Teaching Satellite Attitude Controls and Mission Operations using Satellite Attitude Lab

Shannon Tracy[1], Elijah Howerter[2], Ian Swies[3], Dean Simpson[4], Julia Law[5], Eiade Ali El Wekeil[6] and Chasyn Beachley[7]

Florida Institute of Technology, Melbourne, FL, 32901, USA

It is often difficult for undergraduate and high school students to grasp the concept of how the attitude of a spacecraft is controlled in space when they are being introduced to spacecraft mission controls for the first time. The Satellite Attitude Lab (SAL) consists of a setup for a lab and three software packages that allow students to explore mission control operations in a controlled and simplified environment. Integration of COSMOS by Ball Aerospace, MATLAB, and STK allows for effective commanding and data collection of an educational CubeSat (ESAT). An ESAT is placed on an air bearing inside of a dark environment with a sun simulator pointed at it. There is a window so students can see into the environment and observe how the ESAT behaves when a command is sent to it. A camera acts as the payload of the ESAT allowing it to perform a "mission". The data from the camera is fed to MATLAB computer vision program which identifies a target which the ESAT points at and follows as it moves along the inside of the environment. A visualization of the ESAT in orbit is displayed on a computer monitor to simulate how it would look in a mission control center. Students can perform targeting, detumbling, and sun safe missions by sending commands to the ESAT in addition to analyzing the telemetry that is sent from the ESAT as it responds to commands. This may aid in the understanding of how satellite attitude controls work and how missions are performed.

I. Nomenclature

ESAT	=	Educational Satellite by Theia Aerospace
CC	=	Communication Computer
EB	=	ESAT Environmental Box
P	=	Power
t	=	Time
σ	=	Stefan Boltzman constant ($5.87\times 10^{-8}\,\frac{J}{sm^2k^4}$)
e	=	Emmissivity of solar panels
T	=	Temperature of source
T_L	=	Temperature of solar panel
A	=	Area of sun simulator

[1] Undergraduate student, Florida Institute of Technology, AIAA Student Member.
[2] Undergraduate student, Florida Institute of Technology.
[3] Undergraduate student, Florida Institute of Technology.
[4] Undergraduate student, Florida Institute of Technology.
[5] Undergraduate student, Florida Institute of Technology.
[6] Undergraduate student, Florida Institute of Technology.
[7] Undergraduate student, Florida Institute of Technology.

II. Introduction

Satellites are responsible for providing information that humans rely on daily. Weather, GPS, communication, and military operations are just a few of the functions that rely on information from satellites. Satellite attitude can be a challenging topic to master for some students. A hands-on learning approach can be a solution to help students better understand satellite operations and attitude control. This lab can also be used in high schools to introduce satellite operations. Using this lab in a high school setting can introduce students to the spacecraft design career path that is available with an aerospace engineering degree.

There is a need to upgrade the capabilities of the satellite laboratory and the range of ESAT experiments within the Aerospace Laboratory at the Florida Institute of Technology. The lab is not set up permanently and the experiments that can be performed are quite limited. Currently, the lab is solely based around attitude control and does not give students much exposure to mission operations that are seen in the real world. The goal of this project is to create missions that more closely simulate real-life missions for students to complete, and to decrease the amount of error in solar angle measurement that the current lab setup has. This aims to get more students interested in satellite missions and gives them a taste of how satellites work. The upgraded satellite lab should be permanently set up in the lab and have two to three timed missions that challenge students to orient the satellite to look at a certain point, and to take pictures of the room to find objects.

III. Current Laboratory Setup

The current set-up in the aerospace laboratory includes an ESAT, air bearing, sun simulator, magnets, static blanket, communication system, and monitor to display telemetry. The ESAT is currently set on an air bearing and all the overhead lights are turned off in the room. The monitor displays live telemetry that the ESAT is sending to the communication computer (CC). The ESAT sits on top of the air bearing allowing it to turn freely on one axis with as little friction as possible. The laboratory setup is shown in Figure 1 below.

