CubeSat Control Platform + ACTIV
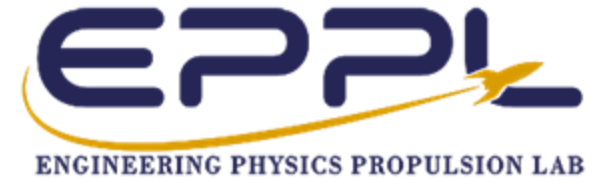## 11/7 Meeting

General Updates/Reminders

Student Research Symposium is on **November 15th** in the union

   -We'll be presenting during **Session 1 (11:00 to 12:15)**

   -The goal is to show up with working demos of the inverted pendulum and ACTIV (1-DoF) as well as the

   assembled expanded 1U CubeSat

   -Any other presentation ideas?

Easy Controls was admitted to TREP Expo!

   -Embry-Riddle's showcase of venture concepts

[Justin]: Database integration, Fusion management, and more

Progress completed this past week

- Added capability to store arm angles in SQ Lite database (100%)
- Redrafted parts of inverted pendulum for clarity (100%)
- Organized Fusion Team (more in following slides) (>50%)
- Expanded IMU Faraday Cage (100%)
- Adjusted pendulum arm
- Assisted in TREP Expo application (>50%)
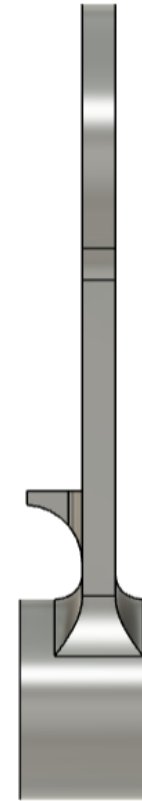- Make PDF document regarding Fusion/Python conventions

Goals for next week

- Design of ACTIV rotary union w/ motor (coming on the 9th) and 1DoF config
- Tune inverted pendulum PID controller
- Begin printing ACTIV rotary unions

