CASCADE

Cubesat Active Systematic CApture DEvice

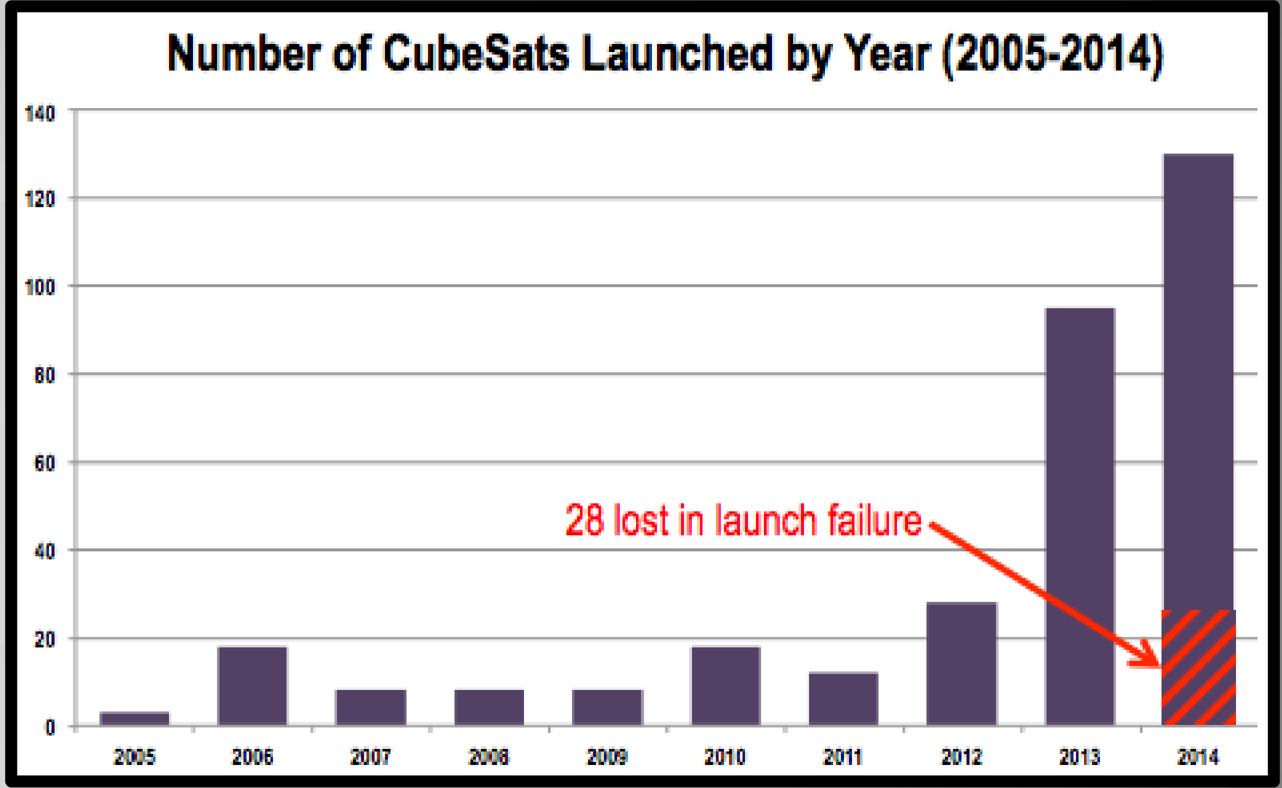
Andrew McBride
Timothy Kiley
Tony Ly
Haoyu Li

Noel Puldon
Keegan Sotebeer
Matthew Fromm
Chad Eberl
Morgan Tilong

Project Purpose and Objectives

Project Motivation

- With the growing industry for CubeSats a method of capturing an uncontrollable CubeSat is desirable.
- Existing CubeSats have little or no propulsive capabilities, with no ability to change the orbit drastically and leaving them stuck if major failures occur.
- Sierra Nevada Corporation would use a capture device and vision system in order to **recover** and **repurpose** CubeSats.



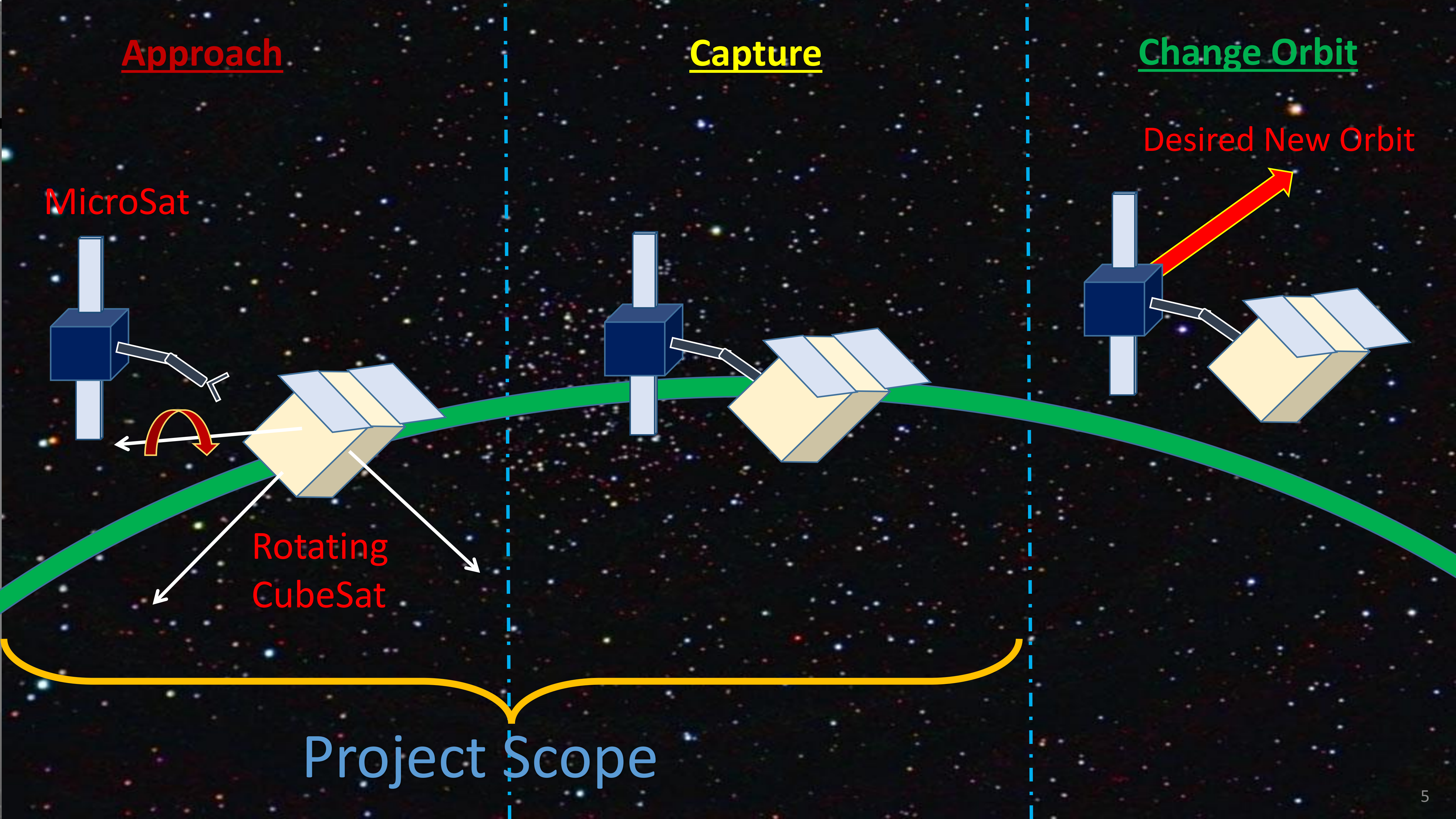
http://www.sia.org/wp-content/uploads/2015/06/Mktg15-SSIR-2015-FINAL-Compressed.pdf

Project Objectives | Design Solution | CPES | Design Reqs. | Risk Plan | Verification & Validation | Project Planning

Project Statement

Team CASCADE will demonstrate the implementation of an algorithm to **autonomously** capture a rotating 3U CubeSat model.

In order to accomplish this goal, Team CASCADE will design and build a CubeSat Recovery System Testbed (**CRST**) used to validate both the **algorithm** and a physical **capture device**.

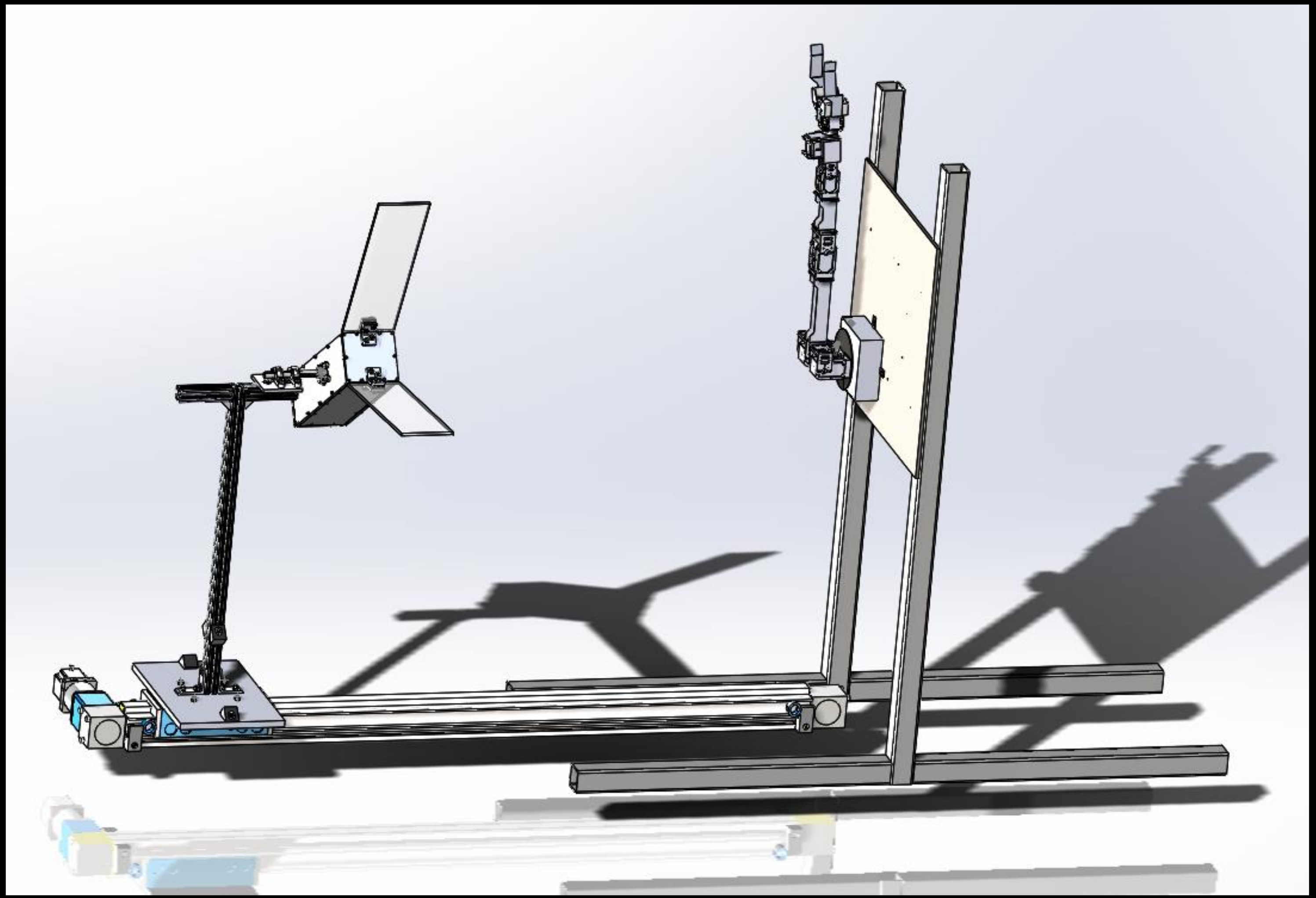

Approach
Capture
Change Orbit
Desired New Orbit
MicroSat
Rotating
CubeSat
Project Scope

Project Scope

Provide a proof of concept of a capture device and algorithm which could be used to recover a CubeSat rotating on **one axis.** This is a step towards an algorithm that could be used for a robotic arm mounted on a SNC micro-satellite .

Our system will use the **Recuv Motion Detection Lab** that gives inputs into our algorithm that are the same as those that would be available from a vision system on orbit.

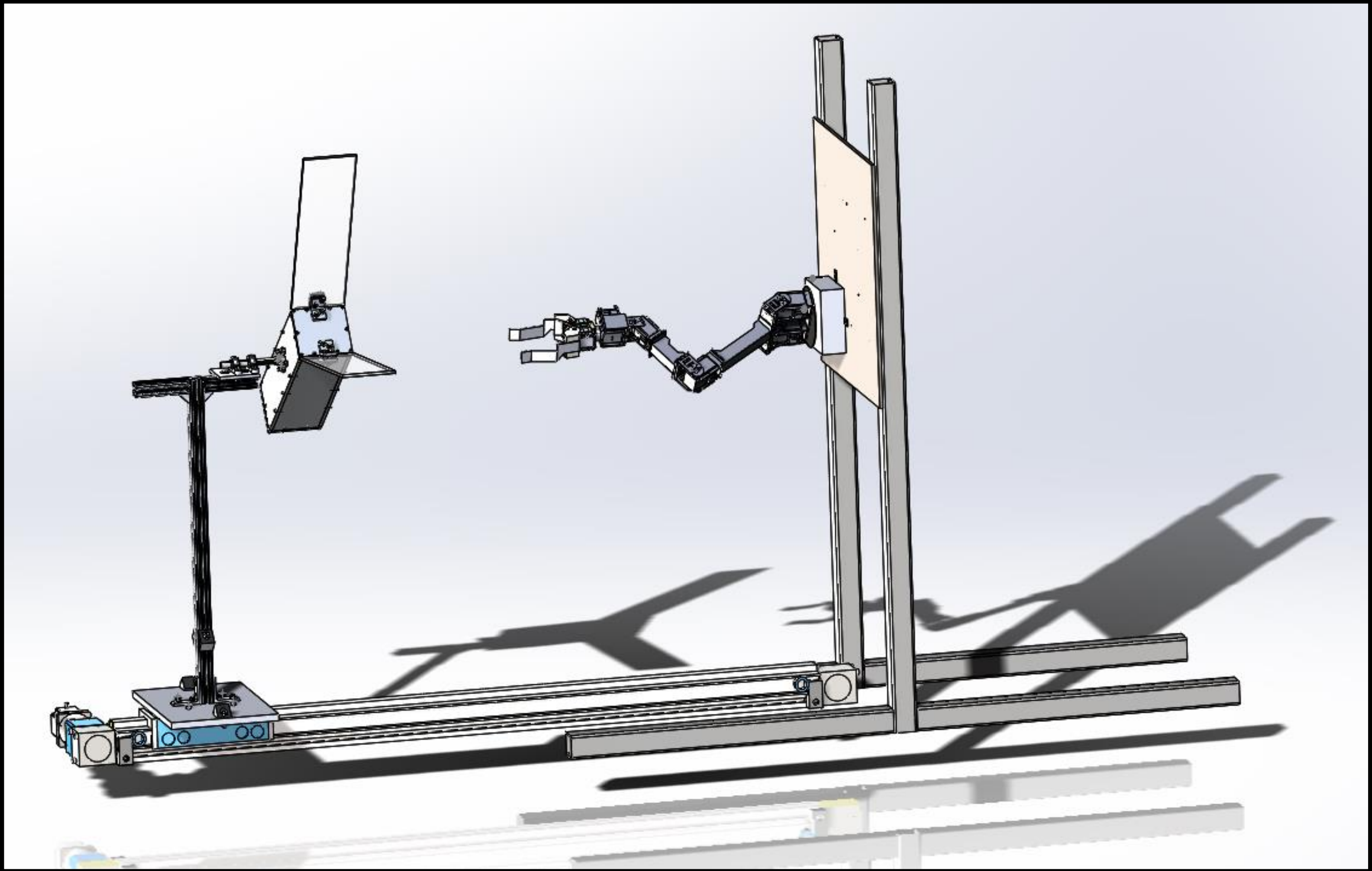
Test Bed CONOPS



0.) Initiation of Demonstration

- Arm stowed in zero torque configuration
- Vicon Cameras start transmitting data to LabView on a personal Laptop.
- LabView used to start the rotation of the CubeSat

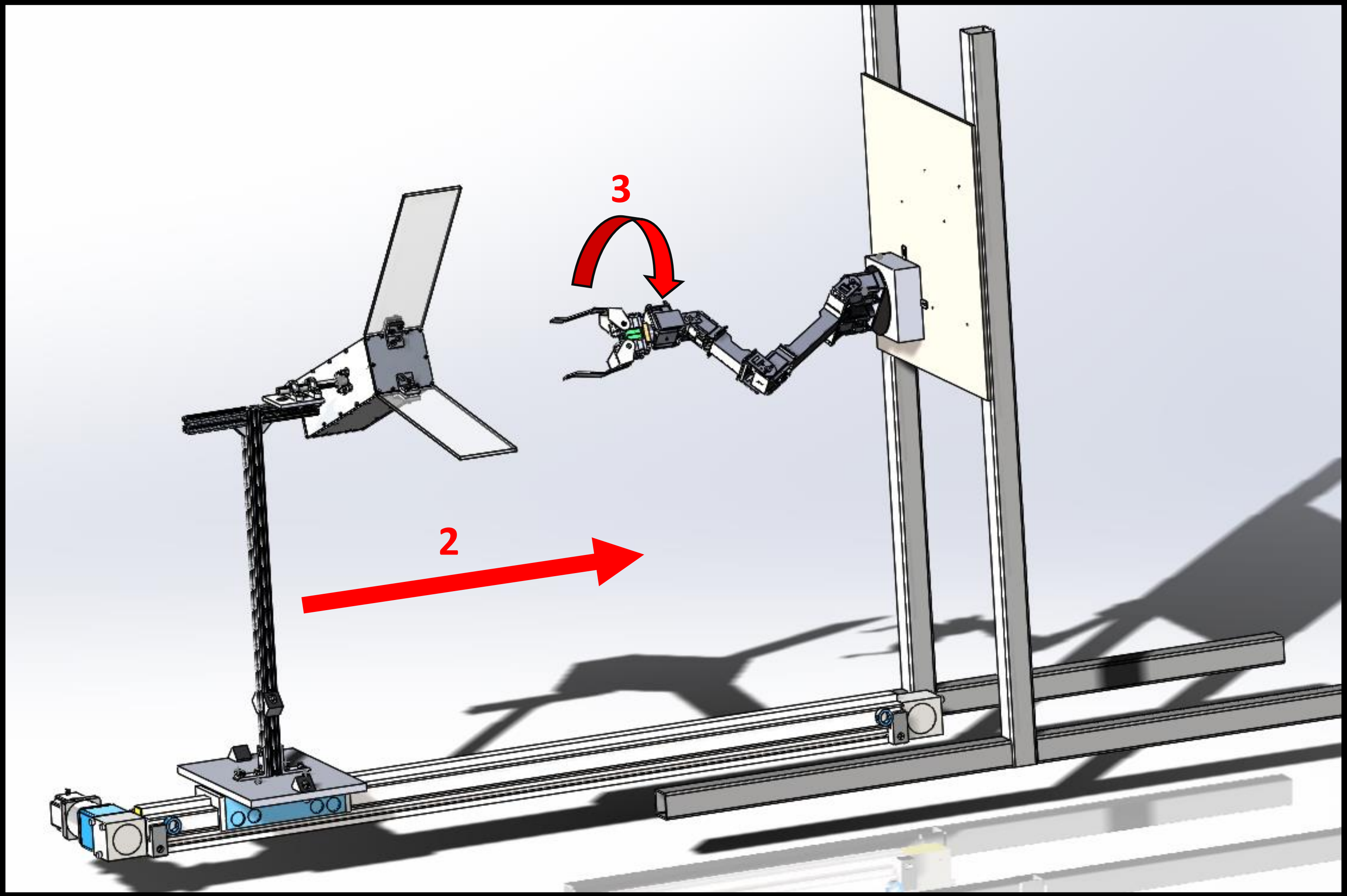
Test Bed CONOPS



<u>1.) Move to Axis of Rotation</u>

- Using Vicon data the axis of rotation will be calculated in LabView
- Commands are sent to the arm to move the end effector to the axis of rotation

Test Bed CONOPS



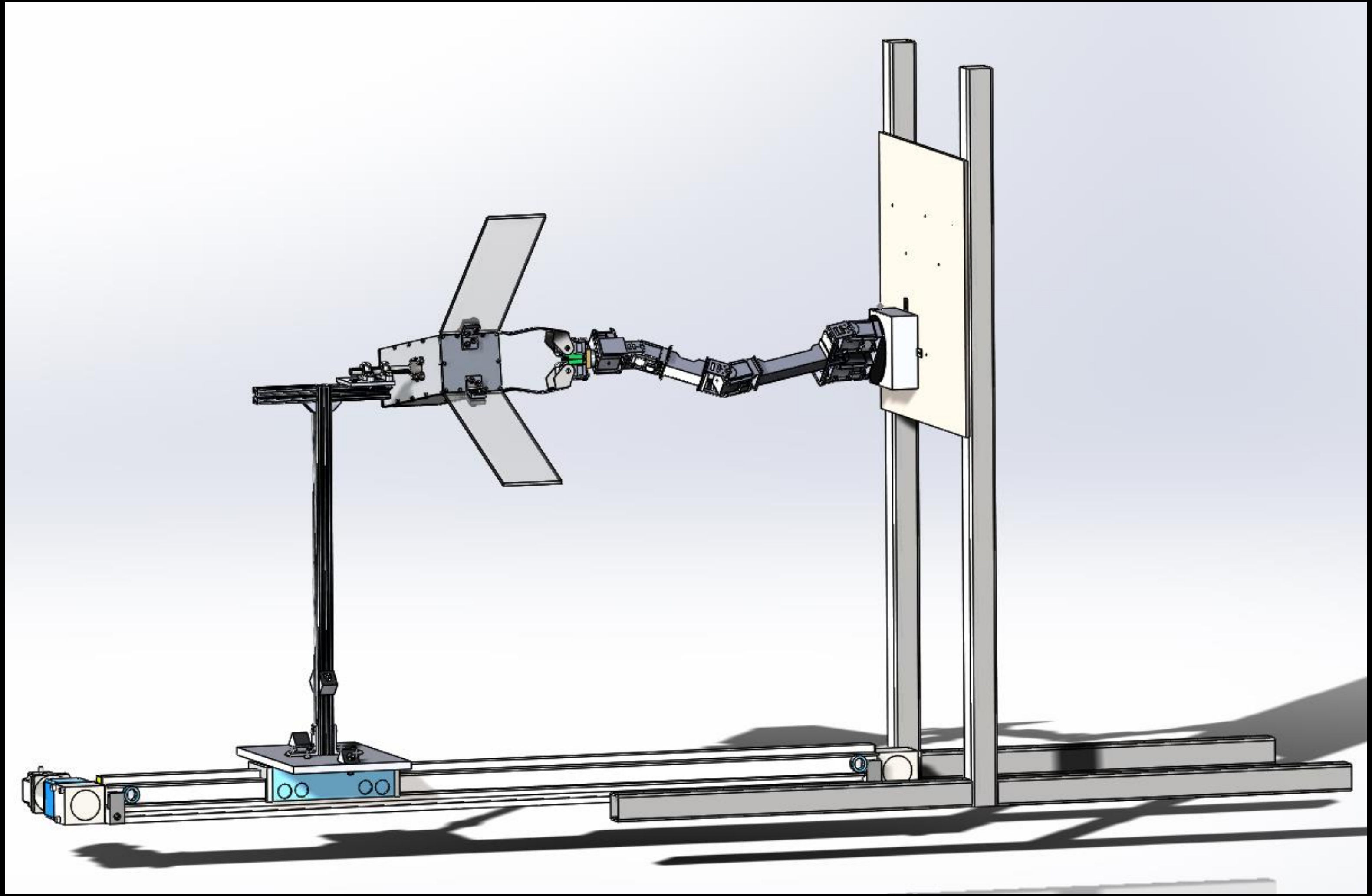
2.) Translate CubeSat

- This phase represents the closing of the relative position between the CubeSat and Capture device
- In space thrusters would be used to approach the CubeSat

3.) Wrist Rotation

- Using Vicon Data the wrist will be sent commands to match the rotation of the CubeSat.

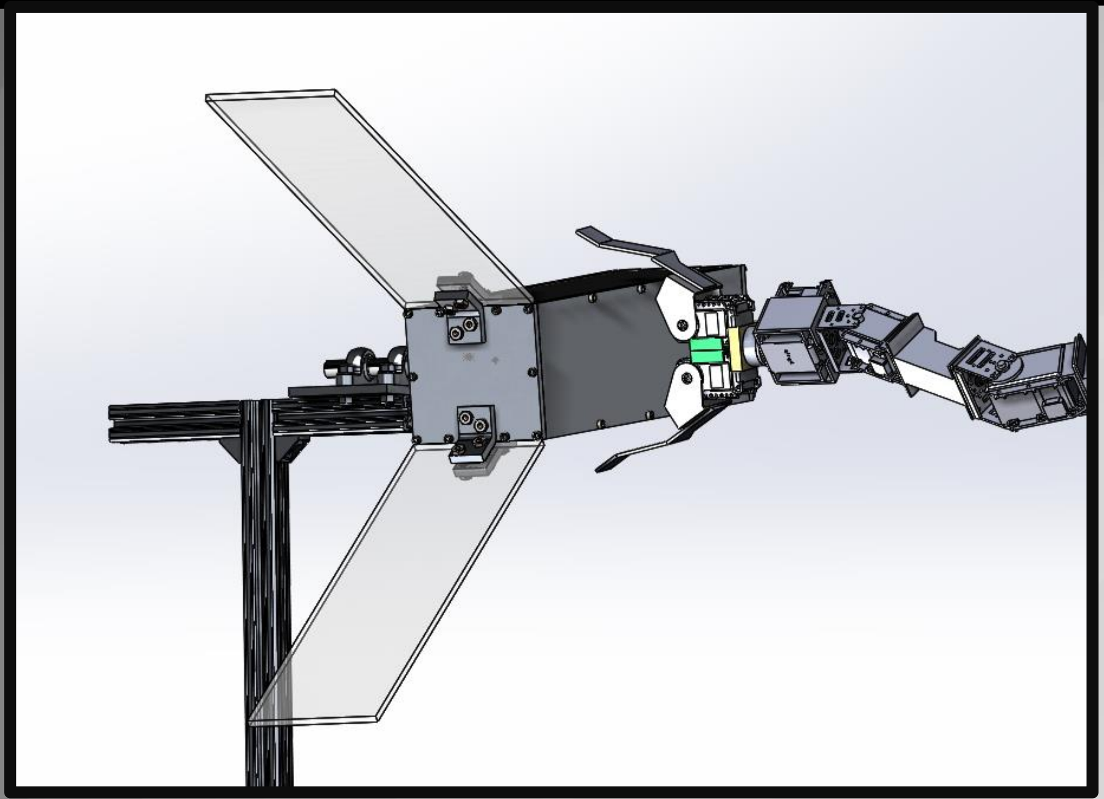
Test Bed CONOPS

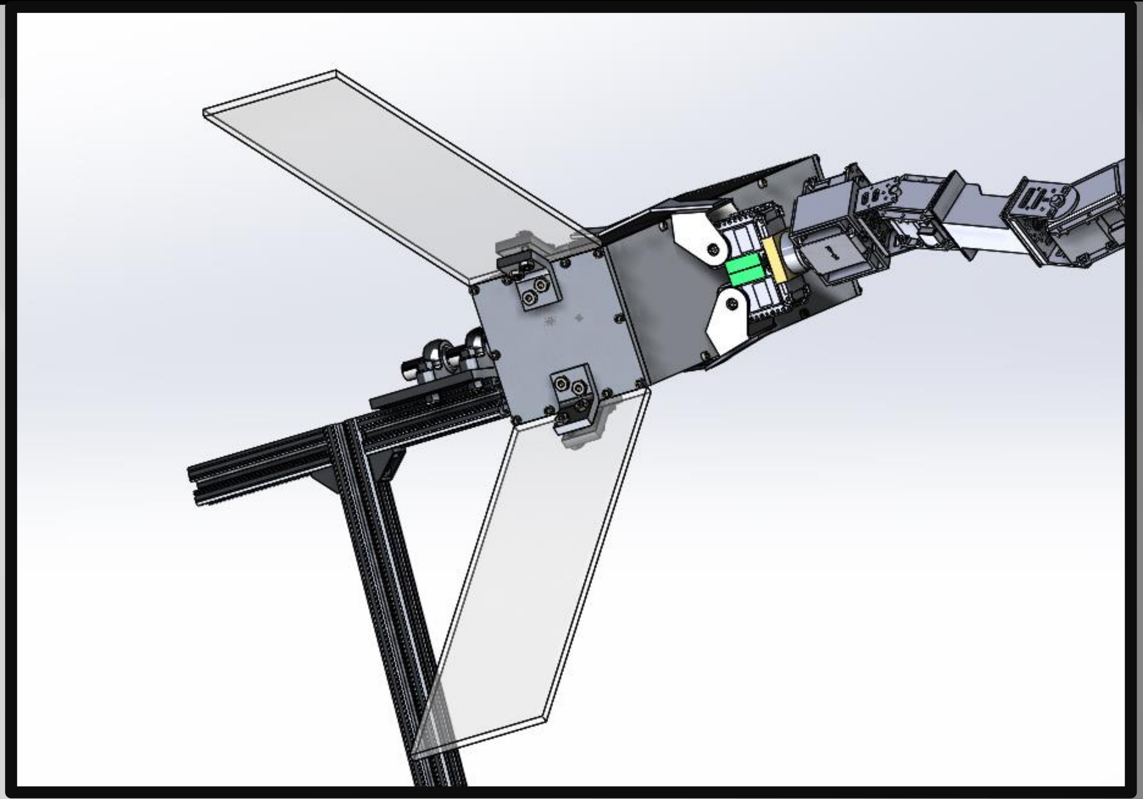


4.) Extend Arm

- Using Vicon data the arm will be sent commands to position the end effector over the CubeSat model.

Test Bed CONOPS





<u>5.) Claw Closure</u>

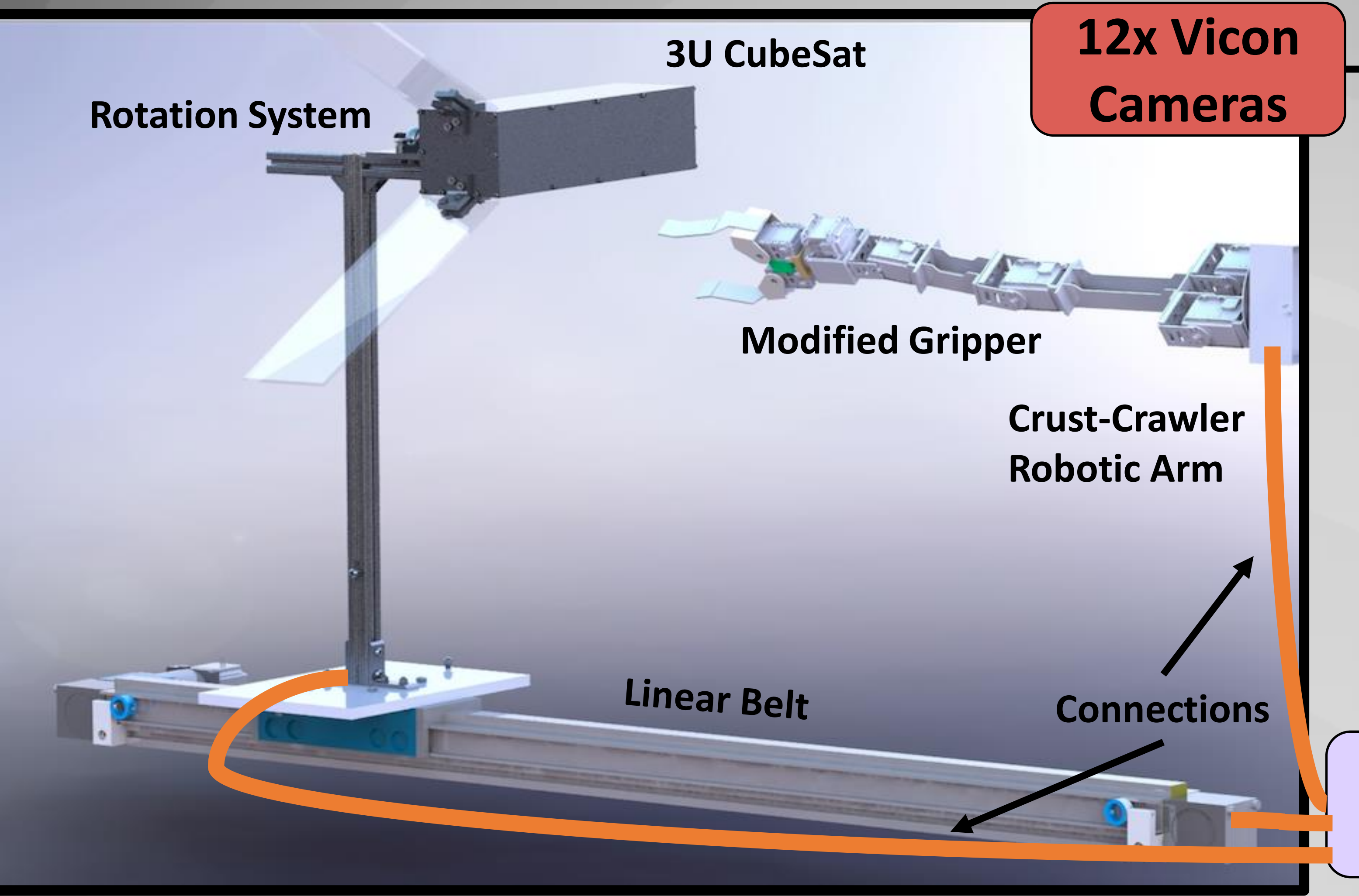
- Finally the claw is closed on the CubeSat surface, capture is confirmed, servo and motors are stopped, and the CubeSat is held for 5 minutes until released.

Levels of Success

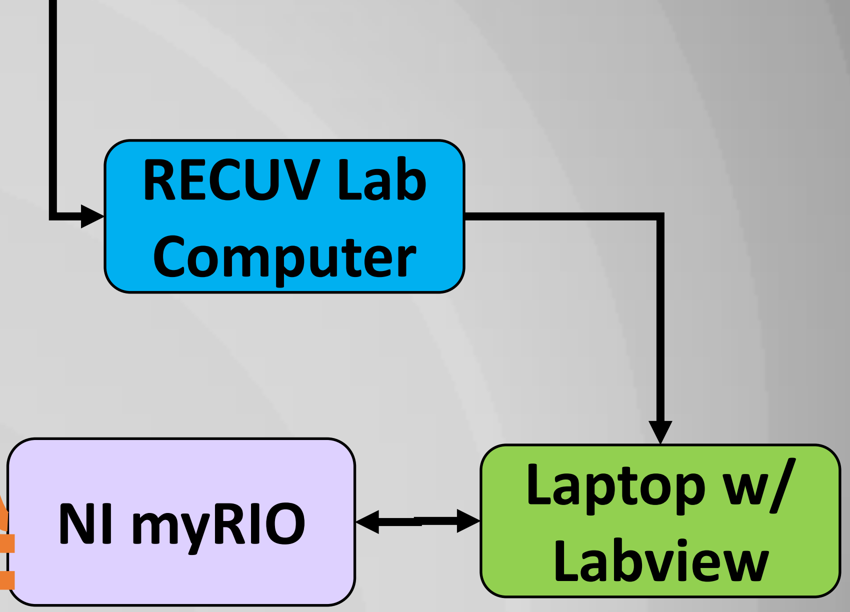
Success Levels	Testbed Demonstration	Capture Device Control
Level 1	1 DOF Translation	Open Loop (Commanded 1 Step at a time)
Level 2	1 DOF Rotation	Closed Loop
Level 3	1 DOF Translation and 1 DOF Rotation	

Design Solution

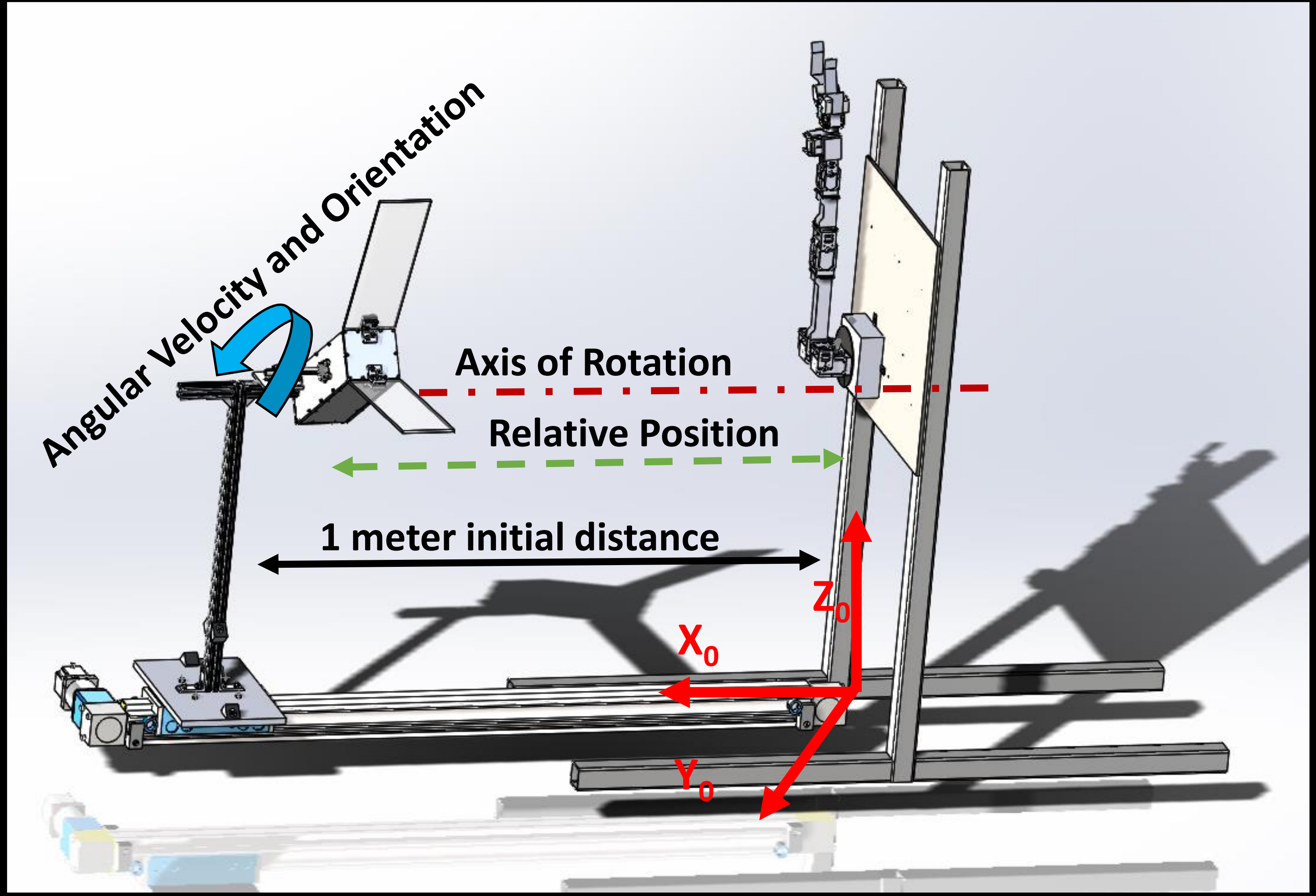
Design Solution



- Linear Belt for **CubeSat Translation**
- Rotation System for **CubeSat Spin**
- Vicon Cameras for **Feedback**
- LabView and NI myRIO for **CDHS**
- Arm and Gripper for **Capture**



Design Solution



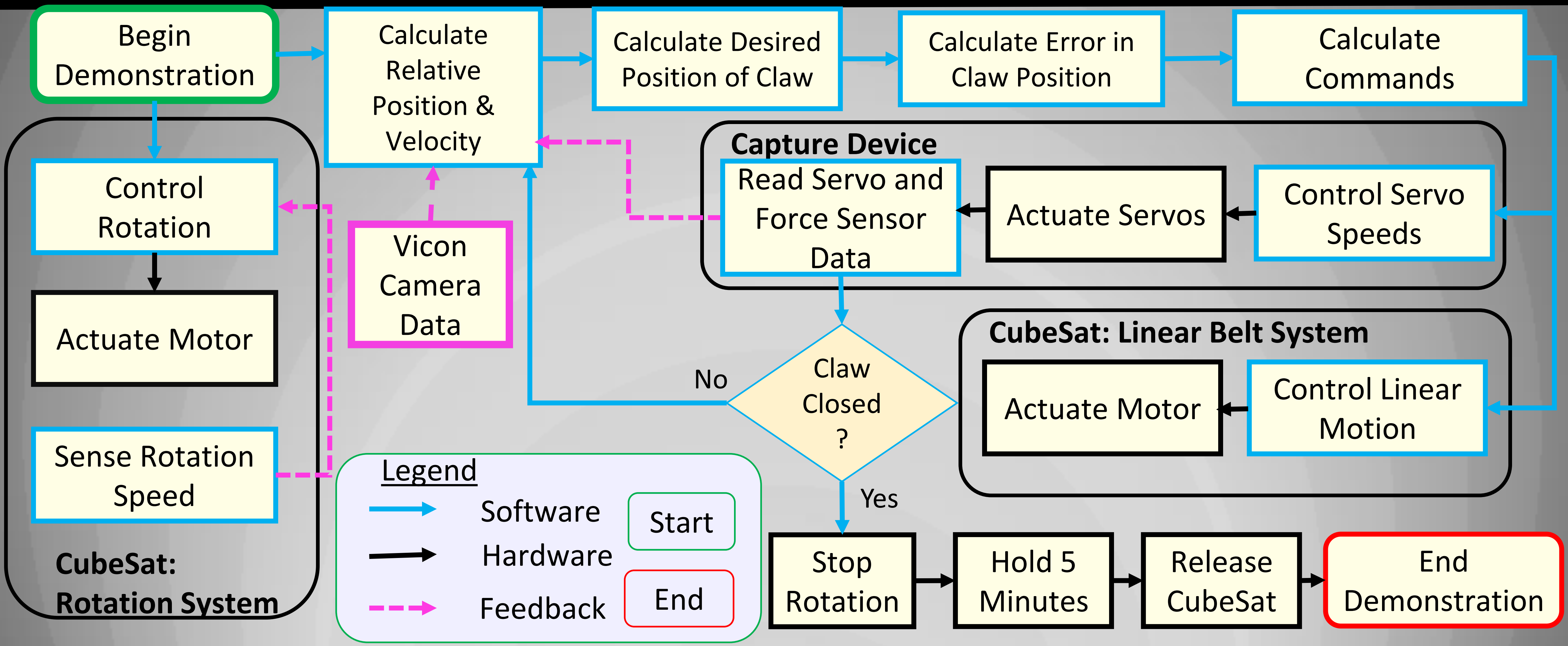
Vicon Data Used to Find:

- Angular Velocity and Orientation of CubeSat
- Axis of Rotation
- Relative position between CubeSat centroid and base of the arm.

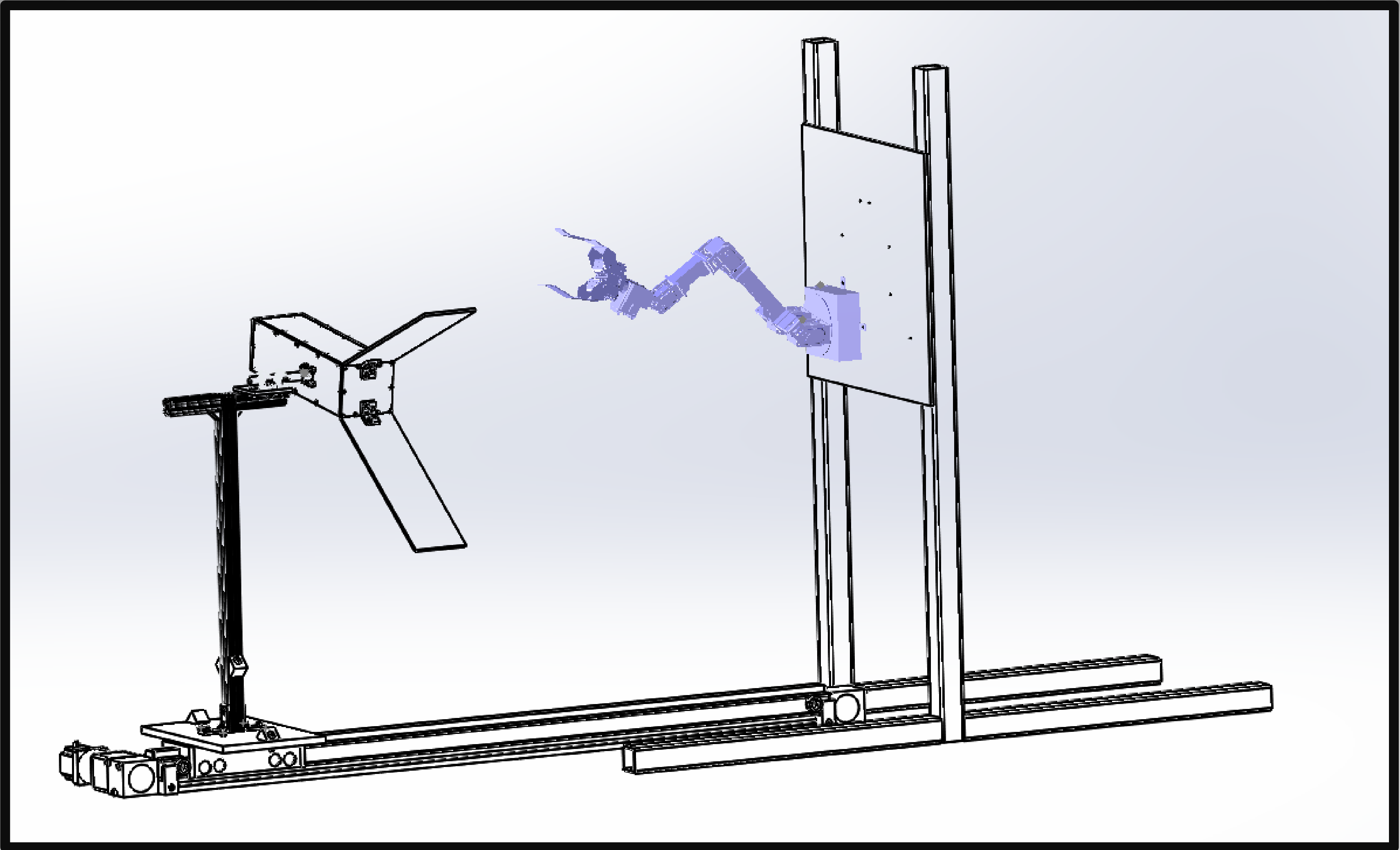
Initial Conditions:

- 1 meter away on x-axis
- Arm in zero torque configuration
- Arm Base **offset from the axis of rotation** on both y and z axis.