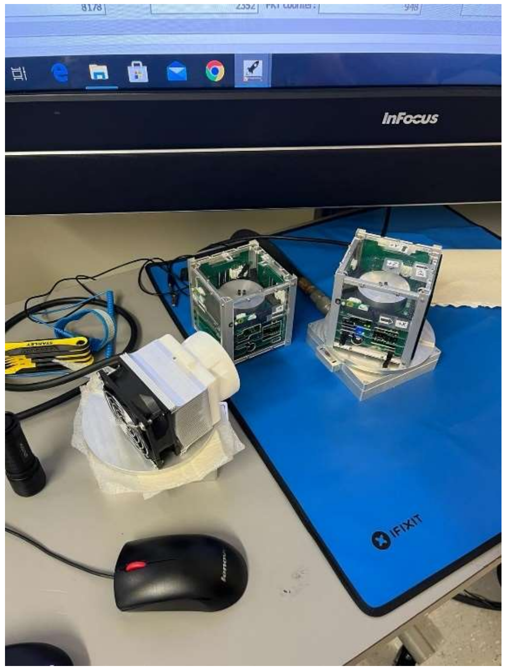


Fig. 1 Current Laboratory Setup

The ESAT itself is a 1U (10mm × 10mm × 10mm)CubeSat that has been simplified for educational purposes. It is shown below in Figure 2.

Fig. 2 ESAT by Theia Space [1]

The Attitude Determination Control System (ADCS) includes one reaction wheel and two magnetorquers. These allow the ESAT to change its attitude. The ESAT is equipped with a PID controller that that activates the reaction wheel and/or the magnetorquers depending on the commanded attitude and the commanded actuators sent by the user. The EPS board is responsible for supplying the ESAT systems with power. The battery can be charged using the two solar panels that are installed on the side, or by plugging it into the wall via a charger. The communication software that the ESAT uses is COSMOS by Ball Aerospace. This program allows the ESAT to send live telemetry to the program and for commands to be sent to the ESAT by the user.

The main issue with the current ESAT setup is that there is a large error in the solar angle that is measured when it is commanded to point at an object (±10°). This is due to light pollution from the monitor and the computer that is part of the communication system is seen by the sun sensors on the ESAT and gives an inaccurate reading for the attitude and causes it to oscillate back and forth.

Currently, there are a few exercises that must be performed by the student during the lab. These include: detumbling the ESAT from a spin, pointing the solar panels at the sun simulator (referred to as "sun safe attitude"), and following a target light source as it moves. Another problem with the current lab is that users perform exercises/mission that do not necessarily relate to how an actual satellite mission is performed. To remedy this, a second lab mission will be added to demonstrate more closely how satellites operate when in orbit.

IV. Improved System Overview and Laboratory Exercises

The solution to the light pollution problem is to put a dark environmental box (EB) over the ESAT setup to reduce any excess light from sources other than the sun simulator. A tinted plexiglass window is installed to allow students to observe how the ESAT behaves within the EB while commands are being sent. A light cone is also installed to the side of the EB to more effectively direct the light from the sun simulator onto the solar panels/solar sensors. This also reduces excess light pollution. This reduces the error in pointing the ESAT to (±1°). The EB will be able to be broken down into sections for ease of storage. The design of the EB will be discussed in further detail in Section 5.

There are two laboratory missions in the improved lab design. The first (known as the "detumble mission") is quite similar to the existing lab exercise where the ESAT is detumbled from a spin and goes into sun safe attitude. A second mission (known as the "Target finding mission") is added to demonstrate how satellites can operate autonomously to perform missions. Both missions will be modeled virtually using ANSYS Systems Tool Kit (STK). The target finding mission autonomously targets an object on the inside of the EB include following a target as it moves around the box by using a Raspberry Pi camera and a computer vision system to send commands to control the attitude of the ESAT. The Raspberry Pi camera is referred to as the payload because it is not part of the ESAT bus. This part of the lab will be commanded by the user once and then they can observe how the ESAT performs the mission by itself. A MATLAB program that utilizes the Computer Vision Toolbox (CVT) is used to relay data from the ESAT telemetry in COSMOS to MATLAB and then to STK. More information on the usage of STK and MATLAB within the system will be discussed in Section 5. In the real world, satellite mission operations have a time window where the satellite is in contact with the ground station on Earth. This window usually begins when the satellite comes up over the horizon and closes when it sets over the other horizon. While the time window in the real world is usually only a few minutes long, there will be a slightly longer time limit imposed on each the lab mission to simulate this. The lab will have three "difficulty levels" of time windows for each mission, 10, 20-, and 30-minute windows. The purpose of difficulty levels in the lab is to allow more time for beginning students to grasp the concept and to not be too overwhelmed by the time constraint. Shorter time windows make it more difficult to complete the mission on time and are intended for students with a good understanding of satellite attitude controls.

