/n - p| \geq \varepsilon) \) may be computed using equation (3.1). The derivation is shown in Appendix A.3.
\[ n \geq \frac{1}{4\varepsilon^2(1 - P(|B/n - p| \geq \varepsilon))} \]  \hspace{1cm} (3.1)

If searching for criteria other than pass-rate, the equation changes and the minimum number of tests would change, therefore it is important to think about what should be gained from a test when determining the sample size.

**Processing and Data Analytics**

Since stochastic modeling involves many runs, computation speed becomes very important. Scripts were made to run simulations in parallel on multiple cores of a machine to reduce computation time. Additional requirements, beyond operating simultaneously on different cores or machines, was the need to save all data, and the need to allow premature termination without any loss. This allows users flexibility to stop and restart runs somewhere in the middle, and also to observe individual tests of interest.

In order to assess performance, a set of post-processing scripts were created. These were designed for individual simulation datasets or aggregate sets. As concerns arise, developers should make new test cases to add to the post-processing set to check for problems. It was important to make sure the post-processing framework allowed for expansion, as new scripts are added.

The purpose of the scripts is to compile and display data quickly for the user to examine. Results should be easy to interpret, automatically aggregated, and still allow a user to delve into a particular case for further examination. For example, the histogram in Figure 3-9 shows docking pass/fail (in color) and increasing initial separation distance along the x-axis. A developer, seeing a correlation between docking failure and large separation distances might wish to look at a particular failed run to determine if there is causation and figure out why large separation is a problem. The post-processing tool should allow the user to locate and process such a run.
A growing suite of post-processing scripts were developed for simulation testing of soft-docking with SPHERES satellites. The goal was to look at a range of input variables and provide some feedback as to what was successful, and what variables are particularly critical.

In order to meet this goal, one must understand the types of outputs of interest. There exist two classifications of success metrics — discrete metrics, and continuous metrics. Discrete metrics are success measures such as pass/fail from docking success or failure. Continuous metrics could be performance measures such as fuel usage during a maneuver. When designing post-processing scripts, it is important to differentiate between these types of variables when plotting results.

For the SPHERES satellite case, the initial list of post-processing examples included various histograms and plots to show performance with respect to various parameter values. In Section 2.4.1, a list of risk factors was created to ensure observation later in design. This list was used as a baseline to select what graphs were necessary. A sampling of such risks, relevant variables, post-processing output, and their corresponding objectives is shown in 3.2.

When designing V&V tools for any soft-dock system, ensuring that tools support observation of risk factors will help with overall risk reduction.
Table 3.2: Post processing scripts

<table>
<thead>
<tr>
<th>Risk Factor</th>
<th>Parameters</th>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor metrology</td>
<td>Ultrasound dropout % and noise level</td>
<td>Histograms of count vs. dropout % or noise level also showing pass/fail in color</td>
<td>Show metrology affect on success</td>
</tr>
<tr>
<td>Weak thruster</td>
<td>Thruster force scaling vector</td>
<td>Histogram of pass/fail (in color) vs. scaling factor or number of bad thrusters</td>
<td>Show thruster performance affect on success</td>
</tr>
<tr>
<td>Magnetic Interference</td>
<td>‘Truth’ distance between lance and magnet</td>
<td>Histogram of minimum distance during test (before docking), and overall count of tests meeting distance threshold while also color plotting pass/fail</td>
<td>Compute percent of potentially successful runs</td>
</tr>
<tr>
<td>Uncertain initial position</td>
<td>Starting distance between satellites</td>
<td>Histogram with pass/fail in color and starting separation distance on x-axis</td>
<td>Show distance affect on success</td>
</tr>
<tr>
<td>Non-zero initial rates</td>
<td>Initial angular rates</td>
<td>Histogram with pass/fail and increasing angular rates</td>
<td>Show if rates affect success</td>
</tr>
<tr>
<td>Target-lock difficulties</td>
<td>Starting time and duration of first target lock</td>
<td>Histogram of times (and pass/fail), plot of trajectory on approach</td>
<td>Show any affect of target-lock or lock-loss on success</td>
</tr>
</tbody>
</table>

3.4 Conclusion

Graphical enhancements of simulations have been shown to increase engineer awareness of risk factors. In this chapter, simulations were developed as a method for testing soft-docking and thus reducing risk, since they allow emulation the 0-g space environment. The key elements necessary for a generic simulation were outlined, followed by an example of how the SPHERES docking simulation was developed. Details on the development of a docking port emulator were shown, along with visualization tools. As a result, during SPHERES docking scenario test runs, the data visualiza-
tion tools now provide quick feedback to users, showing position, orientation, and relative docking angle. In order to gain even more information from the simulation, statistical methods were recommended to find system sensitivity to various sources of uncertainty. A stochastic framework was developed for the SPHERES docking simulation to show how variables can be chosen to sample the relevant soft-docking space. The next chapter will cover results from running such simulations and compare them with hardware testing results.

Essentially, a detailed system model for docking facilitates code development as problems may be found earlier in the design. The process described in this chapter of considering the basic requirements that must be modeled for a docking system, and then applying them to a specific system taking care to model only the most important aspects in the most detail would work for any system. Additionally, adding an stochastic layer allows developers to locate potential difficulties before operation, and can guide testing and redesign. Not only that, but developing these V&V tools early will help in future diagnostic work, since they may be used to try and imitate problems.

Further work for the SPHERES soft-docking V&V tool-set might include:

- Adding three-satellite capability to the model
- Expanding the capability of the Monte Carlo setup by adding more variables
- Creating a scenario-set ‘loader’ to visualize the commanded Monte Carlo test set again from a saved file
- Improving fidelity of the docking-port emulator by making docking success criteria more realistically stringent
- Developing a better post-processing script to categorize primary reasons for failure for mass-simulations
- Code additional statistical analysis post-processing scripts to allow simultaneous comparison of parameters and compare sensitivity
The most important thing to do, however, with soft-docking V&V tools is to apply them to real problems. The next section shows how the framework developed in this chapter was itself validated and improved through iterative ground and ISS testing using SPHERES satellites, as well as the risk-reducing benefits of these simulations.
Chapter 4

Soft-Dock System Analysis, Verification and Validation

The SPHERES facility on the ground, in simulation, and in space provide a test case for system analysis, verification and validation. To begin, data from ground tests was used to improve the simulation. There are aspects of the free-fall environment, however, that are difficult to test on the ground, thus these elements were checked with ISS tests. Once some trust was gained in the simulation, it could be used to identify areas of concern for soft-docking. A thorough simulation analysis was performed to identify key parameters affecting soft-docking. An attempt was made to validate the success of this testing via another ISS science test, however setup issues on the ISS prevented a successful test for true comparison. The simulation was then used as a diagnostic tool to help identify causes of issues seen during ISS testing. Finally, useful information regarding modeling of the docking port was gained, allowing future improvements.

The flow of information between experimental test and simulation is of particular importance for validation. Ground testing supplies parameters to 1-g simulation, which is extrapolated to 0-g simulation, driving development of ISS tests. Feedback from ISS tests further improves simulation, and simulation may also be used to diagnose issues in 0-g. The process of feeding back information from flight testing and simulation is not new, having been done on numerous programs, and suggested
in textbooks as a methods for V&V[15]. It is also described by developers on the MACE program [18], [10]. However, utilizing this framework for ISS experimental autonomous soft-docking testing is recent work.

The schematic in Figure 4-1 shows how testing flowed in this particular docking example using the SPHERES facility on the ground and in space. The lettered steps correspond to processes that will be described in this chapter. First, ground testing was used to correct a 1-g simulation (A). The simulation was designed such that gravity could be ‘removed’ and thus results could be obtained for a 0-g scenario. The resulting flight code was then sent to the ISS for checkout tests (B), and resulting information was fed back into the simulation. Before and during the development of flight code revision 2 (C), much analysis went into improving the algorithms by running Monte Carlo simulations to test out various operational conditions. Gravity was reintroduced in simulation for comparison with more ground tests, and then turned off for more simulations. Another ISS test was carried out (D) and results from simulation were used to diagnose issues found during the test (E).

![Figure 4-1: Hardware testing and simulation in 1-g and 0-g](image)

For each flight code revision, a set of tests was developed by the team. Tests ranged from identifying the mass of the satellite to docking using global metrology.
(ultrasound sensing) and any filtered relative-state data from the docking port. Extra backup tests were created in anticipation of potential problems aboard the ISS. For example, if the docking port camera hardware failed, or global metrology from ultrasound sensors had issues, those tests would be run. Three test options were created, two backups using each source of state information independently, and primary test using the combination. Such contingencies were recommended by Hilton in his thesis on experimentation using the ISS [21].

The remainder of this chapter shows both the process and results by which experiment and simulation were used in creating a validation system. Lessons learned from development for SPHERES satellite docking can be generalized to other soft-docking missions as well, since they have the same fundamental elements.

