

Visual Spacecraft Relative Motion Control using Higher Order Geometric Moments

by

Samantha C. Krening

B.S., University of Colorado at Boulder, 2011

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Master of Science
Department of Aerospace Engineering Sciences
2011

The thesis entitled:
Visual Spacecraft Relative Motion Control using Higher Order Geometric Moments

Written by Samantha C. Krening

Has been approved for the Department of Aerospace Engineering Sciences

Dr. Hanspeter Schaub

Dr. Jeffrey Luftig

Dr. Penina Axelrad

Date_____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Samantha C. Krening

Visual Spacecraft Relative Motion Control using Higher Order Geometric Moments

Thesis directed by Dr. Hanspeter Schaub

This thesis studies passive visual relative control for satellites. The three main problems studied are how to keep the camera pointing at the center of the target, how to move the satellite so the camera can look at the target from a perpendicular orientation or maintain a fixed orientation, and how to maintain a nominal distance from the target.

Visual target tracking within the image uses a statistical pressure snake algorithm, which tracks the outer contour of a target and allows geometric moments to be calculated. The attitude control is based off of tracking the center of the visual target, which is the first moment. The attitude control can treat the inertial target angular velocities as disturbances in the control for slow, smooth reference motions and still be stable. The skewness coefficient g_3 , which is a non-dimensional form of the third moment, can be used as a measure of perpendicularity when certain information is known about the target. A 640x480 pixel resolution camera is used to obtain better skewness and perpendicularity information than the human eye. A distance control does not need to assume the absolute distance, but can maintain a nominal distance by using the image inertia, or second moment information.

Acknowledgements

Dr. Hanspeter Schaub. For the endless 8:00 a.m. verbal quizzes in class, being a very knowledgeable advisor, and being a teacher when there are a lot of other things you could be doing. Thank you.

Dr. Jeffrey Luftig. His inferential statistics classes opened my eyes to the world of data. Without the concepts of EMEN 5005, I would not have had my light bulb moment over Christmas and this thesis would not have been the same.

Dr. Penny Axelrad. Thank you for agreeing to be on my committee at the last minute.

Steve O'Keefe. For helping interface the new GUI with old simulations.

Contents

CHAPTER 1 INTRODUCTION	13
1.1 MOTIVATION	13
1.1.1 <i>Passive vs. Active</i>	16
1.1.2 <i>Inertial vs. Relative</i>	17
1.1.3 <i>Overview</i>	19
1.1 LITERATURE REVIEW	21
1.1.1 <i>Voyager Uranus Encounter</i> ¹⁴	21
1.1.2 <i>Galileo Gaspra Encounter</i> ¹²	22
1.1.3 <i>Configuring the Deep Impact AutoNav System for Lunar, Comet, and Mars Landings</i> ¹⁰	23
1.1.4 <i>Autonomous Helicopter using Feature Tracking</i> ¹⁵	25
1.1.5 <i>STARDUST Wild 2 Encounter</i> ¹⁷	26
1.1.6 <i>Moment Functions in Image Analysis: Theory and Applications</i> ¹	27
1.1.7 <i>Statistical Moments, Jamie Shutler</i> ⁵	28
1.2 OVERVIEW/SCOPE	29
1.2.1 <i>Attitude Control</i>	31
1.2.2 <i>Skewness Control</i>	32
1.2.3 <i>Distance Control</i>	33
CHAPTER 2 PROJECT BACKGROUND.....	34
2.1 UMBRA.....	34
2.2 DATA FLOW	35
CHAPTER 3 ATTITUDE CONTROL	37
3.1 ATTITUDE CONTROL DEVELOPMENT.....	37
3.2 SENSING.....	39
3.2.1 <i>Relative Motion</i>	40

3.2.2	<i>Body Motion</i>	50
3.2.3	<i>Target Motion</i>	52
3.2.4	<i>Summary</i>	53
3.3	ATTITUDE SIMULATION MODULE SETUP	53
CHAPTER 4 VISUAL SKEWNESS CONTROL PROPERTIES		56
4.1	BACKGROUND OF GEOMETRIC MOMENTS	56
4.1.1	<i>Overview</i>	56
4.1.2	<i>Moments Across Disciplines</i>	57
4.2	SKEWNESS PROOF	58
4.2.1	<i>Mass/Image and Statistics Moments are the Same Concept</i>	59
4.2.2	<i>Differences between Mass/Image and Statistics Moments</i>	63
4.2.3	<i>g_3 can be used as a Control Variable for Perpendicularity</i>	64
CHAPTER 5 POSITION CONTROL		79
5.1	CAMERA FRAME SETUP	79
5.2	FULL POSITION CONTROL	79
5.2.1	<i>Position Propagator</i>	80
5.3	SKEWNESS CONTROL DEVELOPMENT	80
5.4	DISTANCE CONTROL DEVELOPMENT	83
5.4.1	<i>Pin-hole Camera Model</i>	83
5.4.2	<i>Control in which Initial Distance is Known</i>	86
5.4.3	<i>Control in which Initial Distance is Not Known</i>	87
5.5	CONTROL FLOW CHART	89
CHAPTER 6 AUTONOMOUS SIMULATION		92
6.1	SIMULATION OVERVIEW	92
6.2	MODULE SETUP	93

6.3	EXAMPLE SIMULATION	97
CHAPTER 7 SUMMARY		99
CHAPTER 8 APPENDIX – MODULES		105
8.1	SIMULATED TARGET ATTITUDE.....	105
8.2	RELATIVE CONTROL LAW	105
8.3	FILTER.....	105
8.4	STATISTICAL PRESSURE SNAKES.....	106
8.5	SIMPLE POSITION PROPAGATOR.....	106
8.6	SKEWNESS CONTROL LAW	107

Tables

TABLE 1: MUKUNDAN'S COMPARISON OF MOMENTS ACROSS FIELDS.....	27
TABLE 2: ATTITUDE VARIABLES.....	37
TABLE 3: ATTITUDE REGULATION INITIAL CONDITIONS	47
TABLE 4: ATTITUDE TRACKING INITIAL CONDITIONS	48
TABLE 5: COMPARISON OF MASS, IMAGE AND STATISTICS MOMENTS.....	63
TABLE 6: COMPARISON OF RECTANGLE AND TRAPEZOID.....	69
TABLE 7: COMPARISON OF POSITIVELY SKEWED SHAPE AND DISTRIBUTION	70

Figures

FIGURE 1: BASIC OBJECTIVE OF VISUAL TRACKING ALGORITHM	14
FIGURE 2: EXAMPLE OF MAINTAINING POINTING WITH VISUAL DATA	14
FIGURE 3: TOP VIEW OF SKEWED ORIENTATION	15
FIGURE 4: COLLISION AFTER FAILURE TO MAINTAIN NOMINAL DISTANCE	15
FIGURE 5: LOSING SIGHT OF THE TARGET AFTER FAILURE TO MAINTAIN NOMINAL DISTANCE	16
FIGURE 6: INERTIAL CONTROL ^{11, 18, 19}	18
FIGURE 7: RELATIVE CONTROL ^{11, 18, 19}	19
FIGURE 8: DOCKING WITH THE ISS ¹⁶	20
FIGURE 9: GASPR FROM GALILEO ¹¹	22
FIGURE 10: TEMPEL-1 COMET VIEWED DURING THE NEXT MISSION.....	24
FIGURE 11: LANDING OPTICAL NAVIGATION LANDMARKS.....	25
FIGURE 12: HELICOPTER DURING A TEST	26
FIGURE 13: LETTER RECOGNITION ⁵	29
FIGURE 14: OVERVIEW OF CONTROLS.....	31
FIGURE 15: ATTITUDE CONTROL NOT WORKING	32
FIGURE 16: ATTITUDE CONTROL WORKING	32
FIGURE 17: SKEWNESS CONTROL NOT WORKING	33
FIGURE 18: SKEWNESS CONTROL WORKING	33
FIGURE 19: DISTANCE CONTROL NOT WORKING	33
FIGURE 20: DISTANCE CONTROL WORKING	33
FIGURE 21: UMBRA SAMPLE MODULE.....	34
FIGURE 22: UMBRA SAMPLE DATA FLOW	35
FIGURE 23: STATISTICAL PRESSURE SNAKES TRACKING A HARD HAT ⁶	36
FIGURE 24: SNAKE CONTROL POINTS ⁶	36
FIGURE 25: ATTITUDE CONTROL NOT WORKING	37

FIGURE 26: ATTITUDE CONTROL WORKING	37
FIGURE 27: ATTITUDE ERROR ESTIMATOR EXAMPLE	41
FIGURE 28: FIELD OF VIEW ⁸	42
FIGURE 29: ATTITUDE ERROR ESTIMATOR GEOMETRY FOR HORIZONTAL CAMERA AXIS.....	42
FIGURE 30: UNFILTERED ANGULAR VELOCITY (TIME).....	45
FIGURE 31: FILTERED ANGULAR VELOCITY (TIME).....	45
FIGURE 32: UNFILTERED ANGULAR VELOCITY (FREQUENCY)	45
FIGURE 33: FILTERED ANGULAR VELOCITY (FREQUENCY).....	45
FIGURE 34: INITIAL ATTITUDE REGULATION CAMERA SIMULATION VIEW	47
FIGURE 35: FINAL ATTITUDE REGULATION CAMERA SIMULATION VIEW	47
FIGURE 36: RELATIVE ATTITUDE CONTROL REGULATION ERROR	48
FIGURE 37: RELATIVE ATTITUDE CONTROL REGULATION TORQUE RESPONSE.....	48
FIGURE 38: ATTITUDE TRACKING WITH PURELY RELATIVE CONTROL	49
FIGURE 39: ZOOMED VIEW OF ATTITUDE TRACKING WITH PURELY RELATIVE CONTROL.....	49
FIGURE 40: BODY/RELATIVE ATTITUDE CONTROL REGULATION ERROR	51
FIGURE 41: BODY/RELATIVE ATTITUDE CONTROL REGULATION TORQUE RESPONSE	51
FIGURE 42: ATTITUDE TRACKING WITH RELATIVE/BODY CONTROL	52
FIGURE 43: ZOOMED VIEW OF ATTITUDE TRACKING WITH RELATIVE/BODY CONTROL.....	52
FIGURE 44: ATTITUDE SIMULATION MODULE SETUP	55
FIGURE 45: COMPARISON OF VARIANCE ³	61
FIGURE 46: SKEWNESS FIGURE FROM MVPSTATS ⁴	62
FIGURE 47: RECTANGLE	65
FIGURE 48: RECTANGLE M_{x3}	65
FIGURE 49: RECTANGLE G_{x3}	65
FIGURE 50: STRAIGHT ON G_{x3}	66
FIGURE 51: TRAPEZOID – SYMMETRIC ABOUT ROTATING AXIS.....	67
FIGURE 52: TRAPEZOID SYMMETRIC ABOUT ROTATION AXIS M_{x3}	67

FIGURE 53: TRAPEZOID SYMMETRIC ABOUT ROTATION AXIS G_{x3}	67
FIGURE 54: TRAPEZOID SYMMETRIC ABOUT ROTATION AXIS STRAIGHT ON G_{x3}	68
FIGURE 55: NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN NEGATIVE AXIS	68
FIGURE 56: NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN NEGATIVE AXIS M_{x3}	69
FIGURE 57: NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN NEGATIVE AXIS G_{x3}	69
FIGURE 58: POSITIVELY SKEWED DISTRIBUTION ⁴	70
FIGURE 59: NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN NEGATIVE AXIS STRAIGHT ON G_{x3}	71
FIGURE 60: TRAPEZOID – NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN POSITIVE AXIS	72
FIGURE 61: TRAPEZOID – NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN POSITIVE AXIS M_{x3}	72
FIGURE 62: TRAPEZOID – NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN POSITIVE AXIS G_{x3}	72
FIGURE 63: TRAPEZOID – NOT SYMMETRIC ABOUT ROTATING AXIS – MORE MASS IN POSITIVE AXIS STRAIGHT ON G_{x3}	72
FIGURE 64: SIMULATION INITIAL VIEW	74
FIGURE 65: SIMULATION INTERMEDIATE VIEW	74
FIGURE 66: SIMULATION FINAL VIEW	74
FIGURE 67: Z-DIRECTION SKEWNESS OF TWO-AXIS TEST	75
FIGURE 68: Y-DIRECTION SKEWNESS OF TWO-AXIS TEST	75
FIGURE 69: SIMULATED TARGET ATTITUDE	75
FIGURE 70: SIMULATION INITIAL VIEW	76
FIGURE 71: SIMULATION INTERMEDIATE VIEW	76
FIGURE 72: SIMULATION FINAL VIEW	76
FIGURE 73: Z-DIRECTION SKEWNESS OF TWO-AXIS TEST	76
FIGURE 74: Y-DIRECTION SKEWNESS OF TWO-AXIS TEST	76
FIGURE 75: SIMULATED TARGET ATTITUDE	76
FIGURE 76: CIRCLE VIEWED FROM A PERPENDICULAR ORIENTATION.....	77
FIGURE 77: CIRCLE VIEWED FROM A SKEWED ORIENTATION LOOKS LIKE AN ELLIPSE	77
FIGURE 78: CAMERA FRAME.....	79
FIGURE 79: PIN-HOLE CAMERA MODEL ⁶	83

FIGURE 80: PIN-HOLE CAMERA MODEL VISUALIZATION ⁷	84
FIGURE 81: CONTROL FLOW CHART	91
FIGURE 82: SIMPLE SIMULATION SKEWNESS VIEW	92
FIGURE 83: INITIAL INERTIAL VIEW	93
FIGURE 84: FINAL INERTIAL VIEW	93
FIGURE 85: INITIAL CAMERA VIEW	93
FIGURE 86: FINAL CAMERA VIEW	93
FIGURE 87: SKEWNESS CONTROL AROUND DEADBAND	97
FIGURE 88: RELATIVE ATTITUDE AROUND DEADBAND	98
FIGURE 89: DISTANCE BETWEEN BODY AND TARGET [M]	98
FIGURE 90: UNFILTERED INERTIA (2 ND MOMENT)	98

Chapter 1 Introduction

1.1 Motivation

The motivation for this thesis is to improve relative control techniques, specifically passive visual relative control of free-flying spacecraft. Visual control means camera data will be used to track a target. *Passive* visual control means inertial information from the target will not be transmitted to the spacecraft that is tracking the target with a camera.

A two-year-old toddler is fairly proficient at passive visual relative control. A two-year-old can easily watch a person walk around a room with her eyes or watch a cookie in her parent's hand get closer and control her hand to rendezvous with the cookie. If a toddler can perform more complex tasks with her eyes, and her eyes are basically providing streaming camera data to her brain, then we can perform a few simpler tasks with similar data. The data are there. We just have to learn how to make sense of it like our brains so easily do. We know that the task is possible—another working system, the human brain, takes the same type of input data and has a working control, so it is possible to create another system that takes similar input data and completes the same tasks.

Consider a robotic assistant satellite outside of the International Space Station (ISS) with an objective of observing a specific panel. For the robotic assistant satellite to complete its objective, it must have an algorithm to identify and track the panel or an object on the panel. Therefore, a visual tracking algorithm must be developed. This visual tracking algorithm must be able to differentiate between the object of interest in the image and the background scenery.

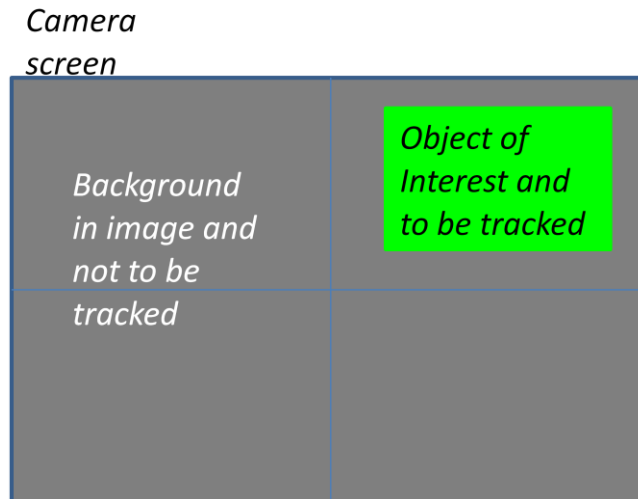


Figure 1: Basic Objective of Visual Tracking Algorithm

The next step in the robotic assistant satellite achieving its objective is to have an algorithm to continue pointing at the panel of the ISS. One method of completing this task is to have a target or object on the ISS panel for the robotic assistant satellite to point the camera at. If the objective is to observe a specific panel and the panel is not in the field of view of the camera, the mission is a failure.

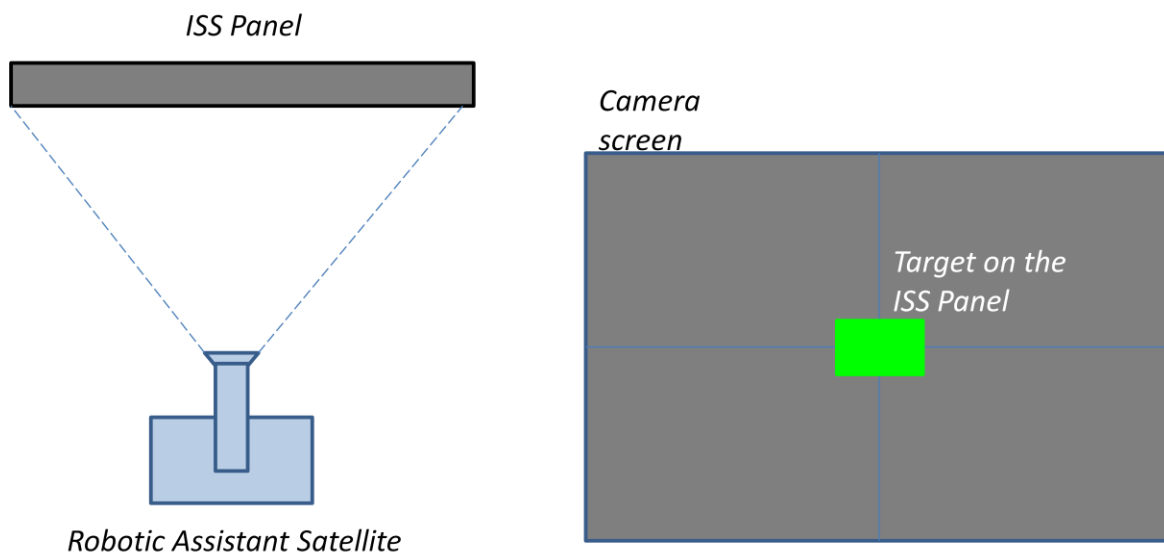


Figure 2: Example of Maintaining Pointing with Visual Data

Another part in achieving the mission objective is to have an algorithm that can recognize if the robotic assistant satellite is viewing the panel from a skewed or angled orientation. This will cause visual distortions in the recorded images.

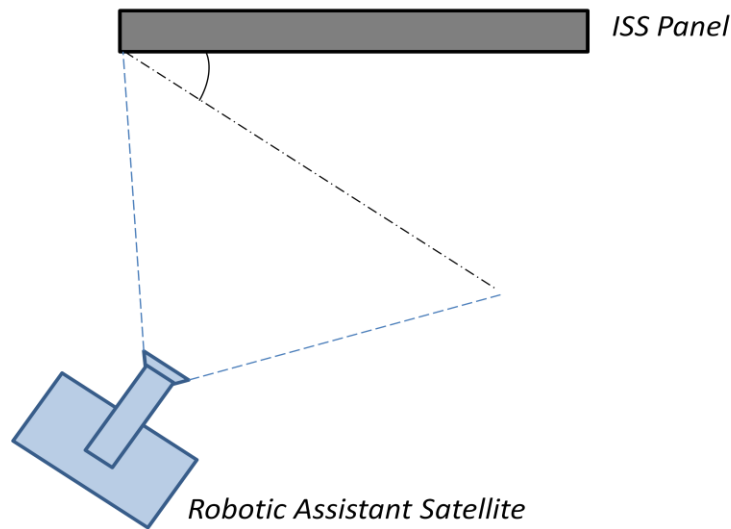


Figure 3: Top View of Skewed Orientation

A final aspect of achieving the mission objective is an algorithm to recognize and maintain the nominal distance between the robotic assistant satellite and the ISS. If the robotic assistant satellite were to collide with the ISS, the result would be space debris in the ISS orbit and possible damage to the ISS. If the robotic assistant satellite were to drift too far away from the ISS, it would lose visual tracking and possibly not be able to be re-docked with the ISS.

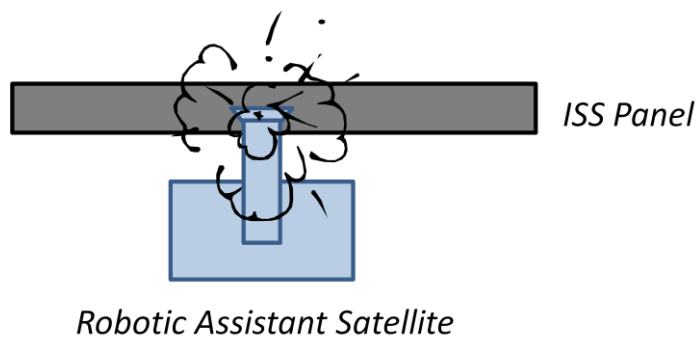


Figure 4: Collision after Failure to Maintain Nominal Distance

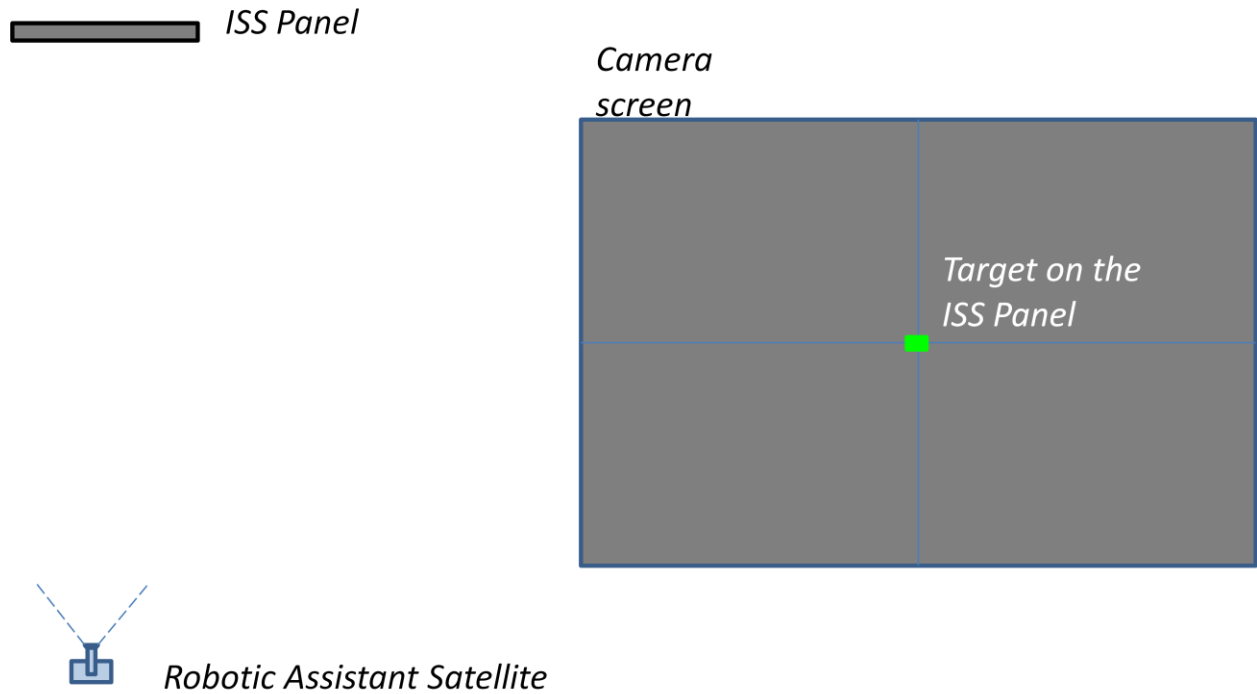


Figure 5: Losing Sight of the Target after Failure to Maintain Nominal Distance

1.1.1 Passive vs. Active

Consider a robotic assistant satellite outside of the International Space Station with an objective of observing a specific panel as in the previous section.

With an active control, the ISS would be required to constantly send position, velocity, attitude, and angular velocity data to the robotic assistant satellite, and mission operations personnel on earth would be involved. Extra equipment besides the camera must be built into the robotic assistant satellite, making it much more expensive.

With a passive control, no ISS information is transferred to the robotic assistant satellite's control. The robotic assistant satellite could be released from the ISS, use its onboard equipment (camera) to track the panel using visual data, store the data, and return to the ISS. No equipment is needed besides the camera, and no mission operations personnel are required to send data to the satellite.

Why are passive relative control techniques important? There are numerous reasons. It is very expensive, complicated, and limiting to be in constant, real-time communication with a target satellite in order to track it. A target satellite would have to incorporate being tracked into its mission by being launched with additional equipment, which is very expensive, and would have to budget the mission operations personnel into the lifetime of its mission to send commands and monitor the target satellite while it is being tracked, which is also expensive. Mission operations personnel would require real-time information of both the target and body satellites' positions and velocities to perform attitude maneuvers to line up the target's antenna with the line-of-sight between the target and body, verify none of the satellites are in safe mode, confirm that no solar flares or atmospheric events have wreaked havoc with the data, and then let the control on the body satellite perform its maneuver. Not only can a lot more go wrong, but this severely limits the tracking algorithm. Only objects that can "talk back" to the body satellite can be tracked. An asteroid, a piece of space debris, or a satellite not specifically launched with specialized equipment could not be tracked.

Passive relative control is another way of saying the target is non-participatory, meaning any object should be able to be tracked.

1.1.2 Inertial vs. Relative

Consider an example of pointing a satellite to track an asteroid. In the inertial control figure below, the ground station first calculates a reference inertial pointing direction from the ground station based on ascension and declination angles. This is generally calculated as a unit direction vector when working with astronomical objects. The ground station then has to calculate the inertial position of the satellite with respect to the ground station. This is a statistical orbit determination problem. At each time step, the direction of the ground station to target and ground station to satellite position vectors can be subtracted and normalized to obtain the relative unit direction vector between the satellite and target. If the camera data is being used for navigation, the satellite does not process the images itself and calculate a control. Instead, the satellite sends the images through space and the atmosphere to the ground station in order for

the ground station to process the images, incorporate optical data into the statistical orbit determination solution, calculate a control, and then send the control up to the satellite.