Before the lab begins, students will be given a brief introduction to satellite attitude controls and the programs they will be using during the lab exercises.

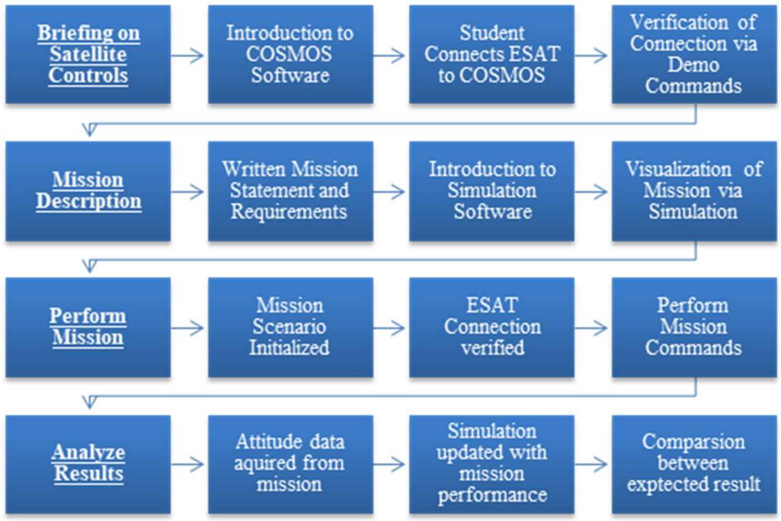


Fig. 4 Functional Flow Block Diagram for Overall Lab Experience

V. Subsystem Design

The integration of each subsystem is crucial to the success of this design. Communication of COSMOS, ESAT, MATLAB, and STK needs to be established in order for each program to run as intended. The simulation and sensor subsystems work quite closely together. The simulation subsystem is responsible for creating graphics in STK. This subsystem also establishes communication between COSMOS and STK and MATLAB. The sensor subsystem is responsible for the MATLAB code for computer vision analysis of a specified target. The purpose of this subsystem is to integrate computer vision to command the ESAT and to demonstrate how it points at and follows a target. This subsystem shall receive and process telemetry using the MATLAB program, and then output a command if needed to have the ESAT follow a target. The hardware subsystem impacts the accuracy of the sensor and simulation subsystems because the dark environment reduces the error in the data that is sent to the other respective subsystems. The Raspberry Pi camera, external battery pack, and laser pointer provide the inputs that make this possible MATLAB code in the sensor subsystem possible. The hardware subsystem is mainly concerned with keeping as much excess light away from the ESAT as possible using the EB. Brackets for holding the Raspberry Pi camera onto the top of the ESAT, sun simulator cone, and height block for the sun simulator are included in this subsystem as well. The Hardware Subsystem is important for data collection that will be used by the simulation and sensor subsystems. All actions in this lab will be compatible with the satellite simulators, as the missions will be based around controlling these devices. The ESAT uses COSMOS for control, which will be integrated with MATLAB and STK. The ESAT is limited as to what hardware can be attached. It is important to note that each subsystem interfaces wirelessly with the others because the ESAT must be able to spin freely about its axis. This is so there is not a bunch of wires to get tangled.