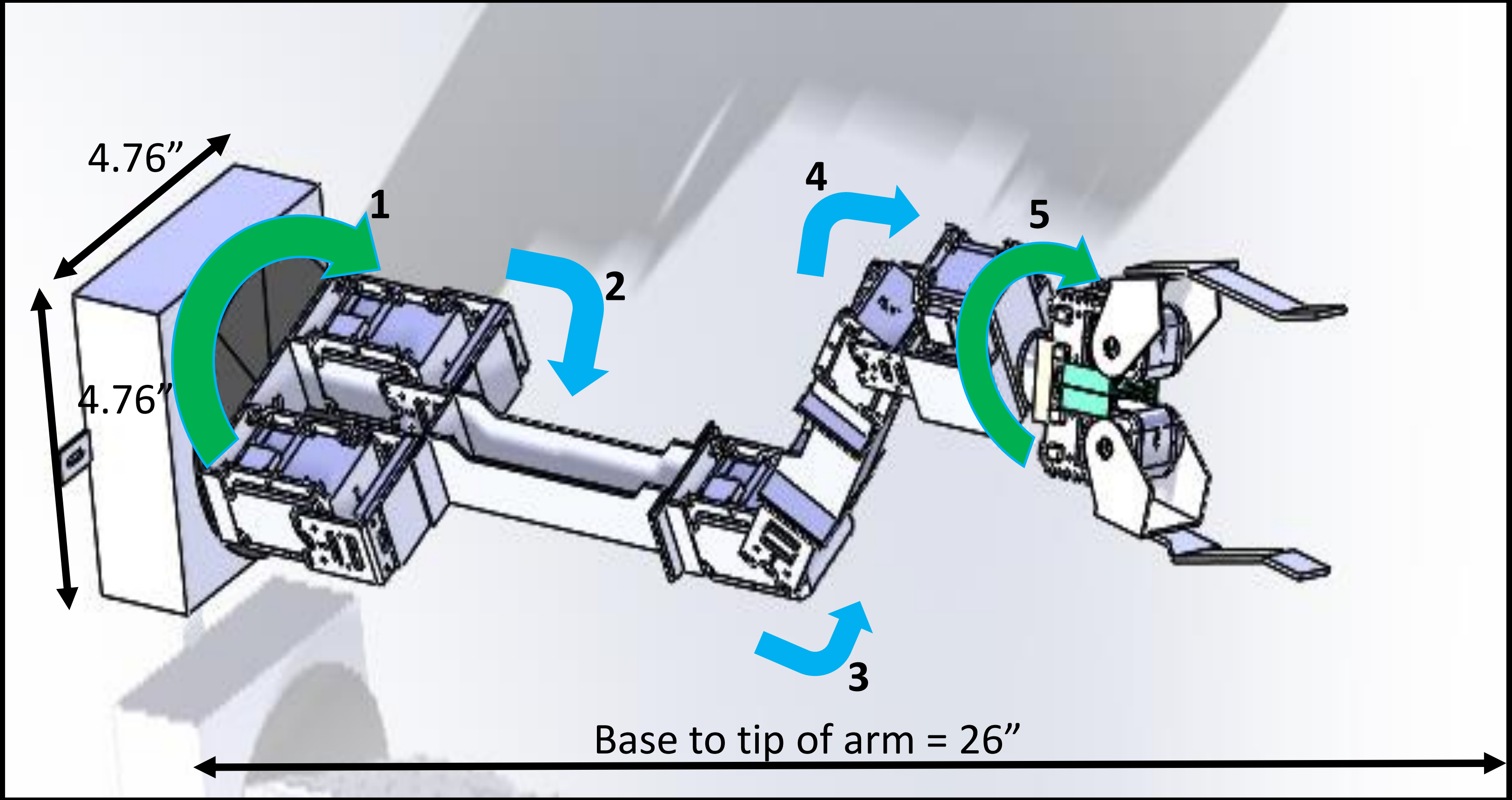
Functional Flow Diagram



Robotic Arm

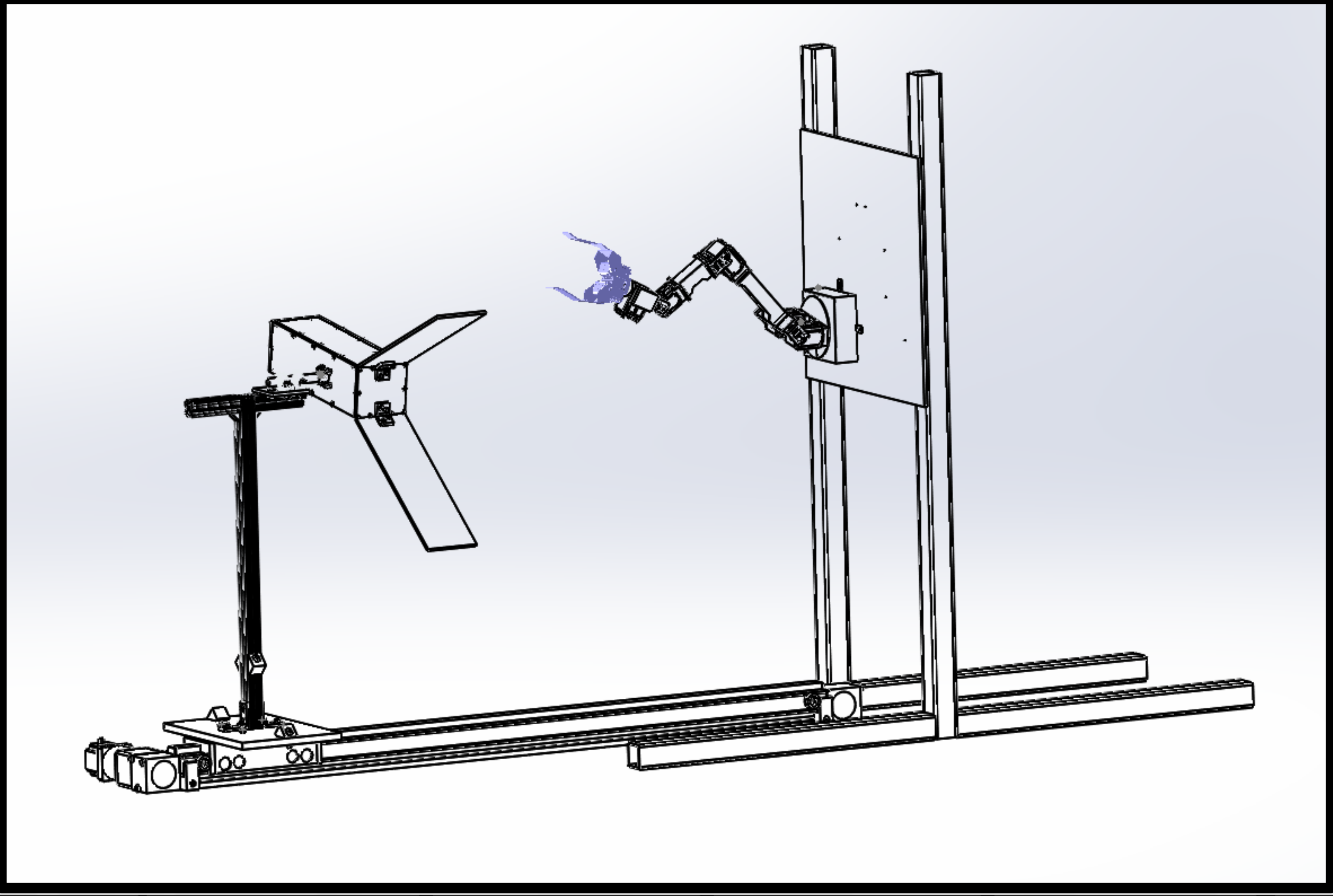


Robotic Arm

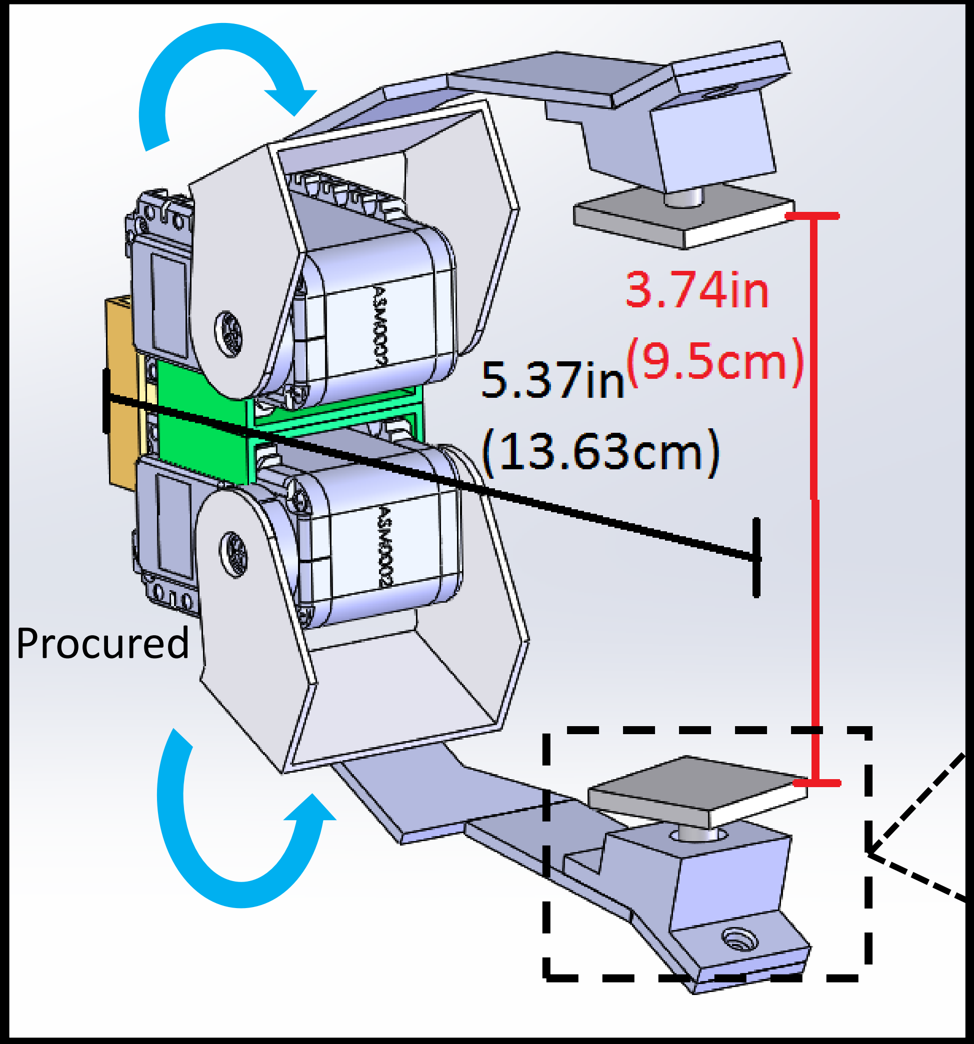


Specifications:
5 DOF arm with 6 servos -Base Rotation (1 Servo) -Base Bend (2 Servos) -Midway Bend (1 Servo) -Wrist Bend (1 Servo) -Wrist Rotator (1 Servo)
Crust-Crawler Modular Arm: we select the link lengths and servo sizes.
Cost: $2,754

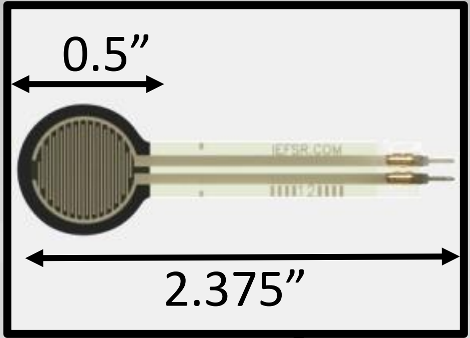
End Effector



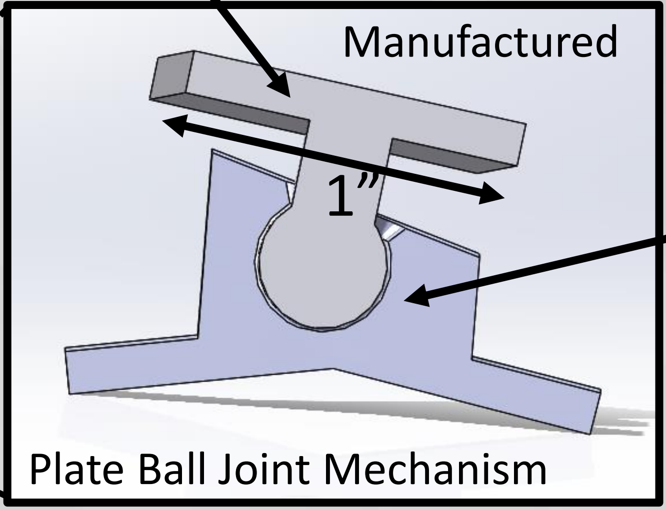
End Effector



Force Sensing Resistor:



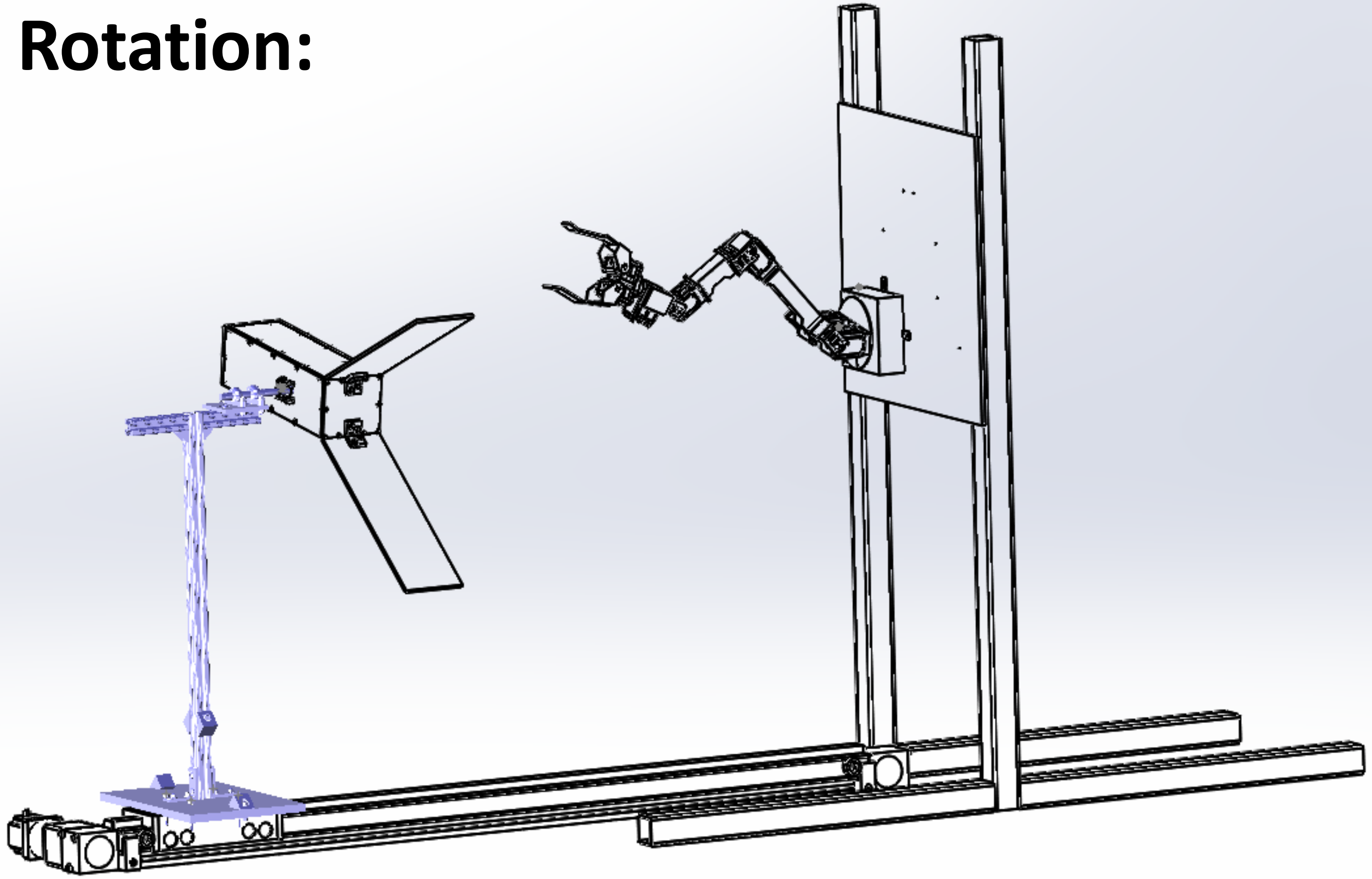
Specifications:
-Rotational Closing with 2 Servos -3D Printed Gripper Plates
Will use force sensing resistors on the plates to verify capture.
Difference amp and low pass filter circuit used for signal conditioning.



Allows for rotation for passive compliance

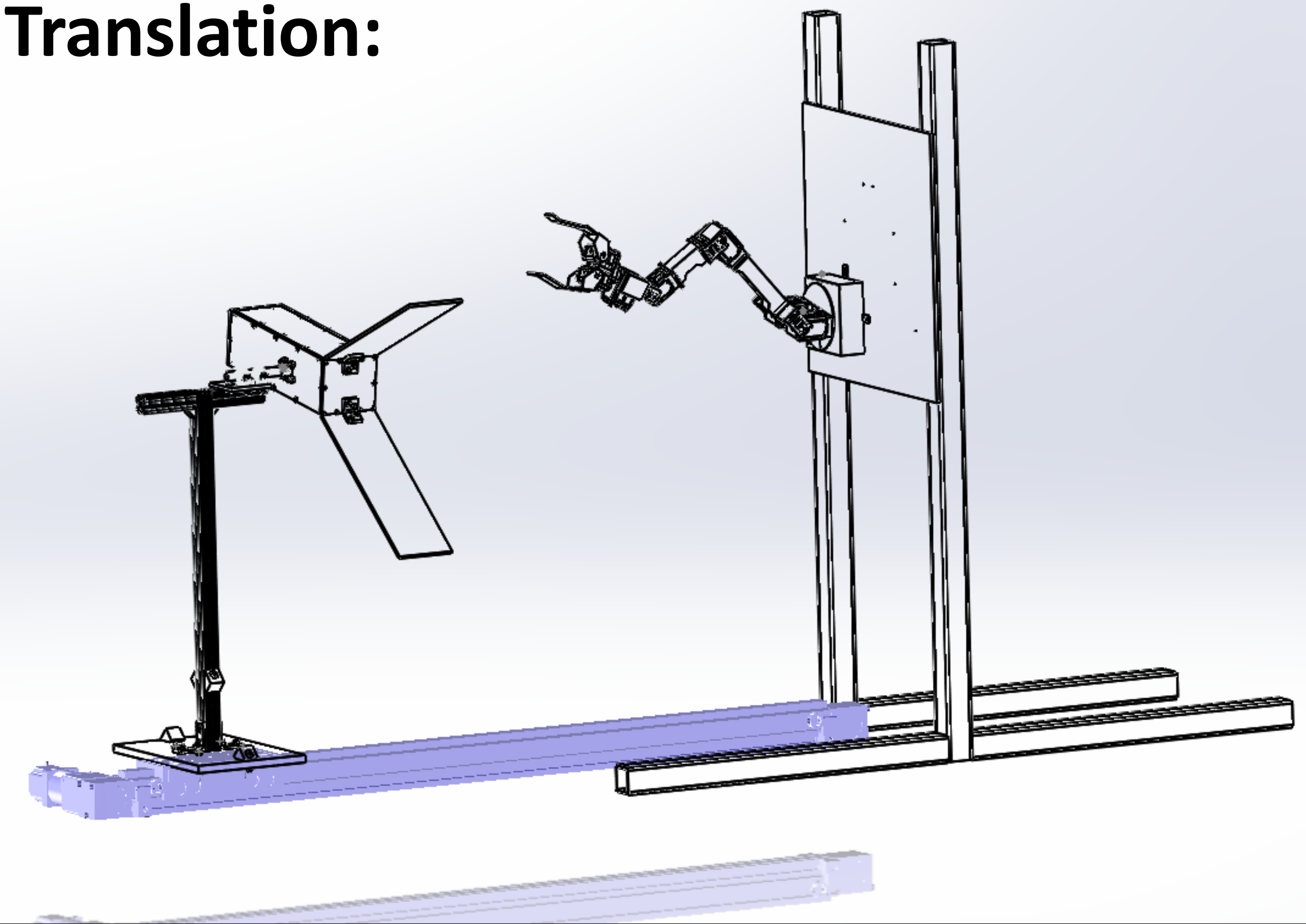
CubeSat Motion

Rotation:

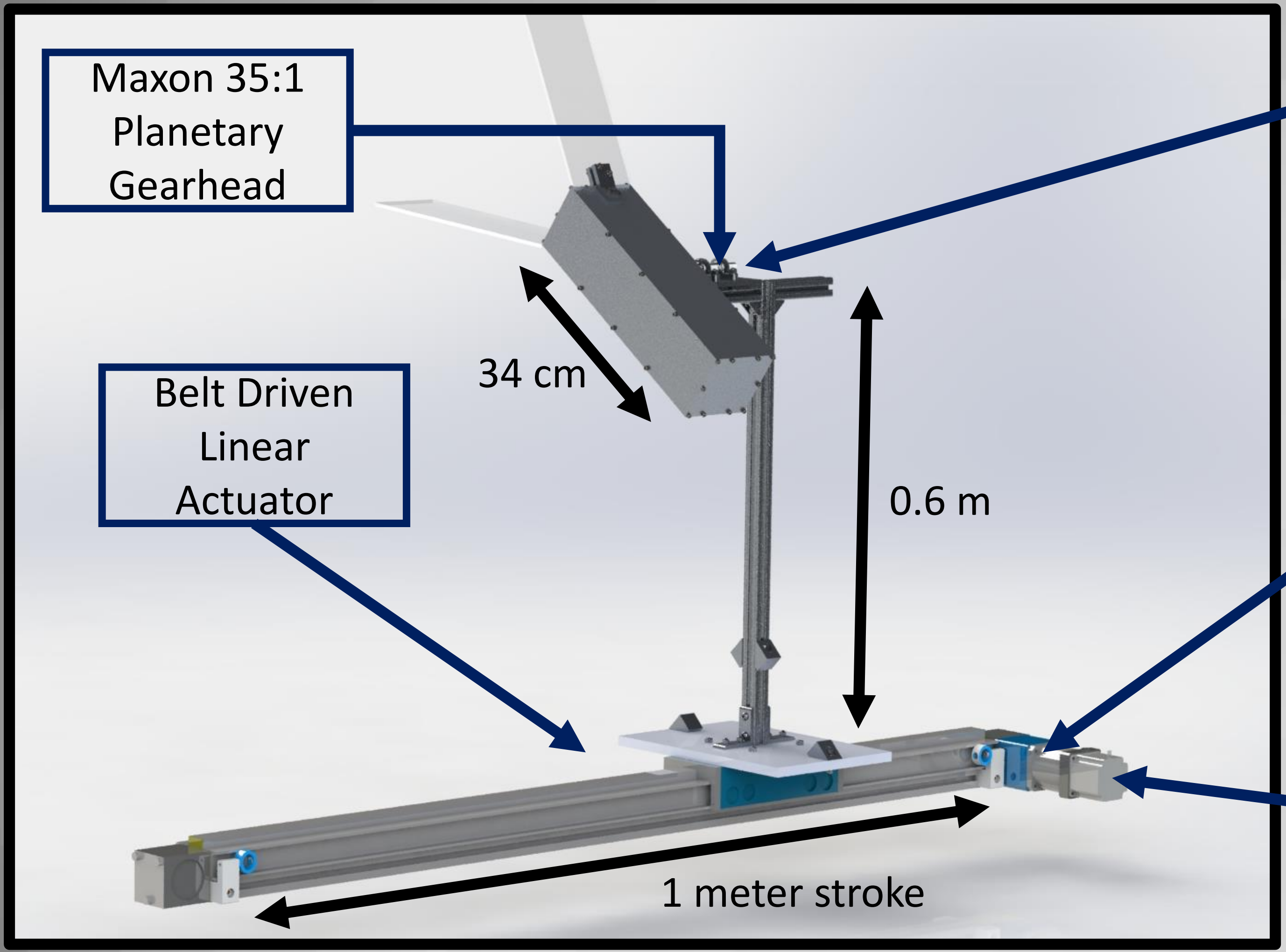


CubeSat Motion

Translation:

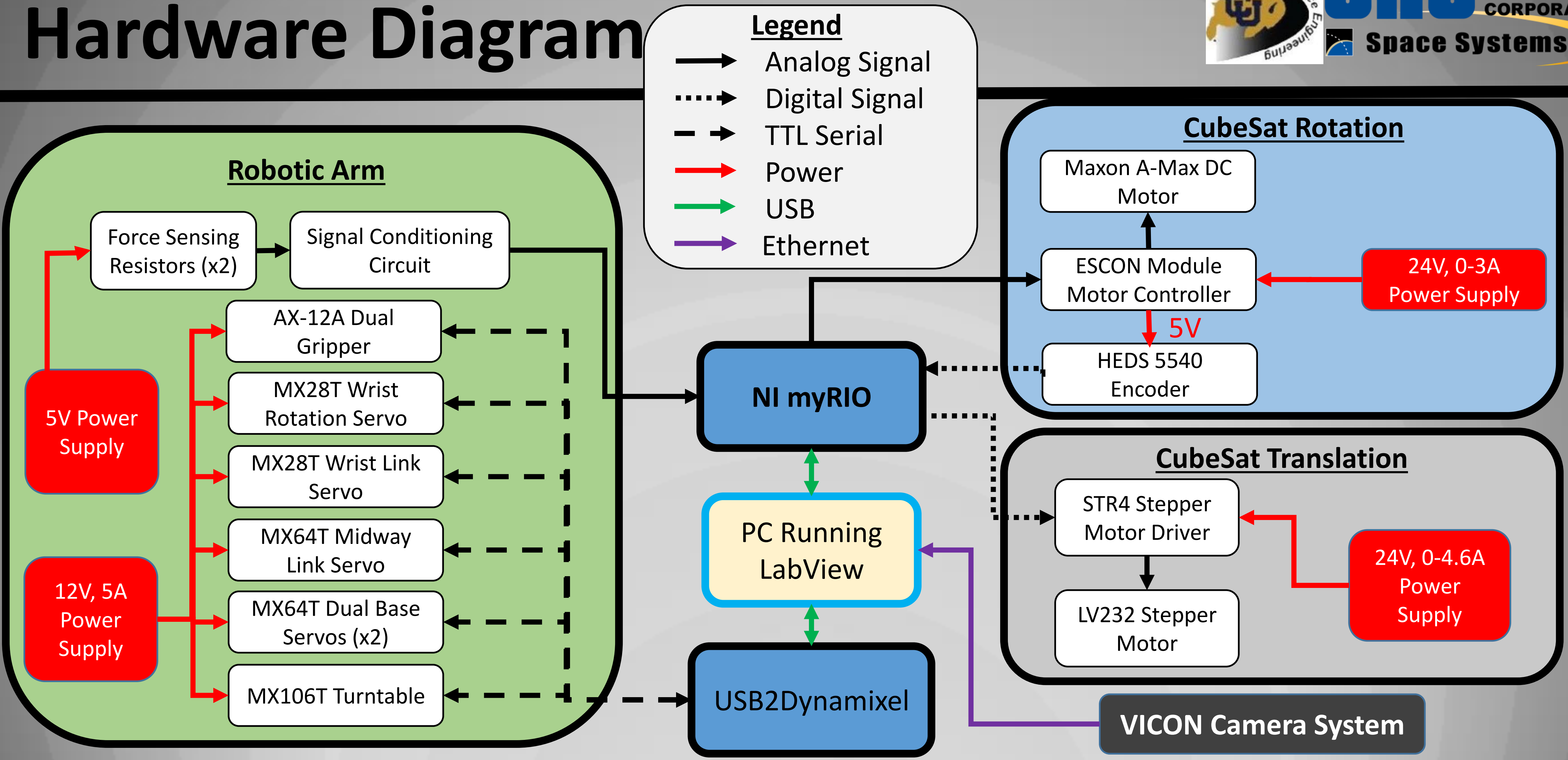


CubeSat Motion

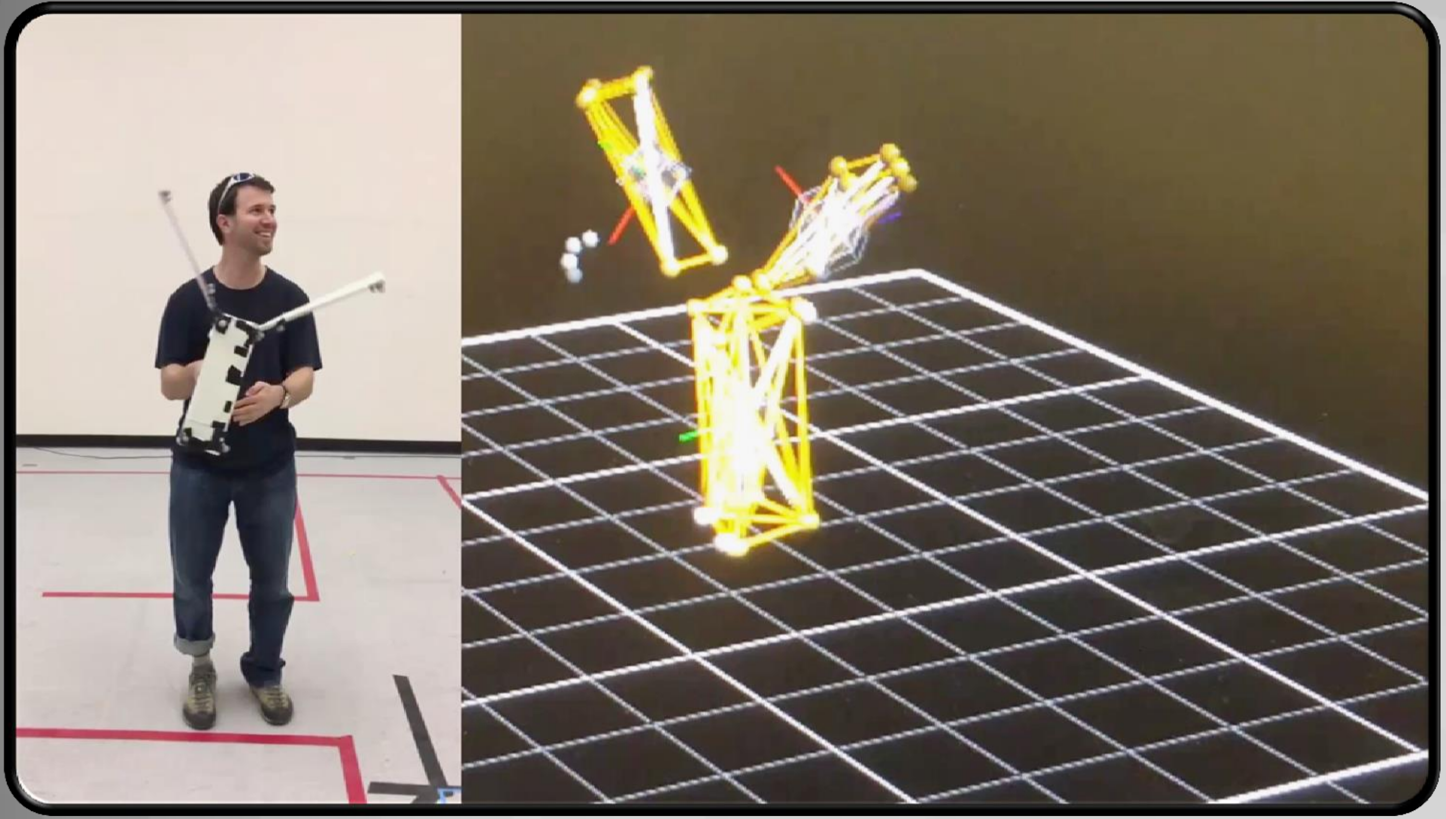


- Linear motion controlled using Vicon feedback in LabView. This simulates proximity operations.
- Rotation held at constant 3 deg/s with PID controller in LabView.

Hardware Diagram



RECUV Vision Lab



Uses infrared Cameras that track an object defined by reflector spheres.

The Vicon System gives us:

[x, y, z, roll, pitch, yaw]

For each sphere and object centroid at 100 Hz. This will be used to find the **axis of rotation** and CubeSat **relative position**.

Specifications:
Interface: Port 800 Ethernet
Average Static Error: 0.0775 mm
Average Latency: 16.87 ms

Software: Overview

(Master)

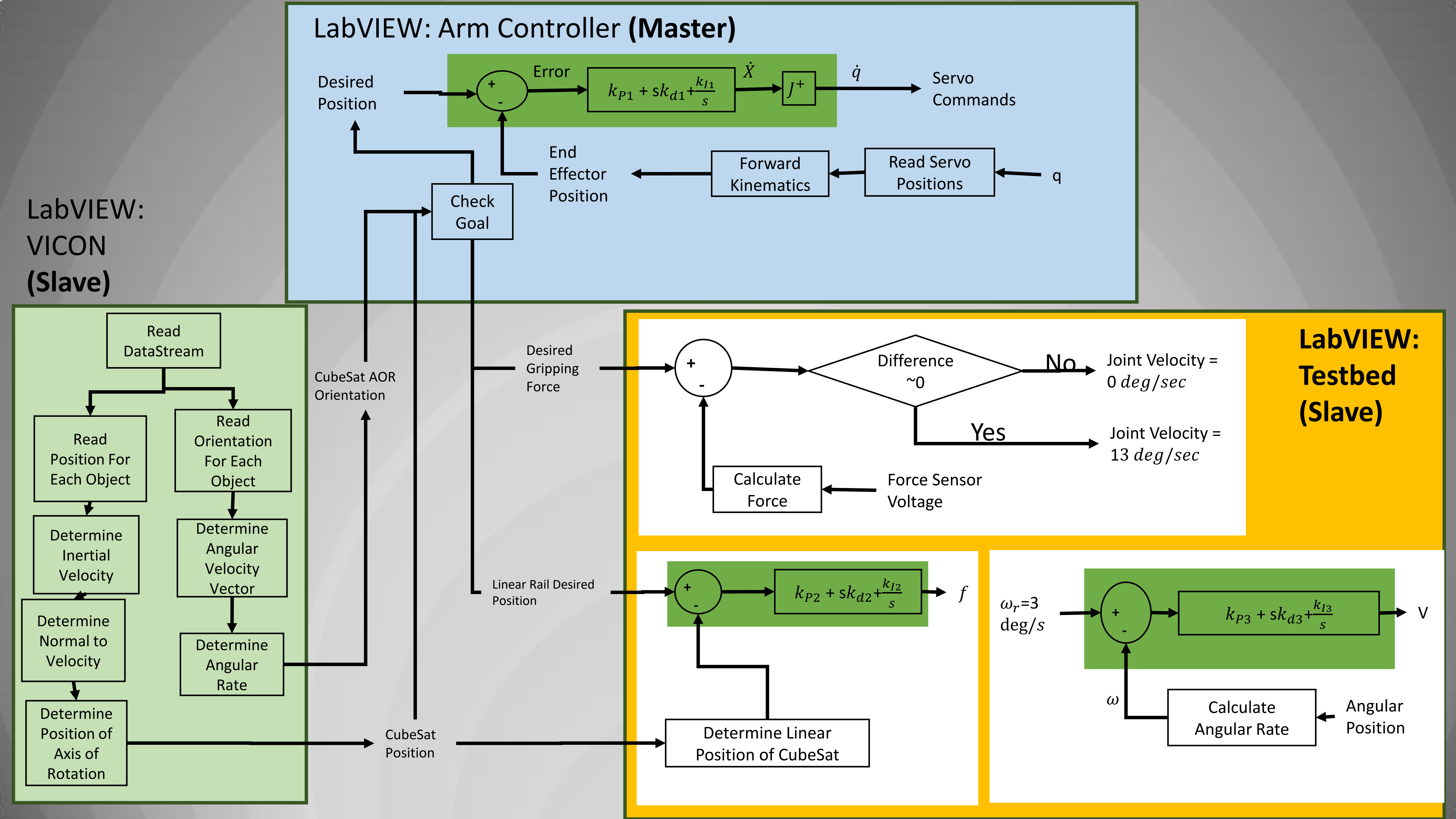
LabView: Arm+ End Effector
- Arm Controller
- Determine Desired Relative Positions for CubeSat and Arm

(Slave)

LabView: VICON
- Read Data from VICON
- Determine Axis of Rotation

(Slave)

LabView: Testbed Controller
- Constant CubeSat Rotation
- Linear Translation


LabVIEW: Arm Controller (Master)
Desired Position
Error
$k_{P1} + sk_{d1} + \frac{k_{I1}}{s}$
$\dot{X}$
J^+
$\dot{q}$
Servo Commands
End Effector Position
Forward Kinematics
Read Servo Positions
q
Check Goal
LabVIEW: VICON (Slave)
Read DataStream
Read Position For Each Object
Read Orientation For Each Object
Determine Inertial Velocity
Determine Angular Velocity Vector
Determine Normal to Velocity
Determine Angular Rate
Determine Position of Axis of Rotation
CubeSat AOR Orientation
CubeSat Position
Desired Gripping Force
Linear Rail Desired Position
LabVIEW: Testbed (Slave)
Difference ~0
No
Yes
Joint Velocity = 0 deg/sec
Joint Velocity = 13 deg/sec
Calculate Force
Force Sensor Voltage
$k_{P2} + sk_{d2} + \frac{k_{I2}}{s}$
f
Determine Linear Position of CubeSat
ω_r=3 $\deg/s$
$k_{P3} + sk_{d3} + \frac{k_{I3}}{s}$
V
ω
Calculate Angular Rate
Angular Position

Critical Project Elements

Critical Project Elements

Critical Project Element	Component
The CRST shall determine the relative position and attitude of the CubeSat and capture device during the demonstration. (FR 1.2)	Vision and Software
The CRST shall command the motion of the capture device during the demonstration (1.3)	Software and Electrical
The capture device shall be able to confirm the capture of the CubeSat after capture without human intervention. (DR 1.5.2)	Software, Mechanical, and Electrical

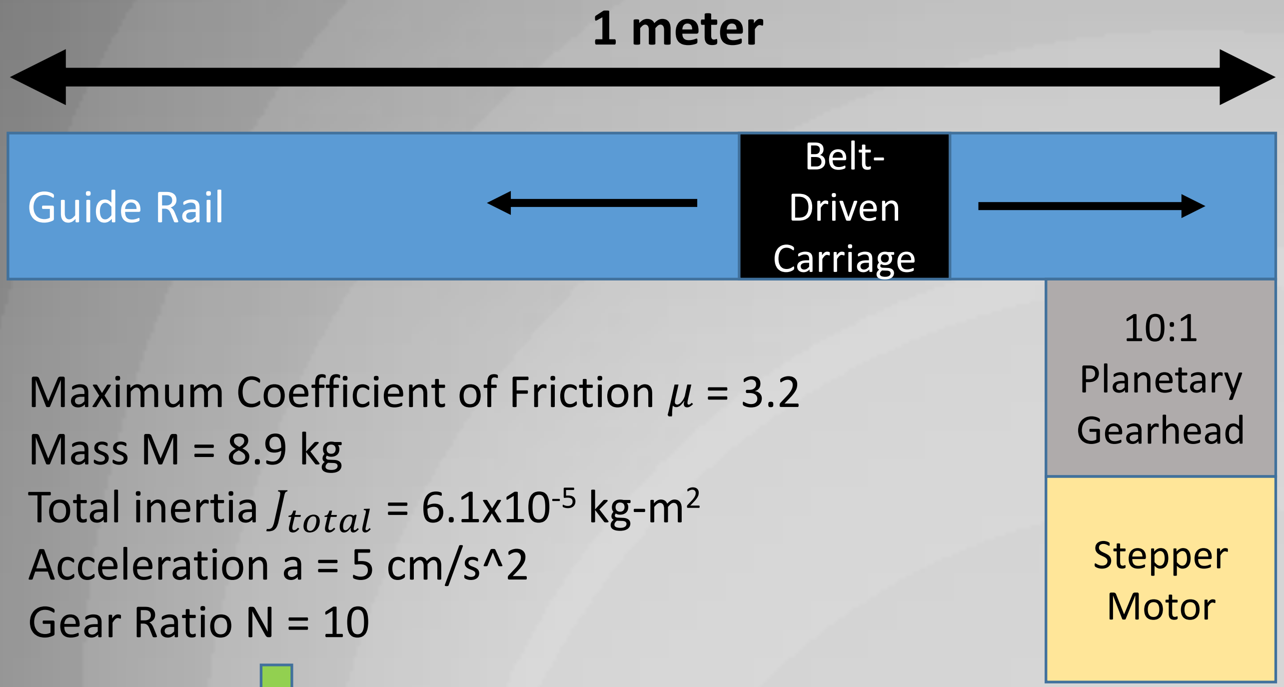
Design Requirements and Satisfaction

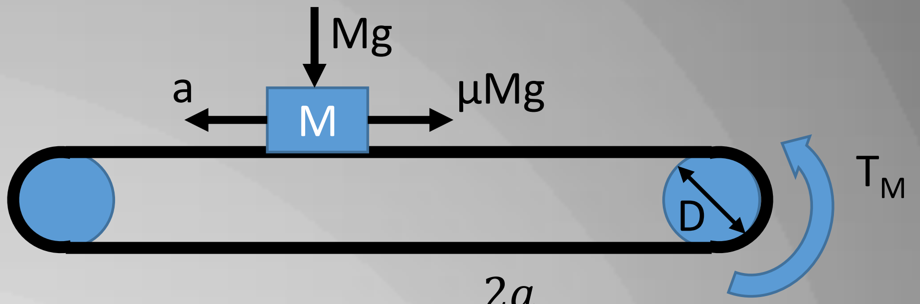
Functional Requirements

Functional Requirements	
1.0	**The CubeSat Recovery System Testbed (CRST) shall demonstrate the successful capture of a physical CubeSat model.**
1.1	The CRST shall demonstrate the motion of a CubeSat analogue during the demonstration.
1.2	The CRST shall determine the relative position and attitude of the CubeSat and capture device during the demonstration.
1.3	The CRST shall command the motion of the capture device during the demonstration
1.4	The CRST shall execute capture of the physical CubeSat model autonomously during the demonstration.