4.1 Ground Testing

On the ground, test satellites and a "flat table" setup allow 3 degree of freedom testing for soft-docking. A photo of the flat-table test setup is shown in Section 1-2. The purpose of ground testing was to (1) obtain baseline parameter calibration information, (2) match simulation to experiment (1-g case), and (3) find additional error sources. Section 2.3.3 describes how ground testing provided the parameters necessary for modeling. The main types of ground tests conducted are described in Table 4.1, and are explained in more detail in this section.

4.1.1 SPHERES Functional Comparison with Simulation

Since the general spacecraft parameters have been validated against the ground test setup and ISS tests on previous projects, the objective was to test parameters specific to the addition of docking ports to the satellites.

At the sub-system level, it is important to match the results of the docking port algorithms in experimental prototypes and in simulation. In hardware, the docking port camera takes a picture that is image-processed to determine a relative state. That information is passed through the unscented Kalman filter (UKF), transformed
Table 4.1: Ground testing

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Determine mean values for transforms, critical dimensions, noise levels etc.</td>
</tr>
<tr>
<td>Function</td>
<td>Check if UKF output matches directions and signs of simulation</td>
</tr>
<tr>
<td>Noise</td>
<td>Match simulation noise levels of relative state to hardware UKF output</td>
</tr>
<tr>
<td>System</td>
<td>Run code intended for ISS (with 1-g modifications) to compare hardware and simulation in 1-g to see if similar performance and end results</td>
</tr>
</tbody>
</table>

into the coordinate frame of the satellite (rather than the docking port), and then sent to the controller.

In the simulation, by contrast, the ‘truth’ relative-state is computed between the satellites, noise added, and then the result is sent to the flight code in lieu of Kalman filter information, as described in Section 3.2.

Since the docking port model is heavily simplified in simulation, it is important to verify its predictions match hardware. An experiment was performed to record data when moving the satellites relative to each other to ensure proper signs and values of attitude quaternions, rates, position and velocity. This process was done twice, once for the original UKF, and then again for the UKF developed later (after the ISS checkout test). Figure 4-2 shows sample results from a ground test of the UKF, that is representative of both tests. Graphs in the first column represent the UKF resultant state vector (transformed into satellite body-coordinates) of the target. Graphs in the second column show the transformed and raw position and quaternion directly after image processing. The photograph in Figure 4-2 shows the view from the docking port camera of the target. The target was rotated in 90 degree steps (near time 30s, 45s, and 60s) approximately around the camera boresight vector, and then tilted in each direction. The filtered attitude quaternion shown in the third row of the first column changes accordingly. Another test was conducted to move the target towards and away from the chaser, and also across the frame.
Figure 4-2: Still frame of video plotting of docking port UKF state data (column 1: position, velocity, quaternion, angular rates) and raw data (column 2: position (transformed to center of mass), raw position (rel. to camera), transformed quaternion, and raw quaternion)

Figure 4-3: Still frame of video plotting of docking port camera imagery, synchronized with plots in Figure 4-2
After plotting experimental data, the simulation was run with the same orientations to check the values of the quaternions. All directions matched, though noise was much lower in simulation. Noise levels had to be adjusted in simulation to account for filtering troubles. The team also worked to improve filter performance using gyroscope data, an upgrade which took place before the second iteration of flight code.

4.1.2 Emergent System Properties

A concerning discovery from ground testing was the tendency for the satellites to dock magnetically when pointing control was poor. Next to the docking port camera are two built-in magnets (see Figure 4-4) to hold a lens cover in place for safe stowage. These would magnetize the lance on the opposite satellite, causing undesired docking.

Figure 4-4: Locations of camera magnets

There was concern that magnetic interference would prevent the satellites from docking properly in space, however removing the magnets before launch would cause delays. Further testing was conducted to measure the sphere of influence of the magnets to see if they would cause a problem, or if the satellites would only dock if control was already insufficient. The amount of force the magnetic connection could handle before breaking was measured (ten times each) at three different distances. Approximating the magnet depth under the surface (1.55mm), and averaging the experimental data allowed the modeling of force vs. distance. A range of exponential factors were used as the factor depends on the magnetic behavior — whether the lance behaves more like a magnetized surface or a magnet. Figure 4-5 shows the
experimental data and modeled magnetic influence on a SPHERES satellite converted
to accelerations assuming a satellite mass of 5.98kg.

Figure 4-5: Experiment and modeled data supporting essential keepout zone > 6mm
(or 4.5mm from the surface) for one magnet

The logarithmic plot shows how acceleration drops off quickly with distance. The
key takeaway here is that satellite thrusters must be able to overcome the magnetic
acceleration given their max acceleration and duty cycle. Since they may only pulse
once per second, for 20% of the time, once the lance is too close the thrusters may not
have time to react. Given acceleration on both satellites, an approximate distance
of 4.5mm from the surface to reach one second to max thruster capability can be
estimated. Since there are two magnets, a keep-out zone of about 1cm from the
magnets should be sufficient.

Given the offset distance (approximately 2.8cm) between the magnets and the
hole, and ground testing of the magnet strength, the problem was considered low risk,
though not insignificant. Contingencies were developed in case it became a problem.
While magnetic forces were not modeled in the simulation, distance between the lance
tips and the magnets could be computed as a metric relating to magnetic influence.
Since the electromagnet force between two magnets (or a magnet and a magnetized
element) scales exponentially with distance, conclusions about mission success could
be drawn from this metric.
Based on this analysis, it was determined that magnetic docking only occurred when SPHERES already failed intended docking, and did not pose a large risk for science. Therefore this issue would not interfere with controller validation in successful runs. Other concerns, like damage to the camera lens itself was not a risk, since the magnets are not exactly behind the lens. Finally, flight version docking port units were near-shipment and the team decided it would have been prohibitive to remove the magnets.

4.1.3 Simulation Parameter Calibration

Some of the most significant contributions of ground testing were the parameter variances measured in testing. The following values were measured on the ground and used in simulation:

- Dimension data (measured and CAD)
- ‘Validation’ of relative state output directions
- UKF noise parameters
- Timing for opening/closing mechanism
- Timing for image processing

4.1.4 Flight Code Improvements

Ground testing revealed that the image processing algorithm mixes up the fiducial points with the gears on the pinch point label, the ultrasound beacons, and the IR receiver holes. Covering the label (shown in Figure 4-6), and the algorithm tuned slightly, it would reduce the occurrence of mix-ups.
Ground testing also revealed issues with the attitude pointing algorithm. The chaser seemed to sometimes point slightly to the side of the target vehicle, thus the pointing code was revisited and revamped. Further modifications were made to the pointing code after checkout (due to time constraints) and once better diagnostic tools were created to help visualize the problem.

4.2 ISS Checkout

During the ISS checkout test, the team verified whether hardware was still functional after arriving at the ISS, whether the electronics interfaced correctly with the new camera and docking port, and whether basic flight algorithms functioned properly. Several attempts were made to autonomously dock. In these runs, the target was commanded to remain stationary while communicating state information to the approaching chaser. Unfortunately, these docking attempts were ultimately unsuccessful, but results from the checkout were used to further train and improve the 0-g simulation.

4.2.1 Anomalies and Discrepancies with Simulation

The checkout successfully demonstrated docking port hardware functionality. Some other observations were made during the test which merited further analysis and influenced development efforts. For instance, performance of the relative-state esti-
mator was questionable since it did not match what was computed based on global metrology. There were large biases in position, and the computed attitude was also offset. Additionally, on one of the satellites, there were high ultrasound pulse drop-outs and the data was extremely noisy (supporting figures may be found in Appendix B-4). This led to doubts about the accuracy of the state information for that satellite. Finally, during one of the tests the satellites magnetically docked, unintentionally.

Figure 4-7 shows information recorded by the IMU during an ISS test that resulted in unintentional docking due to unintended magnetic attraction between the camera lens cover and the docking port lance. The acceleration due to the influence of magnets can be observed from the gradual increase in acceleration in the x direction, and the final exponential increase in acceleration in the z-direction just before collision. Note that if the magnets had not been present, acceleration in all three directions would have stayed close to zero whenever thrusters were not firing.

![Figure 4-7: Target SPHERES bias-corrected raw IMU accelerations before collision during Test 4](image)

Interestingly, in a separate test the satellites are able to compensate for, and even overcome the influence of the magnets, as shown in Figure 4-8. Unfortunately, this other test is ultimately unsuccessful as well due to other factors commanding the
Chaser lance directly towards the magnets instead of the hole. While in the first test the chaser collided with the target directly due to the effect of the magnets, in this other test the chaser approaches, adjusts, but eventually collides with the target before ultimately attaching to the magnets.

Figure 4-8: Chaser SPHERES bias-corrected raw IMU accelerations before collision during Test 6a

Around time 172sec, the satellites were close enough for magnetically-induced accelerations to be easily seen in accelerometer data, however the satellites maintained control authority. This gives the impression that failure to dock was not due to magnetic interference but another problem.