A. Hardware Subsystem

i. Environmental Box

The EB is made out of Medium Density Fiberboard (MDF) lines with black flocking fabric to absorb most of the excess light inside of the EB while the sun simulator is on. This is so the sun sensors on the ESAT are not exposed to much light reflecting off of the surroundings. The EB has dimensions of 29” ×20”×16” with the sun simulator being 18” away from the nearest side of the ESAT. This is to keep the solar panels at their optimal operating temperature of 77°F. The sun simulator can reach temperatures of 150°F. In order to make sure the ESAT is not overheating due to a more concentrated beam of light from the sun simulator a trade study was performed to see what the optimal distance the sun simulator should be at from the ESAT. Equation 1 was used to get a baseline of how far away the sun simulator should be away from the ESAT solar panels. Equation 1 assumes that the sun simulator is a point source and is not shining into the light cone.

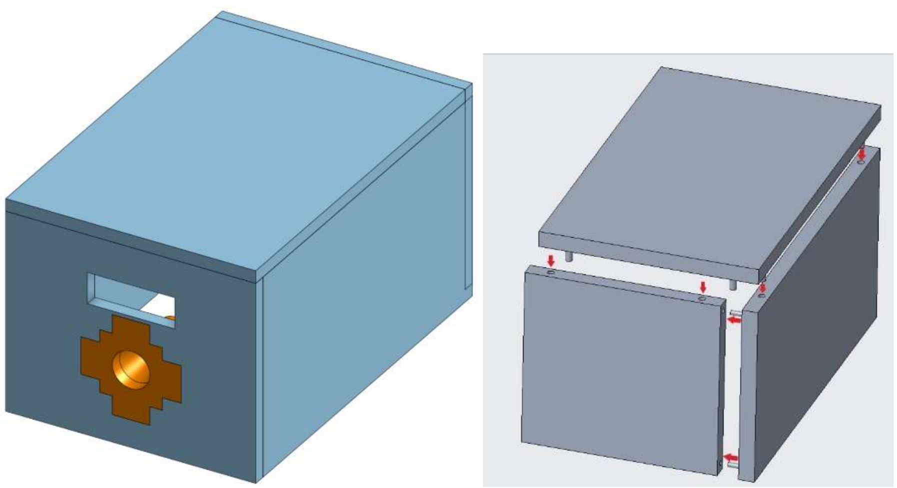
$$T^4 = \frac{P}{4\pi r^2 \sigma e} + T_L^4 \tag{1}$$

This was used as the starting distance for the trade study in Table 1. Table 1 summarizes the findings of the trade study.

Table 1. Solar Panel and ESAT Heat due to Sun Simulator

Time of test (min)	Distance from sun simulator (in)	Solar Panel Temp (F)	Temperature inside EB after 30-minutes (F)
10	18	77.3	75.2
10	17	79.3	-
10	16	81.4	-
10	15	82.1	-

The EB can be broken down into individual pieces and stored easily in a large cabinet. Each panel is labeled and connected to its neighboring panels by wooden dowels that slide into dowel anchors. This is to prevent a bit of wear and tear on the after many setup and breakdown sessions. Latches on each side of the EB secure the panels together and prevent them from coming apart easily. The EB is able to be lifted up and set down without coming apart. The light cone is attached to the front panel of the EB and faces inwards towards the ESAT to direct light from the sun simulator onto the face of the solar panels. The diameter of the sun simulator is no larger than one face of the ESAT to prevent light pollution within the EB. There is a small, tinted window above the light cone so the user can observe the behavior of the ESAT throughout the lab. The window is made of 0.25 in. thick clear acrylic that has 5% window tint overlayed with 20% window tint to reduce outside light from entering the EB. The overhead lights in the lab can stay on during the missions because the ESAT is enclosed. The error in solar angle dropped drastically from ±10° to ±1° during a test with a prototype EB. Figure 5 shows a Computer-Aided Design (CAD) model of the EB.



Front (panels together) *Back (panels apart)*

Fig. 5 CAD of the Environmental Box

ii. Light Cone

The light cone was developed in order to reduce area of light that the sun simulator shines onto the ESAT. It was made just large enough so the light from the sun simulator is roughly the same area as the area of one solar panel on the ESAT. The light cone helps to reduce excess light pollution inside of the EB. It has to withstand temperatures up to 150 °F because it will be right next to the sun simulator. PETG filament for 3D printing was used to fabricate the sun simulator. It will be held on the front panel of the EB using small latches. as used Figure 6 shows the light cone.