FR 1: **The CRST shall demonstrate the motion of a CubeSat analogue during the demonstration.**

CubeSat Translation





$$T_m = \frac{\mu M g + J_{tot}\frac{2a}{D}}{\eta_{belt}\eta_{gear}N}$$

$$J_{total} = \frac{J_{belt\ drive} + J_{load}}{N^2} + J_{gear} + J_{motor}$$

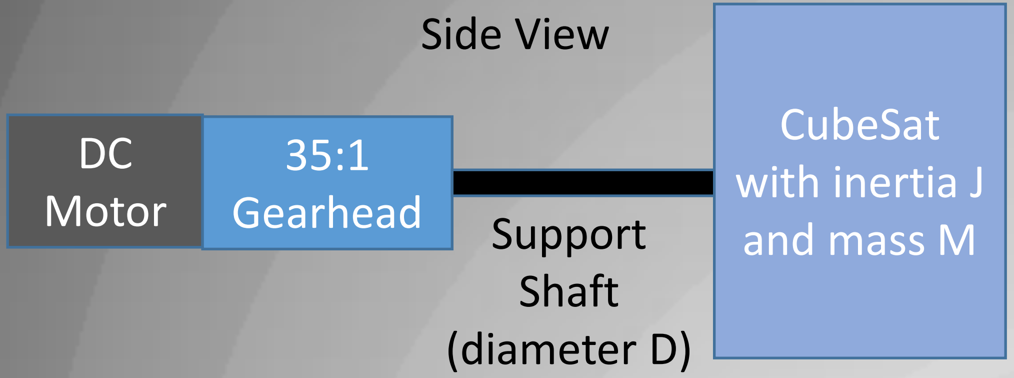
- Maximum Coefficient of Friction μ = 3.2
- Mass M = 8.9 kg
- Total inertia J_{total} = 6.1x10^{-5} kg-m^2
- Acceleration a = 5 cm/s^2
- Gear Ratio N = 10

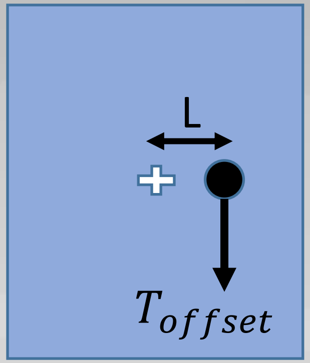
- **Required motor torque = 0.51 Nm**
- **Allowable motor torque = 0.78 Nm**

DR 1.1.2 – CubeSat shall allow for translational motion on one axis

CubeSat Rotation

Side View





Front View

$$T_{required} = T_{friction} + T_{offset} + T_{accel}$$

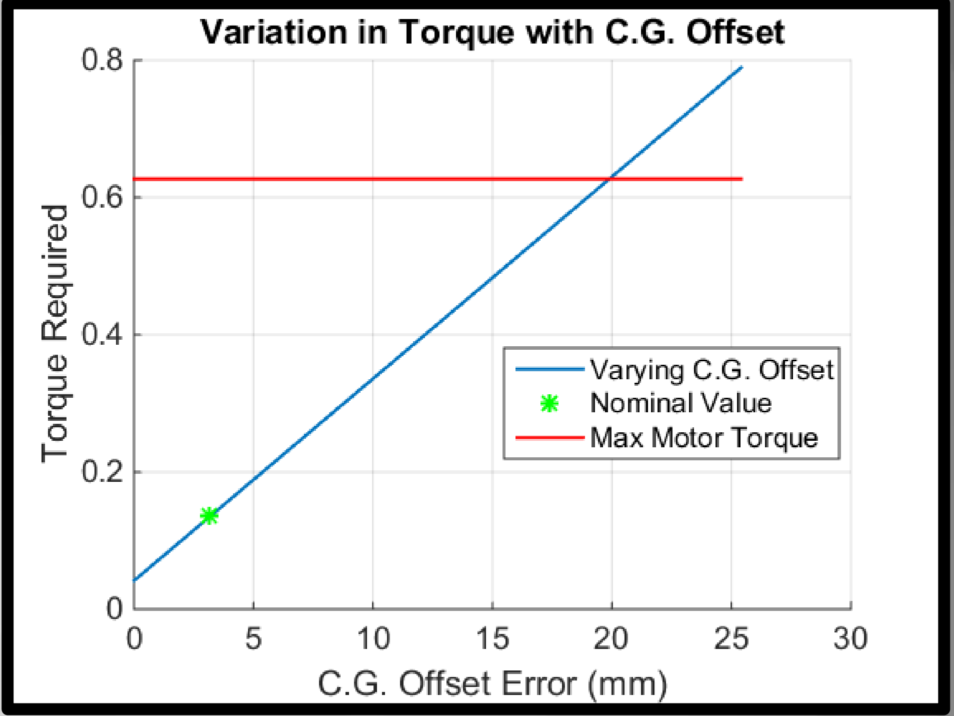
$$T_{required} = \frac{1}{2}\mu M g D + M g L + J\alpha$$

Nominal Values

- Mass M = 3 kg
- Shaft Diameter D = 12.7 mm (0.5 in)
- Coefficient of friction μ = 0.2
- Angular acceleration α = 3 deg/s^2
- C.G. offset L = 3.2 mm (1/8 in)

Predicted Torque Required = 0.14 Nm

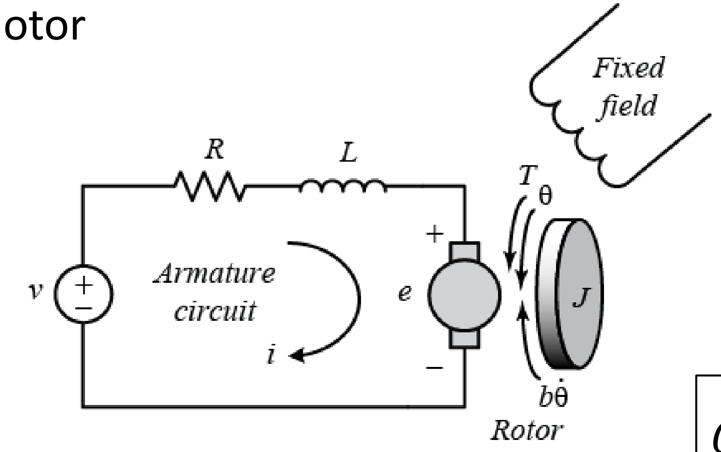
Motor Torque Available = 0.627 Nm

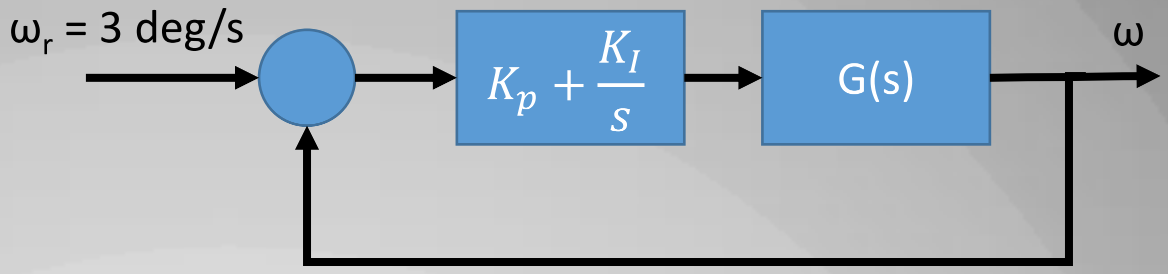


DDR 1.1.3.1-
CubeSat analog shall be capable of a minimum torque of 0.14 Nm to rotate CubeSat

CubeSat Rotation

DC Motor



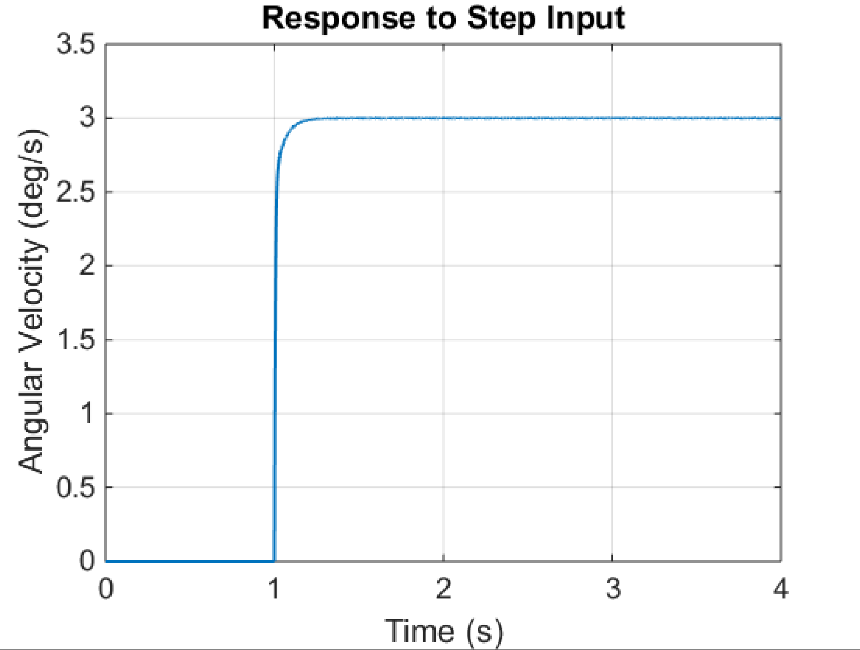


$$G(s) = \frac{\Omega(s)}{V(s)} = \frac{1}{\frac{LJ}{k_t}s^2 + \frac{RJ}{k_t}s + k_b N}$$

$$J = J_{motor} + J_{gear} + \frac{J_{shaft} + J_{cube}}{N^2}$$

- L = Motor Inductance (H)
- R = Motor Resistance (ohms)
- k_t = Torque Constant (Nm/A)
- k_b = Voltage Constant (V/rpm)
- Ω = Angular Velocity (rad/s)
- V = Input voltage (V)
- J = Load Inertia (kg-m^2)
- N = Gear Ratio

DR 1.1.3: CubeSat analogue shall allow for rotation of 3 deg/s about its major axis

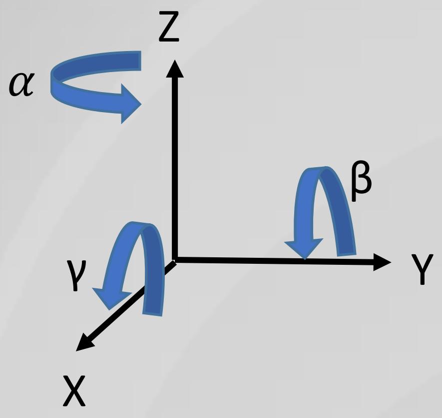


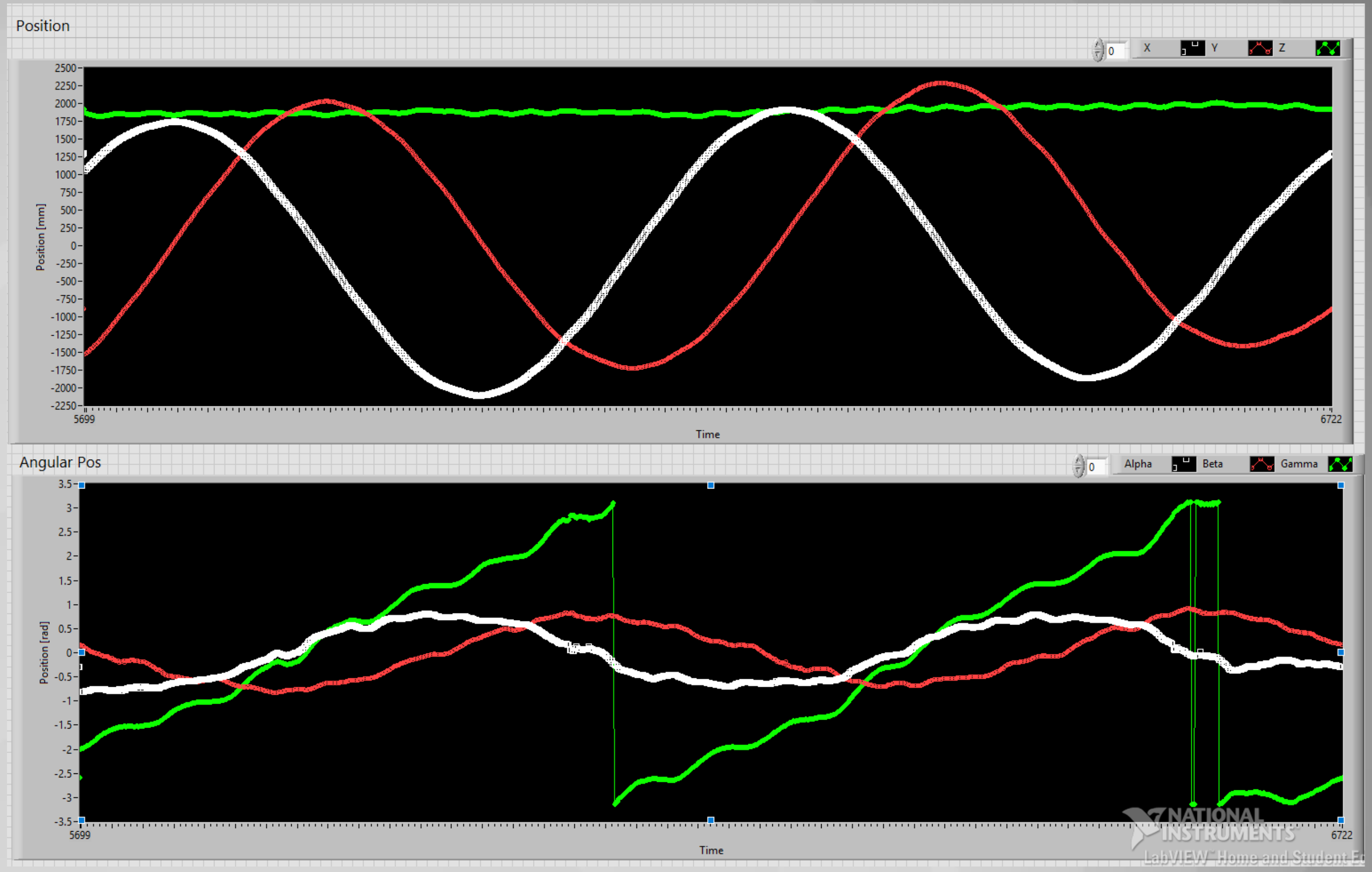
FR2: **The CRST shall determine the relative position and attitude of the CubeSat and capture device during the demonstration.**

VICON Interface

- Utilizes DataStream Software Development Kit
- Uses the dotNET framework directly into LabVIEW as functions
- Functions to read Position, Orientation for each object
- Ethernet TCP Protocol.

FR 1.2.2
The CRST shall communicate with the Vicon Motion Capture System to sense the initial conditions and motion throughout the demonstration.





LabVIEW Timing Methodology

VICON Loop

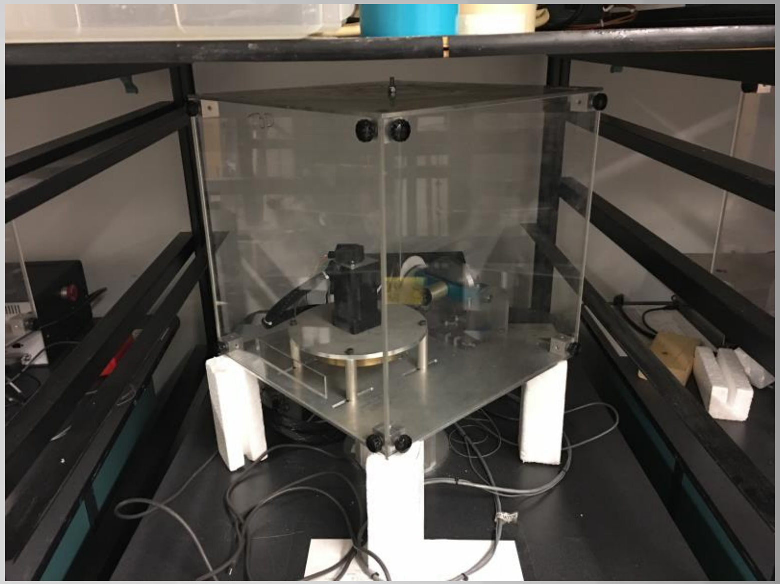
- Utilized LabVIEW Script for VICON Data
- Called MATLAB function that contains same computations for AOR

Control Loops

- Utilized ASEN 3200 Spin Modules to Estimate Loop Timing
- Similar to Expected code in terms of Overhead due to PID Control Loop

Communication Between Loops

- 35.7 u--Neglectable
- Control Loop takes 6ms to run
- Control Loop is 16,706% Faster



Code Timing

VICON Loop
- Utilized LabVIEW Script for VICON Data
- Called MATLAB function that contains same computations for AOR

Control Loops
- Utilized ASEN 3200 Spin Modules to Estimate Loop Timing

Communication Between Loops
- 35.7 us--Neglectable

DR 1.2.2:
The CRST shall send commands at a rate of 10.5 Hz Minimum

Software	Requirement [Hz]	Estimated Performance [Hz]	Margin [Hz]
VICON Loop	10.5	100 .0	89.5
Arm Control Loop	10.5	144.4	133.9
Test Bed Control Loop	13.4	166.6	152.2

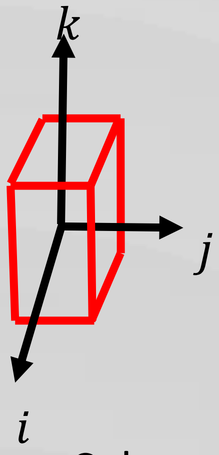
Axis of Rotation

Assumptions:

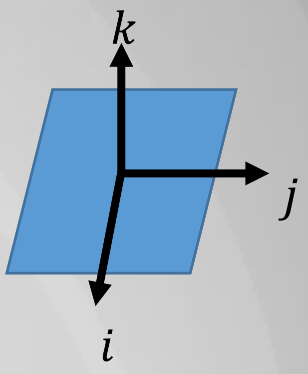
- Grabbing CubeSat While Not Translating On Rail.
- Rigid Body Dynamics.
- "Torque Free Motion"

VICON Output:

- Inertial Position
- Euler Angles

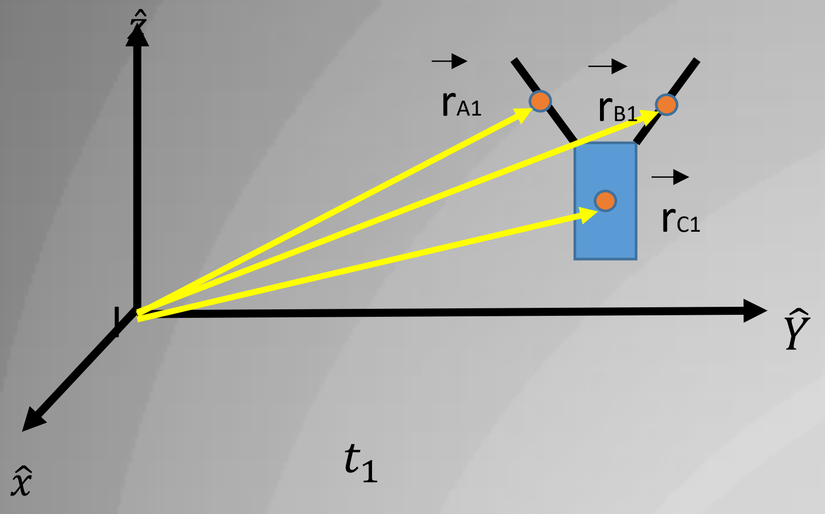


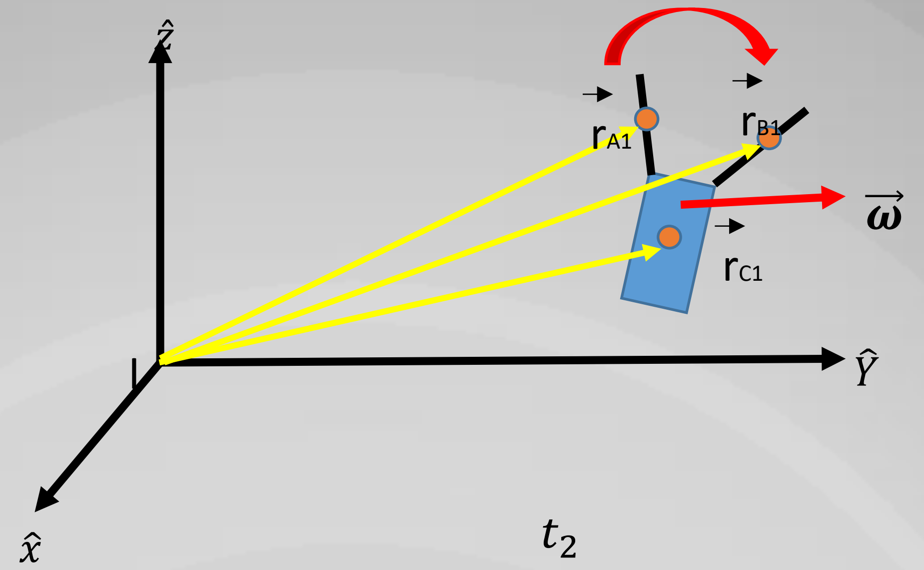
Cube



Solar Panel

Axis of Rotation





Determine Euler Rates:

$$\dot{\vec{\Theta}} = \frac{\vec{\Theta}_{t2} - \vec{\Theta}_{t1}}{\Delta t}$$

Transform Euler Rates to Angular Velocities:

$$[\vec{\omega}]_B = [T]\dot{\vec{\Theta}}$$

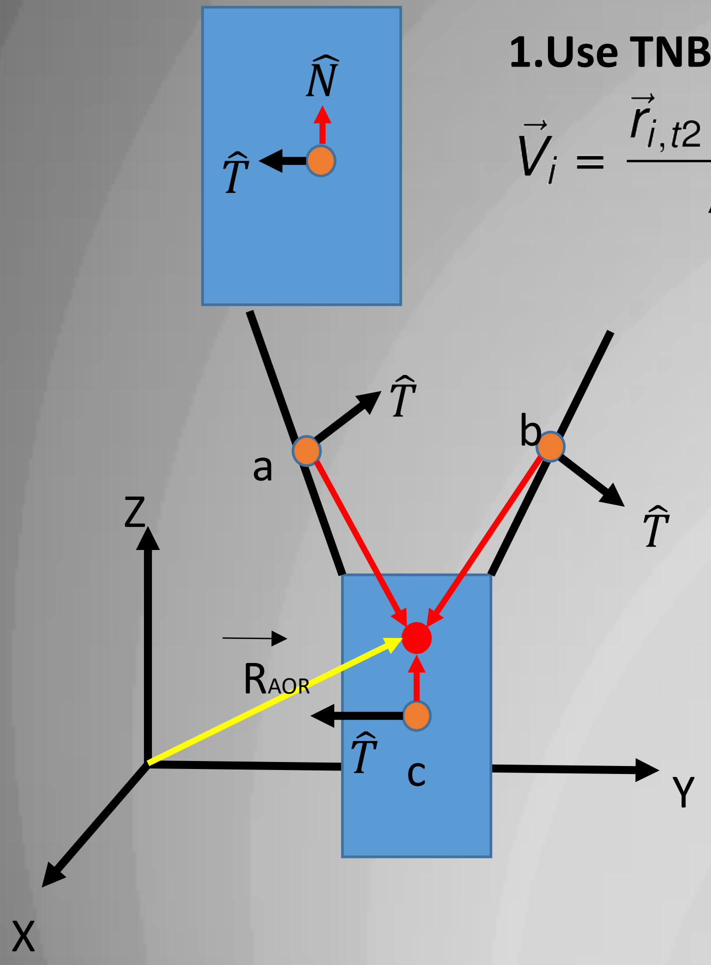
$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Transform Body Coordinates to Inertial Coordinates:

$$[\vec{\omega}]_I = [Q][\vec{\omega}]_B$$

$\vec{\Theta}$ = Euler Angles
$\vec{\omega}$ = Angular Velocity

Axis of Rotation



1.Use TNB frames

$$\vec{V}_i = \frac{\vec{r}_{i,t2} - \vec{r}_{i,t1}}{\Delta t} \qquad \hat{N}_i = \frac{\vec{V}_{i,t3} - \vec{V}_{i,t2}}{|\vec{V}_{i,t3} - \vec{V}_{i,t2}|}$$

2.Determine AOR Using Instantaneous Centers

$\vec{r}_{AOR}$ = Intertial Position of AOR

$$\vec{r}_{AOR} = \vec{r}_a + X_a\hat{N}_a$$
$$\vec{r}_{AOR} = \vec{r}_b + X_b\hat{N}_b$$
$$\vec{r}_{AOR} = \vec{r}_c + X_c\hat{N}_c$$

3.Solve Above Equations for X

X = Constant for $\hat{N}$ to intercept at AOR

$$\vec{r}_c - \vec{r}_b = X_b\hat{N}_b - X_c\hat{N}_c$$
$$\vec{r}_c - \vec{r}_a = X_a\hat{N}_a - X_c\hat{N}_c$$
$$\vec{r}_b - \vec{r}_a = X_a\hat{N}_a - X_b\hat{N}_b$$

9 Equations, 3 Unknowns

4.Substitute Back to Find Position of AOR

1.2.3
The CRST shall Calculate the axis of rotation of the CubeSat model during the demonstration.

CubeSat Position

$$[\vec{r}_{AOR/c}]_B = [Q][\vec{r}_{AOR/c}]_I$$

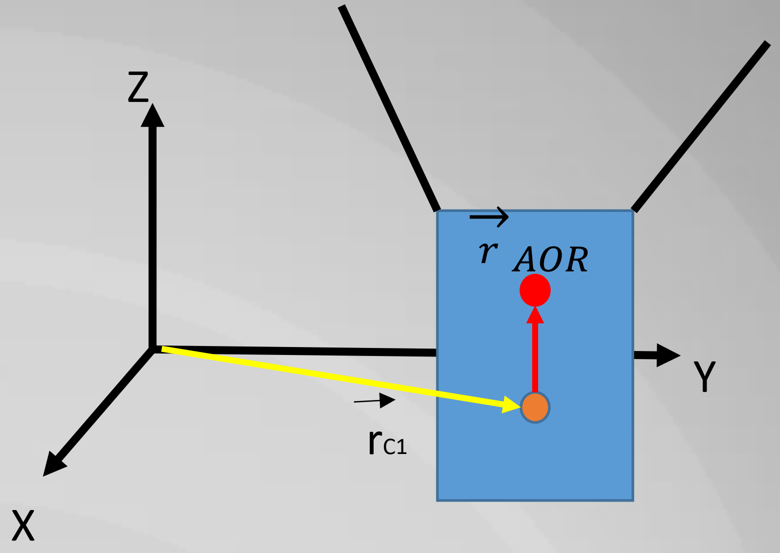
Constant for Rigid Body

$$[\vec{r}_c]_B = [Q][\vec{r}_c]_I$$

Varies with Time

$$[\vec{r}_{AOR}]_B = [\vec{r}_c]_B + [\vec{r}_{AOR/C}]_B$$

$$[\vec{r}_{AOR}]_I = [Q^{-1}][\vec{r}_{AOR}]_B$$



Thus, Inertial Translational Position is the magnitude of the x and y component of $[r_{AOR}]_I$

DR 1.2.3:
The CRST shall determine the relative linear position of the CubeSat model during the demonstration.

Desired Position of End Effector

Goal 1:

- Align End Effector With AOR while Keeping Arm out of Harms Way of CubeSat Translation
- Desired Position is L_{min}
- L_{min} is calculated Numerically From Arm Geometry and Initial Conditions

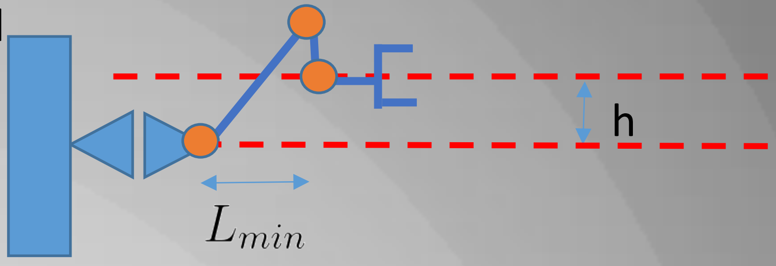
Goal 2:

- Translate CubeSat to 75% of Extension Length (ϵ)

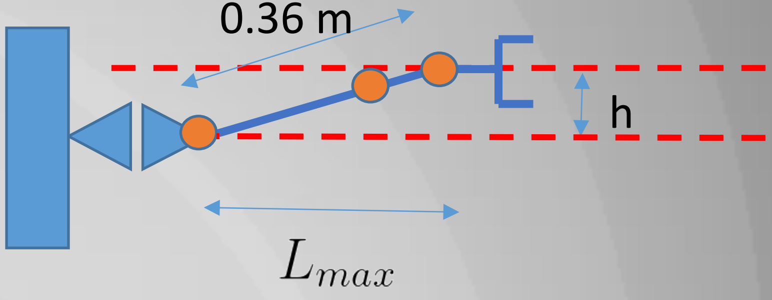
Goal 4

- 75% of Extension Length (ϵ) was chosen to be Desired End Effector Position.
- Prevents Full Extension to Help Protect the Arm.
- Allows for Extension Distance to Protect the Arm from Translation of CubeSat.

Retracted



Fully Extended



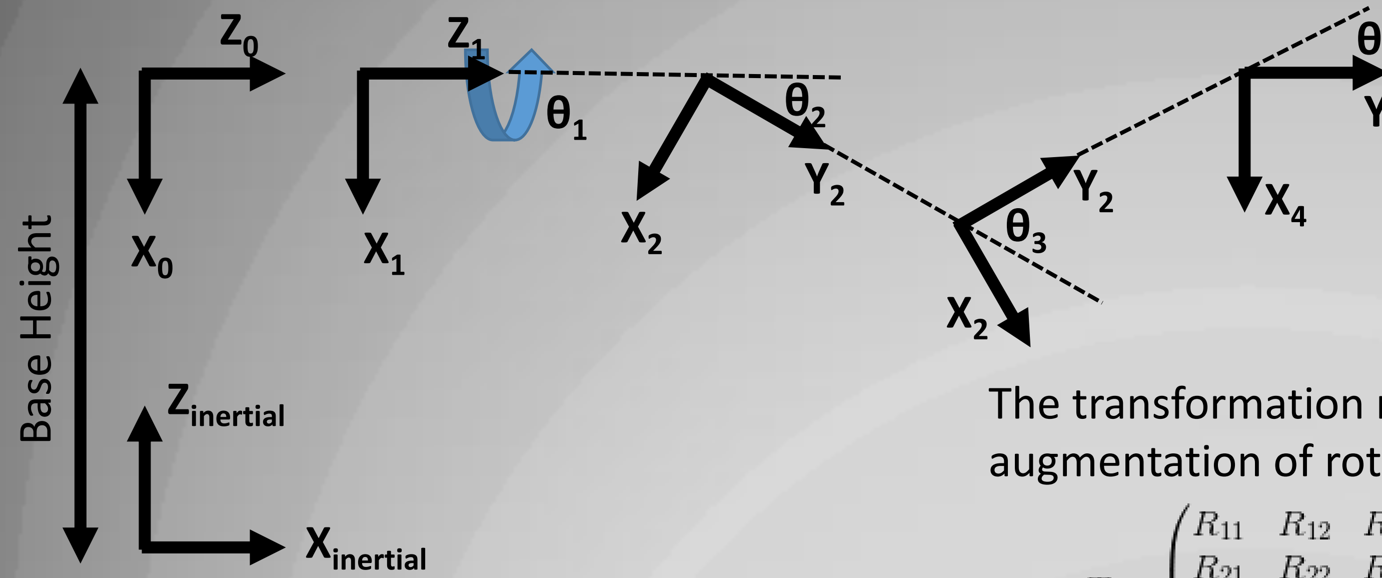
$$L_{max} = \sqrt{0.36^2 - h^2}$$

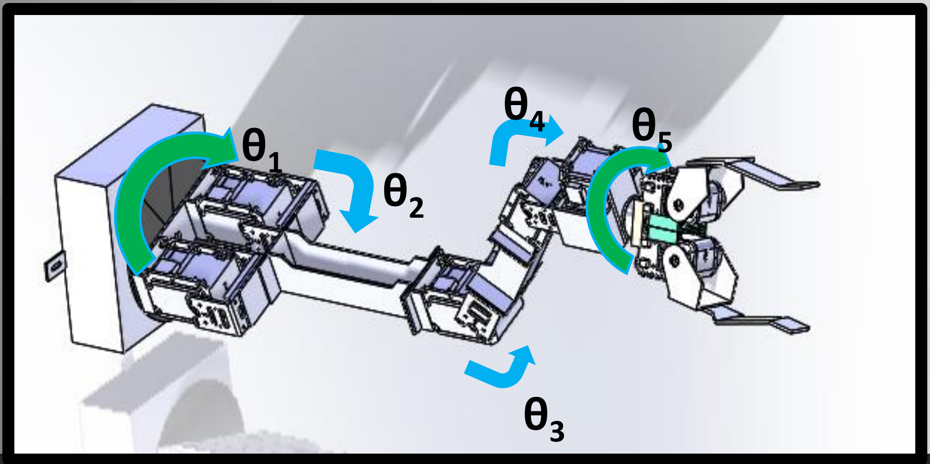
$$\epsilon = L_{max} - L_{min}$$

DR 1.2.4
The CRST Shall Determine the Desired End Effector Orientation and Postion

FR 3: **The CRST shall command the motion of the capture device during the demonstration**

Current Position: Forward Kinematics





The transformation matrix from one joint to the next is the augmentation of rotation matrix R and translation vector (Δx, Δy, Δz)

$$T = \begin{pmatrix} R_{11} & R_{12} & R_{13} & \Delta x \\ R_{21} & R_{22} & R_{23} & \Delta y \\ R_{31} & R_{32} & R_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The transformation from inertial coordinates to end effector coordinates is then:

$$T_{end}^{inertial} = T_0^{inertial} T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_{end}^5$$

This can be used to calculate x, y, z coordinates and roll, pitch, yaw of end effector

DR 1.3.1: The CRST shall calculate the current end effector location and orientation during the demonstration.

Arm Control Theory

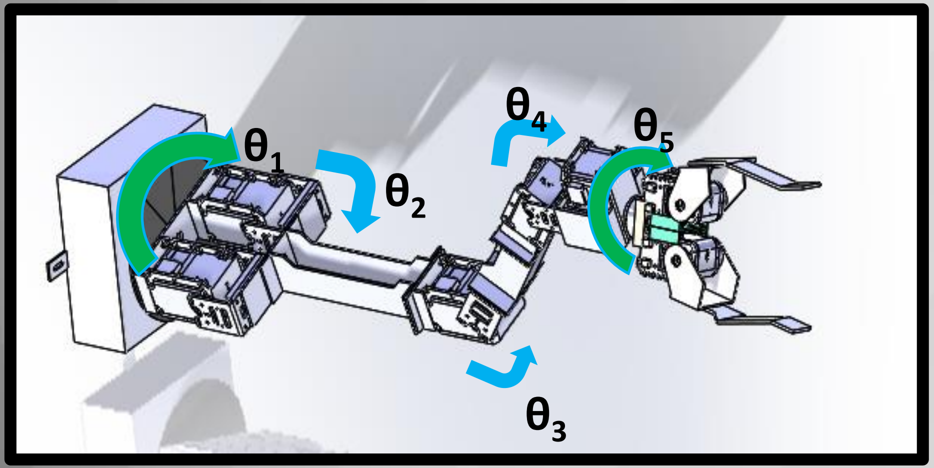
Feedback Control Via the Inverse Jacobian Method

$$\boxed{\dot{q} = J^{+}\dot{X}}$$

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_2 \\ \theta_4 \\ \theta_5 \end{bmatrix} \qquad X = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}$$

- q = Array of joint angles
- X = Position and orientation of end effector in inertial space
- J = Jacobian matrix
- J^{+} = Inverse Jacobian Approximation

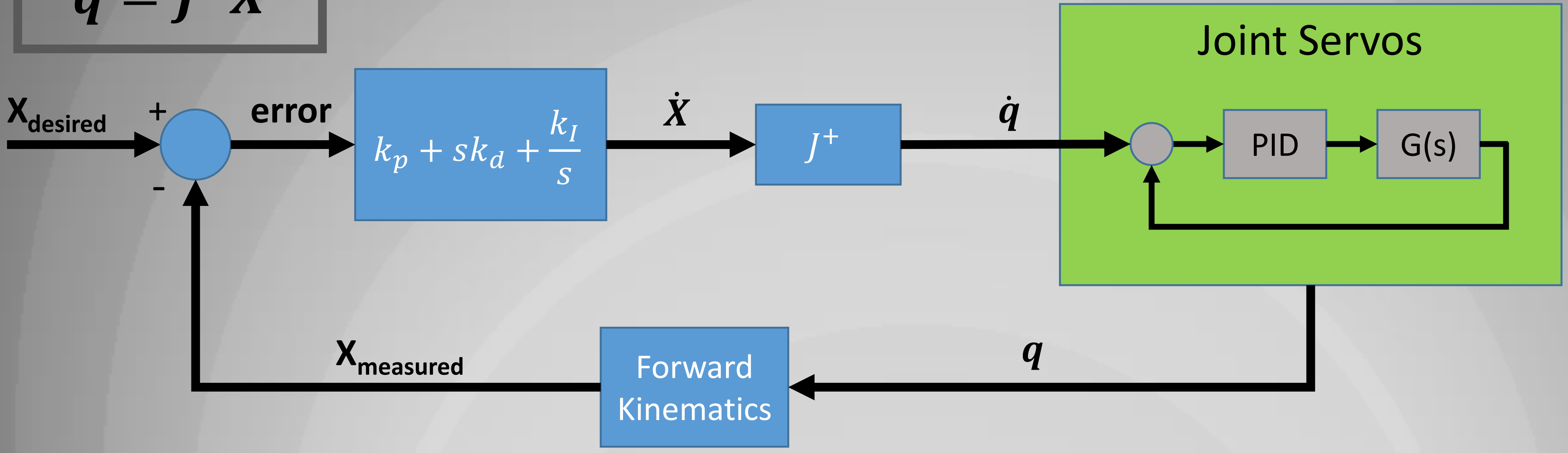
$$J_{6x5} = \begin{pmatrix} \frac{dx}{d\theta_1} & \cdots & \frac{dx}{d\theta_5} \\ \vdots & \ddots & \vdots \\ \frac{d\psi}{d\theta_1} & \cdots & \frac{d\psi}{d\theta_5} \end{pmatrix} \qquad J^{+} = J^{T}(JJ^{T})^{-1}$$



This transformation allows us to control the position and orientation of the end effector in inertial space by setting the joint angular velocities

Arm Control Theory

$$\dot{q} = J^{+}\dot{X}$$



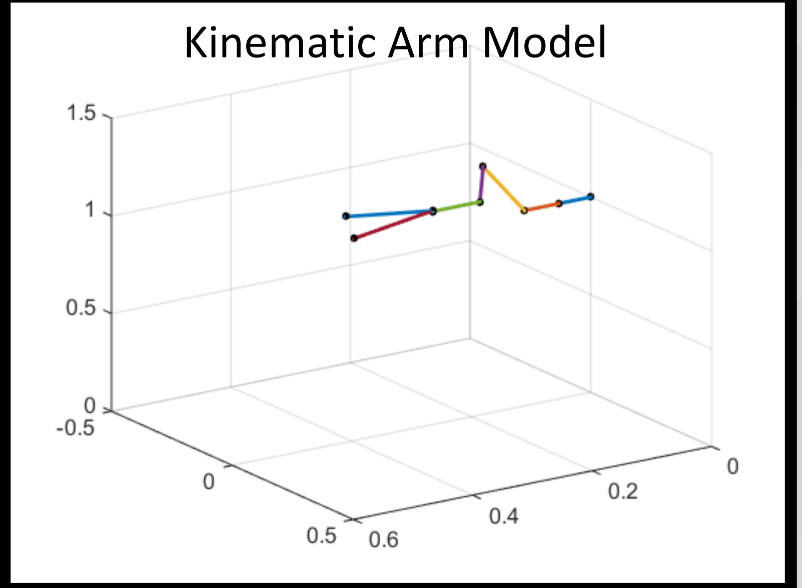
Internal to Dynamixel Servos

Performed by CASCADE Software

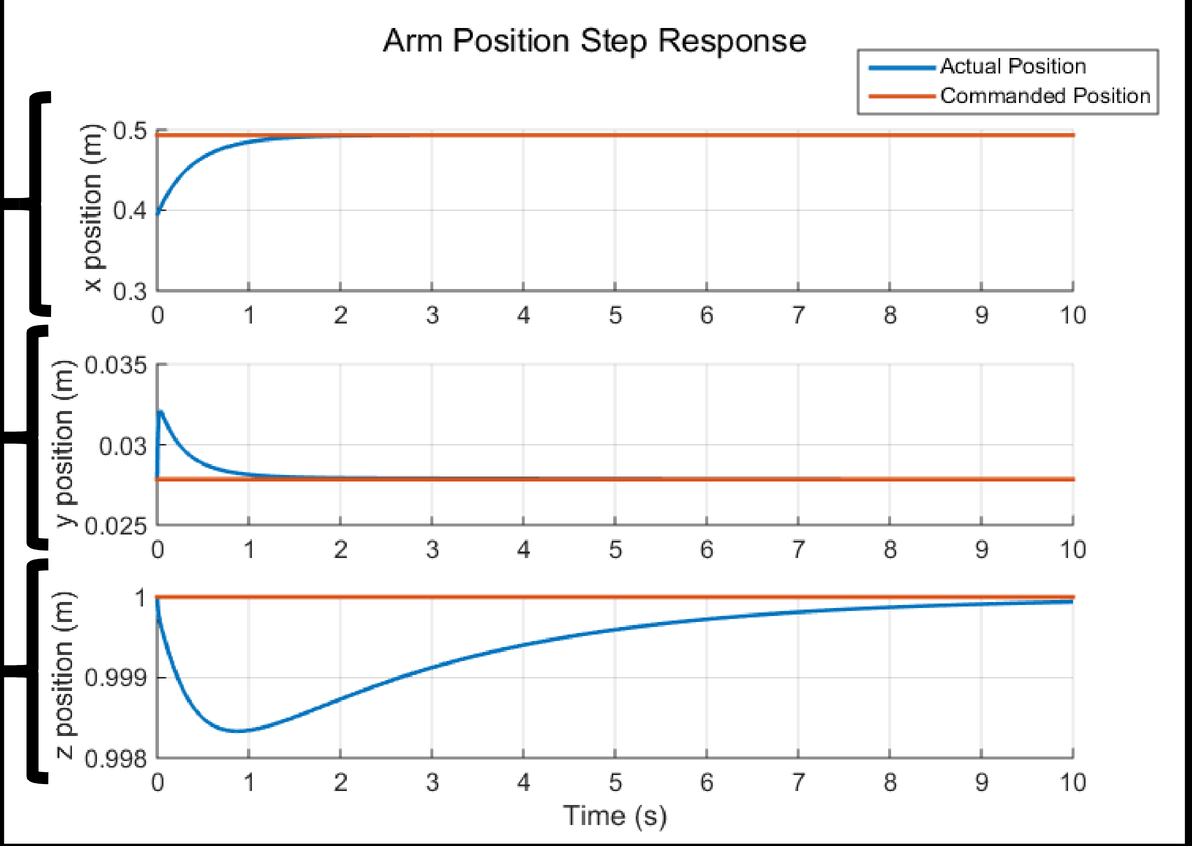
- q = Array of joint angles
- X = Position and orientation of end effector in inertial space
- J^{+} = Inverse Jacobian Approximation

Arm Control

- **Initial End Effector Position:**
  X_0 = (0.40, 0.03, 1.00) m
- **Commanded End Effector Position:**
  X_{ref} = (0.50, 0.03, 1.00) m



Note Differing Axis Limits



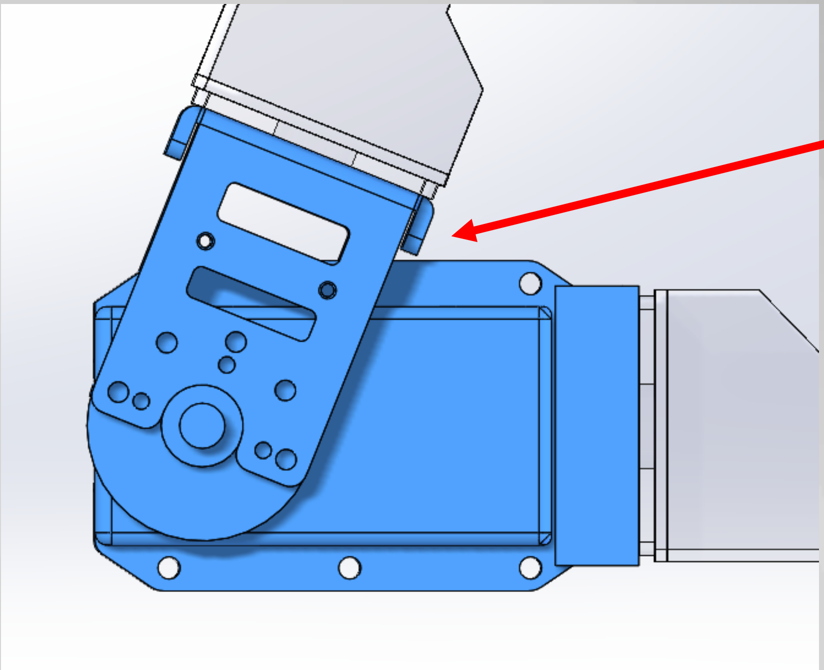
FR 1.3: The CRST shall command the motion of the capture device during the demonstration.

Arm Limits

Physical Angle Constraints

Servo	Physical Angle Limits
Turn Table	-180 < Θ < 180
Base Servo	0 < Θ < 180
Elbow Servo	-110 < Θ < 110
Wrist Servo	-110 < Θ < 110
Wrist Rotator Servo	-180 < Θ < 180

Elbow Physically Constrained

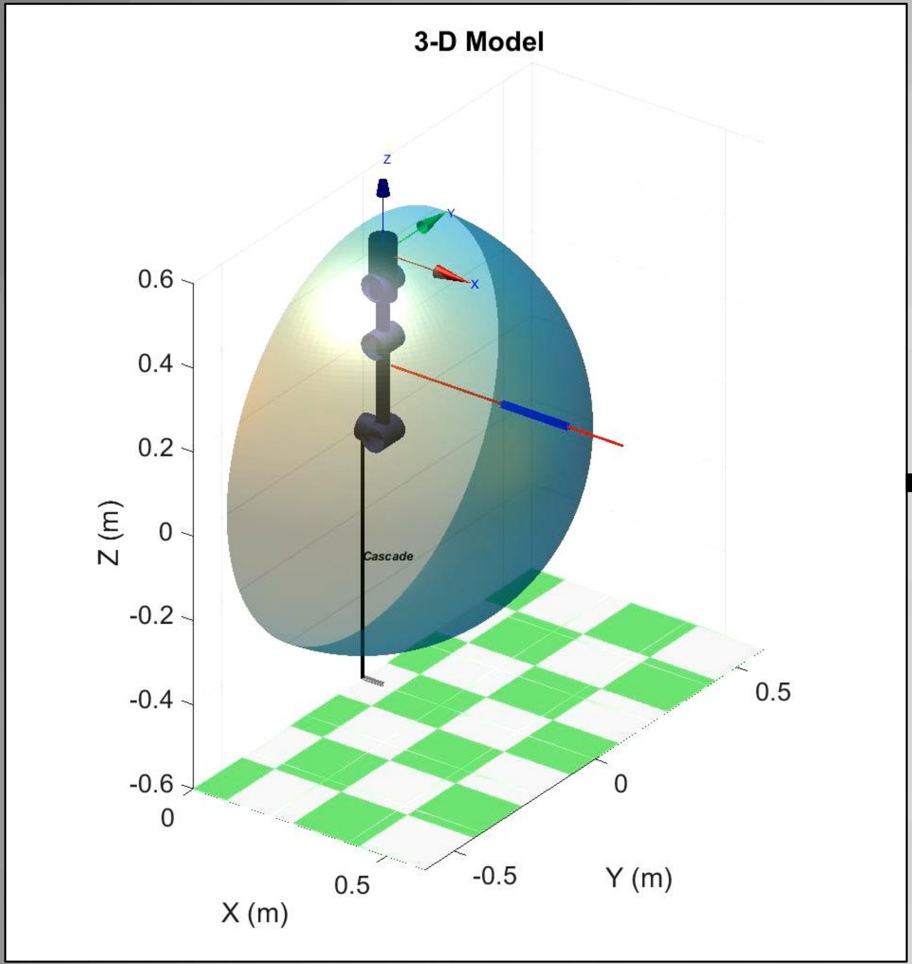


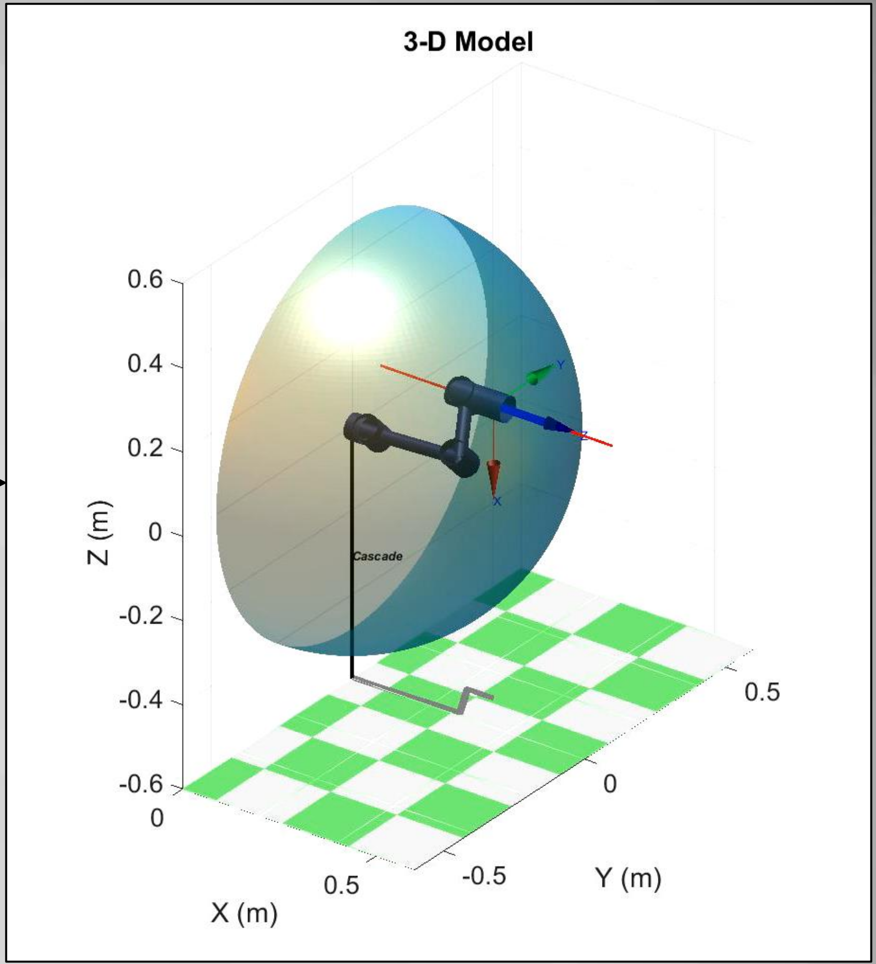
Servo Close to Self Collision

DR 1.3.3: The commands sent for capture device motion shall be within joint servos performance limits.