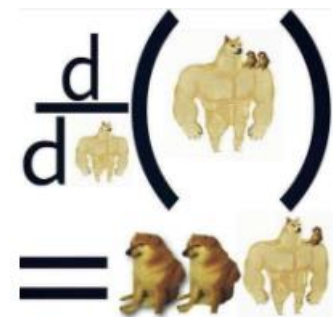**Anticipated Challenges**

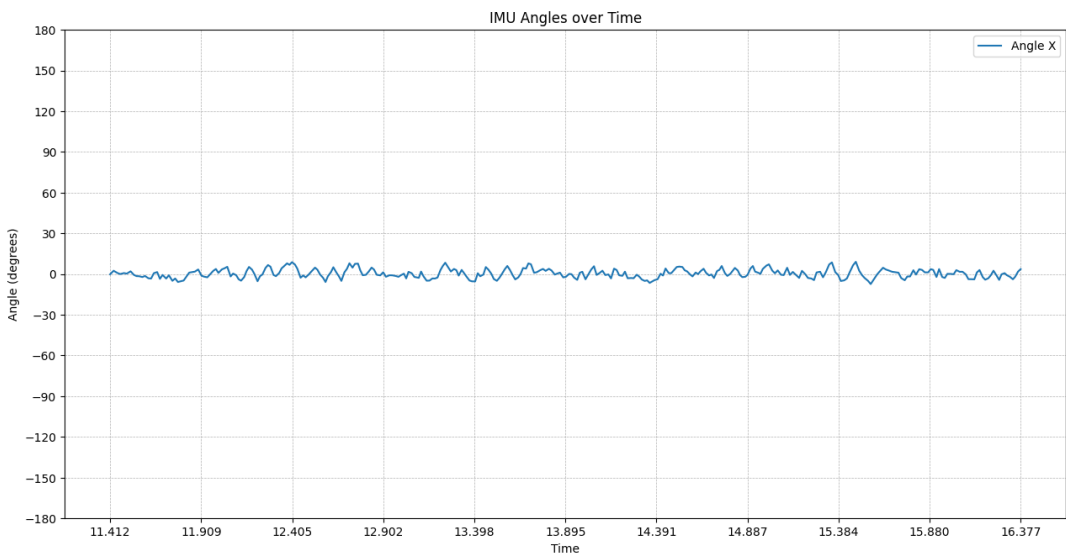- Time crunch with SRS in one week

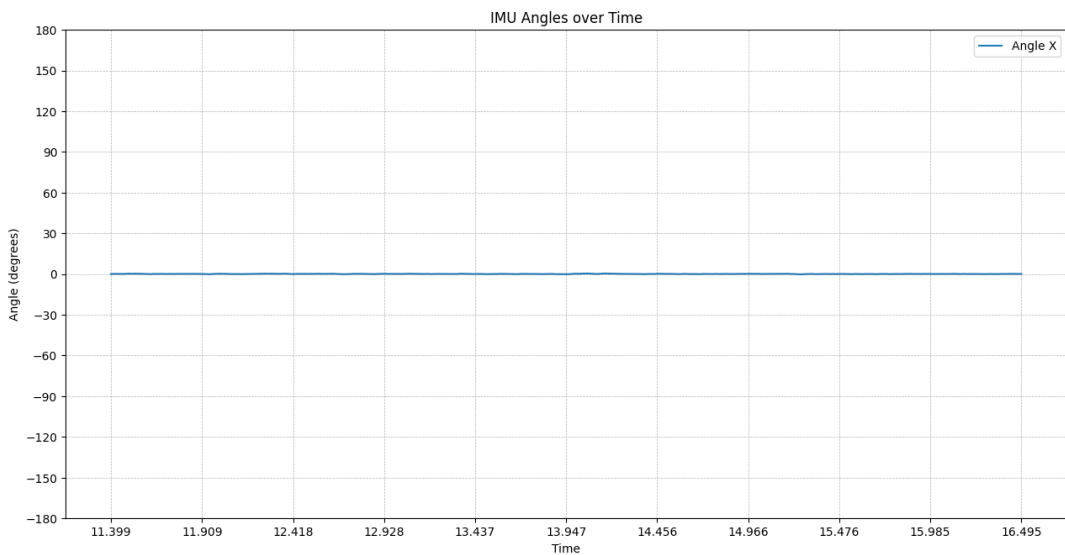**Previous Design**   **Current Design**

[Justin]: Faraday Cage Effectiveness

Without Faraday Cage

With Faraday Cage

[Justin]: Updated Fusion Structure

1) **Separated each platform into its own *Project***
   - Can be found from *Home* page of CubeSat Control Platform

2) **Further broke up each project into clear subsections**
   - Before drafting new parts, please examine existing folders, and subfolders, to find logical location
   - If anyone sees obvious subfolder missing, please add it and post your change to the discord
   - Adding *Archive* sections for outdated/unused parts (add a "- Archive" to the title of any part placed here

3) **Naming conventions**
   - Use clear and descriptive names for parts/assemblies (do not use your name or relative terms such as new, bad, etc.)

4) **A note about assemblies**
   - When inserting parts into an assembly file, insert the parts from their original folder, DO NOT make a copy in the folder of the assembly
   - This measure is to decrease redundant part copies within the repository
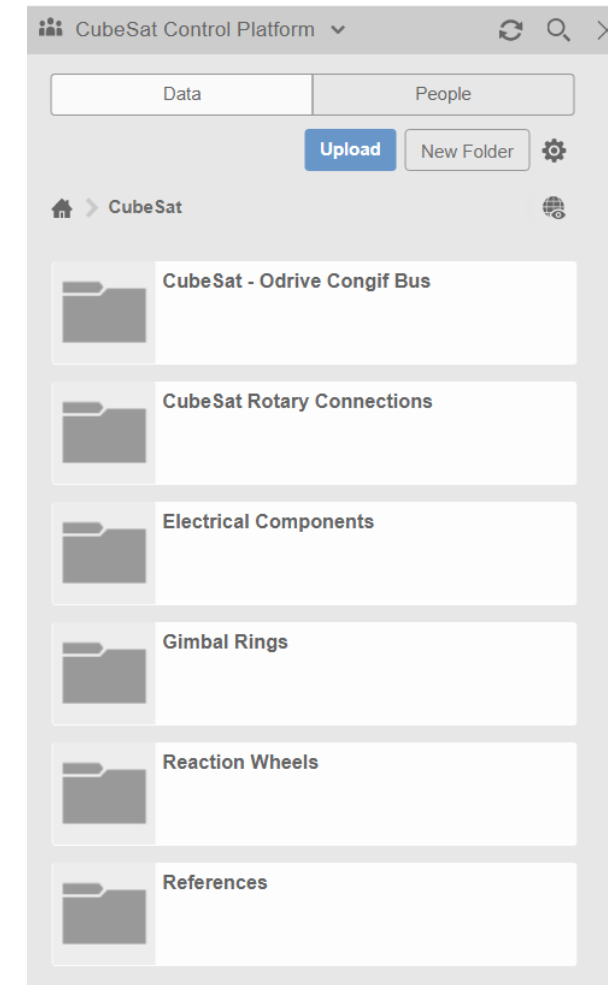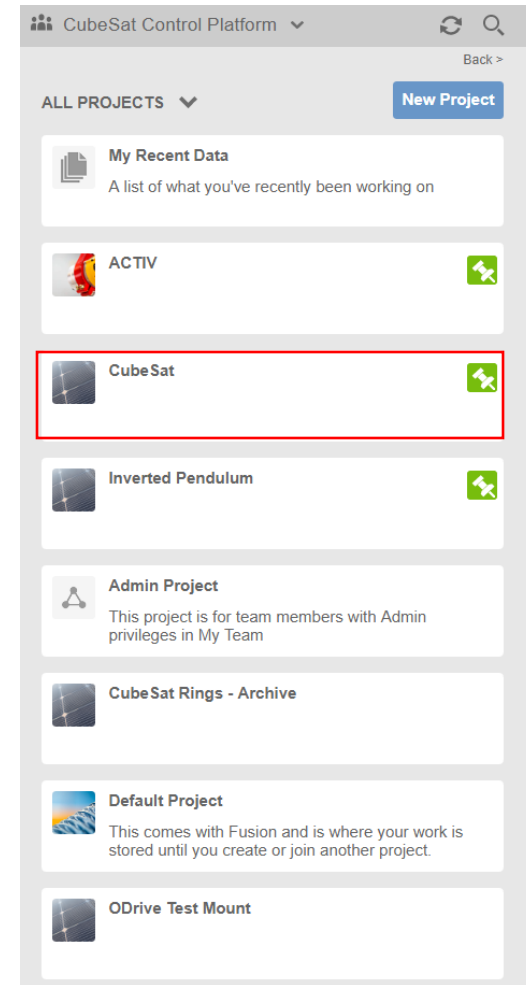
[Isaac] +[2DOF/3DOF Cube Sats]