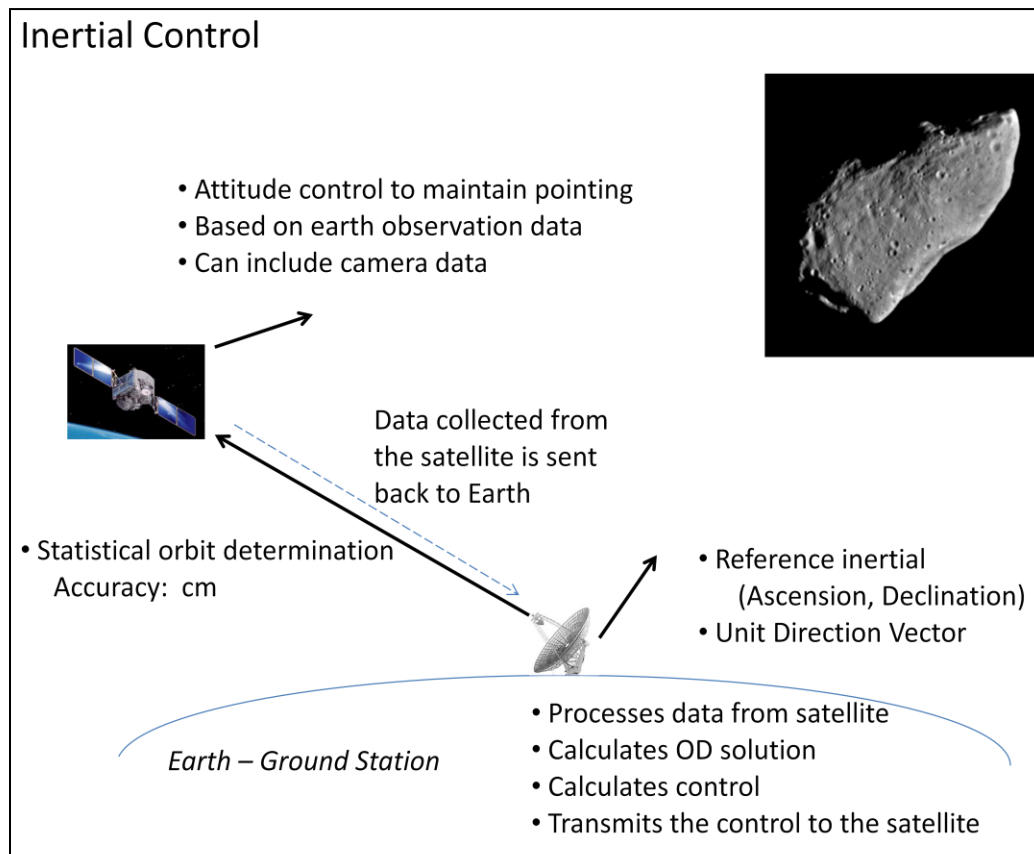


Figure 6: Inertial Control^{11, 18, 19}

In the relative control figure below, the unit direction vector from the ground to the target is known at the initial time because a visual target is not chosen without first knowing where that target is. (A visual target is usually not chosen if it has not already been seen.) The satellite's position must be known at the initial time. These direction vectors can be subtracted at the initial time to give the camera a pointing direction to get the target in its field of view. Once the camera's control starts to track the target, no more data is sent to the satellite. A main advantage of this approach is the satellite is processing its own data and calculating its own control responses. It does not need to wait for the ground station on earth to receive, process, calculate, and transmit the data and responses.

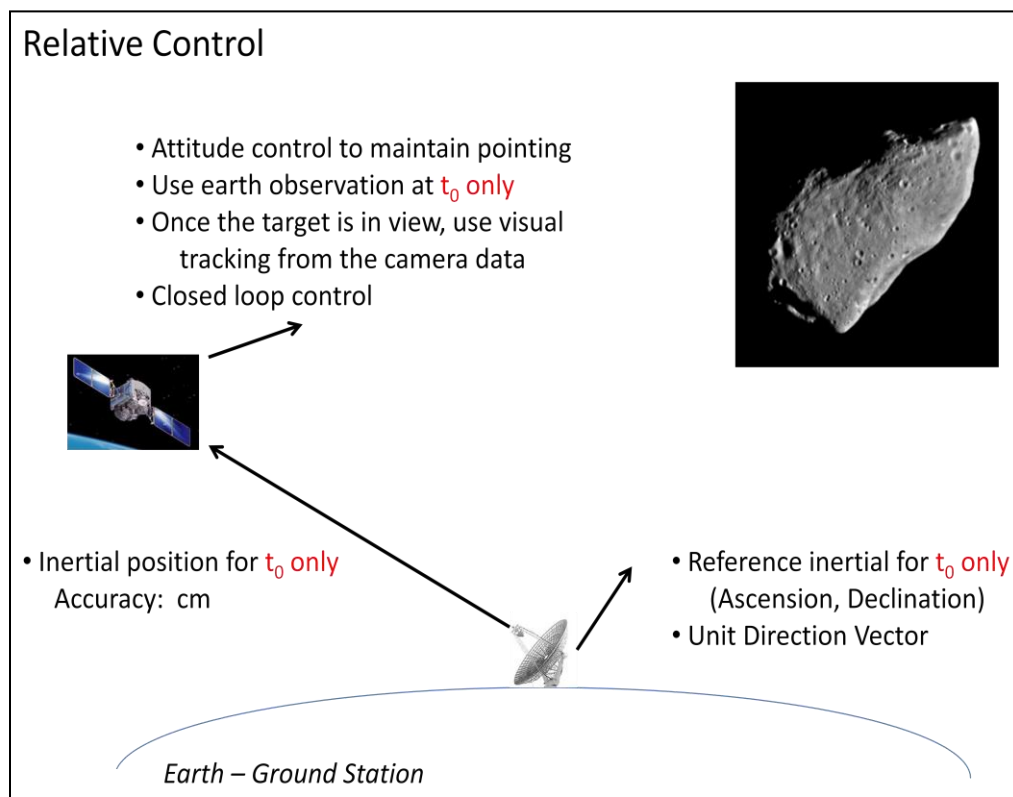


Figure 7: Relative Control^{11, 18, 19}

Several missions discussed in the literature review, especially the Galileo Gaspra encounter, would have benefited greatly from using a relative control approach instead of sending data back to Earth for processing. Technological concerns of twenty years ago are probably what made the mission design decisions, but if those missions were repeated today, different methods should be considered.

1.1.3 Overview

The figure below illustrates the data transfer when a satellite docks with the ISS.

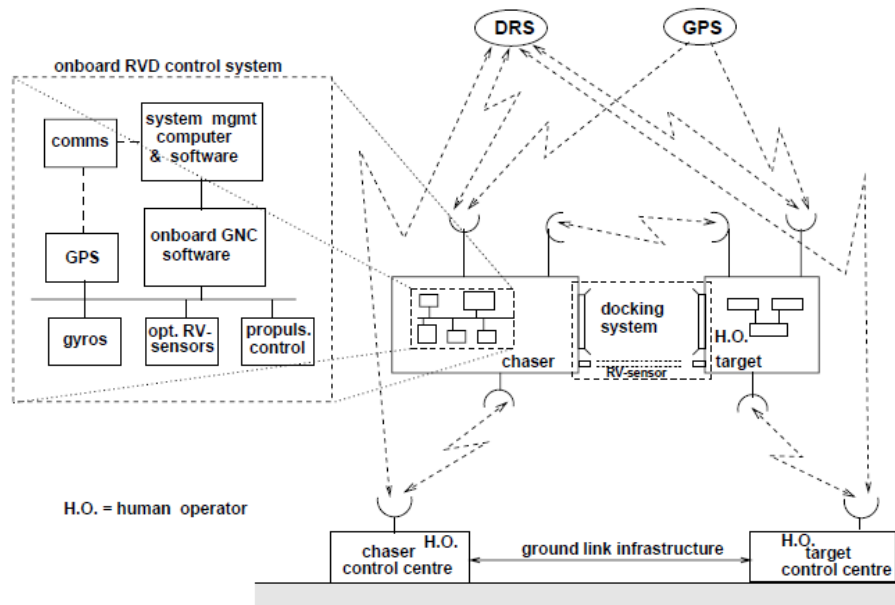


Figure 8: Docking with the ISS¹⁶

The chaser (body) satellite receives information from the chaser ground station, GPS, ISS, and DRS (data relay satellite). The ISS receives information from the ISS ground station, GPS, the chaser satellite, and DRS. The two ground stations must communicate with each other, and both must separately communicate with the DRS.

This means that the chaser's control is an active, inertial control. Everything in the previous sections concluded a passive, relative control is a desired solution for numerous reasons. If one of numerous parts on the chaser, chaser ground station, target, target ground station, GPS, or DRS is not working perfectly, then the maneuver cannot occur. Instrument failures and misalignments happen. Satellites go into safe mode. GPS measurements become flawed. Antennas do not deploy. Data becomes corrupt or lost. As a system is made more complex with more instruments that must work perfectly at synchronized times, and more data that must be transferred through the atmosphere, the less likely it is that this method of active, inertial control will work. The research in passive, relative control is important because it lets the chaser satellite rely on its own data, process its own control, and react instead of relying on external data and an incredibly complex system that is bound to fail.

1.1 Literature Review

1.1.1 Voyager Uranus Encounter¹⁴

The two Voyager spacecraft were initially supposed to study Jupiter, Saturn, and their moons. The satellites were still operational, NASA continued their funding, and Voyager 2 was sent on to explore Uranus and Neptune. Now, both Voyager 1 and 2 are well beyond Pluto¹³. Ten new Uranian satellites or moons were found using visual data from Voyager 2.

Approximately 225 pictures were taken in the 75 days leading up to the encounter with Uranus. The images were scheduled so ideally two or three stars would be in the background of Uranus in each picture. The ascension and declination of the surrounding stars down to a magnitude of 10.0 were required, so a star catalog of the surrounding area of space was developed for this encounter.

Once the image is on the ground at Earth, a technician visually finds the center of each object. The centers of each object are then calculated using box filtering, which is basically a version of the batch filter that uses the technician's visual estimations of the center of each object as the *a priori* information for the filter.

In a filtering method like the batch processor, if the *a priori* information is too inaccurate the solution will not converge. Taking the *a priori* information from a human approximation instead of a calculation is not recommended. The researchers and mission operations personnel obviously made this method work and produced accurate results, but algorithms exist that take the human "eye-ball" estimation out of the process and replace it with machine calculation. Also, if a computer can calculate the solution, the images do not have to be sent back to Earth for a solution to be generated. Computer memory was more of an issue than transferring data across a solar system twenty years ago. Now the technological situation is not the same, so the development, application, and usage of algorithms and control techniques do not need to reflect the issues of the past.

1.1.2 Galileo Gaspra Encounter¹²

The spacecraft Galileo was scheduled for the first interplanetary flyby of an asteroid in 1991. As often happens, something unexpected occurred. Galileo's high-gain antenna did not deploy, which meant that only four pictures of Gaspra, the asteroid, were sent back to Earth before the closest approach to the asteroid. Using a low-gain antenna onboard Galileo instead of the high-gain antenna meant the difference between receiving the image data almost instantly and having to wait more than seventy hours for a single image. The optical navigation techniques previously developed required more than four pictures, so researchers at the Jet Propulsion Laboratory came up with a new algorithm called the single-frame mosaic technique.



Figure 9: Gaspra from Galileo¹¹

1.1.2.1 *Batch Filtering (Voyager) Method*

The plan was to receive at least one hundred pictures in the sixty days leading up to the Gaspra encounter. The same image analysis and optical navigation techniques used for Voyager would have been reused and employed for the Galileo Gaspra encounter. Initially, Gaspra would have been far enough away from the satellite to be treated as a point source similar to the stars in the background. As Galileo approached throughout the passing days, Gaspra would appear larger and three-dimensional information such as

shape and orientation could have been calculated. The same filtering method would have been utilized as Voyager.

1.1.2.2 Single-Frame Mosaic

The basis of the single-frame mosaic method was to leave the camera shutter open and get a time history of Gaspra in each picture instead of using the “point and click” method that takes a single instant in time. Also, while the camera shutter was open, the camera platform was slewed around to different positions. Slewing the camera around decreased the sensitivity to data loss and allowed multiple exposure times to be used in a single picture—one could benefit Gaspra observations more, the other could benefit the background star for orbit determination more. Also, slewing the camera around could let a single picture view a larger amount of space than the field of view of the camera allows.

1.1.2.3 Summary

The single-frame mosaic technique is ingenious and the researchers who came up with it had very little time to develop and implement the idea. Both the single-frame mosaic and traditional techniques depend on data transfer and analysis of the images on Earth. If the satellite could process the images and feed the results into its orbit determination solution, then this issue would not have occurred in the first place.

1.1.3 Configuring the Deep Impact AutoNav System for Lunar, Comet, and Mars

Landings¹⁰

The optical navigation system developed by the Jet Propulsion Laboratory is referred to as AutoNav. Any missions that obtain close up, never before seen images of comets like the recent EPOXI or NEXt missions, use the AutoNav system.



Figure 10: Tempel-1 Comet Viewed during the NExT Mission

The AutoNav system is planned to be expanded to do more than maneuver around comets using optical data—the task of the optical navigation department at JPL is to use AutoNav to precisely land on the Moon, a comet, and Mars. AutoNav uses a system for surface modeling and landmark tracking to estimate the spacecraft position and attitude. The proposed landing system has meter-level accuracy.

During the landing maneuver, visual data is taken from the cameras. The new AutoNav system does not depend on gyroscope or accelerometer data. Before the maneuver, the AutoNav system must have accurate surface images to compare the control to. The AutoNav control is only as accurate as the pre-flight information and the in-flight attitude information taken from the camera data. The figure below shows an example of landmarks in an image.

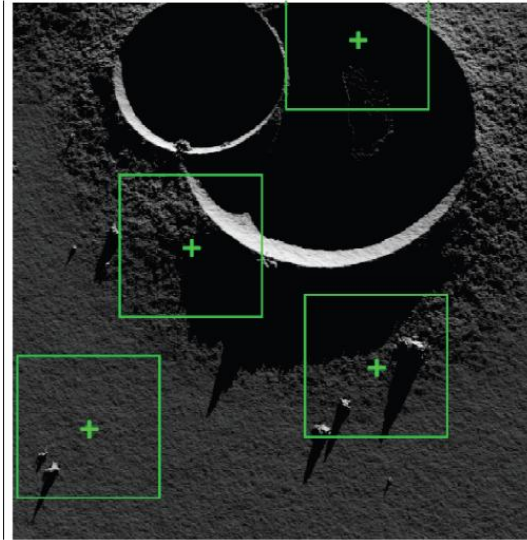


Figure 11: Landing Optical Navigation Landmarks

Using landmarks for landing on something as large as the moon sounds like a decent method because it is unlikely that a landmark of that size would move between the pre-flight surface mapping and the in-flight maneuver. Transferring this method to docking between satellites could be problematic since the landmarks to track on the target satellite would be much smaller and subject to small movement variations.

1.1.4 Autonomous Helicopter using Feature Tracking¹⁵

Researchers combined visual sensing of urban features like windows with GPS (Global Positioning System) and IMU (Inertial Measurement Unit) data to track objects. The helicopter is initialized in a hover mode and a user selects a window for it to track. It tracks the window through successive frames.



Figure 12: Helicopter during a Test

A lot of inertial information is used in this application. However, a technique of identifying windows in an image was explored.

First, the color of the target (window) is determined after the user clicks on the target in the image. Then the image is converted to grayscale by OpenCV software. This is done for processing speed. The pixels of the image with the right color are given a value of '1' while the other pixels are given a value of '0', turning the image into a binary image. Some of the objects in the background might have a similar color as the target and will also have a '0' in the binary image, so a square finding algorithm is used to find the windows. Then, template matching is employed using the user's input of where to look for the target. Once the object is found, tracking can begin. The output to the control is velocity.

This visual sensing technique is sensitive to lighting conditions, and other environment factors were not thoroughly tested. It assumes a square is being tracked, and factors such as the target being partially obscured or viewed from a skewness angle so the square began to look like a rectangle were not pursued. It is another method for visual sensing and target tracking, but is also very limiting in terms of what targets can be tracked under what conditions.

1.1.5 STARDUST Wild 2 Encounter¹⁷

The STARDUST Wild 2 mission collected dust samples from the comet Wild 2 in early 2004. The navigation used a combination of optical and radio techniques.

Images are taken of the comet with stars in the background. The stars are considered inertial. The center of the comet is calculated. The radio and image data are combined to calculate the orbit determination solution. The “centerfinding” uses a moment calculation, but it skips the visual tracking algorithm step and has numerous admitted difficulties and limitations, such as lighting, geometry, noise, etc.

1.1.6 Moment Functions in Image Analysis: Theory and Applications¹

The fields of computer vision and robotics use image analysis techniques for many applications. Moment functions can be used for automatic character recognition, aircraft identification, pattern matching, “invariant pattern recognition, object classification, pose estimation, image coding, and reconstruction¹.”

Mukundan and Ramakrishnan use moment functions to describe geometric features of an image. This thesis differs because something is assumed about the target and then the moment functions are used to gain information about the relative position and orientation of the body satellite with respect to the target.

Mukundan also draws the relationship between the zeroth, first and second moments in statistics and mass. His explanation is summed up in the following table.

Table 1: Mukundan’s Comparison of Moments Across Fields

Moment	Statistics	Mass	Images
0th	Total Probability	Total Mass	Total Area
1st	Expectation (Mean)	Center of Mass	Central Coordinate
2nd	Variance	Inertia	Orientation

The basis of Mukundan’s image analysis with geometric moments is to consider “an image as a two-dimensional intensity distribution¹.” The results of the image’s geometric moments are the total image area, the central pixel coordinate of the image, and the orientation. Mukundan calls the second image moment the orientation, and while it can be used to calculate the orientation of the image within the plane of the camera’s screen, it would be more accurate to call it the inertia of the image. The length of the target in pixels can also be calculated from the second image moment if something is known about the target or image shape.

Besides geometric moments, there are also orthogonal, Legendre, Zernike, and complex moments covered by Mukundan that may be of use to future researchers on this project.

Mukundan used applied moments for pattern recognition, object identification, and attitude and position estimation. He used two methods of attitude and pose estimation: 1) using a pattern on a plane of an object, or 2) using multiple views (cameras) of the object to estimate a particular view direction.

The first method is closest to the desired developments of this thesis, but Mukundan's method requires a pattern to be in constant view of the camera, and anything tracked must be man-made. Also, this method is very numerically intensive and not based on skewness. Another note is it depends on an assumption that the initial distance from the camera to the target is known—something that is not known in any of the proposed simulations.

The second method proposed multiple cameras viewing the target from different positions, compiling all of the multiple images into a library in real time, and calculating relative orientation from the compiled information. This posed many practical issues—it is expensive enough to launch one robotic assistant satellite into space. If a control requires multiple satellites to be operational and communicate with each other in real time in order to complete a mission, then the control option becomes less realistic.

Mukundan saw the parallels between how geometric moments are used for mass, statistics, and images, and even recognized that skewness in an image represents a deviation from symmetry. He did not take that a step further to apply skewness as a way to determine relative orientation.

1.1.7 Statistical Moments, Jamie Shutler⁵

Shutler introduces how researchers in image analysis use the third moment to calculate skewness, which is used as a measure of symmetry. The direction of the asymmetry is determined by looking at the sign of the skewness calculation. Being able to determine the direction of the asymmetry is very important for control applications because if an algorithm can detect that an image is skewed but has no way of

determining in which direction it is skewed, the control will not know in which direction to apply a restoring force to fix the problem.

Some researchers in image analysis are concerned with letter recognition. Using the third moment and skewness helps with letter recognition since skewness is a measure of symmetry about an axis.

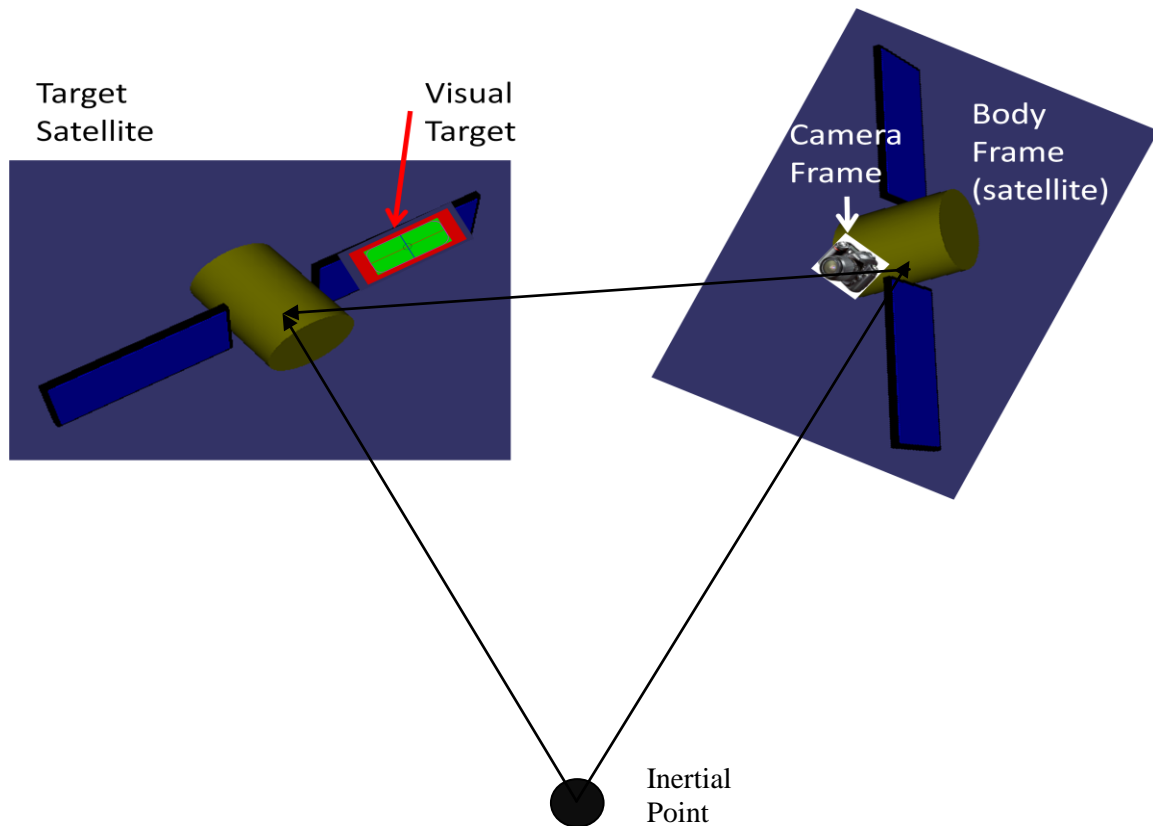


Figure 13: Letter Recognition⁵

The letter “M” is symmetrical about the vertical axis, but not the horizontal axis. This helps identify it is the letter “M”. The letter “C” is symmetrical about the horizontal axis, but not the vertical axis, and this helps identify it is a “C”. This is found by calculating the third moment for each letter in each axis.

1.2 Overview/Scope

The setup of the problem includes two bodies, generally either two satellites or two ground rovers. The satellite with the camera attached is called the *body* satellite; the one that the body is trying to track with a camera is called the *target* satellite.



The target should be tracked based off of information the camera collects. This makes for a relative control because all of the data a camera can collect are relative to the two satellites and not inertial.

The main problems that have to be solved are:

- 1) How can a camera track a target so the center of the target stays at the center of the camera screen?
- 2) How can a camera/computer tell if the target is tilted away from the camera, meaning the camera is not looking at it “straight on”?
- 3) How can the camera/computer figure out distance or changes in distance when only using one camera?

There are three different controls that should work in concert for the simulation to function. The attitude control keeps the center of the target in the center of the camera’s viewing frame. The skewness control

keeps the body satellite “straight on”, or perpendicular to, the target. This is especially important for rendezvous and docking maneuvers. The position control keeps the target at a fixed distance from the body satellite. The three controls are how to put the solutions to the three main problems solved in this thesis into action.

Another way to look at all three controls is to imagine the target satellite at the center of a sphere and the body satellite at the edge of the sphere. The attitude control keeps the body satellite always pointing at the center of the sphere (at the target). The skewness control moves the body satellite along the edge of the sphere until the body is looking at the target from a “straight on” view. The distance control keeps the radius of the sphere constant, as seen in the following figure.

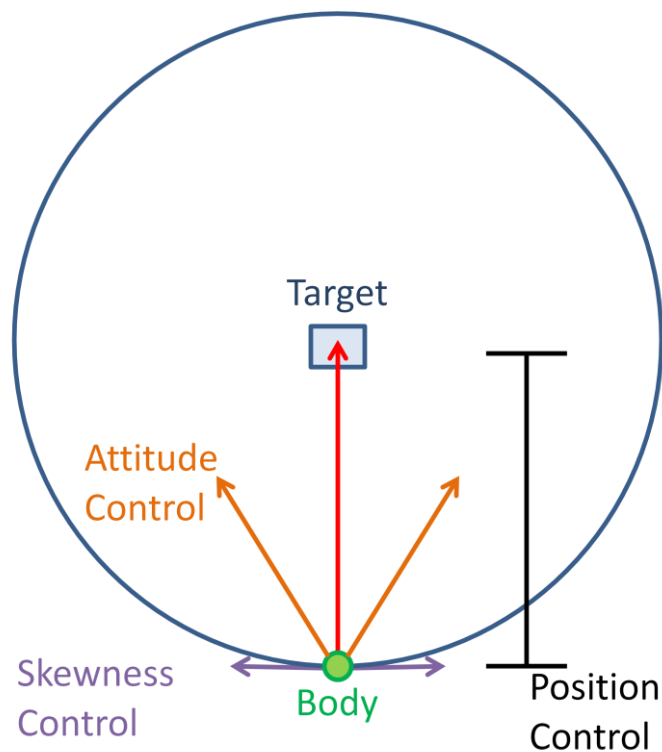


Figure 14: Overview of Controls

1.2.1 Attitude Control

Attitude is a measure of pointing. In terms of a relative control, the center of the camera should be pointing at the center of the target. The attitude control is arguably the most important control, because if

it does not work, the camera will lose sight of the target. Imagine if there is a science mission that is supposed to track an asteroid and the camera cannot even see the asteroid because the body satellite is pointed in the wrong direction. That would lead to an instant mission failure.

The following two figures show the camera frame viewing a target. On the right, the attitude control is working because the center of the camera is pointing at the center of the target. On the left, the attitude control is not functioning correctly because the center of the target is not at the center of the camera frame.



Figure 15: Attitude Control Not Working



Figure 16: Attitude Control Working

1.2.2 Skewness Control

For a skewness control, it is important to get a measure of perpendicularity relative to an object. For maneuvers such as automated rendezvous and docking, it is very important to know how “straight on” or perpendicular the body is to the target.

The following two figures show the camera frame viewing a target. In the right figure, the skewness control is working because the camera is viewing the target from “straight on”. On the left, the skewness control is not working, because the camera is viewing the target from a skewed angle.

