SIERRA NEVADA CORPORATION


# Mechanically Engineered Grappling Arm to Capture Litter and Atmospheric Waste

## Preliminary Design Review

Presenters: *Joseph Beightol, Benjamin Elsaesser, Caleb Inglis, Jack Isbill, Andy Kain, Cedric Leedy*

Team: *Luke Beasley, Joseph Beightol, Benjamin Elsaesser, Sheridan Godfrey, Caleb Inglis, Jack Isbill, Andy Kain, Cedric Leedy, Chris Leighton, Daniel Mastick, Bailey Topp*

Customer: *TJ Sayer - Sierra Nevada Corporation (SNC)*

Advisor: *Francisco Lopez Jimenez*

SIERRA NEVADA CORPORATION

MEGACLAW
University of Colorado
Senior Design

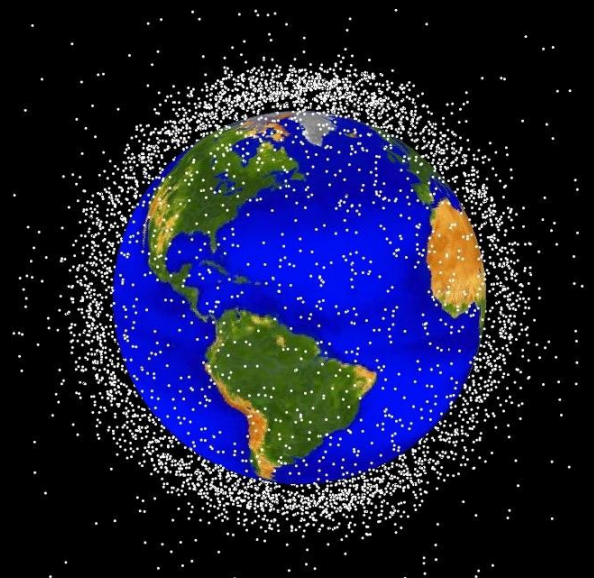
# Project Overview

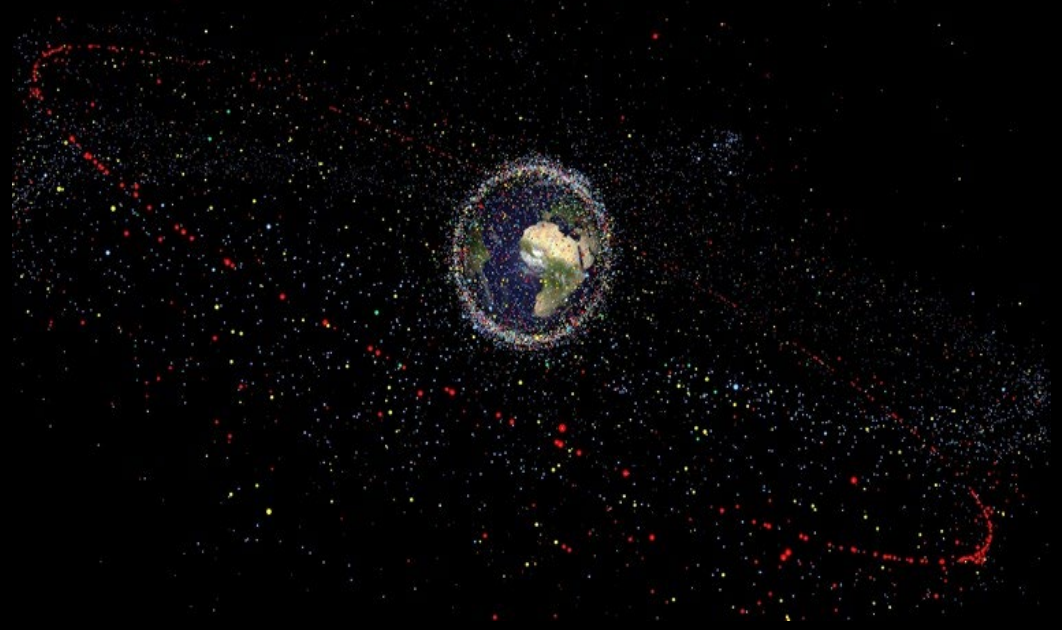
# Project Motivation

**The amount of debris in orbit will only increase if nothing is done to mitigate it.**

**MEGACLAW is a proof of concept for mitigating debris by capturing then deorbiting debris or performing maintenance on broken spacecraft.**