Progress completed this past week

- Changed the reaction wheel CAD
- added the correct configuration to 1.5U
- made new shorter and new location mount holes
- holes for the claw mount

Goals for next week

- adding the electronics area to the open cubesat side

Anticipated challenges

- correctly locating the electronics locatic

EPPL

ENGINEERING PHYSICS PROPULSION LAB
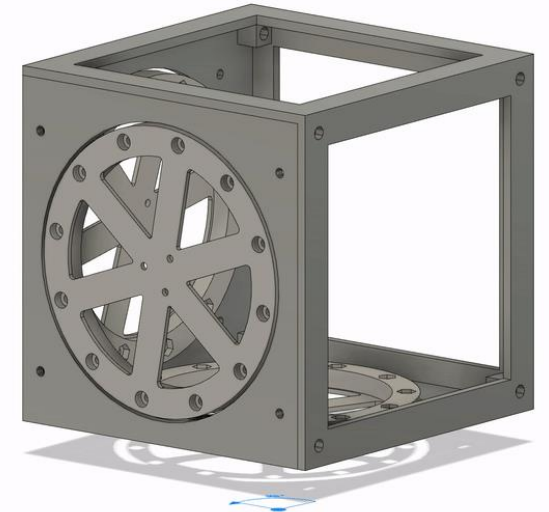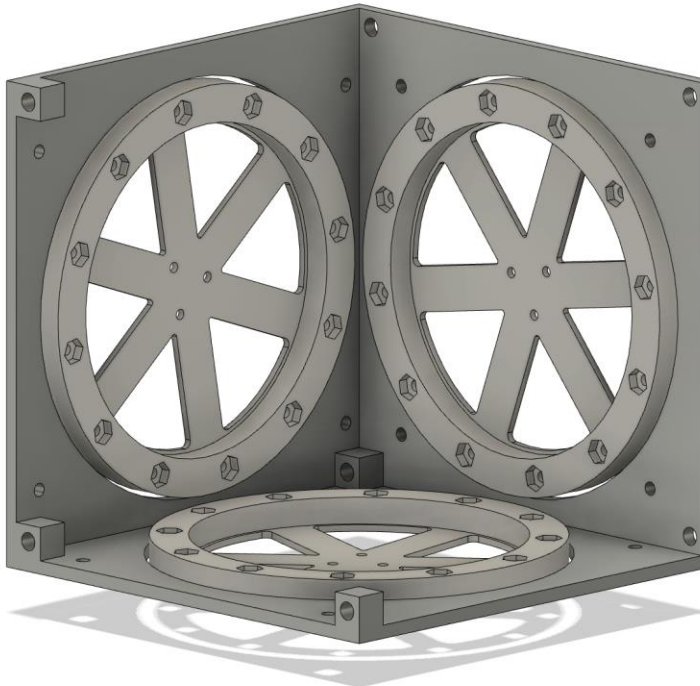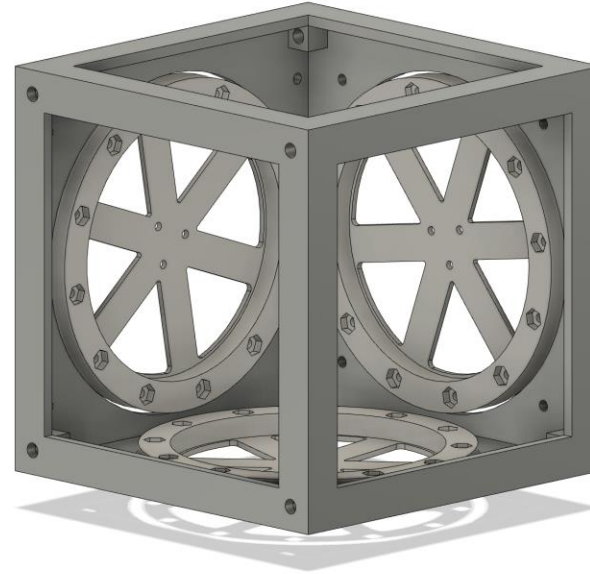
[Ella] + [Inverted pendulum]