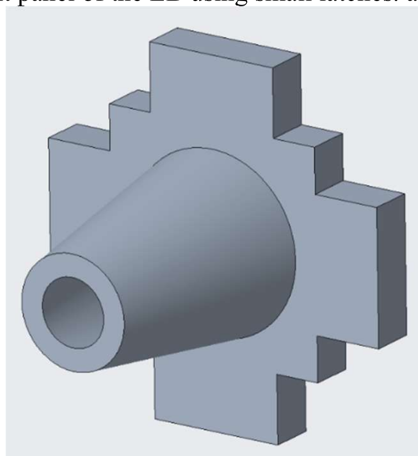


Fig. 6 Light Cone CAD Model

iii. Payload Bracket

A 3D printed bracket was manufactured for the payload. The purpose is to hold the camera and laser pointer steady atop the ESAT and to ensure it does not fall off during the missions. The bracket also angles the camera and laser pointer appropriately to point at where the targets will be on the interior of the box. This can be seen in Figure 7 atop the ESAT. There is a small lip at the bottom that sits on the top of the ESAT. The tolerances are tight so it does not wiggle much while the ESAT is moving.

iv. Raspberry Pi Camera and Laser Pointer

The current payload computer is a Raspberry Pi 4. The Raspberry Pi has both pin connections for the laser pointer to use and a dedicated camera port. Additionally, the Raspberry Pi comes equipped with a firmware package known as “Arducam”, which features a set of console commands that allows control over the attached camera from all aspects from simply viewing the camera feed to adjusting the image size and filters. One final important firmware package is the MATLAB Support Package for the Raspberry Pi Hardware, which allows for a MATLAB user to connect to the RPi and be able to see its inputs from various sources, one of which is notably a camera. This connection between MATLAB and the RPi is functionally similar to the connection between the Arduino IDE and an Arduino board. The Raspberry Pi camera connects to the MATLAB program by reading in the IP address and model into MATLAB. A laser pointer is attached to the payload as well, so the user can see more clearly where the ESAT is pointing. For the purposes of this project, the +X direction is defined as the front of the ESAT. The Raspberry Pi camera/laser pointer setup can be seen in Figure 7.

Fig. 7 Raspberry Pi Camera and Laser Pointer Setup

The Raspberry Pi is powered by its own battery to reduce the draw on the ESAT. This prolongs the ESAT's battery runtime by not having it responsible for giving out more power than it needs to. The battery is a 10000 mAh lithium battery. With an estimated power of the payload being 7.155 W over the set lab runtime of three hours for a power consumption of 21.465 Wh, the battery needed to provide more than the 21.465 Wh to meet the minimum requirement. The chosen lithium battery provides 50 Wh, which leads to an estimated run time of 4 hours and 53 minutes.

B. Sensor Subsystem

i. Live Data Transfer

The design of the computer program portion of the system is meant to be intuitive and easy to use with basic background knowledge of how satellites work. This is why COSMOS is used as the communication link between the ESAT and STK/MATLAB. The motivation for linking multiple programs together is to have more detailed analysis of the telemetry that is being received from the ESAT, to allow it to be processed through the CVT for the Target Finding mission, and to allow STK to simulate the movement of the ESAT virtually.

The first stage of the process refers to the transfer of ESAT telemetry data from the COSMOS into MATLAB. A User Datagram Protocol (UDP) connection is used to link the two. The purpose of this stage is for the transmission scripts to go through all the data packets being sent from the ESAT and parse through them to send only the solar angle byte data from COSMOS to MATLAB. The solar angle provides MATLAB with enough information to know the attitude of the ESAT. From there, the next state is to forward the attitude data that is now in MATLAB data to STK. It is worth noting that the MATLAB script will take the solar angle data, which will be in string format, and convert it to quaternions before being sent to STK.

COSMOS runs off of the Ruby coding language, so to set up the UDP connection between COSMOS and MATLAB certain Ruby scripts within the COSMOS configuration files must be modified. It turns out that COSMOS already has a ruby script dedicated to such a function, so writing a new script is not necessary.