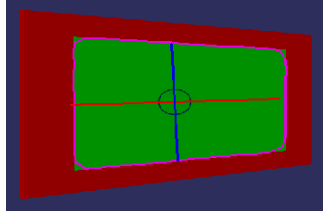


Figure 17: Skewness Control Not Working

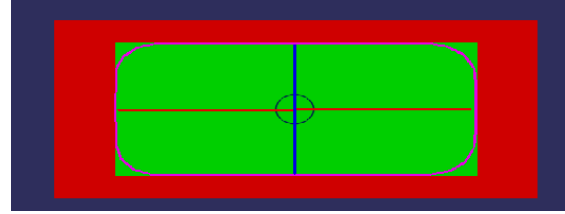


Figure 18: Skewness Control Working

1.2.3 Distance Control

A position control has the job of keeping the body a certain distance away from the target. This is important because you do not want the body drifting too far away from the target, but you also do not want the body colliding with the target. There may be rendezvous maneuvers that are closely controlled, but for the most part, a specified distance is desired for observation and tracking. If the body gets too far away there will not be accurate tracking, the other controls will not work accurately, and tracking will eventually be lost.

The following two figures show the camera frame viewing a target. In the right figure, the distance control is working because the body is keeping a correct, specified distance from the target. On the left, the distance control is not working because the body is not keeping a specified distance from the target. In this case, the body has drifted too far away from the target so the target appears smaller and shows up as fewer pixels in each camera frame.

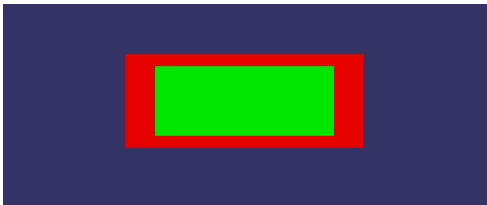


Figure 19: Distance Control Not Working



Figure 20: Distance Control Working

Chapter 2 Project Background

2.1 UMBRA

The software used to develop this project is a simulation framework created by Sandia National Laboratories named UMBRA. UMBRA uses a combination of C++, Tcl/Tk, and OpenGL graphics to quickly and easily simulate and visualize complex systems. In UMBRA, each functional component becomes its own module. Modules can have multiple input and output connectors.

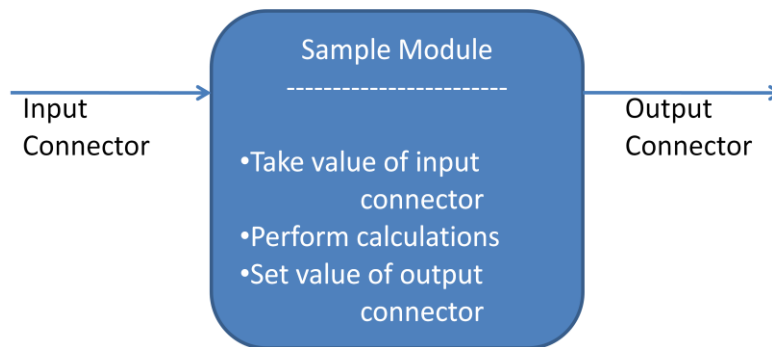


Figure 21: UMBRA Sample Module

Data flows through the input and output connectors at each time step—like electronics, the data flows through the connectors like electrons through wires. To construct a simulation, connect instances of different modules together like LEGOs® or wires connecting different parts of a circuit. Each time step (or update loop, as UMBRA calls it), Module 1 is called, its calculations are completed, the data leaves through an output connector which is connected to Module 2's input connector, Module 2's calculations are completed, data is sent from its output connectors to more modules, and so on.

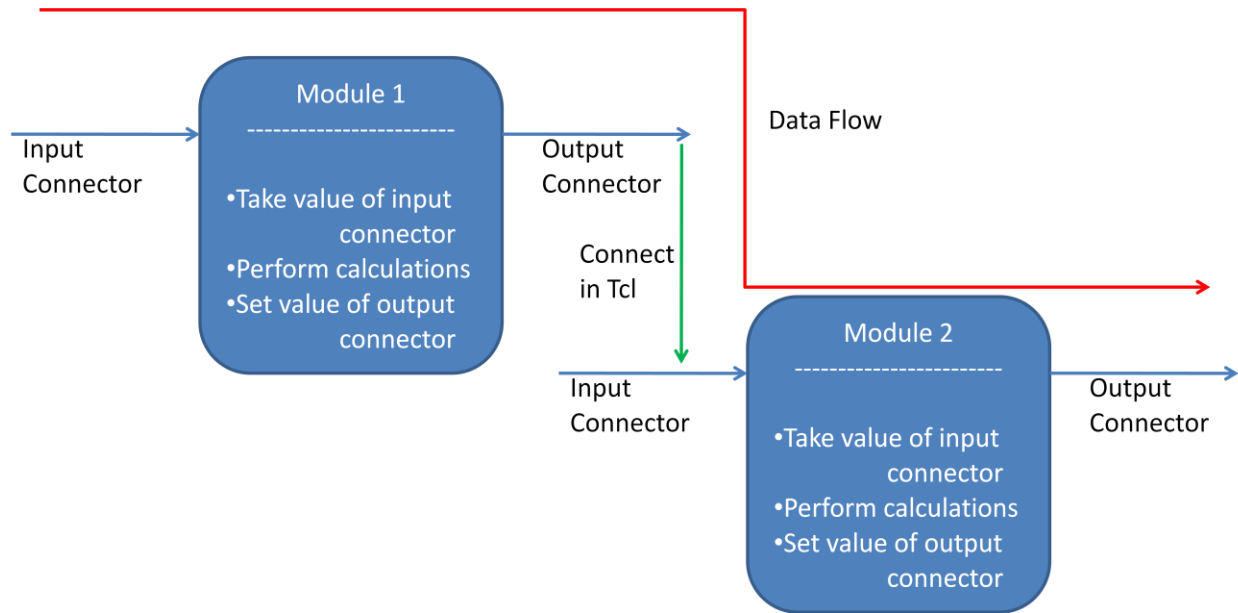


Figure 22: UMBRA Sample Data Flow

All of the modules are created and compiled in C++, but connected together to form a simulation in Tcl. Creating the simulation in Tcl makes UMBRA incredibly flexible because modules can be exchanged without recompiling any code. For example, if a virtual camera is needed in one simulation, connect the virtual camera module. If a real camera is then needed, disconnect the input and output connectors tied to the virtual camera module in the Tcl script, connect them to the real camera module instead, and now real camera data would flow into the simulation and no code would need to be recompiled to make this change. Interchanging a virtual or real robot that moves around the lab follows the exact same principle—they are both modules in UMBRA, so one is chosen when the simulation is created in Tcl and no code needs to be recompiled.

2.2 Data Flow

The data flow of the project comes from sequential camera images being taken. The flow of images is processed by a statistical pressure snake algorithm, which tracks a specific target in the image by hue.

From the statistical pressure snake algorithm, the output is a series of snake control points which make up the outside curve or “snake” of the statistical pressure snake as seen in Figure 24.

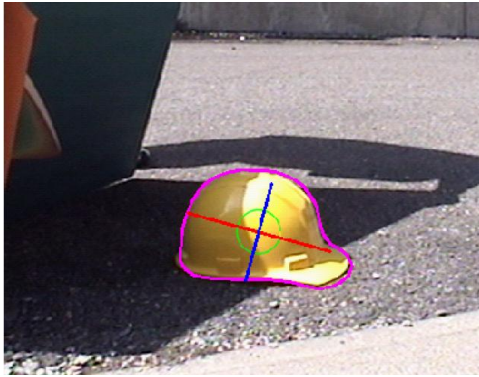


Figure 23: Statistical Pressure Snakes Tracking a Hard Hat⁶

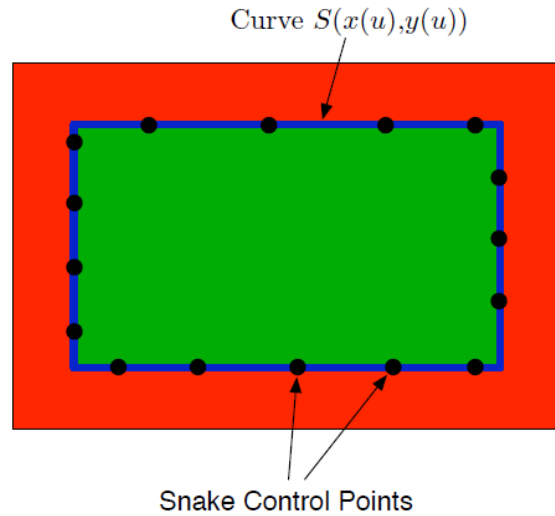


Figure 24: Snake Control Points⁶

Each snake control point has a corresponding (x,y) coordinate in the camera frame which can be used to calculate a moment of any order. In this project, the snake control points are sent to an OpenCV program that calculates the image moments. The statistical pressure snake algorithm works well under varying lighting conditions, can track any shape, and even works when part of the image is obscured.

At this point in the data flow, an attitude error estimator module takes the central coordinates (first moment) of the target and calculates the attitude error between the center of the target and camera. The attitude control determines how the camera must rotate for the center of the target to be in the center of the camera screen. The moment information is sent off to a skewness control law that calculates how the body must move to be perpendicular to target. Finally, information from the image about how many pixels the target takes up can be used to figure out a change in depth, which is sent to the distance control.

Chapter 3 Attitude Control

The objective of the attitude control is for the center of the camera to point at the center of the target. The attitude control is considered to be the most important component of visual control because if it fails the camera will not be able to “see” the target, so none of the other controls will be able to work. The figures below again show what it means for the attitude control to be working or not working. An attitude control would not be working in Figure 25 because the center of the target is not in the center of the camera screen. The attitude control is working in Figure 26 because the center of the target is in the center of the camera screen.

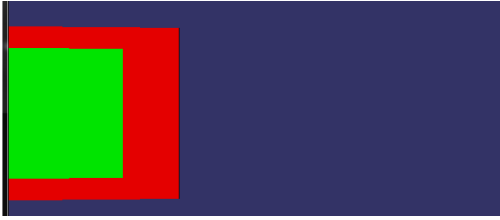


Figure 25: Attitude Control Not Working



Figure 26: Attitude Control Working

3.1 Attitude Control Development

The development of the attitude control starts with Euler’s rotational equations of motion. In order to make the equations less confusing, the variable names and descriptions are provided in the table below.

Table 2: Attitude Variables

$[I]$	Body inertia tensor (assume rigid body)
$\dot{\bar{\omega}}_{B/N}$	Body angular acceleration with respect to the inertial reference frame
$\bar{\omega}_{B/N}$	Body angular velocity with respect to the inertial reference frame
\bar{u}	Unconstrained external torque vector
\bar{L}	Known external torque acting on the body
$\bar{\sigma}_{B/R}$	Attitude error between body and target frames (Modified Rodrigues Parameter) (relative attitude)

$\bar{\omega}_{B/R}$ Angular velocity error between body and target frames (relative angular velocity)

$$\bar{\omega}_{B/R} = \bar{\omega}_{B/N} - \bar{\omega}_{R/N}$$

$\bar{\omega}_{R/N}$ Target angular velocity with respect to the inertial reference frame

$\dot{\bar{\omega}}_{R/N}$ Target angular acceleration with respect to the inertial reference frame

The derivation of the rotational attitude control begins with Euler's rotational equations of motion.

$$[I]\dot{\bar{\omega}}_{B/N} = -[\tilde{\omega}_{B/N}][I]\bar{\omega}_{B/N} + \bar{u} + \bar{L} \quad (3.1)$$

The tilde operator is matrix notation for the cross product².

$$[\tilde{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3.2)$$

The Lyapunov function given below is meant to drive both the relative attitude and relative angular velocity to zero.

$$V(\bar{\omega}_{B/R}, \bar{\sigma}_{B/R}) = \frac{1}{2} \omega_{B/R}^T [I] \bar{\omega}_{B/R} + 2K \ln(1 + \bar{\sigma}_{B/R}^T \bar{\sigma}_{B/R}) \quad (3.3)$$

The derivative of the Lyapunov function must be calculated to determine stability. Because the Lyapunov function is a scalar quantity, the derivative can be taken with respect to any reference frame. The body frame makes the calculation simple since the inertia tensor does not vary with respect to the body frame.

$$\dot{V}(\bar{\omega}_{B/R}, \bar{\sigma}_{B/R}) = \omega_{B/R}^T [I] {}^B \frac{d}{dt} (\bar{\omega}_{B/R}) + \omega_{B/R}^T K \bar{\sigma}_{B/R} \quad (3.4)$$

The relation for the body derivative of the relative angular velocity is given below.

$${}^B \frac{d}{dt} (\bar{\omega}_{B/R}) = \dot{\bar{\omega}}_{B/N} - \dot{\bar{\omega}}_{R/N} + [\tilde{\omega}_{B/N}] \bar{\omega}_{R/N} \quad (3.5)$$

Plug the relation for the body derivative of the relative angular velocity into the derivative of the Lyapunov function.

$$\dot{V}(\bar{\omega}_{B/R}, \bar{\sigma}_{B/R}) = \bar{\omega}_{B/R}^T \left[[I] \dot{\bar{\omega}}_{B/N} + [I]([\tilde{\omega}_{B/N}] \bar{\omega}_{R/N} - \dot{\bar{\omega}}_{R/N}) + K \bar{\sigma}_{B/R} \right] \quad (3.6)$$

Plug Euler's rotational equations of motion into the derivative of the Lyapunov function.

$$\dot{V}(\bar{\omega}_{B/R}, \bar{\sigma}_{B/R}) = \bar{\omega}_{B/R}^T [-[\tilde{\omega}_{B/N}][I] \bar{\omega}_{B/N} + \bar{u} + \bar{L} + [I]([\tilde{\omega}_{B/N}] \bar{\omega}_{R/N} - \dot{\bar{\omega}}_{R/N}) + K \bar{\sigma}_{B/R}] \quad (3.7)$$

Force the derivative of the Lyapunov function to be negative definite by setting it equal to a negative definite function.

$$\bar{\omega}_{B/R}^T [-[\tilde{\omega}_{B/N}][I] \bar{\omega}_{B/N} + \bar{u} + \bar{L} + [I]([\tilde{\omega}_{B/N}] \bar{\omega}_{R/N} - \dot{\bar{\omega}}_{R/N}) + K \bar{\sigma}_{B/R}] = -\bar{\omega}_{B/R}^T [P] \bar{\omega}_{B/R} \quad (3.8)$$

Solving for the control, the result is the same as the unconstrained control law in H. Schaub's Analytical Mechanics of Space Systems².

$$\bar{u} = -K \bar{\sigma}_{B/R} - [P] \bar{\omega}_{B/R} + [I](\dot{\bar{\omega}}_{R/N} - [\tilde{\omega}_{B/N}] \bar{\omega}_{R/N}) + [\tilde{\omega}_{B/N}][I] \bar{\omega}_{B/N} - \bar{L} \quad (3.9)$$

The equation above is the control that would be ideal for the system in a perfect world if all information were perfectly known. However; this simply is not the case. What information is known? How well can we get each of these values for a control? What issues are there? These are some questions the next section on sensing will delve into.

3.2 Sensing

The overall goal of this thesis is passive relative visual control. Stated simply, the goal is to use data from a camera to track objects. Similar to a toddler eagerly watching her parent's hand that has a cookie in it, we are giving a robot an eye (camera) and giving its brain (computer) algorithms to process the data so it can track an object for us. We want the robot to be able to track our cookies. Or point flashlights for astronauts. Or "watch" something to collect scientific data.

If the overall goal is to use that camera data for the control equation developed in the previous section, what part of the camera data gets matched up with what part of the control? How is it done? What are the other values in the control law and where do they come from? Can we even get some of those values?

Does the control law work if we cannot get some of those values? What if the values we get are faulty? Does the control law still work? Analyzing the terms in the full Lyapunov control equation will answer these questions.

3.2.1 Relative Motion

The relative attitude and angular velocity terms in the full Lyapunov control are $\bar{\sigma}_{B/R}$ and $\bar{\omega}_{B/R}$.

3.2.1.1 *Origin*

The relative attitude ($\bar{\sigma}_{B/R}$) ultimately is obtained from the camera data, but the calculation of the target center comes from the statistical pressure snake module, and the translation of the pixel offset into the relative attitude MRP comes from the attitude error estimator module.

The relative angular velocity ($\bar{\omega}_{B/R}$) is calculated from an equation that uses the numerical differentiation of the relative attitude (Eqn 3.17). Therefore, the relative angular velocity is dependent on a time history of sequential camera frames, but also on the statistical pressure snake and attitude error estimator modules.

The statistical pressure snake algorithm first finds the outside edge of the target. The information about the shape of the object is used to calculate the zeroth, first, second, and third moments. The first moment is the coordinate of the image center in pixels. Consider this as attitude error for the scope of this thesis. An explanation follows.

The goal of the attitude control is for the center of the camera screen to point at the center of the target. Imagine that the center of the target is off of the center of the camera's screen by 'x' pixels to the right and 'y' pixels up like Figure 27. A calculation must translate the 'x' pixels to the right into a desired yaw rotation and the 'y' pixel up into a desired pitch rotation. Then, a control must rotate the camera through the resulting yaw and pitch rotations so the camera will be pointing at the center of the target. Notice that using the center of the target yields a two-axis control since the camera will only rotate in two-axes (yaw

and pitch). This means the attitude control does not correct for any third-axis, or roll, rotations. Visually, this corresponds to ensuring the target is always in view but not correcting for any spin within the camera's view. The target could be spinning around its central pixel, and as long as the central pixel does not waver from the center of the camera screen, the two-axis attitude control will not notice any variation. The attitude control is developed using Modified Rodrigues Parameters, so the yaw and pitch rotations are translated into MRPs.

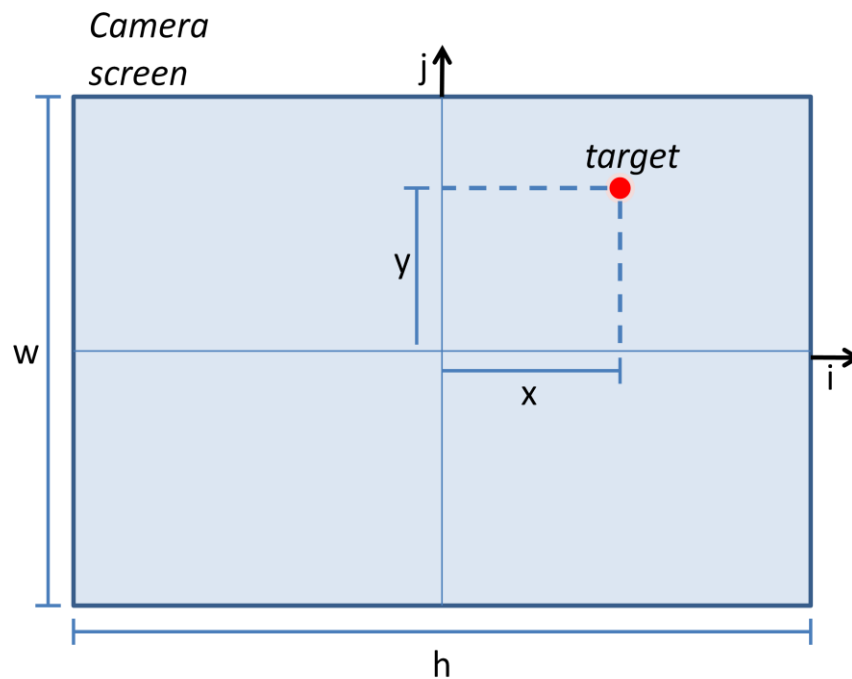


Figure 27: Attitude Error Estimator Example

The field of view of a camera shows the spread angle the camera can view in an image in each axis. The field of view is required to translate the pixel offset in each axis into the two-axis rotation the camera must undergo in order to view the center of the target at the center of the camera screen.

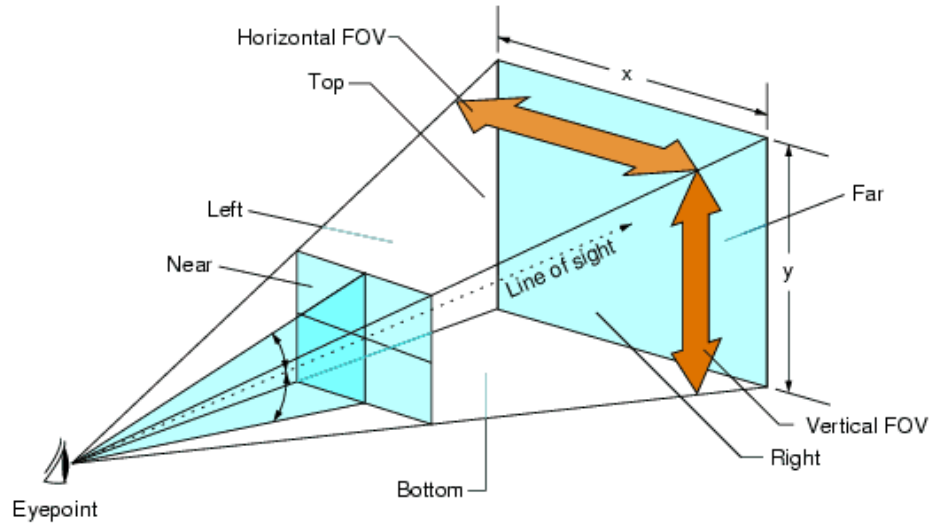


Figure 28: Field of View⁸

To translate the pixel offset into a rotation, the relation between the pixel and the angle must be calculated using the field of view in that axis.

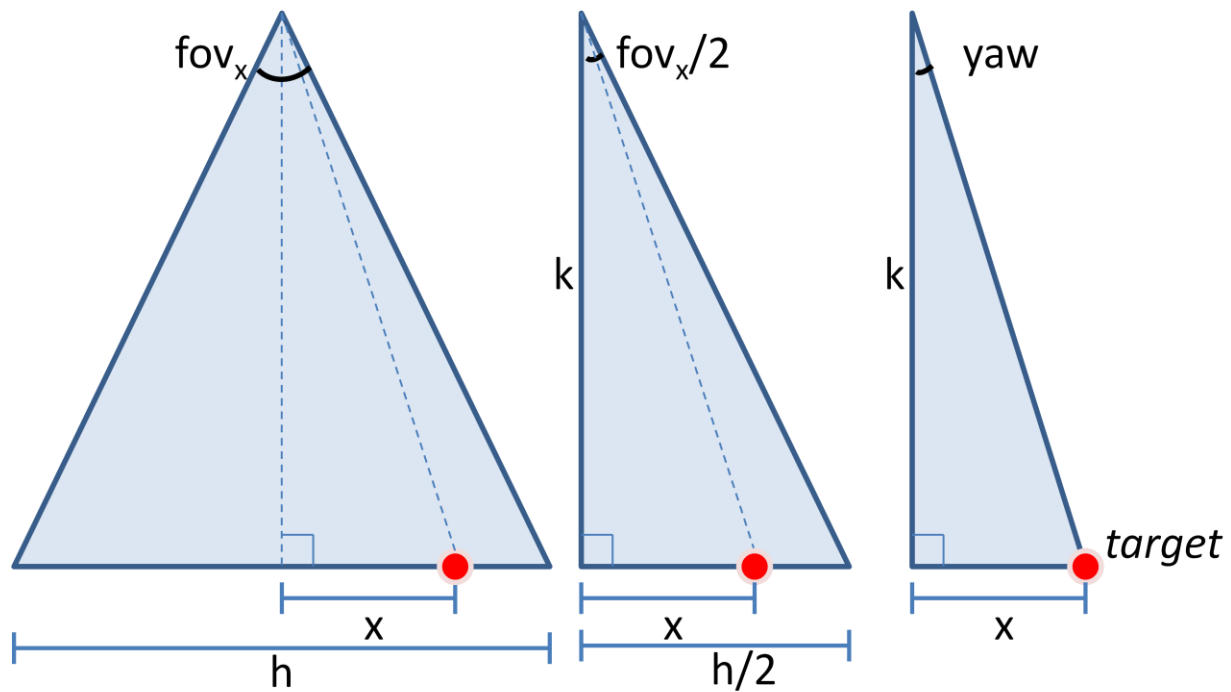


Figure 29: Attitude Error Estimator Geometry for Horizontal Camera Axis

From the far right triangle in Figure 29, there is the relation between the yaw and perpendicular distance from the focal point to the target shown below.

$$\tan(yaw) = \frac{x}{k} \quad (3.10)$$

The attitude error estimator is supposed to calculate yaw, so the equation above can be solved for yaw.

$$yaw = \tan^{-1} \left(\frac{x}{k} \right) \quad (3.11)$$

The perpendicular distance from the focal point to the target is unknown, but the value of k can be substituted using relations from the middle triangle of Figure 29.

$$\tan \left(\frac{fov_x}{2} \right) = \frac{h}{2k} \quad (3.12)$$

$$k = \frac{h}{2 \tan \left(\frac{fov_x}{2} \right)} \quad (3.13)$$

Plug the relation for k into the equation for yaw.

$$yaw = \tan^{-1} \left[\frac{2x}{h} \tan \left(\frac{fov_x}{2} \right) \right] \quad (3.14)$$

The same process is done for pitch except the vertical field of view and vertical size of the camera screen in pixels must be used. Once the yaw and pitch are calculated, an Euler rotational sequence can be constructed and translated into MRPs. The software module developed in UMBRA for the attitude error estimator was created by another researcher²⁰.

Once the attitude is calculated in MRPs, sequential camera frames can be used to differentiate the attitude and calculate the relative attitude time derivative $\dot{\bar{\sigma}}_{B/R}$. The relative angular velocity and the relative attitude derivative can be calculated from each other using the following equation².

$$\dot{\bar{\sigma}}_{B/R} = \frac{1}{4} [B] \bar{\omega}_{B/R} \quad (3.15)$$

$$[B] = (1 - \sigma^2)[I_{3 \times 3}] + 2[\tilde{\sigma}] + 2\sigma\sigma^T \quad (3.16)$$

The attitude error estimator module gives the control module a relative MRP value for each time step. Sequential time steps can be numerically differentiated to calculate $\dot{\tilde{\sigma}}$. Numerically differentiating noisy data leads to noise amplification, so a low pass filter is used. The data will be noisy even if a simulation environment is used with virtual modules (ex: virtual camera) because the boundary pixels of a target introduce noise in an image. Using a visual tracking algorithm to calculate the moments will reduce the noise compared to calculating the moments directly from the image with noisy boundary pixels, but in either case the relative attitude measurement will have some noise. Please see Figures 30-33 for numerical low pass filter justification.

Since $\dot{\tilde{\sigma}}$ is the variable that is known, the equation above must be rearranged to solve for the relative angular velocity as shown in the following equation.

$$\bar{\omega}_{B/R} = 4[B]^{-1}\dot{\tilde{\sigma}}_{B/R} \quad (3.17)$$

The graphs below show the relative angular velocity. Unfiltered, it is subject to noise amplification because it is calculated from the numerical differentiation of the relative MRPs, which will have some noise from the statistical pressure snakes and camera data. Numerically differentiating noisy data will lead to noise amplification. Putting this into a control without a filter could drive the control unstable. Everything could be correct with the control theory, but the numerical inputs would lead to an unstable solution. A low pass filter will reduce the noise, but will also introduce an element of lag into the control. In this system, the increased performance in the control dwarfs any concerns about lag. The angular velocity term is needed for a stable control, and the filter is needed to use the calculated angular velocity.