4.2.2 Simulation Validation

The data captured during the in-space tests can be used to make direct one-to-one comparisons between simulation model prediction and experimental measurements and thereby increase the fidelity of the simulation where needed. The simulation can be set to run with the same initial position, velocity and rates seen during the ISS test.
Figure 4-8 shows plots for one of the test runs showing actual 0-g results and simulated 0-g results. In particular, it shows the ISS position track along with the simulated track for a test of docking using only ultrasound-based state information. The simulated track does not appear to start with the exact same initial conditions since the first 18 seconds are spent initializing the state estimator, so the initial conditions had to be back-propagated. After initialization and reaching some stability, the simulated track looks similar to the actual track, however with some key distinctions. First, the commanded track for simulation does not change near 45 seconds. Second, the simulated track is much smoother than the actual track. Third, the y-position on the ISS test remains with an offset. Finally, perhaps most importantly, the simulated track passes the last gate shortly after 120 seconds and proceeds to dock, while the ISS test fails.

Figure 4-9: Chaser position vs. time for simulated ‘truth,’ simulated ‘actual,’ ISS state estimate, and ISS commanded state
Analyzing the data, sources of problems ranging from coding errors which caused incorrect position commands when initial conditions deviated far from nominal were identified, as seen in Figure 4-9 at a time of 45 seconds.

Other tests were used to validate the simulation as well. After refueling, a test was run using a combination of ultrasound-based state estimate (i.e. global metrology) and relative state estimates from the camera. Results from this test are included in Appendix B.1 for reference.

In general, simulation relative-state estimates, were much more accurate than those on orbit. Simulations also produced more cleaner results than on the ISS. On the ISS test, the chaser satellite passed through some initial gates and even achieved target lock (though it was not supposed to use the relative information for attitude control), but deviated on approach to far for docking. In simulation, the deviations were small enough to still allow docking. Additionally, further research was needed to account for discrepancies observed between the relative-state estimates and the global metrology.
4.2.3 Simulation Model Improvements

In general, the simulation model proved to be too idealistic, allowing docking when docking did not occur. Numerous improvements were made to the simulation to make it more realistic. Better diagnostic tools were added as well, to aid in observation. New test features were added, such as the capability to easily initialize a scenario into the docked state, and reset the docking mechanism after undocking. The following list highlights various key improvements:

- Graphics improvements (shown in Figure 3-5)
- Approach visualization (shown in Figure 3-4)
- Geometric-center and center of mass confusion errors
- Relative-state estimate noise and inaccuracy modeling
- More stringent/realistic docking entrance-criteria
- Post-processing for magnetic interference
- Stochastic simulation framework development (described in Section 3.3)

Despite consideration, magnetic effects were not added to the simulation dynamics model to account for the issues encountered with the mechanical design of the SPHERES, as discussed earlier. Instead, automation scripts that execute outside of the simulation were developed to estimate the expected distance between the magnetic parts of the SPHERES and to raise warning flags if a threshold is surpassed.

4.2.4 Flight Code Improvements

When the SPHERES were docked, the state estimator (which uses a one-SPHERE model) diverged after attempting motion, as seen in Figure 4-11 by the difference between the actual separation and the estimator-computed distance. Thus during future docked tests, the estimator must be modified to either include a docked model dynamics or put more weight on sensor measurements than predictive state.
Figure 4-11: The estimator-computed distance between satellites diverges to an impossible 3.5 meters, even though the SPHERES satellites are rigidly docked throughout the test.

Latency was also of concern, since any delays in providing state information to the controller results in a phase lag. By modifying the code, the team reduced the time delay between computing the relative state estimate and sending it to the controller by allowing the information to stream directly after computation. The streaming modification was also applied to the simulation, as it had to interface correctly with the flight code.

Major modifications were made to the system of gate checks and pointing times for the docking scenario. Dependence on initial positioning was eliminated by forcing the chaser to remain inside a virtual cylinder for approaching the target when moving forward. Care was also given to ensure the pointing algorithm would not cause the satellites to oscillate when pointing towards each other.

4.3 Risk Assessment Enabled by Monte Carlo Simulation

Using the data gathered through the experimental docking attempts aboard the ISS, many modifications were made to both the validation simulation and autonomous GN&C code. Although the satellites were unable to successfully dock, the data and insights gathered through testing allowed the team to iterate on making modifications.
to the simulation and code. Additionally, better diagnostic tools were developed to help with assessment. Between the checkout test in September, and the first science test in March, the capability to run Monte Carlo analyses was created (as described in 3.3). The purpose was to build a risk assessment tool and apply it to soft-docking to assess system performance, and identify areas that require modifications or additional testing in hardware. This section discusses a subset of the various scenarios that were run to determine system sensitivity to certain variables using the framework designed in Section 3.3. Results of these analyses and the team’s response are described.

4.3.1 Test Design: Selecting Variables and Parameters

Benefits of simulation include the capability of performing numerous runs in parallel, inexpensively, while emulating dynamics which are difficult to simulate on the ground. A major driver of creating the simulation and Monte Carlo testing framework was to allow testing of numerous cases to get statistical confidence in predictions. Experimental design concepts of randomization, isolation, and repeatability were used in the development of these simulation tests, as recommended in Hilton’s thesis [21]. It is important to determine not only overall confidence, but sensitivity to different variables by holding the others constant and focusing variability on a particular aspect of the problem.

To begin, tests were run varying numerous parameters simultaneously with expected distributions and then graphed against various metrics to get a sense for which parameters were most influential. Then these parameters were studied in more detail through additional simulation. The purpose of the study is to identify risks. For soft-docking, risks may stem both from unexpected failures and also system uncertainty. ‘Failures’ could be problems such as low fuel or communication issues. In this analysis, focus was given to reducing risk due to uncertainty in elements which influence control, rather than component failures. Uncertainty might be found in initial conditions, dynamics, sensing, and actuation. Each area was examined through simulations.

In Table 3.1, variables accessible for tuning were described. Now ranges and dis-
tributions for a subset of these variables were selected to run numerous scenarios. Deciding which variables were held constant, and to what value, was also important. For each area of uncertainty (initial state, dynamics, sensing and actuating), simulation sets were designed, run, and analyzed to assess risks and develop mitigation strategies as necessary. Table 4.2 shows areas of uncertainty, variables, and logic for testing them. These variables are a subset of those made accessible by the stochastic simulation tools developed in Section 3.3, and relate to the risks from Table 2.3. The remained of this chapter discusses these elements in more detail.

Table 4.2: Stochastic simulation testing for risk assessment

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial State</td>
<td>Position, velocity, attitude, angular rate</td>
<td>Uniform distributions to survey operation-space; normal distributions for performance tests</td>
</tr>
<tr>
<td>Dynamics</td>
<td>Acceptable incoming dock velocity and region; mass; inertia</td>
<td>Test success when contact dynamics are restricted; see performance when mass or inertia change</td>
</tr>
<tr>
<td>State Estimation</td>
<td>Global metrology noise; beacon drop-rate; estimator noise; UKF noise</td>
<td>Test sensitivity of system to different noise sources by simulating both a uniform distribution over a reasonable space, and a normal over the expected performance</td>
</tr>
<tr>
<td>Actuation</td>
<td>Thruster force; thruster direction matrix; thruster noise level</td>
<td>Vary noise level on thrusters, as well as force to see performance sensitivity</td>
</tr>
</tbody>
</table>

4.3.2 Initial State Conditions

A strong correlation was found between docking success and satellite initial states. Different types of tests were also more sensitive to different elements of initial state. For instance, starting satellites too far apart prevented the satellites from docking.
This was because the satellites could not reach the starting locations in time to make the gate, thus ended up approaching at an angle. The histogram in Figure 4-12 shows results of a Monte Carlo run varying initial position and keeping all other variables in nominal ranges.

Figure 4-12: Success histogram for varying SPHERES initial separation distance before GN&C code modifications with only a 56% pass rate for a wide distribution of starting positions.

A modification to the flight code, based on this analysis, was to program the chaser to remain within a cylindrical track towards the target before moving closer, rather than reach specific way-points. This reduced dependence on starting position. Initial position is much less of a driving factor in docking success for the modified flight code when using both metrology-based state information in combination with relative state information from the docking port camera.

During tests which navigated using only docking port camera data and not global metrology, issues were encountered however. One expects this scenario to fail if the fiducial markings start outside the field of view of the camera, thus for this test the astronaut would be instructed to position the satellites such that the markings are in the field of view. Starting locations in simulation were assumed to be within a tight range for the satellites to point at each other. However, they failed to dock when
initial spin rates and velocities were too high, causing a loss of target lock before the system could stabilize. Figure 4-13 shows percent docking success across increasing initial angular rates.

