Integrated Spacecraft Autonomous Attitude Control Testbed

Jacob Romeo[a], Dylan Ballback[a], Kyle Fox[a], and Sergey Drakunov[a]

[a]Embry-Riddle Aeronautical University, 1 Aerospace Blvd, Daytona Beach, Florida, USA, Engineering Physics Propulsion Lab

ABSTRACT

ISAAC is a 3D-printed pneumatic spacecraft for attitude control system development in a 3-axis gimbal ring. This allows for simulated free-space movement of a cold gas thruster-controlled probe in a controlled test environment. The purpose of this open-sourced control platform is to allow students, professors, and researchers to test their control algorithms on real hardware in real-time. The end goal is to have a website allowing anyone to upload their code and watch it run via live stream. The spacecraft uses a pneumatic system to mimic cold gas thrusters by using compressed air as a means of propulsion. The delivery system uses solenoids to control the thrust, stabilizing the craft. The hardware is simple and consists of custom Arduino printed circuit boards (PCB), a Raspberry Pi, an Inertial Measurement Unit (IMU) for total orientation data, and 2 LiPo batteries. The craft is entirely 3D printed, including the mounts for the components, to be accessible for future research and upgrades. The attitude controller will be integrated into the website **easycontrols.org**, which will allow anyone interested, both students and researchers alike, to upload their Python control algorithm and watch it run on hardware in real-time. The website will have built-in functions and examples, allowing the user to create their algorithm easily. A proof of concept of this system has been the application of a sliding mode controller in one axis of the gimbal rings. Future work can include the application of more modern control methods for students and facilities to display and follow.

Keywords: Spacecraft Controls, Education, Hardware-in-the-loop, Pneumatic, 3D-Printing, Testbed, Open-Sourced, IMU, Sliding-Mode Controller

1. INTRODUCTION

The need for a modular and robust controls platform for testing different software and hardware variations on a spacecraft has been shown multiple times before the Integrated Spacecraft Autonomous Attitude Control (ISAAC) Testbed. These projects had the goal of testing specific control algorithms, thruster types, or payload configurations to better understand spacecraft dynamics. Two instances that share many similarities to the ISAAC platform and contain much of the inspiration behind it are the Autonomous Re-configurable Craft (ARC) prototype and NASA's Asteroid Free Flyer. The Asteroid Free Flyer project, which was led by NASA Engineer Dr. Michael DuPuis, created a prototype spacecraft to fly in and around asteroid fields' microgravity.[1] The Free Flyer was tested extensively within the Engineering Physics Propulsion Lab (EPPL) and this project became the inspiration for ARC. ARC was created as a completely undergraduate-led project to create a small attitude control and hardware test platform for the EPPL to allow for modular hardware design.[2]

ISAAC design intent was to overcome the limitations of the Asteroid Free Flyer and ARC, by simplifying the design and manufacturing process along with making the project open-sourced. This scope created three major designs and completed versions. The first version consisted of a two-layer design similar to the Asteroid Free Flyer such that the solenoids, piping, wire routing, and internal air tanks on the bottom layer, leaving the top layer for control computers, weight balance/unbalancing, and batteries seen on the left side of Figure 1. This design was partially completed before the realization that the gimbal rings that were designed to support the spacecraft would be too large and heavy to support the larger design. After this setback, it was decided that the next version would use 3D-printed gimbal rings and the design would be an octagon-shaped plate with a center cut hole to hold one air tank seen in the middle of Figure 1. This second version was built and tested for a short time before deciding to 3D-print the craft instead. The third version greatly simplified the design of the spacecraft by removing the center tank and using modified distributed solenoid control, which allowed for ease

of use and assembly for people who only had access to a 3D printer. This also cut down the overall weight of ISAAC, allowing the gimbal rings to be smaller and not as robust, cutting manufacturing times and material.