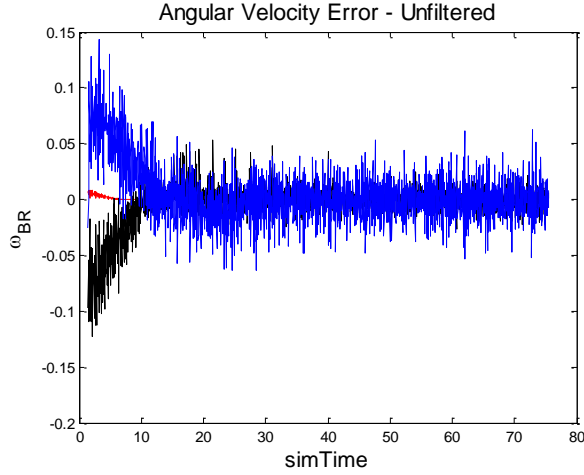


Figure 30: Unfiltered Angular Velocity (Time)

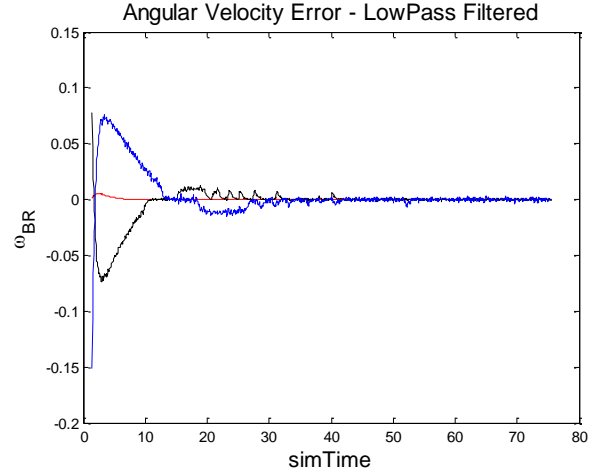


Figure 31: Filtered Angular Velocity (Time)

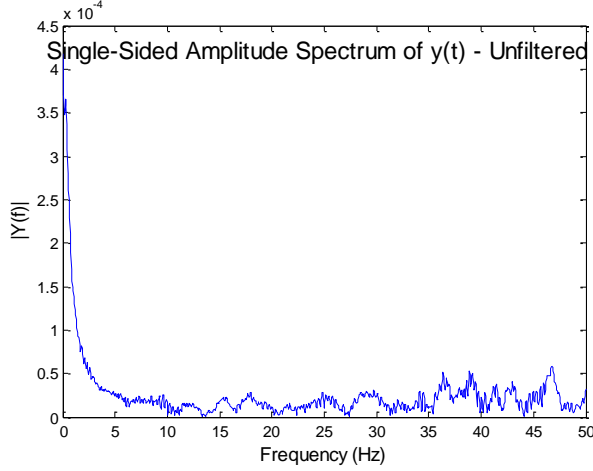


Figure 32: Unfiltered Angular Velocity (Frequency)

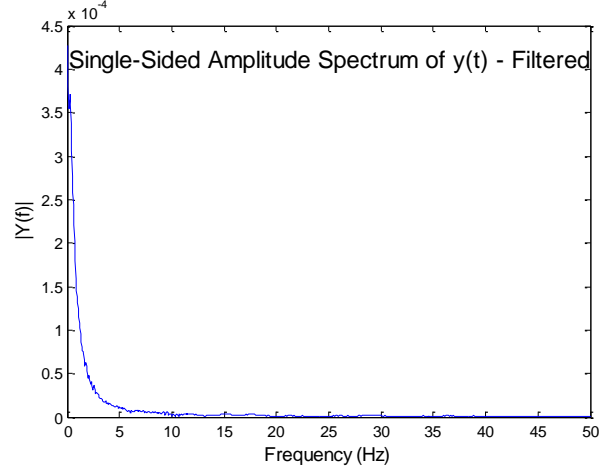


Figure 33: Filtered Angular Velocity (Frequency)

Since at least one variable required in the attitude control is subject to noise amplification and requires a low pass filter ($\bar{\omega}_{B/R}$), a filter module was created in UMBRA. Any number of states can be filtered using the same filter module after initialization. Fourteen filter options are programmed, including first order low pass, second order low pass, first order high pass, recursive, bandpass, and notch. For details, see the appendix.

3.2.1.2 Control

If only the relative attitude terms are available for the control, the resulting control law is reduced to the following. Note that for a purely relative control, the inertia tensor of the body satellite does not need to

be known. The only information that must be known is data coming from the camera. If the angular velocity term is used, it is calculated by the numerical differentiation of the relative Modified Rodrigues Parameter $\bar{\sigma}_{B/R}$ (as explained in a previous section 3.2.1.2), so a filter is required. A filter introduces lag into the control. If this is not desired, the following purely relative control can be used for very smooth reference motions, but a lot of the dynamics of the system will be treated as disturbances by the control. The system must be known very well, including the overall motion of the spacecraft and target, before choosing an appropriate control if some of the system dynamics are going to be treated as disturbances in the control.

The output of each attitude control, u , is a torque vector that will act to rotate the spacecraft.

3.2.1.2.1 Purely Relative Attitude Control – No Filter

It is not recommended to use the purely relative attitude control $\bar{u}_{RELNoFilter}$ unless it is absolutely known that the reference motion is incredibly slow and smooth and this control can work for the chosen system. This control has no angular velocity term and treats many of the system terms as disturbances. This control was not shown to be asymptotically stabilizing for the presented simulations and was therefore not chosen as an adequate control for the underlying reference motion.

$$\bar{u}_{RELNoFilter} = -K\bar{\sigma}_{B/R} \quad (3.18)$$

3.2.1.2.2 Purely Relative Attitude Control – Filter

For the purposes of the example simulations, the purely relative attitude control with a filter worked, but since the inertia and body angular velocity was known, the relative/body control was chosen.

As explained in a previous section (Section 3.2.1.2), $\bar{\omega}_{B/R}$ must have a low pass filter applied before it can be used in a control because it is subject to noise amplification. If $\bar{\omega}_{B/R}$ is not filtered, it can drive the control unstable.

$$\bar{u}_{RELFilter} = -K\bar{\sigma}_{B/R} - [P]\bar{\omega}_{B/R} \quad (3.19)$$

3.2.1.3 Results

3.2.1.3.1 Attitude Regulation

For an attitude regulation simulation, the simulation starts with an attitude error. The reference conditions to be tracked are all zero. With the chosen attitude description of Modified Rodrigues Parameters, each of the three elements must be brought to and kept at zero.

Table 3: Attitude Regulation Initial Conditions

$\bar{\sigma}_{R/N} = [0 \ 0 \ 0]^T$	Target attitude from the Simulated Target Attitude module
$[10 \ 0.8 \ 0.8]$	Initial Position offset between Camera and Target [x y z]
$[I] = \text{diag}[10 \ 20 \ 30]$	Body inertia tensor
$K = 15$	Relative attitude gain
$[P] = \text{diag}[5 \ 5 \ 5]$	Relative angular velocity gain (if the control uses it)
$\bar{L} = [0 \ 0 \ 0]^T$	Known external torque

The initial view from the camera looked like the following figure. As you can see, the center of the target is not in the center of the camera screen, so there is an initial attitude error that the attitude control law will drive to zero.

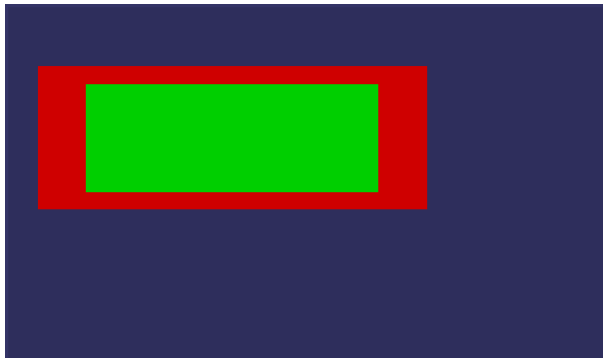


Figure 34: Initial Attitude Regulation Camera Simulation View

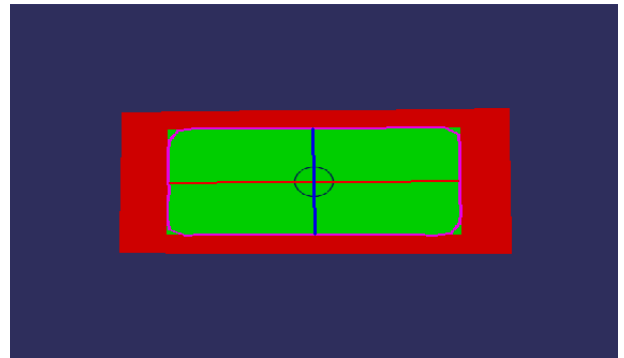


Figure 35: Final Attitude Regulation Camera Simulation View

The relative attitude is driven to zero, which is what the full Lyapunov control would be expected to do.

The relative control works with the given regulation simulation.

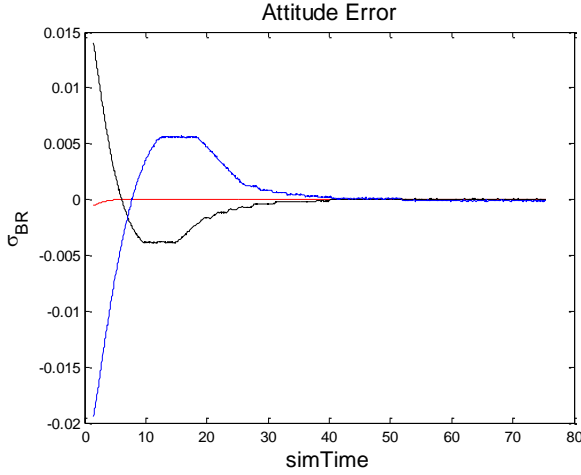


Figure 36: Relative Attitude Control Regulation Error

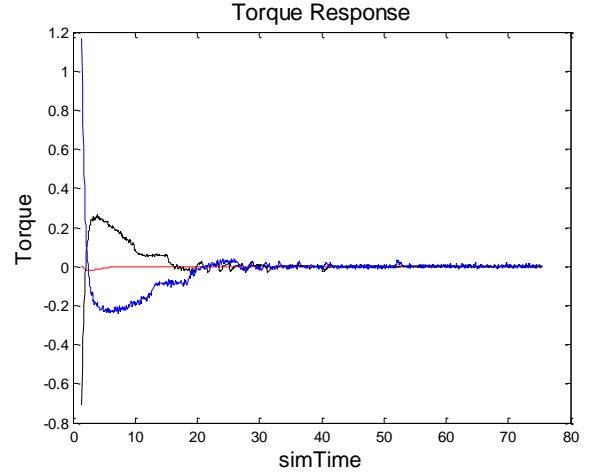


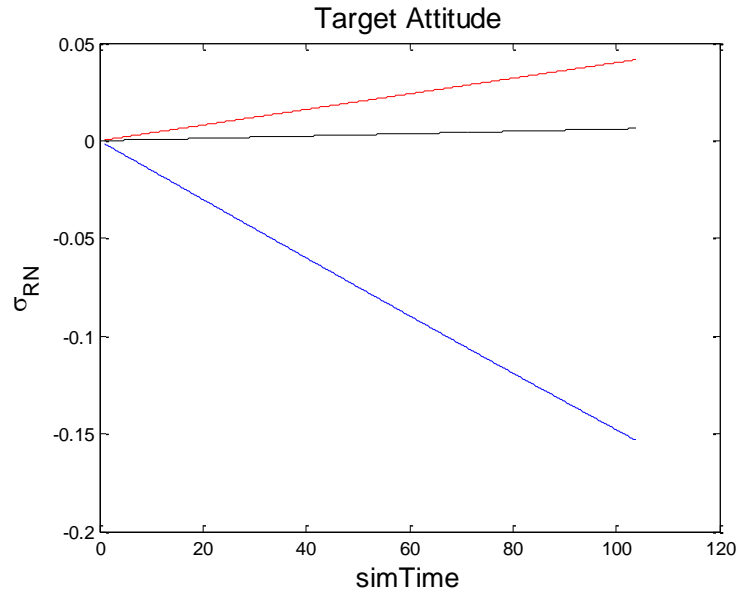
Figure 37: Relative Attitude Control Regulation Torque Response

3.2.1.3.2 Attitude Tracking

Attitude tracking differs from regulation because the target is moving according to some reference motion instead of being stationary. The simulated target attitude is shown for a certain length of time. The simulated target attitude is not subject to noise and will not vary across each simulation.

Table 4: Attitude Tracking Initial Conditions

$\bar{\sigma}_{R/N} = \begin{bmatrix} 0.4\sin(0.005t) \\ 0.3\sin(0.001t) \\ 0.5\sin(-0.015t) \end{bmatrix}$	Target attitude from the Simulated Target Attitude module
$[10 \ 0.8 \ 0.8]$	Initial Position offset between Camera and Target [x y z]
$[I] = \text{diag}[10 \ 20 \ 30]$	Body inertia tensor
$K = 15$	Relative attitude gain
$[P] = \text{diag}[5 \ 5 \ 5]$	Relative angular velocity gain (if the control uses it)
$\bar{L} = [0 \ 0 \ 0]^T$	Known external torque



With the tracking problem, the relative attitude control drives the error toward zero. The control is treating some of the dynamics of the system as disturbances, so the error does not stay ideally at zero, but the error remains bounded to a very small error within 10^{-3} MRP.

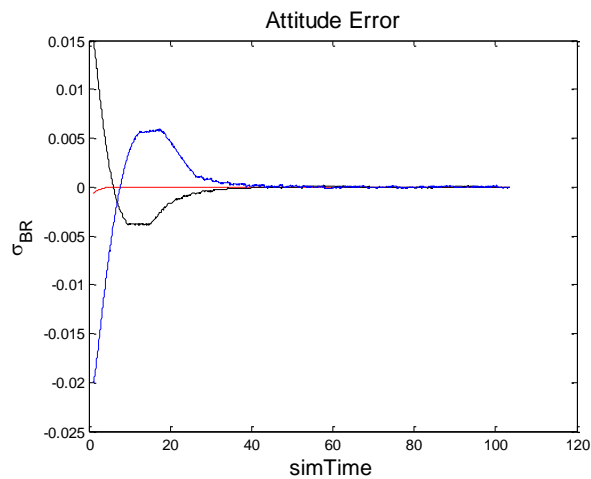


Figure 38: Attitude Tracking with Purely Relative Control

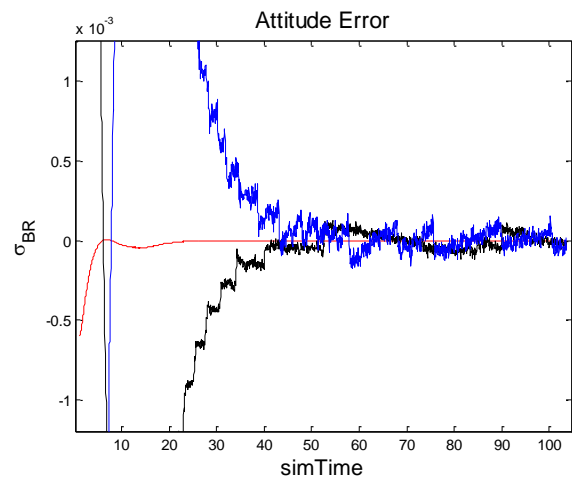


Figure 39: Zoomed View of Attitude Tracking with Purely Relative Control

3.2.2 Body Motion

The body angular velocity term in the full Lyapunov control is $\bar{\omega}_{B/N}$.

3.2.2.1 Origin

The body attitude term of the control law ($\bar{\omega}_{B/N}$) is obtained from an onboard gyroscope. In general, it is a good assumption that the body attitude is known. These measurements will be imperfect and subject to noise. Noisy data can be filtered and redundancies can be built in, but there may be certain cases or missions in which the body attitude is not included in the control, such as if the spacecraft inertia cannot be assumed as a known. For the following numerical simulations, the body attitude terms were not corrupted to simulate the noise of an onboard gyroscope.

3.2.2.2 Control

If the relative and body terms are available for the control, the resulting control law is reduced from Section 3.1 to the following.

$$\bar{u}_{RelBody} = -K\bar{\sigma}_{B/R} - [P]\bar{\omega}_{B/R} + [\tilde{\omega}_{B/N}][I]\bar{\omega}_{B/N} \quad (3.20)$$

For the purposes of the proposed simulations, the purely relative attitude control with a filter (Section 3.2.1.3.2) and the relative/body attitude control $\bar{u}_{RelBody}$ were both functional. Because the inertia of the spacecraft and the body angular velocity could both be assumed as known, the relative/body attitude control was chosen after the studies in the following section (Section 3.2.2.4).

3.2.2.3 Results

3.2.2.3.1 Attitude Regulation

The same initial conditions were chosen for the body/relative regulation simulation as the relative regulation simulation in the previous section.

The body/relative attitude is driven to zero, which is what the full Lyapunov control would be expected to do. The body/relative control works with the given regulation simulation.

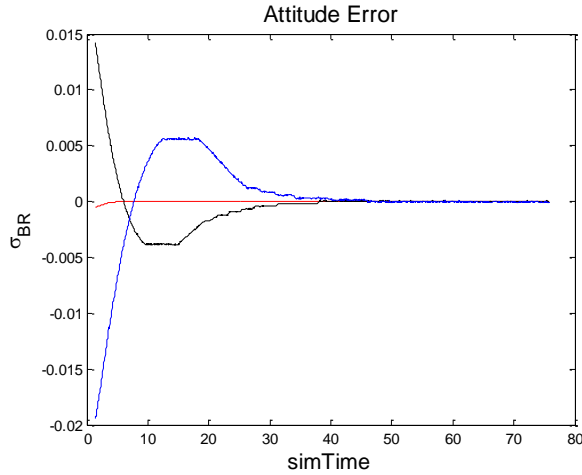


Figure 40: Body/Relative Attitude Control Regulation Error

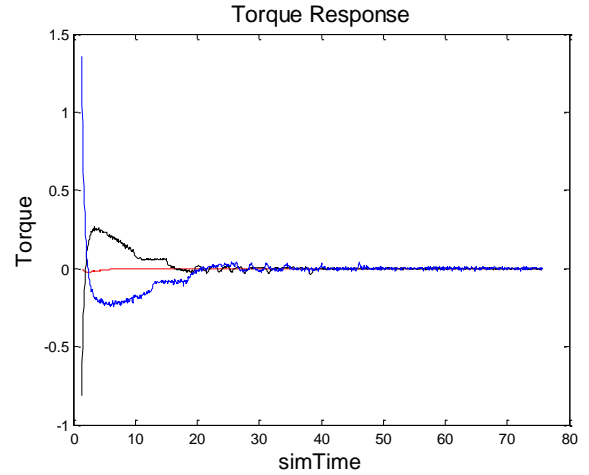


Figure 41: Body/Relative Attitude Control Regulation Torque Response

3.2.2.3.2 Attitude Tracking

The same simulation conditions were chosen for the body/relative tracking simulation as the relative tracking simulation in the previous section.

The body/relative attitude control drives the relative attitude to zero during the provided tracking conditions, even though some of the system dynamics are treated as disturbances. Since there are system dynamics being treated as disturbances and mostly compensated for by the control gains, the attitude error is not asymptotically driven to zero but the error remains below 10^{-3} MRP.

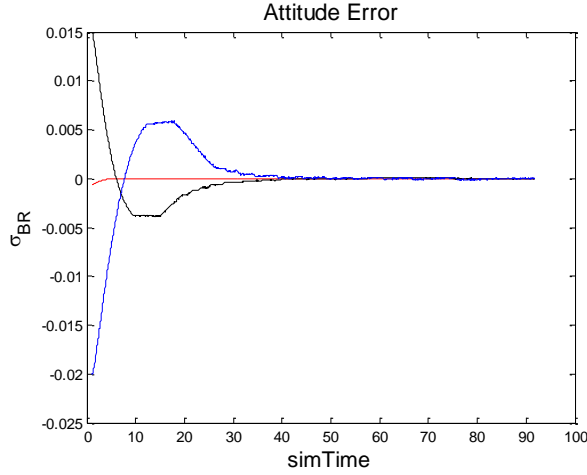


Figure 42: Attitude Tracking with Relative/Body Control

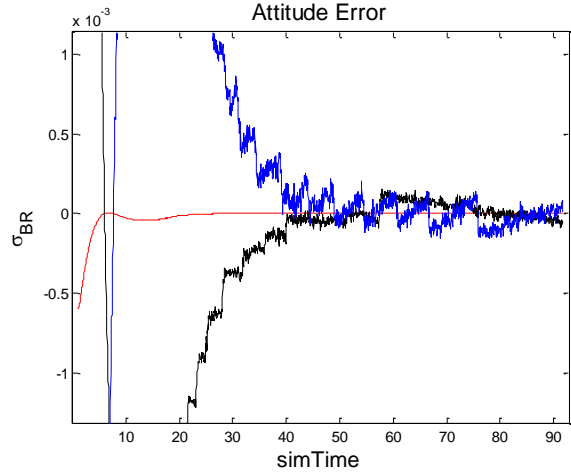


Figure 43: Zoomed View of Attitude Tracking with Relative/Body Control

3.2.3 Target Motion

The target angular velocity and angular acceleration terms in the full Lyapunov control are $\bar{\omega}_{R/N}$ and $\dot{\bar{\omega}}_{R/N}$.

3.2.3.1 Origin

The target inertial terms cannot be obtained directly from the target satellite's measurement devices without violating the overall goal of this thesis: *passive* relative visual control. If both the relative and body angular velocities are known, the target angular velocity can be estimated using the equation below.

$$\bar{\omega}_{R/N} = \bar{\omega}_{B/N} - \bar{\omega}_{B/R} \quad (3.21)$$

Analytically this is very simple, but there could be numerical issues since $\bar{\omega}_{R/N}$ will be subject to noise.

The relative angular velocity $\bar{\omega}_{B/R}$ is subject to noise amplification and must be filtered. The body angular velocity $\bar{\omega}_{B/N}$ is from a gyroscope measurement and will be filtered. This means that the target angular velocity is based on two separate sources of noise and two pre-filtered values.

The target angular acceleration term will be calculated by numerically differentiating the target angular velocity and will be subject to noise amplification. The target angular acceleration will need to be filtered, making it the third layer of filtering and subject to lag.

The presented simulations include smooth, slow reference motions. In future research, if sharp, fast reference motions are included, the full attitude control with better estimation of the target inertial terms should be pursued. It is a numerical issue to be aware of, but outside the scope of this thesis.

3.2.3.2 *Control*

Once the relative, body, and target terms are all located and estimated, the full Lyapunov unconstrained control law can be employed.

3.2.4 *Summary*

Both the purely relative and relative/body controls performed almost identically well in the given simulations. Orbital motion is about smooth, slow changes and reference motions, so either control would work. However, the relative/body control contains more information and is closer to the full control developed in Section 3.1. The body angular velocity could be obtained by gyroscopes while working with hardware and the attitude propagator module in the UMBRA simulation. The inertia of each spacecraft is known to a good approximation before flight, but if this is not known, the purely relative control should be used. The purely relative control can track smoothly varying reference motion such as the slowly varying sinusoidal reference motion of the attitude tracking simulation.

3.3 *Attitude Simulation Module Setup*

The Simulated Target Attitude module provides the simulated reference Modified Rodrigues Parameters of the simulation. The next three modules form a framework for the output of the Simulated Target Attitude module to be connected to the target's attitude in the virtual world, so when the Simulated Target Attitude module output is at some value, the GeoObject target module in the virtual world also has that same attitude. This makes it so the target (a rectangle, sphere, satellite, etc.) can follow a reference

attitude trajectory. Note: the information from the Simulated Target Attitude module is not shared with the rest of the simulation. This would be the same as the target satellite being in constant communication with the body satellite, and is in direct violation of the project goals.

The Virtual Camera module is in the same virtual world as the GeoObject target module. The Virtual Camera records camera data of the target—the streaming update of individual camera shots make the camera data similar to a movie of whatever the camera is pointing at (hopefully the target). The data from the Virtual Camera then flows through the Statistical Pressure Snake module where the outer boundary of the target is tracked and moment information, such as the center of the image and inertia in pixels, is calculated. The central coordinates of the target calculated from the Statistical Pressure Snakes are sent to the Attitude Error Estimator module, which calculates the relative Modified Rodrigues Parameters of the camera and target frames given the offset of the center of the target from the center of the camera's screen (see Section 3.2.1).

The Attitude Error Estimator module gives the Relative Control Law module the relative attitude between the body and target frames (a.k.a. attitude error). The Relative Control Law module acts to drive this relative attitude to zero. Sequential relative attitude measurements can be used to calculate the derivative of the relative attitude, which can be used to calculate the relative angular velocity. The relative angular velocity would be subject to noise amplification since it would be calculated by numerically differentiating a noisy value, which is where the Filter module comes in. The Attitude Propagator module propagates the body attitude given the applied control torque from the Relative Control Law module so the simulation can keep track of how the body satellite is rotating.

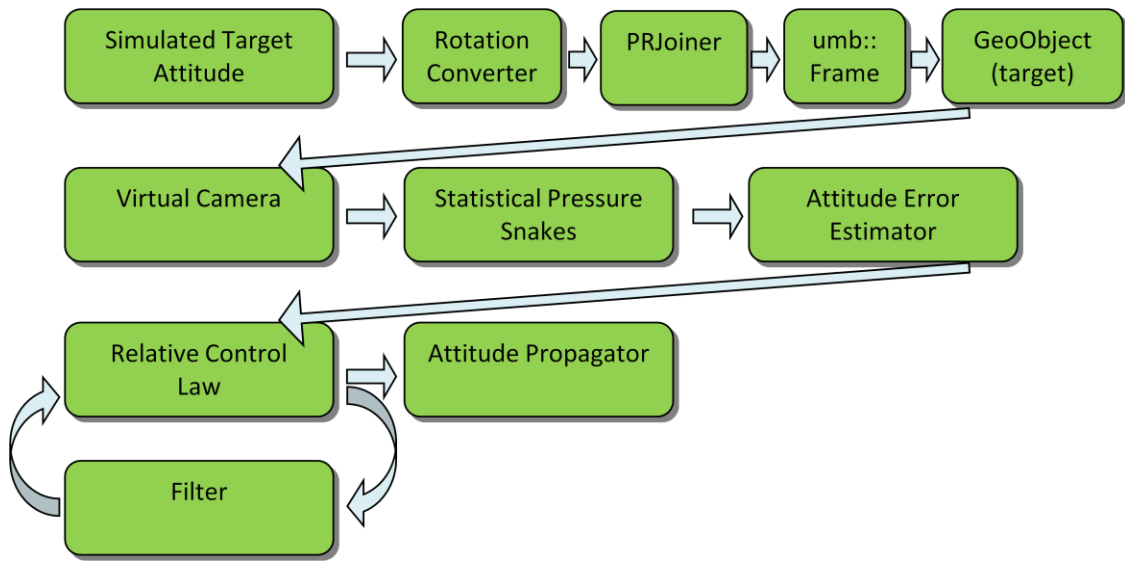


Figure 44: Attitude Simulation Module Setup

Chapter 4 Visual Skewness Control Properties

4.1 Background of Geometric Moments

Geometric moments have been used in a myriad of ways. Whether used to find the center of mass of a paper clip or automatic aircraft identification, moment equations all start from the same basic formula.