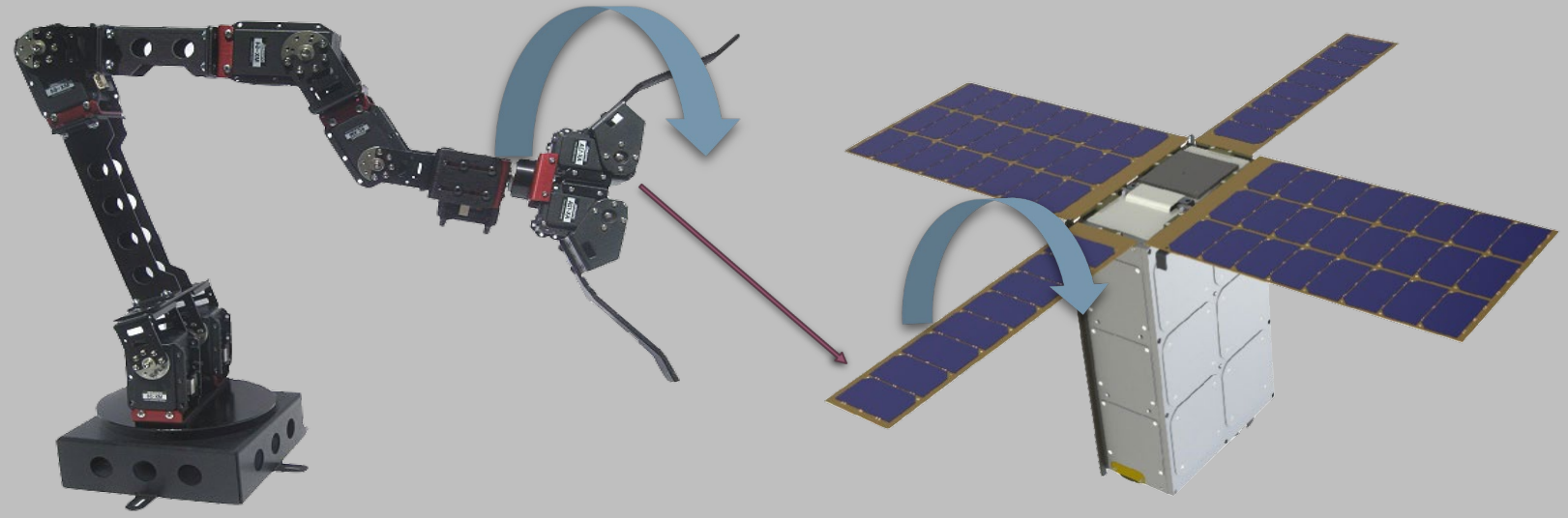
**Heritage: CASCADE, KESSLER**





# Project Statement

- MEGACLAW shall use a robotic arm equipped with an end effector to grapple a grapple point on a flat plate spinning on a motor at a constant rate, which simulates a solar panel on a 6U CubeSat rotating about a single axis of rotation.

