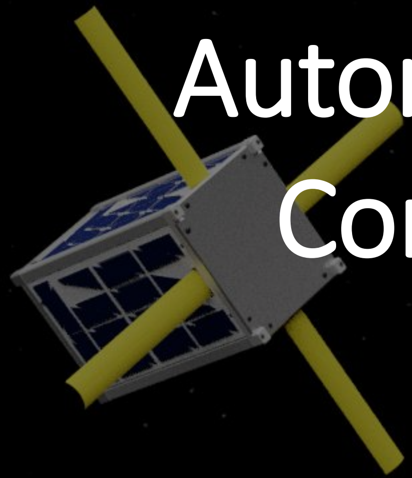
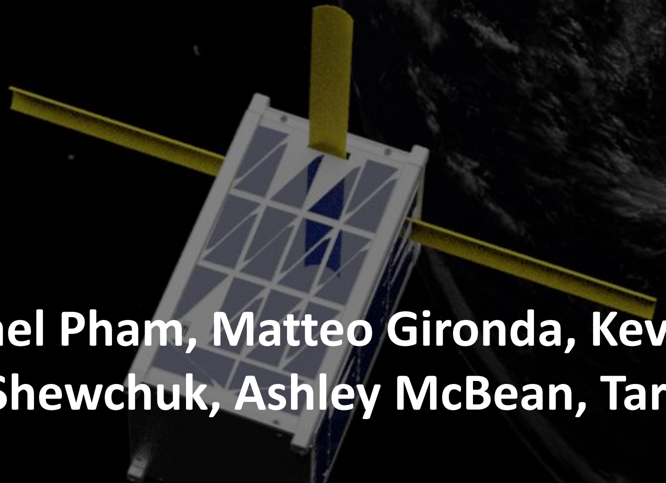
A Novel Approach to an Autonomous and Dynamic Satellite Control System Using On-Orbit Machine Learning