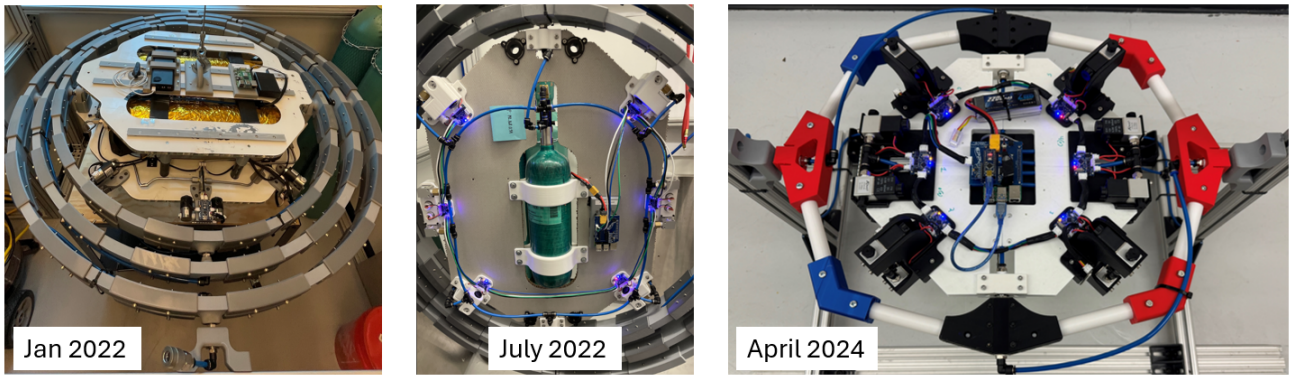

Figure 1: NASA's Freeflyer updated, ISSAC Rev 1, Current Rev 3 ISAAC.

This paper serves to describe the design and control theory behind ISAAC as well as the Easy Controls platform. It will cover the chronological order of ISAAC versions as well as the detailed hardware used to run the base control system and Easy Controls.

2. DESIGN AND METHODOLOGY

The design objective of the Integrated Spacecraft Autonomous Attitude Control (ISAAC) was to achieve cost-efficiency, simplicity, and reliability enabling the system to be open-sourced and reproducible by other research entities and academic institutions. Drawing inspiration from NASA's Freeflyer, the project aimed to create a system that could be replicated and utilized within academic environments, promoting a deeper engagement with spacecraft control systems. A significant portion of the system leverages 3D printing technology, complemented by inexpensive electronics, and simplified design ensuring that ISAAC remains within the reach of a diverse user base. Below in Figure 2, the ISAAC render displays its current design with a single gimbal ring which enables rotation in two degrees of freedom (DOF), where over 80 percent of the parts seen are 3D printed.

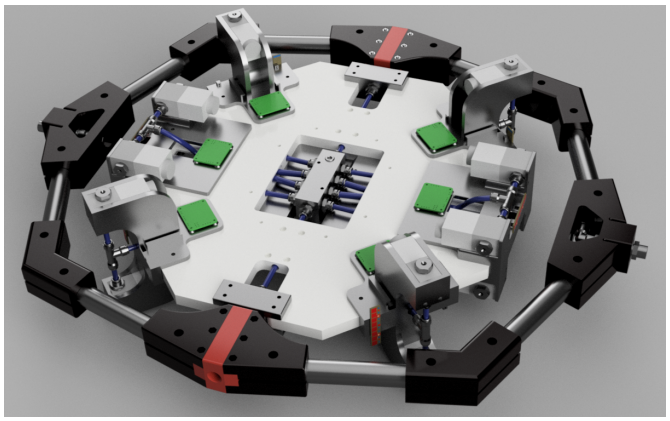

Figure 2: ISSAC Rev 3 with 1 Gimbal Ring CAD Rendering

To create a system that is compact and space-efficient, ISAAC's design integrates channels for the pneumatic tubing, connecting the central air distribution block to each gimbal ring and the six thruster modules. This can be seen below in the cross-section Figure 3, where these internal channels are only made possible by 3D printing the spacecraft's body. A major design choice was the elimination of onboard compressed gas storage, a feature

that necessitated a tank and introduced complexity. This simplification not only facilitated the open-source aspect of the spacecraft but also made it feasible for other educational and research institutions to replicate and innovate upon the design without large expenses.