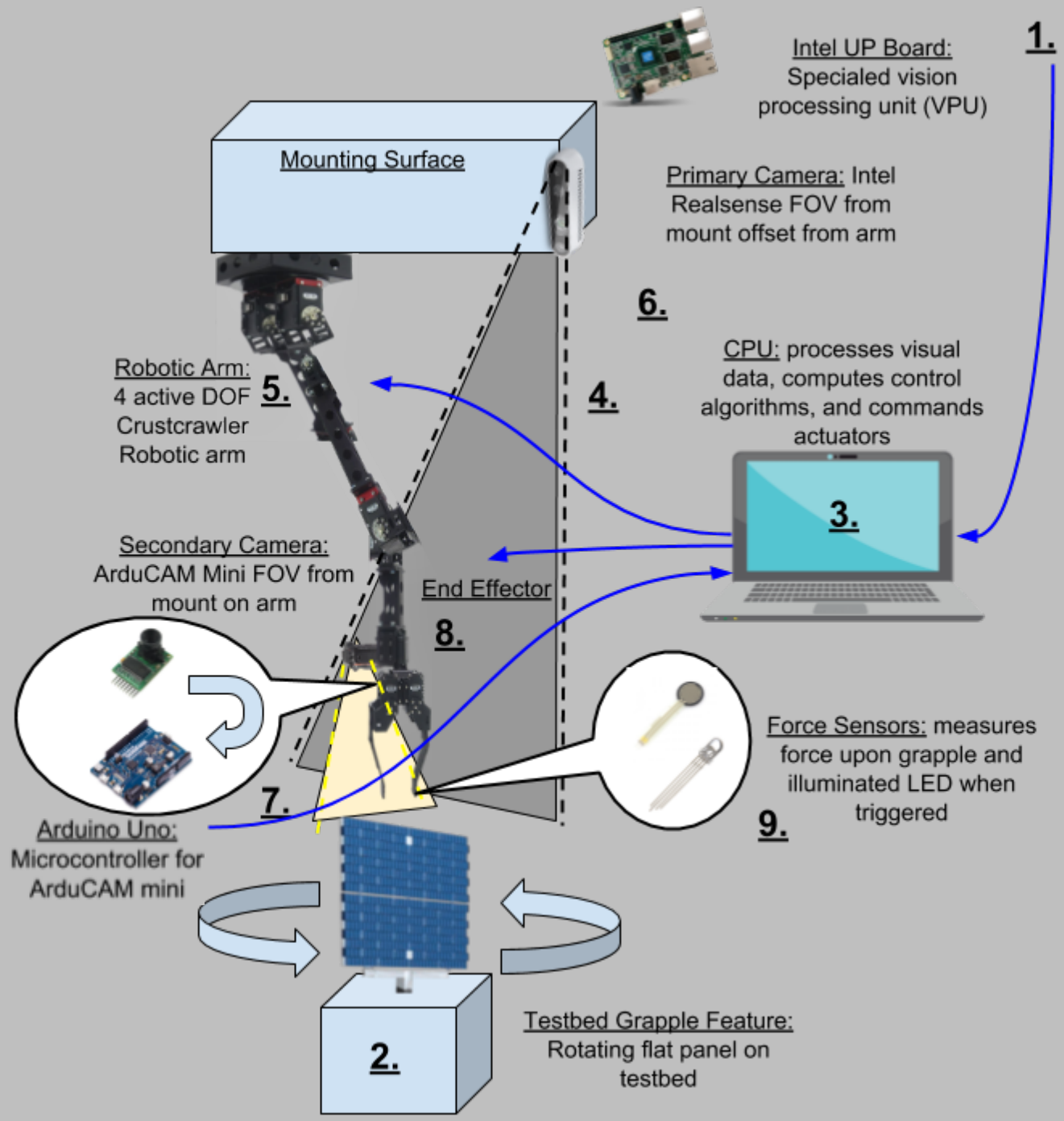


SIERRA NEVADA CORPORATION

# Big Picture CONOPS

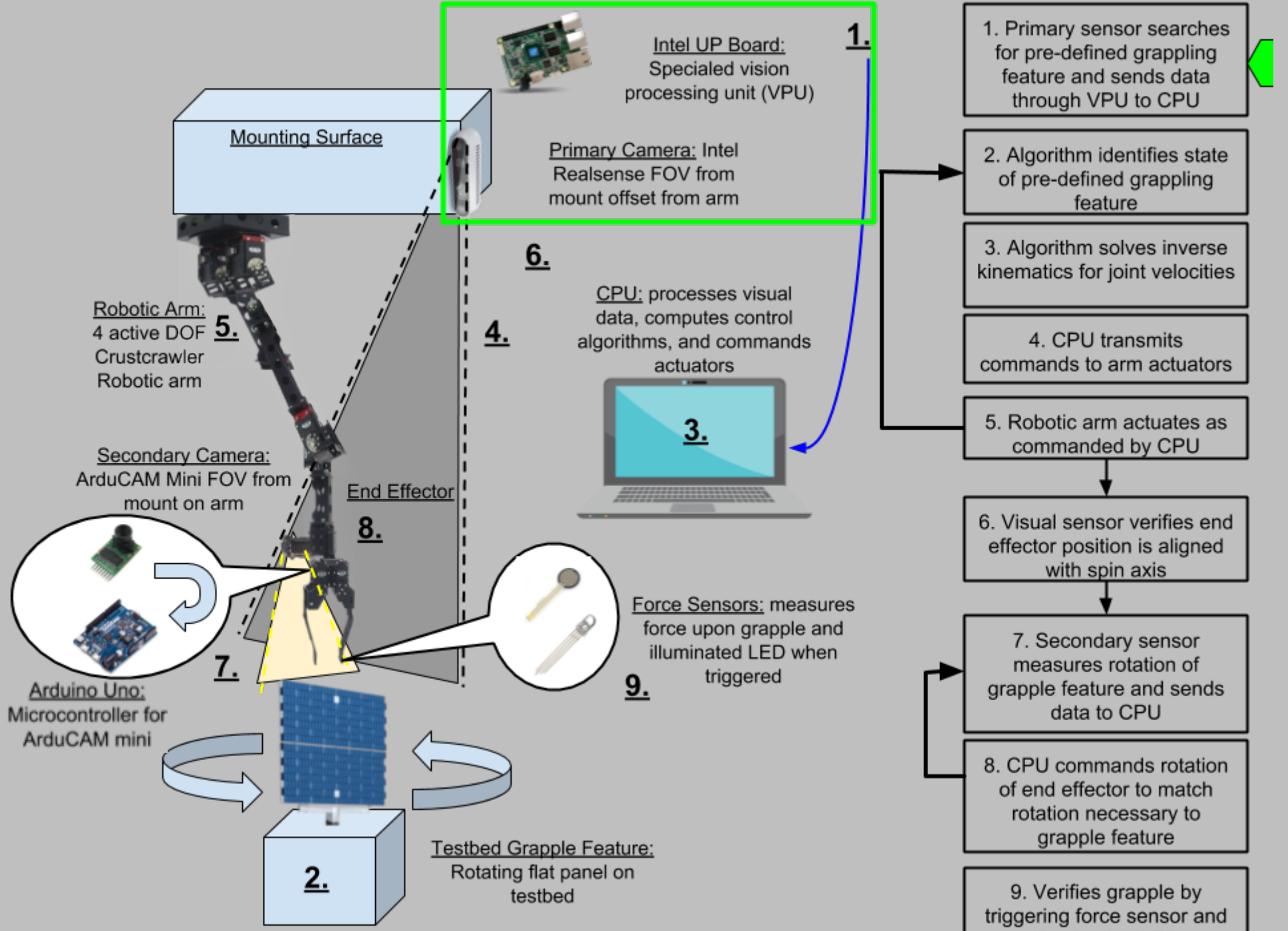
SIERRA NEVADA CORPORATION