4.1.1 Overview

The general, raw, non-central moment equation for one axis is the following equation in which m_p is the moment, p is the order, $\rho(x)$ is the density function, and x is the variable²¹. $E[*]$ is the notation for expectation. This general moment equation can be applied to statistics, image analysis, or mass. There are differences, such as what is used for the density function.

$$m_p = E[x^p] = \int_{-\infty}^{\infty} x^p \rho(x) dx \quad (4.1)$$

The general, central moment equation for one axis that can also be applied to statistics, image analysis, or mass is the following equation²¹. The central moment is useful for looking at tendencies around the mean of the data set, image, or mass, and is what will be utilized in this research. For simplicity of notation, the subtraction of the mean will be dropped from further equations even though the central moment equation will be used.

$$m_p = E[x - E(x)]^p = \int_{-\infty}^{\infty} [x - E(x)]^p \rho(x) dx \quad (4.2)$$

The general moment equation for mixed-axes is the following equation¹. The mixed-axis equation is not used for statistics applications because data for statistical tests are in one dimension.

$$m_{pq} = \iint_{\zeta} x^p y^q f(x, y) dx dy, \quad p, q = 0, 1, 2, 3 \dots \quad (4.3)$$

For a continuous body, the equation below is the relation for a moment about the x -axis M_{x_p} of any order p , non-uniform density $\rho(x, y)$.

$$M_{x_p} = \iint_A y^p \rho(x, y) dx dy \quad (4.4)$$

4.1.1.1 Density Functions

For statistics applications, the variable x in the moment equations is random. The density function $\rho(x)$ for statistics is the probability density function. There are specific requirements that must be met for the probability density function to hold for statistics applications. Some of these make intuitive sense, such as the probability of an event occurring cannot be less than zero percent and cannot be greater than one hundred percent. Let the probability P of an event occurring when the variable x is a certain value x_j be equal to the density function at that value.

$$P(x = x_j) = \rho(x_j) \quad (4.5)$$

The probability is constrained by the following²¹:

$$1. \quad 0 \leq P \leq 1 \quad (4.6)$$

$$2. \quad P(-\infty \leq x \leq \infty) = 1 \quad (4.7)$$

$$3. \quad \text{For } x_k > x_j \quad P(x_j \leq x \leq x_k) = P(x \leq x_k) - P(x \leq x_j) \quad (4.8)$$

For mass applications, the density function in the moment formula is the density function of the body. The physical mass per volume characteristics of the material at each coordinate point within the body determines the density function.

For image applications, the density function in the moment formula is the intensity function of the image. This is described by the hue of each pixel.

4.1.2 Moments Across Disciplines

The equations for geometric moments are simple calculus, but the applications for geometric moments split off into many different fields. Many dynamics fields have used geometric moments to describe the

characteristics of a physics object like area, center of mass, and inertia. Statisticians have used moments in order to describe data. Researchers in image analysis and computer vision have used geometric moments for numerous applications mentioned in Mukundan including automatic character recognition, aircraft identification, pattern matching, “invariant pattern recognition, object classification, pose estimation, image coding, and reconstruction¹.” In each field, some of the moments have been visualized and interpreted differently due to the way they have been applied. A lot can be learned by looking across the different fields. In particular, statisticians are not constrained to think in terms of three-dimensional real bodies that have mass and other properties, so the way they have visualized some of the higher moments (especially the third) has been very helpful. The same concepts of geometric moments apply across all fields; after all, the same equations are being used.

4.2 Skewness Proof

Previous researchers in image analysis have recognized that skewness in images is a measure of symmetry, but no one has applied that to a measure of perpendicularity to determine if the target is “tilted” with respect to the camera, meaning the camera should move around the target until it is in the perpendicular or “straight on” orientation.

There are three important points readers should be aware of in the following sections:

- 1) If the first image/mass moment has the same concept as the first statistics moment, and the second image/mass moment has the same concept as the second statistics moment, then the third image/mass moment has the same concept as the third statistics moment. There are multiple image analysis resources that have proven using the third moment in image analysis with the same concept as the third statistics moment, but geometric moment concepts will make more sense if readers are exposed to this rather than being pointed to references.
- 2) Differences between mass and statistics moments come from the questions being asked. The concepts and equations are the same.

- 3) The non-dimensional skewness value g_3 can be used as a control variable for perpendicularity when certain knowledge is known about the target.

4.2.1 Mass/Image and Statistics Moments are the Same Concept

The point of this section is to convince the reader that the third geometric moment can be used in image analysis as a measure of skewness and symmetry, just like in statistics. This concept has been proven and is shown in Mukundan¹. Shutler even shows an example of letter recognition and how using the skewness coefficient allows different letters to be recognized in image analysis⁵.

4.2.1.1 First Geometric Moment

Thinking of moments in mass starts out as a calculus concept you can literally wrap your fingers around. Using the moment equation for a continuous body, the zeroth mass moment is the following equation, which is the total mass of the object.

$$M_{x_0} = \iint_A \rho(x, y) dx dy \quad (4.9)$$

Since the mass of a continuous body does not vary, it is the same for both axes; therefore, the zeroth moment for both axes must be equal.

$$M_{x_0} = M_{y_0} \quad (4.10)$$

Using the moment equation for a continuous body, the first mass moment is the following equation. The distance from the x -axis to the infinitesimal coordinate of the continuous body is y . The first moment divided by the total system mass (the zeroth moment), equals the y -coordinate center of mass \tilde{y} . Find a pen, envelope, or some sort of small object and try to balance it on your finger. The point at which it is balanced is its center of mass coordinate.

$$M_{x_1} = \iint_A y \rho(x, y) dx dy \quad (4.11)$$

$$\tilde{y} = \frac{M_{x_1}}{M_{x_0}} \quad (4.12)$$

In statistics, the first moment represents the mean of the data set. In mass, the first moment represents the center of mass or the mean of the object. In image analysis, the first moment represents the central coordinate pixel of the object. If constant density and a “straight on”, perpendicular orientation are assumed, the central coordinate pixel from image analysis and the center of mass from a mass moment will be identical. The density function used in the moment function for the mass moment is the physical density of the object (mass per volume), whereas the function used in the image moment is an intensity function based off of hue. The two will generally yield similar results, but remember that our eyes can sometimes play “tricks” on us. For example, a circle from a perpendicular orientation will look like an ellipse from a skewed orientation, which is why *a priori* knowledge must be known about the target before starting the control (see later sections).

4.2.1.2 *Second Geometric Moment*

Using the moment equation for a continuous body, the second mass moment is the following equation, which is the inertia of a body’s axis. Inertia is the distribution of mass around the center of mass. The classic physics example is the difference between a hollow or solid cylinder of equal mass rolling down a hill. The mass is distributed differently around the center of mass for the hollow cylinder (far away from the center of mass) than the solid cylinder in which the mass is closer to the center of mass, so the inertias are different. The hollow cylinder has a larger inertia than the solid cylinder because the hollow cylinder’s mass is distributed farther from its mean.

$$M_{x_2} = \iint_A y^2 \rho(x, y) dx dy \quad (4.13)$$

In statistics, the second geometric moment is called the variance. The variance can easily be seen in a data set as how the data are distributed around the mean. If all of the data are close to the mean, the variance is low. If the data are spread out and not near the mean, the variance is high. It makes intuitive

and visual sense. In the figure below, Curve A has a larger variance because the data are distributed further around the mean. Curve B has a lower variance because the scores, values, or observations for this data set are distributed closer to the mean.

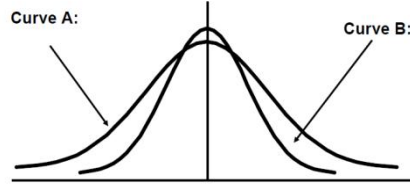


Figure 45: Comparison of Variance³

In statistics, mass, and image analysis, the second geometric moment represents the distribution of *stuff* around its mean.

4.2.1.3 Third Geometric Moment

Using the moment equation for a continuous body, the third mass moment is the following equation.

$$M_{x_3} = \iint_A y^3 \rho dx dy \quad (4.14)$$

It does not have a defined meaning or application in terms of rigid body analysis yet. Some researchers have started using the third moment in terms of image analysis, and this research continues their work.

In statistics, the third moment is often non-dimensionalized using the second moment to create a value called g_3 given below. In the equation below, m_3 is the third moment, m_2 is the second moment, and n is the number of points in the data set. For these purposes, n is the number of snake control points from the statistical pressure snake algorithm.

$$g_3 = \left[\frac{n(n-1)}{n-2} \right] \frac{m_3}{m_2^{3/2}} \quad (4.15)$$

The variable g_3 is known as the skewness of the data set. After calculating proper statistical tests, the value of γ_3 , the population variable, can be inferred, but that delves into the branch of inferential statistics and is different than the topic discussed here.

The figure below shows what skewness in a data set means. A positively skewed data set means a lot of the individual scores, values, or observations are lumped off in the lower end of the scale. In the upper graph in which the distribution is positively skewed, both g_3 and m_3 (third moment) will be positive (remember that m_2 , the variance, like inertia, can never be negative). In the lower figure in which the distribution is negatively skewed, both g_3 and m_3 will be negative.

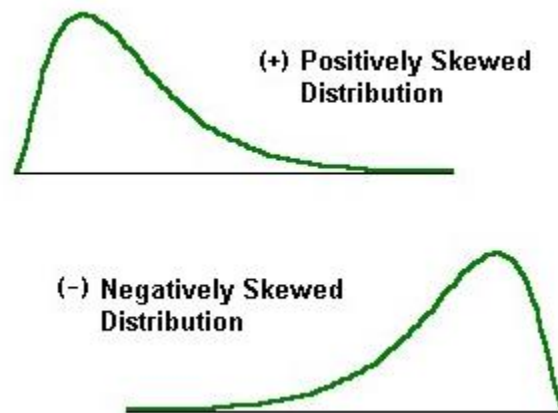


Figure 46: Skewness Figure from MVPStats⁴

The following is what Mukundan wrote about skewness as applied to image analysis.

“The third-order moments μ_{30} , μ_{03} denote *skewness* of the image projections. Skewness is a statistical measure of the degree of deviation from symmetry about the mean. If an image is symmetrical about the line $x = x_0$, then $\mu_{30} = 0$. We can therefore consider μ_{30} as a departure from symmetry about the mean axis $x = x_0$. Since μ_{20} is always positive, we can divide the μ_{30} term by $(\mu_{20})^{3/2}$ to get a non-dimensional quantity. Thus the coefficients of skewness of an image about the x-axis and y-axis are¹”

$$S_x = \frac{\mu_{30}}{\mu_{20}^{3/2}} \quad (4.16)$$

$$S_y = \frac{\mu_{03}}{\mu_{02}^{3/2}} \quad (4.17)$$

Mukundan realized that the same concepts about the third moment and skewness as applied to statistics could also be applied to image analysis. The field of image analysis has utilized the third moment and skewness as a measure of symmetry for applications such as letter recognition⁵.

Since skewness is a measure of symmetry for statistics and image analysis, and we know that geometric moments come from calculus and their concepts are the same across the fields to which they are applied, it makes sense that skewness is also a measure of symmetry in mass. This research only uses skewness as a measure of symmetry in image analysis, which has been proven by other researchers and used in other applications^{1,5}, but it is important to understand that a geometric moment of a specific order has the same concept no matter which field or application it is applied to.

The table below shows what the 1st-3rd geometric moments mean for mass, image, and statistics applications.

Table 5: Comparison of Mass, Image and Statistics Moments

Moment	Mass	Image	Statistics
1st	Center of Mass (Mean)	Central Pixel of Target (Mean)	Mean
2nd	Inertia (Distribution around Mean)	Inertia (Distribution around Mean)	Variance (Distribution around Mean)
3rd	Symmetry	Skewness, Symmetry	Skewness, Symmetry

4.2.2 Differences between Mass/Image and Statistics Moments

One way to think of the differences between statistics and mass moments is to think of the questions they are trying to answer. Take an example of a cell phone. For a mass moment, the question would be: what

is the total mass? For a statistics moment, the question would be: if you were to draw one particle out of the cell phone, what would you expect its individual mass to be? For the mass moment, the entire mass continuum of the cell phone would be included in the calculation (meaning every particle in the cell phone, or the entire research population). For the statistics moment, a random sample would be drawn from the research population, meaning a random sample of particles would be taken from the cell phone. When dealing with the rotation of satellites, we want to know the characteristics of the entire body, not the expected characteristics of each individual particle that makes up that satellite. The geometric moment equations used in mass and statistics moments are exactly the same—the questions being asked differ, and so what is put in to the equations differs slightly. The concepts and equations are the same; we just have to recognize which tools should be used for which jobs.

4.2.3 g_3 can be used as a Control Variable for Perpendicularity

In statistics and all the resources of image analysis, the third moment is said to represent skewness and symmetry about an axis. However; this research presents the idea that the third moment can be applied to represent more in specific cases, specifically a measure of perpendicularity. To show this, a series of shapes have been tested by slewing them off to a certain angle (approximately -30 degrees), and rotating them through approximately positive 30 degrees in one axis. The response of the third moment will tell some characteristics of the third moment across many shapes and how it can be applied.

4.2.3.1 Rectangle

If a rectangular target is slewed in one axis, the third moment becomes zero in the active axis when the rectangle is viewed from a “straight on”, perpendicular, or zero skewness orientation from the camera’s perspective, and is non-zero at every other configuration.



Figure 47: Rectangle

The graphs below show that both the third moment and g_3 skewness of the rectangle are negative when the camera views the rectangle from a negatively-skewed orientation, zero when the rectangle is viewed perpendicularly, and positive when the rectangle is viewed from a positively-skewed orientation.

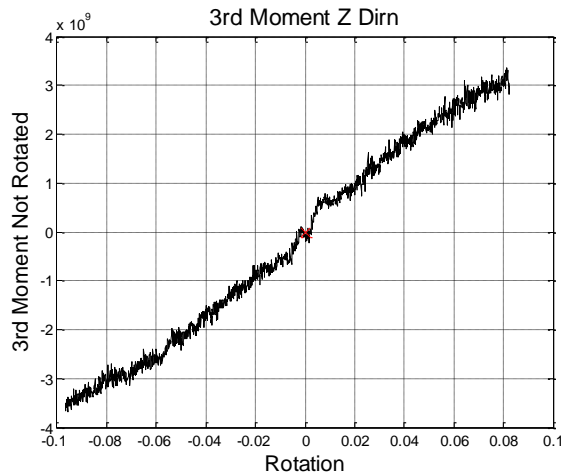


Figure 48: Rectangle M_{x3}

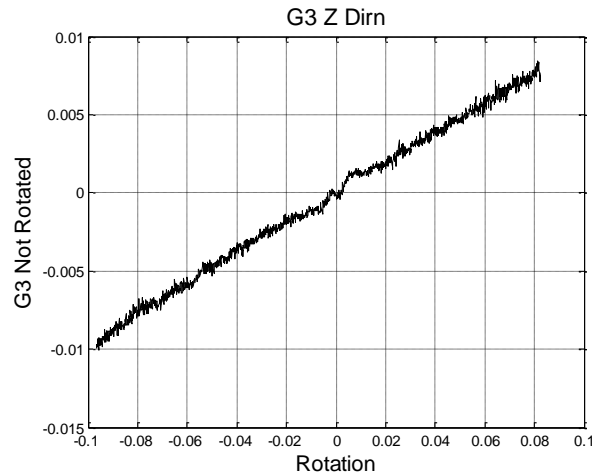
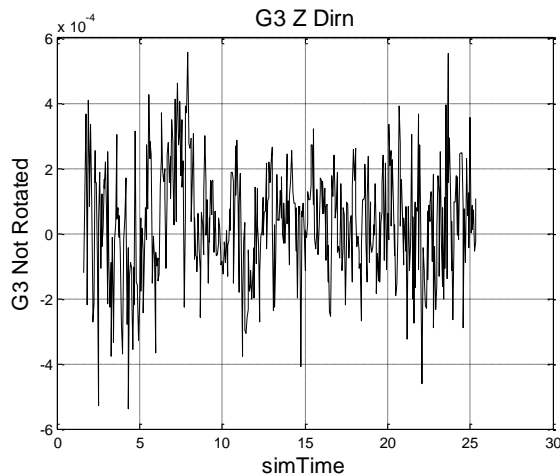


Figure 49: Rectangle g_{x3}

The graph below shows the g_3 , or skewness, value across time when viewing a stationary rectangle from a “straight on” orientation. The measurements from the statistical pressure snake are noisy, but the g_3 measurements across time when the camera views the rectangle at the “straight on”, or perpendicular orientation are approximately zero.



Straight On Average

Average M_{x3}	1.47845e+007
Average g_{x3}	2.95093e-005

Figure 50: Straight On g_{x3}

These are promising initial results because it indicates that both the third moment and g_3 can be used as a measure of perpendicularity for a rectangle. The camera is viewing the rectangle from a perpendicular orientation with the third moment or g_3 at a reference value of zero. This happens at zero rotation and no other time. When the third moment is zero, g_3 will also be zero and vice versa. Therefore, both are measures of perpendicularity for a rectangle. Also, the relationship on both graphs that the reference value only pertains to perpendicularity and no other orientation means that both the third moment and g_3 can be used as control variables for a rectangle. The fact that both are nearly-linear relationships allows for a simple control.

For a rectangle, the third mass moment and g_3 are measures of perpendicularity, meaning they can tell visually whether the rectangle is being viewed from “straight on” or at an angle. This is very important, especially for rendezvous and docking maneuvers. Imagine knowing that the target was directly in the center of the camera screen (attitude control), but not realizing the body satellite was approaching the target at an angle of 30 degrees. A docking maneuver would not be successful and a collision would probably occur.

Next, an investigation of whether the third mass moment and g_3 measures of perpendicularity hold for other shapes was carried out.

4.2.3.2 Trapezoid – Symmetric about Rotating Axis

If a trapezoid that is symmetric about the rotating axis is slewed in one axis, the third moment becomes zero in the active axis when the trapezoid is viewed from a perpendicular or zero skewness orientation from the camera's perspective, and is non-zero at every other configuration.

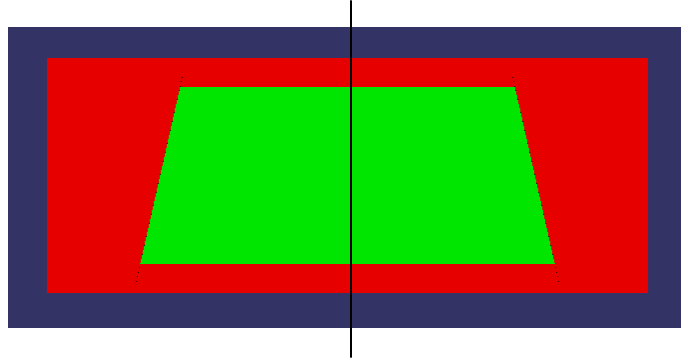


Figure 51: Trapezoid – Symmetric about Rotating Axis

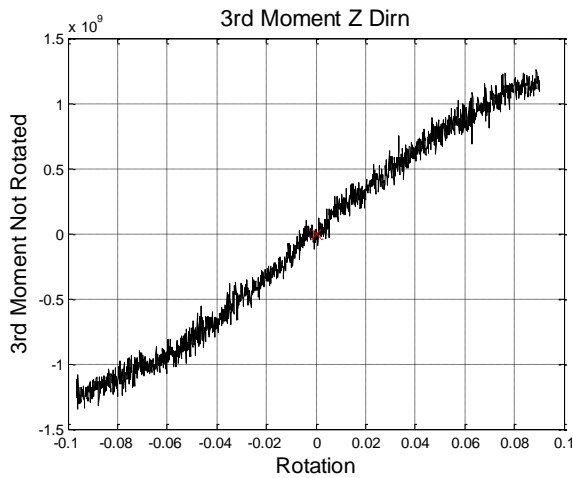


Figure 52: Trapezoid Symmetric about Rotation Axis M_{x3}

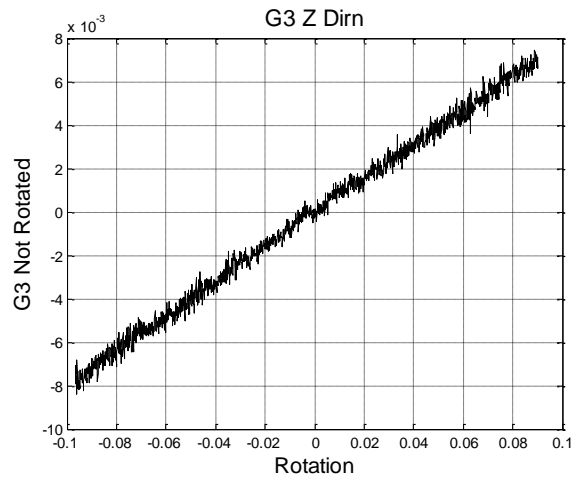
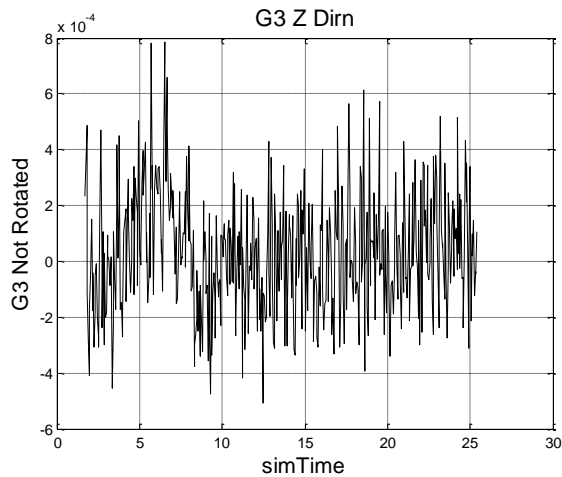


Figure 53: Trapezoid Symmetric about Rotation Axis g_{x3}

The graph below shows the g_3 , or skewness, value across time when viewing a stationary trapezoid that is symmetric about the rotating axis from a “straight on” orientation. Even with the noisy measurement, the g_3 measurement across time when the camera views the symmetric trapezoid axis at the “straight on”, or perpendicular orientation is approximately zero.



Straight On Average

Average M_{x3}	6.75079e+006
Average g_{x3}	3.06812e-005

Figure 54: Trapezoid Symmetric about Rotation Axis
Straight on g_{x3}

4.2.3.3 Trapezoid – Not Symmetric about Rotating Axis – More Mass in Negative Axis

If a trapezoid that is not symmetric about the rotating axis but has more mass in the negative rotating axis is slewed in one axis, the third moment has a positive reference value in the active axis when the trapezoid is viewed from a perpendicular or zero skewness orientation from the camera's perspective, and is different from that reference value at every other configuration.

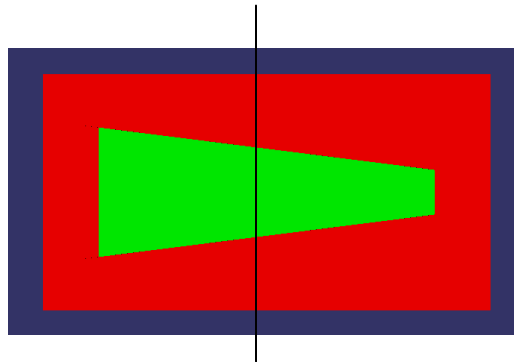


Figure 55: Not Symmetric about Rotating Axis – More Mass in Negative Axis

The moment graph below is different from that of the rectangle. When the trapezoid is first viewed by the camera from a negatively skewed orientation, the short side of the trapezoid is closest to the camera and the long side (where the most mass and therefore the most pixels are) is farthest away from the camera. This rotation causes the width of the trapezoid to appear smaller which makes the dimensions of the

trapezoid seem different—the trapezoid looks a little more like a rectangle or a square, which causes the third moment to be closer to zero. As the rotation brings the trapezoid more toward the “straight on” or perpendicular orientation, the short side of the trapezoid rotates away from the camera and the long side pivots closer to the camera. This rotation causes the width of the trapezoid to appear to lengthen as seen by the camera, which causes the trapezoid to look less symmetrical and causes the third moment to move further from zero. As the rotation continues toward the camera’s positively-skewed orientation, the long side is closest to the camera and the short side of the trapezoid is rotated farther away from the camera, which would make the trapezoid appear even less symmetrical, which causes the third moment to increase. However, this rotation also causes the width of the trapezoid to appear smaller from the camera’s point of view which makes the trapezoid again appear a little more like a rectangle or square and causes the absolute value of the moment to decrease. The net effect on the third moment is a decrease toward zero. The g_3 skewness value, however, increases since it is divided by a power of the second moment (inertia) which will decrease as the width of the trapezoid decreases.