In the period when the ESAT is connected, all different data packets are displayed on COSMOS. The purpose of the inclusions in the ruby script is to parse through all the different packets, isolate the solar angle packet, convert it into string format, and send it to a MATLAB script. Within the edited ruby script, the IP address of the laptop running MATLAB is indicated, as well as the port number [4]. When running the live data scripts, the bytes must be sorted so that only the solar angle is transmitted. It is worth noting that the port number should be the same in both the MATLAB receiver script and the COSMOS ruby script.

For the Target Finding mission, the live data transfer flow would follow Figure 8.

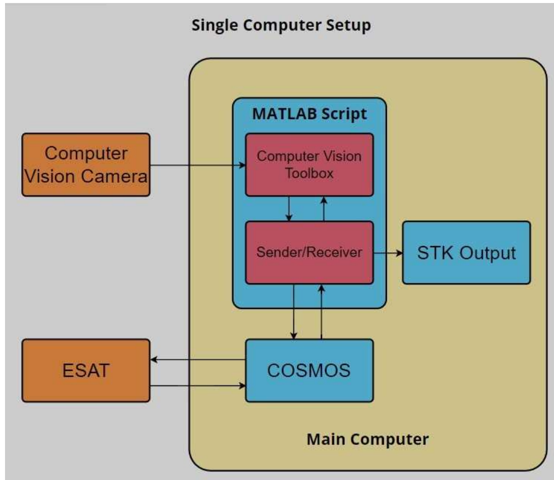


Fig. 8 Flow of Live Data for Target Finding Mission

ii. MATLAB Computer Vision Toolbox (CVT)

Since the ESAT's already existing onboard sensors are working properly, the main sensor this subsystem is adding to the ESAT is the computer vision camera. This will simulate how real spacecraft control their attitude to point at specific recognized objects and is the main focus of the Target Finding Mission. This camera system will be able to detect a circle drawn on in the inside of the box we are creating by using the MATLAB computer vision toolbox. From here, the computer vision camera system will be sending live camera data through UDP to a MATLAB script running on the main PC. How the computer vision script works is by looking for circles that have radii in a certain range and grayscale values in a certain range, where the radius of the circle is measured in pixels. This is needed so the computer vision camera can filter out any objects that are not the target (such as the circular lens of the sun simulator, for example). As for how the computer vision script tells what a circle is, it uses edge detection and Hough transform algorithms, where the edge detection algorithm just detects edges and the Hough transform algorithm converts these edge points into a possible circle in the image. This possible circle is compared to what the CVT's definition of a circle is and is displayed on the screen if it is within a specified margin of "circularness" defined by the toolbox. The computer vision system can be tested by simply waving around an image of a circle in front of a camera. It uses coordinates as seen in Figure 9 below to be able to slew the ESAT to a specific attitude.

Fig. 9 Computer Vision Toolbox Shape Recognition Test

C. Simulation Subsystem

The simulation subsystem provides the user with a visually appealing, data-focused display of the current mission. This ensures focus on the educational aspects of the attitude lab through mimicking modern day ground systems that entice students and inspire them. The simulation subsystem also seamlessly integrates with both the hardware and sensor subsystems to provide a smooth data transfer process from ESAT and its sensors to the monitor.

After researching and comparing capability through modeling examples, the choice to use ANSYS STK was made mainly because of the better graphics and ease of use when compared to MATLAB System Took Kit. In an educational environment, students should be interested in what they are learning. Visuals were weighted the most when comparing applications. STK offers a great GUI that can make the lab feel like the students are working on a real-world mission and making a difference. Cost was also weighted equivalent to visuals due to the budget constraint of $3,000. The simulation subsystem needs to be backed by great applications and capabilities without draining the entire project budget on application licenses. The ability to integrate with ESAT was of high importance. Without the capability for a stable connection to the satellite, the simulation subsystem will not be able to accurately display and analyze what the students do.