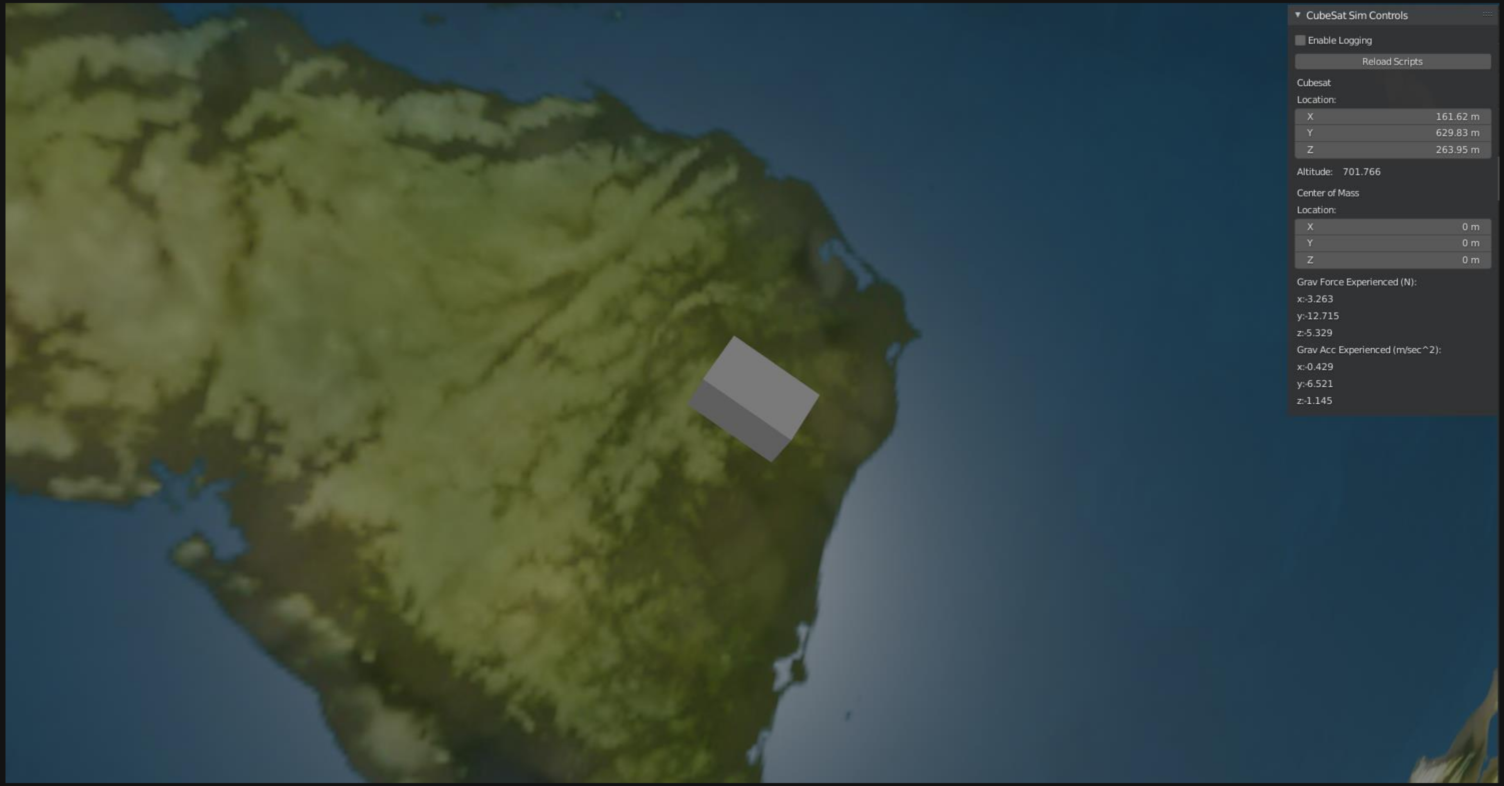


Maximum Wilder-Smith, Michael Pham, Matteo Gironda, Kevin Kwik, Michael Gee, Ryan Toomer, Nicholas Shewchuk, Ashley McBean, Tarek Elsharhawy

California State Polytechnic University Pomona – Bronco Space

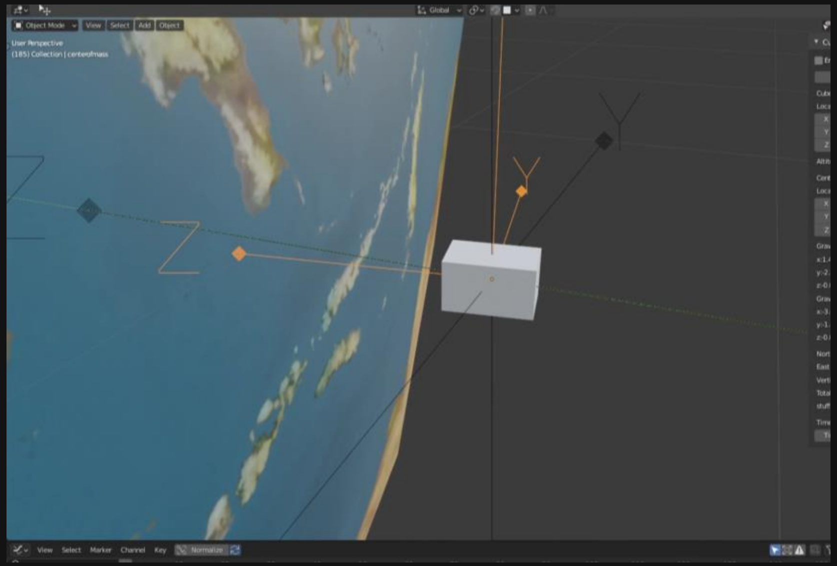
Objectives

- Purpose is to create a machine learning friendly environment for virtual missions