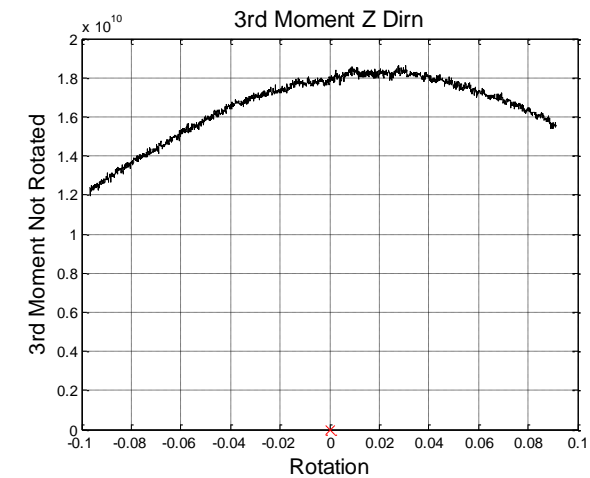


Figure 56: Not Symmetric about Rotating Axis – More Mass in Negative Axis M_{x3}

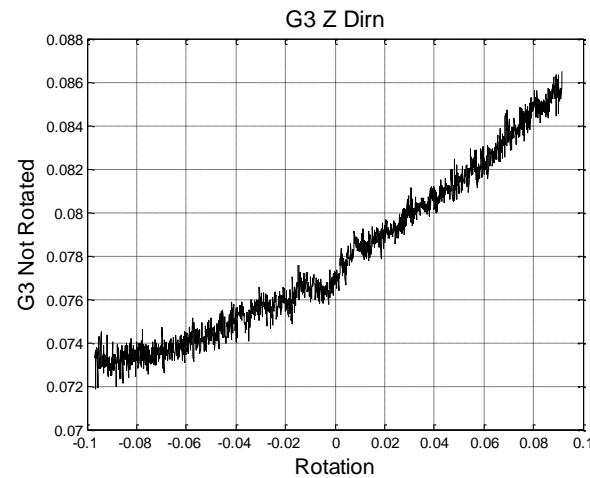
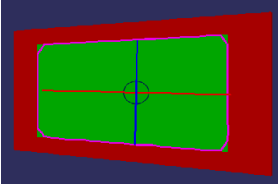
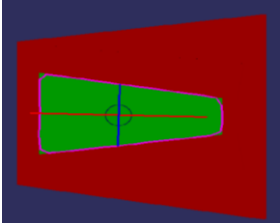

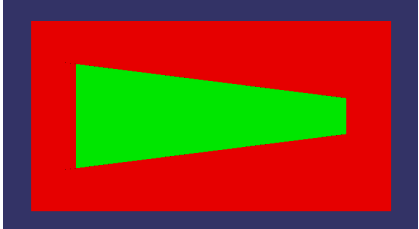
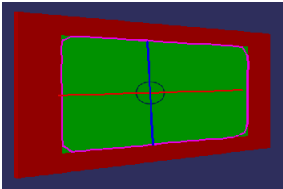
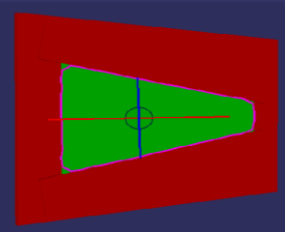


Figure 57: Not Symmetric about Rotating Axis – More Mass in Negative Axis g_{x3}

Table 6: Comparison of Rectangle and Trapezoid

Camera View	Rectangle	Trapezoid – Not Symmetric about Rotating Axis – More Mass in Negative Axis
-------------	-----------	--

<p>Negatively Skewed</p> <p>$\sigma_{BR} = [0 \ 0 \ -0.2]$</p>		
<p>Perpendicular Zero Skewness</p> <p>$\sigma_{BR} = [0 \ 0 \ 0]$</p>		
<p>Positively Skewed</p> <p>$\sigma_{BR} = [0 \ 0 \ 0.2]$</p>		

First, look at the rectangles. Notice that when the rectangle is positively skewed, it looks the same as a non-symmetrical trapezoid that has most of its mass in the negative axis. The third moment and skewness values show symmetry, and a non-symmetric trapezoid with most of its mass in the negative axis will not have calculated third moments and skewness variables that indicate symmetry. The statistics graph about skewness shown below is similar to the trapezoid with many of the (x,y) pixel coordinates in the negative axis; this similarity is reflected in the skewness calculation.

Table 7: Comparison of Positively Skewed Shape and Distribution

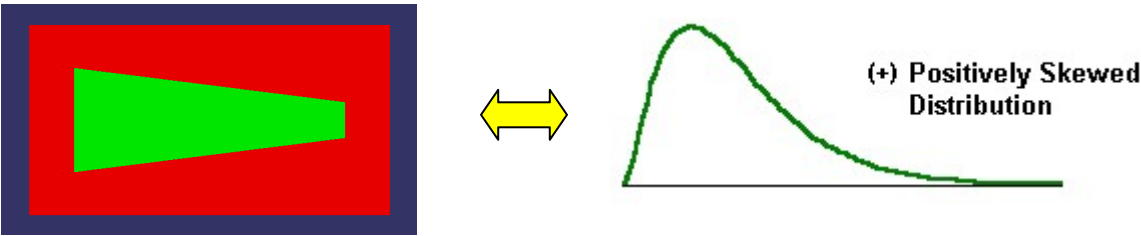
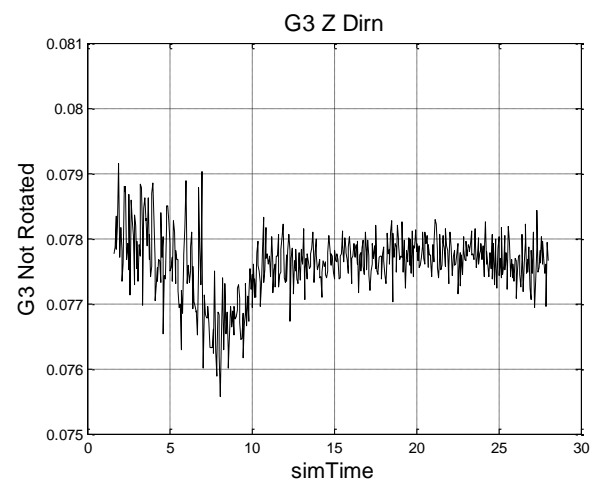


Figure 58: Positively Skewed Distribution⁴

Returning to Table 6 (Comparison of Rectangle and Trapezoid), since the rectangle is a symmetric shape, if the camera view is skewed the rectangle also looks skewed in exactly the same way. With the trapezoid, it is much harder for human eyes to differentiate. The calculation of g_3 is more sensitive and

accurate than the human eye since it can tell partial degree deviations in perpendicularity whereas the human eye can determine a difference at multiple degree deviations. The particular trapezoid chosen always looks positively skewed even when the camera is viewing the trapezoid from a negatively skewed angle. (Look at the red rectangle in the background that the green trapezoid is resting on if your eyes are “playing tricks” on you).



Straight On Average	
Average M_{x3}	1.83014e+010
Average g_{x3}	0.07760

Figure 59: Not Symmetric about Rotating Axis – More Mass in Negative Axis Straight on g_{x3}

4.2.3.4 *Trapezoid – Not Symmetric about Rotating Axis – More Mass in Positive Axis*

If a trapezoid that is non-symmetric about the rotating axis, has more mass in the positive axis, and is skewed in one axis, the third moment will have a negative reference value in the active axis when the trapezoid is viewed from a perpendicular or zero skewness orientation, and will not be that reference value at any other orientation.

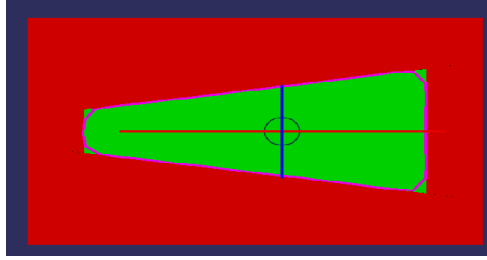


Figure 60: Trapezoid – Not Symmetric about Rotating Axis – More Mass in Positive Axis

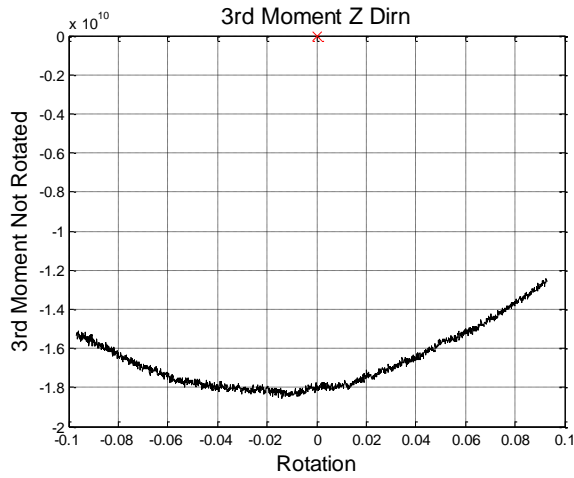


Figure 61: Trapezoid – Not Symmetric about Rotating Axis – More Mass in Positive Axis M_{x3}

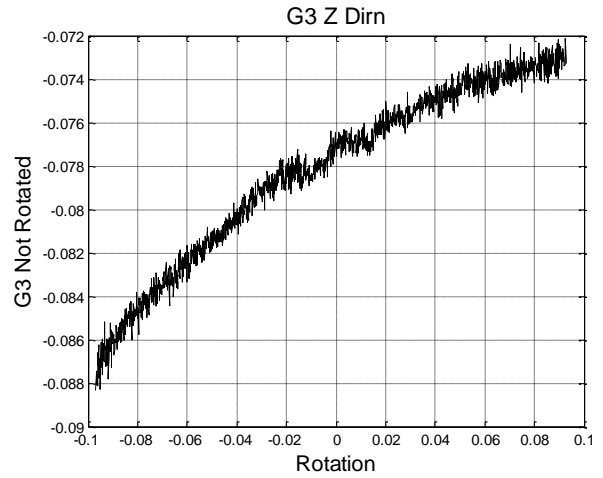


Figure 62: Trapezoid – Not Symmetric about Rotating Axis – More Mass in Positive Axis g_{x3}

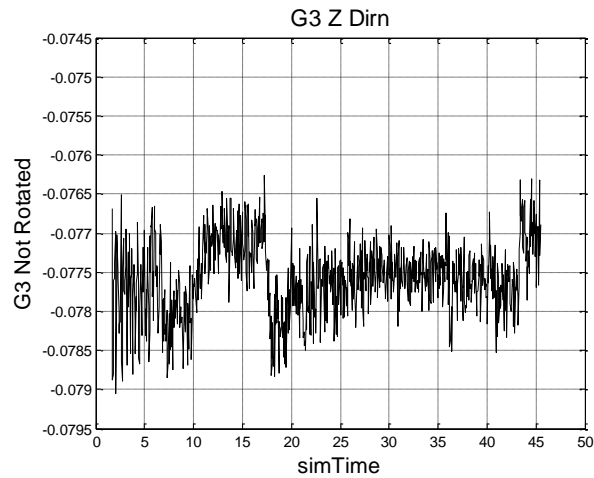


Figure 63: Trapezoid – Not Symmetric about Rotating Axis – More Mass in Positive Axis Straight On g_{x3}

Straight On Average

Average M_{x3}	-1.83391e+010
Average g_{x3}	-0.07757

4.2.3.5 Summary – One Axis Rotation

Shape	Symmetric or side of axis with most mass (negative, positive)	Third Moment M_{x3} at zero rotation	Skewness g_3 at zero rotation	Use M_{x3} as a control variable. (Xr at zero rotation and not Xr at every other rotation)	Use g_3 as a control variable.
Rectangle	Symmetric	0	0	Yes	Yes
Trapezoid	Symmetric	0	0	Yes	Yes
Trapezoid	-	+	+	No	Yes , there is just an Xr offset to be taken into account in the control.
Trapezoid	+	-	-	No	Yes , there is just an Xr offset to be taken into account in the control.

If something about the shape and its offset is known, then the control will effectively be driven to its offset instead of zero. The skewness g_3 is still a measure of perpendicularity, only a g_3 of zero for this shape does not correspond to a perpendicular orientation.

The fundamental knowledge the third moment and skewness value g_3 convey is symmetry. If some shape is being viewed “straight on” and is not symmetrical about an axis, the sign of the third moment in that axis will verify the direction and magnitude of asymmetry. Researchers in image analysis working on letter recognition have used this to their advantage to recognize whether a letter is symmetrical and about which axis⁵.

This research presents an idea that the third moment can be applied to tell relay information about perpendicularity, or the relative orientation between the body and target, and that is exactly what that controls will take advantage of. In order to use the skewness value g_3 for perpendicularity, information must be known about the target symmetry to start off with, because if it is not symmetric there will be an

offset that must be taken into account since the “straight on”, perpendicular configuration will correspond to a skewness value that is nonzero.

4.2.3.6 Skewness in Two Axes – Rectangle Linear Motion in Two Axes

First, an extension of the one-axis test was tested. Instead of only slewing about the z-axis, the target would also slew about the y-axis. This test shows that the skewness value for both axes can be extracted at the same time, and therefore the skewness of both axes can be controlled. This is important because if only one axis could be determined, then in a real-time application how would the software know which axis to choose? More importantly, what is gained by controlling skewness in one axis but not the other? Remember: with respect to a camera screen, there are only two axes (horizontal and vertical).

Similar to the one-axis tests, the camera is held inertially-fixed and the target is rotated. This is the same relative test. In a real-life example, the target would be moving according to some unknown reference motion and the camera/body would move in response, but this is a fixed test to look at the nature of how the skewness in both axes compares to the relative position between the camera/body and target.

4.2.3.6.1 Test 1 – Initially Positively Skewed in Both Axes

The time dependent values of the simulated target attitude for the simulation are given below.

$$\bar{\sigma}_{R/N} = \begin{bmatrix} 0 \\ -0.1 + t * 0.01 \\ -0.1 + t * 0.01 \end{bmatrix} \quad \text{Target attitude from the Simulated Target Attitude module}$$

The beginning, middle, and final positions for the first test are shown below.

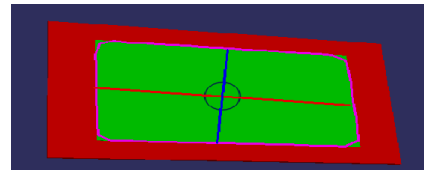
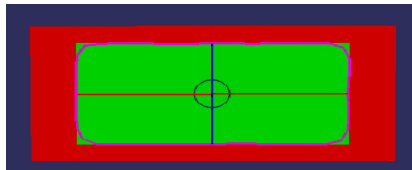
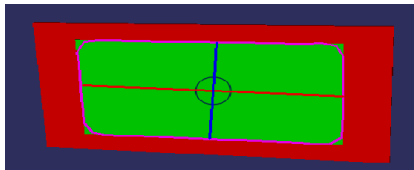


Figure 64: Simulation Initial View

Figure 65: Simulation Intermediate View

Figure 66: Simulation Final View

The final results show that both directions can be used and controlled simultaneously.

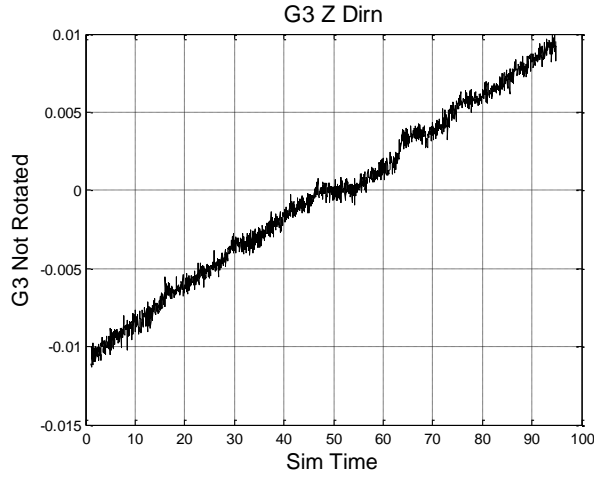


Figure 67: Z-Direction Skewness of Two-Axis Test

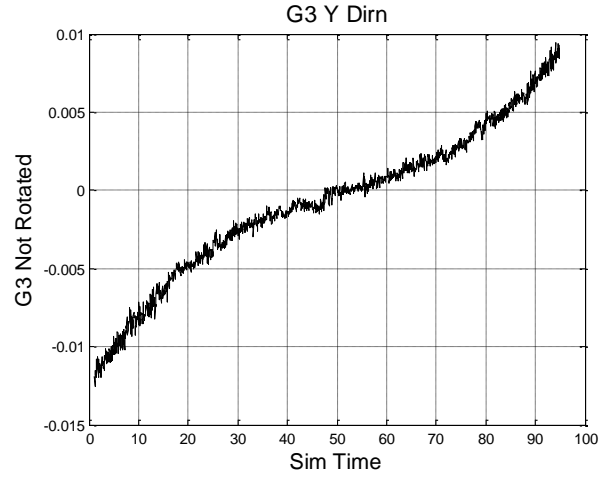


Figure 68: Y-Direction Skewness of Two-Axis Test

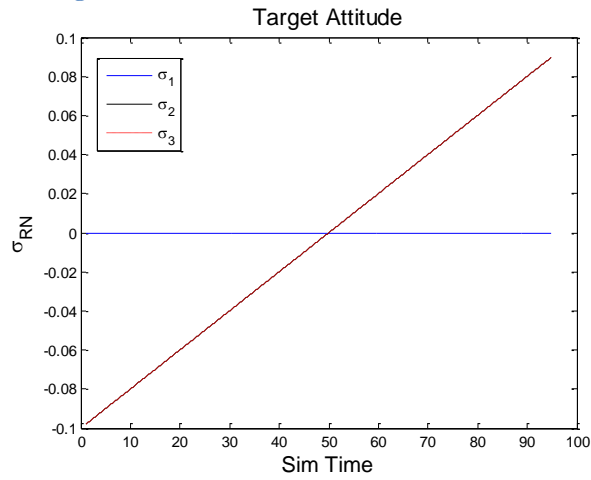


Figure 69: Simulated Target Attitude

4.2.3.6.2 Test 2 – Initially Positively Skewed in Z and Negatively Skewed in Y

The time dependent values of the simulated target attitude for the simulation are given below.

$$\bar{\sigma}_{R/N} = \begin{bmatrix} 0 \\ 0.1 - t * 0.01 \\ -0.1 + t * 0.01 \end{bmatrix} \quad \text{Target attitude from the Simulated Target Attitude module}$$

The beginning, middle, and final positions for the first test are shown below.

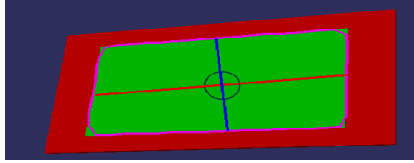


Figure 70: Simulation Initial View

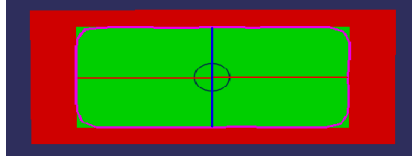


Figure 71: Simulation Intermediate View

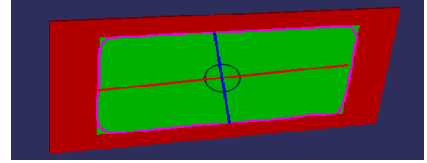


Figure 72: Simulation Final View

The final results show that both directions can be used and controlled simultaneously.

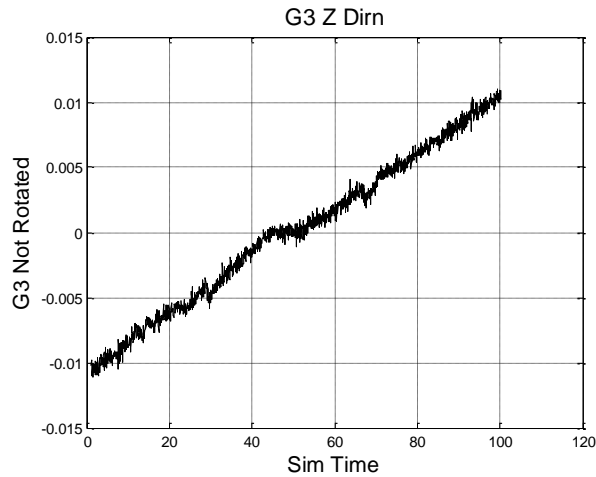


Figure 73: Z-Direction Skewness of Two-Axis Test

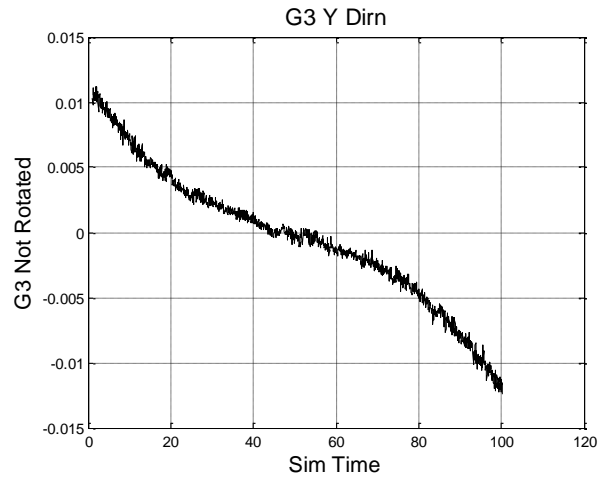


Figure 74: Y-Direction Skewness of Two-Axis Test

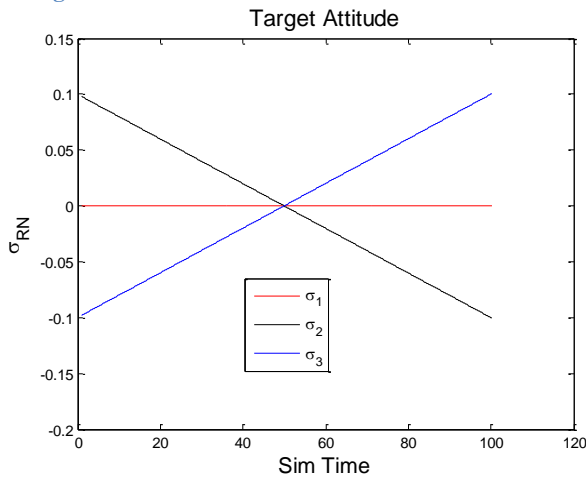


Figure 75: Simulated Target Attitude

4.2.3.7 Assumptions about the Target

For the purposes of these tests, the target has to be two-dimensional. Right now, this means a decal would have to be placed on a three-dimensional object that someone wants to track. This can be extended

in future research, but the proof of concept is to prove that the skewness value g_3 can be used as a measure of perpendicularity in a control. The targets can be expanded and the control made more complicated in future research. If the shape of the target is to be chosen, such as a decal that will be placed on the side of the object to be tracked, a shape in which the axes can be visually told apart should be chosen. A circle or a square should never be chosen. Rectangles are an ideal choice, especially for the statistical pressure snake algorithm. Choosing a symmetric target/decal is preferred because the reference skewness value for the control will be zero, but this is not necessary (this is shown in prior sections). Information must be known about the target before the control is initialized. Visual data is being used, so if the target is initially a complete unknown (approaching a comet that has not been viewed closely before), then use the visual data to model the target before starting the control. This is an investigation for future research. This research was based on the idea and proof of concept of skewness as a measure of perpendicularity.

To explain more thoroughly why information is required about the target before initializing the control, consider a circle viewed at varying skewness angles.

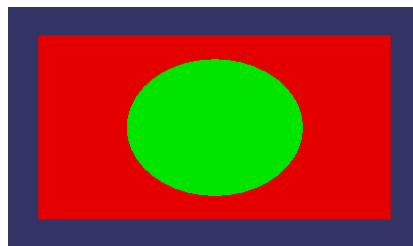


Figure 76: Circle Viewed from a Perpendicular Orientation

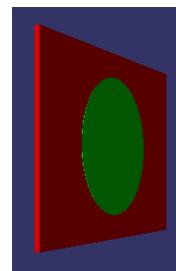


Figure 77: Circle Viewed from a Skewed Orientation looks like an Ellipse

Considering the elliptical picture on the right, there are two possibilities.

1. The target is an ellipse (not true since reality is known in the test case), or

2. The camera is viewing the target from a position such that the target appears to be an ellipse when it is a different shape in reality.

The skewness angle between the camera and target can make shapes appear distorted, or like other shapes.

If this concept is extended to any two-dimensional target, the following are true.

1. The target will have a reference skewness $g_{3\text{target}}$. The reference will be zero if the target axis shape is symmetrical.
2. The observed $g_{3\text{observed}}$ will not equal $g_{3\text{target}}$ unless the camera is viewing the target from a straight on orientation.
3. There is no way to know $g_{3\text{target}}$, and therefore no way to know if the camera is viewing the target perpendicularly using the skewness method, if information is not known about the target. Ex: is a circle or an ellipse being viewed? If it is a circle, it is being viewed from a skewed orientation. If it is an ellipse, it could be viewed from a straight on orientation, but even that is not known without the specific $g_{3\text{target}}$.

Chapter 5 Position Control

5.1 Camera Frame Setup

For the sake of consistency within the project, the same frame unit direction vectors were maintained as another researcher on the project created²⁰. The frame is not consistent with the field of image analysis, but as long as the frame is in the right-handed coordinate system and accounted for within the simulations, the frame does not matter. There are an infinite number of frames that could be chosen. Researchers in image analysis do not choose to work in this frame because it makes the positive camera screen quadrant similar to the second Cartesian quadrant, which is negative in one axis.

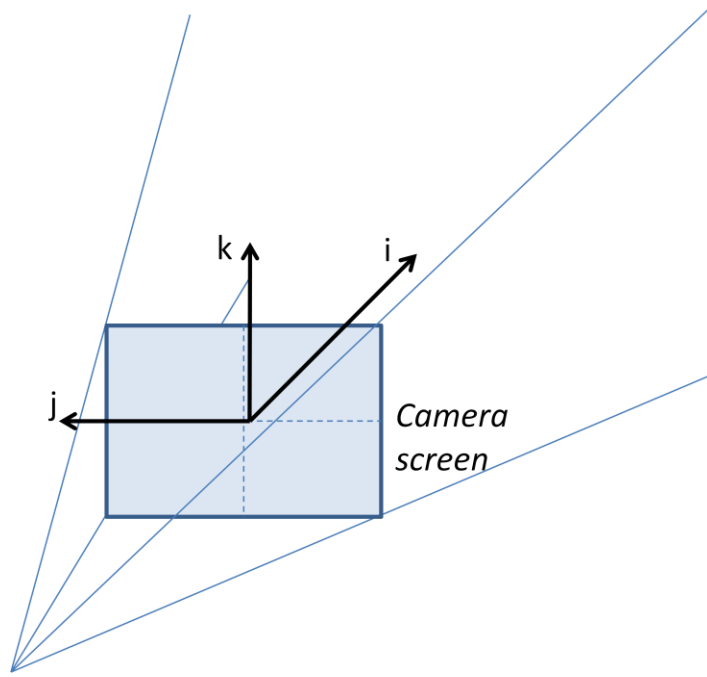


Figure 78: Camera Frame

The camera frame representation in equation form is the following.

$$C = \{\hat{i} \quad \hat{j} \quad \hat{k}\} \quad (5.1)$$

5.2 Full Position Control

The full position control incorporates both the skewness and distance controls, which control different axes. The position control is calculated in the camera frame within the position control module and

translated into the inertial frame by a frame converter module. The control output must be translated into inertial coordinates within the numerical simulation in order to propagate the camera's position. In a true maneuver, the control and external forces would act on the satellite and make it move. In a simulation, these forces must be simulated and propagated.

$$\bar{u}_{position} = {}^C \begin{bmatrix} u_{distance} \\ u_{skew_j} \\ u_{skew_k} \end{bmatrix} \quad (5.2)$$

5.2.1 Position Propagator

In order to propagate the motion of the body satellite, a position propagator is required. The body satellite obeys Newton's second law of motion in which the sum of the applied forces (\bar{F}) equals its mass (m) times its acceleration (\bar{a}).

$$\sum \bar{F} = m\bar{a} \quad (5.3)$$

For the deep space simulations, the gravity of being in orbit around a planet and any perturbations were not considered. The only applied force to the body for the proof of concept was the applied force from the skewness and distance controls ($\bar{u}_{position}$). For the proof of concept case, Euler integration was used to integrate the acceleration for velocity and integrate velocity to calculate position. Please see Appendix 8.5 for the module overview.

For future work, additional external forces should be included to create a more advanced simulation including gravity from orbital maneuvers around a planet, solar radiation pressure, and effects from being in orbit around a planet such as J_2 and drag.