Progress completed this past week

-Created lock for inverted pendulum to start straight up

-Dylans idea

-Made the new washer to go in front of the bearings to lock them in

Goals for next week

Anticipated challenges

"I want it to be aesthetically pleasing."

- Justin

[Ryan] + [Software]

[Spoooooky]
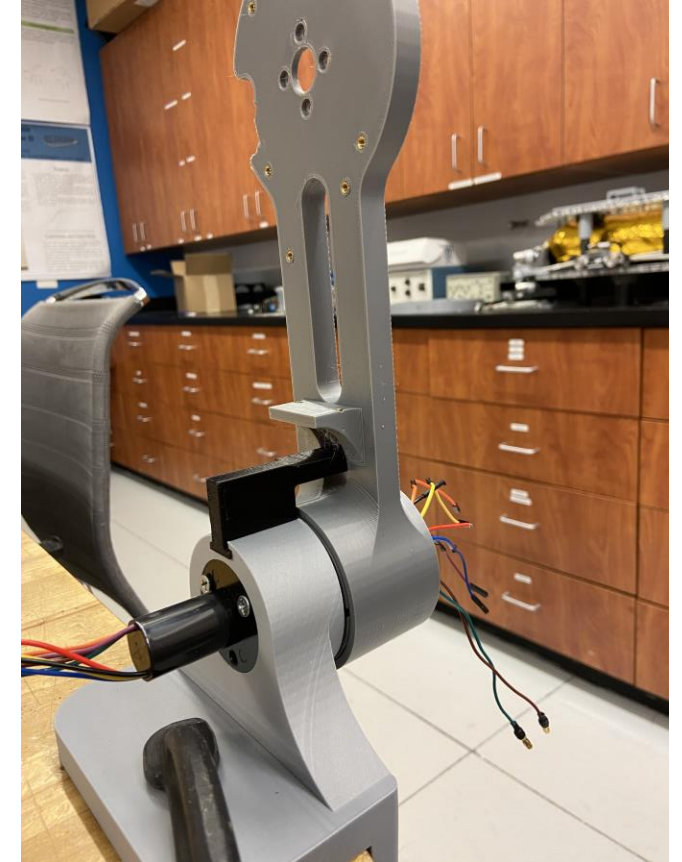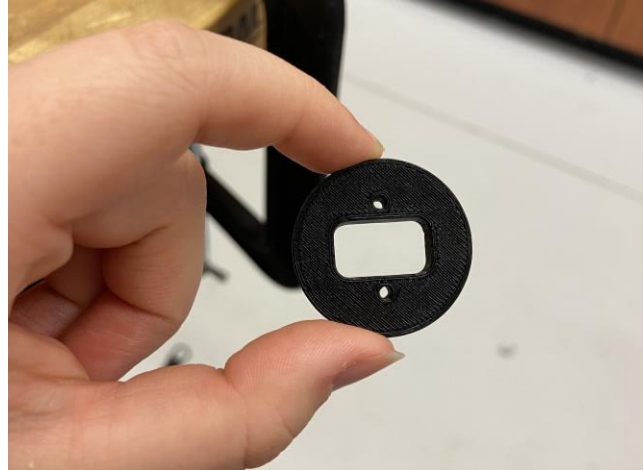
Progress completed this past week

-Finished Sensor parent class and LSM9DS1 subclass basic functionality

-Finished estimator parent class and added tilt, quest subclasses with basic functionality

-Started data storage class.