Objectives

- Requires kinematics simulation, sensor simulation, and TensorFlow integration



Objectives

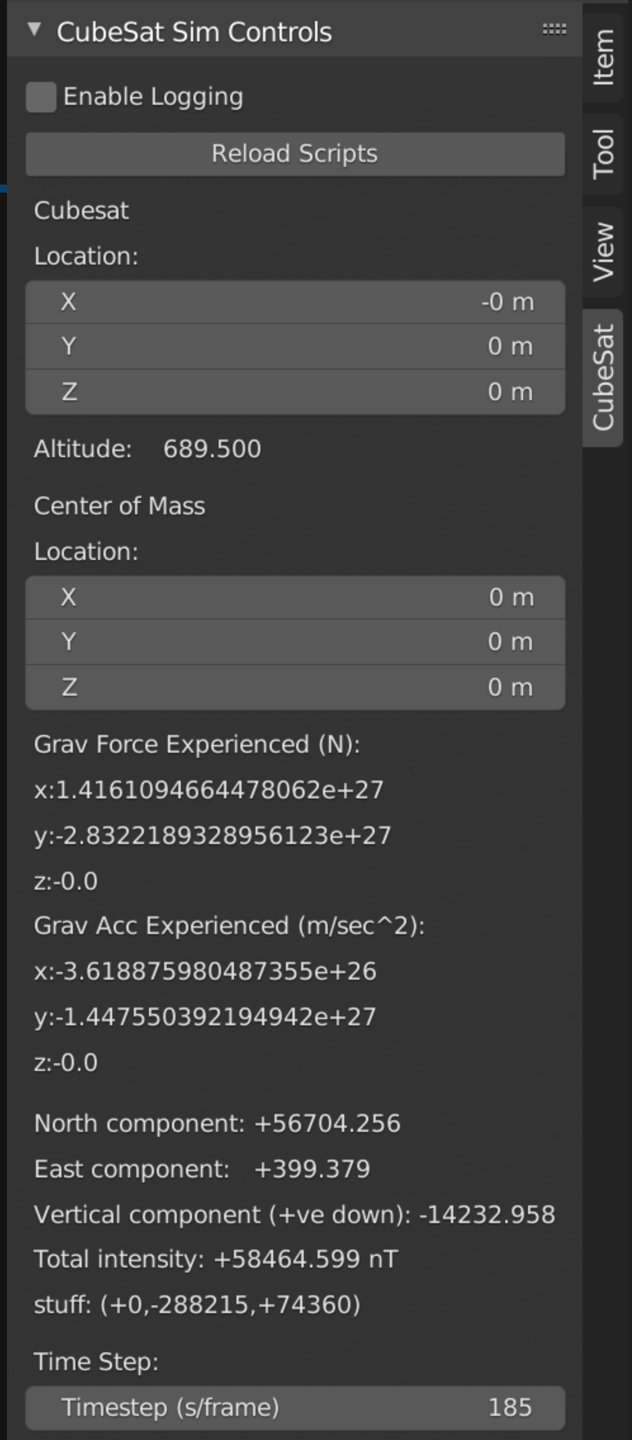
- Simulator is built off the Blender 3D animation system
- Uses a series of custom Python scripts