# Specific Mission CONOPS



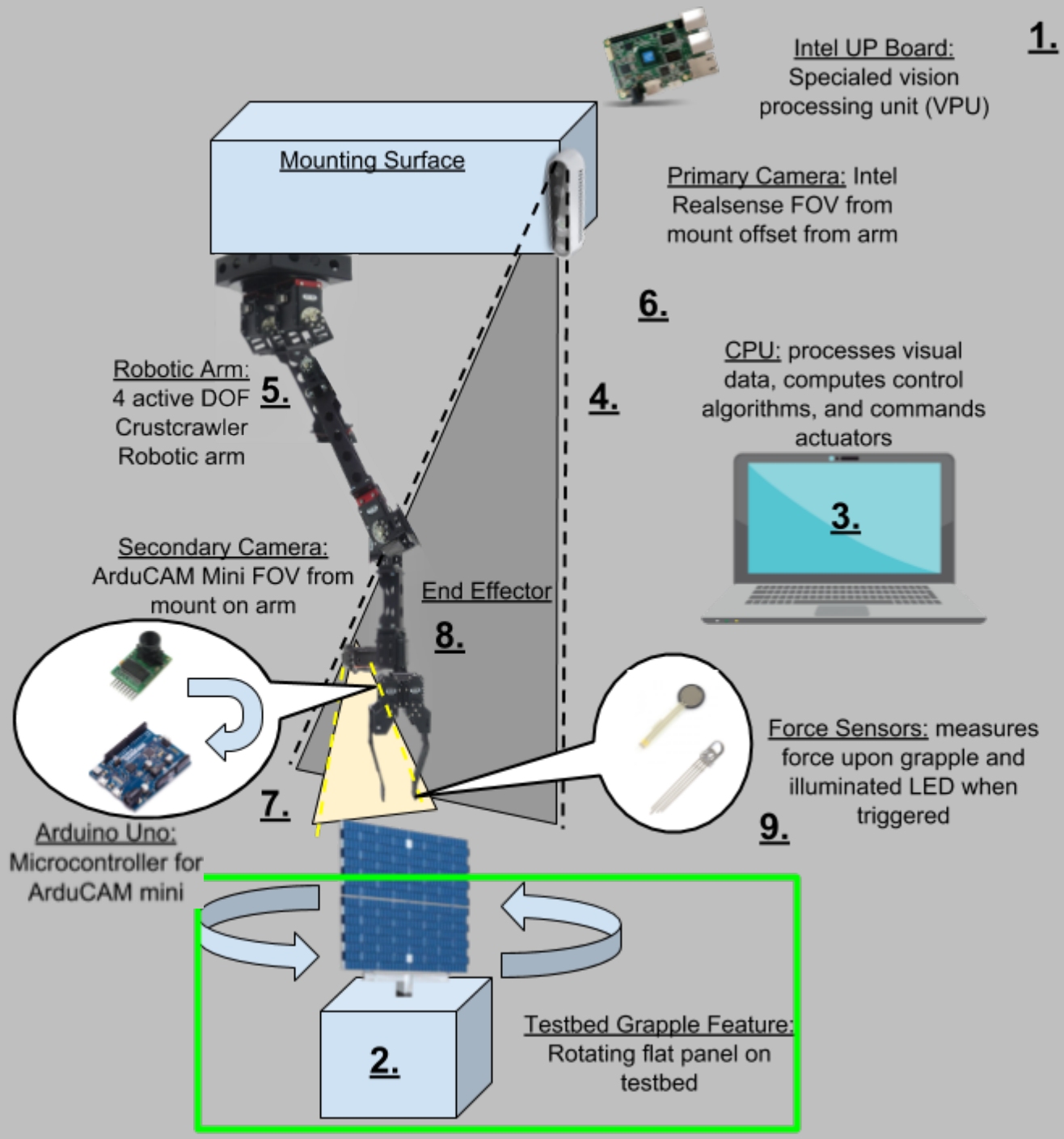

1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED

MEGACLAW
University of Colorado
Senior Design