5.3 Skewness Control Development

Initially, the control will start at some arbitrary skewness value g_3 .

The skewness control uses the skewness value and the time history of the skewness. The skewness control is only applied in the j- and k-directions of the camera frame. In order to propagate the resulting

motion of the body caused by the control forces, the body frame control forces must be rotated into the inertial frame. This rotation is done outside of the skewness control module—everything in the skewness module is calculated in terms of the camera body frame.

To derive a control for skewness, two assumptions were made relating to the slow, steady motion of the reference motion. For future research, a different Lyapunov function could be developed and tested for the skewness control.

The applied force of the control is approximately equal to the mass times the acceleration. This is a close approximation for smooth, slow motion that does not take gyroscopic effects and sharp motion into account. As the spacecraft moves, the skewness will change as a response. The rate of this change is assumed to be proportional to the rate of the spacecraft, and so the second derivative of the skewness is assumed to be proportional to the acceleration of the spacecraft. This does not take gyroscopic terms into account.

$$u_j \cong m\ddot{y} \cong m\ddot{g}_3 \quad (5.4)$$

The Lyapunov function is given below. Since a space application is being developed, an acceleration control must be created. The output of the control is a force, not a velocity. The camera should not drift and have velocity build up with the acceleration control, which is why the rates must also be driven to zero within the control.

$$V(g_3, \dot{g}_3) = \frac{1}{2}P_1g_{3j} + \frac{1}{2}m\dot{g}_{3j} \quad (5.5)$$

To prove stability, the derivative of the Lyapunov function must be calculated.

$$\dot{V}(g_3, \dot{g}_3) = \dot{g}_3(P_1g_{3j} + m\ddot{g}_{3j}) \quad (5.6)$$

Plug in the equations of motion.

$$\dot{V}(g_3, \dot{g}_3) = \dot{g}_3(P_1g_{3j} + u_j) \quad (5.7)$$

Force the Lyapunov function derivative to be negative definite by setting it equal to a negative definite function.

$$\dot{V}(g_3, \dot{g}_3) = \dot{g}_3(P_1 g_{3j} + u_j) = -P_2 \dot{g}_{3j}^2 \quad (5.8)$$

Solve for the control.

$$u_{skew_j} = -P_1 g_{3j} - P_2 \dot{g}_{3j} \quad (5.9)$$

The same stability proof holds for the k axis.

$$u_{skew_k} = -L_1 g_{3k} - L_2 \dot{g}_{3k} \quad (5.10)$$

The attitude control must be working at the same time as the skewness control. Remember: the attitude control is the most important control. Without the attitude control, the target would move out of sight of the camera and none of the other controls would work, and the whole point of getting data from the target would be a failure (mission failure). More importantly, the attitude control must be working to constantly re-aim the camera. The skewness control will move the target more or less in a sphere around the target—never getting closer or farther away—but this will only happen if the attitude control is functioning properly on top of the skewness control. This is an elegant solution in terms of robustness because it means operators do not need to worry about the body satellite crashing into the target.

Let's look more in depth as to why the skewness control only moves in a sphere. The skewness control will output a control in the \hat{j} direction. Imagine that you are standing with a target a few paces in front of you and that the skewness control force makes you take a step to your right ($-\hat{j}$ direction). Because you have taken a step directly to your right, the target is no longer directly in front of you. The attitude control will respond to this by rotating you directly where you stand to your left. Now you are pointed more toward the target. During the next skewness control, the control will cause you to take a step to your right again. Look at your feet. Because the attitude control caused you to rotate, your new step to

your right from the skewness control is going to be at an angle compared to the previous step to your right from the skewness control. Can you see that you'll be moving in approximately a circle around the target? If you imagine this same process in two directions, you'll be moving in a sphere around the target.

5.4 Distance Control Development

5.4.1 Pin-hole Camera Model

It is helpful to understand the pin-hole camera model to understand the development of the distance control. The pin-hole camera model is based on simple geometry.

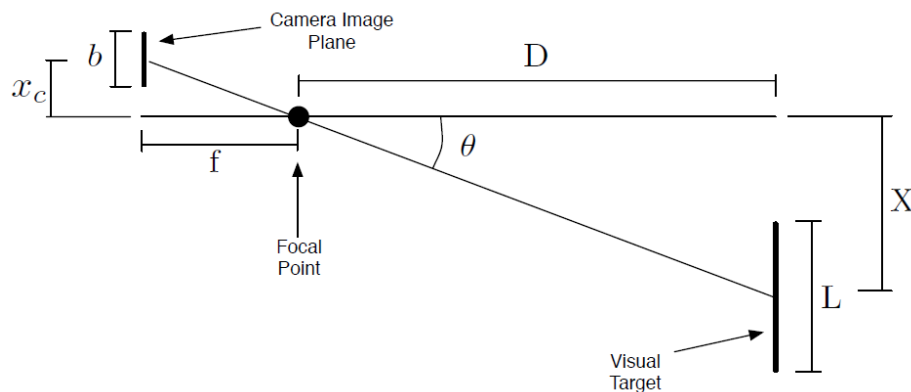


Figure 79: Pin-hole Camera Model⁶

In the pin-hole camera model, the horizontal line is the direction from the center of the camera screen straight out perpendicularly to the screen's plane. L is the length of the target, X is the distance from the horizontal direction line to the center of the target, D is the distance from the focal point to the target, f is the focal length, b is the length of the target as seen on the camera's screen (given by the statistical pressure snakes), x_c is the center of the target image as seen on the camera screen (also given by the statistical pressure snakes), and θ is the attitude error. The next image helps visualize what all of this means. The values of X and L correspond to the visual target in the pin-hole camera model or the big tree. The x_c and b values correspond to the little tree that shows up on the camera screen.

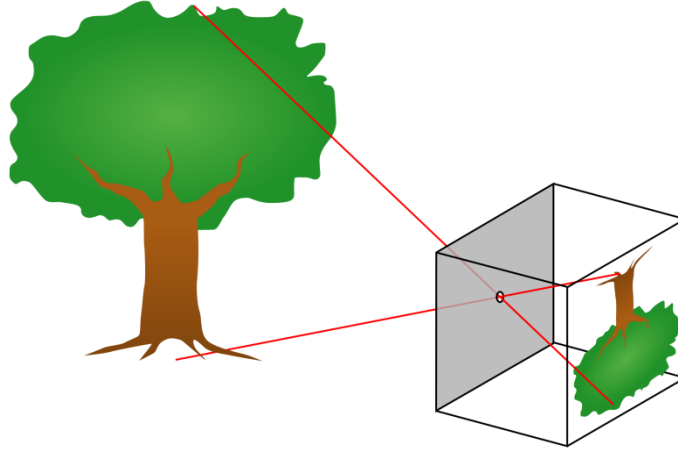


Figure 80: Pin-hole Camera Model Visualization⁷

Using the properties of triangles and the pin-hole camera model, the focal length can be calculated using the center of the image x_c and the attitude error θ .

$$f = \frac{x_c}{\tan\theta} \quad (5.11)$$

Using the pin-hole camera model, if similar triangles are created that start from the horizontal line and extend to the farthest edges of L and b, the resulting relation is:

$$\frac{D}{X + \frac{L}{2}} = \frac{f}{x_c + \frac{b}{2}} \quad (5.12)$$

$$D \left(x_c + \frac{b}{2} \right) = f \left(X + \frac{L}{2} \right) \quad (5.13)$$

Using the geometry of the pin-hole camera model: $X = D \tan\theta$. Plug the relation for X into the equation.

$$D \left(x_c + \frac{b}{2} \right) = f \left(D \tan\theta + \frac{L}{2} \right) \quad (5.14)$$

Solve for D.

$$D = \frac{\frac{fL}{2}}{x_c + \frac{b}{2} - f \tan \theta} \quad (5.15)$$

Plug in the relation for the focal length.

$$D = \frac{\frac{fL}{2}}{x_c + \frac{b}{2} - x_c} \quad (5.16)$$

Simplify the equation.

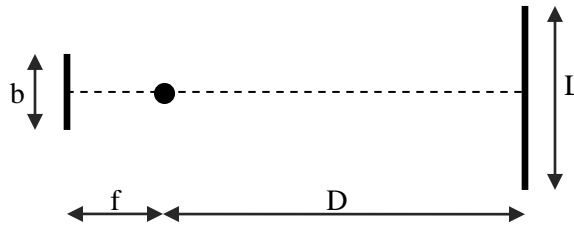
$$D = \frac{fL}{b} \quad (5.17)$$

The depth calibration factor γ changes for different distances D . Within the software, the data for b will come from the statistical pressure snakes.

$$\gamma = Db = fL \quad (5.18)$$

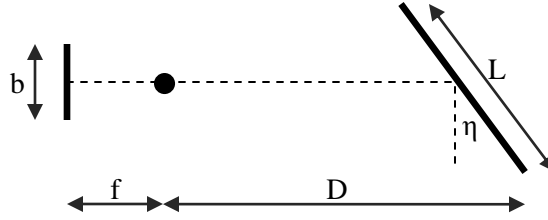
5.4.1.1 *Simple No Skew – No Attitude Error Camera Model*

In the following simplified camera model developed in the course of this work, there is no attitude error or skewness. The target is being viewed from a “straight on”, zero skewness orientation with the center of the target at the center of the camera screen.



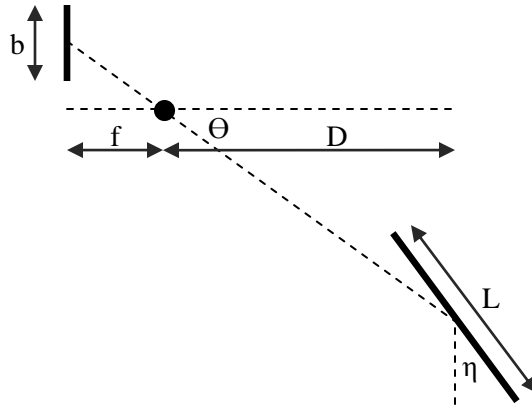
5.4.1.2 *Simple Skewed – No Attitude Error Camera Model*

The alternate camera model developed in this work shows no attitude error, but the target is skewed. This is the same setup used in Section 4.2.3.



5.4.1.3 *Skewed and Attitude Error Camera Model*

This alternate camera model developed in this work is more complicated than the simple pin-hole model because it shows both an attitude error and a skewed target. This is a much more realistic view. The controls developed will reorient the camera so the end result is the simple camera model in Section 5.4.1.1 with no attitude error, no skewness, and also with the camera maintaining a fixed separation distance from the target.



5.4.2 **Control in which Initial Distance is Known**

If the initial distance is known, the body satellite can be put at the initial distance, the statistical pressure snakes will provide b , and the calibration value γ will be calculated using $\gamma = Db$ for the entire simulation if that specific distance should be maintained.

The research on the topic develops a distance control based off of a calibration in which the distance is known, but this is not practical. A robotic assistant satellite in space is not going to unfurl a tape measure,

perfectly zero out skewness (which already requires two controls to be working), maintain a distance control in space (which already requires a distance control to be working), precisely measure how far away the body satellite (my robotic assistant in this example) is from the target, reel in the tape measure, do the calibration, and then start the new distance control. This does not make sense. There are other ways to measure distance like radar, but this requires an additional instrument to be launched with the spacecraft, another test that has to be done each time the control is used, and one more thing that could cause the system to fail.

The initial distance is never known in the proposed simulations, so an alternate distance control method has been developed. A lot of information can be gathered from a camera, including the information to build a distance control, so there is no reason to launch another instrument into space or assume a distance is known when it never would be.

5.4.3 Control in which Initial Distance is Not Known

The more realistic simulation is one in which the initial distance is not known. The control will have no idea how far away the body satellite is from the camera, but it will maintain that distance.

If the distance between the body and target is getting smaller (camera is getting closer to the target), the target will appear to get larger in the camera's screen, so the second moment the statistical pressure snake outputs will increase. If the distance between the body and target is getting larger (camera is moving away from the target), the target will appear to get smaller in the camera's screen, so the second moment from the statistical pressure snake will decrease. This relation assumes that the skewness is not changing and ideally zero. If the skewness is changing, it will affect the second moment. If the skewness is zero, the second moment will be larger than if the skewness is either positive or negative because it will make the shape seem smaller since the camera will be viewing the target at an angle. Since the skewness must be relatively constant, the skewness control must be working before the distance control. The skewness

control requires the attitude control to be working first, so the distance control essentially requires a three-tiered control simulation.

Instead of a calibration parameter based off of the initial distance, the calibration parameter is based off of an ideal second moment. If a simulation starts at an arbitrary skewness and the skewness control works to bring the target to the “straight on” or perpendicular orientation, the distance control will remain inactive until the skewness control brings the target to the “straight on” position. At that point, the control will save the second moment values that were sent from the statistical pressure snakes. These second moments (inertias) are the nominal calibration values for the control. As long as the skewness remains in control, if the second moment increases, the distance control knows the target is getting too close and responds by moving the camera backward; if the second moment decreases from the nominal, the distance control knows the target is too far and moves the camera forward. The control forces calculated by the distance control are in the camera body frame. Another module must rotate the control into the inertial frame before the control forces are propagated. I_2 is the current inertia value from the statistical pressure snakes, and $I_{2_{nom}}$ is the nominal calibration inertia. The distance control only affects the axis directly between the center of the camera screen and the center of the target.

$$dI_2 = I_2 - I_{2_{nom}} \quad (5.19)$$

To derive the Lyapunov control, an assumption must be made that the motion is smooth and steady. No sharp turns or gyroscopic effects will be taken into effect. A task for future research could be to refine the Lyapunov function and input equations of motion to account for higher order dynamics.

$$u_x \cong m\ddot{x} \cong md\ddot{I}_2 \quad (5.20)$$

The Lyapunov function is given below. Since the simulation is being created for a space environment, an acceleration control was developed. This means both the position and rate term difference terms must be driven to zero.

$$V(dI_2, d\dot{I}_2) = \frac{1}{2}K_1dI_2^2 + \frac{1}{2}md\dot{I}_2^2 \quad (5.21)$$

The derivative of the Lyapunov function must be calculated to determine stability.

$$\dot{V}(dI_2, d\dot{I}_2) = d\dot{I}_2(K_1dI_2 + md\ddot{I}_2) \quad (5.22)$$

Substitute the equation of motion into the Lyapunov derivative.

$$\dot{V}(dI_2, d\dot{I}_2) = d\dot{I}_2(K_1dI_2 + u_x) \quad (5.23)$$

Force the Lyapunov derivative to be negative definite by setting it equal to a negative definite function.

$$\dot{V}(dI_2, d\dot{I}_2) = d\dot{I}_2(K_1dI_2 + u_x) = -K_2d\dot{I}_2^2 \quad (5.24)$$

Solve for the control.

$$u_{distance} = -K_1dI_2 - K_2d\dot{I}_2 \quad (5.25)$$

5.5 Control Flow Chart

The following figure shows one option for a simulation control flow chart. The control blocks show the name of the control, if that control depends on any other controls, and what data the control depends on. Red blocks represent questions or criteria within the control, purple blocks correspond to possible outcomes, and green blocks are actions or calculations within the control.

The attitude control is not dependent on any other control. The attitude control can be thought of as the “master” control—as previously stated, without the attitude control, none of the other controls will function. More importantly, without the attitude control, the camera would not be able to view the target and the mission would be a failure. The attitude control depends on data from the camera. More specifically, the statistical pressure snake algorithm outputs the first moment of the target, which is the central pixel position of the target, and the attitude error estimator translates the pixel position into a two-

axis rotation that the attitude control attempts to match. In this control setup, the attitude control always runs, meaning no deadband is implemented.

The skewness control is dependent in the attitude control, which is discussed previously. Remember: this is what makes the skewness control move along an approximate sphere around the target. The skewness control depends on the camera data, specifically on the third moment in both axes provided by the statistical pressure snake module. The skewness control does have a deadband. If the skewness measure g_3 is not within the deadband, the skewness control will continue to drive g_3 toward the reference value. If g_3 is within the deadband, the skewness control will be turned off. If it is the first time the skewness measure has been driven within the deadband, the second moment will be saved as a nominal value and the distance control will be activated.

The distance control depends on both the attitude and skewness controls. The distance control depends on the camera data, specifically the second moment (inertia) provided by the statistical pressure snake module. The distance control also depends on the nominal inertia value, which means the distance control cannot be initialized until the skewness control has driven the skewness measure g_3 into the deadband the first time.

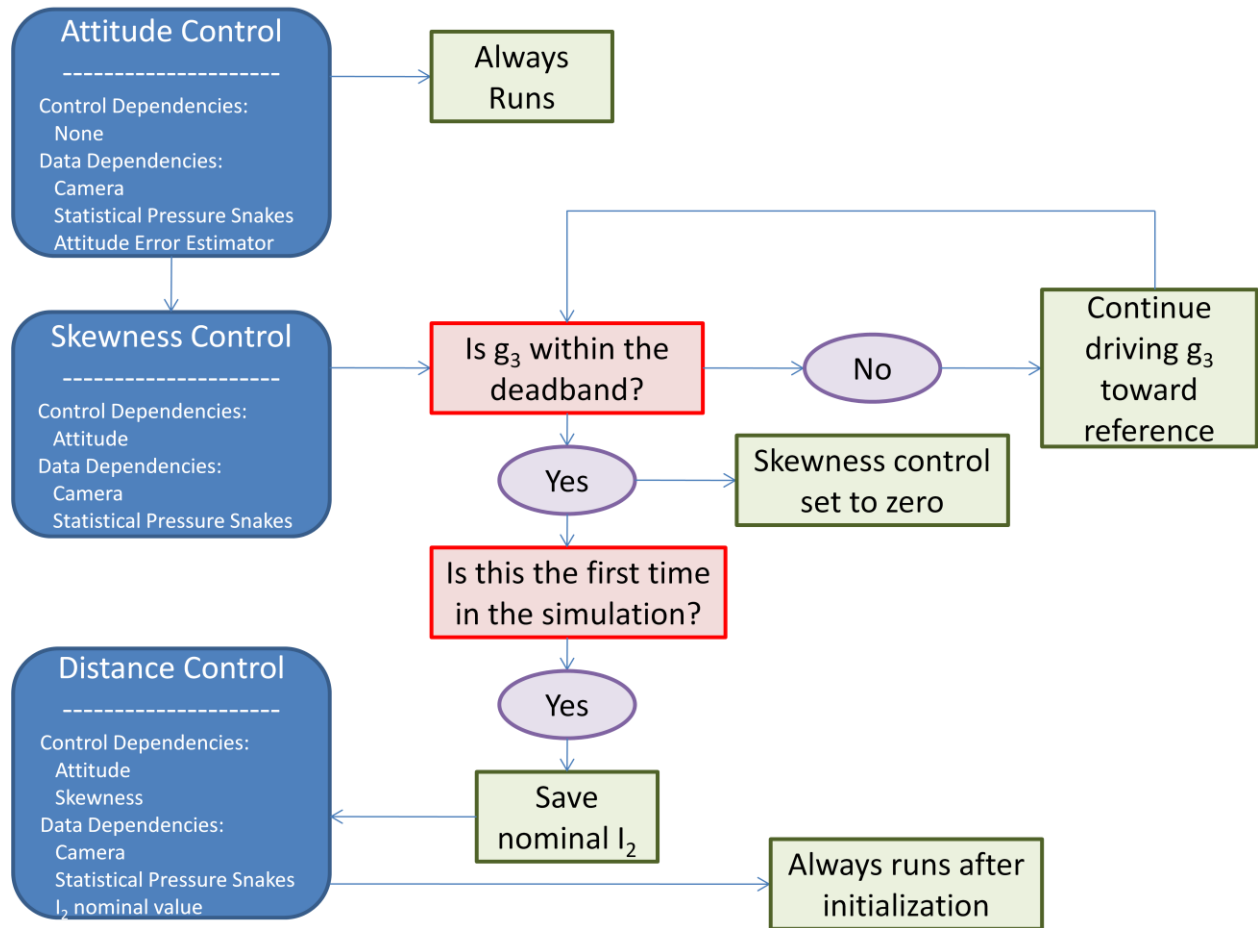


Figure 81: Control Flow Chart

Chapter 6 Autonomous Simulation

6.1 Simulation Overview

The overall simulation starts at some arbitrary skewness for a target that is known (like the rectangle).

The reason the target has to be known is the skewness profile for different orientations and skewness value at the perpendicular orientation must be known initially. If the mission objective is to observe a previously unknown asteroid or shape, then use the camera to retrieve the initial shape data, model the shape, and find the skewness profile before starting the control (Section 4.2.3.7). Having the skewness profile is very important because 1) it determines the reference value of the control, and 2) it determines if this control, a different control, or if any control is possible for the given shape.

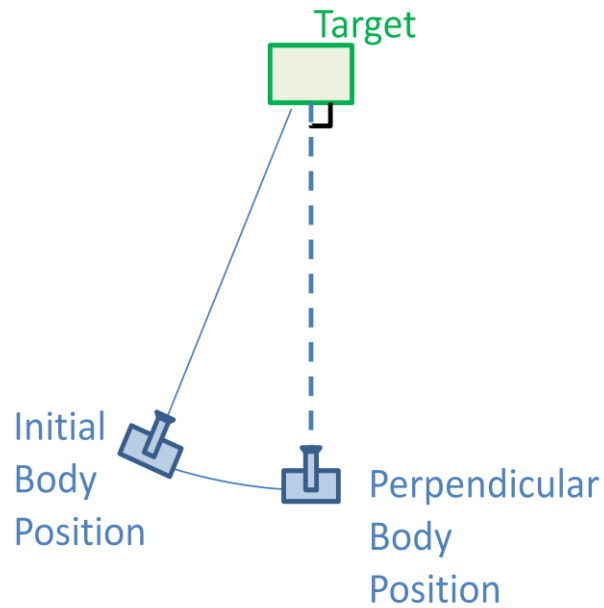


Figure 82: Simple Simulation Skewness View

The simulation then drives the skewness to the reference value (zero for symmetric axes). Once the skewness control reaches the deadband, the maximum value for the inertia is stored as a calibration for the distance control, which engages to maintain the separation distance between the body and target.

After this point, if the skewness value leaves the deadband, both the distance and skewness controls act in concert. Remember: they control different body axes.

The following figures show the visual representation of the simulations from UMBRA. The initial inertial view and initial camera view show the camera viewing the target from a skewed angle. As the controls work, the final camera view shows the camera viewing the target from a “straight on” or perpendicular orientation.

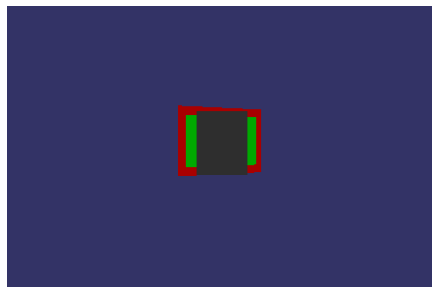


Figure 83: Initial Inertial View

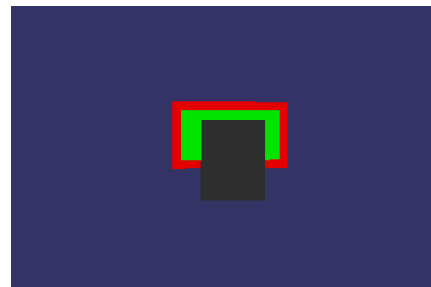


Figure 84: Final Inertial View

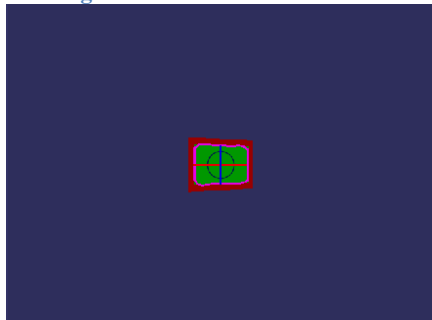


Figure 85: Initial Camera View

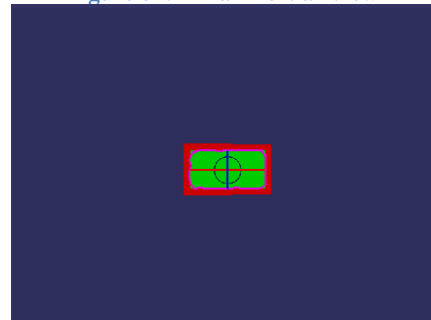


Figure 86: Final Camera View

6.2 Module Setup

The simulation starts with Frames.Virtual. The output of Frames.Virtual goes to a few different places: two frames named RectDemoFrame and CamDemoFrame, and one frame converter.

Next is the branch that deals with the target. Frames.Virtual’s output is connected to RectDemoFrame’s input. The Simulated Target Attitude module generates the simulated MRPs, and Rotation Converter and PRJoiner modules are needed to get the MRP at each time step into the proper format to be connect to RectDemoFrame’s offset. Then, RectDemoFrame’s output is connected to the target frame’s offset. All of this allows the target to have the attitude specified by the Simulated Target Attitude module.

The next branch is more complicated. It deals with the camera, the data flow of the camera, the controls that are built off of the data, and the movement that the controls want the camera to make.