![Histogram from uniform distribution of initial angular rates around Z-axis (tank-axis) showing docking success](image)

Figure 4-13: Histogram from uniform distribution of initial angular rates around Z-axis (tank-axis) showing docking success

Essentially, when spin rates are too high, the estimator does not have sufficient time to converge on a solution and control the vehicle before the fiducial markings leave the field of view. The chart is not exactly symmetric either, because of the spacecraft inertia, which would cause the vehicle to tip in a direction even with a purely-axial initial spin. Spinning about the boresight, however, resulted in almost a 100% docking success rate (given other variables were kept constant).

As a result of initial state condition simulations, code was modified to reduce dependence on starting location and orientation. For the camera-data-only backup test, 0.05rad/s (or close to 3 °/s) was thought to be a high tipoff rate for the astronaut to impart accidentally. Since it was a backup test, despite concerns the risk was accepted. The team resolved to inform the astronauts to be "extra careful" during satellite release and perform the test multiple times if there were issues.
4.3.3 Unmodeled Dynamics

Uncertain dynamic properties may be difficult for a system to handle, given the lack of modeling. In Section 4.2.4, state estimator divergence was shown to occur when the satellites were docked due to insufficient modeling of the dynamics.

As mentioned in Section 3.2, contact dynamics were highly simplified in this model, and due to time limitations were not studied in this analysis. Recent efforts have begun however to use the existing tools to determine contact velocities and location. In the future, including a method of allowing a random distribution of forces to emulate the uncertainty of the exact contact point during docking might help with identifying potential difficulties. Additionally, on-orbit test data will help with providing insight as to what contact dynamics might look like from the perspective of the vehicle.

Another concern was the potential for plume impingement from the opposite satellite interfering with control. Looking at the accelerometer data, however, there were no observable accelerations consistent with thrusting from another satellite.

Finally, during ground testing, concern was raised over magnetic docking (as described in Section 4.1.1). The satellites also magnetically docked during the first docking test on the ISS. Magnetic interference was not modeled in the simulation, or accounted for in flight GN&C code.

Computations from Section 4.1.1 showed that a clearance of approximately 1cm around each magnet should be sufficient to prevent unintended docking. Therefore, simulations were run with processing scripts to quantify how often approaches were too close. Figure 4-14 shows the range of close-approaches for a set of successful runs, and the percentage that had each approach distance. Close-approaches greater than 4cm were not successful, because the lance would not have entered the hole.
Essentially magnetic interference could still cause a problem given the current soft-docking architecture and code, especially when relative state estimates are noisy like in this example. Here, approximately 7% of the ‘successful’ runs would have docked magnetically instead. Looking instead at a sample of successful runs and what the separation actually is, one can see potential issues with oscillations, high contact velocities and angled approach. Contact dynamics will require further investigation.

Of the un-modeled dynamics described, magnetic effects and contact dynamics were considered the most risky. Given the modeling showed less than 7% potential for trouble with a highly diverse dataset, and those runs appeared to have worse incoming contact angles, it seemed that runs that would due to magnetism would have also failed for bad incoming angle. Thus risks for magnetic effects were considered the same as those for contact dynamics, and thus accepted.

4.3.4 State Estimate Accuracy and Noise

Just like unknown dynamics, state uncertainty can lead to control issues. For docking spacecraft, sufficient knowledge of the relative state between the vehicles is important to null out rates, point accurately, and approach with a slow velocity (for soft-docking,
rather than a collision). This begs the question of how accurate and precise the relative state knowledge must be for successful docking. Also, what parameters are most influential in creating this knowledge? To answer these questions, two main sources of state information and their influence on the system were examined: ultrasound-based and vision-based.

Ultrasound Metrology-Based Noise and Errors

The question was raised how the system would perform with poor global metrology conditions, since the chaser was dropping many pulse readings during the checkout test. Statistical simulation tests were run on the PD, passive docking test across a range of drop-out percentages to see the effect of poor metrology. A histogram (see Figure 4-15) showing bins of percent drop-out shows which tests docked and which failed. For this test, only the metrology for the chaser sphere was varied, while the initial state conditions remained fixed.

![Histogram Sphere 1 Metrology Ok Threshold, N=31](image)

**Figure 4-15:** Histogram showing pass/fails for an N-sample run with various drop-percentages for metrology without any camera data

As shown in the plot, docking becomes difficult when global metrology information is sporadic, becoming effectively impossible by the time 80% is dropped.

Metrology noise was also a variable for testing. Another docking test simulation set was run, this time keeping most parameters and initial conditions fixed and just varying the metrology noise. For a set of runs, increasing the metrology noise (as expected) adversely affects pointing and positioning making docking impossible. The
histogram shown in Figure 4-16 graphs dependence on noise level, showing in green and red which tests successfully docked.

![Histogram Sphere 1 Met DistNoise, N=81](image)

Figure 4-16: Histogram showing pass/fails for an N-sample run with increasing noise level on metrology data

The results here are not very surprising. Once noise gets up to above about 2cm, the size of the docking port hole itself, tests start having difficulty.

While this analysis clearly showed that poor global metrology would adversely affect performance, the probability of poor metrology was considered unlikely considering past performance (though there was some difficulty during checkout, behavior was better during other ISS tests). In general, sources contributing to poor global metrology included things like lights on the ISS (these conflicted with infrared timing signals), which astronauts were requested to turn off in the test procedures. Nothing additional was done to mitigate the chance of poor global metrology and any risks were accepted.

**Docking Port Camera-Based Noise and Errors**

Since the accuracy of the ultra-sound system for SPHERES was predicted to be insufficiently accurate for mechanical docking, a relative navigation system was designed, as described in Section 1.5 [32].

Much analysis (by various team members) went into improving the Kalman filter for docking using information from the docking port cameras. The simulation placed the combination of image processor and Kalman filter in a "black-box" but allowed
noise levels for the relative state to be modified. Initial experiments were conducted to gather statistics on the output of the Unscented Kalman Filter (UKF) to determine realistic values for the noise parameters, which were used in the first docking port emulator [32]. Troubles persisted with the velocity and rate estimates produced by the filter however, thus the UKF was enhanced with IMU data for future flights. As testing progressed and the UKF continued to evolve, more testing was conducted and new estimates for noise were measured and included in simulation.

Using the stochastic simulator, the noise on the relative state estimate for the docking port emulator was varied to see its effect on performance. Figure 4-17 shows how as the noise on the synthesized UKF estimate for angular rate increases, the satellites have more trouble docking.

![Figure 4-17: Histogram showing pass/fails for an N-sample run with various noise levels for relative state estimate of angular rates](image)

Interestingly, the variance has to be quite high to actually prevent docking (though contact dynamics have been over-simplified in this model). Targeting estimator vari-
ance below 0.001rad/s should not be overly strenuous. However, while the stacked histogram shows pass rates for variation in angular rate estimates, this is just one portion of the error. The entire relative state will actually be noisy, and the combination of all these errors will certainly degrade performance.

Given the high dependence of success on low noise, much effort was placed into improving the performance of the UKF by members of the team.

4.3.5 Actuator Performance

Even with perfect sensing, actuator noise could potentially hinder successful operations.

For SPHERES, since discrepancies between expected thrust and actual thrust were identified during the ISS checkout (for example, one thruster was firing less than the others), thruster output was examined as a variable. Physically, thrust might be weak when the tank is low on gas, the regulator is not set correctly, or a thruster valve is obstructed. Some simulations were run, shown in Figure 4-18 to see what the sensitivity would be to changing thruster force.

![Figure 4-18: Histogram showing pass/fails for an N-sample run varying thruster force](image)

However, when all of the thrusters are set to carry the same level of reduced force, there are no noticeable effects on performance. Further analysis by varying thruster forces individually might reveal more interesting results, so the simulation was modified to accommodate such parameter toggling.
Given the low sensitivity of the system to small differences in actual thrust levels, and the low level of control the team had over the performance on the ISS, any risks associated with thrusting were accepted.

4.4 ISS Science Test and Comparison with Simulation

On March 25, 2016 SPHERES docking tests were attempted on the ISS. Unfortunately, while the satellites came close, no docking was achieved using the mechanical docking port. An examination into why follows in this section.

4.4.1 Simulation Contributions to ISS Science Test Design

Based on observations made using various V&V tools and mass simulations, many weaknesses were identified. Several key code modifications between the ISS checkout and first science test were as follows:

- docking corridor for glideslope rather than positioning gates
- improved pointing algorithm
- different pointing (active/passive target) tests
- new and improved UKF for relative state information
- updated mass and inertia parameters

Concerns still existed before the test, however, as simulation still behave more smoothly than in hardware, and simulations did not show 100% success.

Simulations succeeded in predicting issues prior to ISS testing, despite the fact that simulated conditions are cleaner than real experimental data. Those predictions and concerns contributed to development of risk mitigation strategies. Since not all risks could be reduced entirely, some was accepted. Accepted risks included effects from
noisy relative state estimation, poor global state estimation, magnetic interference, and unmodeled contact dynamics.