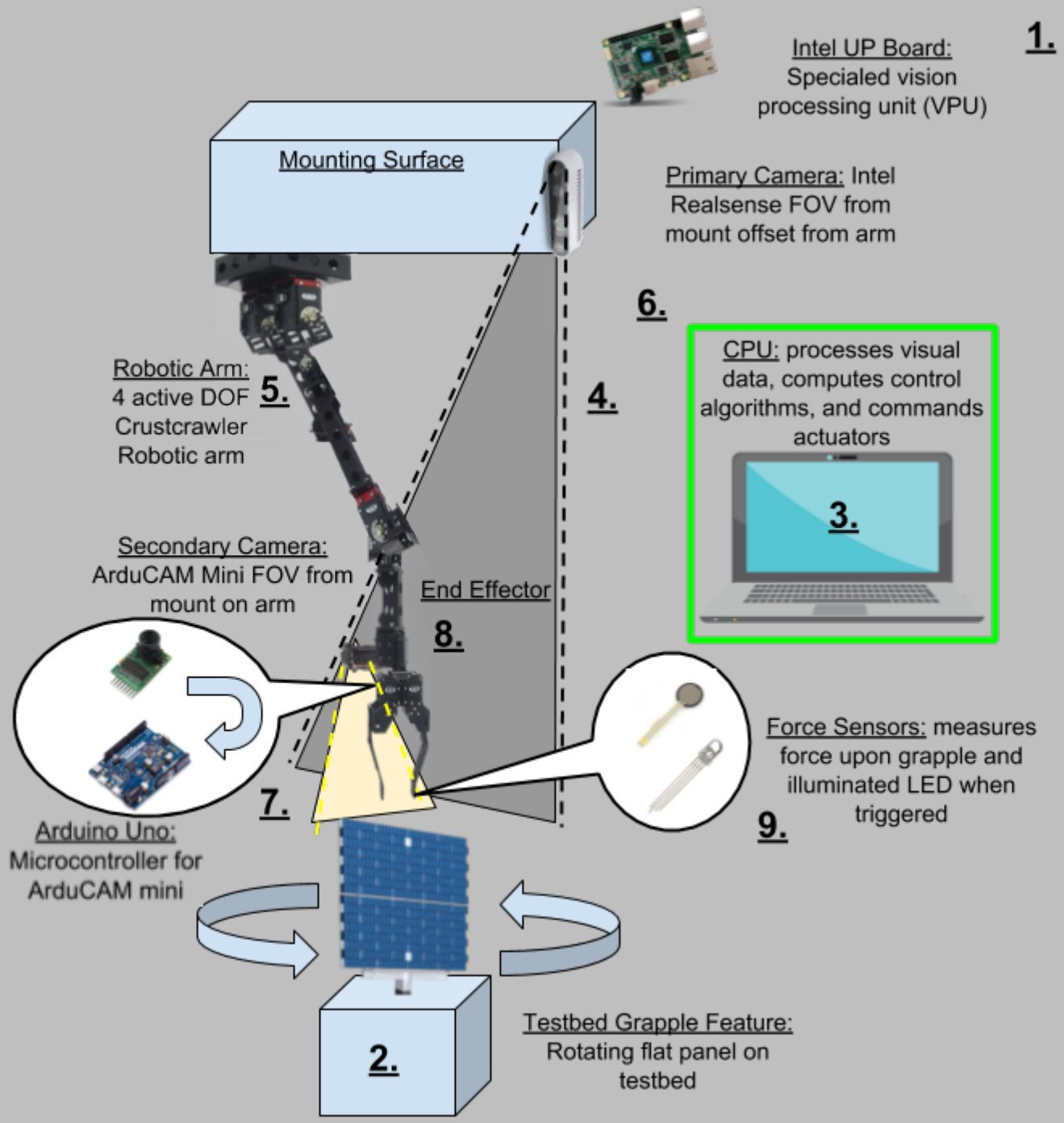



1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED

MEGACLAW
University of Colorado
Senior Design

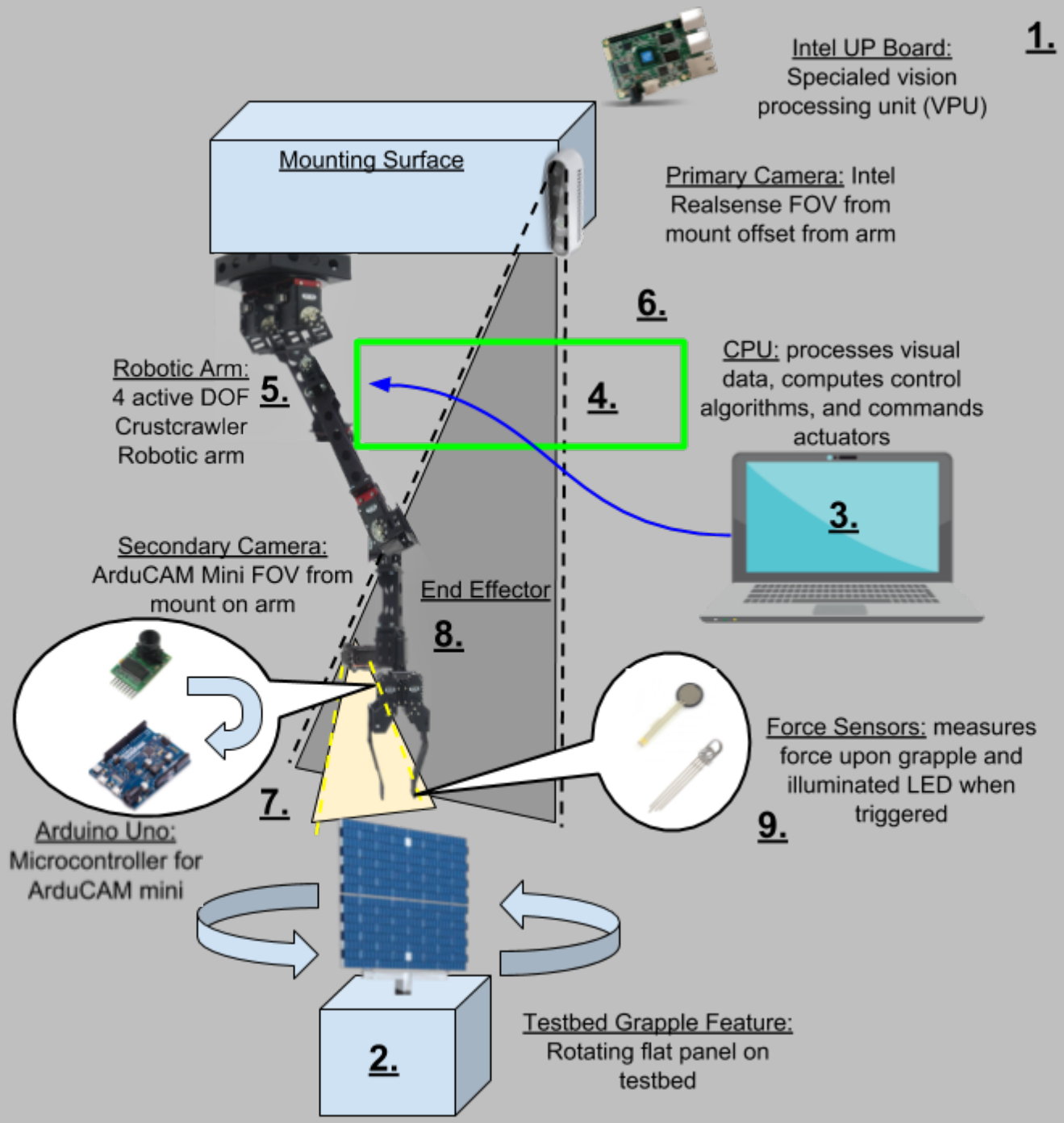



1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED




1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED

MEGACLAW
University of Colorado
Senior Design




1.