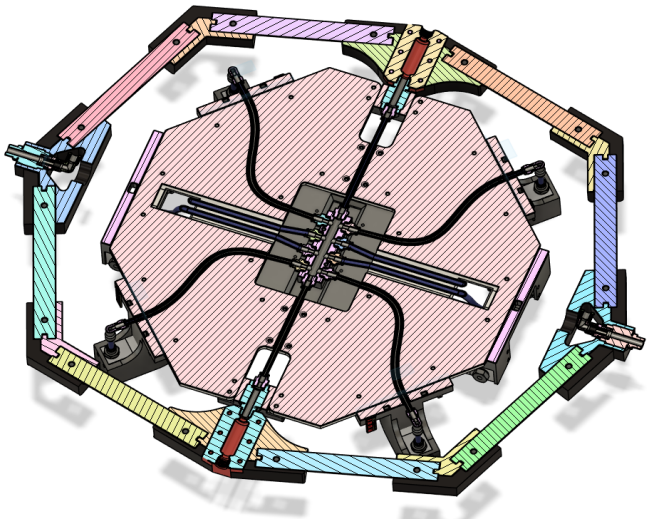

Figure 3: Cross Section of ISAAC and 1 Gimbal Ring.

The onboard tanks needed to be refilled and had a limited run time. Previously, the NASA Freeflyer had two onboard tanks pressurized to 2,200 psi[1] and presented a danger from the pressurization and discharge of the tanks. In addition, these tanks were only able to run for less than 10 minutes. By using the in-house air supply, there is an unlimited run time for ISAAC and less of a safety hazard. Thus getting rid of the tanks rids the hazard and makes operating safer as an entirety, simplifying the overall spacecraft and reducing the overall mass, while allowing for a longer run time.

One of the most challenging aspects of the project was the design and 3D printing of the gimbal ring, which are required to allow for ISAAC to freely rotate in 3 DOF. To reach the main goal of ISSAC of easy manufacturing, the rings were segmented into twelve parts to fit an average-sized printer bed. This segmentation was critical to maintaining the structural integrity and functionality of the gimbal rings while still allowing for easy assembly with common hardware. Furthermore, the connection points of the gimbal rings to the rotary unions on the spacecraft required 1/4" NPT threads, which had to be air-tight under pressures of 90-100 psi. Achieving this level of precision necessitated the use of a 0.25mm nozzle during the 3D printing process, ensuring the dimensional accuracy needed for an air-tight seal with the plastic 3D printed threads. Currently, there is only a single gimbal ring that allows for ISSAC to rotate in 2 DOF, while the third gimbal ring is currently being designed to allow ISSAC to rotate in all 3 DOF.

3. ELECTRICAL AND SOFTWARE

ISAAC employs a custom-designed printed circuit board (PCB) with an ATMega328P microcontroller (akin to an Arduino Nano), dual-channel MOSFETs for solenoid control, and I2C communication for streamlined wiring. Six thruster modules, with solenoids oriented for both vertical and horizontal thrust, facilitate maneuvering can be seen in Figure 4. Visual feedback is provided by LED strips that illuminate during solenoid activation. A Raspberry Pi manages the centralized control with a custom PCB attached on top (HAT) of the board, housing an Arduino Nano, IMU (MPU9250), and DC converter, all powered by two 3S LiPo batteries.