STK utilizes object model commands much like Java. In an STK scenario, every object created is the parent object. The object's properties exist on a lower level of the parent object. For each mission script, the a scenario needs to be initialized. MATLAB acts as the parent scenario because it creates the scenario and its objects, as well as all the properties and descriptions the object contains. It is important to note in each block the parent object is called back, and a variable is created to be the connecting command to the parent object. Every STK script begins at a high level and then proceeds to lower-level objects. STK also has options for each object to add to the scenario. Attaching a "setTime" command to each parent object scenario for a certain analysis period is how the timer is set for each mission. Every object has its own specific set of commands to call. When creating an object such as a satellite, orbit properties, 3D graphics properties, and even create other objects such as cameras on the satellite can be accessed. Figure 10 shows the SKT simulation that is shown on the monitor during a test run of the detumble mission.

Fig. 10 STK Visual Output for Detumble mission

VI. Mass and Power Budgets

It is important to not exceed the maximum amount of power of the ESAT so that the components do not get damaged. The Raspberry Pi on the payload has power limits on each of the peripherals. Staying equal to or underneath the power is a must to ensure that each part of the payload has enough power being supplied to them. Table 2 outlines the power budget of the payload. There are no USB accessories being used, so no power is being used there. There is a single GPIO power pin being used to supply enough power to the laser pointer. The laser pointer uses 3.3V and a current of 30 mA, which results in a power draw of 0.099. The camera will always use the maximum power. This is acceptable because the Raspberry Pi and the camera are both designed to use the maximum power given by the camera port. None of the payload parts require more power than what the Raspberry Pi can provide.

Table 2. Payload Power Budget

Payload Power Budget			
Peripheral	**Power Used [W]**	**Max Power [W]**	**Margin [W]**
USB	0	6	**6**

GPIO	0.099	0.165	**0.066**
Camera	0.990	0.990	**0**

Table 3 shows the mass budget for the air bearing. Mass is an important part of this project because of the air bearing. This piece of equipment is quite delicate because if the surface gets scratched even a little it will alter the way the air flows around the bearing causing it to wobble and spin tremendously. Mass of the ESAT and camera payload must be taken into account because they will be sitting atop the air bearing. The air bearing is to be set no higher than 25-30 psi. This is to reduce the wobbling of the ESAT while on the air bearing. This is to avoid risking damage to the air bearing, especially when testing because excessive wobbling can cause it to more than normal and scratch/score the bearing. It can be seen from Table 3 that all of the ESAT components are well below the maximum allowed mass on the air bearing.

Table 3. Mass Budget

Mass Budget						
Part	**Payload**	**ESAT**	**Turntable**	**Total**	**Max Allowed**	**Margin**
Mass [lb]	0.919	1.577	1.036	3.532	15	**11.468**

VII. Conclusion

In conclusion, the Satellite Attitude Lab is an educational product that can help enhance the learning of students when it comes to understanding the tough concepts of satellite mission operations and attitude control. Through the exercises in this lab, students can experience what the basics of a satellite mission operation involves, as well as a better understanding of how satellites move in space. There is a substantial amount of future work that can be done on this project, such as upgrading the payload capabilities or merging the three software programs into one user interface, so it is easier to access. Educating future students about satellite controls and mission operations is important because by teaching future generations of engineers, new technology may be available faster.

Acknowledgments

The authors would like to thank our faculty advisor Dr. Eric Swenson and our industry advisor Ron Fussel for their guidance and feedback on this project. In addition, the authors thank the graduate student assistants for this project George Nehma and Patrick Quinn for their support during the design process. The authors would also like to thank Ryan Sendall, another undergraduate student at Florida Tech who dedicated his time to assisting with the live data transfer portion of this project.

References

[1] Theia Space. "*Let Us Introduce You to ESAT*". (n.d.). https://www.theia.eusoc.upm.es/esat/

[2] ANSYS. *"STK Programming Help*." Product Help, help.agi.com/stkdevkit/index.htm. Accessed 5 Nov. 2023.

[3] Ball Aerospace. *"Cosmos Installation*." (n.d.). https://ballaerospace.github.io/cosmos-website/docs/v5/installation

[4] Ball Aerospace. *"Cosmos Tools."* (n.d.). https://ballaerospace.github.io/cosmos-website/docs/v4/tools

[5] Agi. "Simple real-time data integration." (n.d.). https://www.agi.com/blog/2020/07/simple-real-time-data-integration