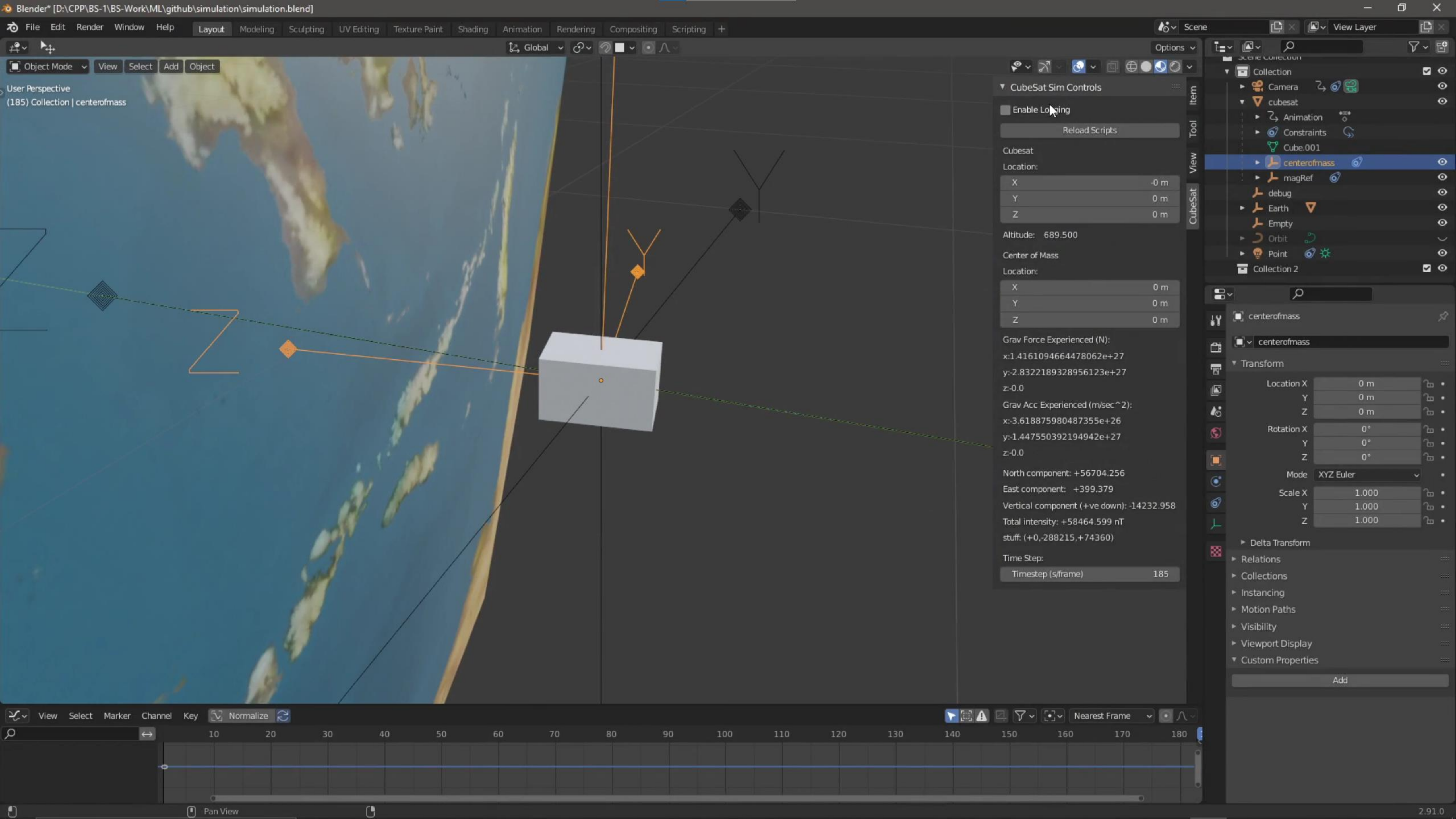
Simulation Environment: Blender UI

- Open source and built with Python
- Open to modification
- Pre-existing documentation and community
- Easy to create additions
- Designed to ease development

Simulation Environment: Custom UI

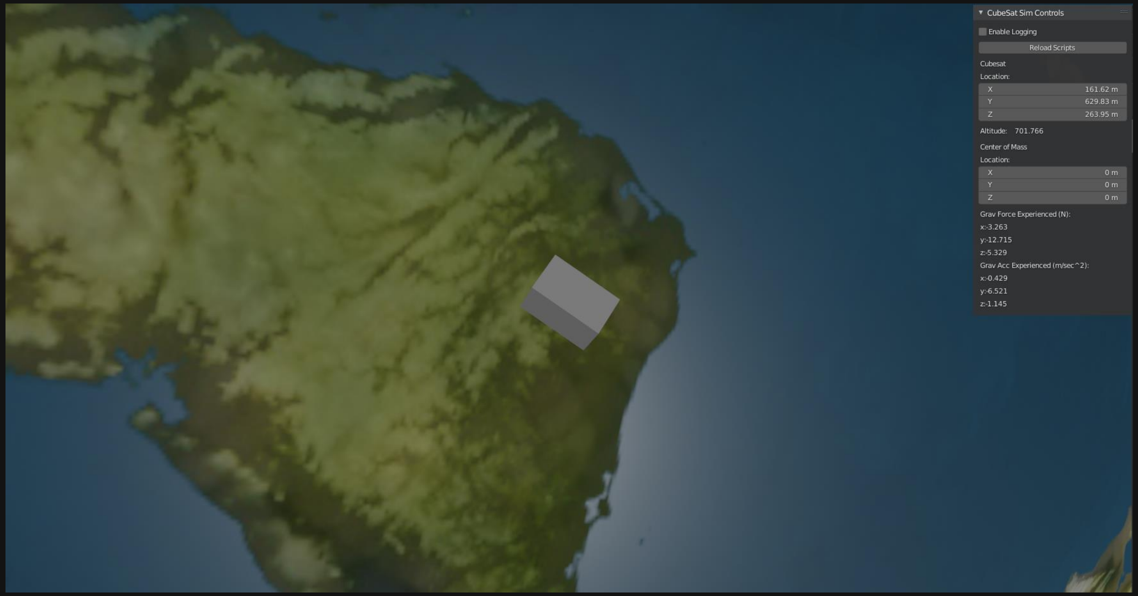
- Mainly within existing UI panel
- Provides:
  - Debugging controls
    - Help with active development of system
  - CubeSat properties
    - Location, altitude, center of mass
  - Calculated environmental properties
    - Gravitational force/acceleration, magnetic forces