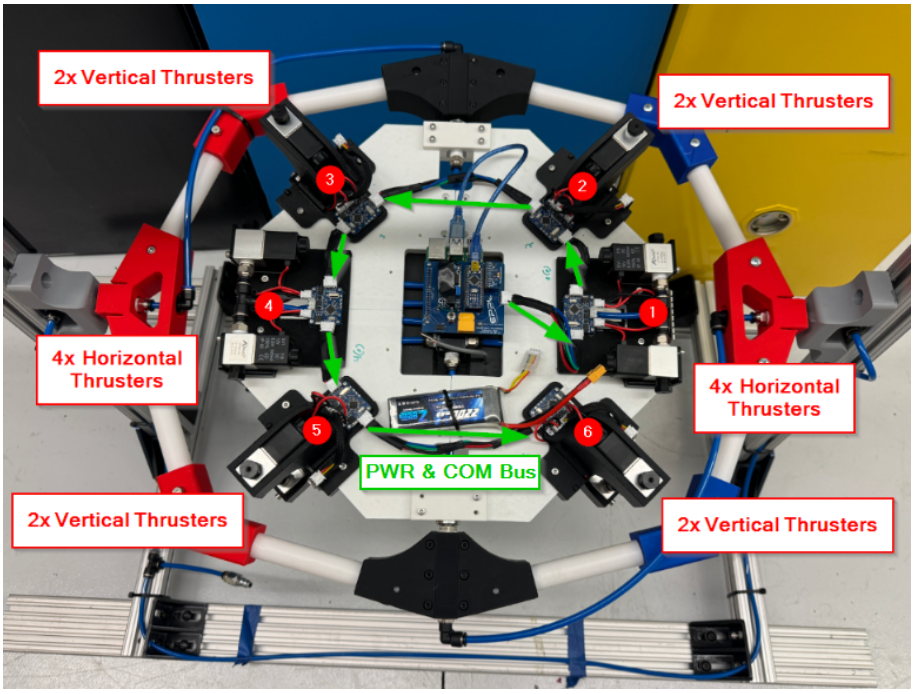

Figure 4: Annotated ISAAC Thruster Modules/ Power and Communication Bus Configuration.

To cycle the solenoids there is a 12V voltage impulse required to open and close, thus the need for the 3s LiPo batteries. However, the Raspberry Pi and the IMU use 5V whereas the Nano and IMU on the HAT use 3.3V. Because of this, a Buck Converter is being used to step down 12V to 5V, while the Pi powers the Nano and IMU from an internal source. The Nano communicates to the IMU via I2C, reading the data and sending it to the Pi via USB serial. The Pi then uses the IMU data as active feedback inside the controller and can fire any individual thruster solenoid control board by using I2C. Recently we have been experimenting with an upgraded IMU (BNO055) that directly connects to the Raspberry Pi via I2C with 5V and simplifies the overall design by eliminating the Nano on the Pi hat and simplifying the code which previously retrieved the IMU data via USB Serial. Additionally, the BNO055 IMU has onboard filtering and outputs directly to the quaternion state which will be required for the 2 DOF and 3 DOF ISSAC attitude controllers described below.

4. CONTROL THEORY

There were a few iterations to control ISAAC's attitude, the first was simply using if-else statements based on its current position measured by the IMU. Where the craft would change its direction based on hard-coded angles which would control which thrusters would fire, this would lead to a large amount of overshoot.

The second iteration incorporated a simple sliding mode controller. In this approach, as shown in Equation 1, the sliding surface is calculated based on the spacecraft's current error in position, angular velocity, and a control gain constant kd.[3] The objective of the sliding surface is to reach zero, indicating that the system has achieved its target state. The calculation of the sliding surface is as follows:

$$s = kd \cdot error + \dot{\theta} \tag{1}$$

In Equation 2, the result from the sliding surface calculation is input into the signum function *sgn*, which determines the direction of the control action:

$$u = sgn(s) \tag{2}$$

This function yields a value of 1 for a positive input and -1 for a negative input. For ISAAC, these values directly control the thruster solenoids: a result of 1 triggers the thrusters to fire in the positive direction, while -1 causes them to fire in the negative direction, effectively controlling the spacecraft's attitude to its intended orientation.