Goals for next week

-Add a test.py program to demo functionality prior to pushing to github (tonight after I take a tiny creeper trick or treating)

-Incorporate SQL into data storage class (needs to be done before filtering/filtered estimators)

- Add TODO comments

Anticipated challenges

-Ya boi managed to get a technical interview and has to study

-I don't have a way to test data collection with current program. May need to add serial functionality.

COPY O.G.

EPPL
ENGINEERING PHYSICS PROPULSION LAB
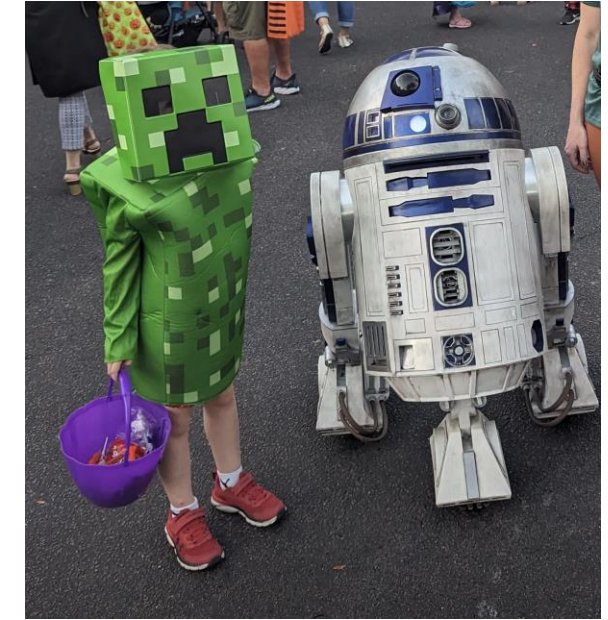
[Dylan] + [More CAN O-Drive Control]

Progress completed this past week

- **Get CAN Control**
  - **Torque Control**
  - **Controlling multiple motors at once**

- **Work on learning ROS 1...**
  - Maybe we can use this to make our lives easier?
- **TREP expo poster**

Goals for next week

- Implement simple PID on Inverted Pendulum w/ Justin
  - Pending new Inverted Pendulum Arm
- Get preliminary design for 3DoF CubeSat Electronics
- 3D print and assembly ISSAC 3DoF CubeSat
- Try and get 3D print going of the new gimbal rings
- Want to work more on livestreaming data with WebSocket