1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU

2. Algorithm identifies state of pre-defined grappling feature

3. Algorithm solves inverse kinematics for joint velocities

4. CPU transmits commands to arm actuators

5. Robotic arm actuates as commanded by CPU

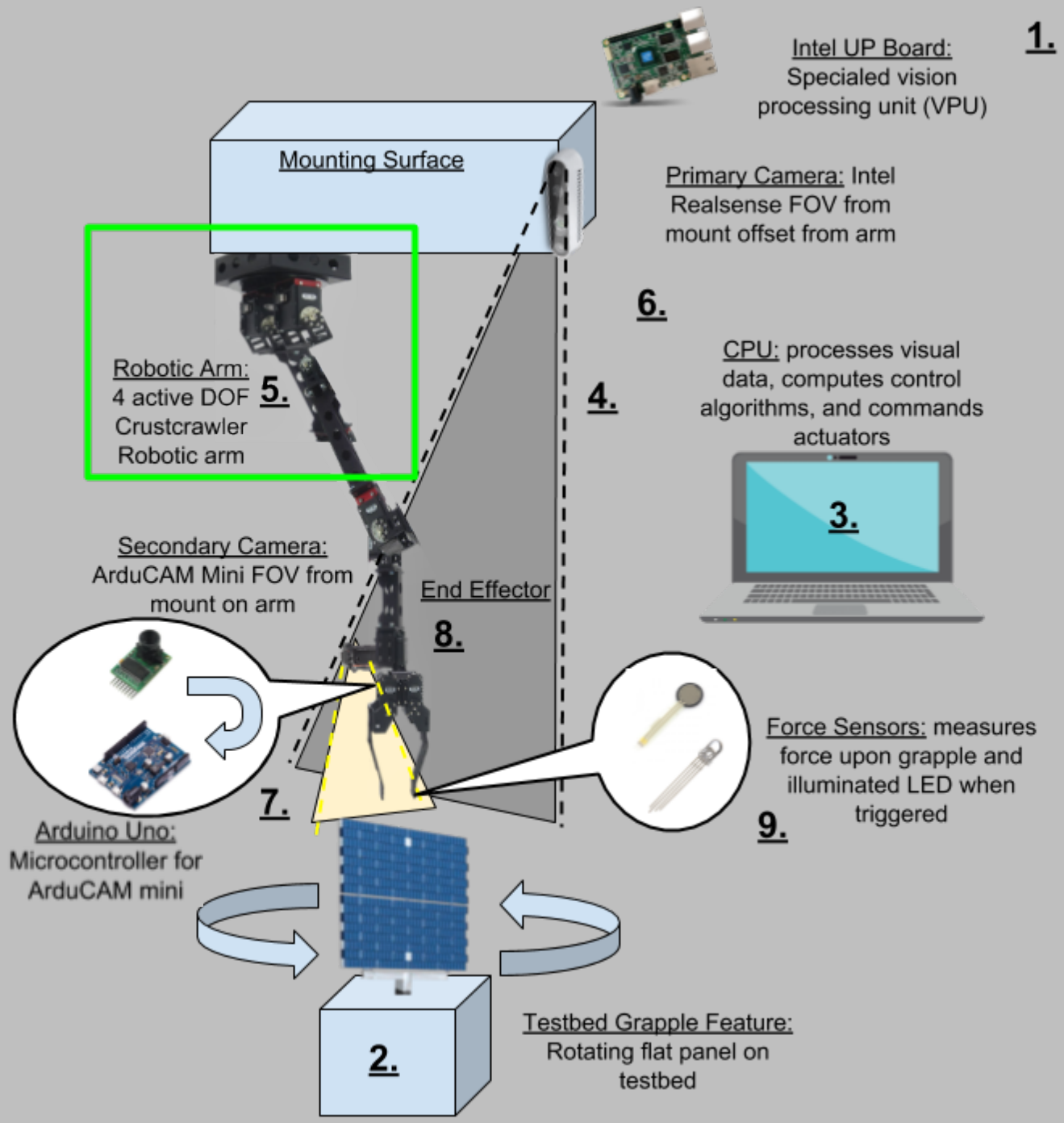
6. Visual sensor verifies end effector position is aligned with spin axis