4.4.2 ISS Science Test Synopsis

The general approach for docking was to initialize, distance-approach, and then glide-slope to a dock. Additionally, there was a test where the satellites were docked manually and expected to move. This test, however, was short on propellant, thus the expected trajectory was not followed.

Unfortunately, there were circumstances on-orbit which prevented docking. Additionally, due to difficulties with various other operations, less time was available for testing. Only a few of the cases were run, since time was needed to fix Wi-Fi and gas tank troubles. Results from the limited number of tests that were run show noisy ultrasound beacon information on the chaser vehicle, leading to poor state estimation. Raw data is included in appendix Figure B-4. The chaser satellite had tremendous trouble maintaining its commanded path, showing oscillations not present in prior tests. Due to these issues, the team elected to schedule a facility maintenance session before resuming science operations.

4.4.3 Comparison with Simulation

After the ISS science test, the data was processed for comparison with simulation in order to both improve the simulation, and to diagnose what went wrong during the test. Initially, it was thought that the simulation in some respects did not reflect reality, since using the same initial conditions the vehicles would dock in simulation but not in space. However, by varying parameters within simulation, such as ultrasound noise, similar effects could be observed. The satellites fail to dock, and also exhibit the same wavering seen during the ISS test.

In this manner, simulation was able to assist in diagnosing, and ultimately identifying, a contributor to the inability to dock. Further investigation into other sources of problems for these tests is currently underway.
4.5 Conclusion

Iterative experimentation and simulation contributed to improvements in the performance of flight code as well as simulation fidelity. Initially, lack of sufficient V&V was a primary contributor to unsuccessful docking attempts during the ISS checkout test. Several problems that could have been identified and solved on the ground were overlooked due to time constraints restricting testing and development of V&V tools. These were found post-checkout as diagnostic tools were created. To remedy the issue, significant effort was put into testing the system and improving simulation and analysis tools. Using the approach from Chapter 2 and the tools developed in Chapter 3, simulations were run to assess risks for soft-docking. The most influential parameters affecting docking and performance (from HW, Sim and Space assessment) were those which affected uncertainty in position and attitude, either in the form of poor global state estimation from ultrasound or poor filtering of docking port camera data. Dependence on initial location was something that could be reduced by design.

Additionally, simulation was able to identify some of the difficulties of docking, such as remaining steady enough (especially during initial target lock) to keep the target’s fiducial markings in the field of view of the chaser’s sensor. From the coding perspective, it may be possible to better tune controller gains to reduce oscillations during approach (especially during relative navigation), though this will require further analysis.

For the future, simulation and V&V enhancements might include:

- Global state estimation error-introduction

- More accurate modeling of UKF state, including dropping the first 7 estimates and associated delays

- Better timing and modeling for target-lock for proper state estimate hand-off between global and relative

- Ability to bias UKF estimator
• Post-processing for contact dynamics analysis (incoming velocity at impact)

Unfortunately, much in part due to test-setup difficulties (ultrasound issues) on the ISS during the second test, successful docking with the mechanical interface has yet to be achieved. Thus true validation has not yet been done with the SPHERES. The V&V framework created throughout this process, however, was helpful in diagnosing what went wrong during various tests and will also be helpful in reducing risk of future tests as the model becomes higher fidelity. Additionally, lessons learned about risk factors are applicable to any soft-docking mission, and not just SPHERES. Such modeling and V&V tools may even be applied to post-dock as well. In the next chapter, a subset of the modeling techniques created are used for analysis of an adaptive control mission with a target with uncertain parameters.
Chapter 5

Adaptive Joint Attitude Control Case Study

A detailed system model with diagnostic tools may be used in many phases of design. While it is especially useful for verification and validation after a preliminary design is complete, it can also be helpful in the initial concept development phase of a program. Through modeling, developers may become aware of potential difficulties earlier in the design and thus save time and money by avoiding problems.

In this chapter, a subset of the models and risk reduction processes established in this thesis were applied to study post-dock attitude control using adaptation. The goal is first to show how such processes may be applied to new problems, and second, to advance current efforts to overcome post-dock control difficulties in space given target vehicle parameter uncertainty. Adaptive control methods show potential to perform well under such circumstances, and thus warrant a closer look. Much of this work was also described in IEEE Aerospace conference proceedings [23].

The configuration of the target object may be unknown or changing. The case of jointed maneuver once a vehicle docks with a passive object (one that is neither resisting nor actively supporting motion, as described in 2.1.1) with some unknown physical parameters was selected here for study with adaptive algorithms. While the adaptive attitude control approach described here is not revolutionary, its application to a real-world system and the development of a process for testing the adaptive
algorithm on docking satellites in space is unique.

Motivation for further research in adaptive control is given, followed by development approach, and applicable dynamics and models. Then, controller design is examined. A comparison is made between an adaptive approach and a generic PID controller that has not been tuned to the new docked state physical parameters. Next, results are shown from when the controller was implemented in the SPHERES hardware for a simple code-validation test on a 3 degree of freedom hardware test setup. Finally, conclusions discuss limitations and future extensions.

5.1 Controller Development and Test Approach

The first step in developing the control system was to mathematically describe the dynamics the vehicles would encounter. These equations had to be simulated. Details may be found in Appendix A.1. After that, the satellites themselves could be modeled, using available state and parameter information. Next, the controllers could be designed and integrated into the simulation. Once a closed-loop system was created, effort went into ensuring that references signals were sufficient to test the design space and the control goals of stability, robustness and performance.

For controls analysis, a MATLAB Simulink framework was created to allow simulated testing of system response to various reference commands and compare performance. The simulation consists of a reference signal generator, the controller, the plant that emulates the dynamics for the individual or docked satellites, a feed-through for sensors, and data analysis tools. Various controller designs can be tested in this framework side by side. The simulation developed for this test includes modules developed and described previously in Chapter 3.

Testing stability, performance, and robustness is the goal. Care was given to design cases that would test stability by spanning the operational envelope, and assess performance by selecting nominal conditions. Robustness was tested by simulating performance under disturbances. For checking general functionality, initial conditions were varied and multiple tests were run in succession. All cases were tested in both the
docked and undocked configurations. Below is a table showing various combinations of reference signals used in testing:

Table 5.1: Reference command combinations

<table>
<thead>
<tr>
<th>Position Step (0 Velocity)</th>
<th>Quaternion Step (0 Angular Rates)</th>
<th>Constant Angular Rate (Continuous Quaternion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Hold</td>
<td>- Hold</td>
<td>- Position hold</td>
</tr>
<tr>
<td>- Step +/- each direction or axis</td>
<td>- Step +/- each direction or axis</td>
<td>- Position step</td>
</tr>
<tr>
<td>- Random attitude or position</td>
<td>- Random attitude or position</td>
<td></td>
</tr>
</tbody>
</table>

Of particular interest is the feasibility of various commands. Not all orientations and all positions can be accessed directly from one another without destabilizing the system. For example, commanding a large off-axis translation while holding a particular orientation will induce torques that cannot be overcome with this hardware. Thus test cases had to be chosen that did not over-strain the system. A perfectly tuned PID controller should perform adequately, otherwise there would be no hope for the adaptive version. By design, the implicit goal of an adaptive PID is to achieve the ideal PID performance.

After a controller design passed initial simulation testing, the controller was implemented in C-code just as it would in SPHERES flight code. Once programmed, the code can be run in a more sophisticated simulation which emulates the sensor noise, timing delays, thruster disturbances, and other subtleties [46], [24], [36]. This simulation has already been validated with data results from operations in space on board the ISS. Simulations were run using each controller with the same reference mission for comparison. For this analysis, only a standard PID and an adaptive PID were compared because the standard PID not only has heritage on undocked flights, but flew on a docking test on the ISS. Future efforts will involve comparing more types of controllers.

Finally, the control code was implemented in hardware on the ground for a simple
test on a 3 degree of freedom hardware test setup (described in Section 1.5) to verify function.

5.2 PID Baseline Control

5.2.1 Controller

PD and PID controllers have operated successfully on SPHERES satellites for years. The appropriate gains were computed based on analysis from flight testing onboard the ISS. Docking ports, however, are new to the ISS. For this analysis, since the exact response for the SPHERES satellites with a docking port attached to it has not been validated, the gains were initially tuned to match the expected physical parameters of the satellites with a docking port. Using results from the first hardware checkout test on the ISS, gains were further tuned using a validated approach [31].

5.2.2 Limitations

Due to the fixed nature of the PID gains, limitations exist. When the true system center of mass moves away from the expected center of mass, force commands induce more and more torque on the system. Additionally, the increased inertia means torque commands have less and less effect. Ideally, the gains for the torque commands would increase after docking, however this system does not adapt. In general, this means the system will not be damped correctly and may even go unstable should conditions deviate too far from nominal.

5.3 Adaptive Controller Design

5.3.1 Adaptive Controller Selection

Several different types of adaptive controllers were examined before selecting an adaptive PID for further analysis. An adaptive PID and a model reference adaptive
controller (MRAC), were modeled. The MRAC showed much promise, and may outperform the adaptive PID, as there has been much study into how to account for actuator saturation using this control method. However it was not selected for further study since it would not be easy to directly compare it with a non-adaptive PID. Future work may involve implementation of such a design.