Arm Limits

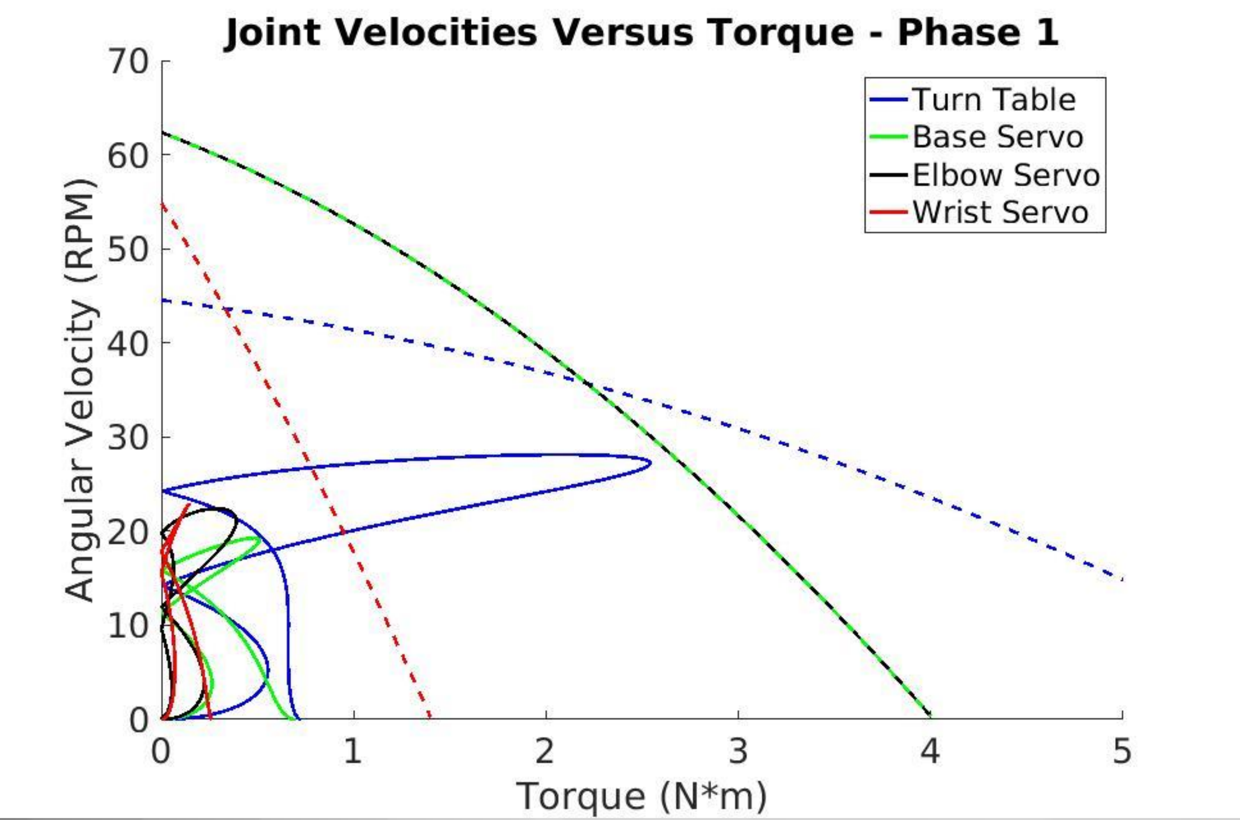
Simulation of Phases 1 & 4

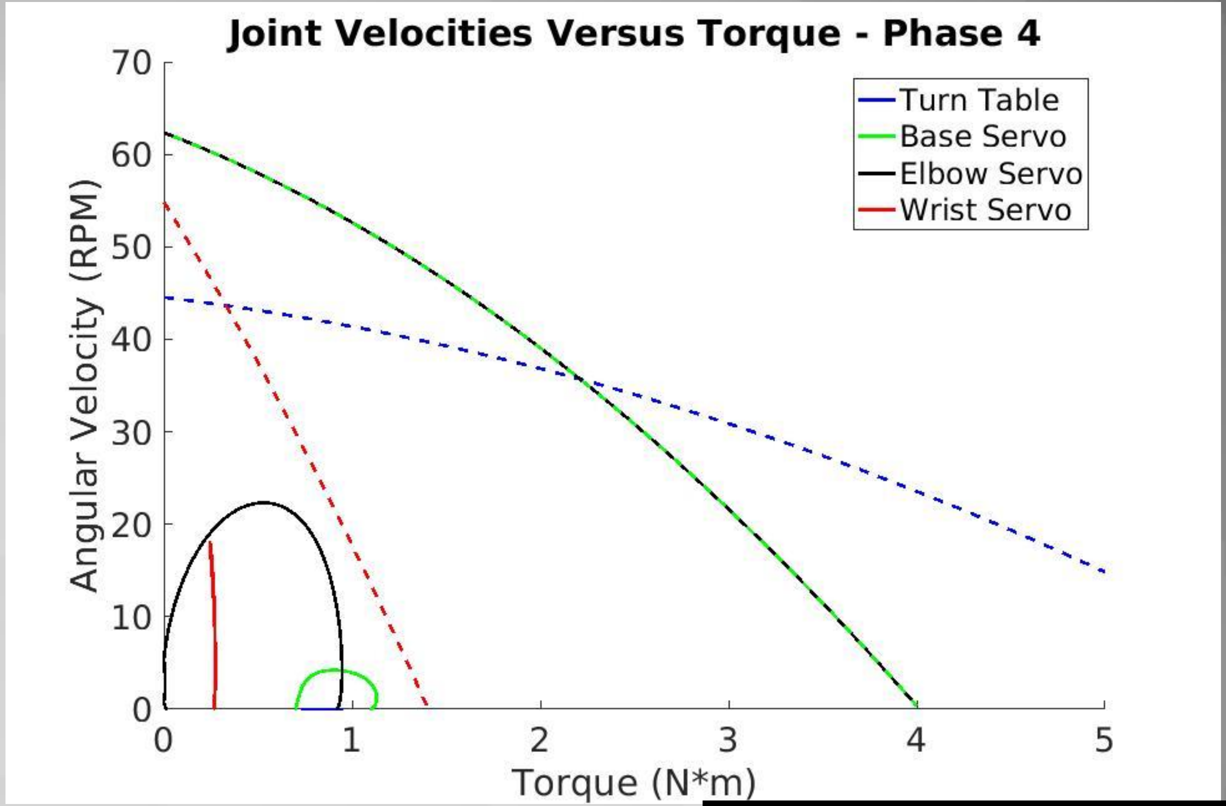




Arm Limits

Servo Performance





DR 1.3.3: The commands sent for capture device motion shall be within joint servos performance limits.

FR 4: The CRST shall execute capture of the physical CubeSat model autonomously during the demonstration.

Arm Limits

Peak & Avg Power Feasibility

Phase 1

	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	17.23 W	15.66 W	5.94 W	3.85 W	0	42.68 W
6 s	11.93 W	16.25 W	9.45 W	3.85 W	0	41.48 W

Phase 4

	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	8.29 W	24.76 W	11.23 W	3.92 W	0	48.2 W
6 s	8.29 W	23.92 W	10.92 W	3.85 W	0	46.98 W

DR 1.4.1: The capture device shall have an average power of no more than 100 W

DR 1.4.2: The capture device shall have a peak power of no more than 168 W

Arm Limits

Peak Current Draw Feasibility

Phase 1

	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	1.44 A	1.31 A	0.49 A	0.32 A	0	3.56 A
6 s	0.99 A	1.35 A	0.79 A	0.32 A	0	3.46 A

Phase 4

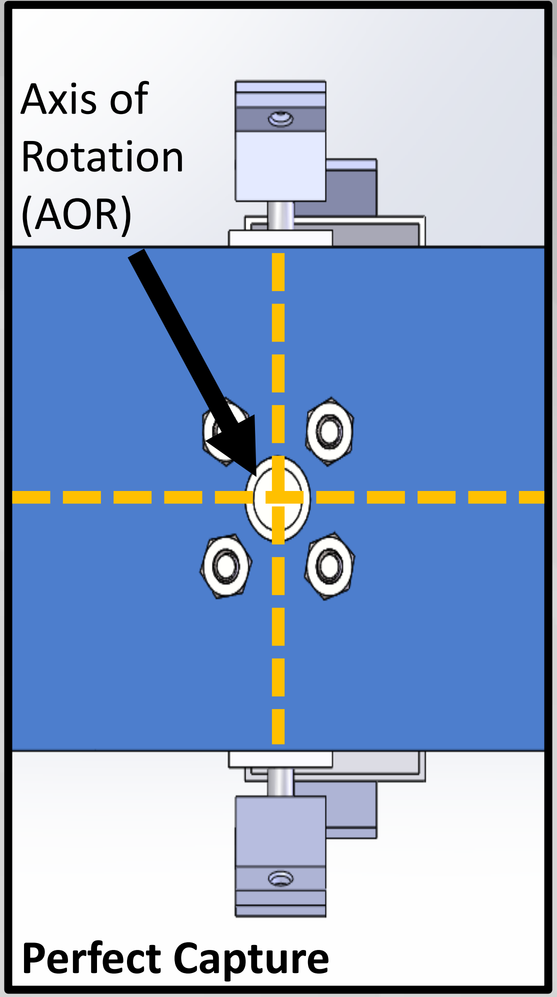
	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	0.69 A	2.06 A	0.94 A	0.33 A	0	4.02 A
6 s	0.69 A	1.99 A	0.91 A	0.32 A	0	3.92 A

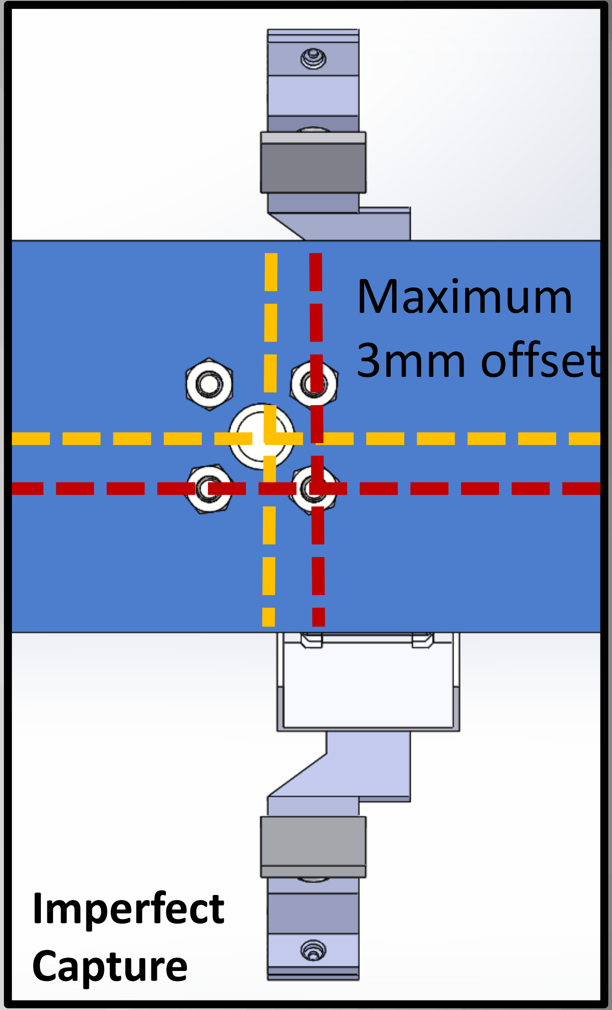
DR 1.4.3: The capture device shall have a peak current draw of no more than 10 A

Claw Compliance

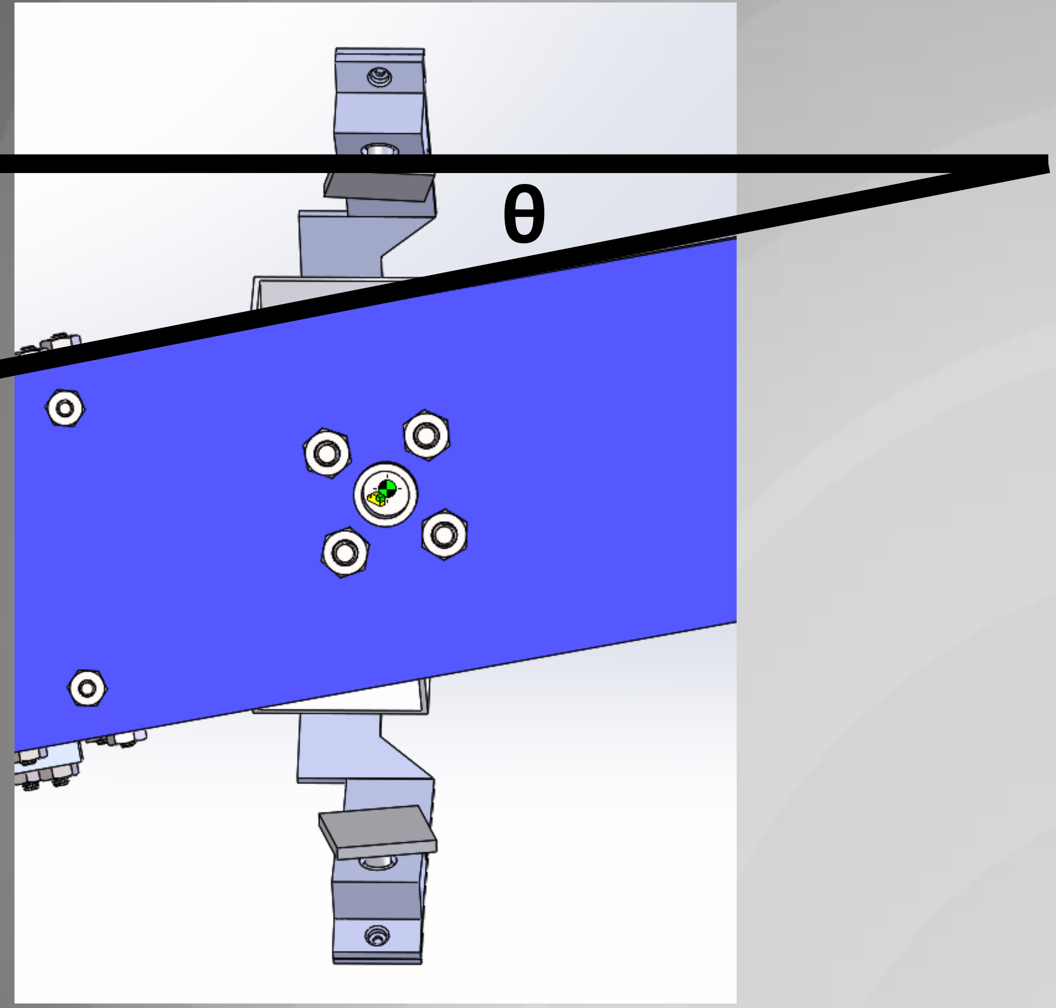
Perfect Capture Assumes:
- Claw perpendicular to CubeSat
- Gripper surface contacts on AOR and corresponding axes
- Claw perfectly matches CubeSat rotation phase
- Gripper plates contact surface simultaneously

Imperfect Capture Assumes:
- Gripper surface not at AOR
- Gripper plates do not contact simultaneously
- 3 mm position offset from servo resolutions





Claw Compliance



Phase Error:

- Ball joint compensates for equivalent 10° of offset
- Based on servo specs and control timing the max expected offset is 0.1°.
    - 0.08° from servo positioning tolerance
    - 0.018° from latency in sending commands
- Error in CubeSat rotation motor can be ignored since Vicon is used to measure orientation

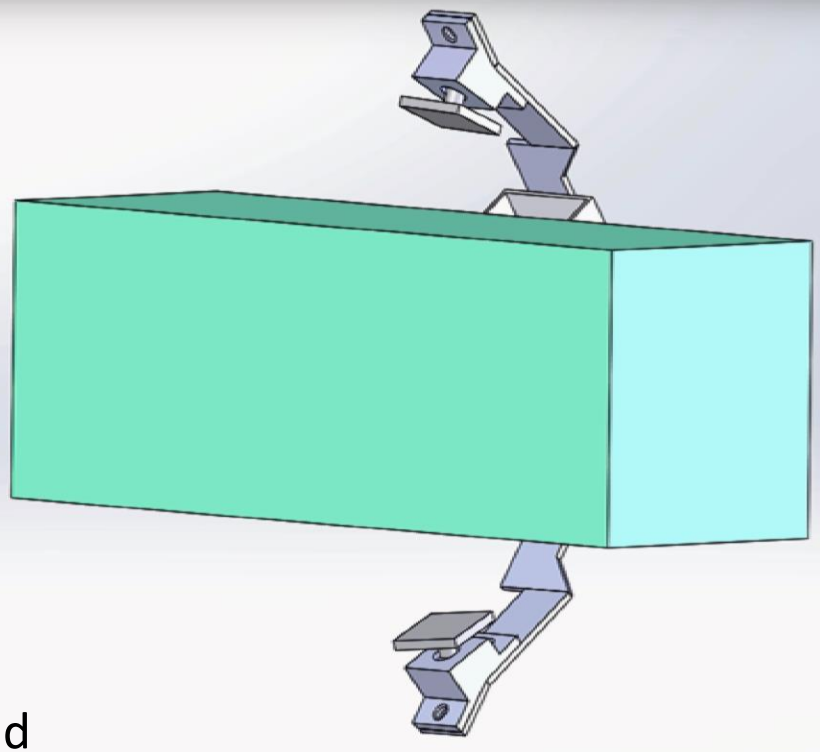
Claw Compliance

DR 1.4.7-The capture device shall be able to confirm the capture of the CubeSat without human intervention.

DR 1.4.6-The capture device shall be able to release from the CubeSat after capture without human intervention.

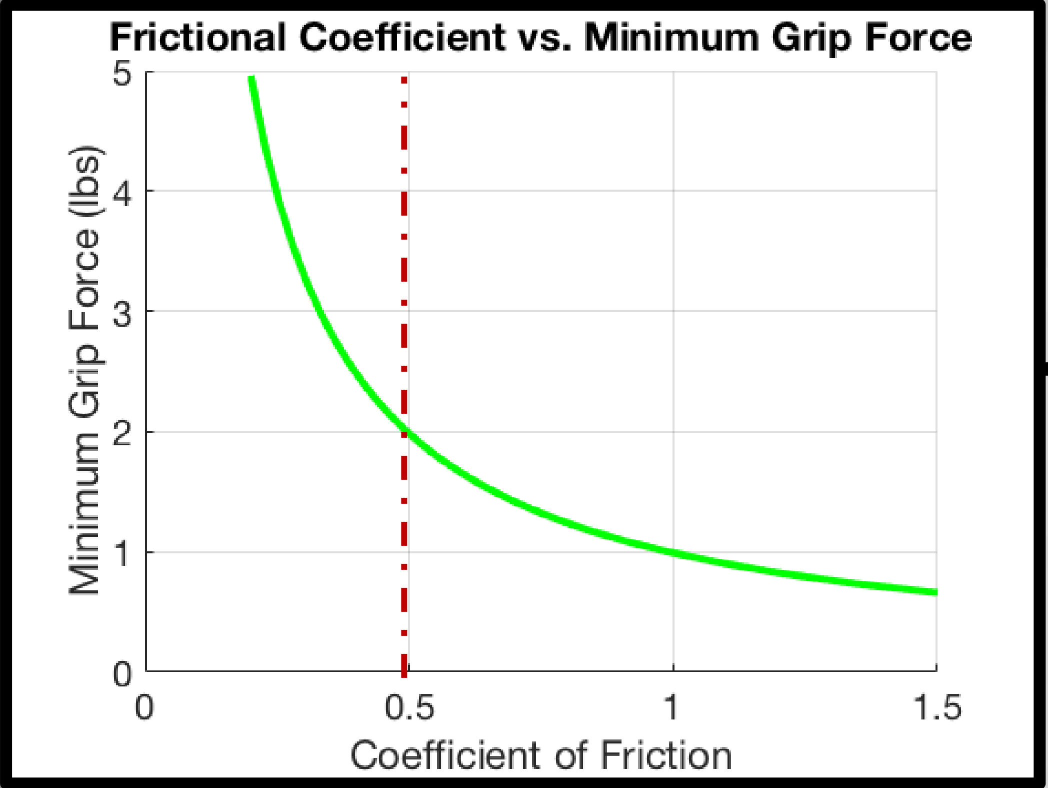
Imperfect Capture Mitigation:

- Ball joint compensate up to 10° of planar offset
- Load from force sensor reaches capture threshold
  - First contact servo shuts off
  - Second servo completes closure



Given the 3mm offset in y and z axis and 0.1° phase offset → **Gripper plates still make direct contact**

Claw Minimum Grip Force



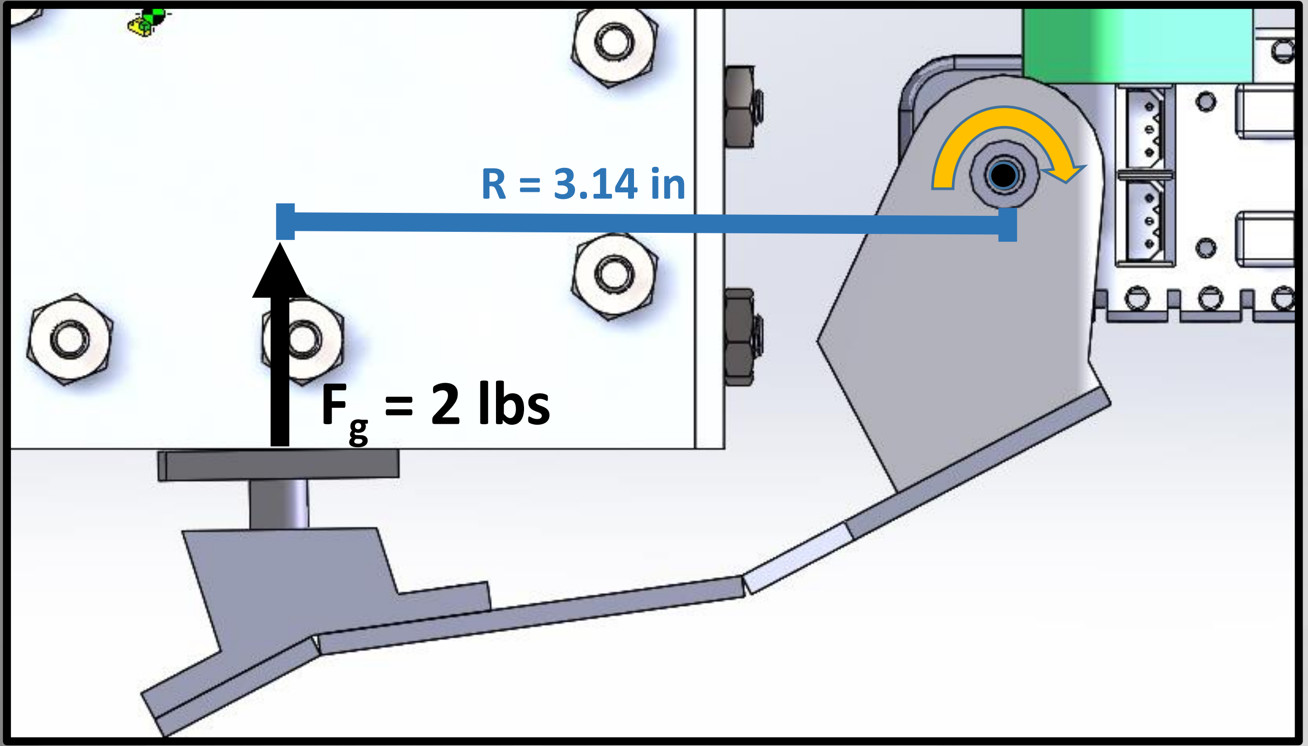
$$F_g = \frac{I\,\delta\omega}{2\mu d\,\delta t}$$

Assumptions:

- Coefficient of friction no less than 0.5.
- Force Sense Resistor (non-uniform) contact to Aluminum surface
- Capture Time (**δt**) is greater than 0.01 seconds
- d = distance from CG to outer surface of CubeSat = 1.87”
- $\delta\omega$ = CubeSat speed = 0.05 rad/s

F_g = 2 lbs (assuming min friction)

Claw Torque Required



- Determine the minimum gripping force to hold CubeSat (confirm capture)
- Calculate torque on independent gripper servo T = R x F
- Verify torque on servo does not exceed servo specifications

Required Torque	Servo Stall Torque	Margin
100.48 oz-in	226 oz-in	44.46%

DDR 1.4.5: The end effector of the capture device shall have a minimum grip strength of 4 oz.

Project Risks

Risk Introduction

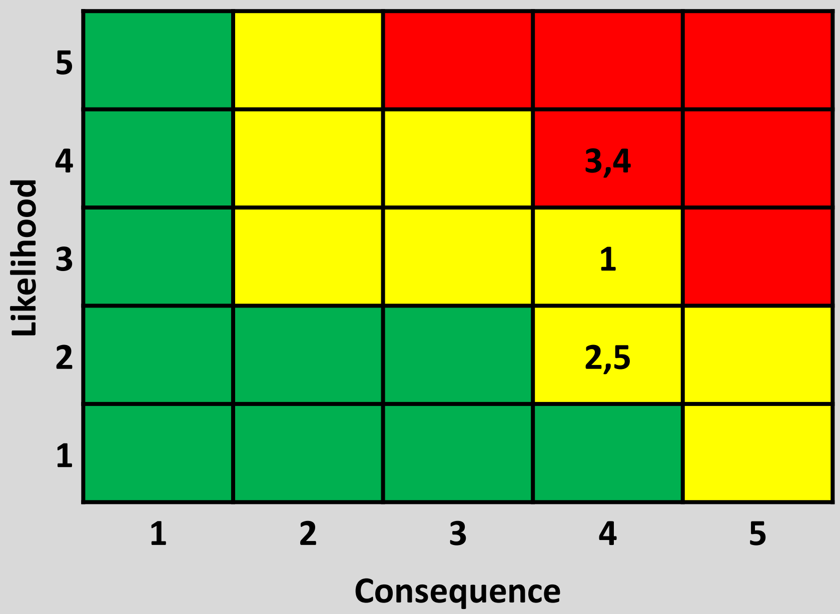
Context: Risk Analysis conducted for highest level of success – 1 DOF Rot./Trans., Closed-loop Autonomy

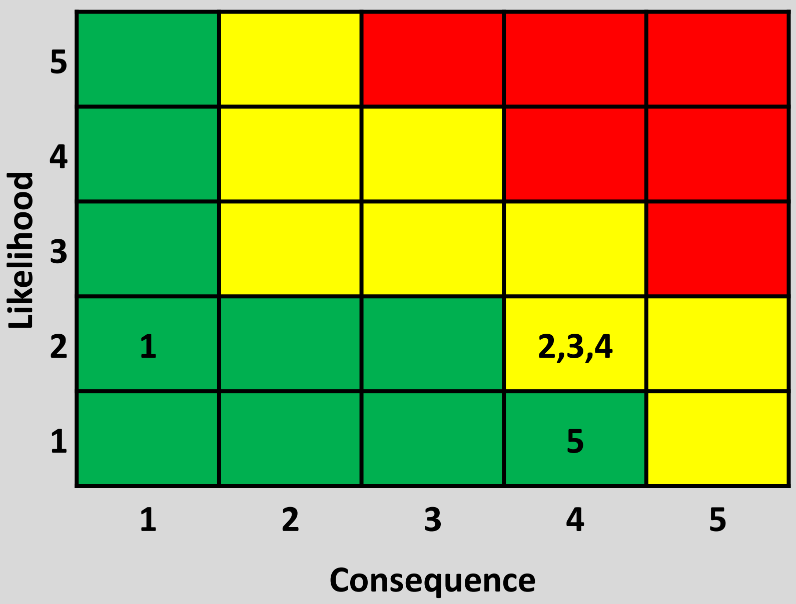
Level	Likelihood
1	1 in 10000
2	1 in 1000
3	1 in 100
4	1 in 10
5	1 in 2

Level	Consequence
1	Minimal Impact
2	Schedule slip < 1 week
3	Schedule slip < 2 weeks, some minor requirement not met
4	Schedule slip < 4 weeks, major requirements not met
5	Project Failure, irreplaceable components damaged beyond repair

Risk Matrix and Mitigation

Risk	Description	Mitigation
R1	Overcurrent to CS rotation motor during capture confirmation, costs increase & schedule slips	Hardware and Software Fail Safes
R2	Capture confirmation fails, demo failure	Force sensor testing prior to final demo
R3	LabVIEW errors, testing schedule slips	40% of team devoted to LabVIEW development
R4	Control Algorithm errors, testing schedule slips	20% of team devoted to Control development
R5	Motor malfunction, costs increase & schedule slips	New motors, testing, margin available





More Risks in Backup Slides

Verification and Validation

Verification/Validation Overview

Dynamics Model Validation

- Accuracy of claw placement along axis of rotation (AOR)
- Power Requirements Verification

Validates DR …

Validates DR 1.4.1 – 1.4.4

Timing Verification

- Phases 3 – 5 (Critical)

Validates DR 1.4.9

Data Sheet Verification

- Servos, motors, force sensors

Subsystem and Full System Tests

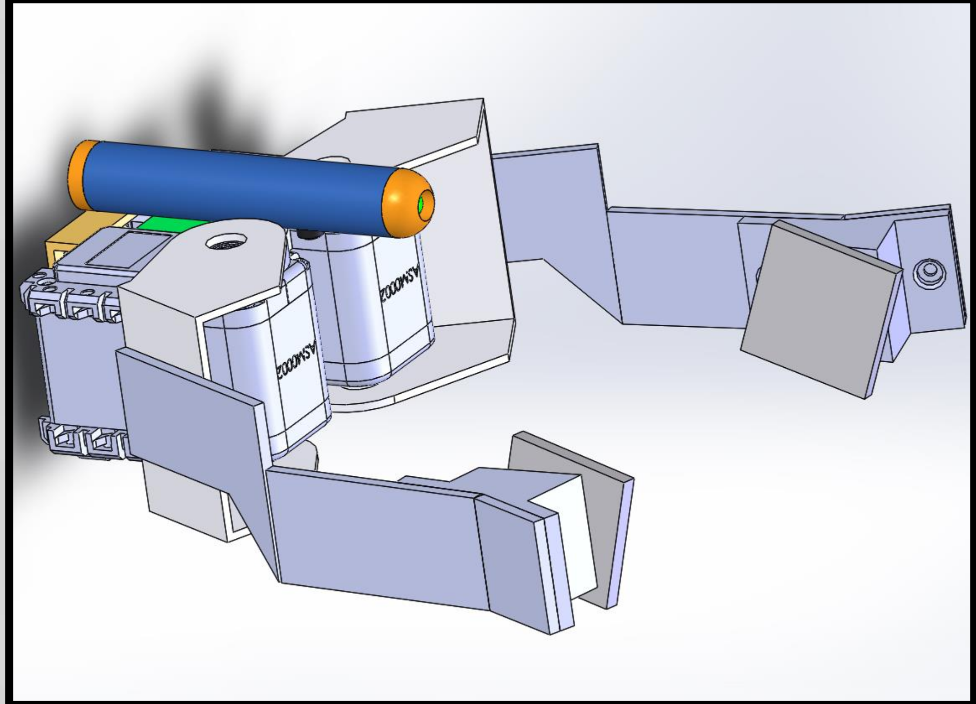
Validates FR 1.1 – 1.4

Dynamics Model - Accuracy

Objective	Validate Accuracy of Claw to align with AOR
Test	Axis of Rotation Accuracy
Date	January 31st
Duration	4 hr
Location	Senior Design Space in Engineering Center

Equipment	Resolution	Procurement
Laptop	-	Owned
Dynamixel MX-64R Servo	10^-3 rad	Purchased
USB2Dynamixel Controller	10^-3 rad	Purchased
CrustCrawler Pro-Series Arm	-	Purchased
Vicon	10^-3 m	Borrowed

Data Needed	Resolution Needed	Sampling Rate
Servo Position	10^-3 rad	100Hz
Laser Pointer radius	10^-3 m	
IR Sensor Position	10^-3 m	100Hz



Dynamics Model - Power

Objective	Verify Power Requirements
Test	Comparison of performance: Torque & Current
Date	February 15th
Duration	4 hr
Location	Senior Design Space in Engineering Center

Data Needed	Resolution Needed	Sampling Rate
Torque	10^-3 Nm	100Hz
Current	mA	100Hz
Voltage	mV	100Hz

Equipment	Resolution	Procurement
Laptop	-	Owned
Dynamixel MX-64R Servo	10^-3 rad	Purchased
USB2Dynamixel Controller	10^-3 rad	Purchased
CrustCrawler Pro-Series Arm	-	Purchased

Project Planning

Organization Chart

SIERRA NEVADA CORPORATION
Space Systems

CASCADE

- **Henry "Lad" Curtis** — Customer
- **Jelliffe Jackson** — Advisor
- **Andrew McBride** — Project Manager
  - **Noel Puldon** — Systems Lead
    - **Timothy Kiley** — Software Lead
      - **Tony Ly** — Algorithm Lead
      - **Haoyu Li** — Modeling Lead
    - **Keegan Sotebeer** — Hardware Lead
      - **Chad Eberl** — Mechanical Lead
      - **Morgan Tilong** — Electrical Lead
      - **Matthew Fromm** — Manufacturing Lead

Work Breakdown Structure

Aerospace Engineering

SNC SIERRA NEVADA CORPORATION Space Systems

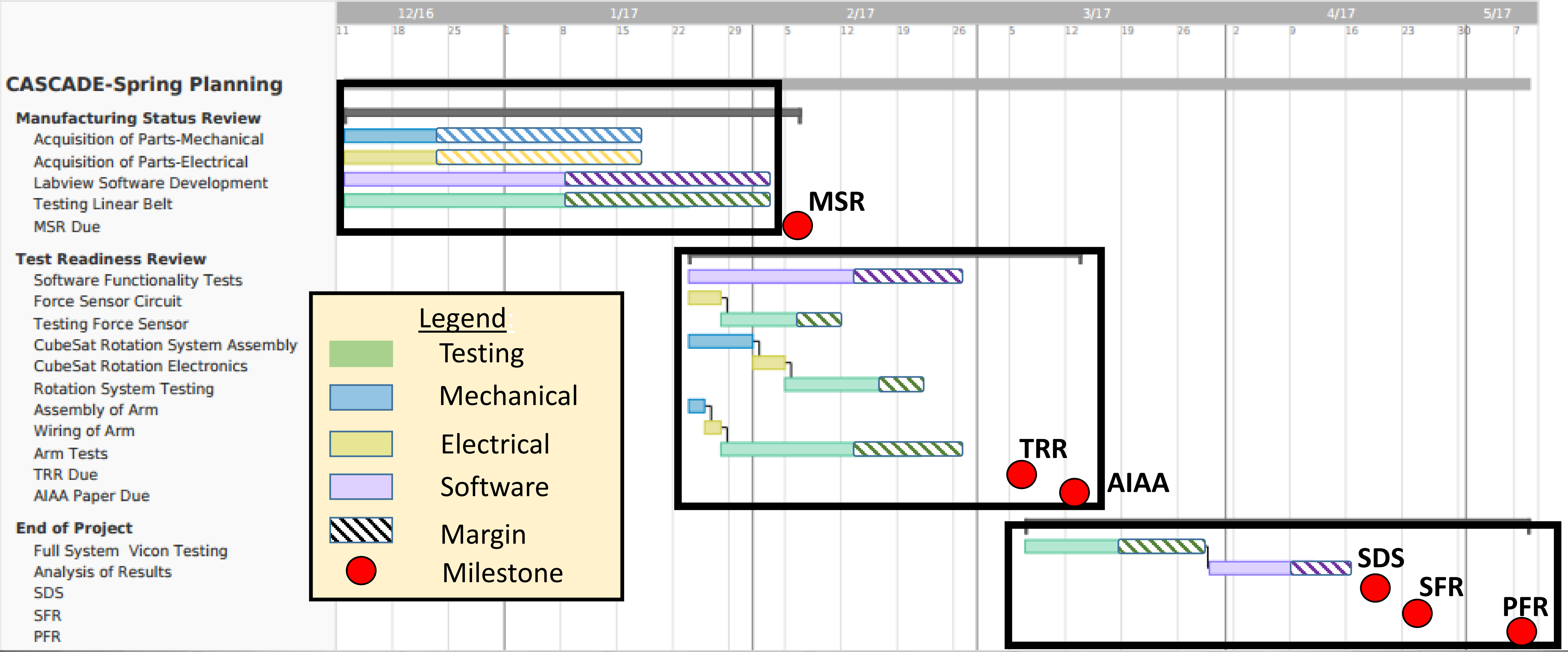
CASCADE

- **Management and Systems**
  - Interface Doc.
  - Schedule
  - Budget
  - Requirements
  - Risk Matrices
  - Order Parts
- **Hardware**
  - Electrical Schematic
  - SolidWorks
  - Dynamics Models
  - Power Budget
  - Robotic Arm Assembled
  - Rotation System Assembled
  - Force Sensor Circuit Built
- **Commands and Data**
  - Software Flow
  - NI DAQ Selected
  - Algorithm Functions
  - Timing Diagram
  - Vicon-Labview Data Stream
  - Executable LabView-Arm
  - Executable LabView-Rotation
  - Executable LabView-Translation
- **Testing**
  - Test Plan Doc.
  - Separate Test Data
  - Vicon Test Data
- **Deliverables**
  - CDR
  - FFR
  - MSR
  - TRR
  - AIAA
  - SDS
  - SFR
  - PFR

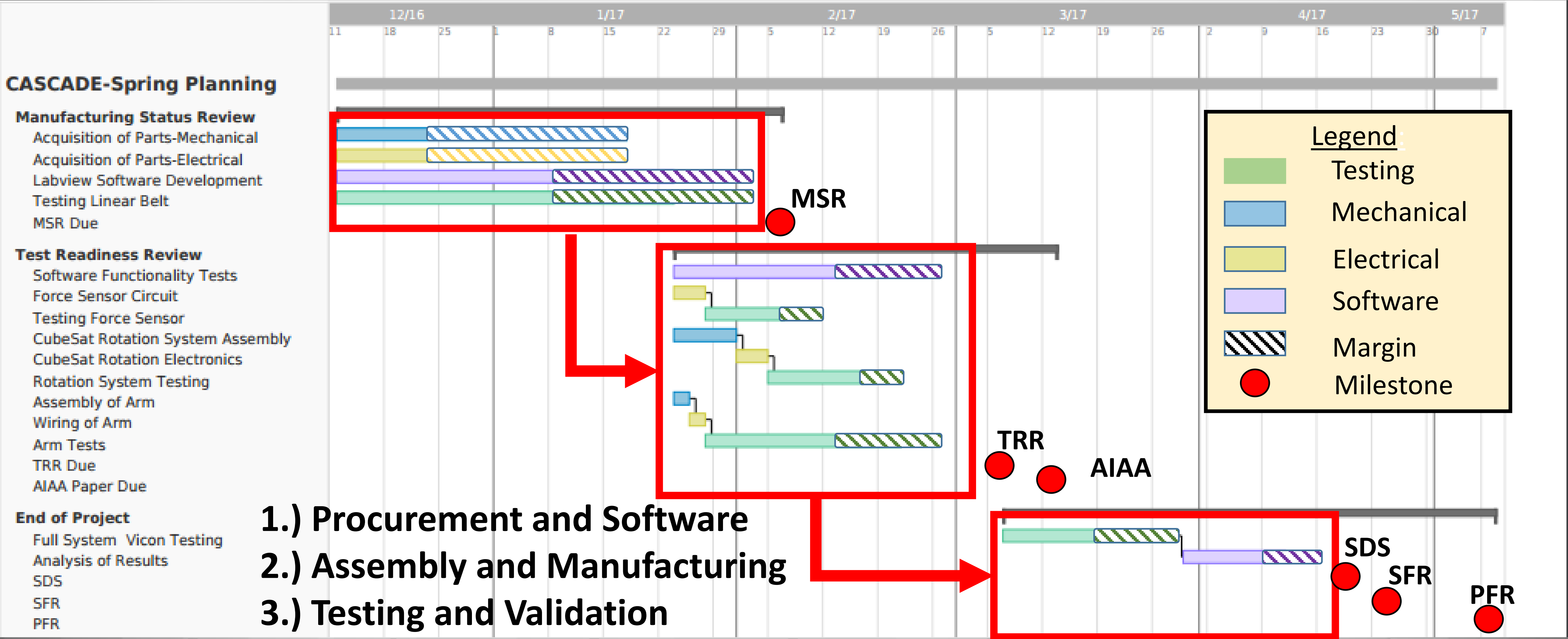
Legend:

- Complete
- Not Complete

Work Plan


12/16
1/17
2/17
3/17
4/17
5/17
CASCADE-Spring Planning
Manufacturing Status Review
Acquisition of Parts-Mechanical
Acquisition of Parts-Electrical
Labview Software Development
Testing Linear Belt
MSR Due
Test Readiness Review
Software Functionality Tests
Force Sensor Circuit
Testing Force Sensor
CubeSat Rotation System Assembly
CubeSat Rotation Electronics
Rotation System Testing
Assembly of Arm
Wiring of Arm
Arm Tests
TRR Due
AIAA Paper Due
End of Project
Full System Vicon Testing
Analysis of Results
SDS
SFR
PFR
Legend
Testing
Mechanical
Electrical
Software
Margin
Milestone
MSR
TRR
AIAA
SDS
SFR
PFR

Critical Path



1.) Procurement and Software

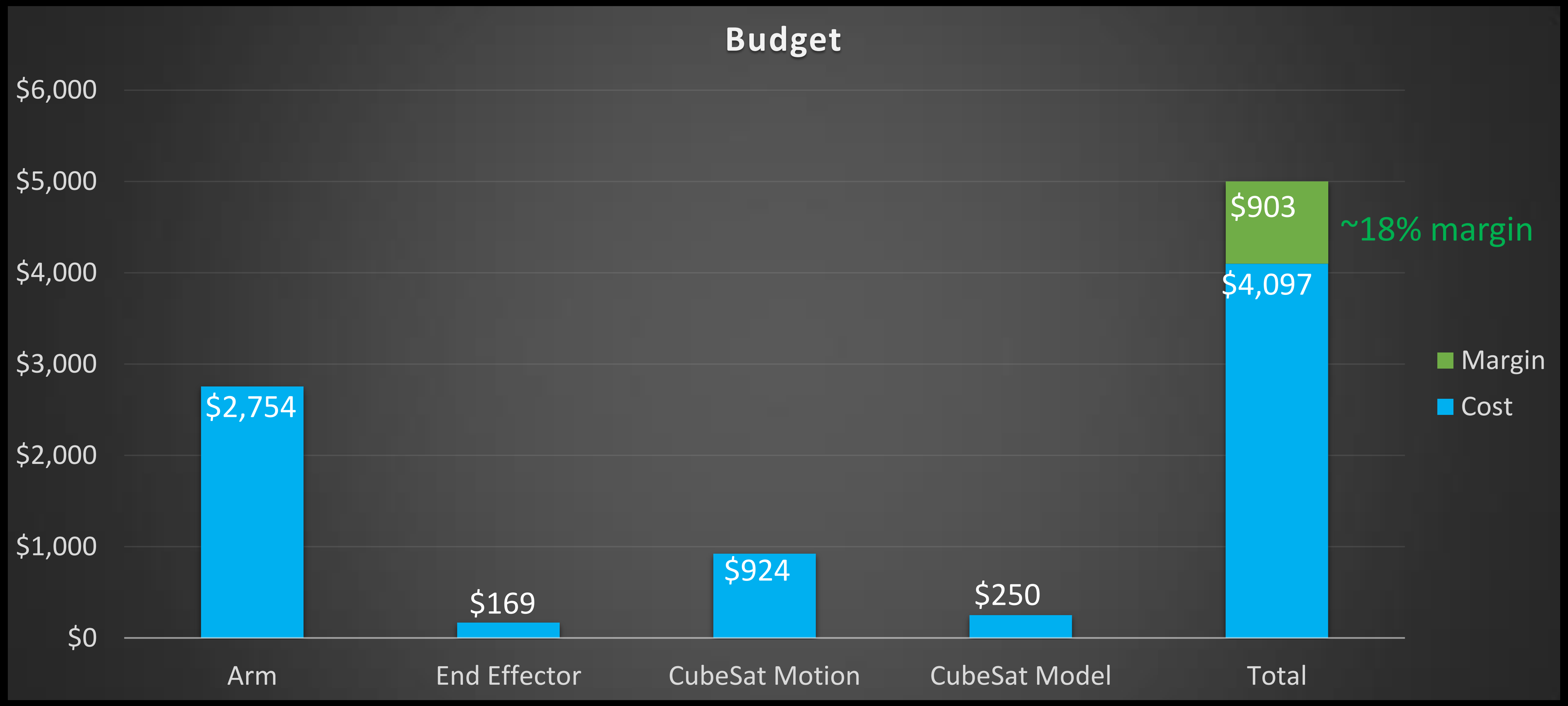
2.) Assembly and Manufacturing

3.) Testing and Validation

Build and Test Plan:

Week:	Testing Goal:	Key Date:
1-2 (1/17 -1/30)	Acquire Parts, Test Linear Rail, Software Development Finalized	MSR Due (2/6)
3-4 (1/31 -2/13)	Manufacture and Assemble Rotation System and Arm	TRR Due (3/6)
5-7 (2/14-2/27)	Test Arm, Rotation of CubeSat, and Force Sensor individually.	TRR Due (3/6)
8 (2/28-3/6)	Prepare for TRR with Dry Runs	
9 (3/7-3/13)	Submit AIAA paper, **Full Vicon Tests**	AIAA due (3/13)
10-13 (3/14-4/10)	**Full Vicon Tests** and Analysis	SDS Due (4/19)

Cost Plan- Overall Budget



Acknowledgements

Team CASCADE would like to acknowledge PAB members Professor Jackson and Professor Nabity for their continued assistance as we work to understand and develop solutions for this project. We would like to acknowledge AES faculty staff Trudy Schwartz, Matt Rhode, and Bobby Hodgkinson. We would also like to thank Professor Nikolaus Correll from the CSCI department for help with autonomous robotics. Steve McGuire for help understanding the RECUV lab. Finally, Brandon Antoniak for running around in RECUV for us.

References

- Correll, N., *Introduction to Autonomous Robots*, Magellan Scientific, 2016.
- AbuQassem, M. R., “Simulation and Interfacing of 5 DOF Robotic Arm,” thesis, 2010.
- Schaal, S., “Jacobian Methods for Inverse Kinematics and Planning.”
- Buss, S. R., “Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse, and Damped Least Squares Methods.”
- “State of the Satellite Industry Report,” *SIA - The Voice of the Satellite Industry* Available: .
- “Robotis Dynamixel Actuators in LabVIEW,” *Robotis Dynamixel Actuators in LabVIEW - National Instruments* Available: http://www.ni.com/white-paper/12557/en/.
- “ROBOTIS e-MANUAL,” *ROBOTIS e-MANUAL* Available: http://support.robotis.com/en/.
- “VICON DataStream SDK Developer’s Manual”. VICON. URL: http://andrewd.ces.clemson.edu/courses/cpsc412/manuals/Vicon%20DataStream%20SDK%20Manual.pdf
- P.I. Corke, “Robotics, Vision & Control”, Springer 2011, ISBN 978-3-642-20143-1
- Cheah, C. C., Kawamura, S. and Arimoto, S. (1999), Feedback control for robotic manipulator with an uncertain Jacobian matrix. J. Robotic Syst., 16: 119–134. doi:10.1002/(SICI)1097-4563(199902)16:2<119::AID-ROB5>3.0.CO;2-J

Thank you
from Team CASCADE!

Andrew McBride	PM
Noel Puldon	Systems
Tim Kiley	Software
Keegan Sotebeer	Hardware
Tony Ly	Algorithm
Haoyu Li	Modeling
Chad Eberl	Mechanical
Matthew Fromm	Manufacturing
Morgan Tilong	Electronics

FR 1.1 Breakdown

FR 1.1: The CRST shall demonstrate the motion of a CubeSat analogue during the demonstration.
DR 1.1.1: The CubeSat analogue shall allow for translational motion about only one axis.
Source: Dictated by scope/ levels of success
<u>*Verification:* Inspection</u>
DDR 1.1.1.1: The CubeSat analogue shall employ a motor with a minimum torque of 0.06 Nm to translate the CubeSat.
Source: Derived based on mass properties and friction of linear rail.
<u>*Verification:* Test</u>
DDR 1.1.1.2: The linear translation of the CubeSat shall be commanded to perform within motor performance limits.
Source: Needed in order to avoid the risk of overcurrent to the motor.
<u>*Verification:* Demonstration</u>
DR 1.1.2: The CubeSat analogue shall allow for rotational motion about only one axis.
Source: Dictated by scope/ levels of success
<u>*Verification:* Inspection</u>
DDR 1.1.2.1: The CubeSat analogue shall employ a minimum torque of 0.14 Nm to rotate the CubeSat Model.
Source: Derived based on angular velocity of the CubeSat and its mass properties along with frictional torques.
<u>*Verification:* Test</u>
DR 1.1.3: The CubeSat model shall weigh 3kg..
Source: Customer Requirement.
<u>*Verification:* Inspection</u>

FR 1.2 Breakdown

FR 1.2: The CRST shall determine the relative position and attitude between the CubeSat and capture device during the demonstration.

DR 1.2.1: The CRST shall communicate with the Vicon Motion Capture System to sense the initial conditions and motion of the CubeSat relative to the base of the robotic arm (origin) throughout the demonstration.

Source: Dictated by highest levels of success

Verification: Test

DR 1.2.2: The CRST shall determine the axis of rotation of the CubeSat model during the demonstration.

Source: Needed to align the end effector with the CubeSat for ease of capture.