Blender* [D:\CPP\BS-1\BS-Work\ML\github\simulation\simulation.blend]
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
Scene
View Layer
Global
Options
Object Mode View Select Add Object
User Perspective
(185) Collection | centerofmass
CubeSat Sim Controls
Enable Logging
Reload Scripts
Cubesat
Location:
X -0 m
Y 0 m
Z 0 m
Altitude: 689.500
Center of Mass
Location:
X 0 m
Y 0 m
Z 0 m
Grav Force Experienced (N):
x:1.4161094664478062e+27
y:-2.8322189328956123e+27
z:-0.0
Grav Acc Experienced (m/sec^2):
x:-3.618875980487355e+26
y:-1.447550392194942e+27
z:-0.0
North component: +56704.256
East component: +399.379
Vertical component (+ve down): -14232.958
Total intensity: +58464.599 nT
stuff: (+0,-288215,+74360)
Time Step:
Timestep (s/frame) 185
Item
Tool
View
CubeSat
Collection
Camera
cubesat
Animation
Constraints
Cube.001
centerofmass
magRef
debug
Earth
Empty
Orbit
Point
Collection 2
centerofmass
Transform
Location X 0 m
Y 0 m
Z 0 m
Rotation X 0°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Visibility
Viewport Display
Custom Properties
Add
View Select Marker Channel Key Normalize
Nearest Frame
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Pan View
2.91.0

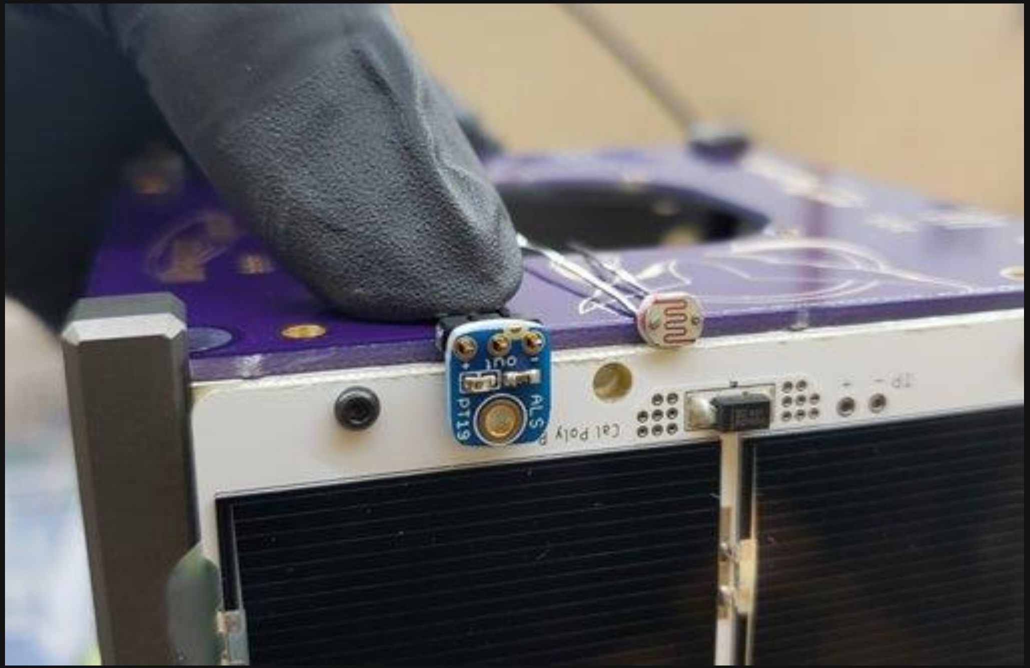
Spacecraft Simulation: Gravity Properties