In selecting the choice method, a number of questions required answers. For instance, what parameters are unknown? Is it simply necessary to control the system, or do the true parameter values need to be determined? What control over the input is acceptable (i.e., can it be made persistently exciting)? Are there system constraints like actuator saturation limits? Answers to these questions guided selection. The decision tree in Figure 5-1 shows how one might determine the type of control system best suited for an application.

Figure 5-1: Decision tree for deciding type of controller

For the joint maneuver control problem, there were large uncertainties, limited system knowledge, and parameter information was not necessary. Thus direct control was selected. Here, one satellite docks to another without "knowing" the inertia or mass of the other target satellite, or the location of the new center of mass. This leads to three key changes:

- Force commands now produce torques as well (due to new center of mass location)
Commanded torque and force magnitudes stay the same, but act about the new center of mass, influenced by the combined inertia.

Increase in mass means more force is required for the same acceleration.

In a state-space representation of the SPHERES plant, this means both the A and B matrices change. The thruster mixing matrix (the conversion from forces and torques to twelve thruster firings) will change as well (and may or may not be part of the B matrix depending on the designer’s choice).

There are also two options for control command, force and torque, or direct thruster control. Adaptation with direct thruster control would, in theory, allow a more optimal solution than the constraint of force and torque commands. Only positive thrusts are physically possible, however, requiring a mapping between negative commands and negative direction thrusters. A potential extension for adaptation would be to allow a gain factor for each individual thruster. This would help with thruster-failure cases where a thruster is not operating at full thrust for some mechanical reason (which has happened on orbit).

Another choice is whether to combine position and attitude control, or treat them separately. Since forces induce torques in this joint system, if information between the position and attitude controllers is not shared, commanded position changes will appear as external torque disturbances to the attitude controller. For this example, a combination position and attitude adaptive controller was modeled, to see if there were advantages to integrating the two given the coupling between force commands and torques. It did not show significant improvement over the independent controllers, thus an adaptive PID attitude controller combined with a standard PID position controller was used.

5.3.2 Control Laws

For this adaptive PID controller, the plant was assumed second order of the form in equation (5.1) below, where the true matrix parameters J and B are unknown and must be estimated adaptively. This format allows changes to both the first and second
order terms, which include the inertia and center of mass dependencies. Background for the adaptive PID approach and corresponding stability analysis can be found in lecture notes [3]. The basic controller design itself is not a novel contribution of this thesis, however the implementation, and simulation results for spacecraft attitude control of docked satellites is unique.

\[
\frac{1}{s(js + B)} \quad J > 0
\]  

(5.1)

Matching conditions were chosen to select the constant gains \( K \) and \( \lambda \), such that the gains \( K_p^* \) and \( K_d^* \) are the tuned position and derivative gains from the PID controller for the standard configuration inertia.

\[
K = K_d^*
\]  

(5.2)

\[
\lambda = \frac{K_p^*}{2K}
\]  

(5.3)

Attitude error equations are then derived from the desired state vector \( x_d \) from (A.5) and the current state vector \( x \) via a combination of equations A.10, (5.4) and (5.5). Equation (A.10) ensures that quaternion errors will tend to zero.

\[
e_1 = \ddot{x}_d + 2\lambda \dot{e} + \lambda^2 e
\]  

(5.4)

\[
e_2 = \dot{e} + 2\lambda e + \lambda^2 \int e dt
\]  

(5.5)

Adaptive gain matrices \( \dot{J} \) and \( \dot{B} \), representing the difference between the estimated and true values of plant parameters \( J \) and \( B \), are computed from the integrals of equations (5.6) and (5.7).

\[
\dot{J} = e_1 e_2
\]  

(5.6)
\[ \dot{B} = \dot{x} e_2 \]  

(5.7)

The commanded torque is computed below:

\[ \tau = \ddot{J} e_1 + \dot{B} \dot{x} + K e_2 \]  

(5.8)

Figure 5-2 shows how these equations fit together into a system model. The reference signal (orange) is used to compute errors (e, e1 and e2), which feed into the adaptive gain matrix computations. The gains are multiplied by errors to compute the necessary torques (yellow). The forces and torques are converted to commands using the thruster matrix, and thus control the plant. Adaptation halts when the error (e) is small, and inside a ‘deadzone’ (see purple).

![Figure 5-2: Adaptive model diagram](image)

### 5.3.3 Stability Analysis

For proving mathematical stability, select the Lyapunov equation (5.9):

\[ V(e_2, \ddot{J}, \dot{B}) = \frac{1}{2} (J e_2^2 + \ddot{J}^2 + \dot{B}^2) \]  

(5.9)

Taking the derivative of equation (5.9) and simplifying using (5.6) and (5.7) gives:

\[ \dot{V} = -K e_2^2 \leq 0 \]  

(5.10)
Since the derivative of the Lyapunov equation is negative semi-definite, $V(0) = 0$, and the below sequence of bounds can be inferred, finally proving stability via Barbalat’s Lemma [22], [34].

\[ e_2 \in L^2, e_2 \in L^\infty, \dot{e}_2 \in L^\infty, \bar{J} \in L^\infty, \bar{B} \in L^\infty, e \in L^\infty \] (5.11)

Note, this simply shows mathematical stability, and does not mean the system is ensured stable given all implementation intricacies.

### 5.3.4 Robustness

To prevent adaptation when errors are sufficiently small ($|MRP_{1,2,3}| < 0.1$), an adaptation dead-zone was implemented on both the adaptive gain matrices $J$ and $B$ as shown in (5.12) and (5.13). The dead-zone value was chosen to be large enough to capture runaway, but small enough that biases would not cause performance issues. Without the dead-zone, not only did the controller spend fuel adapting unnecessarily, but the gains could grow unchecked with any minor disturbance (such as a force command) when errors are small. Dead-zones help with robustness when persistent excitation is not guaranteed [22], [34].

\[
\begin{align*}
\dot{\bar{J}} &= \begin{cases} 
e_1 e_2 & |MRP_{1,2,3}| > 0.1 \\
0 & |MRP_{1,2,3}| \leq 0.1 \end{cases} \\
\dot{\bar{B}} &= \begin{cases} \dot{x} e_2 & |MRP_{1,2,3}| > 0.1 \\
0 & |MRP_{1,2,3}| \leq 0.1 \end{cases}
\end{align*}
\] (5.12) (5.13)

### 5.3.5 Further Improvements

The current design could be made more robust by adding actuator saturation protection, and adaptive gain runaway protection like projection [34]. These would increase robustness to disturbances as well. A real flight system should require these safeties.

In particular, actuator saturation protection may be added by adding an addi-
tional adaptive gain, as outlined in the work of Karason and Annaswamy [25]. Initially, actuator saturation times were considered short for torque commands in this application, thus protection was not included. However, results demonstrate the need for improvement in future work. For a model reference adaptive control approach, Karason, Annaswamy and Levretsky actually all worked out solutions for actuator saturation protection [25], [28]. Perhaps a similar approach could be used in this PID case, though it would require making a reference model to estimate the controllable error.

5.4 Adaptive Controller Verification and Validation Process

5.4.1 Standalone Simulation Results

The simulation results in Figure 5-3 and Figure 5-4 show the post-dock response to a step attitude command during position hold—an arbitrary pure rotation of \(-67\) degrees about the z-axis.
The PID controller is highly under-damped, while the adaptive PID changes gains quickly before locking on values for the adaptive gain matrices J and B in the dead-zone once the MRP error becomes low. Since this particular case involves a pure axial rotation, the cross terms in the gains do not move to different values. Both succeeded in performing the position hold through the attitude step.

Numerous additional cases were run, many exhibiting more adaptation than the example above. In general, the simple PID was severely under-damped, and suffered during combination maneuvers when forces would exert additional torques. In
simple cases the adaptive controller would outperform the PID controller. The adaptive controller exhibited limitations as well, specifically when saturation limits were hit. Integrating a reliable method for handling actuator saturation in the adaptive algorithm would be essential for robust control on a spacecraft.

Additionally, the identical cases were run for the undocked case (see Figure 5-5). As expected, the PID (which was tuned for this case) outperformed the adaptive method, but adaptation still produced stable results.

![Undocked PID Angular Rates](image1)

![Undocked PID Torque Cmd](image2)

Figure 5-5: Undocked PID controller attitude response

5.4.2 Integrated Simulation Results

Simulations were run to assess stability and performance of the adaptive controller as well. Figure 5-6 is a typical scenario, beginning with the two SPHERES disjoint, where the primary vehicle approaches straight on, docks (modeled as a purely inelastic collision), and then attempts a position step followed by a 180 degree rotation.
During the pre-dock phase, the standard controller performed better, as expected, because energy was not wasted adapting an already tuned solution. The adaptive attitude controller often performed better than the standard controller in the high-fidelity simulation docked case, using the single-satellite gains and physical parameters as the baseline for both controllers. It was not superior in all cases, however, and still exhibited oscillatory motion when commanded rotations required thruster saturation.