Verification: Test

DR 1.2.3: The CRST shall determine the relative linear position of the CubeSat model during the demonstration.

Source: Needed to bring the CubeSat to the grab zone of the arm.

Verification: Test

DR 1.2.4: The CRST shall calculate the desired end effector location and orientation during the demonstration.

Source: Needed to align the end effector with the CubeSat for ease of capture

Verification: Test

FR 1.3 Breakdown

FR 1.3: The CRST shall command the motion of the capture device during the demonstration.

DR 1.3.1: The CRST shall calculate the current end effector location and orientation during the demonstration.

Source: Dictated by scope/ levels of success.

Verification: Inspection

DR 1.3.2: The commands sent for capture device motion shall be within joint servos performance limits.

Source: Derived based on mass properties and friction of linear rail.

Verification: Test

DDR 1.3.2.1: The CRST shall send commands at a minimum rate of 10.5 Hz.

Source: In order to meet power requirements, operation of the motors must be below the stall current

Verification: Test

FR 1.4 Breakdown

FR 1.4: The CRST shall execute capture of the physical CubeSat model autonomously during the demonstration.

DR 1.4.1: The capture device shall have an average power of no more than 100W.
Source: Customer Requirement
Verification: Test

DR 1.4.2: The capture device shall have an peak power of no more than 168W.
Source: Customer Requirement
Verification: Test

DR 1.4.3: The capture device shall have an peak current draw of no more than 10A.
Source: Customer Requirement
Verification: Test

DR 1.4.4: The capture device shall have an peak voltage draw of no more than 28V ± 6V unregulated
Source: Customer Requirement
Verification: Test

DDR 1.4.5: The end effector of the capture device shall have a minimum grip force of 1.1 N.
Source: Needed to capture the CubeSat based on the coefficient of friction between the force sensors and the CubeSat.
Verification: Test

Continued on next slide…

FR 1.4 Breakdown

FR 1.4: The CRST shall execute capture of the physical CubeSat model autonomously during the demonstration.

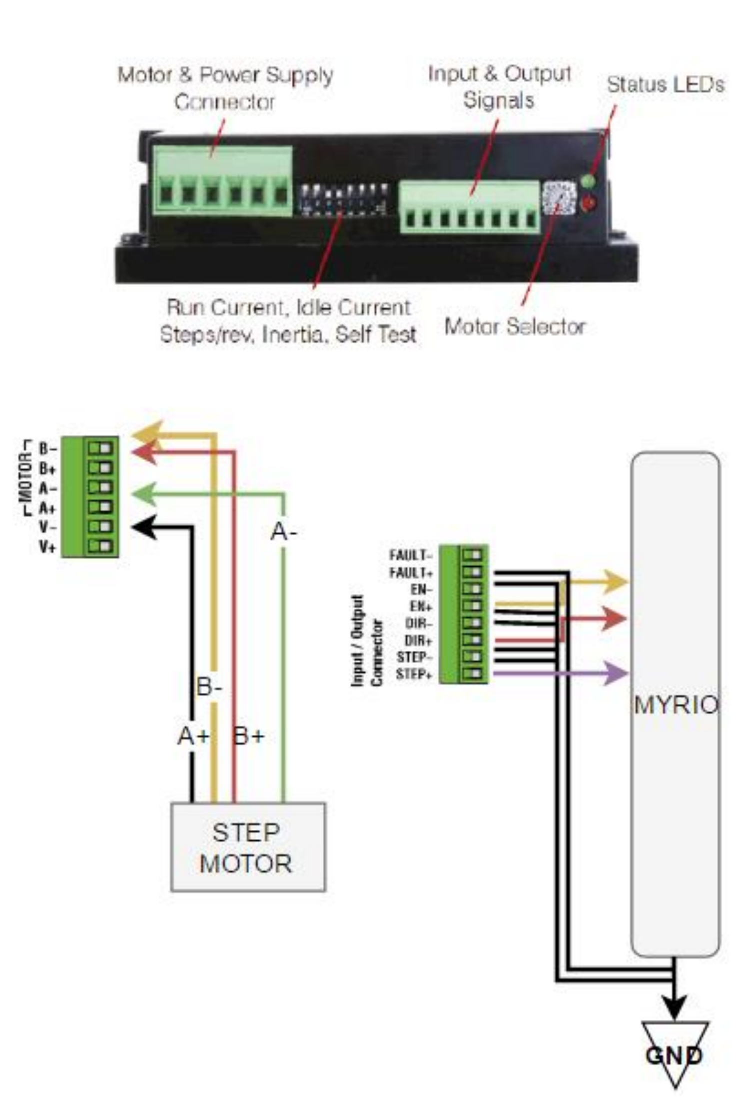
DR 1.4.6: The capture device shall be able to release from the CubeSat after capture without human intervention.
Source: Dictated by scope/ levels of success
Verification: Demonstration

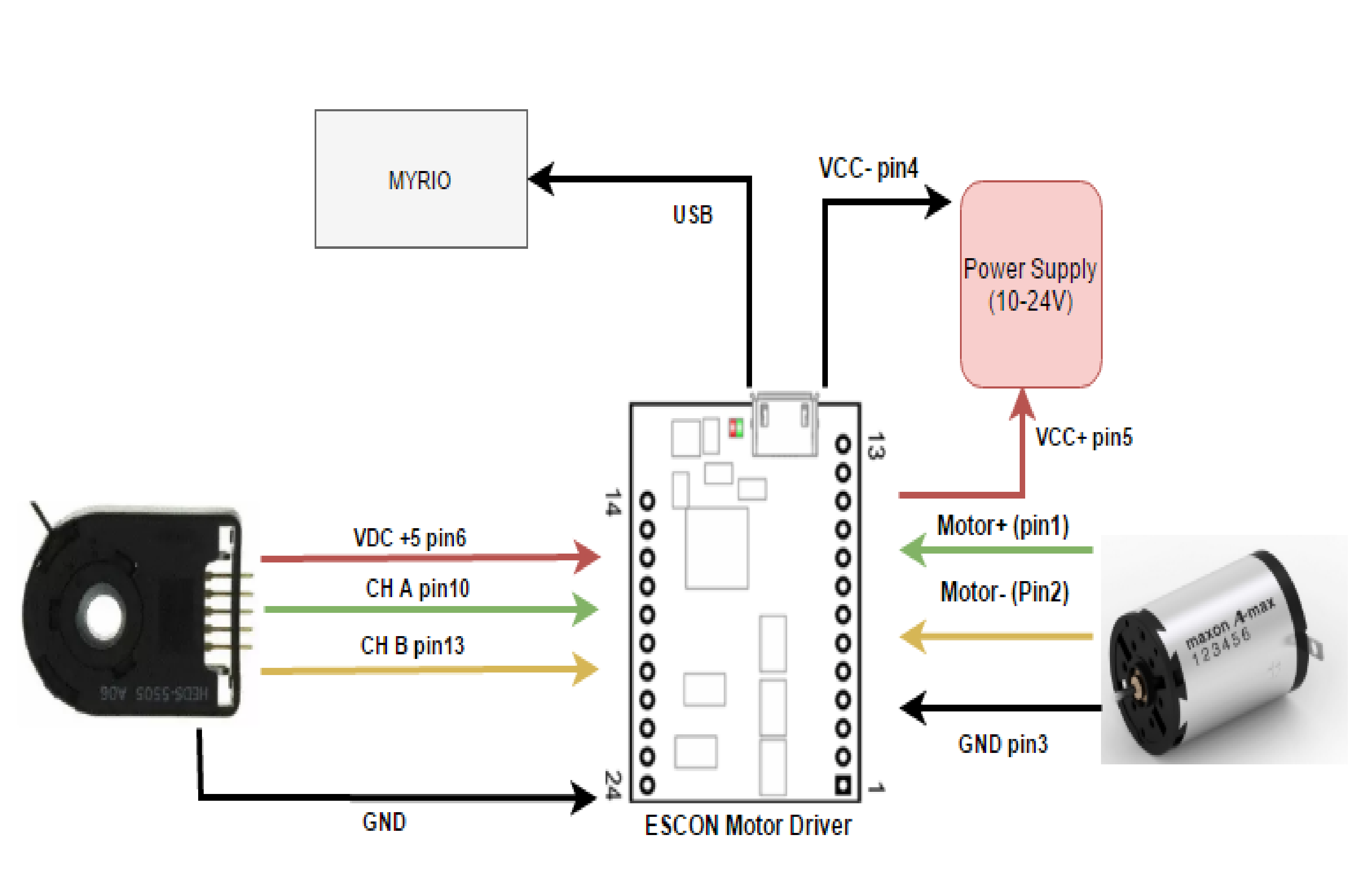
DR 1.4.7: The capture device shall be able to confirm the capture of the CubeSat after capture without human intervention.
Source: Dictated by scope/ levels of success
Verification: Demonstration

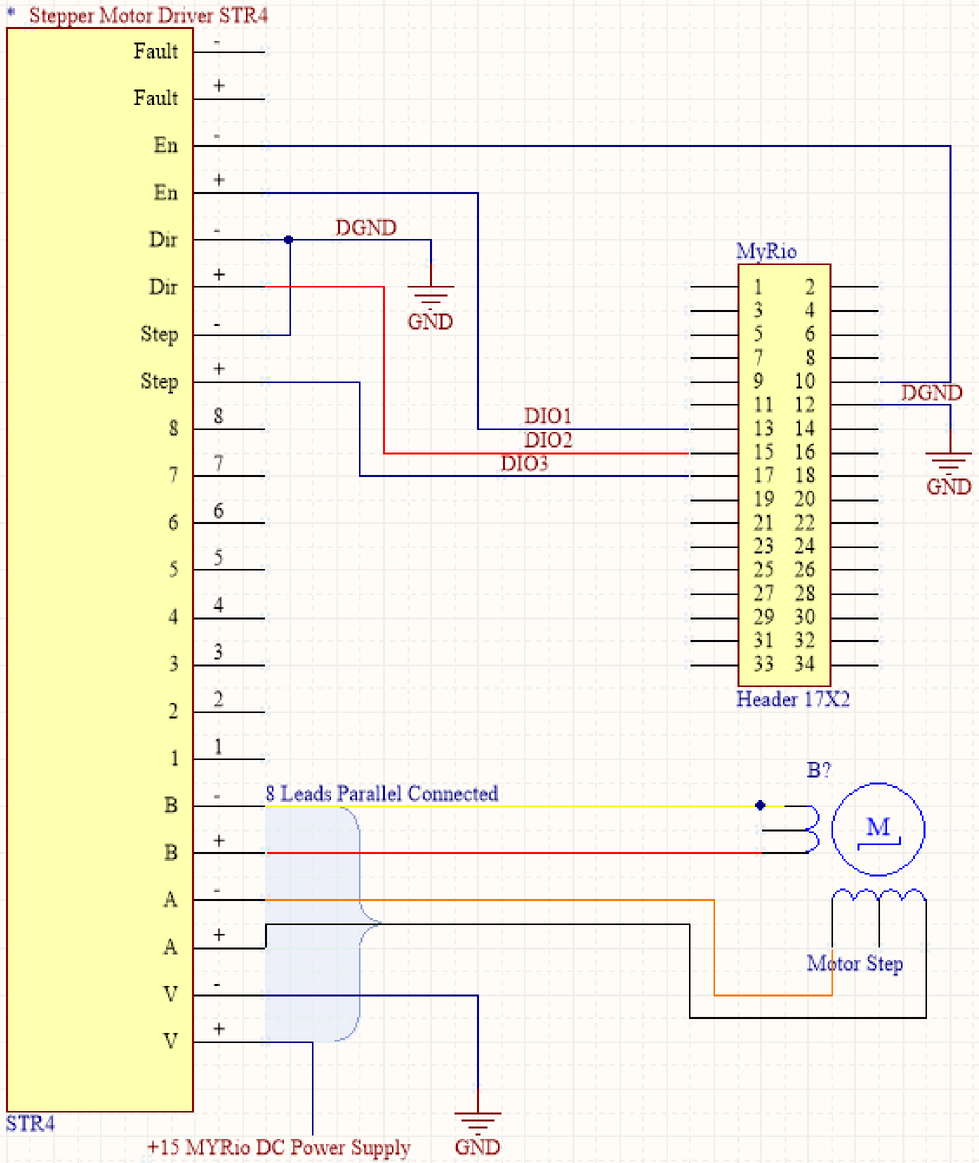
DR 1.4.8: The capture device shall confirm capture of the CubeSat model in less than 30 minutes.
Source: Customer Requirement
Verification: Test

DDR 1.4.9: The capture device wrist shall rotate less than two revolutions from its initial orientation.
Source: Needed to reduce the possibility of severing the wires to the force sensors.
Verification: Test

Motor Schematics







The STR drives have three inputs:
• STEP: a high speed digital input for step pulse commands, 5-24 volt logic
• DIR: a high speed digital input for the direction signal, 5-24 volt logic
• EN: a 5-24V input for removing power from the step motor. When the EN input is activated the motor is disabled. Activating then de-activating the EN input clears alarms and faults, and re-enables the motor in the case of drive faults.
Fault: This output can be used to drive LEDs, relays and the inputs of other electronic devices like PLCs (programmable logic controller). Aslo a fault output notifies the user that peak current was reached

---

SET Current switches (1 and 2) to reduce current from 100% t0 70%

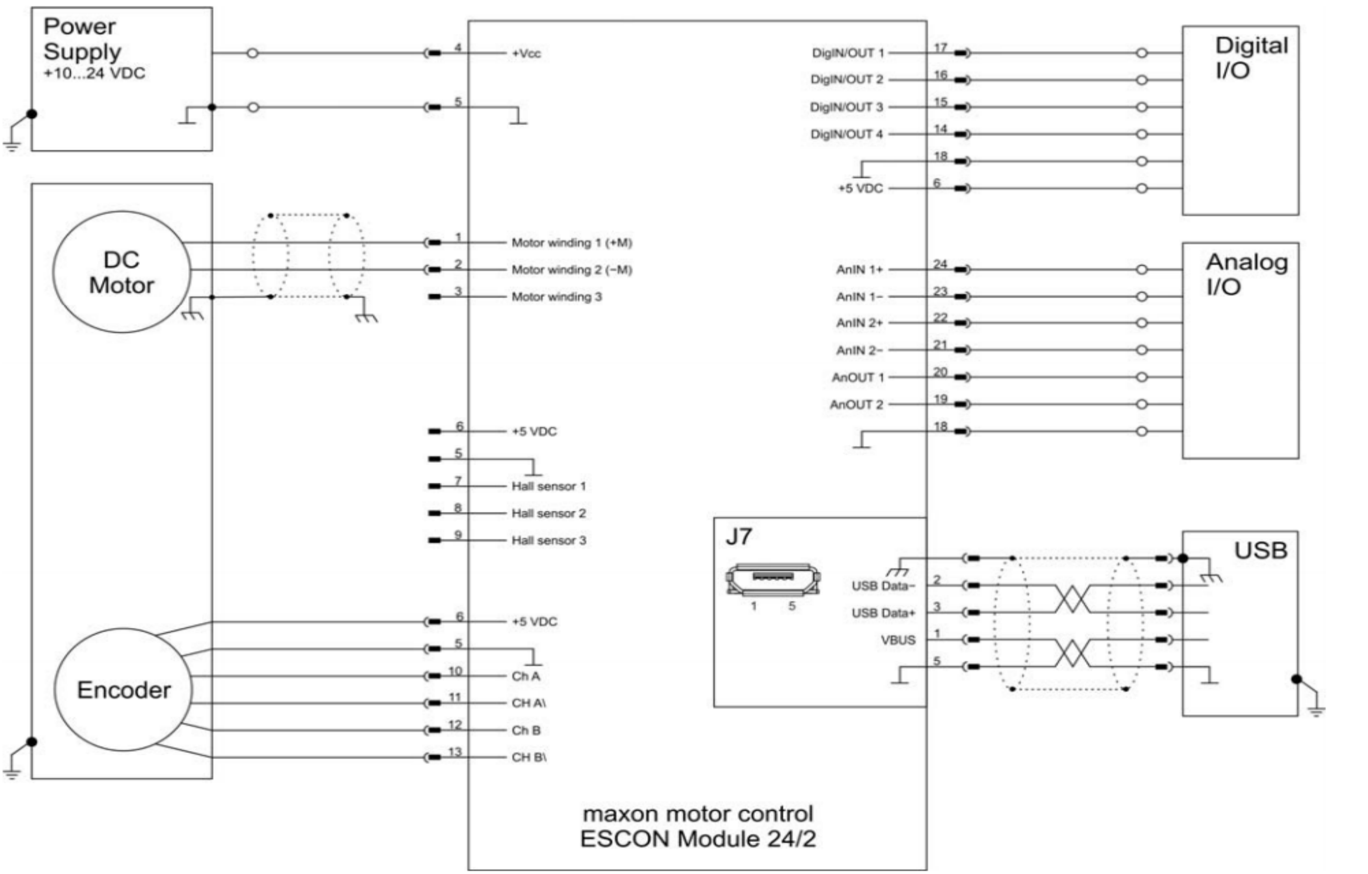
We can inhance by 0-4x or 4-10x anti-resonance and electronic damping features which greatly improve motor performance by using switch 3.

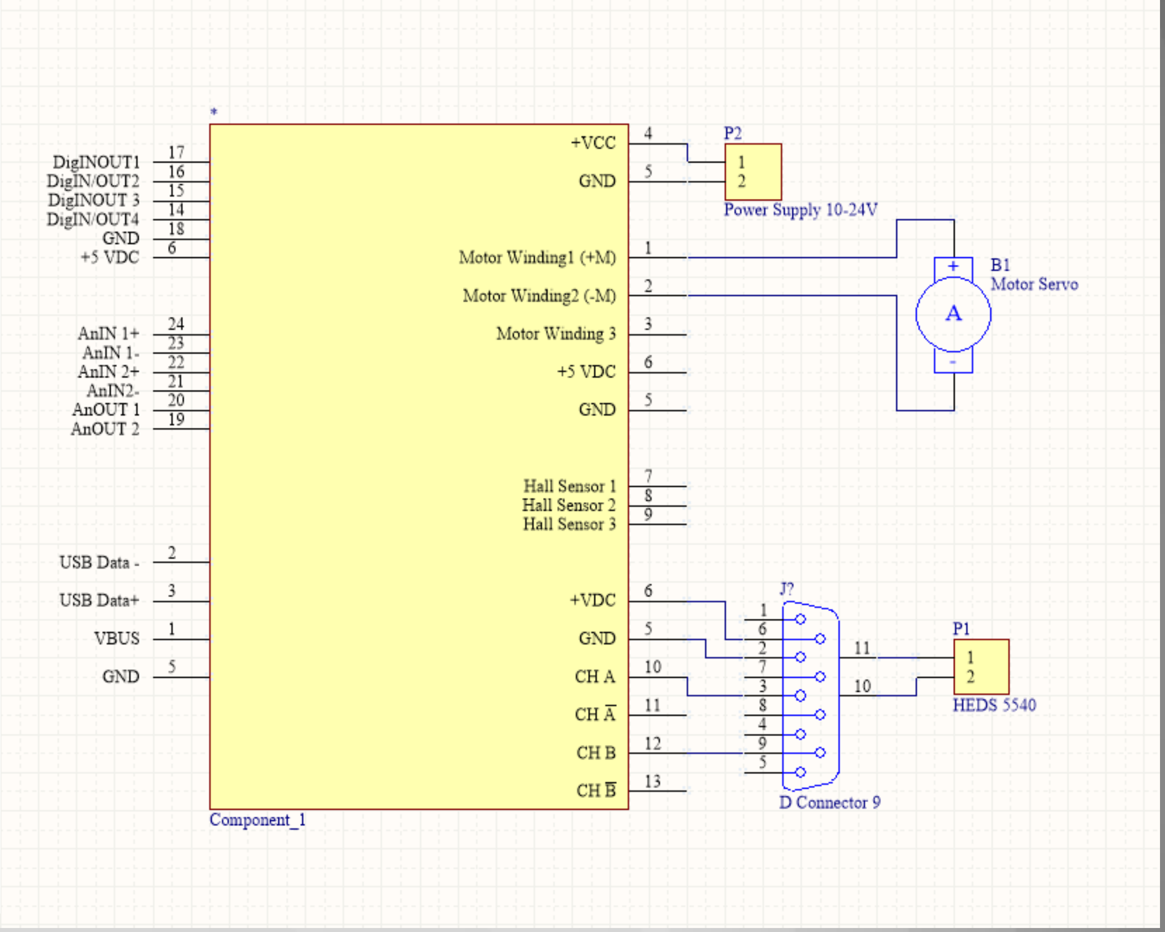
Turning Switches 5,6,7 on.
Smaller step sizes result in smoother motion and more precise speed, but also require a higher step pulse frequency to achieve maximum speed. The smallest step size of the STR drives is 1/20,000th of a motor turn. To command a motor speed of 50 revolutions per second (3000 rpm) the step pulses frequency must be 50 x 20,000 = 1 MHz.

Idle Current: Switch 4 on or off. Reduces current to 70% or 50%.

Switch 8: Self Test.

MAXON DC MOTOR WITH ENCODER





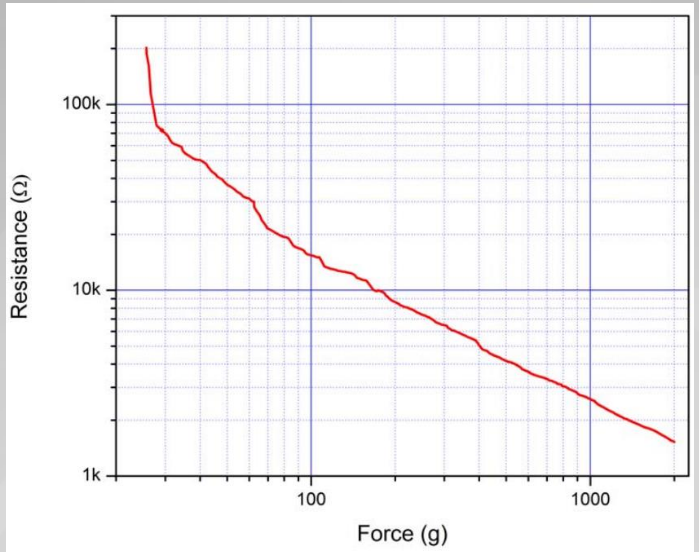
GPS-4303 Instek DC Power Supply

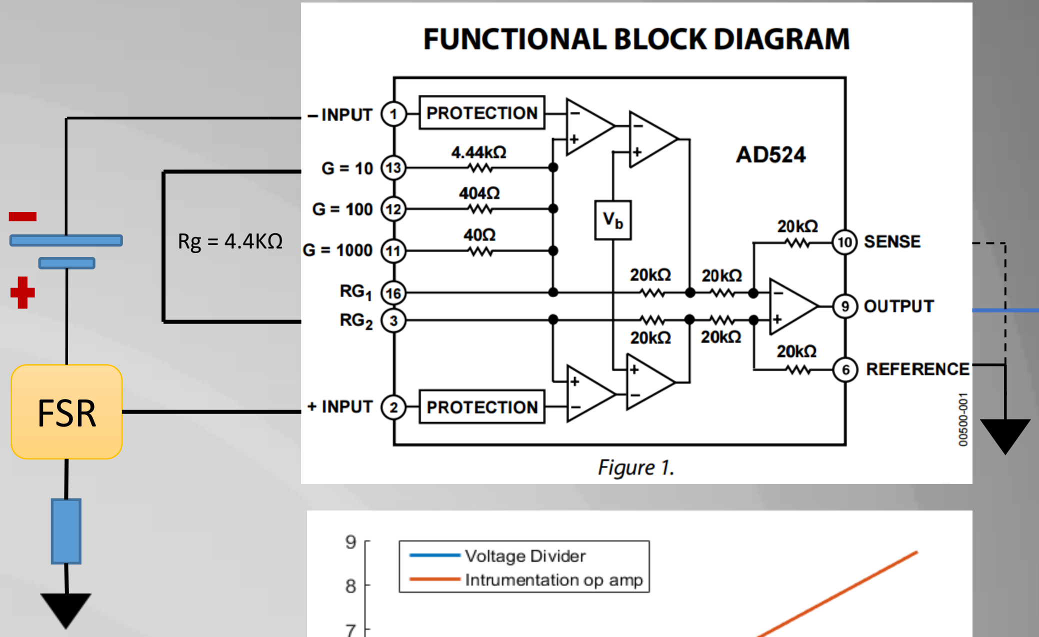
	CH1	CH2	CH3
Voltage	0-30 V	0-30 V	2.2-5.2 V
Current	0-3 A	0-3 A	1 A (max)
Power	0-90W	0-90W	2.2-5.2W
Component (Plug In)	**ESCON Module Motor Controller**	**STR4 Stepper Motor Driver**	**FSR**
Operating Power Range	10-25V	24-48V	6 - 18 V
	0-62.5W (250W max)	26.88- 71.04W (216W max)	>1W (5mA)

- **Adjustable 200W Power Supply with 4 isolated outputs.**

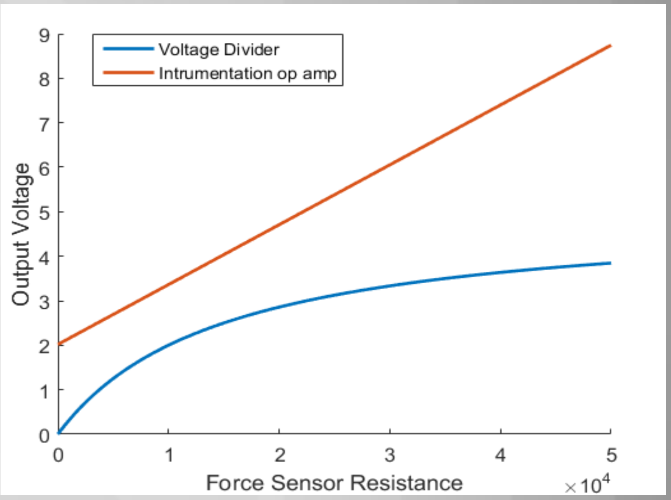
Force Sensing Resistor

- AD524 precision instrumentation amplifier will be used for anti aliasing and filtering.
- An Rg of 4.44 KΩ will be chosen to produce a gain of 10 and a Common mode rejection ratio of -100db (noise rejection).
- Assuming every resistor in our instrumentation amplifier is the same at 20Kohms:





$$V_{out} = \left(\frac{R_p + R_{Fsr}}{5v * R_p}\right) * \left(1 + 2\,\frac{20\ kohms}{R_g}\right)$$



R_{FSr} : *Force Sensor Resistance*

R_p: *Parallel resistance*

R_g: *Gain resistance*

Cutoff Frequency for FSR

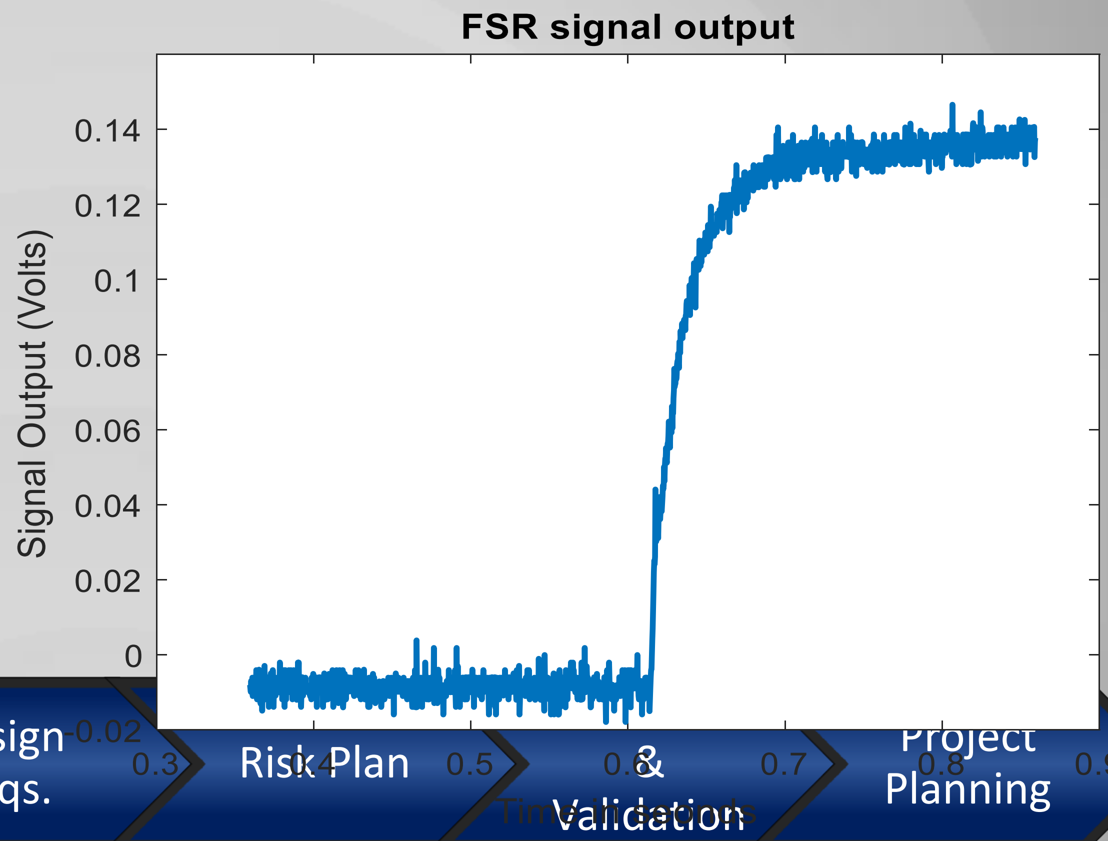
- The FSR's ability to respond to high speed human contact is illustrated in the plot to the left .
- The frequency of this response calculated calculated to be 7.07 Hz.
- There for a low pass filter with time constant of 22.5ms will be used to attenuate frequencies above the cutoff.

$$t(10\%\ to\ 90\%) = 2.2\tau = \frac{\mathbf{0.35}}{f_c}$$

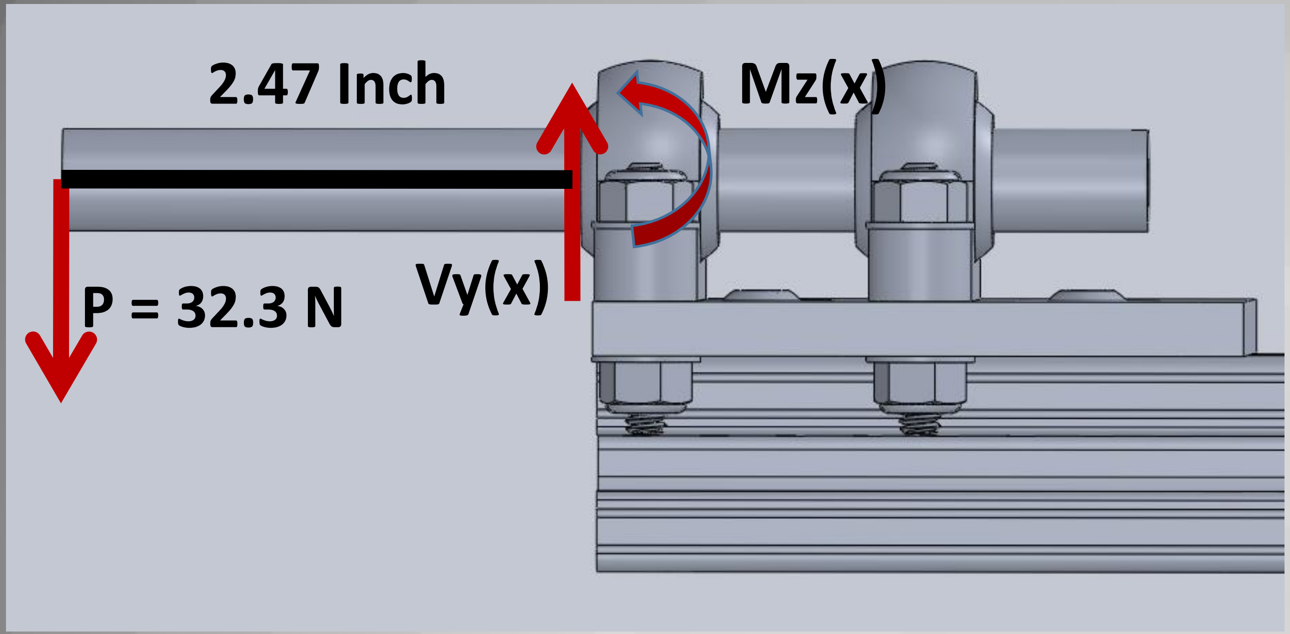
$$f_c = \frac{0.35}{0.0495s} = \mathbf{7.07Hz}$$

$$\tau = \frac{0.35}{7.07Hz*2.2} = \mathbf{0.0225s}$$

	Volts (mV)	Time
10%	14.7mV	0mS
90%	132mV	49.5mS



Deflection Of Support Shaft -Setup



$$Mz(x) = -Px \quad EIzzV''(x) = Mz(x) = -Px$$

$$EIzzV'(x) = \frac{-Px^2}{2} + C1$$

$$EIzzV(x) = \frac{-Px^3}{6} + C1x + C2$$

$$V'(L) = 0$$

$$V(L) = 0$$

$$V(x) = -\frac{P}{6EIzz}(x^3 - 3L^2x + 2L^3) = -\frac{P}{6EIzz}(L-x)^2(2L+x)$$

$$V(0) = V_A = -\frac{PL^3}{3EIzz} \quad Izz = \frac{\Pi d^4}{64}$$

Assumptions:

-Bearings hold shaft to behave like cantilever

-Point mass at end simulates max deflection

Support Shaft- Sensitivity Analysis

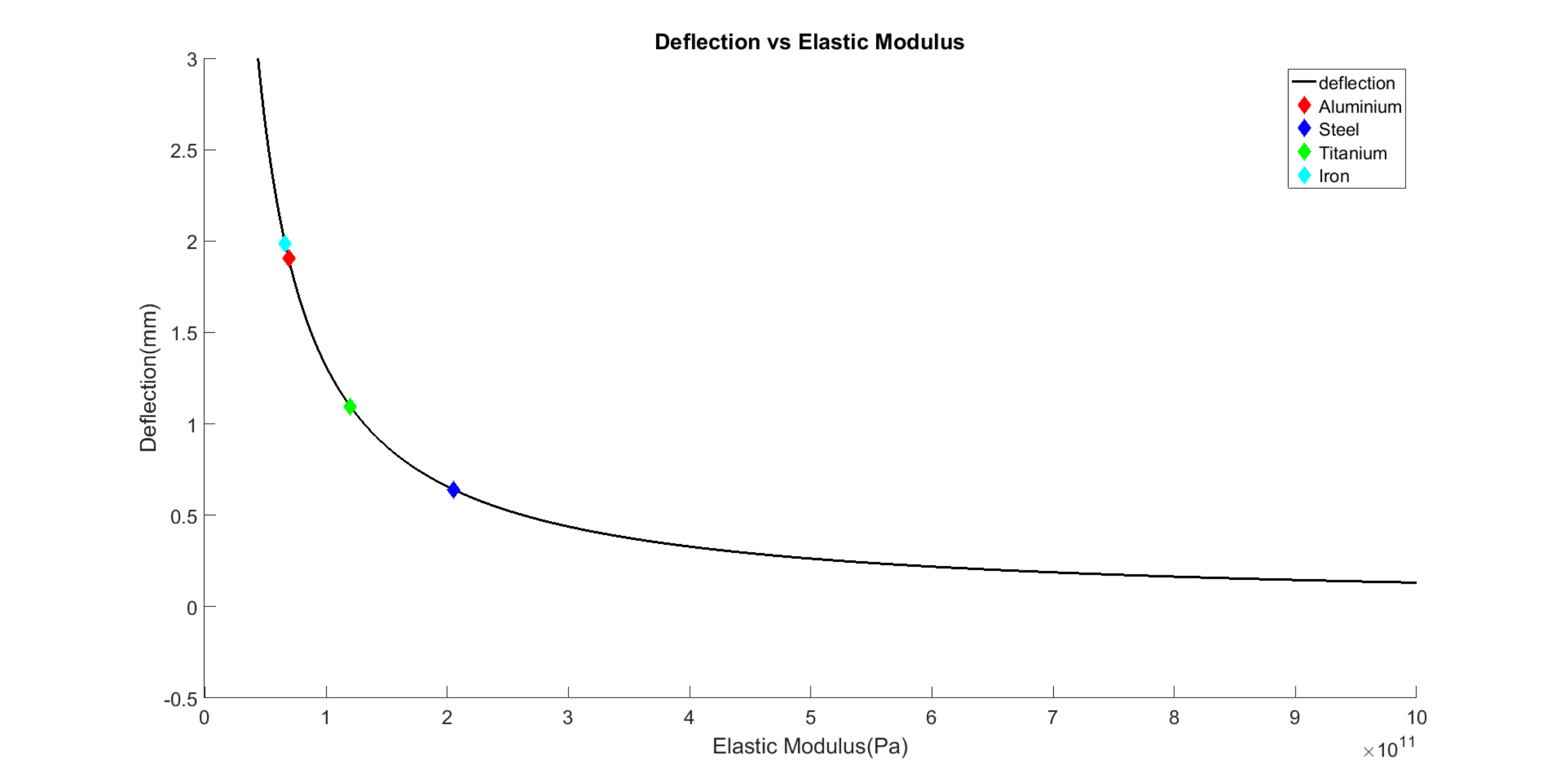
Material Modulus of Elasticity

E Aluminium = 69e9 Pa

E Mild Steel = 2.05e11 Pa

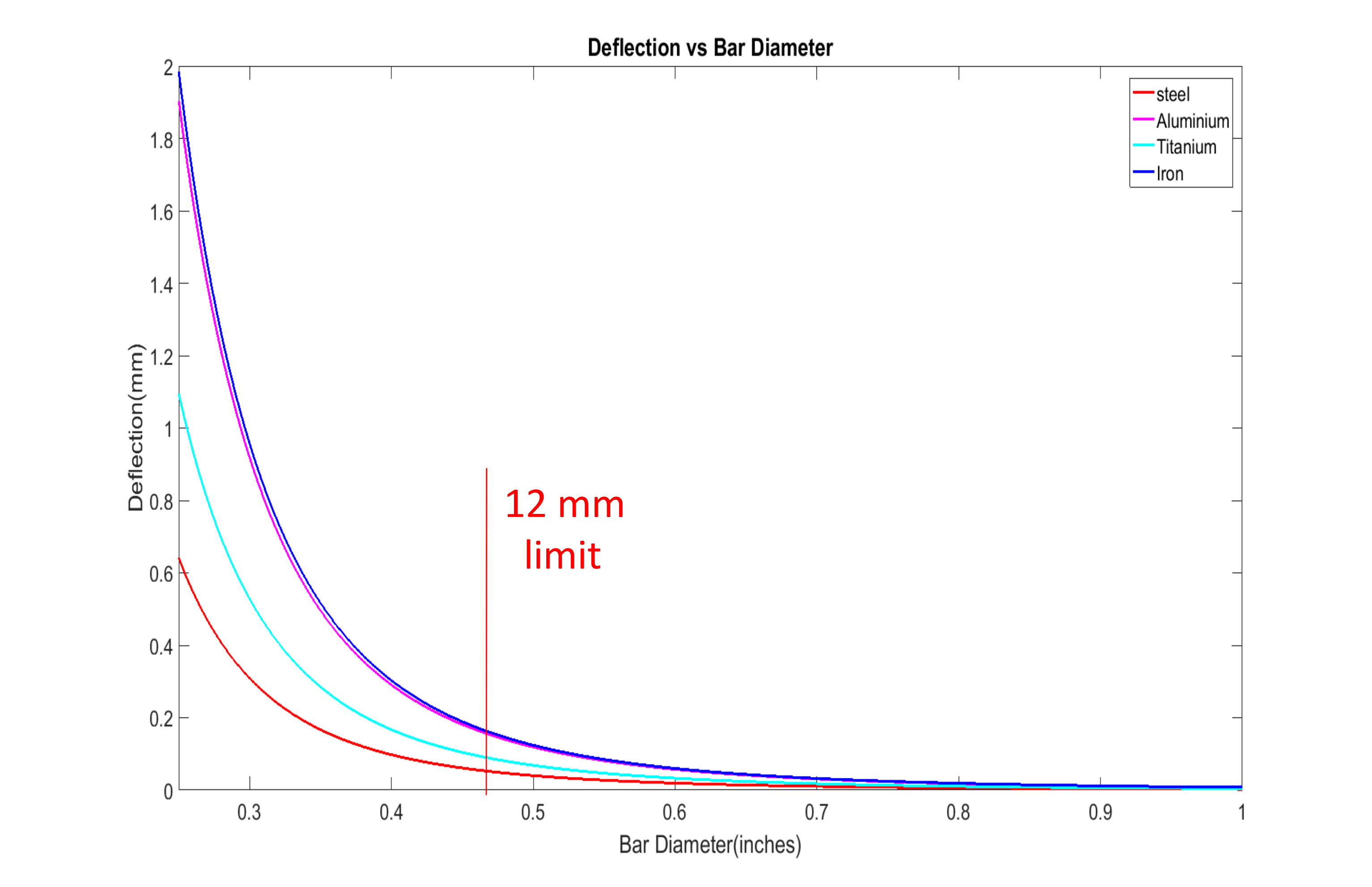
E Titanium = 1.2e11 Pa

E Iron = 6.62e10 Pa



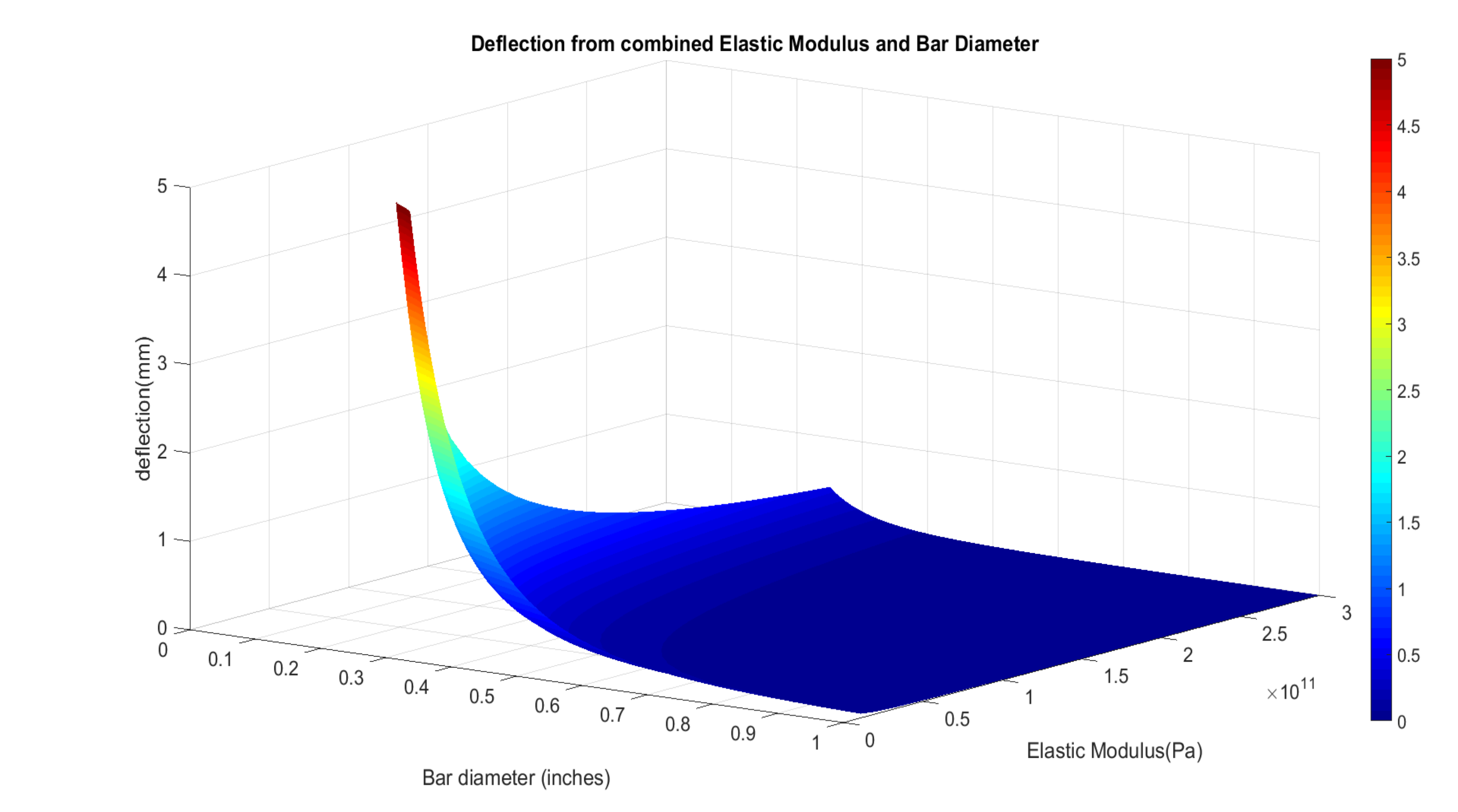
Higher Modulus of Elasticity results in decreased shaft deflection

Support Shaft- Sensitivity Analysis



Increased shaft diameter results in less deflection, but diameter is limited by motor couplings available

Support Shaft- Sensitivity Analysis



Conclusion:
Maximize rod diameter and elastic modulus

Deflection Of Support Shaft

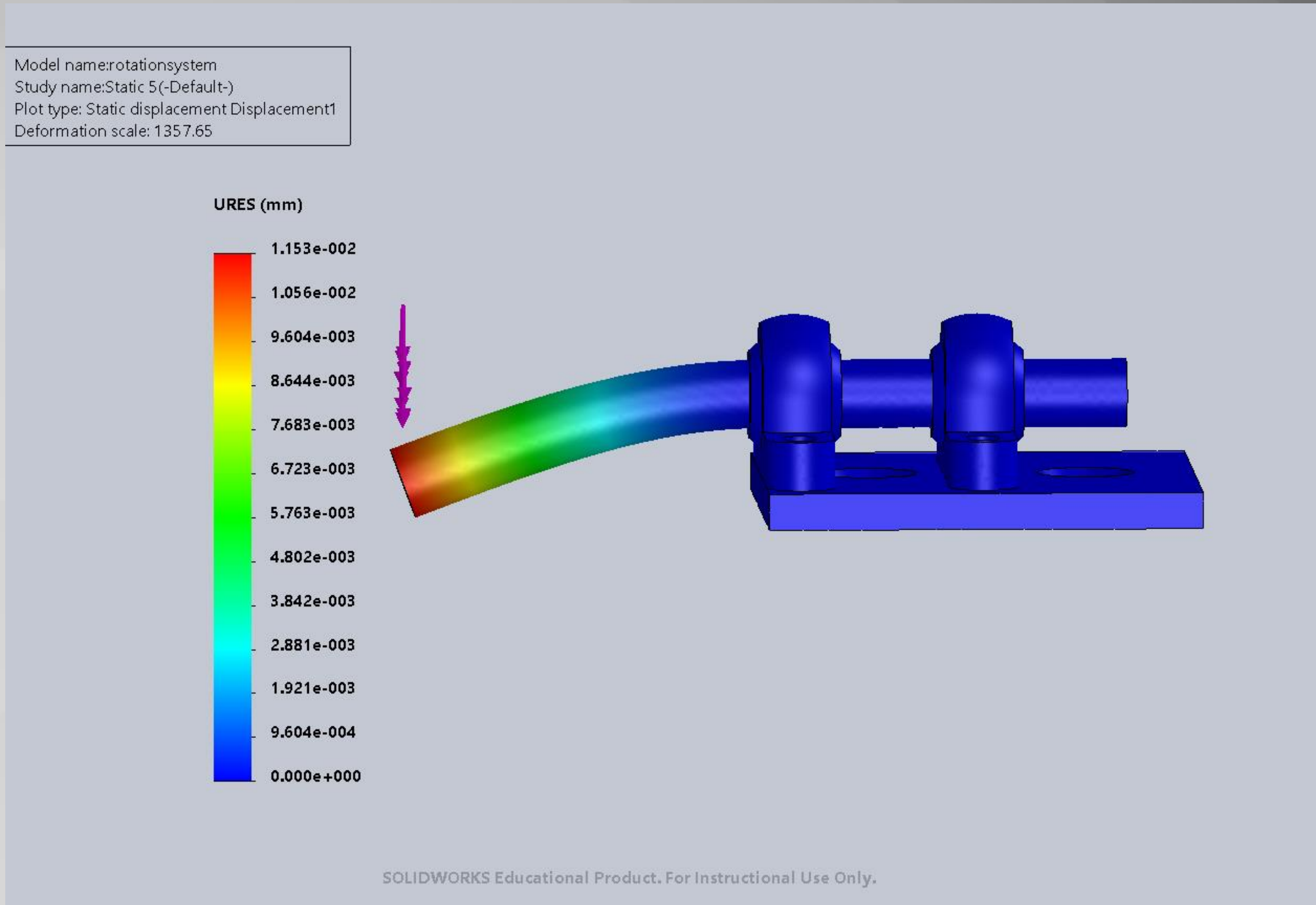
Design Choices:

Material: Mild Steel

Bar Diameter: 12mm (.4742 inches)

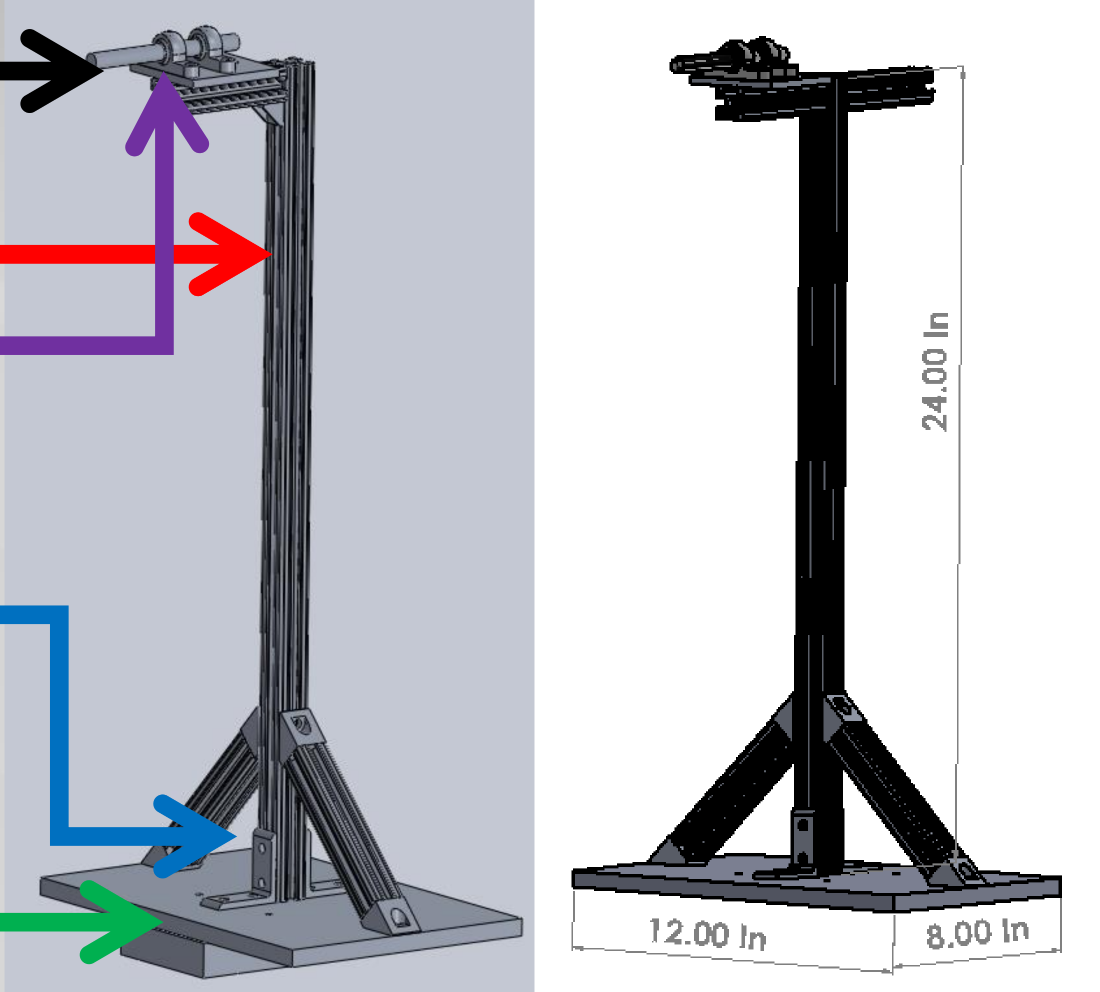
Analytic Deflection: 0.0118 mm

Simulated Deflection: 0.0115 mm



Rotation System - Geometry

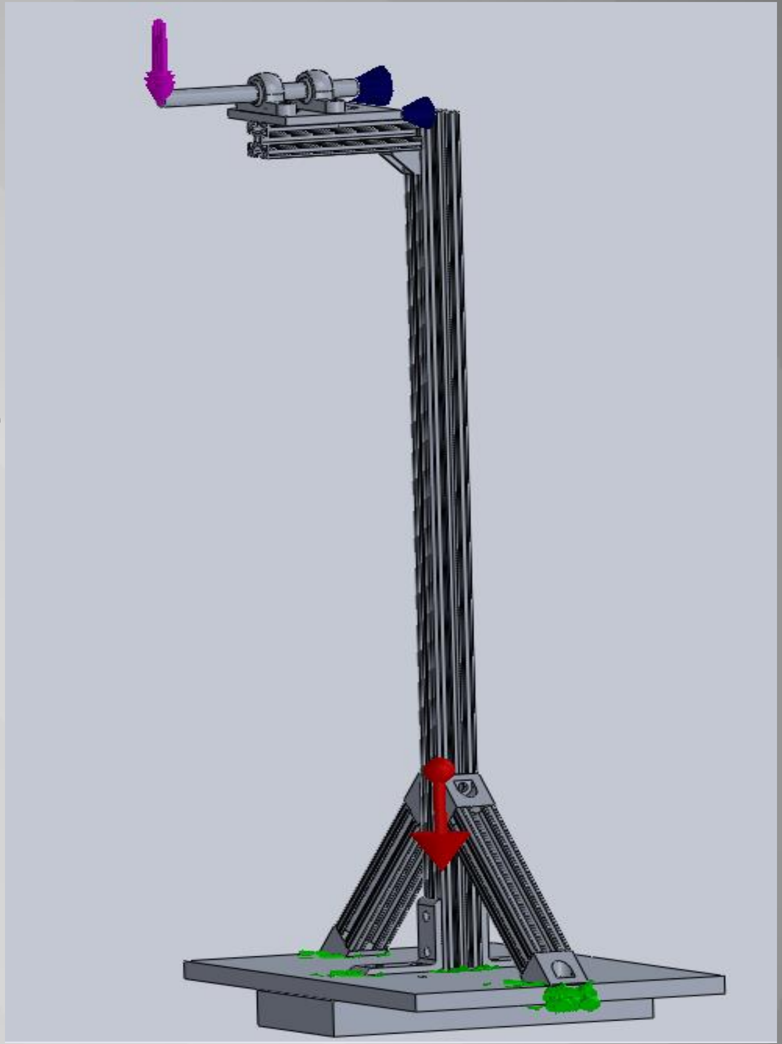
- Solidworks Model
  - Aluminum T-slot Framing
  - 12 mm mounted sleeve bearings
  - 12 mm support shaft
  - ERV carriage interface plate
  - Support legs and corner brackets



Rotation System - Simulation

Solidworks Model Boundary Conditions:

- Force of the CubeSat weight directed as point force on end of support member
- Gravitational force directed through entire structure
- Bearing support frame rigidly bonded to move with remaining structure
- Remainder of frame bonded to replicate fixtures
- Frame fixed to the interface plate

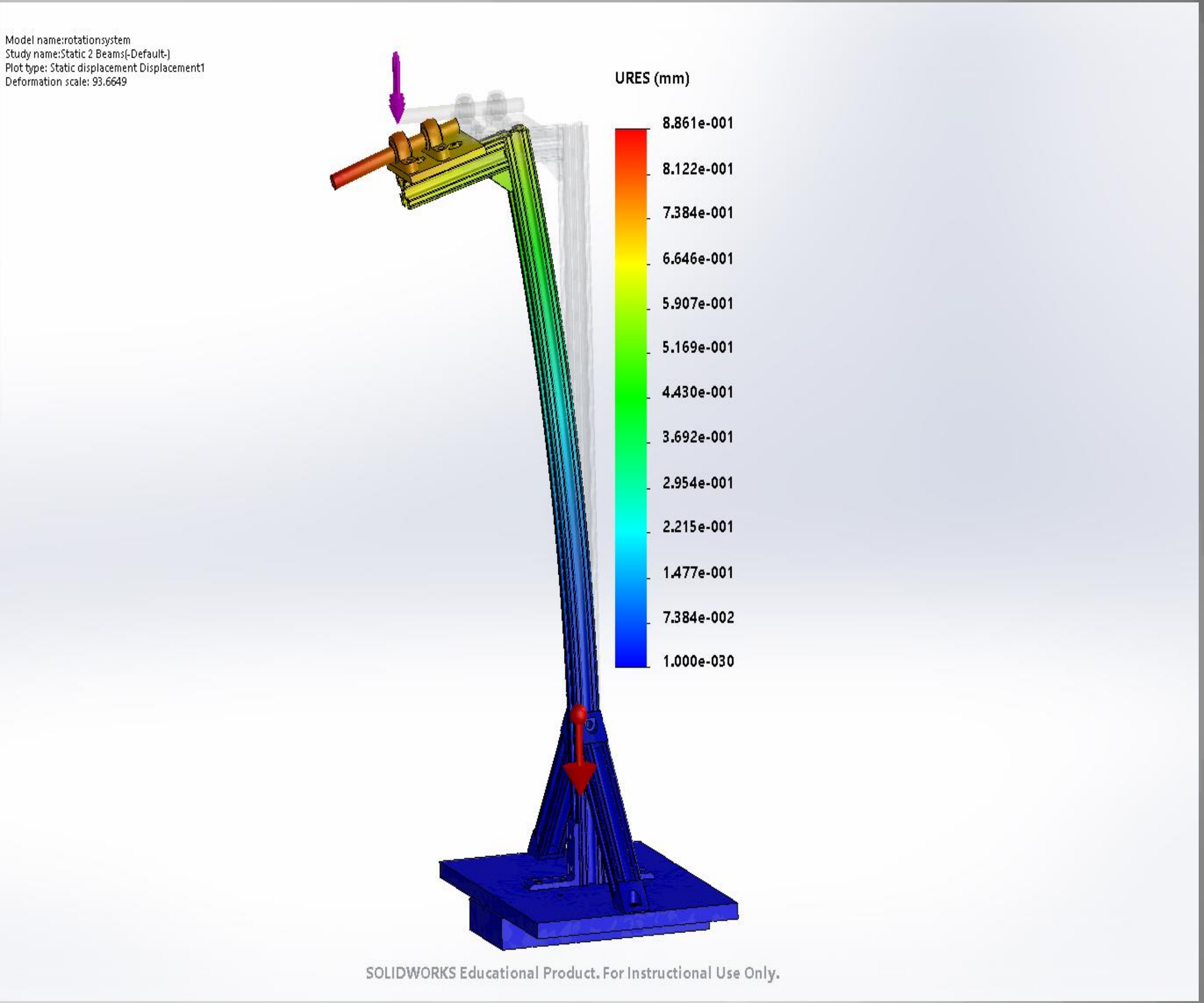


Rotation System - Simulation

Solidworks Model Deflection

Structural Deflection: .8861 mm

Resulting angular offset:
Center Mounted: 1.015 °
Externally Mounted: .5077 °



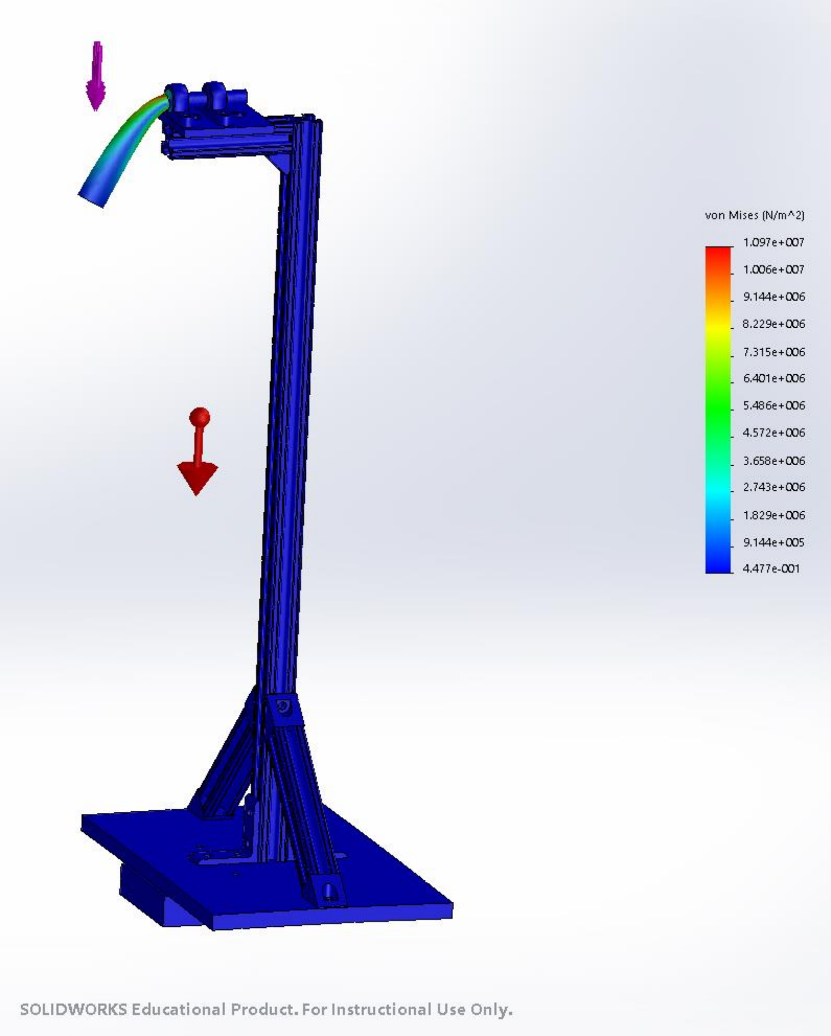
Rotation System - Simulation

Solidworks Model Stress

Maximum Stress: 1.1e+007 $\frac{N}{m^2}$

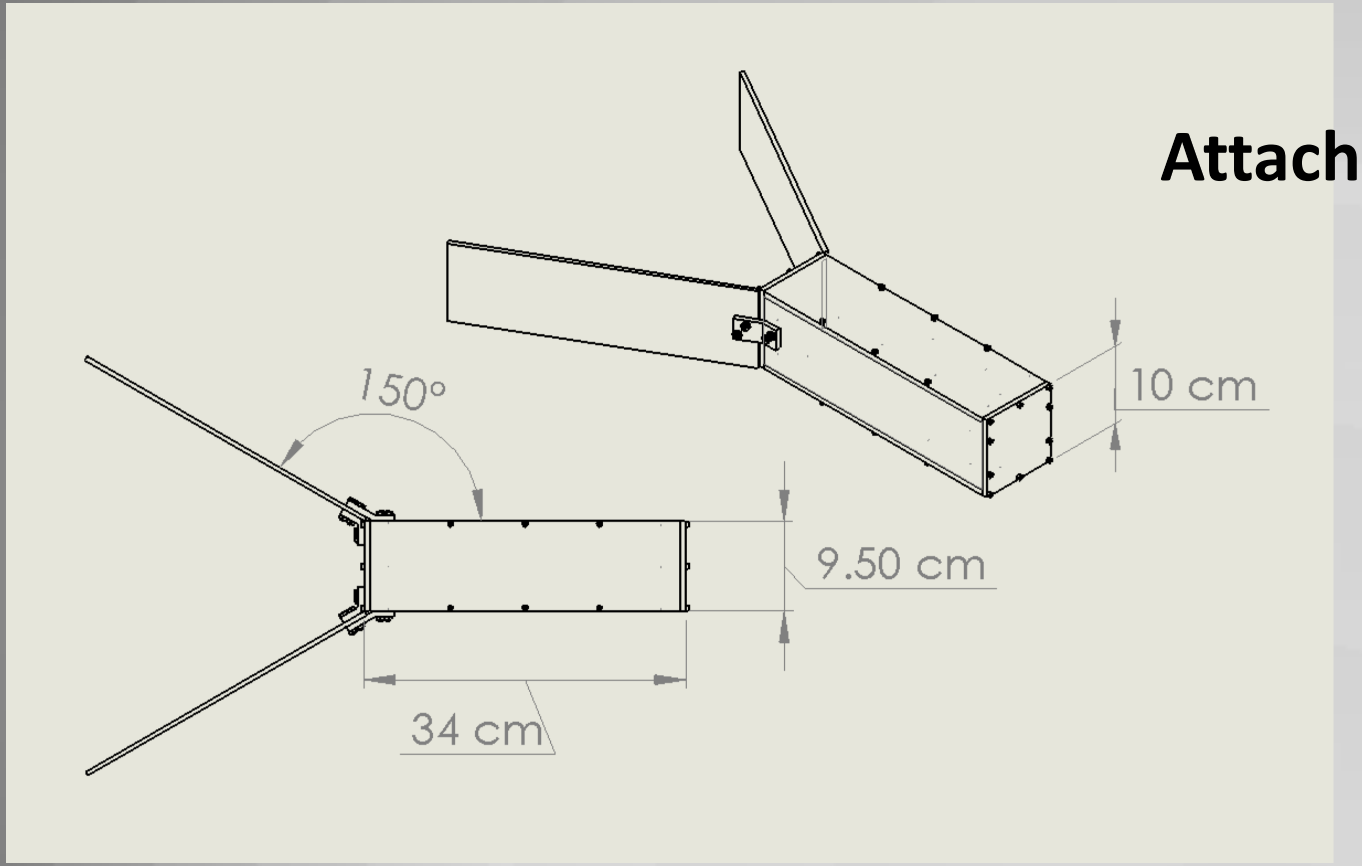
Aluminum yield strength: 5.515e+007 $\frac{N}{m^2}$

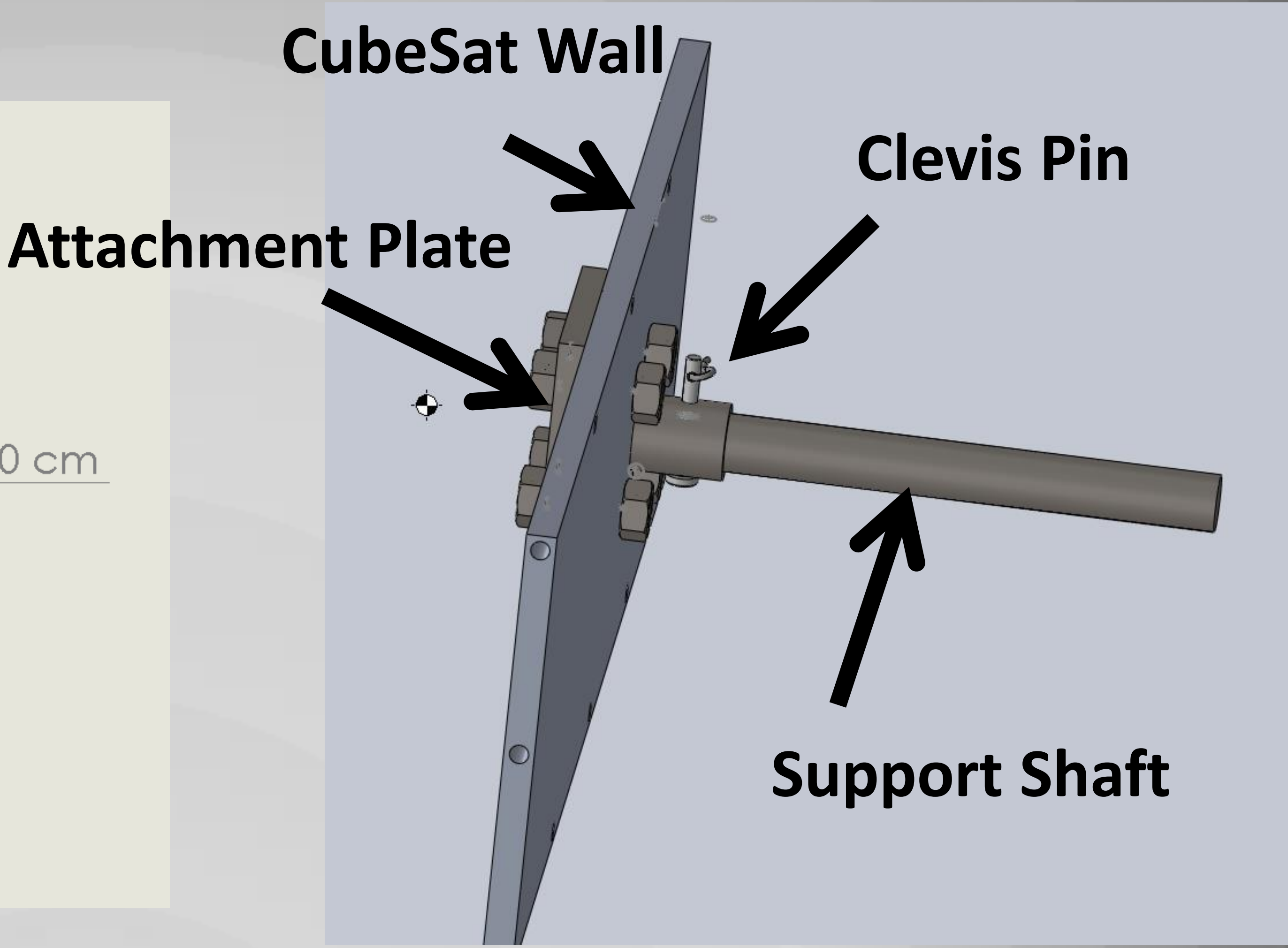
Steel yield strength: 4.2e+008 $\frac{N}{m^2}$



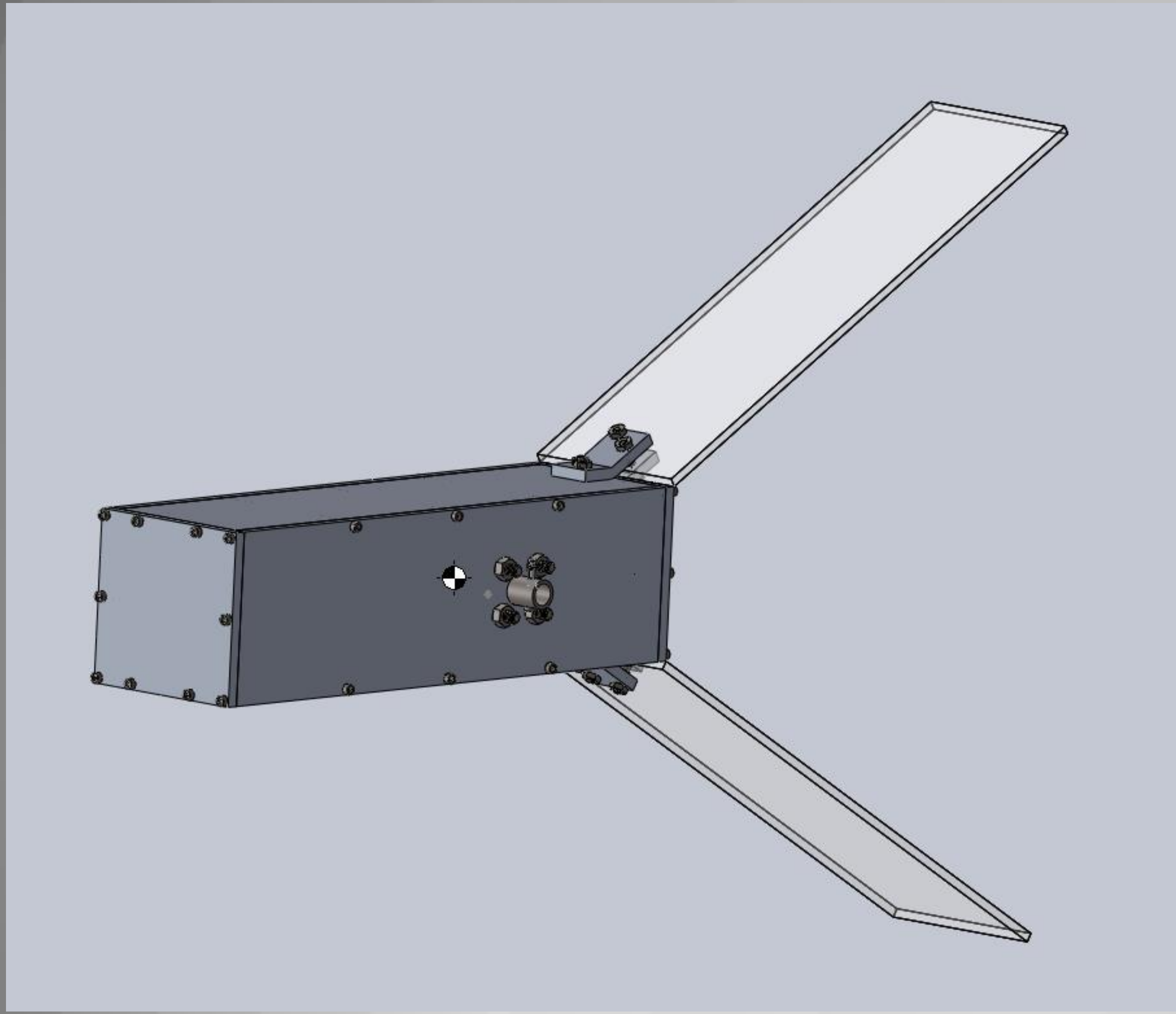
CubeSat Integration

CubeSat Dimensions:





CubeSat Design



- ¼” Aluminum Walls
- Acrylic ‘Solar Panels’
- Countersunk screw fasteners
- Total weight 3.2 kg – additional weight allotted to ballast
- Steel interface plate along CG

Torque Requirements

$$\tau_{required} = \tau_{offset} + \tau_{friction} + \tau_{acceleration}$$

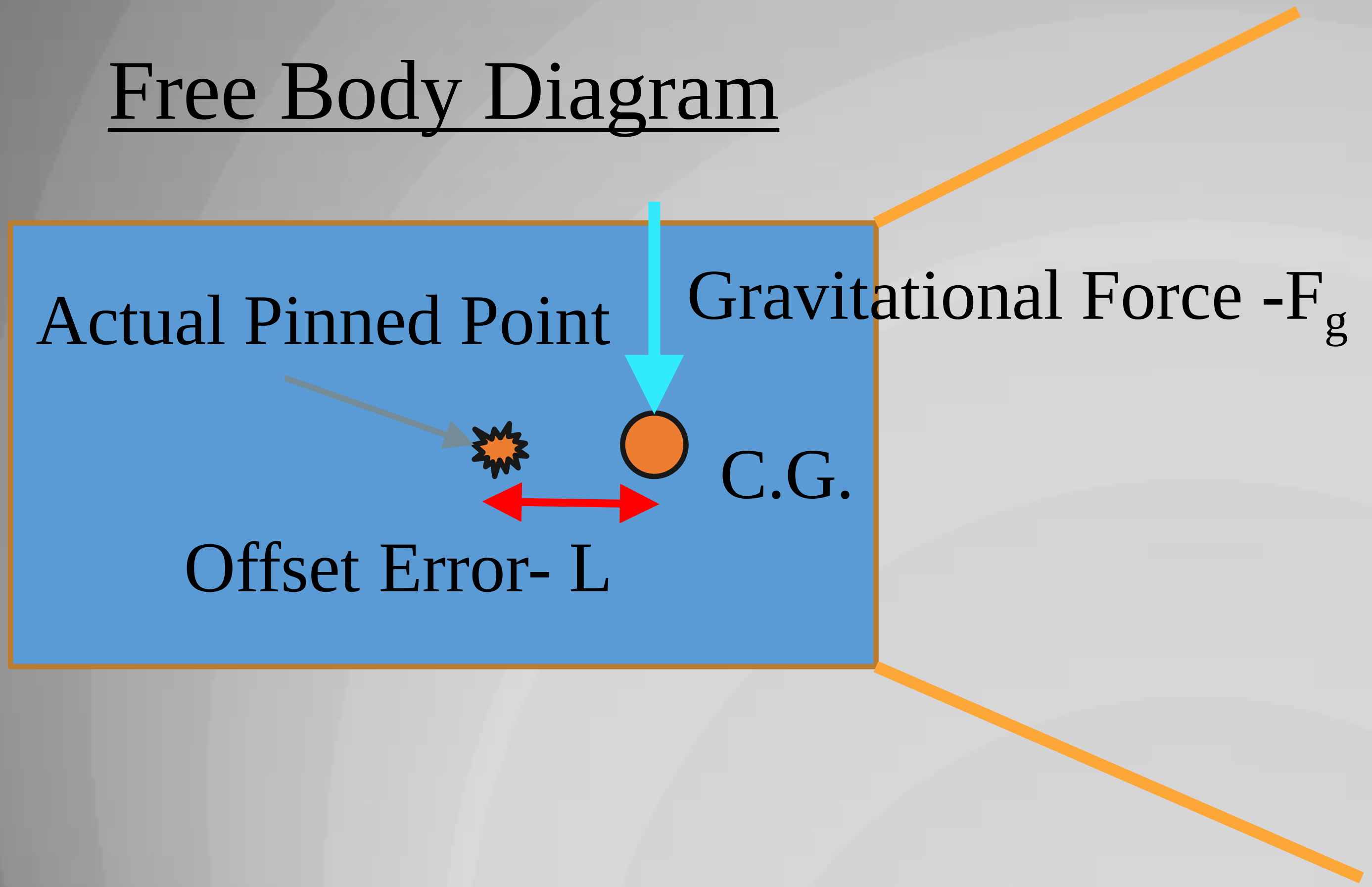
τ_{offset} : Torque due to gravitational force

$\tau_{friction}$: Torque due to motor and bearing friction

$\tau_{acceleration}$: Torque for angular acceleration

Torque From Offset

Free Body Diagram



$$\tau = F_g \delta L$$

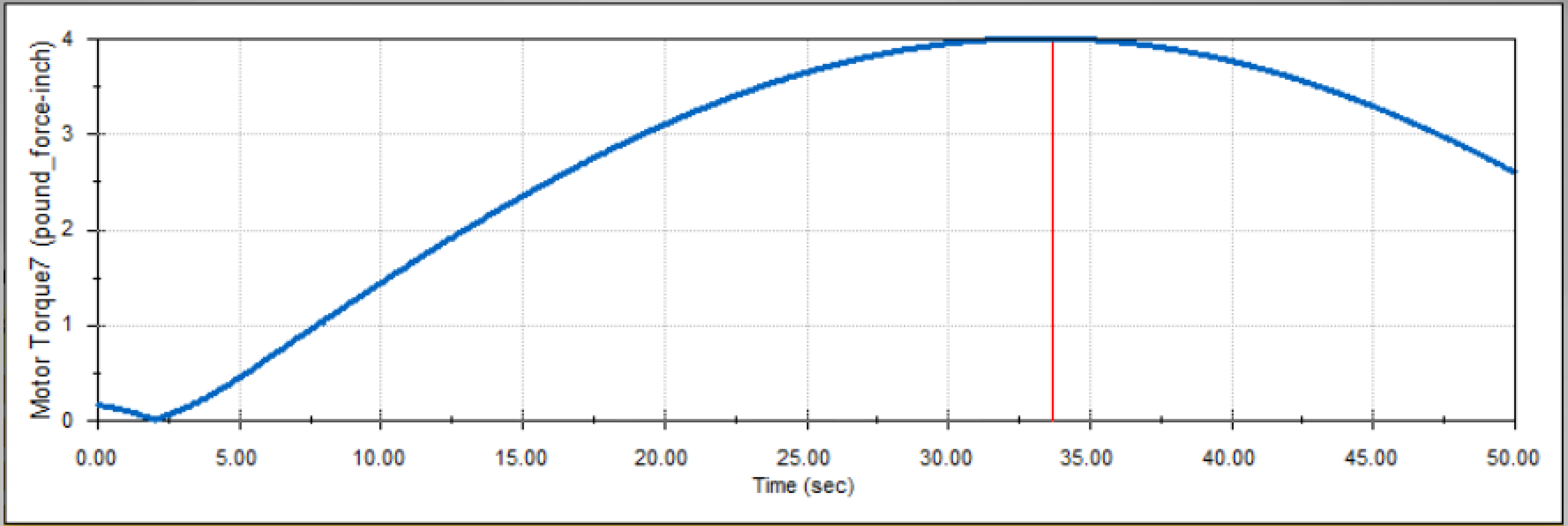
$$= (mass)(9.81)\delta L$$

Frictional & Acceleration Torque

$$\tau_{friction} = \frac{1}{2}\mu_f F_g d \qquad \tau_{acceleration} = I\alpha_{desired}$$

Approximations	
Offset Torque (0.5 in error)	52.1788 oz-in
Frictional Torque	10.4191 oz-in
Acceleration Torque (3 deg/s^2)	0.6073 oz-in
<u>Total</u>	63.2052 oz-in

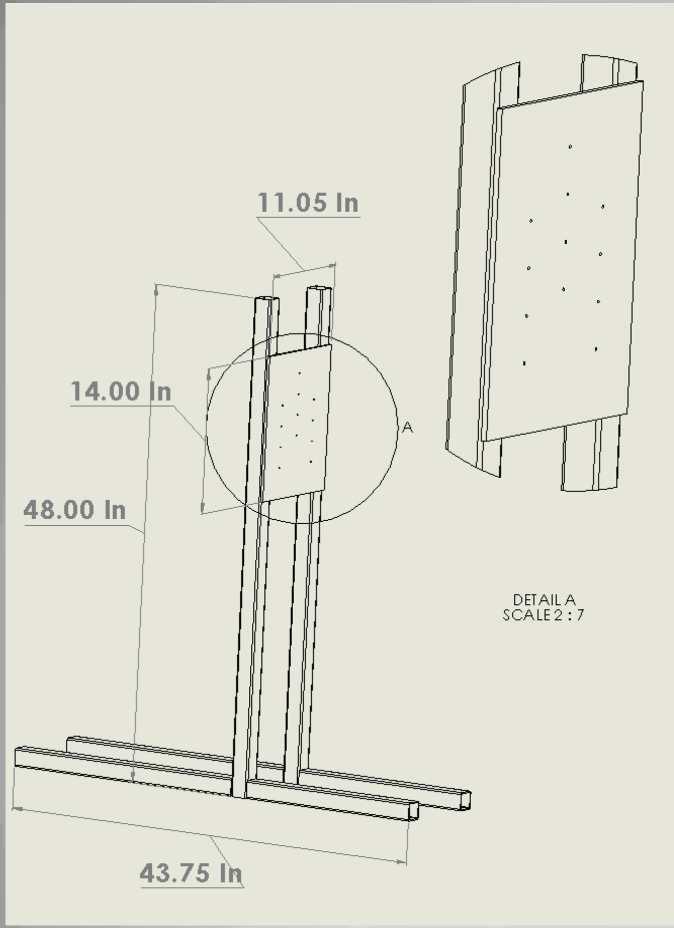
Solidworks Simulation - Torque



Maximum calculated torque from Solidworks: 4 lbf-in (64 oz-in) of torque
Within 2% of analytic solution

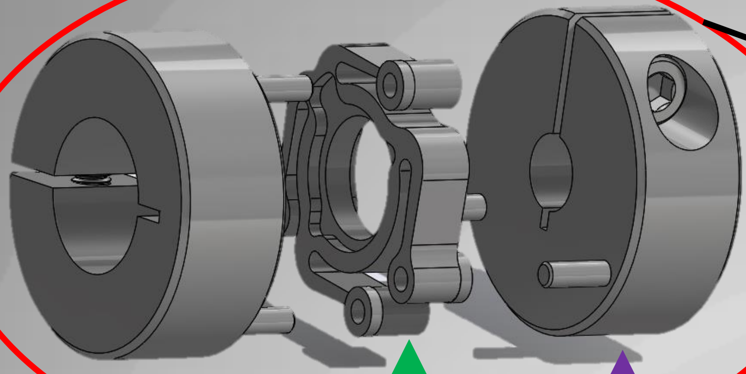
Capture Device Support Structure

Support Dimensions:

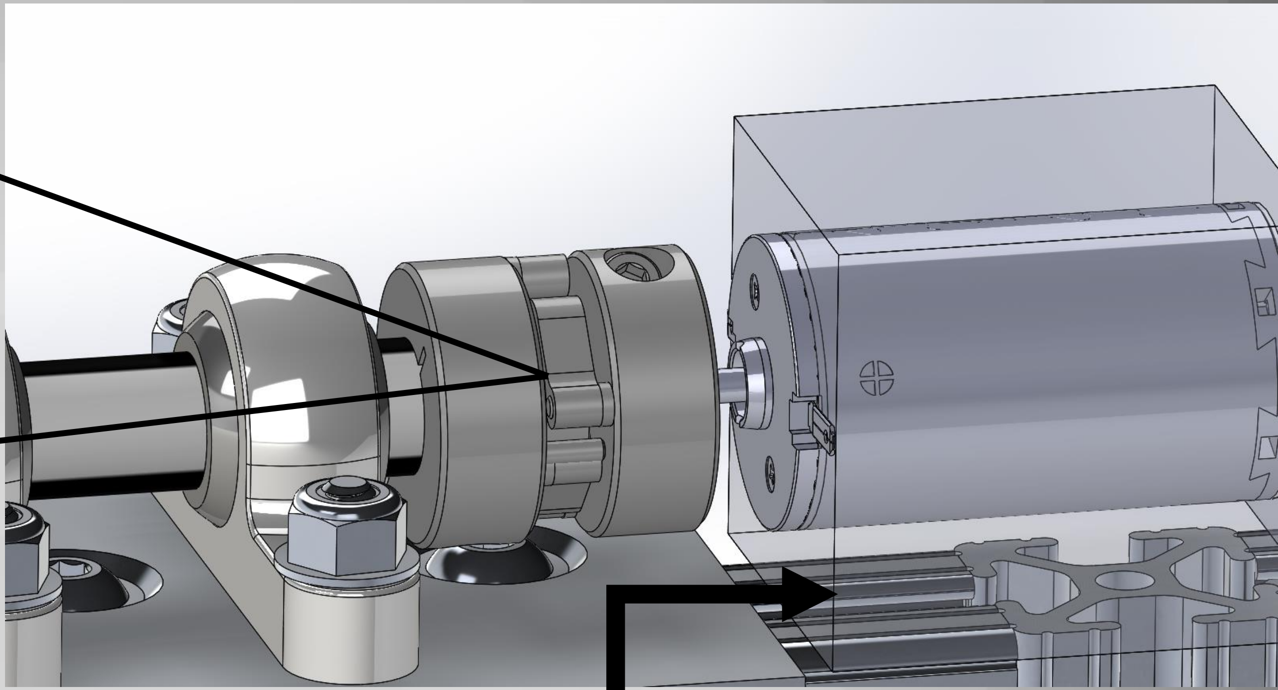


- Holes in linear series provide positions for displacement of capture device in 4 increments from 0 to 6 inches
- 4 legs weighted down instead of fixed to allow versatility in positioning

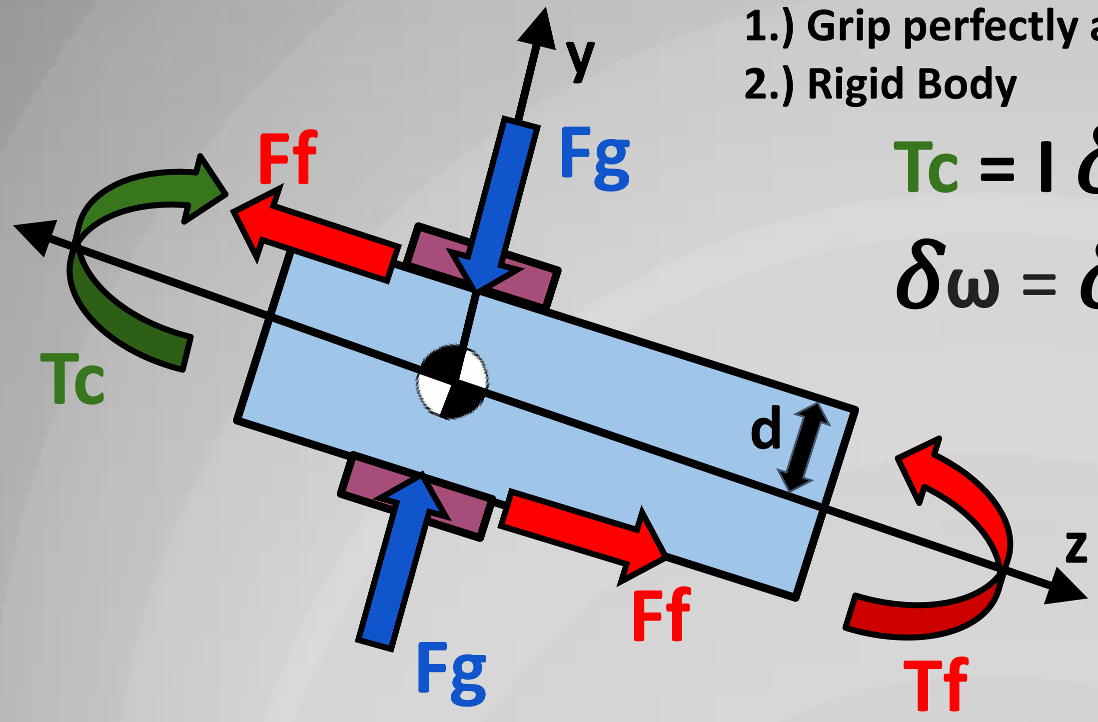
Rotation System - Motor Coupling



- Coupling Disc
- 6 mm Shaft Coupling
- 12 mm Shaft Coupling
- Motor Mounting Block



Minimum Grip Force Derivation



Assumptions:

1.) Grip perfectly at C.O.G
2.) Rigid Body
3.) Perfect Contact
4.) CubeSat Motion Stopped in δt

$$Tc = I\,\delta\alpha \qquad Tc = I\,\delta\omega/\delta t$$

$$\delta\omega = \delta\alpha\,\delta t$$

$$Ff = \mu Fg$$

$$\rightarrow Tf = 2\mu Fg d$$

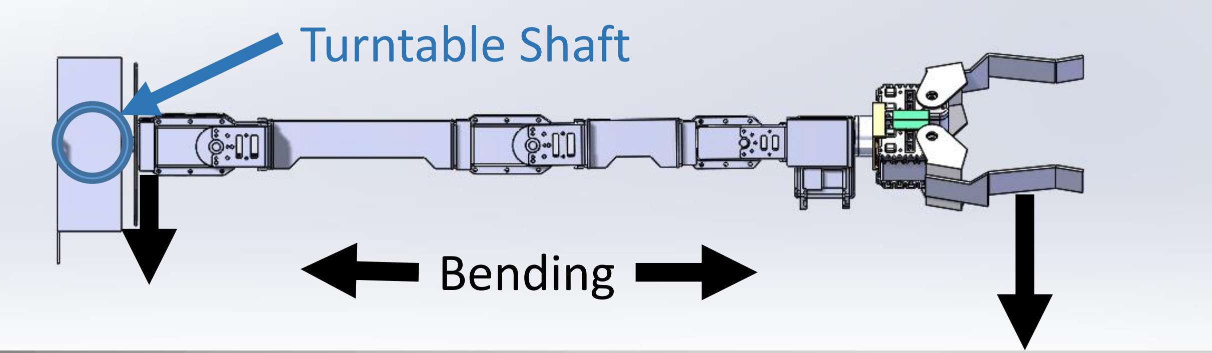
$$\rightarrow Tf = Tc$$

$$\rightarrow \boxed{Fg = \frac{I\,\delta\omega}{2\mu d\delta t}}$$

3D Printer Specs

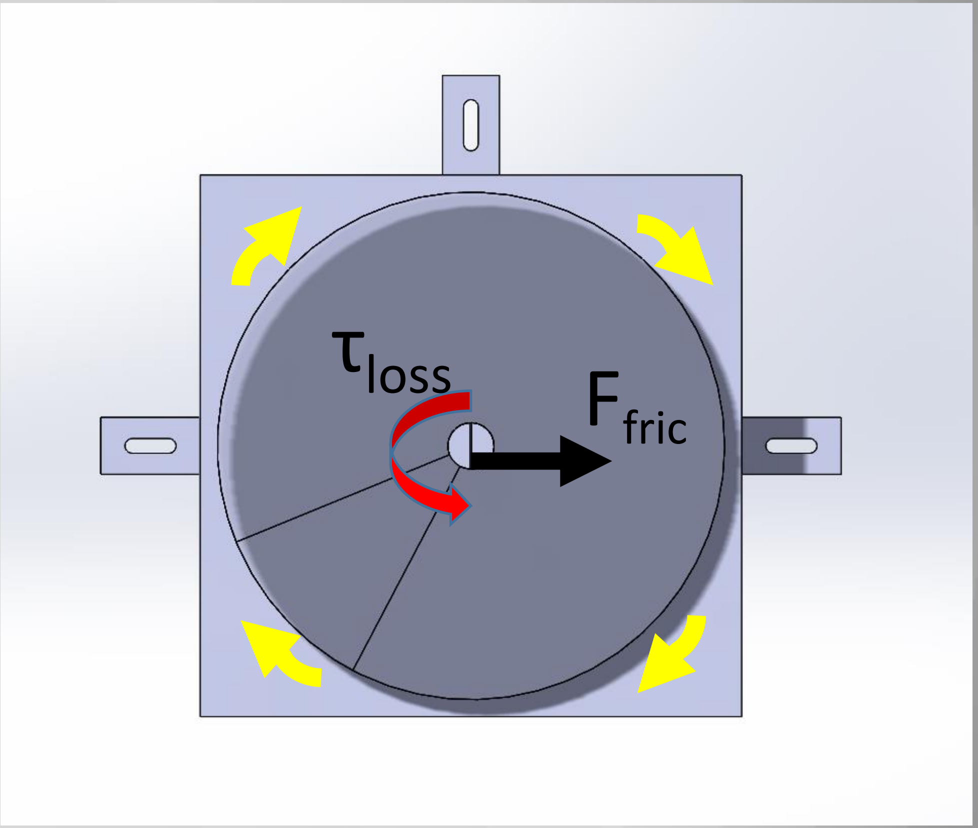
- Objet30 Pro print with VeroWhite material
- Location: ITLL, CU Boulder
- 0.1 mm (0.0039 in.) accuracy, geometry dependent
- Ball Joint: ~$10/print