- Gravitational Force and acceleration experienced by satellite
- Both vector quantities
- Calculations based on Newton's Law of Universal Gravitation



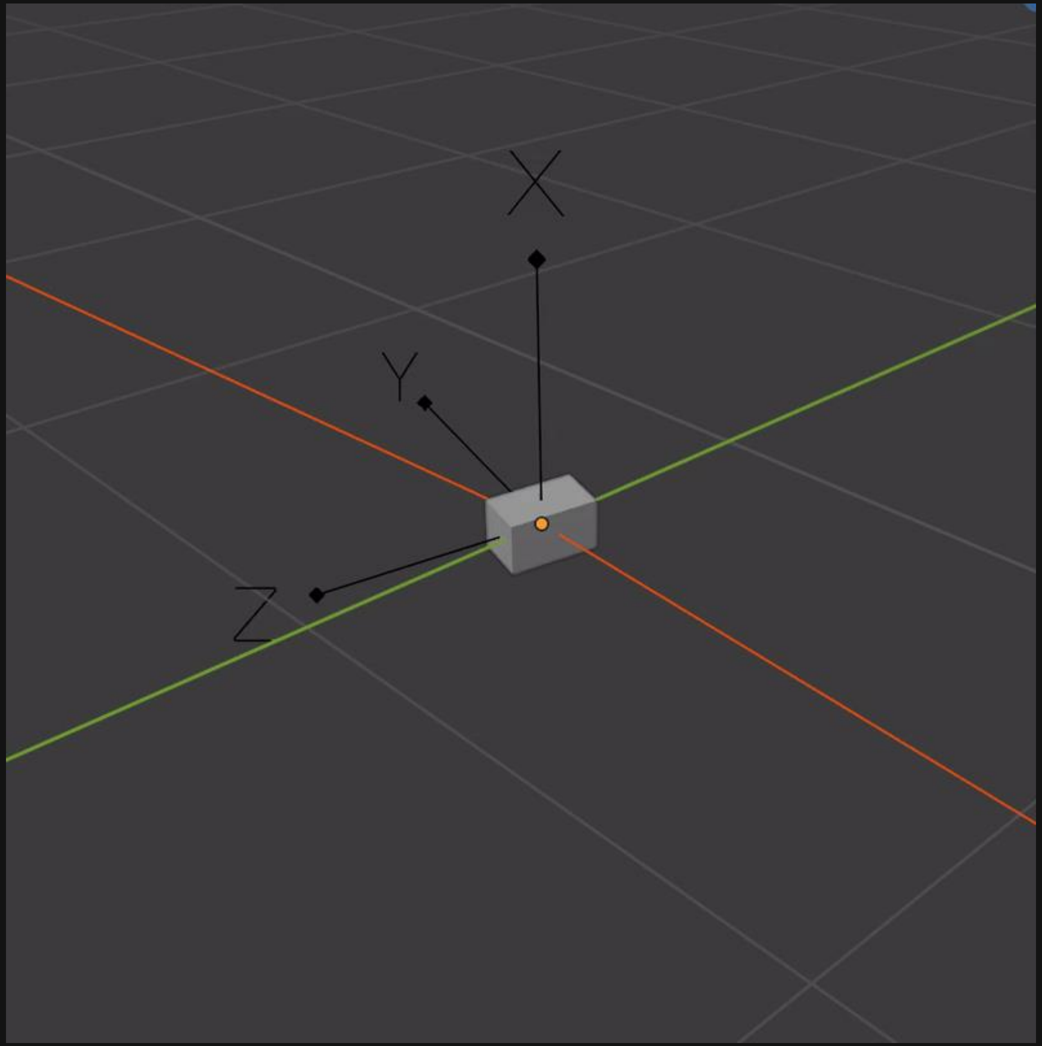
Spacecraft Simulation: Sun Sensors

- Rough approximation made through trigonometric approach
- Does not consider light reflected off any planetary bodies
- Minimum and maximum lux approximations based on data from an ALS-PT19-315C photoresistor