EasyControls.org

EMBRY-RIDDLE Aeronautical University DAYTONA BEACH, FLORIDA

EPPL — ENGINEERING PHYSICS PROPULSION LAB

TREP EXPO — ENTREPRENEURSHIP EXPO

Dylan Ballback, Justin Hartland, Ella Cheatham, Adam Duke, Jacob Romeo, Jacob Salazar, Isaac Stitt, Ryan Taylor, Vishal Ramisetty, Vishwam Rathod, Jacob Raynor, Kyle Fox
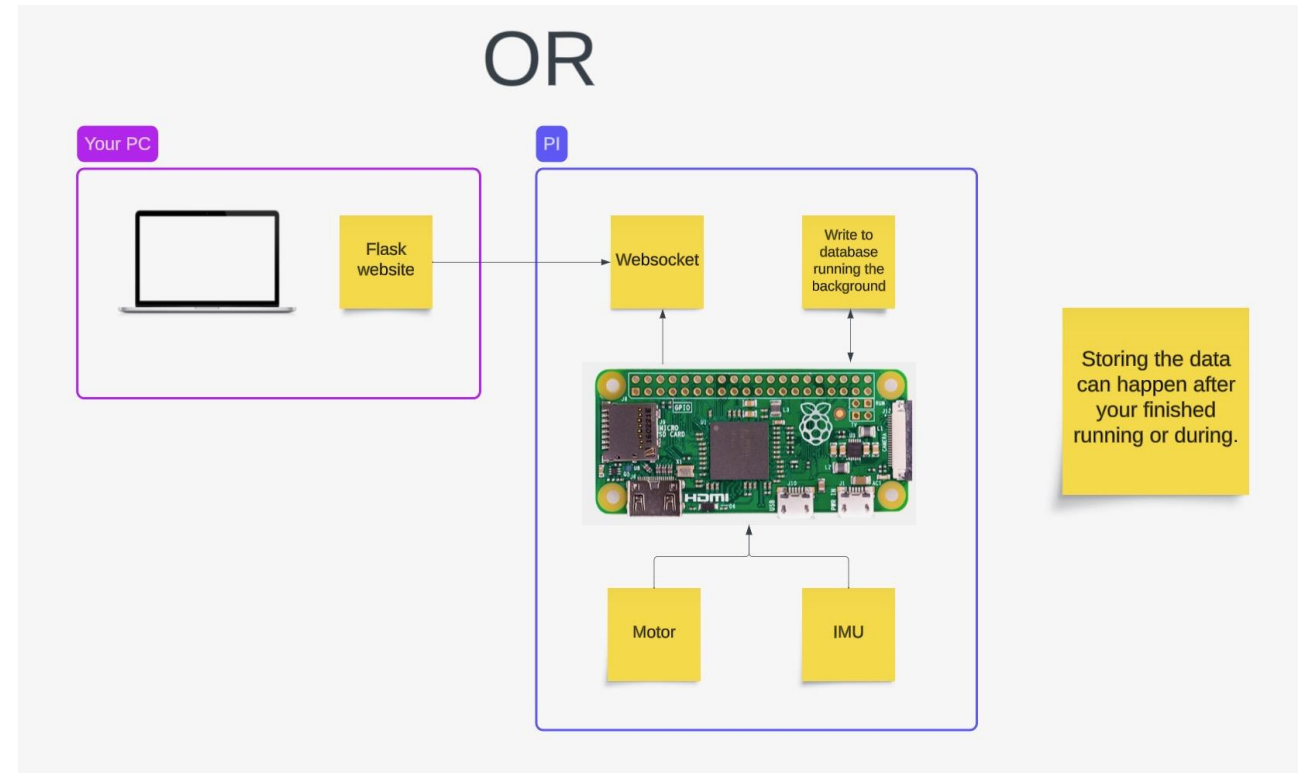
COMPANY PURPOSE

- EasyControls.com allows for students and researchers to accelerate and expand their knowledge of control systems and algorithms. This educational space gives people the ability to complete hardware-in-the-loop testing in real-time on physical hardware through the internet.

PRODUCT

- Monetizing the use of EasyControls.org
  - Able to charge $/min of runtime
  - Able to sell educational controls courses with a focus on using EasyControl.org to learn.
- Educational Kits (Raw Material Cost)
  - Inverted Pendulum ($400)
  - CubeSat ($900)
  - Spacecraft ($1200)

PROBLEM

- Learning Controls can be overwhelming with complex mathematics and equations, and it is very hard to visualize and connect the dots between controlling a system in real life and the abstract mathematical equations/formulas.
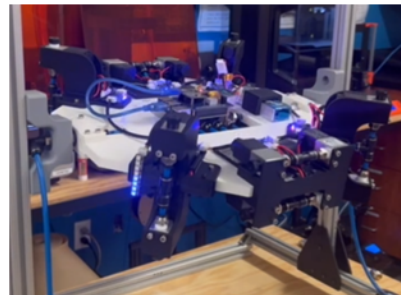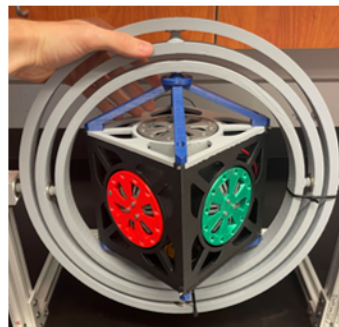
Momentum Wheel

Pivot Angle
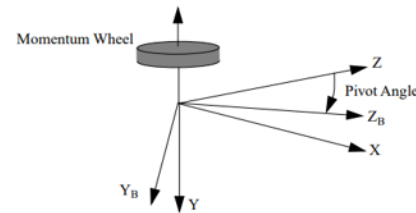
$$\omega_1 = \Omega_1$$
$$\omega_2 = \Omega_2 + b_2^T \omega_1$$
$$\omega_3 = \Omega_3 + b_3^T \omega_2$$
$$h_3 = I_3 \omega_3$$
$$h_2 = I_2 \omega_2 + b_3 h_3$$
$$h_1 = I_1 \omega_1 + b_2 h_2$$

SOLUTION

- EasyControls.org will allow for anyone online can upload their own controls algorithm and watch a live stream of how their algorithm performs on hardware in real-time.
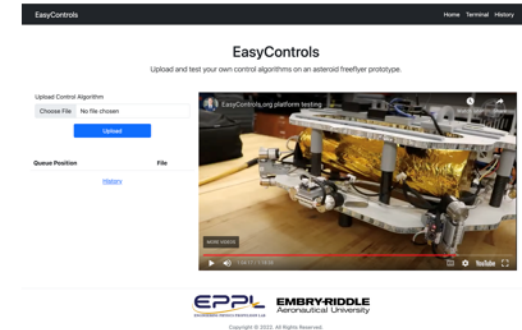
MARKET POTENTIAL/COMPETITION

- Controls professors being able to use this as a teaching aid while in class or for homework/project assignments.