The results from using this simple sliding mode controller can be seen in Figure 5 below, where the target angle was set to 60 degrees. The bottom right plot in Figure 5 shows the error angle over time, which can be seen to converge to 0, although it is clear that there is chatter where the spacecraft is +/- 10 degrees of error, due to the use of the *sgn* function. Around the 35-second mark, an external disturbance was introduced to the spacecraft as can be seen in Figure 5, and shortly after the controller then re-stabilizes the spacecraft at its target angle. Additional trials were run with a target angle of 45 degrees (Figure 7) and 90 degrees (Figure 8), which can be seen below in Appendix A.

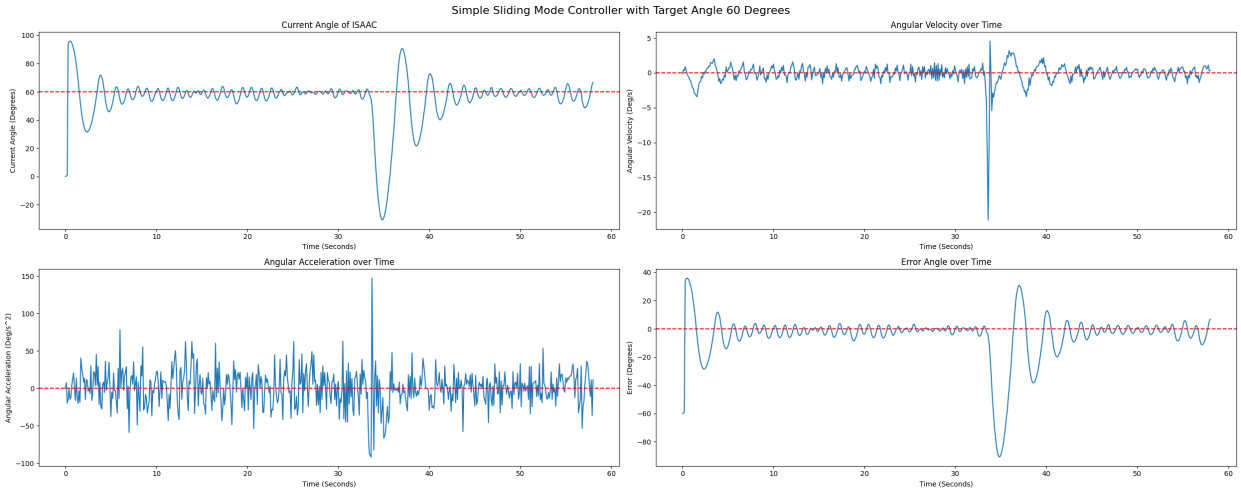

Figure 5: Simple Sliding Mode Controller - Target Angle 60 Degrees Data.

Current work is on implementing a more advanced sliding mode controller with a PID observer to control ISAAC attitude in 2/3 DOF. The goal is to have the PID observer implemented inside the sliding mode control to reduce the oscillations that can be seen above in Figure 5 error angle over time. This method would be switching from utilizing Euler angles and moving forward to using quaternions (q) by collecting the current spacecraft state from the IMU and converting it to quaternions, which the IMU does. This data comes in matrix form, representing a vector that is used to derive the angular velocity (ω). Equations 3-5 are the first steps to solve for the angular velocity using the quaternions and the arbitrary variables $l_{x,y,z}$.

$$l_{x,y,z} = \frac{q_{1,2,3}}{\sqrt{q_1^2 + q_2^2 + q_3^2}} \tag{3}$$

$$\theta/2 = cos^{-1}(q_0) \tag{4}$$

$$log(q) = l_{x,y,z}(\theta/2) \tag{5}$$

Using these equations to solve for the angular velocity in Equation 6 where λ is a control parameter based on the system.

$$\vec{\omega} = \omega_{x,y,z} = -\lambda log(q) = \vec{\sigma} = \sigma_{x,y,z} \tag{6}$$

Where σ is the error in each degree of freedom (x, y, z). This is then plugged into Equation 7 to solve for the torque (τ) the attitude controller creates.