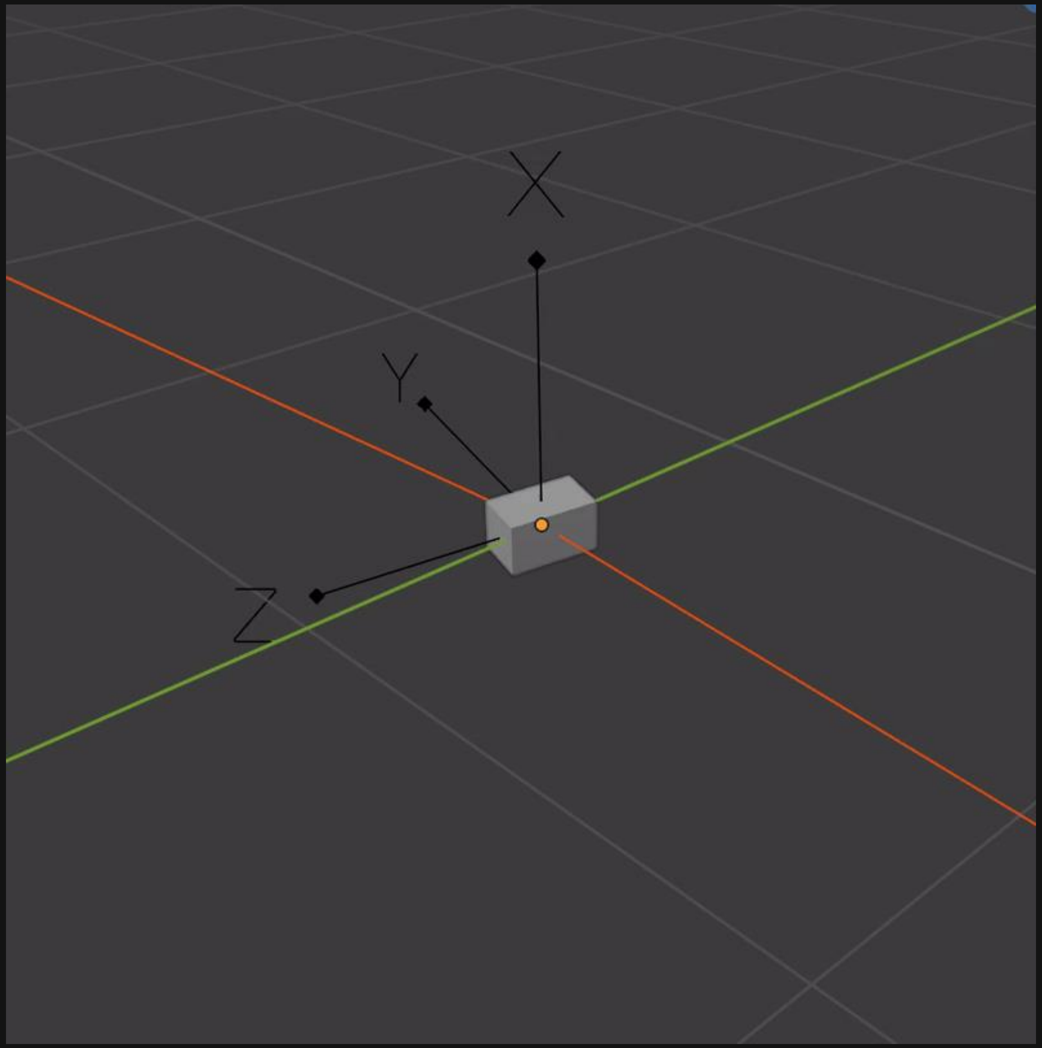
Spacecraft Simulation: Angular Velocity

- Defined as the change in the satellite's orientation per frame step
- Two Functions
  - Reads angular velocity every frame step
  - Calculates new positions based off desired angular velocity input
- All calculations use Quaternions