Bearing Friction



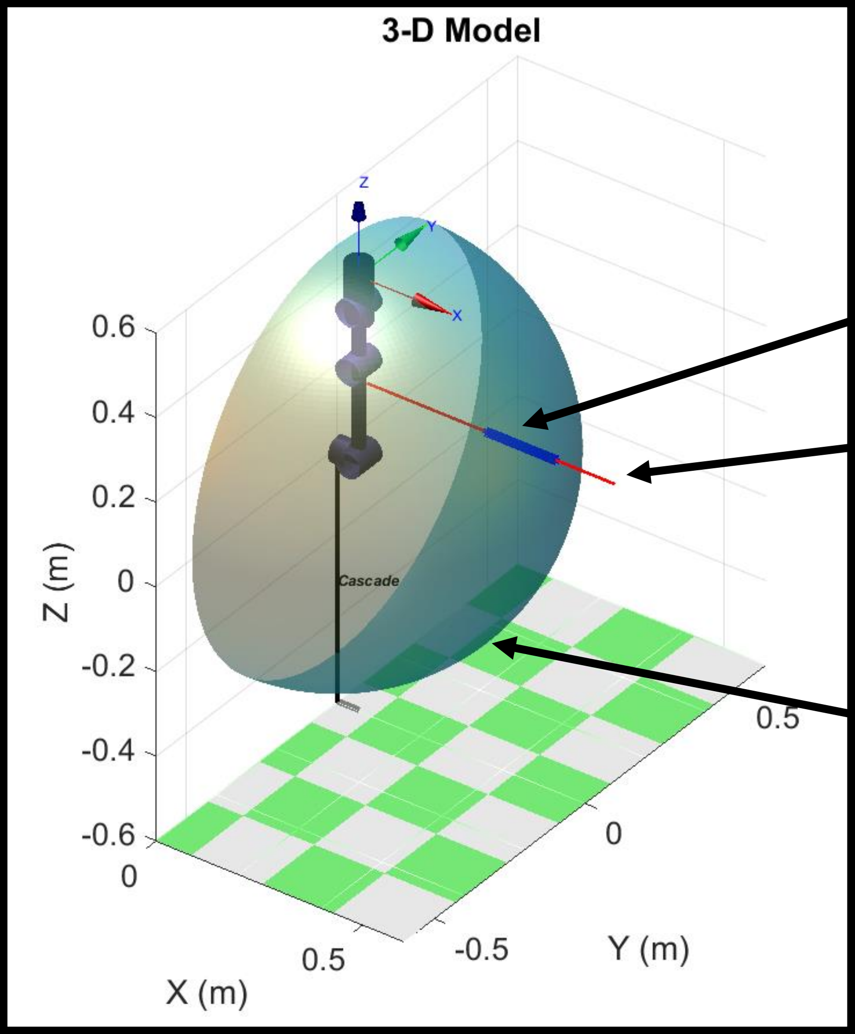
$$\tau_{loss} = R \times F = R_{shaft}F_{fric}$$

- Weight of the arm induces bending at the support -> turntable shaft
- Decreased servo performance
- Friction force is SMALL
  - Shaft supported by 4 bearings
  - Unable to quantify torque
- Feedback from servos to confirm the desired angle is achieved



Arm Limits

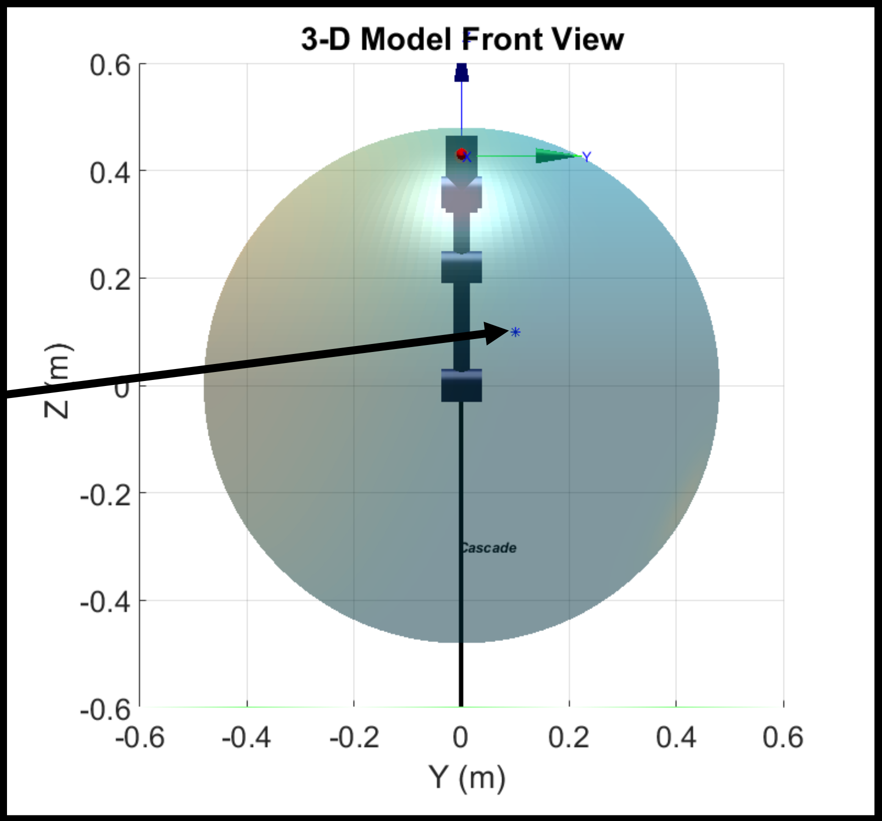
Example Case



Allowable Reach on Spin Axis

Spin Axis

Sphere of Reach



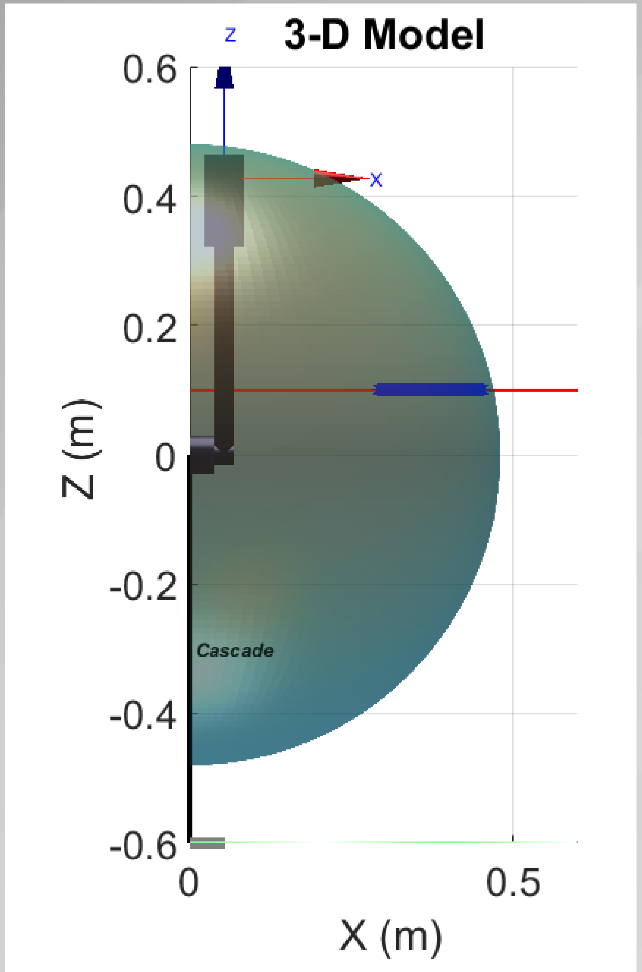
Spin Axis Assumption

Horizontal Offset : 0.1m Vertical Offset : 0.1m

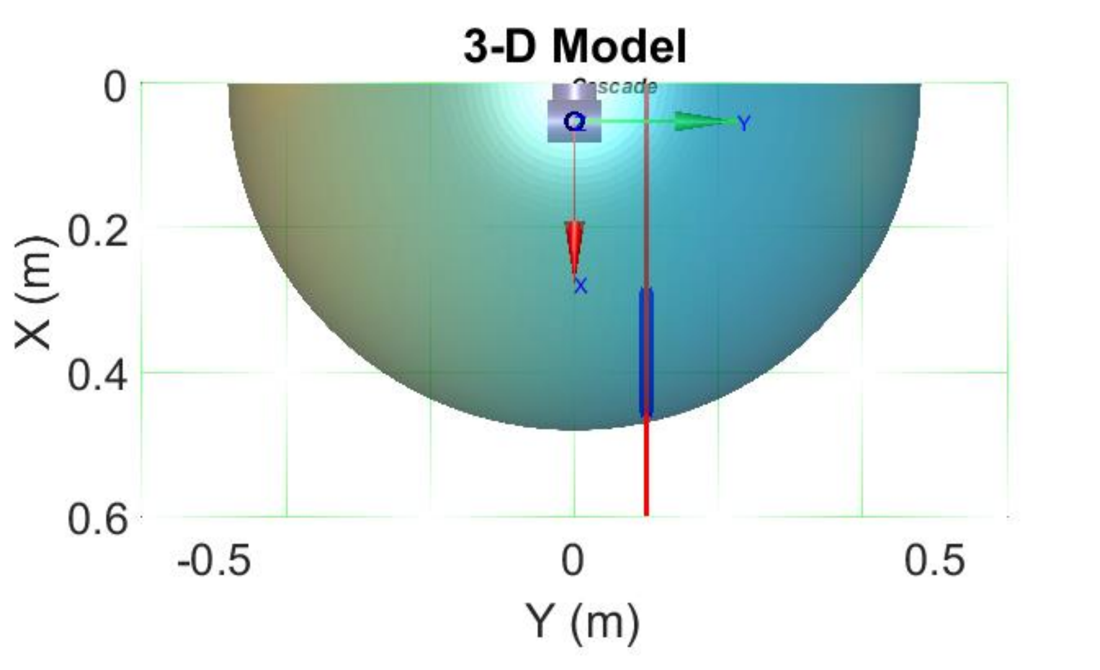
Arm Limits

Example Case Backup

Side View

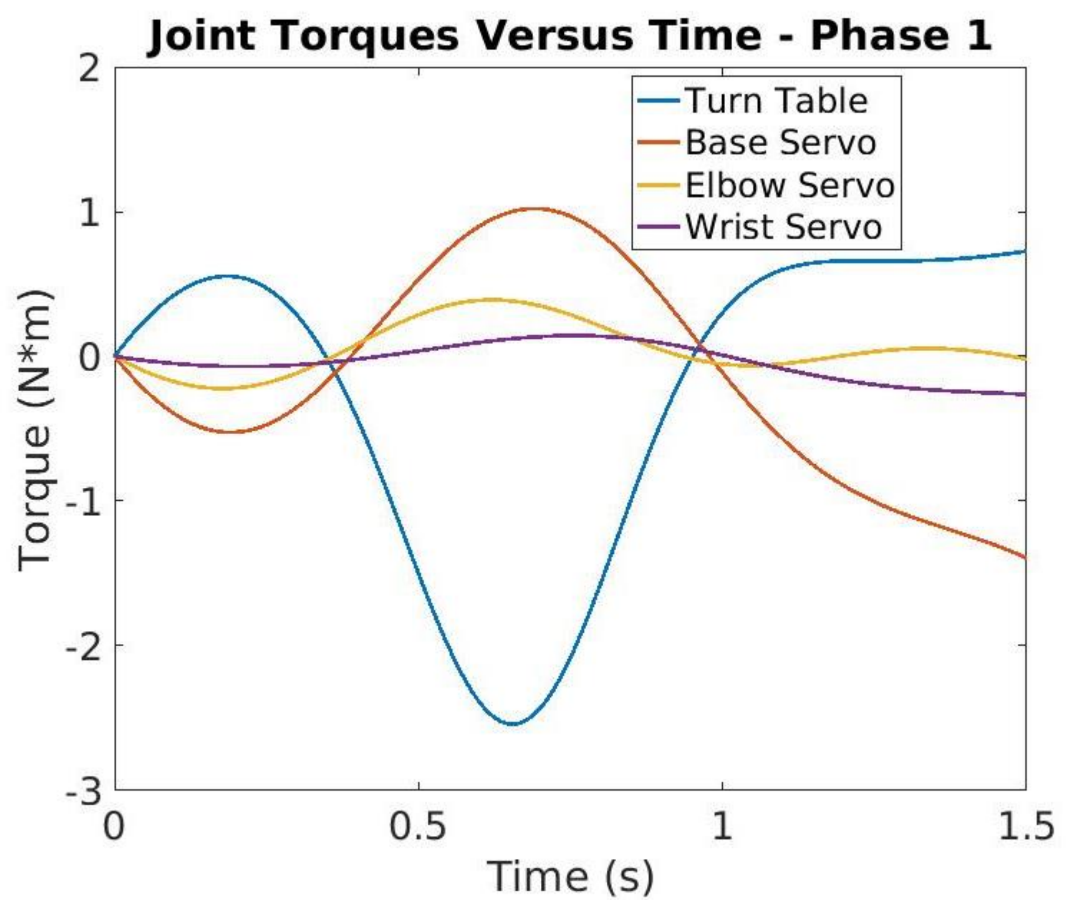


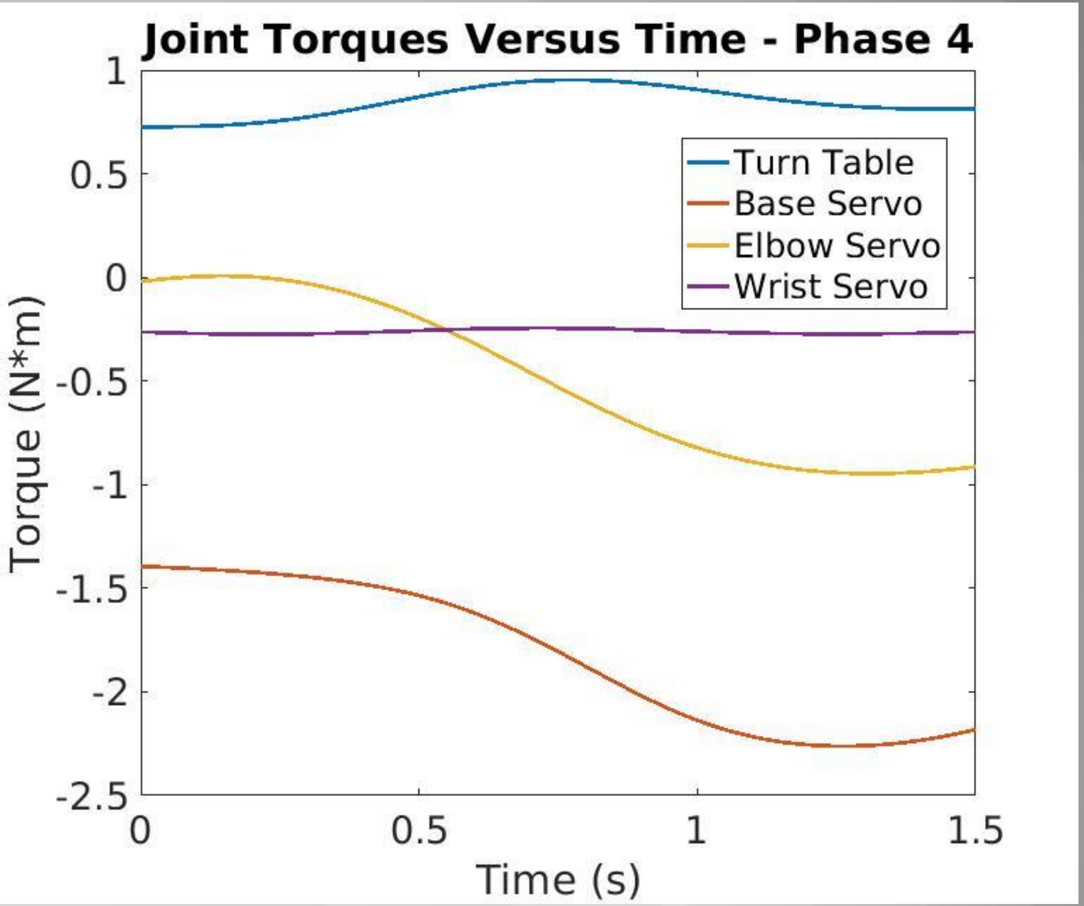
Top View



Arm Limits

Servo Performance -backup





Arm Limits

Peak & Avg Power Feasibility

Phase 1

	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	17.23 W	15.66 W	5.94 W	3.85 W	0	42.68 W
6 s	11.93 W	16.25 W	9.45 W	3.85 W	0	41.48 W

Phase 4

	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	17.23 W	15.66 W	5.94 W	3.85 W	0	42.68 W
6 s	11.93 W	16.25 W	9.45 W	3.85 W	0	41.48 W

DR 1.4.1: The capture device shall have an average power of no more than 100 W

DR 1.4.2: The capture device shall have a peak power of no more than 168 W

Arm Limits

Peak Current Draw Feasibility

Phase 1

	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	0.67 A	3.79 A	1.57 A	0.44 A	0	6.47 A
6 s	0.51 A	3.09 A	1.3 A	0.38 A	0	5.29 A

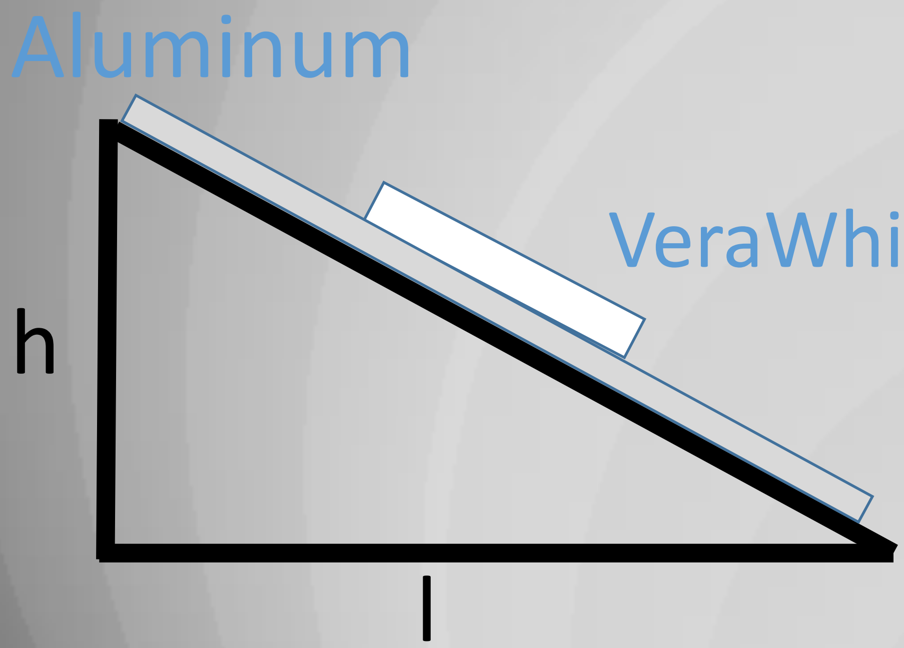
Phase 4

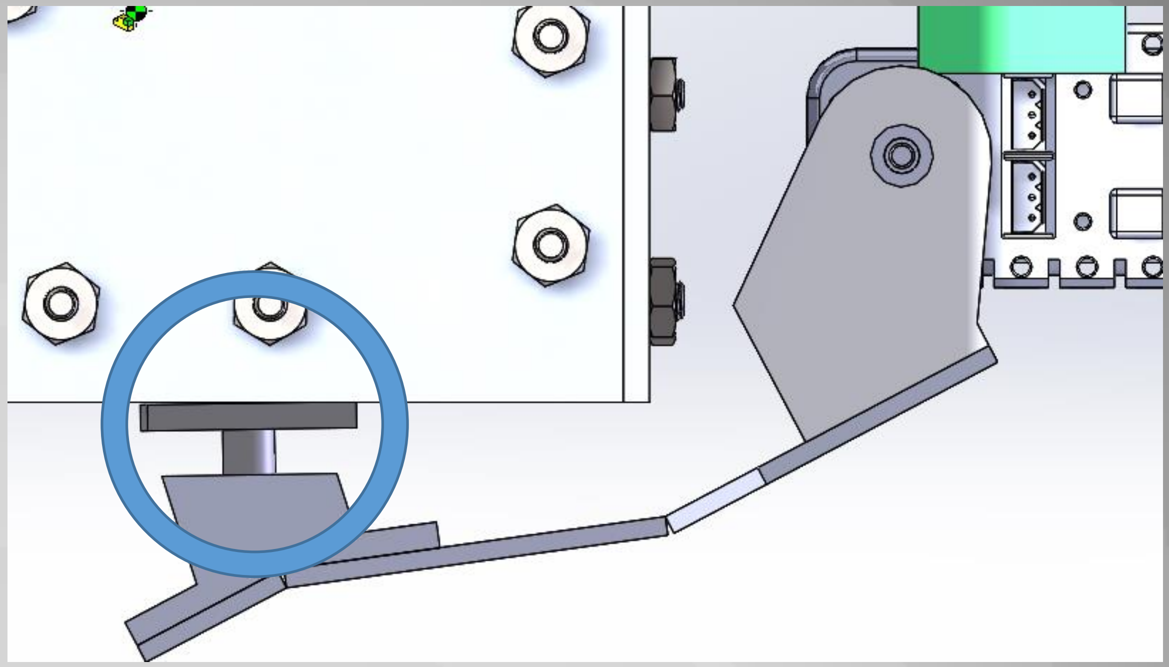
	Turn Table Servo	Base Servo	Elbow Servo	Wrist Servo	Wrist Rotator	Total
1.5 s	0.5 A	1.34 A	1.02 A	0.21 A	0	3.06 A
6 s	0.5 A	1.27 A	0.98 A	0.21 A	0	2.96 A

DR 1.4.3: The capture device shall have a peak current draw of no more than 10 A

Gripper Friction

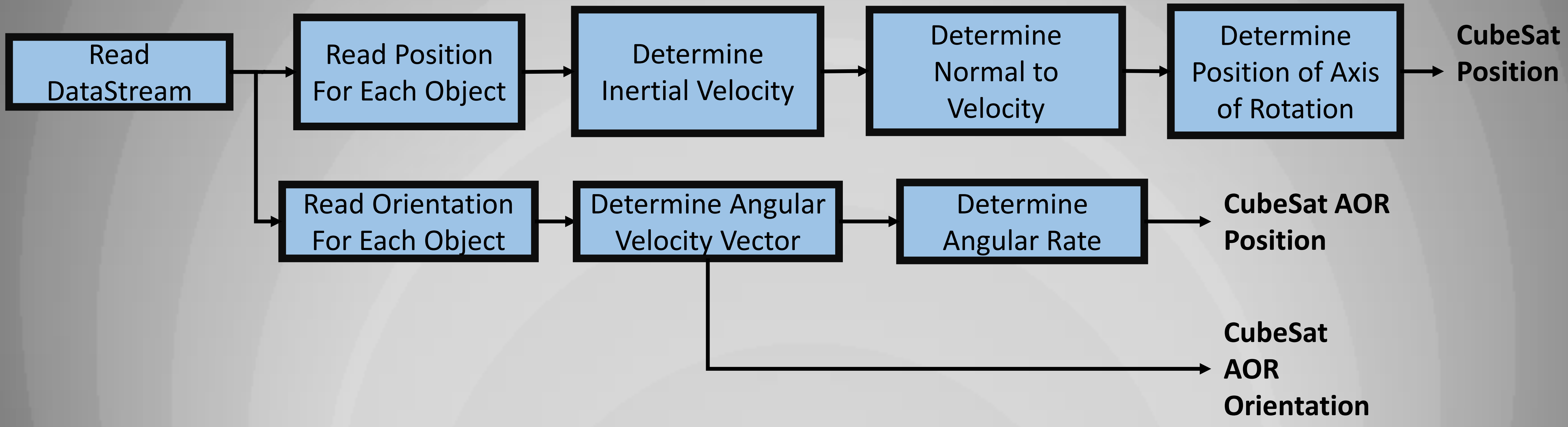
- Experimentally Determine – Ramp Test



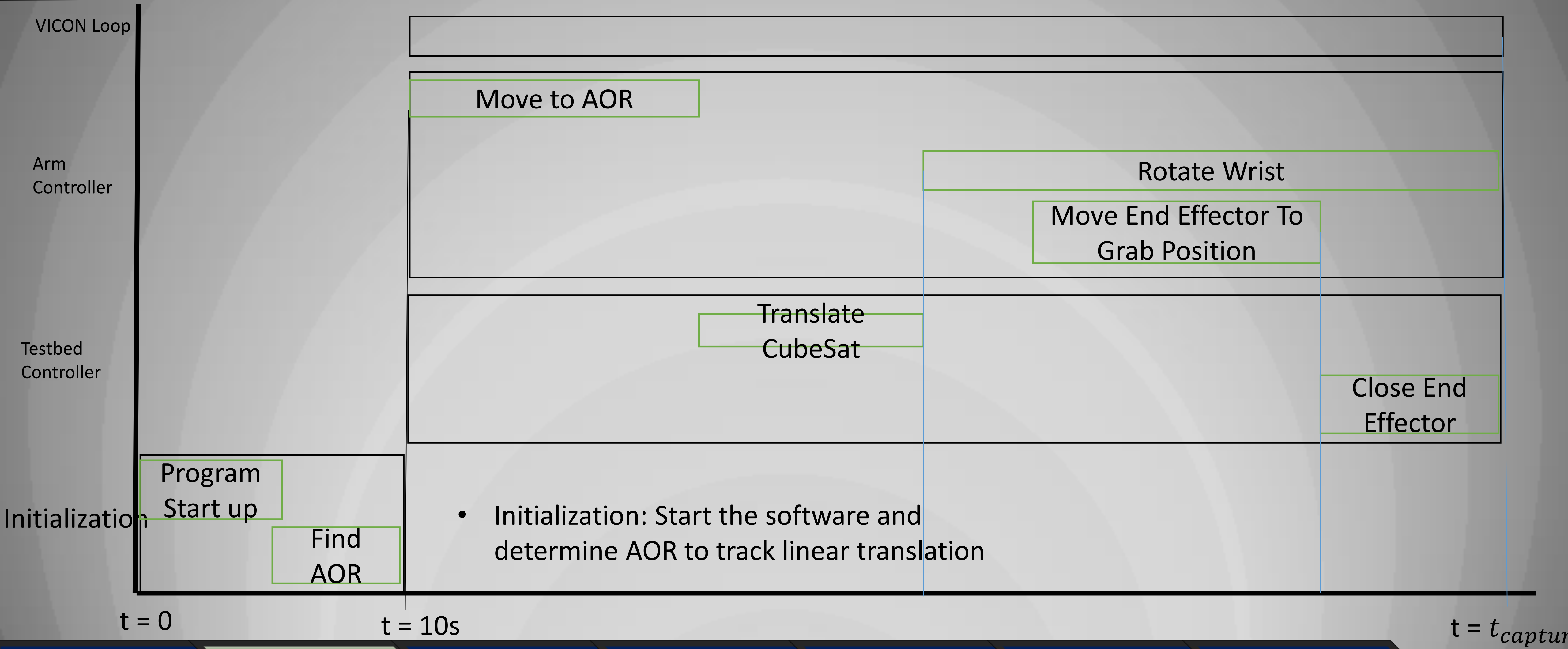


- Adjust height until plate slips -> $F_r = h/l$
- $F_r = \mu_s N$

Software: VICON

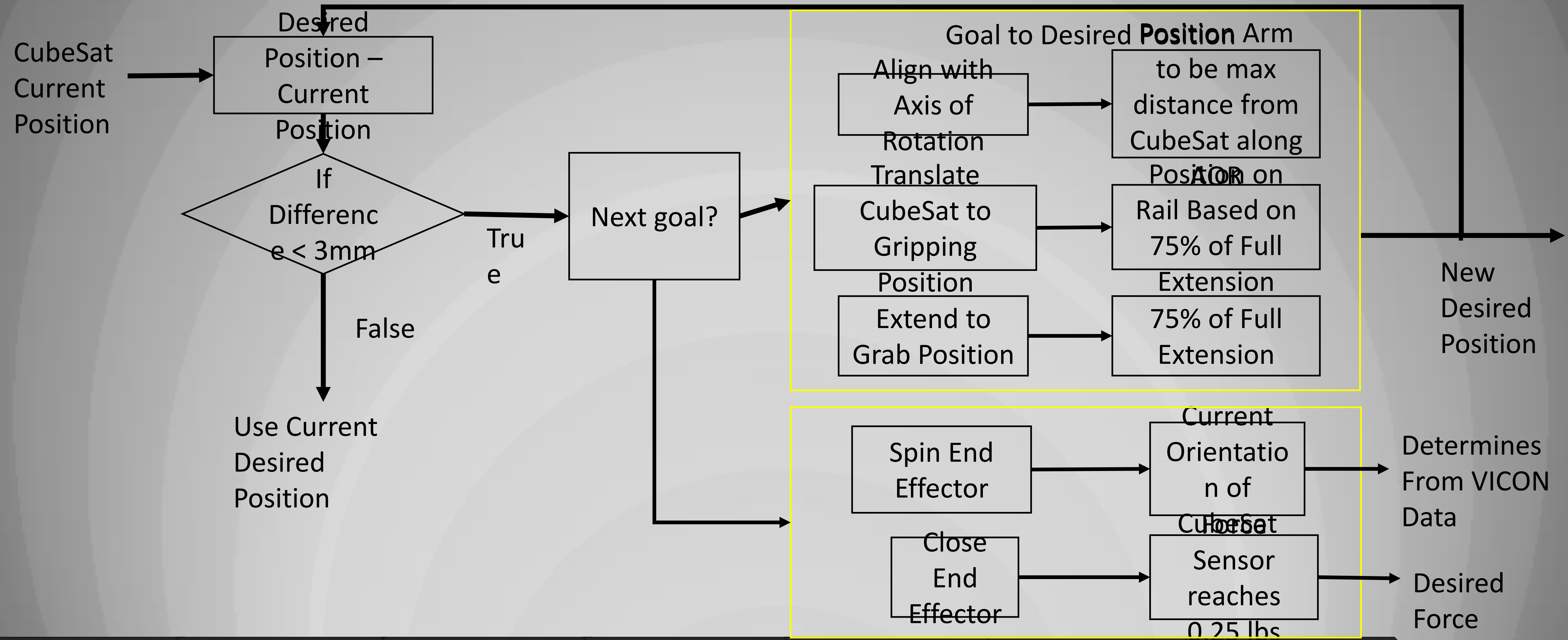


Software: Time Scheme



- Initialization: Start the software and determine AOR to track linear translation

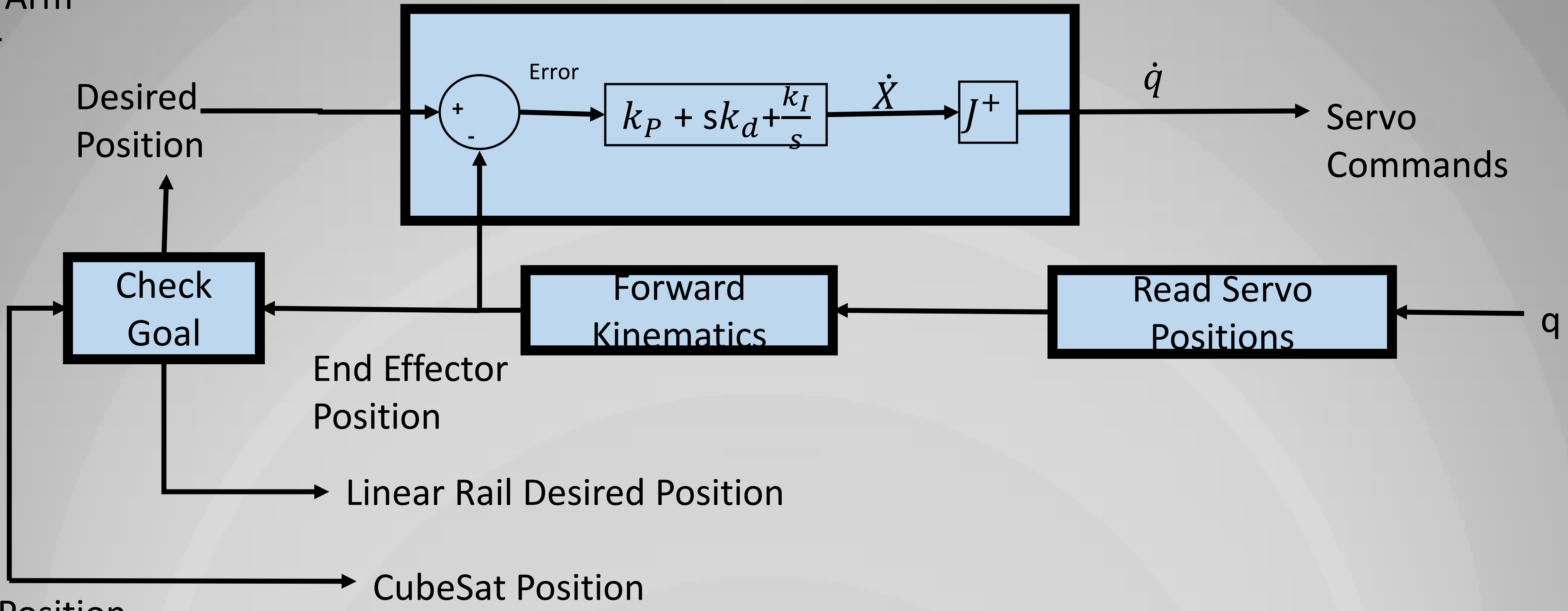
Goal



CubeSat Current Position
Desired Position – Current Position
If Difference < 3mm
True
False
Use Current Desired Position
Next goal?
Goal to Desired Position
Align with Axis of Rotation
Position Arm to be max distance from CubeSat along AOR
Translate CubeSat to Gripping Position
Position on Rail Based on 75% of Full Extension
Extend to Grab Position
75% of Full Extension
New Desired Position
Spin End Effector
Current Orientation of CubeSat
Determines From VICON Data
Close End Effector
Force Sensor reaches 0.25 lbs
Desired Force

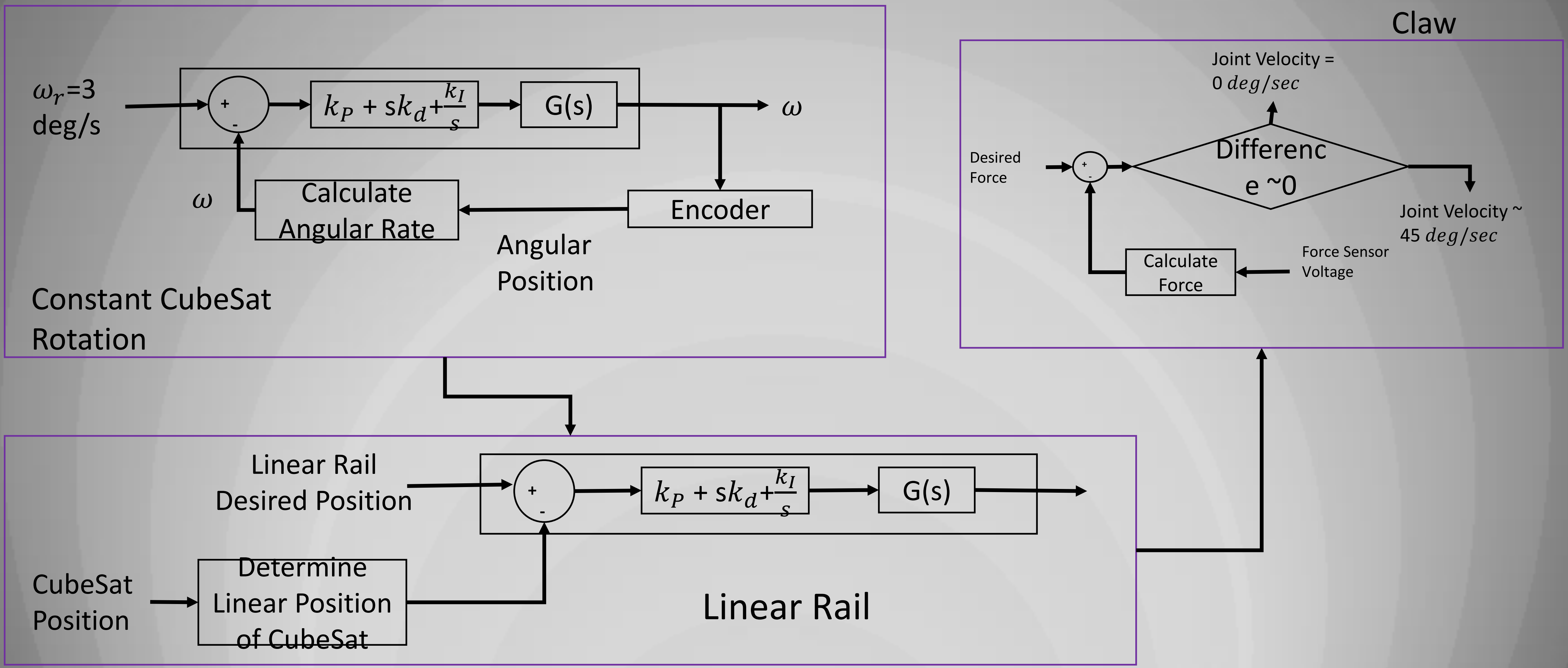
Software: Arm

LabVIEW: Arm Controller



CubeSat AOR Position, Orientation + CubeSat Position

TestBed Controllers



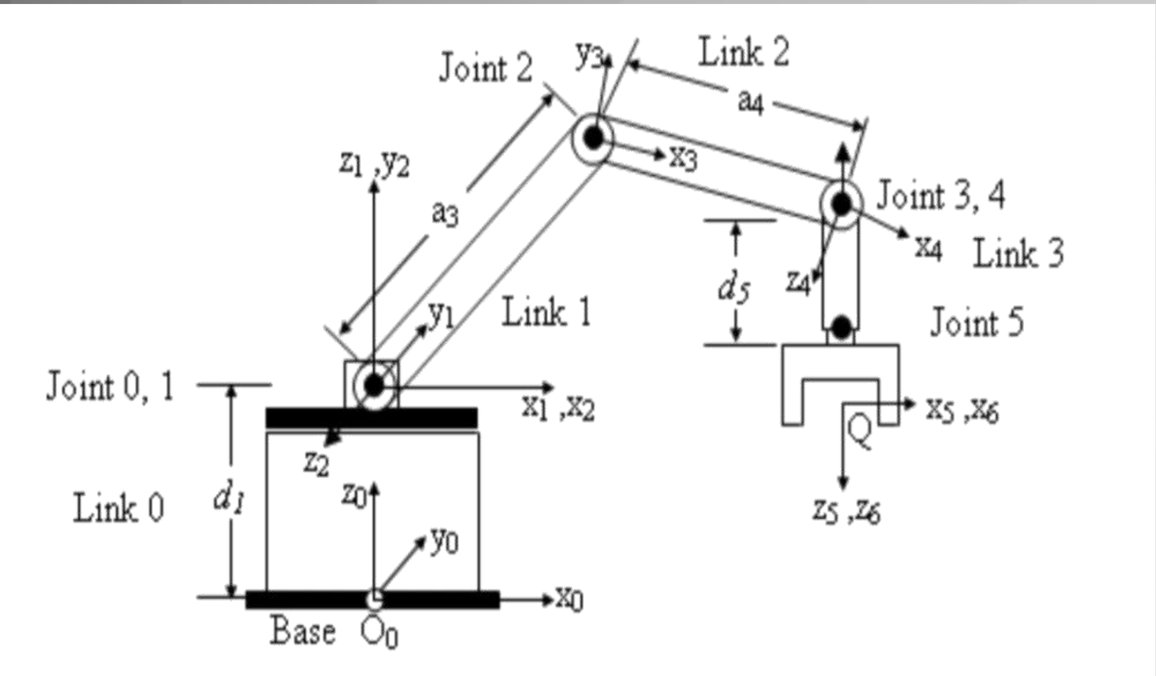
Jacobian Inverse Techniques

Because the Jacobian is a 6x5 matrix, its inverse must be approximated in order to control the robotic arm

1. Jacobian Transpose Method
   - Assume $J^{-1} \approx J^T$
   - This method has been used for inverse kinematics by others in the past
   - It is fast to calculate and avoids singularity issues but can lead to steady state error
2. Pseudo Inverse Method
   - Assume $J^{-1} \approx J^T(JJ^T)^{-1}$
   - Produces the optimal joint angle trajectories in a least squares sense
   - When two or more joints line on one of the principal axes, J is not full rank, which causes J^{-1} and consequently the joint angular velocities to blow up to infinity
3. Damped Least Squares Method
   - Assume $J^{-1} \approx J^T(JJ^T + \lambda^2 I)^{-1}$, where λ is system dependent and is typically determined experimentally
   - This method is a way of modifying the pseudo inverse method so that near singularities, the joint angular velocities are driven closer to zero, rather than to infinity
   - This can have issues with local minima and is slightly more computationally intensive

Denavit-Hartenberg parameters

Joint angle	θ_j	The angle between the x_{j-1} and x_j axes about the z_{j-1} axis	Revolute joint variable
Link offset	d_j	The distance from the origin of frame $j-1$ to the x_j axis along the z_{j-1} axis	Prismatic joint variable
Link Length	a_j	The distance between the z_{j-1} and z_j axes along the x_j axis	constant
Link Twist	α_j	The angle from the z_{j-1} axis to the z_j axis about the x_j axis	constant
Joint type	σ_j	$\sigma = 0$ for a revolute joint, $\sigma = 1$ for a prismatic joint	constant



$$^{j-1}A_j(\theta_j, d_j, a_j, \alpha_j) = T_{RZ}(\theta_j)(T_z(d_j)T_x(a_j)T_{RX}(\alpha_j))$$

$$\begin{pmatrix} cos\theta_j & -sin\theta_j cos & sin\theta_j sin\alpha & a_j cos \\ sin\theta_j & cos\theta_j cos\alpha_j & -cos\theta_j sin & a_j sin \\ 0 & sin\alpha_j & cos\alpha_j & d_j \end{pmatrix}$$

$$0 \quad 0 \quad 0 \quad 1$$

For a revolute joint, θ_j $\alpha_j = 0$

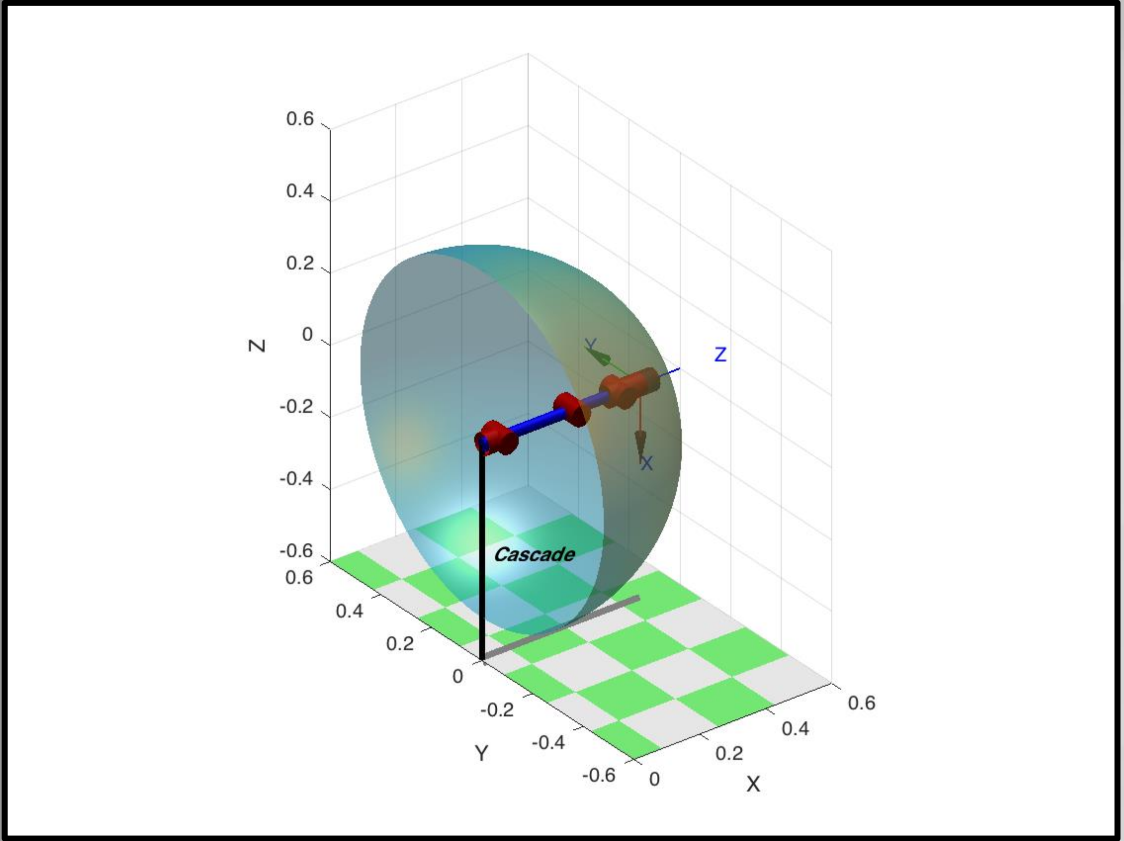
DH Parameters for Cascade Arm

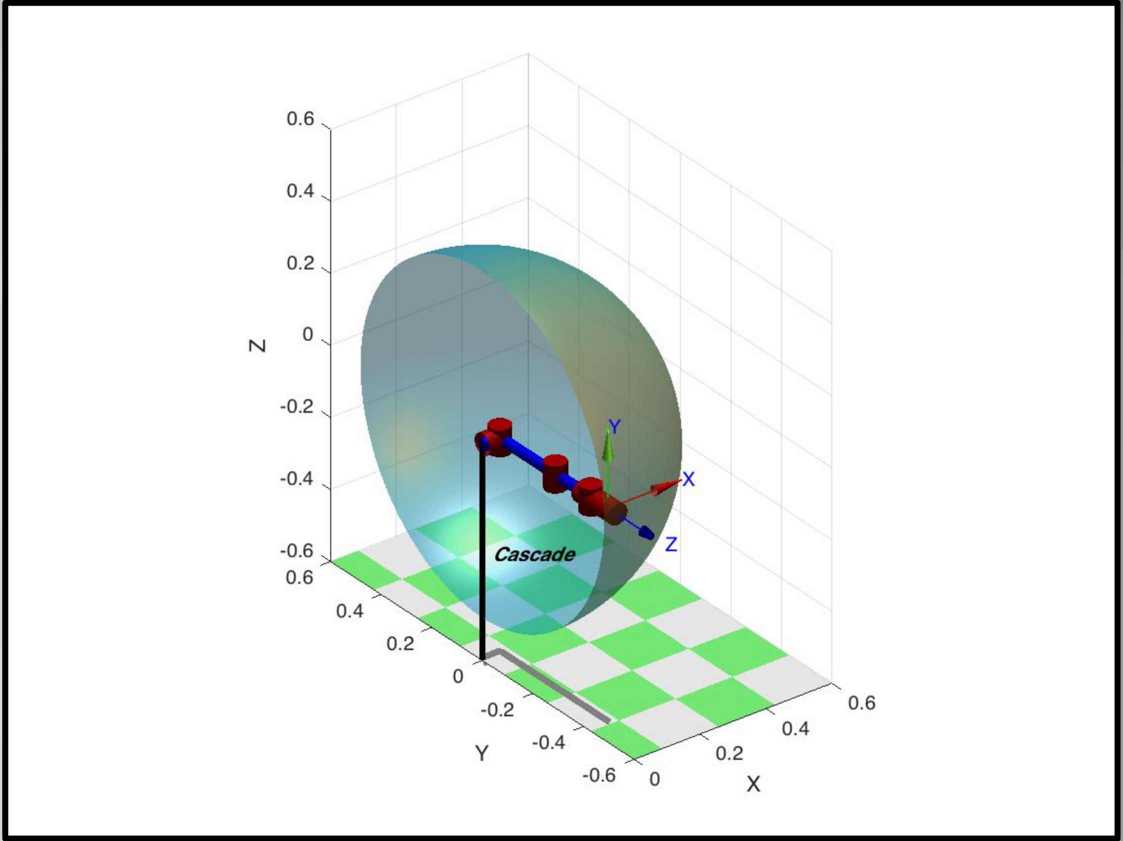
Cascade (6 axis, RRRRRR, stdDH, fastRNE)

j	theta	d	a	alpha	offset
1	q1	0.053	0	1.571	0
2	q2	0	0.22	0	0
3	q3	0	0.14	0	0
4	q4	0	0	-1.571	-1.571
5	q5	0.068	0	0	0
6	q6	0	0	0	0

$$\text{grav} = \begin{matrix} 0 \\ 0 \\ 9.81 \end{matrix} \quad \text{base} = \begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \quad \text{tool} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Gravitational Torque