$$\vec{\tau} = \tau_{x,y,z} = -\tau_{max} \cdot sgn(\sigma_{x,y,z}) \tag{7}$$

Where *sgn* is the sign function which is explained above. Equations 8 and 9 determine the positional correction for ISAAC, and are derived using the angular velocity and torque which were solved for, and the quaternion state given by the IMU.

$$\dot{q} = \frac{1}{2} q \cdot \vec{\omega} \tag{8}$$

$$J\dot{\vec{\omega}} = -\vec{\omega} X J\vec{\omega} + \vec{\tau} \tag{9}$$

5. WEBSITE

The Easy Controls website is in development to enable remote access for writing and executing control algorithms on the Integrated Spacecraft Autonomous Attitude Control (ISAAC) hardware in real time. Presently, it features an on-site coding interface, allowing users to directly input and execute their codes on ISAAC hardware, with the outcomes displayed on the website. This functionality is illustrated in Figure 6, highlighting the embedded Python Integrated Development Environment (IDE) for streamlined code adjustments directly on the website, bypassing the need for external IDEs for code modifications and uploads. The site, powered by Flask, is hosted on ISAAC's Raspberry Pi, which also stores trial data in a local database, offering users the option to download the raw data for further analysis.

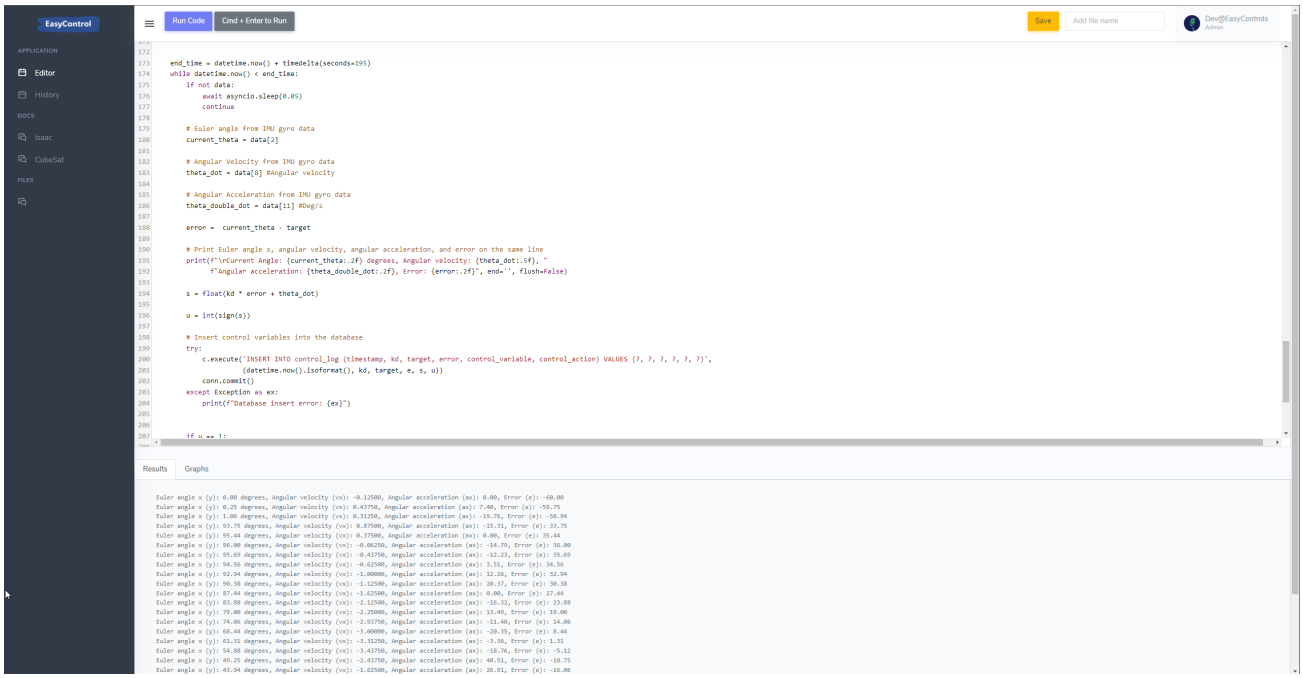

Figure 6: Easy Controls current development example of writing code and running it on ISAAC through the website.