- Existing educational controls platforms on the market.
  - Balancing Robot Kit ($85)
  - HiWonder Robotic Arm and Car ($600)
  - Quanser Inverted Pendulum ($5000)
  - EyasSat Classroom Satellite ($12350)

1DoF Spacecraft Prototype

1U 3DoF CubeSat Prototype

Inverted Pendulum Prototype

O-Drive CAN Commands

CMD ID	Name	Sender	Start Byte	Name	Type	Description
0x000	Get_Version	ODrive_Axis0	0	Protocol_Version	uint8	Always reported as 2
			1	Hw_Version_Major	uint8	hw_version_major
			2	Hw_Version_Minor	uint8	hw_version_minor
			3	Hw_Version_Variant	uint8	hw_version_variant
			4	Fw_Version_Major	uint8	fw_version_major
			5	Fw_Version_Minor	uint8	fw_version_minor
			6	Fw_Version_Revision	uint8	fw_version_revision
			7	Fw_Version_Unreleased	uint8	fw_version_unreleased
0x001	Heartbeat	ODrive_Axis0	0	Axis_Error	uint32	<axis>.active_errors <axis>.disarm_reason
			4	Axis_State	uint8	<axis>.current_state
			5	Procedure_Result	uint8	<axis>.procedure_result
			6	Trajectory_Done_Flag	uint8	<axis>.controller.trajectory_done (0: Flase, 1: True)
0x002	Estop	Master	Empty Payload	-	-	ESTOP_REQUESTED
0x003	Get_Error	ODrive_Axis0	0	Active_Errors	uint32	active_errors
			4	Disarm_Reason	uint32	disarm_reason
0x004	RxSdo	None	0	Opcode	uint8	0: Read, 1: Write
			1	Endpoint_ID	uint16	Endpoint ID as found in flat_endpoints.json
			3	Reserved	uint8	-
			4	Value	uint32	Data type and length depend on endpoint ID
0x005	TxSdo	None	0	Reserved0	uint8	-
			1	Endpoint_ID	uint16	Endpoint ID as found in flat_endpoints.json
			3	Reserved1	uint8	-
			4	Value	uint32	Data type and length depend on endpoint ID
0x006	Set_Axis_Node_ID	Master	0	Axis_Node_ID	uint32	node_id
0x007	Set_Axis_State	Master	0	0	uint32	requested_state

0x009	Get_Encoder_Estimates	ODrive_Axis0	0	Pos_Estimate	float32	<axis>.pos_vel_mapper.pos_rel <axis>.pos_vel_mapper.pos_abs Depends on: ODrive.Controller.Config.absolute_setpoints
			4	Vel_Estimate	float32	<axis>.pos_vel_mapper.vel
0x00b	Set_Controller_Mode	Master	0	Control_Mode	uint32	control_mode
			4	Input_Mode	uint32	input_mode
0x00c	Set_Input_Pos	Master	0	Input_Pos	float32	input_pos
			4	Vel_FF	uint16	input_vel
			6	Torque_FF	uint16	input_torque
0x00d	Set_Input_Vel	Master	0	Input_Vel	float32	input_vel
			4	Input_Torque_FF	float32	input_torque
0x00e	Set_Input_Torque	Master	0	Input_Torque	float32	input_torque
0x00f	Set_Limits	Master	0	Velocity_Limit	float32	vel_limit
			4	Current_Limit	float32	current_soft_max
0x011	Set_Traj_Vel_Limit	Master	0	Traj_Vel_Limit	float32	vel_limit
0x012	Set_Traj_Accel_Limits	Master	0	Traj_Accel_Limit	float32	accel_limit
			4	Traj_Decel_Limit	float32	decel_limit
0x013	Set_Traj_Inertia	Master	0	Traj_Inertia	float32	inertia
0x014	Get_Iq	ODrive_Axis0	0	Iq_Setpoint	float32	Iq_setpoint
			4	Iq_Measured	float32	Iq_measured
0x015	Get_Temperature	ODrive_Axis0	0	FET_Temperature	float32	<axis>.motor.fet_thermistor.temperature
			4	Motor_Temperature	float32	<axis>.motor.motor_thermistor.temperature
0x016	Reboot	Master	Empty Payload	-	-	reboot()
0x017	Get_Bus_Voltage_Current	ODrive_Axis0	0	Bus_Voltage	float32	vbus_voltage
			4	Bus_Current	float32	ibus
0x018	Clear_Errors	Master	Empty Payload	-	-	clear_errors()
0x019	Set_Absolute_Position	Master	0	Position	float32	set_abs_pos()
0x01a	Set_Pos_Gain	Master	0	Pos_Gain	float32	pos_gain
0x01b	Set_Vel_Gains	Master	0	Vel_Gain	float32	vel_gain
			4	Vel_Integrator_Gain	float32	vel_integrator_gain
0x01c	Get_Torques	ODrive_Axis0	0	Torque_Target	float32	effective_torque_setpoint
			4	Torque_Estimate	float32	torque_estimate