5.4.3 Adaptive Controller Testing on SPHERES Hardware

For hardware testing, a ground platform exists where algorithms can be loaded into SPHERES placed on ‘air carriages’ that float on low-friction glass-top table for 3DoF testing [46].
Besides the obvious gravitational inconvenience which restricts operations to 3DoF, the setup (Figure 5-7) has some additional limitations. It is not truly frictionless, the table is not perfectly flat, and SPHERES must be placed in carriages that alter the inertia characteristics of the SPHERES. Given these realities, only basic functionality can be tested in hardware and performance on the table does not necessarily reflect performance in space. The benefit of hardware testing, however, is to gain confidence in system functionality with realistic timing and sensing.

Once the controller was designed and tested in simulation, an implementation of it was written in C-code to run on the SPHERES satellites. Running the code on the glass table helps to validate the software. A simple one-satellite only test was conducted, performing an attitude hold during a simple position step command followed by an attitude adjustment. The inertia parameter set in the model at the start of the test was different from the true inertia. The vehicle succeeded in maintaining control authority.

Future work entails running more strenuous tests on the glass-table to validate the controller, including docked maneuvers, with the expectation of running SPHERES
flight experiments on-board the ISS.

5.5 Conclusion

Adaptive attitude control for spacecraft has potential for spacecraft applications when physical parameters are uncertain. This work motivated why an adaptive approach may be helpful for applications like autonomous spacecraft docking, spacecraft servicing, and robotic assembly. A formulation of an adaptive attitude controller for an experimental space flight program was tested and developed for future space implementation, a significant step towards reducing risk in using such types of controllers in space. A comparison was made with a baseline PID controller, and advantages were seen in cases where parameters varied drastically from expected values. Limitations of the design, such as actuator saturation handling, were identified for further work. Future study should involve not only improvement on design through iterated simulation, but experimentation in a true environment to avoid the simplifications of modeling.

One recommendation for further investigation would be to allow the adaptive controller to control the twelve thrusters individually, rather than commanding predetermined force and torque pairs. This may enable the controller to command forces which would induce a much greater torque than the torque pair, and thus avoid saturation.

Additional controller types, such as MRAC or non-adaptive controllers, may also be assessed using this approach in the future. The particular choice of the non-adaptive and adaptive PID controllers was highlighted here since non-adaptive PID control will be used with tuned gains for other SPHERES tests, providing flight data for comparison. In the future, more types of controllers may be tested and selected for SPHERES docked testing on the ISS.
Chapter 6

Conclusion

6.1 Summary

This thesis presented a verification and validation framework development approach for application to spacecraft soft-docking with uncertain dynamic properties. The work was motivated by recent interest in challenging autonomous missions such as spacecraft servicing, reconfiguration and debris removal, as well as a history of incidents due to the complexity of such tasks.

In Chapter 1, the problems for autonomous soft-docking and adaptive control were outlined, providing motivation for further study. More specifically, NASA, DARPA and the USAF have all identified autonomous soft-docking as a priority for future missions for both scientific and security reasons. The expense of in-space testing and difficulties associated with ground testing for space missions (along with various in-space failures) drive the need for developing targeted V&V processes to reduce risks for docking. The history and current state of autonomous docking, as well as adaptive attitude control methods for spacecraft, were summarized to further illustrate the need for V&V using flight-testing. Utilizing the SPHERES facility, in simulation, on the ground, and in space provides a method for validating the validation tools.

Chapter 2 described some pre-existing risk reduction methods like FMECA, STPA and some experimental design principles to demonstrate how they could be integrated and applied to soft-docking. The notion of passive and collaborative vehicles was in-
introduced revealing a difference in strategy required for such risk reduction. Using these techniques and knowledge of the application, a V&V framework was devised. This risk-reduction approach combines system modeling, diagnostics, experimentation, and Monte Carlo methods to develop a comprehensive V&V solution specifically for autonomous docking. An example was provided using SPHERES to show how insights from risk analysis should fold into V&V processes.

Next, in Chapter 3, a verification and validation framework was developed by modeling the SPHERES docking system and creating diagnostic tools and stochastic capability for simulated testing. Care was taken to provide an interface to actual flight software, to increase confidence in software function. Details for modeling system dynamics as well as the docking port were described to give insight on how one would develop such models for any system. Risk factors outlined in the Chapter 2 were converted to variables that were used as the basis for stochastic simulation testing.

Chapter 4 applied the tools developed in the Chapter 3 to show how iterations between ground testing, simulation, and testing in space could improve confidence in both the SPHERES soft-docking system, and the presented verification and validation approach. Setbacks during initial checkout, including a lack of V&V tools were identified and remedied. A master simulation assessment was conducted using experimental design methods of randomizing a large number of variables or examining the influence of variables individually to determine areas of sensitivity applicable to both SPHERES and generic soft-docking missions.

And finally, Chapter 5 applied what was developed in part of this thesis to the problem of post-dock attitude control to demonstrate how strengths and weaknesses of a design could be discovered through simulated and hardware testing.

Despite the lack of flight evidence for soft-docking on SPHERES, the process presented by this thesis was useful in both reducing risk and identifying issues. Space facility problems like propellant tank issues and noisy ultrasound-based state estimation were identified and accepted risks. Additional considerations, such as relative estimation errors, unexpected initial conditions, and interference from un-modeled
dynamics, were all brought to light by V&V techniques utilized in this research. Modeling even predicted issues with relative-state-only navigation in close range, as tips-offs could move the target out of view during simulation. Even advantages and setbacks with adaptive attitude control techniques were found through careful simulation of the system in its environment.

Certainly there were many lessons learned from development throughout this research. One salient lesson is how simulation may (and should) guide testing, and how testing results should feed back into simulation. Simulation was most useful in testing conditions that could not be tested on the ground (rotations and locations only accessible in 6DoF), as well as gaining breadth since only a limited number of tests could be performed in hardware.

6.2 Contributions

Primary thesis contributions are listed here:

- Developed integrated verification and validation approach and used it to create spacecraft docking risk-reduction tools, including system models, stochastic tests, and improved diagnostics

- Applied verification and validation methods and tools to SPHERES through simulation, ground testing, and ISS testing

  - Identified important soft-dock parameters

  - Improved system model

  - Analyzed performance of 6 DoF testing using V&V process

- Demonstrated feasibility of adaptive attitude control for post-dock maneuvers with partly-unknown object by developing and testing (in simulation and on ground hardware) a controller and applying a subset of V&V techniques as well as system modeling to increase probability of future success
6.3 Future Work

Numerous areas for further work exist given the breadth of coverage of this work: from V&V, to autonomous soft-docking and adaptive attitude control. First and foremost, completing a true demonstration of SPHERES satellite autonomous soft-docking on the ISS would be invaluable. Second, often the most deleterious mishaps stem from "unknown-unknowns" and false "knowns." For example, extremely poor metrology was known to cause issues, however there was false confidence in metrology quality. Further research into risk management and V&V for identifying potential unknown-unknowns and false-knowns by removing the human from the loop and relying on autonomy to find risks in an unbiased fashion may provide useful results. Secondary to these, below is a list of proposed extensions by topic:

- Satellite flight software updates
  - Fix metrology/state estimator problem
  - Tune controller gains

- SPHERES simulation model improvements
  - Improve simulation even more by making docking criteria more stringent
  - Add third satellite to docking simulation for multi-satellite reconfiguration

- Stochastic simulating improvements
  - Global state estimation error-introduction
  - More accurate modeling of UKF state, including dropping the first 7 estimates and associated delays
  - Better timing and modeling for target-lock for proper state estimate hand-off between global and relative
  - Ability to bias UKF estimator
  - Post-processing for contact dynamics analysis (incoming velocity at impact)
- Add multi-set run post-processing functionality for pooling together numerous simulation campaigns for statistical purposes

- Parameter Sensitivity and Uncertainty Analysis
  - Use risk reduction for more sensitivity analysis
  - Continue to develop more post-processing tools which automatically assess relative sensitivity to various parameters

- V&V
  - Research into automated detection of "unknown-unknowns"
  - Research into relaxation of false "knowns" for detection of potential issues

- Adaptive attitude control
  - Develop better performing adaptive controllers which handle actuator saturation
  - Conduct more ground hardware testing to see controller robustness to disturbances
  - Test and compare different types of adaptive control (MRAC, PID, etc.)
  - Validate adaptive attitude control in space for post-dock maneuvers with uncertainty
  - Run Monte Carlo assessment for performance under uncertainty
Appendix A

Appendix: SPHERES Simulations

A.1 Spacecraft Dynamics and State

A.1.1 Spacecraft Attitude Dynamics

Generally, for spacecraft in orbit, the position and attitude dynamics can largely be decoupled. Spacecraft that only use reaction wheels for transferring angular momentum are not even capable of translating. SPHERES however, solely use gas thrusters and are not equipped with wheels. Using thrusters to command a force can cause some undesired rotation, due to imperfections in the propulsion system. If however the location of the center of mass changes, like in the post-dock case, commands that would previously have produced a pure force now induce a torque on the vehicle as well. Therefore, there exists a coupling between forces and torques in a thruster-actuated satellite system like this one.