Future developments for the Easy Controls website are targeted at enriching user experience and advancing controls education. The website will introduce live streaming capabilities to enable real-time monitoring of spacecraft trials, coupled with a comprehensive video archive of past sessions. This will allow users to select, view, and analyze specific trials, with controller data displayed graphically and synchronized with the corresponding videos. To promote learning and engagement, the site aims to provide an interactive learning environment where users, even those without access to physical hardware, can run example code and observe its performance on actual spacecraft hardware, bridging the gap between simulation and real-world application. This feature will include sample code, detailed documentation, and video tutorials, all designed to help users effectively utilize the Easy Controls platform for developing and testing spacecraft attitude control algorithms.

6. CONCLUSION

ISAAC is a 3D-printed and lightweight attitude controller that will allow anybody with an internet connection to upload their control algorithm onto the Easy Controls website and watch the spacecraft run in real time. This allows students and researchers alike to learn applications of control theory. Not just the theoretical side of controls, which ends up being complicated and often confusing to students. ISAAC embodies a marriage of affordability and high technical functionality, drawing on the legacy of NASA's Freeflyer while paving the way for widespread academic and research engagement in spacecraft control systems.

Soon, the Easy Controls website will include tutorials and tips to get started on your algorithm. This will attract students and people who yearn to learn about controls and will help them progress faster because they will have access to real-time hardware. This will enhance their learning experience by enabling the connection between the abstract math required for spacecraft controls and how the physical spacecraft system behaves. This is an anticipated feature of Easy Controls at Embry-Riddle and will be tested on campus before being produced publicly. Currently, Easy Controls and ISSAC are planned to be utilized while teaching spacecraft attitude dynamics, enabling the professor to teach their lesson and the theory, then visit the Easy Controls website and show the students the real-life application of what they are learning.