O-Drive CAN Python Class

Started on the code layout:

- Maybe need some help with this code!

```python
import board
import can


class ODriveCAN:
    """
    A class for setting up O-Drive motor controllers using CAN comunication

    Attributes:
        Specifically for setting up CAN comunication between Raspberry Pi and CAN Communication Type:
            canBusID (String): Can Bus ID should be default "can0" but if you have muilitiple can buses
            on your device you can modify here

            canBusType (String): python-can package CAN communication type we by default us "socketcan"

        O-Drive Controller Specific Attributes:
        nodeID (integer): The node ID can be set by the
    """
    def __init__(self, canBusID, canBusType, nodeID):
        self.canBusID = canBusID
        self.canBusType = canBusType
        self.nodeID = nodeID


    def setAxisNodeID(self):
        """
        Sets Axis NodeID for an O-Drive Controller through CAN BUS

        Set_Axis_NodeID: 0x06
        """

        pass


    def setAxisState(self):
        """
        Set Axis State for an O-Drive Controller through CAN BUS

        CAN Set_Axis_State: 0x07
            Axis_Requested_State:
                Undefined:                          0x0
                Idle:                               0x1
                Startup_Sequence:                   0x2
                Full_Calibration_Sequence:          0x3
                Motor_Calibration:                  0x4
                Encoder_Index_Search:               0x5
                Encoder_Offset_Calibration:         0x6
                Closed_Loop_Control:                0x7
                Lockin_Spin:                        0x8
                Encoder_DIR_Find:                   0x9
                Homing:                             0xA
                Encoder_Hall_Polarity_Calibration:  0xB
                Encoder_Hall_Phase_Calibration:     0xD
        """

        pass
```

```python
    def setControllerMode(self):
        """
        Set the O-Drive Controller Mode type

        Attribute:
            CAN Set_Controller_Mode: 0x0B
                    Control_Mode:
                        Voltage_Control:    0x0
                        Torque_Control:     0x1
                        Velocity_Control:   0x2
                        Position_Control:   0x3

                    Input_Mode:
                        Inactive:       0x0
                        Passthrough:    0x1
                        VEL_Ramp:       0x2
                        Pos_Filter:     0x3
                        Mix_Channels:   0x4
                        Trap_Traj:      0x5
                        Torque_Ramp:    0x6
                        Mirror:         0x7
                        Tunning:        0x8
        """
        pass



    def getAxisEncoderEstimates(self):
        """
        Get Encoder Estimates for specific O-Drive Controller Axis through CAN BUS

        CAN Get_Encoder_Estimates: 0x09
                        - Pos_Estimate
                        - Vel_Estimate

        Attributes:
            Axis_ID


        Returns:
            Pos_Estimate
            Vel_Estimate
        """
        pass
```

Visual System Updates

- Jacob and Justin scheduling a talk with Dr. Henderson over zoom tomorrow.

- Star tracking is supposedly a pain to replicate in a lab setting

- Example for lab-based star tracker testing: https://www.hindawi.com/journals/ijae/2013/505720/

[Assignee] + [Task Title]

Progress completed this past week

-[Discuss progress]

-Highlight based on: <mark style="background:lime">Complete</mark>, <mark style="background:yellow">>50%</mark>, <mark style="background:red"><50%</mark>

Goals for next week

-[Discuss goals]

Anticipated challenges

-[discuss challenges, request assistance if needed]

[Relevant photos if needed]