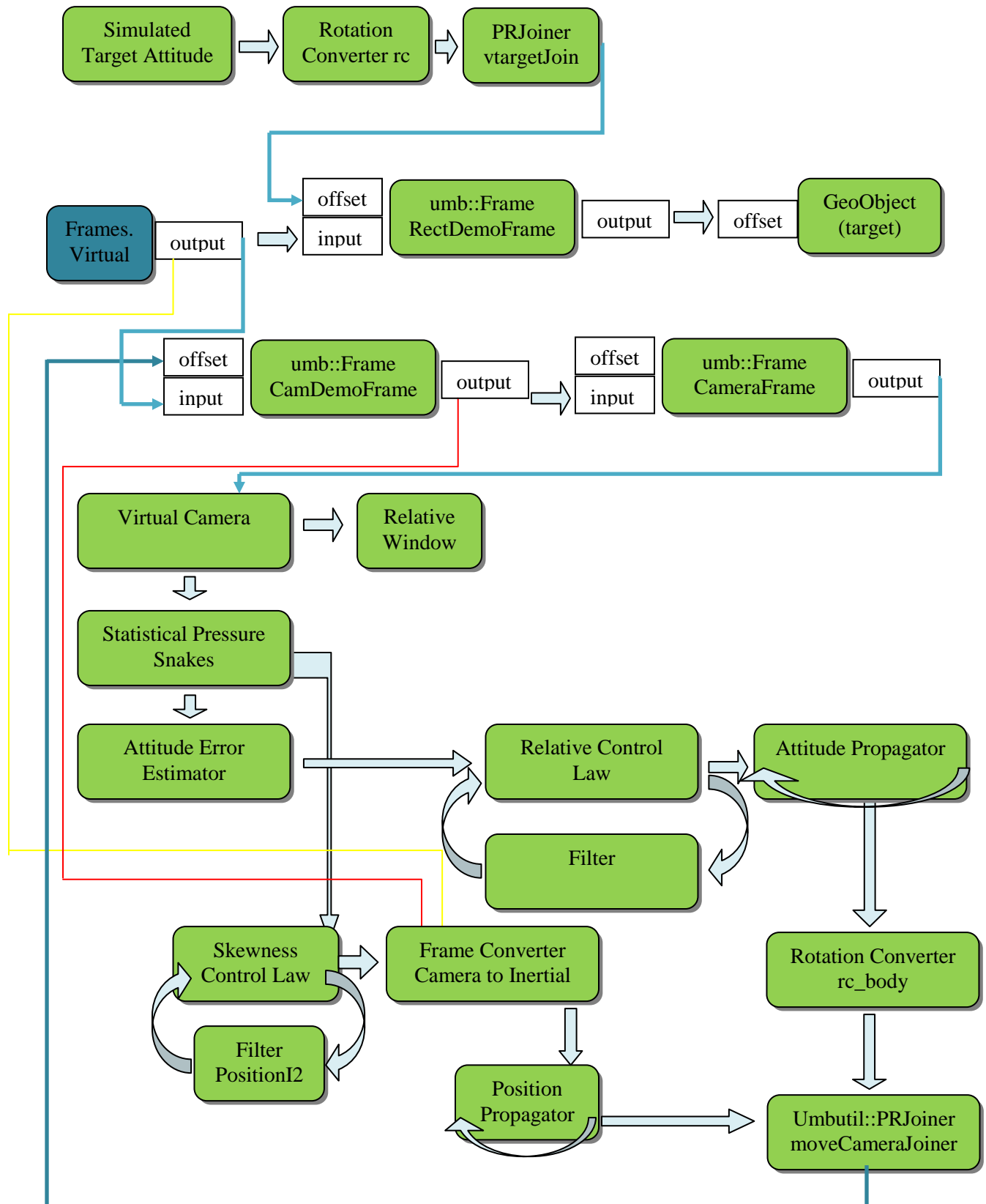
Frames.Virtual's output is connected to CamDemoFrame's input. CamDemoFrame's output is connected to the CameraFrame's input. The CameraFrame is connected to the Virtual Camera.

The output of the Virtual Camera's window frame is connected to the relative window in the GUI to ensure that the relative window in the GUI is seeing what the camera sees during a simulation. The Virtual Camera is attached to the Statistical Pressure Snake module so the camera data can be used to calculate moment information and track the target (assuming the target is in the camera's field of view). The Statistical Pressure Snake module is connected to the Attitude Error Estimator so the relative attitude can be calculated from the moment information (MRPs from the center of image pixel coordinate). The Attitude Error Estimator transfers the relative attitude to the Relative Control Law module, which needs a filter to deal with the noisy differentiated MRPs that lead to calculating the relative angular velocity. The output control of the Relative Control Law module goes to the Attitude Propagator module, which propagates the camera/body attitude. This part of the branch works to control the relative attitude; basically, it tries to keep the center of the target in the center of the camera's screen.

Another of the camera's sub-branches deals with the position controls, both skewness and distance. Data from the Statistical Pressure Snakes module is sent to the Skewness Control Law module. (Even though it is named the Skewness Control Law module, it is really a position control module since it contains all of the position controls, both skewness and distance). The Skewness Control Law module is connected to a filter that feeds back to the control module because some of the data needs a filter. The Skewness Control Law module sends its control output vector to a Frame Converter module, because the Skewness Control Law module calculates its control in camera/body frame components, whereas the Position Propagator must use inertial coordinates in order to not violate Newton's laws. The Frame Converter uses CamDemoFrame as its Source and Frames.Virtual as its Destination. The Frame Converter sends the converted control, now in inertial coordinates, to the Position Propagator module. This part of the

branch works to control the relative position; basically, it tries to keep the camera in the “straight on” or zero skewness configuration while keeping a fixed separation distance from the target.

The body attitude goes to a Rotation Converter module to get the MRPs in another format and then goes to the rotation input of the PRJoiner moveCameraJoiner. The inertial position output from the Position Propagator module is the position input of moveCameraJoiner. The PRJoiner moveCameraJoiner is a feedback connector all the way back to the offset of the frame CamDemoFrame because body attitude and body inertial position make up in information needed to construct the camera/body frame.



6.3 Example Simulation

The following shows a simulation that started with the target offset at a skewness of 9.16^0 (relative MRP of 0.04) from the camera. The skewness control first drives the skewness down within the deadband of 0.458^0 . The position control engages once the skewness control first enters the deadband. The skewness control does not apply a control within the deadband, so the skewness will drift within the deadband. When the skewness drifts out of the deadband, the skewness control will re-engage and work simultaneously with the position control to bring the skewness back within the deadband.

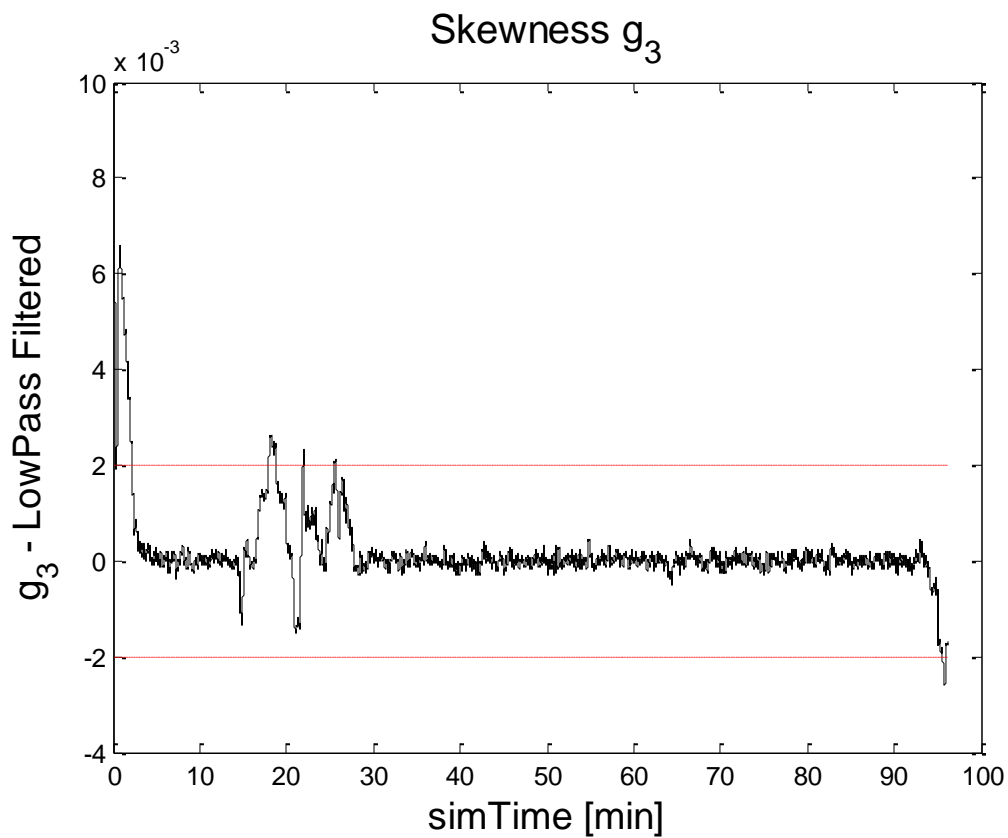


Figure 87: Skewness Control around Deadband

The relative attitude is stable, and the relative distance between the camera and target varies less than 1% throughout the 100 minute simulation. The unfiltered inertia (second moment) varies initially as the skewness is pushed within the deadband and the position control works to converge, but then the inertia remains noisy around a constant value.

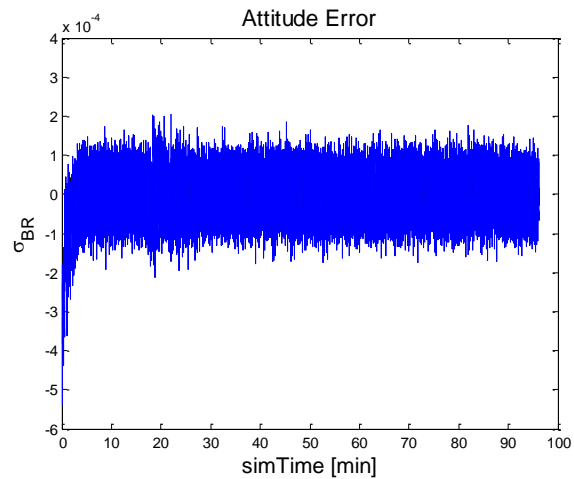


Figure 88: Relative Attitude around Deadband

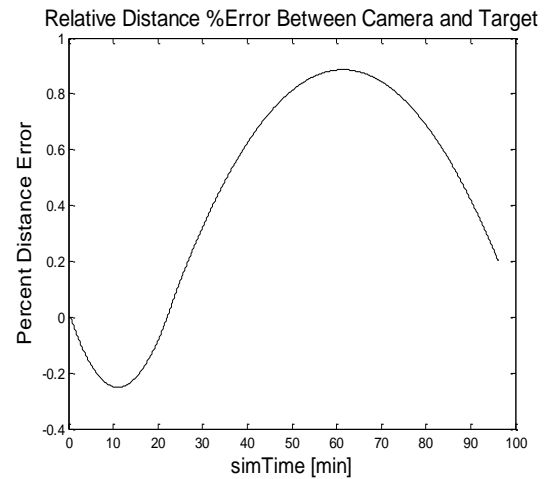


Figure 89: Distance between Body and Target [m]

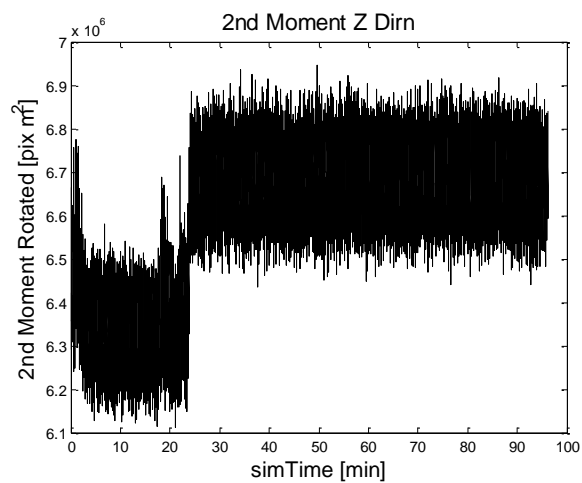


Figure 90: Unfiltered Inertia (2nd Moment)

Chapter 7 Summary

Once again, consider the robotic assistant satellite outside of the ISS with an objective of observing a specific panel.

As previously discussed, the first step of the robotic assistant satellite achieving its objective is to have a visual tracking algorithm in order to tell the difference between the object to be tracked, such as a decal on the panel, and everything else that constitutes the background of the image. In this research, the statistical pressure snake algorithm is used for visual tracking. The statistical pressure snake algorithm, tracks the outside contour of a target by hue, does not assume anything about the shape of a target, is robust against lighting conditions, and even functions with partial obscuring of the target. The statistical pressure snake module also calculates the geometric moments of the target.

It is important to not leave the visual tracking algorithm step out of the control process. Many current missions do not have a visual tracking algorithm. The STARDUST Wild 2 encounter did not have a visual tracking algorithm, which caused issues with the calculation of the central pixel of the comet. The newly proposed AutoNav system uses landmarks based on surface mapping to compare pre- and in-flight pictures. The new AutoNav system is a step in the right direction because it identifies something in an image, but it does not actively identify the target in real time, track the target in real time based off of incoming data, and feed that data into the control. The data is only used as a comparison to pre-flight, inertial surface mapping, and therefore not productive toward relative control techniques. The visual sensing algorithm developed for the autonomous helicopter is very limited. It only works for specific shapes and lighting conditions, and is based on an active, inertial control with visual input. It would be interesting for the researchers to test if the visual algorithm is necessary with all of the inertial information included in the control (height of the window, latitude, longitude, gyroscope, accelerometer, GPS, etc.). Even if the visual sensing algorithm is limiting and not robust against various conditions, it is promising that other researchers are looking into similar algorithms.

By using the statistical pressure snake algorithm, the bulk of the noise from tracking a target that occurs around the contour of the image is canceled out, which allows for a more constant central pixel moment calculation. If future missions incorporate a visual tracking algorithm step, the center finding results will be improved. The issue is not the moment calculation, but rather the noisy data being input to the moment calculation since the visual tracking algorithm, which provides for a more-constant central pixel, is being skipped.

The second step of the robotic assistant satellite achieving its objective is to have an algorithm that allows it to continually point at the ISS panel or target on the ISS panel. In this research, the statistical pressure snake algorithm provides the first moment calculation, which is the central pixel of the target. The attitude error estimator translates the central pixel of the target into a two-axis rotation that will align the camera and target. Since the resulting snake control points from the statistical pressure snake are used to calculate the first moment and this provides for a more constant center, the resulting rotation did not need to be filtered. For smooth, steady, orbital motions without gyroscopic effects, the relative attitude and relative angular velocity is enough to produce a stabilizing control. To model the full dynamical motion with gyroscopic effects, the target's inertial information can be estimated from the relative and body inertial information, so the target can remain passive.

Other missions skip the visual tracking algorithm stage, so the data input to the first moment calculation is exceedingly noisy and the resulting center location is inaccurate. Missions such as Voyager 2 used human "eye-ball" input to mark the center of objects on an image, used that as *a priori* information for a filter, and then iterated to find a better solution for the central pixel. The STARDUST Wild 2 encounter skipped past the visual tracking algorithm stage and proceeded straight to the center finding calculation. Because of that, the mission had issues with noise, lighting, geometry, etc. The new AutoNav system will have similar difficulties because the noise of the image for center finding will be put into the first moment calculation just like the other methods. Using a visual tracking algorithm would reduce the noise around the boundary of the image and therefore reduce the noise in the calculated central pixel.

Another step in achieving the mission objective is to have an algorithm that recognizing the skewed orientation of the camera with respect to the target. Information must be known about the target before initializing the control. This control is based off of the third moment and can be used to either steer the camera/body satellite to a position where it is viewing the target from a perpendicular orientation, or it can maintain the initial skewness angle of the camera with respect to the target. This research is infinitely better than anything used in current missions since skewness is not currently used as a measure for a control. Current missions by comets, for example, carry the satellite along an orbit by the comet and re-point the satellite toward the comet for observations (attitude problem) as the satellite's orbit carries it near the comet. An active position control that maintains the satellite's orientation relative to the comet is not used in that case. For smaller satellites near Earth, such as the robotic assistant satellite, position controls to maintain orientation using skewness are also not used.

The final aspect for achieving the robotic assistant satellite achieving its object is the algorithm to maintain a nominal distance between the camera and target. The method developed in this research is important because it only uses visual information from one camera (second moment), does not require additional equipment to be launched, and does not assume impractical information to be known. Other resources^{1,6} assume the satellite would have to be maneuvered to the ideal position, held in a constant position while the distance between the camera and target is measured, calibrated, and then the distance control can be initiated.

Passive, relative, visual control techniques are an incredibly important investigation for future satellite missions. Passive, relative controls are important because they allow the satellite to process its own data and create a simpler system that depends on fewer external data and events that could fail. Visual control techniques are important because a seemingly endless amount of information can be gained by the data in images.

Bibliography

- [1] Mukundan R., and K. R. Ramakrishnan. Moment Functions in Image Analysis: Theory and Applications. World Scientific. 1998.
- [2] Schaub, Hanspeter, and John L. Junkins. Analytical Mechanics of Space Systems. 2nd ed. American Institute of Aeronautics and Astronautics Education Series. 2009.
- [3] Luftig, Jeffrey. "EMEN 5005 Lecture 2 Descriptive Statistics." University of Colorado – Boulder. Fall 2010.
- [4] MVPStats Help Pages. "Skewness/Kurtosis." < <http://www.mvpprograms.com/help/common/distributions/SkewnessKurtosis> >
- [5] Shutler, Jamie. "Statistical Moments." Department of Electronics and Computer Science, University of Southampton, United Kingdom. < http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/node7.html > 15 Aug. 2002.
- [6] Southward, Charles M. "Autonomous Convoy Study of Unmanned Ground Vehicles using Visual Snakes." Master's Thesis. Virginia Polytechnic Institute and State University, 2007.
- [7] Wikipedia. "Pin-hole Camera Model Picture." < <http://en.wikipedia.org/wiki/File:Pinhole-camera.svg> >
- [8] Ksgamedev. "Field of View Diagram." < <http://ksgamedev.files.wordpress.com/2010/01/fov-diagram.png> > 3 Mar 2011.
- [9] Jet Propulsion Laboratory. < <http://stardustnext.jpl.nasa.gov/mission/images/HRI.jpg> >
- [10] Riedel, J. Ed, Tseng-Chan Wang, Roberg Werner, Andrew Vaughan, David Myers, Nickolaos Mastrodemos, Geoffrey Huntington, Christopher Grasso, Robert Gaskell, David Bayard. "Configuring

the Deep Impact AutoNav System for Lunar, Comet, and Mars Landing.” AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, Aug. 18-21, 2008

[11] NASA, JPL photo credit. < http://solarsystem.nasa.gov/multimedia/display.cfm?IM_ID=2087 >

[12] Vaughan, R. M., J. E. Riedel, R. P. Davis, W. M. Owen, Jr., S. P. Synnott. “Optical Navigation for the Galileo Gaspra Encounter.” AIAA/AAS Astrodynamics Conference, Hilton Head Island, SC, Aug 10-12, 1992, Technical Papers (A92-52051 22-13). Washington, American Institute of Aeronautics and Astronautics, 1992, p. 361-369.

[13] Jet Propulsion Laboratory. < <http://voyager.jpl.nasa.gov/science/planetary.html> >

[14] Synnott, S. P., A. J. Donegan, J. E. Riedel, J. A. Stuve. “Interplanetary Optical Navigation: Voyager Uranus Encounter.” Astrodynamics Conference, Williamsburg, VA, August 18-20, 1986, Technical Papers (A86-47901 23-13). New York, American Institute of Aeronautics and Astronautics, 1986, p. 192-206. NASA-supported research.

[15] Mejias, Luis and Saripalli, Srikanth and Campoy, Pascual and Sukhatme, Gaurav

(2006) Visual Servoing of an Autonomous Helicopter in Urban Areas Using Feature

Tracking . *Journal of Field Robotics* 23(3-4):pp. 185-199.

[16] Fehse Wigbert. Automated Rendezvous and Docking of Spacecraft. Cambridge University Press. 2003.

[17] Bhaskaran, Shyam, Nick Mastrodemos, Joseph E. Riedel, Stephen P. Synnott. “Optical Navigation for the STARDUST Wild 2 Encounter.” Proceedings of the 18th International Symposium on Space Flight Dynamics (ESA SP-548). Jointly organised by the German Space Operations Center of DLR and the European Space Operations Centre of ESA. 11-15 October 2004, Munich, Germany., p.455

[18] Boeing.com < http://www.boeing.com/defense-space/space/bss/factsheets/376/syncom/antenna_n.jpg >

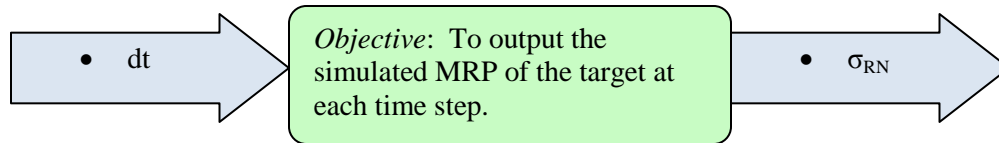
[19] Space Today < http://www.spacetoday.org/images/Sats/MilSats/DSCS_SatInSpaceLockheedMartin.jpg >

[20] Dunn, Daniel. "A Hybrid Hardware and Software Simulation Environment for Relative Orbit Motion Studies." Master's Thesis. University of Colorado at Boulder, 2011.

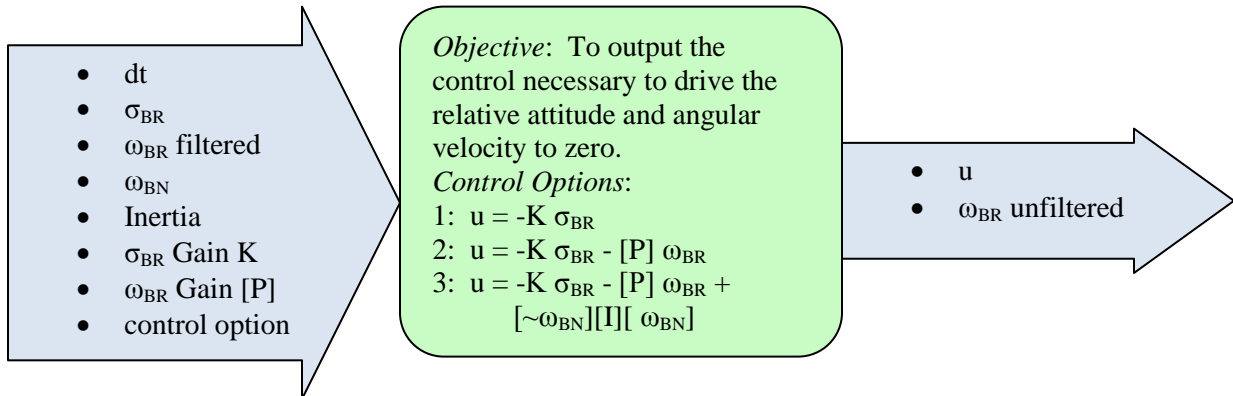
[21] Winer, B. J. Statistical Principles in Experimental Design. 2nd ed. McGraw-Hill, Inc. 1971.

Chapter 8 Appendix – Modules

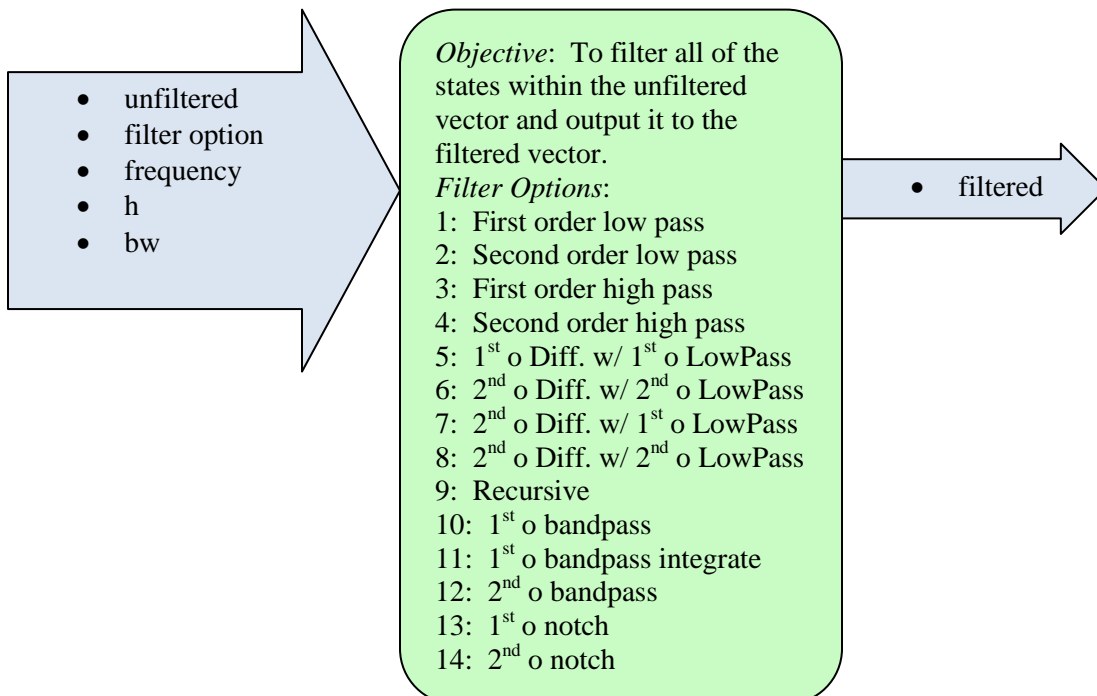
8.1 Simulated Target Attitude



8.2 Relative Control Law

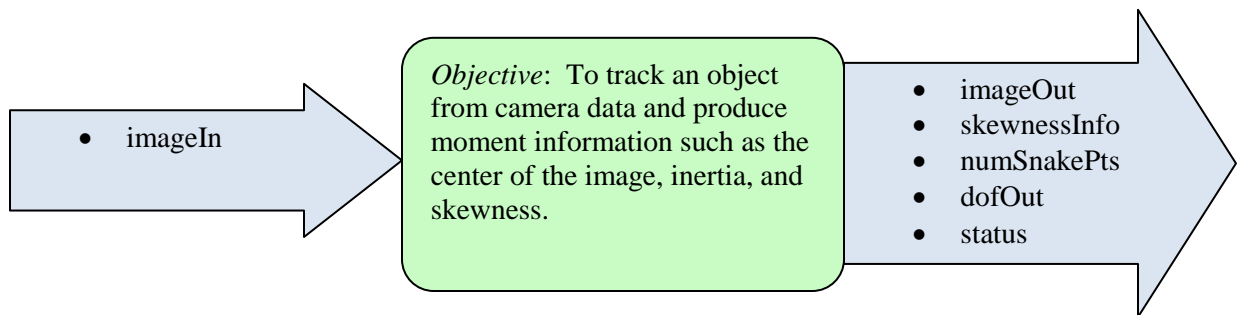


8.3 Filter

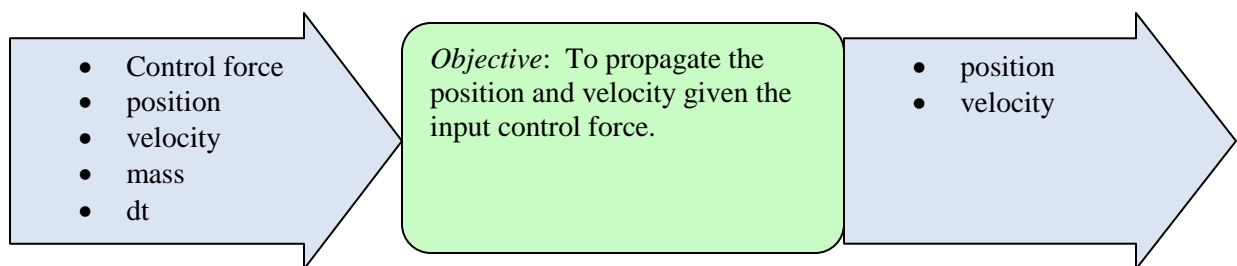


8.4 Statistical Pressure Snakes

The statistical pressure snake module and supporting files were already created and functioning before this project began. More information was needed from the statistical pressure snakes, so more outputs were extracted from the program and a few of the supporting files and functions were changed. Not all of the inputs and outputs are listed here; only the connectors that correspond to the tracking modes dealt with in this thesis, mainly rectangle tracking. For complete information, please see the statistical pressure snake documentation.



8.5 Simple Position Propagator



8.6 Skewness Control Law