For this problem, two types of coordinate frames are defined — a global-frame and a body-frame. The transformation between the two frames is represented as a four element quaternion, scalar last. When SPHERES are docked together, each SPHERES keeps its body frame, however the pair moves in unison and the system center of mass changes along with the system mass and inertia.
The satellite behavior is modeled according to the following equations [44], [5]:

\[
\dot{Q} = \frac{1}{2} \Omega Q
\]  
(A.1)

\[
\dot{\omega} = I_3^{-1}((-\omega \times I_3 \omega) + \tau)
\]  
(A.2)

\[
\tau = \tau_{cmd} + r \times (F_{cmd} + F_d) + \tau_d
\]  
(A.3)

In equation (A.1), \( Q \) represents the 4-element quaternion rotation (defined here scalar-last) between the global frame and the body frame, where \( \dot{Q} \) is its derivative. The variable \( \Omega \) represents the 4x4 skew matrix of the body-frame angular rates, the three element vector \( \omega \).

Equation (A.2) serves to find the rate of change of angular rates (\( \dot{\omega} \)) from known quantities like the 3x3 spacecraft inertia matrix (\( I_3 \)), and the torque (\( \tau \)) provided by the thrusters. Computing the actual system torques in equation (A.3) means summing the commanded torque, the cross product of the vector \( r \) the body-frame position vector between the applied force and the system center of mass, and any disturbance torque (\( \tau_d \)). The applied force consists of the thruster-commanded force (\( F_{cmd} \)), and any disturbance force (\( F_d \)) applied by the thrusters.

### A.1.2 Spacecraft State Determination

For simulation purposes, the above dynamic equations are implemented using discrete approximations. The vehicle 13-element state-vector shown below in equation (A.5), consisting of position, velocity, attitude quaternion, and angular rate, is propagated in each step based on controller force and torque commands, as well as any modeled random disturbances.

\[
x_{full} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & Q_1 & Q_2 & Q_3 & Q_s & \omega_x & \omega_y & \omega_z \end{bmatrix}^T
\]  
(A.4)
For the attitude-only computations used in the adaptive controller, only the last 7 elements of the state vector are used, resulting in a state shown in (A.5).

\[
x_{\text{att}} = \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_s & \omega_x & \omega_y & \omega_z \end{bmatrix}^T
\]  

(A.5)

\[
x = \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix}^T
\]  

(A.6)

\[
\dot{x} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T
\]  

(A.7)

\[
\ddot{x} = \begin{bmatrix} \dot{\omega}_x & \dot{\omega}_y & \dot{\omega}_z \end{bmatrix}^T
\]  

(A.8)

Given we would like to compute the error (e) between the desired state \((x_d)\) and the actual state \((x)\), the quaternion values are replaced with MRP values.

### A.1.3 State-Vector Error Computation

The error between the estimated state vector and the reference state vector is simply the difference between them, with the exception of the quaternion which is handled differently.

By definition, quaternions are non-unique in that a coordinate frame transform may be represented by more than one quaternion. For instance reversing the sign of a quaternion represents the same rotation. Therefore simple subtraction will not provide meaningful error information. Instead one can find the error quaternion, the necessary transform between the desired quaternion \((Q_d)\) and estimated quaternion \((Q)\) which is the first four elements of the state vector \(x\) from (A.5), by using equation (A.9) below. Here quaternions are multiplied \((\otimes)\) via quaternion multiplication rules and \(*\) represents the quaternion conjugate.

\[
Q_e = Q_d \otimes Q^*
\]  

(A.9)
A shortcoming of the error quaternion, for our purposes, is that it is normalized to unity making it difficult to minimize errors. A well-documented solution is to convert the error quaternion into Modified Rodriguez Parameters to allow error minimization, using the follow equation [32], [13].

\[
e = [M RP_1, M RP_2, M RP_3] = \left[ \frac{Q_{e1}}{4(Q_{e4} + 1)}, \frac{Q_{e2}}{4(Q_{e4} + 1)}, \frac{Q_{e3}}{4(Q_{e4} + 1)} \right] \tag{A.10}
\]

### A.1.4 Spacecraft Parameters

For reference, the following physical parameters were used in simulation to provide a realistic scenario for the undocked and docked cases. These parameters are based on the SPHERES.

**Table A.1: SPHERES Physical Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Sat Inertia ([km^2])</td>
<td>\begin{bmatrix} 0.0289 &amp; -0.0003 &amp; 0.0011 \ -0.0003 &amp; 0.0560 &amp; 0.0002 \ 0.0011 &amp; 0.0002 &amp; 0.0588 \end{bmatrix}</td>
</tr>
<tr>
<td>Docked Sats. Inertia ([km^2])</td>
<td>\begin{bmatrix} 0.0582 &amp; -0.0161 &amp; 0 \ -0.0161 &amp; 0.6382 &amp; 0 \ 0 &amp; 0 &amp; 0.6443 \end{bmatrix}</td>
</tr>
<tr>
<td>Docked Center of Mass offset ([m])</td>
<td>([-0.2097, 0.0062, 0.0178])</td>
</tr>
<tr>
<td>Docked Mass ([kg])</td>
<td>11.96</td>
</tr>
<tr>
<td>Maximum Thrust per valve ([N])</td>
<td>0.095</td>
</tr>
<tr>
<td>Maximum Torque per valve ([Nm])</td>
<td>0.0092</td>
</tr>
</tbody>
</table>

### A.1.5 Docked Attitude Control Case

For the case of joint maneuver once the primary vehicle has attached to an unknown target, only the primary is given authority to command thrusters. The target vehicle is assumed completely passive in that it does not actively help or resist motion.
A.2 Operating Instructions for Running Monte Carlo Simulation

A.3 Monte Carlo Sample Size Computation

For quickly approximating how many simulations should be run, the GUI shows the relation (for a binomial pass/fail distribution) between number of samples, error of mean, and confidence using Chebyshev’s inequality, and assuming a mean of 50% (which is worst-case). For example, to get 95% confidence that the pass rate (from your simulations, whatever that may be) is correct to ± error (ε), you need at least a number of runs (n). The actual (conservative) lower bound for a given confidence could then be determined using Chebyshev’s inequality after the success probability (p) is determined via post-processing.

\[ P(|X - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2} \]  \hspace{1cm} (A.11)

For a binomial random variable (B), the variance is...

\[ \sigma^2 = np(1 - p) \]  \hspace{1cm} (A.12)

Assume that \( X = (\sum_{i=1}^{n} B_i)/n \) and the mean \( \mu = p \... \)

\[ Var \left( \frac{B}{n} \right) = \left( \frac{1}{n} \right)^2 Var(B) = \left( \frac{1}{n^2} \right) np(1 - p) = \frac{p(1-p)}{n} \]  \hspace{1cm} (A.13)

Given worst-case, \( p=1/2 \... \)

\[ P(|B/n - p| \geq \varepsilon) \leq n/(4\varepsilon^2) \]  \hspace{1cm} (A.14)

Since we want confidence, we take

\[ 1 - P(|B/n - p| \geq \varepsilon) \]  \hspace{1cm} (A.15)
Solving for \( n \... \)

\[
n \geq \frac{1}{4\varepsilon^2(1 - P(|B/n - p| \geq \varepsilon))} \tag{A.16}
\]

## A.4 Simulation Code

Please contact Dr. Alvar Saenz-Otero of the Space Systems Laboratory at MIT for access to SPHERES code developed in support of this thesis.
Appendix B

Appendix: Result Graphs

B.1 ISS Checkout Test

Figure B-1 and B-2 show a comparison of commands, ISS actual results, and expected results from simulation using the ISS initial conditions.
Figure B-1: Test 6d chaser position, actual and commanded on ISS, and simulated using ISS initial conditions

Figure B-2: Test 6d chaser quaternion, actual and commanded on ISS, and simulated using ISS initial conditions

B.2 ISS Science Test

The SPHERES facility state-estimation system is based on a timed IR pulse and ultrasound chirps from beacons planted around the volume [42], [36]. On the ISS on both the checkout test, and even more so on the science test, the estimation on one of the satellites was fairly poor. Noisy and dropped data from ultrasound beacons contributed to the problem. Figures B-3 and B-4 contrast the difference in spurious and false beacon readings between good and bad beacon two on the two satellites during the ISS Science test in March 2016.
Figure B-3: Clean Beacon Data
Figure B-4: Noisy Beacon Data
Bibliography