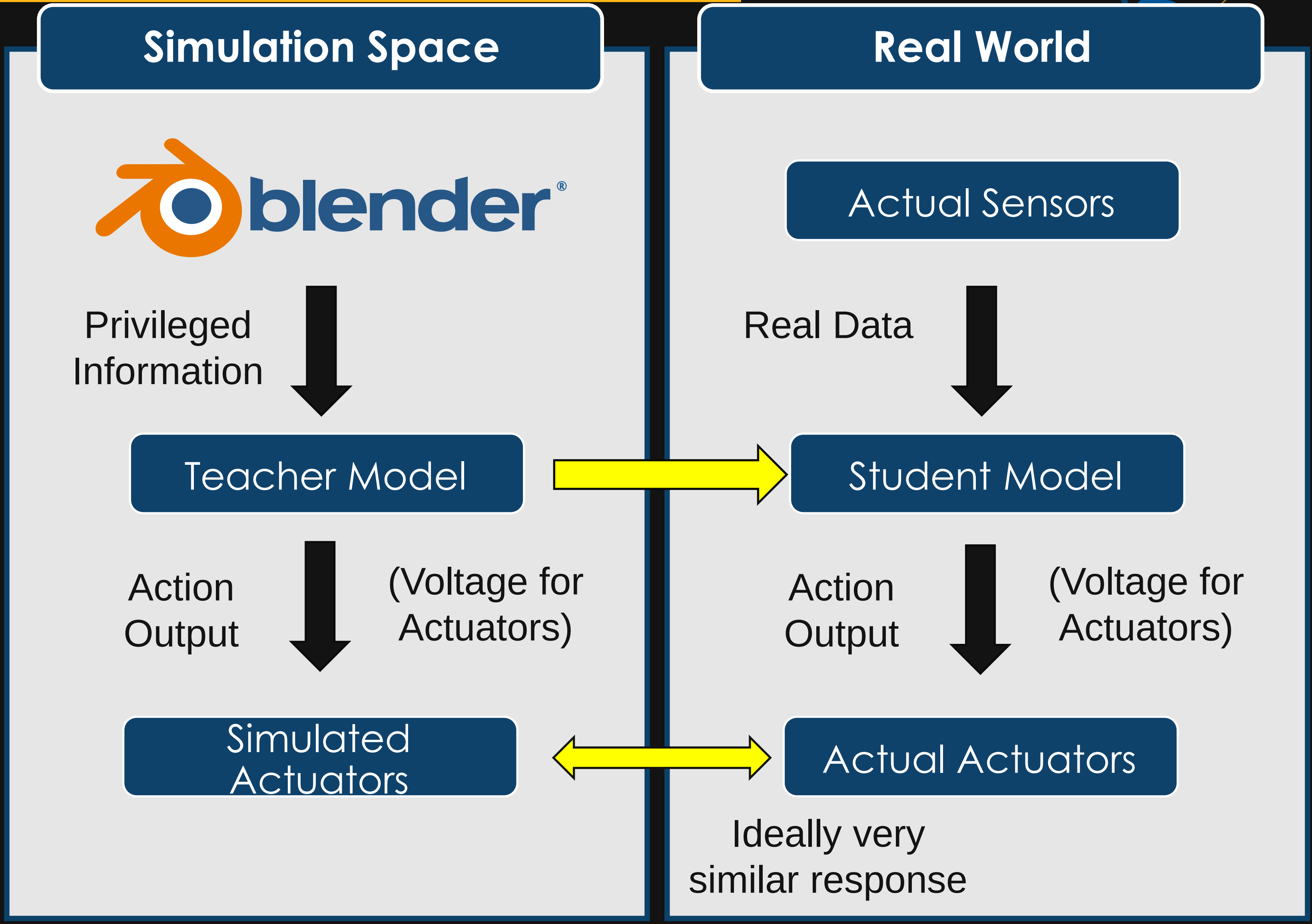
Angular Velocity Demonstration

- Orientation set function
- Angular velocity set to 10 degrees about each axis
- Time step of 1 frame per second



Control System: ML Architecture

- Based on the Teacher-Student architecture
- Teacher will be heavily trained on the simulation with all data
- Student model then uses teacher weights and biases as reference for real world operation
- Similar abstract understanding between models



Control System: Use Case

- Currently designed for a 3 magnetorqurer system
- Inputs consist of 6 sun sensors and IMU data
- Easy to adjust actuation/input
- Model will be verified in laboratory testing
- Designed to be deployed as a small network file compatible with Python and C++ software