APPENDIX A. ADDITIONAL CONTROLLER DATA

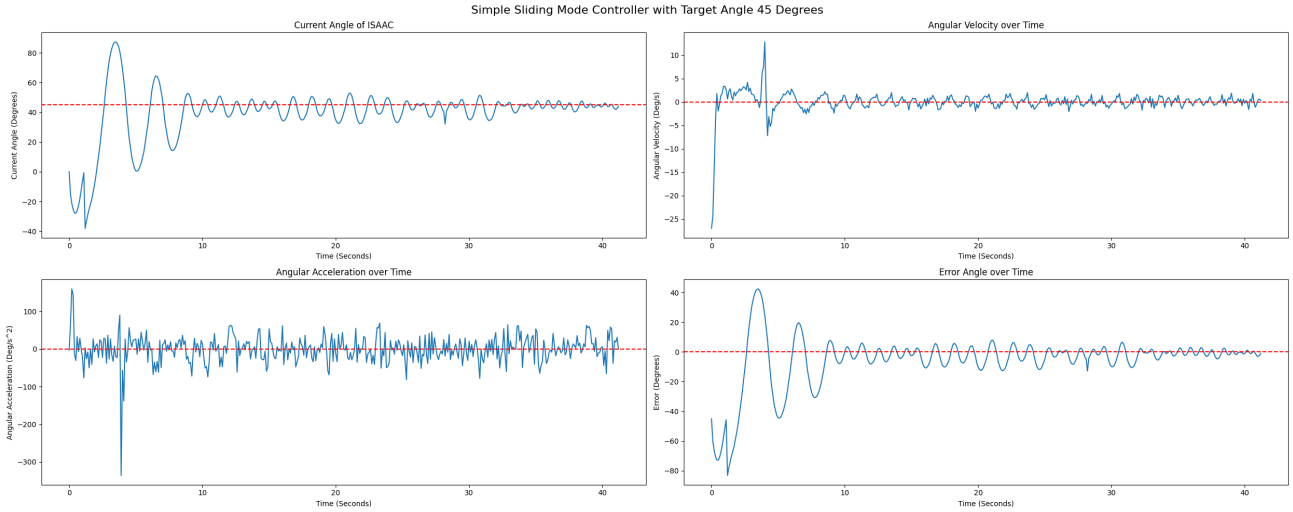

Figure 7: Simple Sliding Mode Controller with Target Angle 45 Degrees

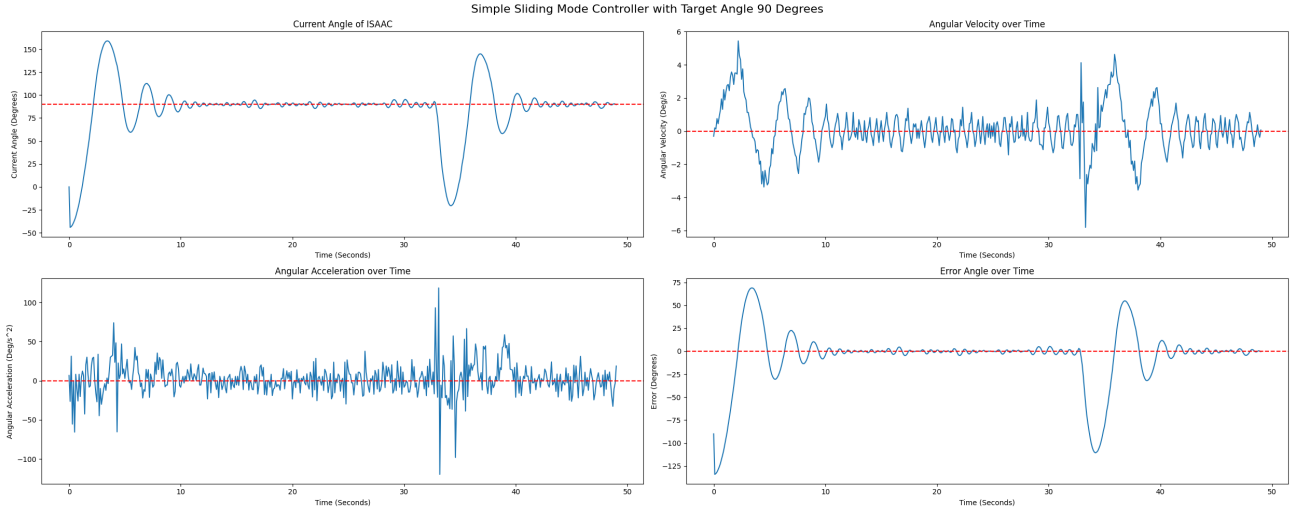

Figure 8: Simple Sliding Mode Controller with Target Angle 90 Degrees

ACKNOWLEDGMENTS

We would like to give a generous thank you to Susan Adams, Donna Fremont, the ERAU Physics Department, and the Office of Undergraduate Research for their assistance in our grant management and guidance. We would also like to thank John Rose for his wit and positivity he always provided in the early morning hours of our research. Lastly, we would like to thank Daniel Wilzack for all of his mentorship in the development of ISAAC and the Easy Controls website.

REFERENCES

[1] Kitchen-McKinley, S. J., "Variable structure feedback control with application to spacecraft with small thrust propulsion systems," *ERAU DOCTORAL DISSERTATIONS AND MASTER'S THESES* (2017).

[2] Nadeau, J., Rukhaiyar, A., Pastrana, F., Serafin, P., and Topolski, C., "Erau arc autonomous reconfigurable craft spacecraft prototype," (2018).

[3] Decarlo, R. A. and Żak, S. H., "A quick introduction to sliding mode control and its applications 1," (2008).