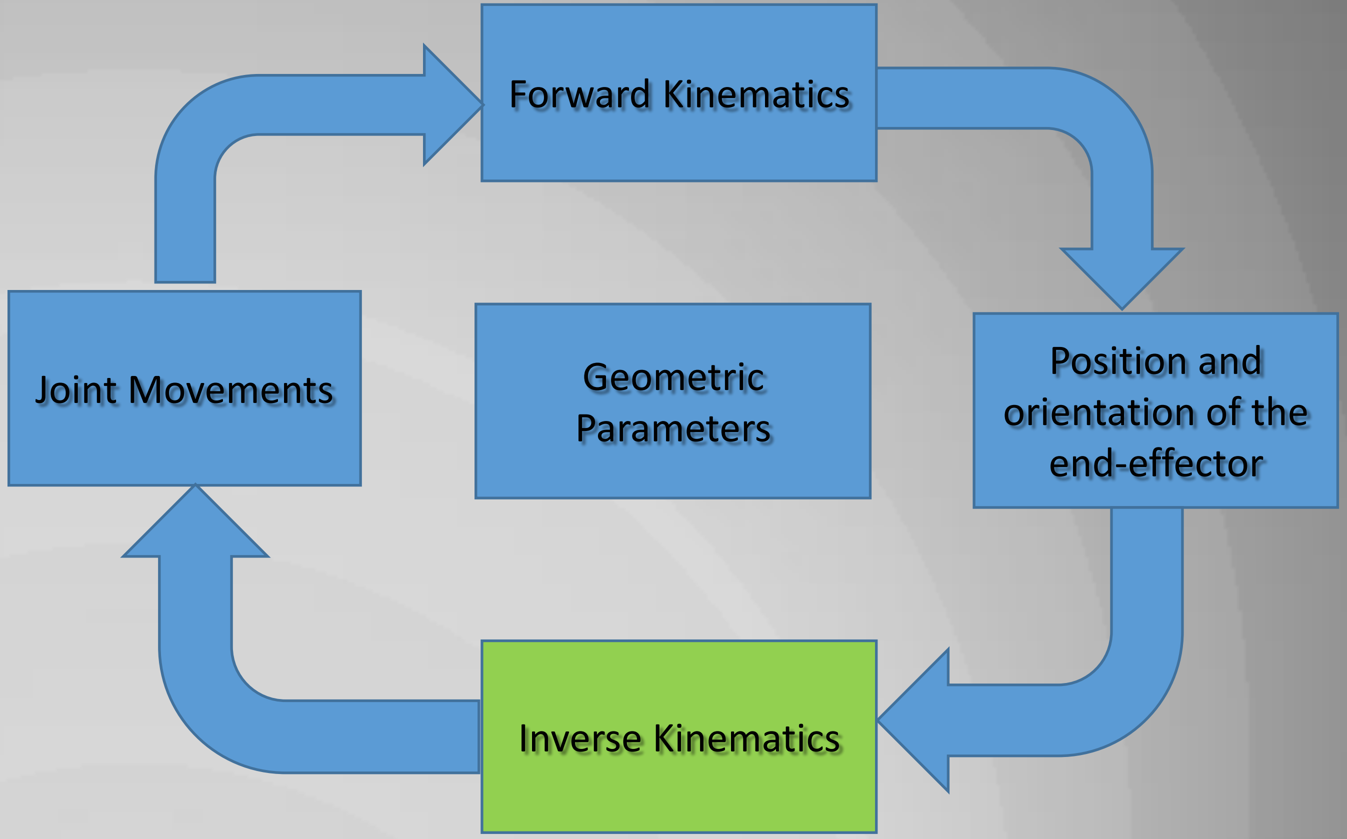


$$T_{grav} = [0, 3.312\,, 1.373\,, 0.372\,, 0\,, 0]\mathrm{N} \cdot m$$

$$T_{grav} = [-3.312\,, 0\,, 0\,, 0\,, 0, 0]\mathrm{N} \cdot m$$

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T g$$

q : *vector in joint coordinates*
$\dot{q}$: *velocity in joint coordinates*
$\ddot{q}$: *accelerations in joint coordinates*
M: *joint − space inertia matrix*
C: *Coriolis and centripetal coupling matrix*
F: *friction force*
G: *gravity loading*
Q: *actuator forces asscoiated with the generalized coordinates* q
J: *manipulator Jacobian*



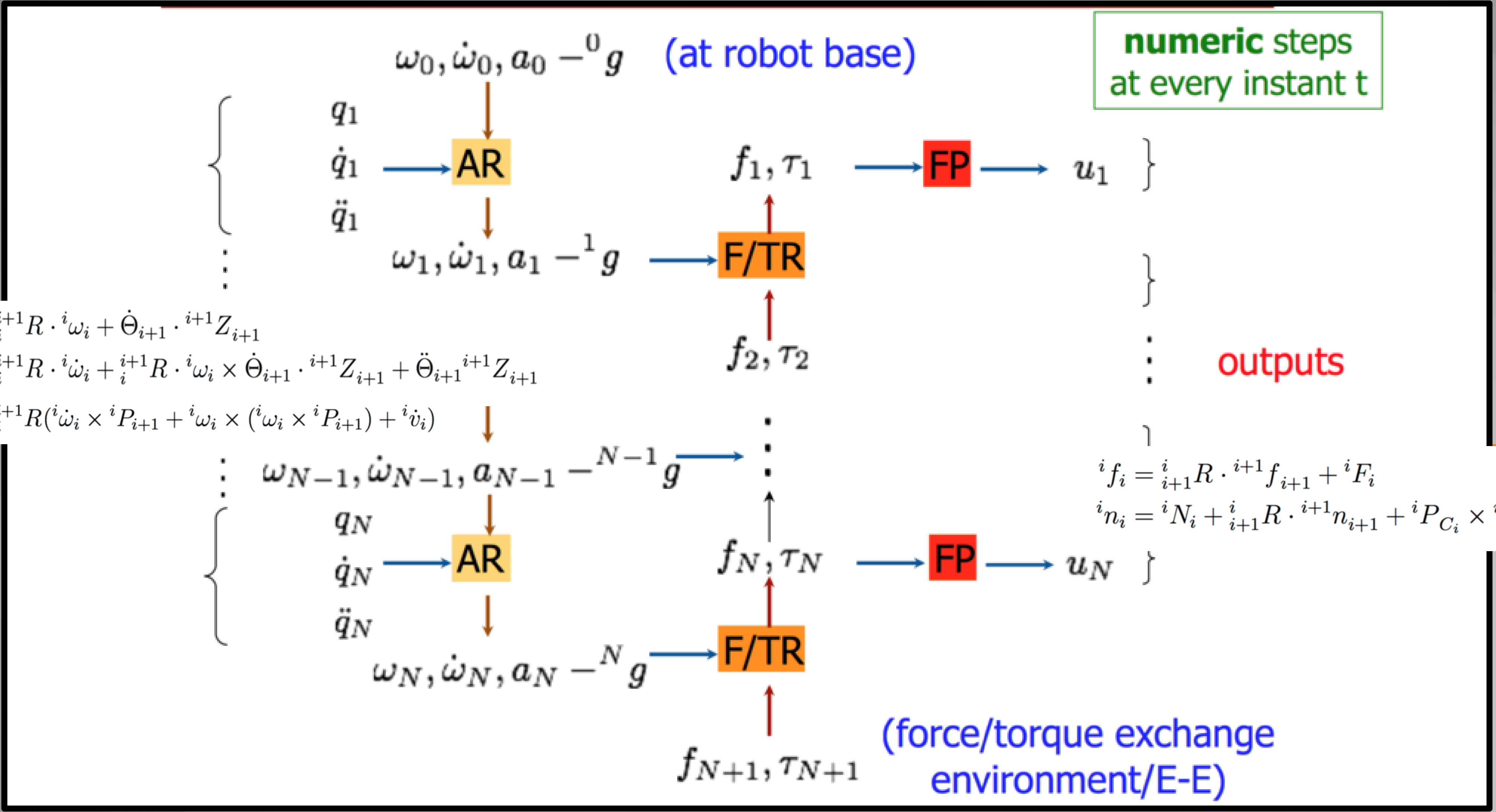
Can be calculated using Newton-Euler Algorithm:

$$\boldsymbol{Q = D(q, \dot{q}, \ddot{q})}$$

(Using .rne function)

This equation describes the manipulator rigid-body dynamics and is known as the inverse dynamics – given the pose, velocity and acceleration it computes the required joint forces or torques

Computational Scheme for Inverse Dynamics Using Newton Euler Algorithm



$${}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \cdot {}^i\omega_i + \dot\Theta_{i+1} \cdot {}^{i+1}Z_{i+1}$$

$${}^{i+1}\dot\omega_{i+1} = {}^{i+1}_{i}R \cdot {}^i\dot\omega_i + {}^{i+1}_{i}R \cdot {}^i\omega_i \times \dot\Theta_{i+1} \cdot {}^{i+1}Z_{i+1} + \ddot\Theta_{i+1}{}^{i+1}Z_{i+1}$$

$${}^{i+1}\dot v_{i+1} = {}^{i+1}_{i}R({}^i\dot\omega_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^i\dot v_i)$$

$${}^if_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1} + {}^iF_i$$

$${}^in_i = {}^iN_i + {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^iP_{C_i} \times {}^iF_i + {}^iP_{i+1} \times {}^{i}_{i+1}R^{i+1}f_{i+1}$$

Jacobian Inverse Techniques

The relationships between joint velocity $\dot{\theta}$, and linear and angular velocities , $\dot{p}$ and $\dot{w}$ are:

$$\dot{p} = J_p(\theta)\dot{\theta}$$
$$\dot{w} = J_w(\theta)\dot{\theta}$$

$$v = J(\theta)\dot{\theta}$$

$$J(\theta) = \frac{v}{\dot{\theta}} = \begin{cases} J_{vi} \\ J_{wi} \end{cases}$$

For a 5 DOF robotic arm, the Jacobian matrix is equal to

$$J(q) = \begin{bmatrix} z_0 \times (o_5 - o_0) & z_1 \times (o_5 - o_1) & z_2 \times (o_5 - o_2) & z_3 \times (o_5 - o_3) & z_4 \times (o_5 - o_4) \\ z_0 & z_1 & z_2 & z_3 & z_4 \end{bmatrix}$$

J_{vi} *angular velovity for revolute joint i*

J_{wi} *linear velovity for revolute joint i*

Gripper Servo Wiring

- Make wires long enough to rotate wrist 2 full rotations
  - With CubeSat spinning at 0.5 rpm, this gives us 4 minutes to extend the arm, close the gripper, confirm capture, and end demonstration
  - Implement software failsafe where upon nearing the 4 minute mark, the arm will back out and rotate the wrist back to it's starting position before re-attempting capture

Pros	Cons
Simplest in terms of hardware interface	Adds complexity of timing requirement
Lower cost – no additional components	Added risk of breaking wires

- Other options considered but ruled out
  - Slip ring
    - Adds unwanted noise and difficulty with hardware interface
  - Microcontroller driving gripper servos
    - Adds complexity of programming another device and a slip ring would still be needed for power

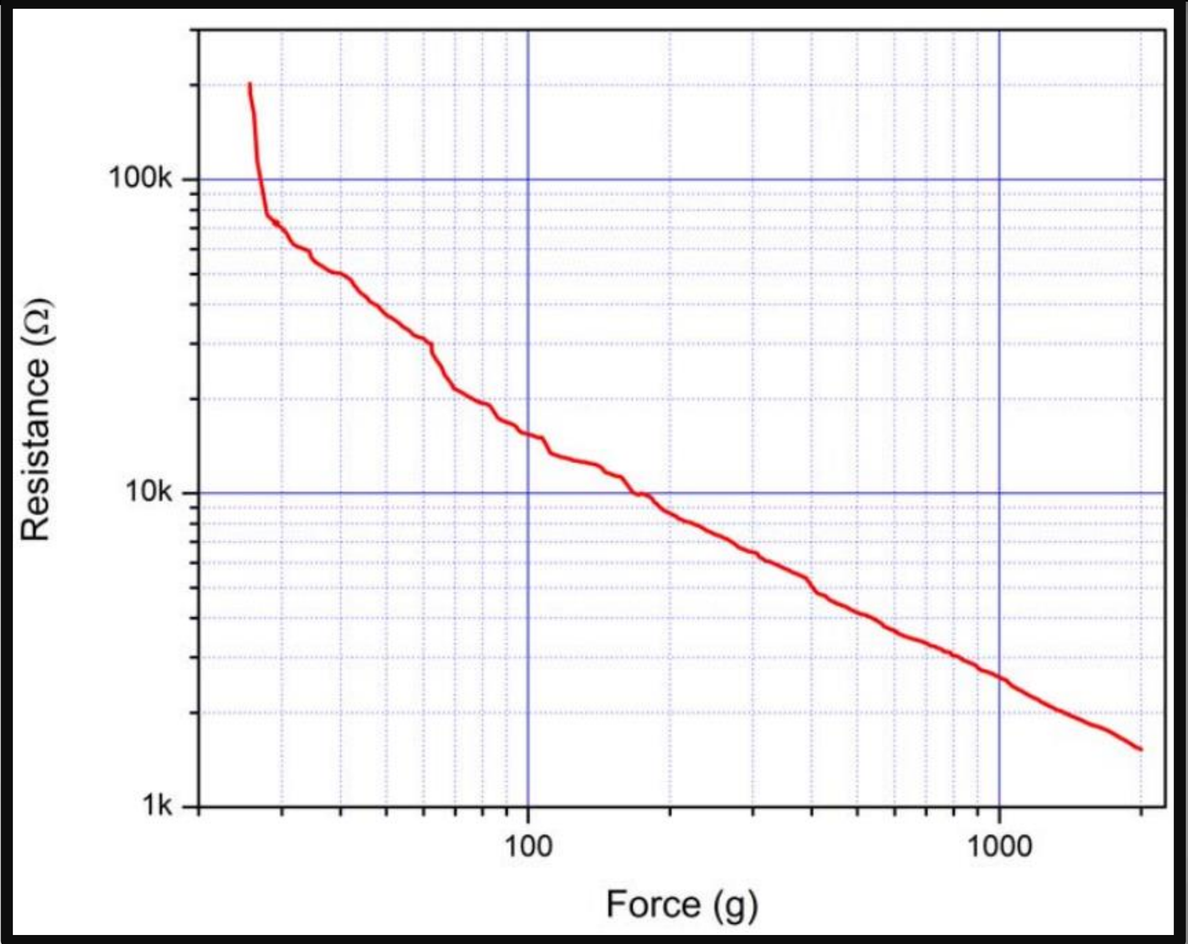
Force Sensor Calibration

$$V_{out} = \left(\frac{R_p + R_{Fsr}}{5v \ * R_p}\right) * \left(1 + 2\,\frac{20\ kohms}{R_g}\right)$$

R_{Fsr} : *Force Sensor Resistance*

R_p: *Parallel resistance*

R_g: *Gain resistance*



Apply Known Weights → Measure Voltage → Calculate FSR → Fit to Datasheet Resistance

Servo Command Protocol

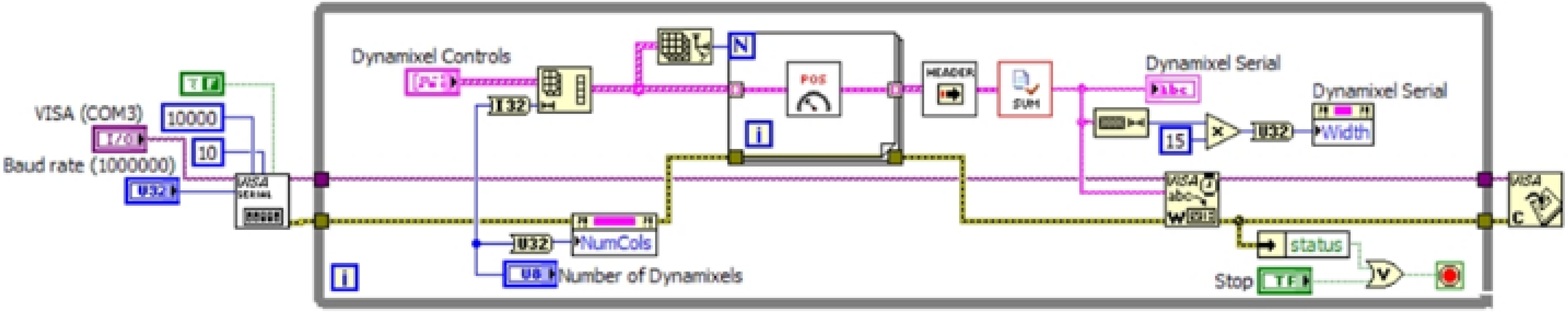
START	ALL	T.LEN	SYNC	INSTRUCTION	LENGTH	ID 1	P 1	P 2	ID N	P 1	P 2	C.SUM

START: (Hex 0XFF 0XFF) The double FF initializes communication between the COM and the Dynamixels.
ALL: Broadcast ID (Hex-0XFE
) to all Dynamixels, disables return of status packets
T.LEN: Total Length, Uses the formula (NxL)+N+4, where N is number of Dynamixels, L is the length of the Control Table Bytes, and 4 is the length of the header bytes to be used by the checksum (Manual uses the formula (L+1)xN+4 this achieves the same solution as the formula used).
SYNC: The Sync Write (Hex 0x83) defines the command being used
INSTRUCTION: Starting Byte Address from Control Table (Goal Position (L) 0X1E).
LENGTH: L is the length of the Control Table bytes used
ID P1 P2 IDN P1 P2: Dynamixel ID and Position Control (Position sub VI)
C.SUM: (CheckSUM sub VI)

http://www.ni.com/white-paper/12557/en/

- Commands for a write instruction.
- SYNC and INSTRUCTION bits can be changed for different servo commands
- Check Sum transmitted at the end of each write command for error checking.

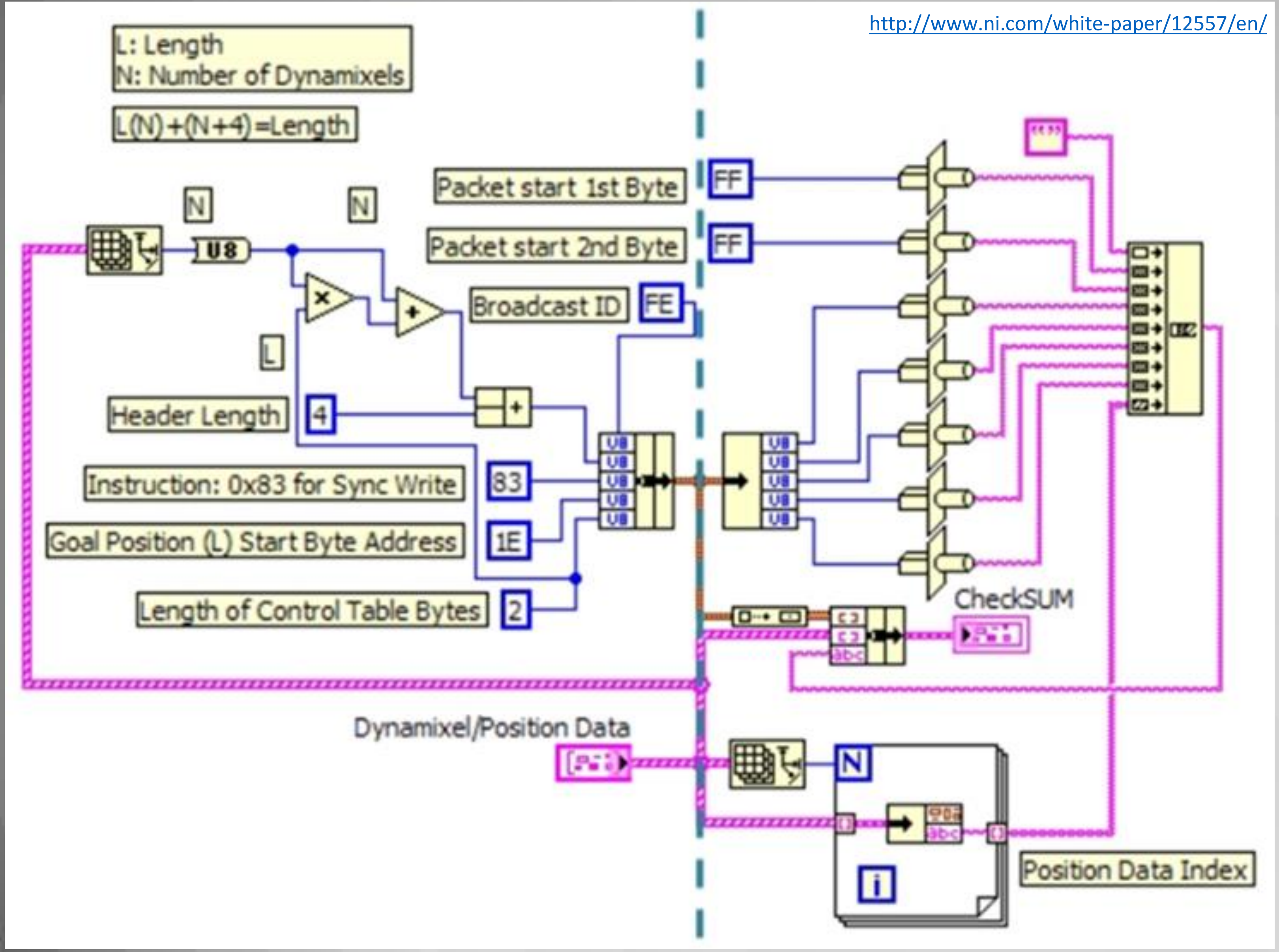
Servo Commands-Labview



http://www.ni.com/white-paper/12557/en/

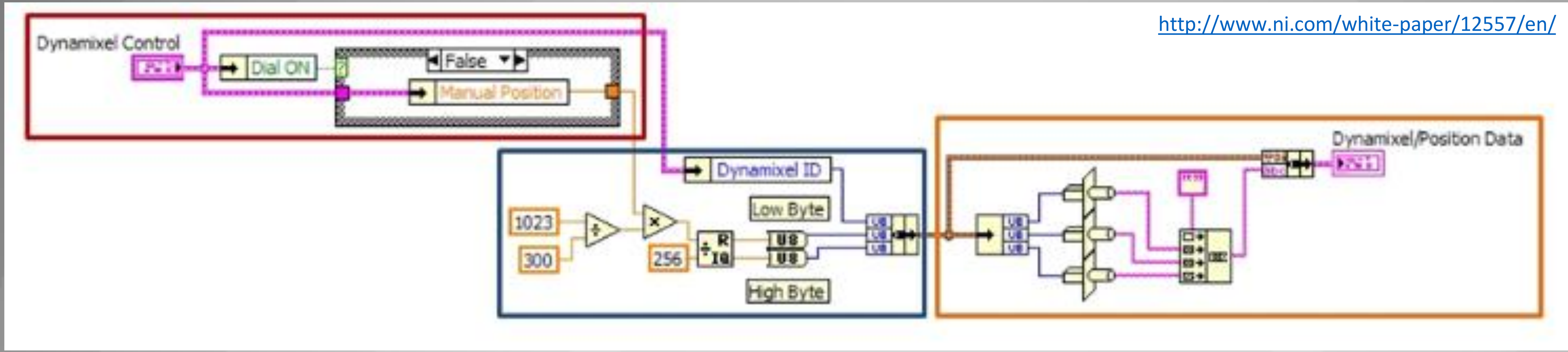
- Top Level LabView for writing to servos
- Will be modified to fit within our software hierarchy

Servo Commands-Labview

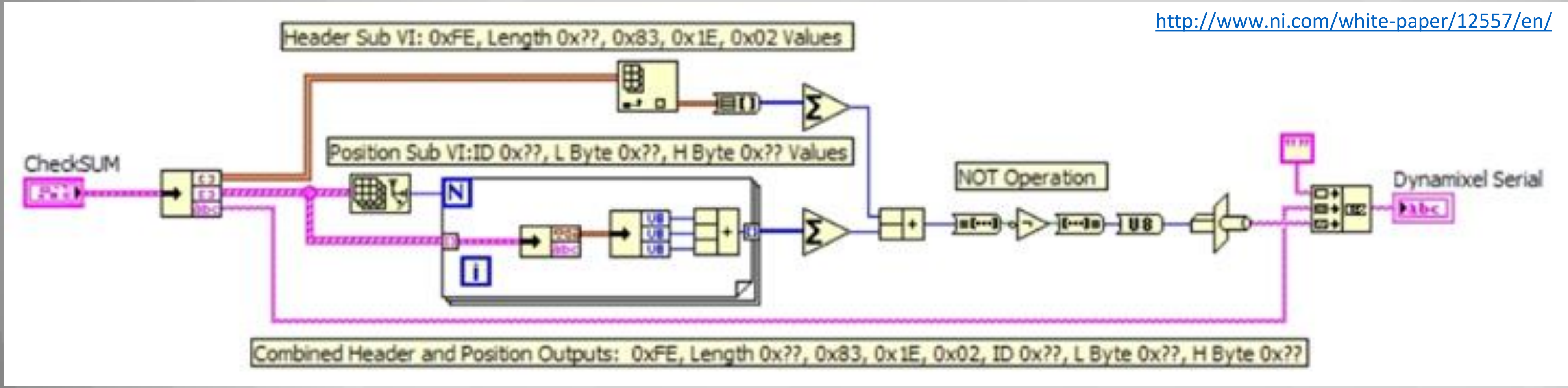


- Header VI
- Left of dotted line: SYNC and INSTRUCTION parameters set
- Right of dotted line: Data is bundled and concatenated into a final package to be sent to the servos.

Servo Commands-Labview



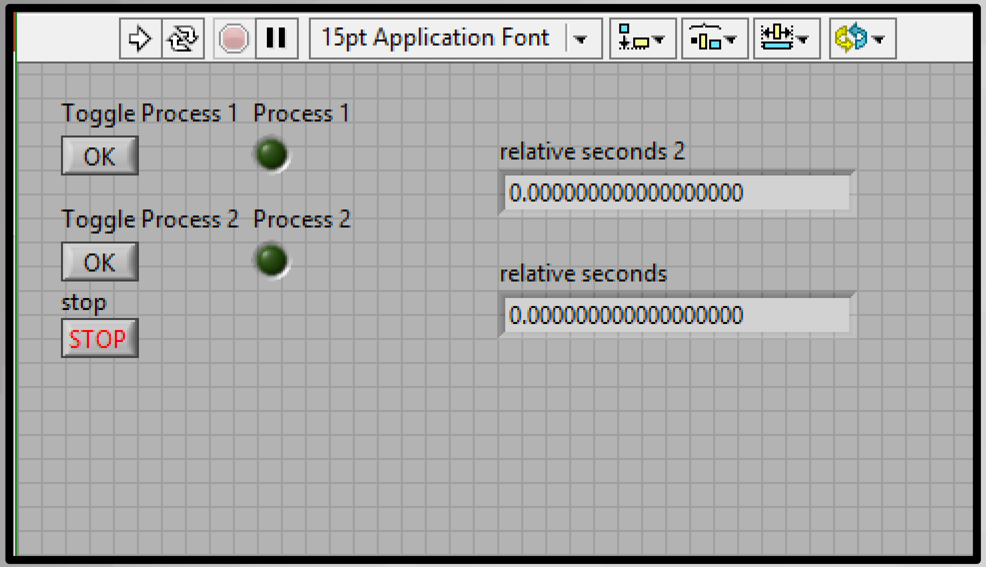
- Position Data VI for forming desired position into bytes, ours will be for velocities instead.

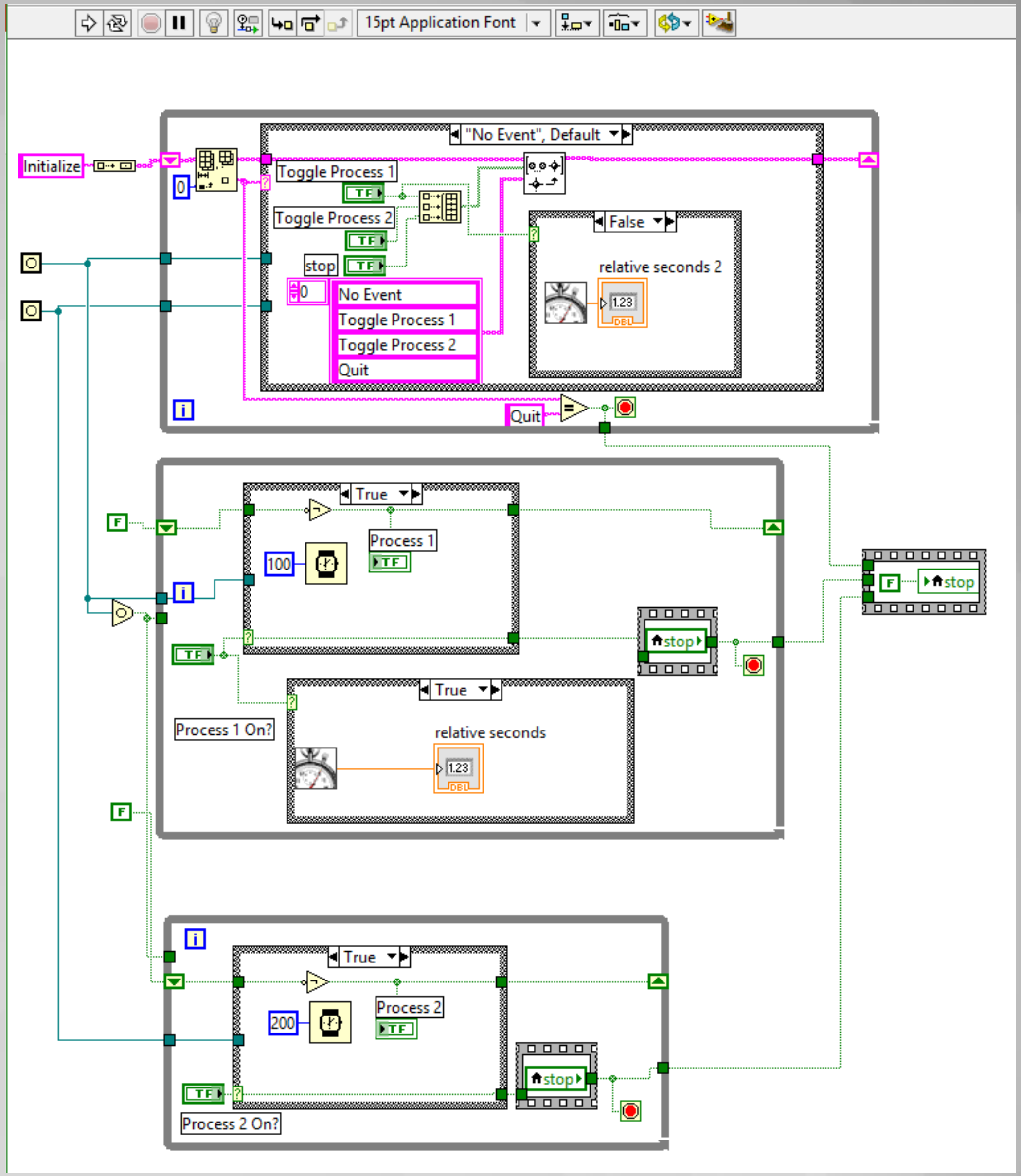


- Check SUM VI for framing and bit errors.

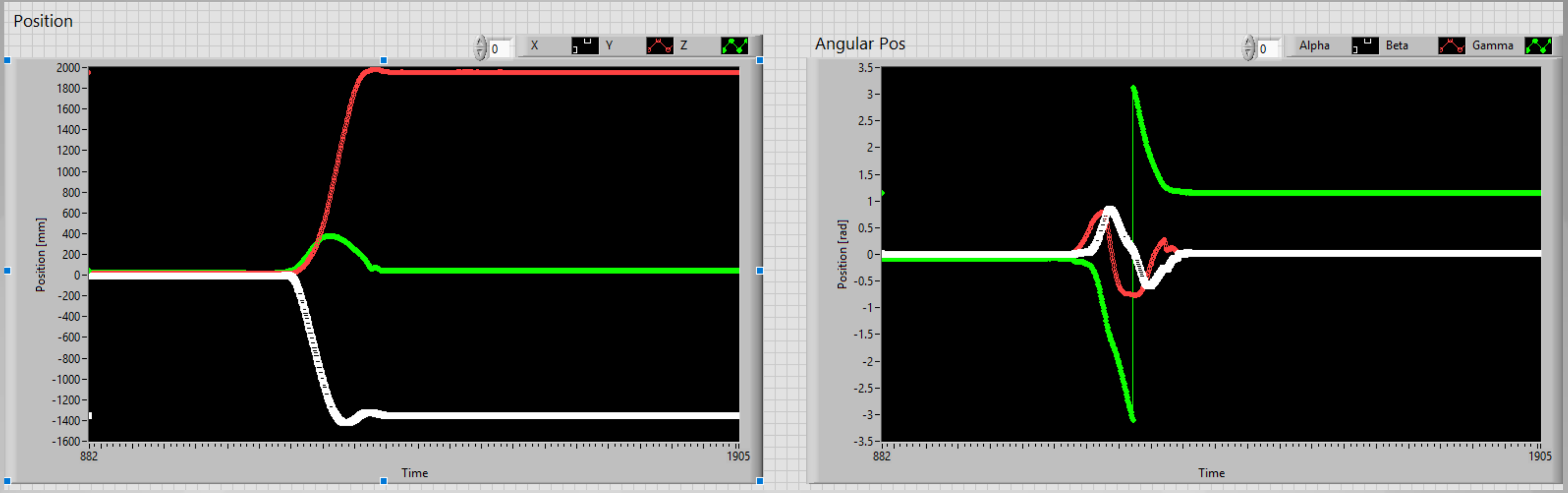
Backup: Timing Diagram

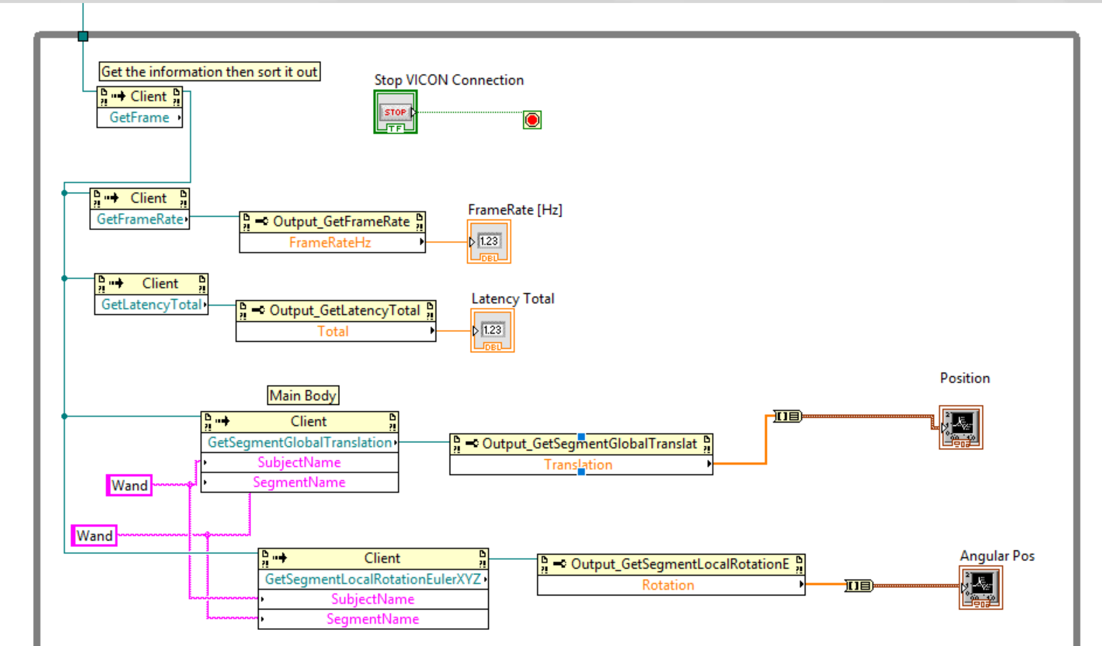
- Master Slave Message Timing
- 35.7 Microseconds between loops





VICON Interface Backup





Software Derived Requirements

Testbed Control Requirement

Based on 5 cm/s Translation Rate on Linear Rail

- Force Grippers are 2.54 cm on a 10 cm Surface Parallel to Linear Rail
- Takes 0.746 s until Force Grippers start are not fully on CubeSat
- f = 1.34 Hz Implies f_s = 13.4 Hz

Arm Control Requirement

Phase: CubeSat spins at 3 degrees/sec to cover 10 degrees of Phase Offset

- f = 0.3 Hz Implies f_s= 3.3 Hz

Arm Servos: Max Unloaded Angular Speed of 63 RPM = 1.05 Hz

- Implies f_s = 10.5Hz
- Since Jacobian Solve for End Effector and Joints, Used highest Rate Required

VICON Requirement

Shall Supply Arm Control Loop with Data at the f_s rate of 10.5 Hz

LabVIEW Timing Backup

LabVIEW: Testbed

- 2 General PID Controller

LabVIEW: Arm + End Effector

- Inverse Jacobian
- Just need to add time to compute Inverse Jacobian in LabVIEW (x2)
  In 3 different ways to solve
- $\delta t =$ ~0.9ms
- f = 144 Hz

LabVIEW: VICON

- Programed Own Code
- Read DataStream at 100 Hz
- Programed AOR Calculation into MATLAB
- Called MATLAB Script from LabVIEW (X5 for margin)
- Added no time since Loop doesn't run until Data is received

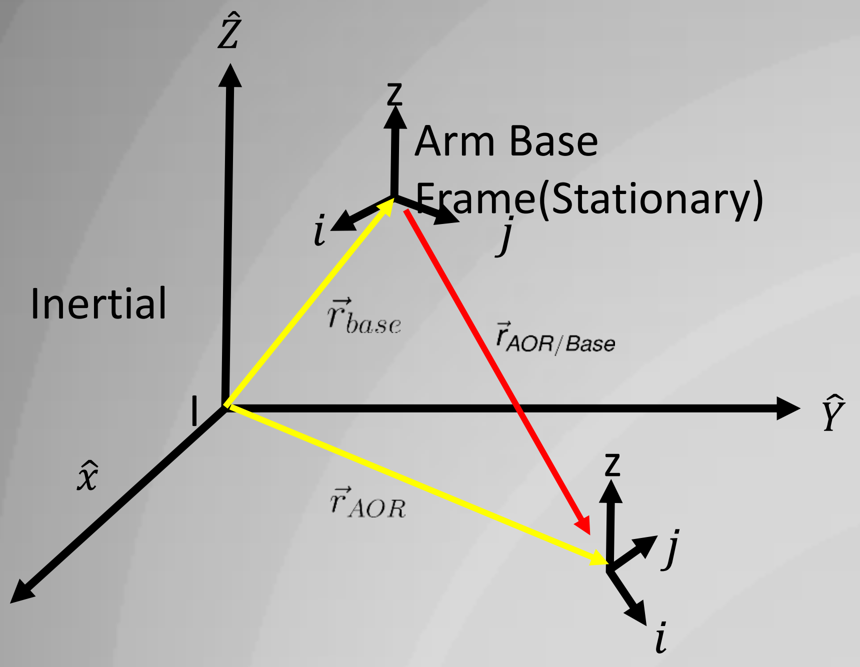
$$T = \texttt{Period of Loop}$$

$$T_{Equ} = 6.02[ms]$$

$$T_{new} = T_{Equ} + \sum_{i=1}^{N} \delta t_i$$

Desired Position: End Effector

1. Convert to Arm Base Frame



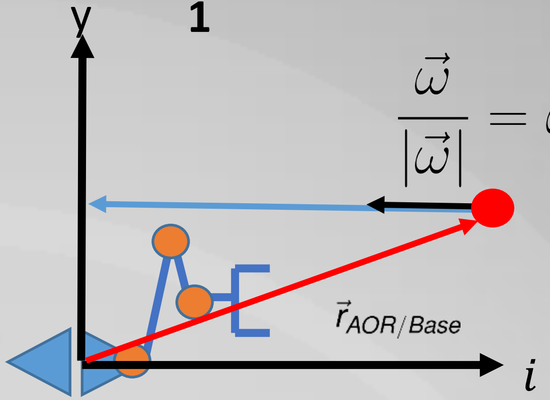
CubeSat Body Frame (Moving)

$$[\vec{r}_{AOR/Base}]_{Base} = [Q][\vec{r}_{AOR/Base}]_I$$

$$[\vec{\omega}]_{Base} = [Q][\vec{\omega}]_I$$

2. Determine Initial Guess for Goal 1

$$\frac{\vec{\omega}}{|\vec{\omega}|} = \hat{\omega}$$



Base—Top Down View

$$\vec{r}_{AOR} + X_{AOR}\hat{\omega} = \begin{bmatrix} 0 \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} + X_{AOR} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 \\ Y \\ Z \end{bmatrix}$$

Use Initial guess to iterate to Determine Feasible Position for Goal 1.

Desired Position of End Effector

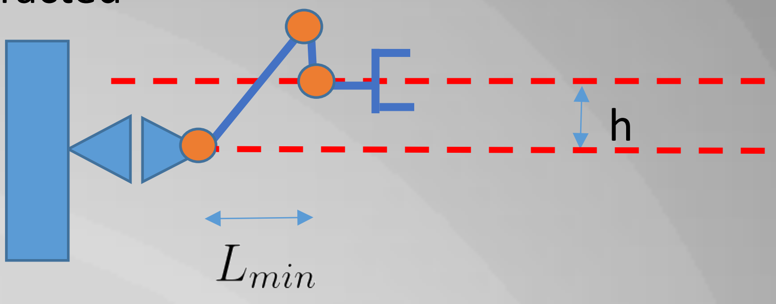
$$L_{max} = \sqrt{0.36^2 - h^2}$$

$$\epsilon = L_{max} - L_{min}$$

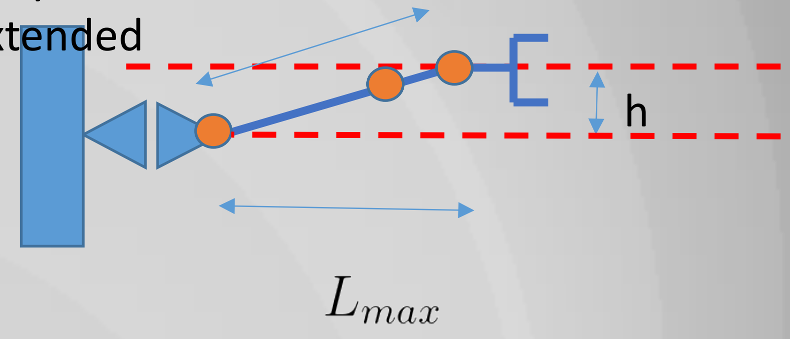
$$\epsilon = \texttt{Extension Length}$$

L_{min} is Numerically Approximated

Retracted



Fully Extended



Verification/Validation Overview

	Subsystems: CubeSat	Subsystems: Arm	Subsystems: Claw	Full System
Sensors	Vicon Rotary Encoder **FR 1.2, 1.3**	Servo Feedback **FR 1.3**	Servo Feedback Vicon Force sensors **FR 1.3**	Vicon **FR 1.3**
Motion	Rotation Performance Translation Performance **FR 1.2**	Arm Performance **FR 1.5**	Claw Performance **FR 1.5**	Levels of Success **FR 1.1, 1.5, 1.6**
Software	LabVIEW Control Phase 2 Movement Capture Confirmation **FR 1.2, 1.6**	Algorithm & LabVIEW control Phase Checks **FR 1.3, 1.4, 1.6**	Capture Confirmation **FR 1.5, FR 1.6**	Levels of Success **FR 1.1-1.6**
Models	Dynamics Model **FR 1.2**	Dynamics Model **FR 1.5**		

CubeSat Risks

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	SEV	Potential Cause(s)/ Mechanism(s) of Failure	OCC	Current Process Controls	DET	RPN
Rotation of CubeSat (CS)	Motor malfunction	Can't redo demo w/o replacement	6	Overheating	4	Testing for nominal operating range prior to demo	3	72
Translation of Cubesat	Motor malfunction	Can't redo demo w/o replacement	6	Overheating	4	Testing for nominal operating range prior to demo	3	72
Control of CS Translation	Control loop inadequate	Demo failure	6	Poor design, gain limitation on control	3	Testing for nominal operating range prior to demo	3	54
Control of CS Rotation	Control loop inadequate	Demo failure	6	Poor design, gain limitation on control	3	Testing for nominal operating range prior to demo	3	54
Motor Shut off @ end of demo	CS motor or wrist servo stalls	Motor damage, can't redo demo w/o replacement	6	Poorly designed failsafes, inaequate testing	3	Testing for nominal operating range prior to demo	3	54

Robotic Arm Risks

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	SEV	Potential Cause(s)/ Mechanism(s) of Failure	OCC	Current Process Controls	DET	RPN
Arm Motion	Arm damages itself	Demo failure	6	Code error	**5**	Testing prior to final demo	4	120
Claw Motion	Servo hits stall torque	Demo failure, servo damaged, can't redo demo w/o replacement	6	Code error	**5**	Testing prior to final demo	3	90
Capture Confirmation	Pressure sensors inadequate to detect valid capture	Demo failure	6	Sensor placement, sensitivity; electrical failure; bad sensor	4	Testing prior to final demo	5	120

Robotic Arm Risks

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	SEV	Potential Cause(s)/ Mechanism(s) of Failure	OCC	Current Process Controls	DET	RPN
Arm Motion	Arm offset from AOR	Too much torque on arm joints	6	Code error	**5**	Testing prior to final demo, IR sensors on claw for checking absolute position.	4	120
Wrist Rotation	Wires wrap around too many times and break	Demo failure, claw and wiring potentially damaged	6	Timing error in code	**5**	Testing prior to final demo, software shutoff if wrist rotates too far	3	90

Verification/Validation Overview

	Subsystems: CubeSat	Subsystems: Arm	Subsystems: Claw	Full System
Sensors	Vicon Rotary Encoder **FR 1.2, 1.3**	Servo Feedback **FR 1.3**	Servo Feedback Vicon Force sensors **FR 1.3**	Vicon **FR 1.3**
Motion	Rotation Performance Translation Performance **FR 1.2**	Arm Performance **FR 1.5**	Claw Performance **FR 1.5**	Levels of Success **FR 1.1, 1.5, 1.6**
Software	LabVIEW Control Phase 2 Movement Capture Confirmation **FR 1.2, 1.6**	OpenRave Algorithm & LabVIEW control Phase Checks **FR 1.3, 1.4, 1.6**	Capture Confirmation **FR 1.5, FR 1.6**	Levels of Success **FR 1.1-1.6**
Models	Dynamics Model **FR 1.2**	Dynamics Model **FR 1.5**		

Backup Slides Appendix

Purpose and Objectives	Design Solution		Requirements		Risks		Project Planning		
CONOPS	Model	RECUV	FReqs	Arm Control Theory	Intro	Dynamics Model-Accuracy	Org Chart		
	Geometry	Software	CubeSat Translation	Arm limits	Matrix	Dynamics Model - Power	WBS		
	Flow Diagram	LabView	CubeSat Rotation	Claw Compliance	Verification		Work Plan		
	Robotic Arm		VICON Interface				Critical Path		
	End Effector		LabVIEW				Test Plan		
	CubeSat Motion		Axis of Rotation				Overall Budget		
	Hardware Diagram		Position of Effector						