7. Secondary sensor measures rotation of grapple feature and sends data to CPU

8. CPU commands rotation of end effector to match rotation necessary to grapple feature

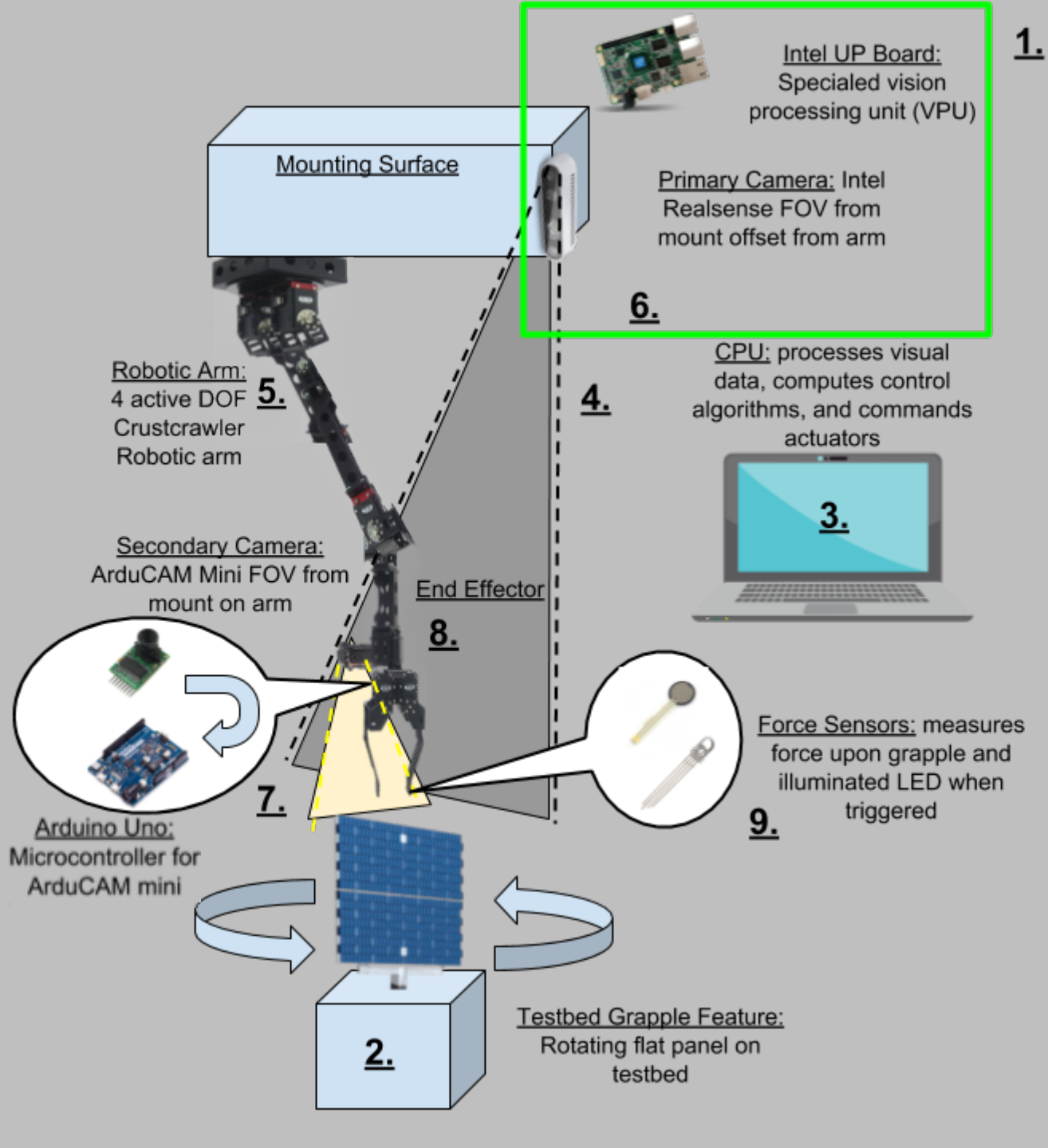
9. Verifies grapple by triggering force sensor and illuminating LED

MEGACLAW
University of Colorado
Senior Design



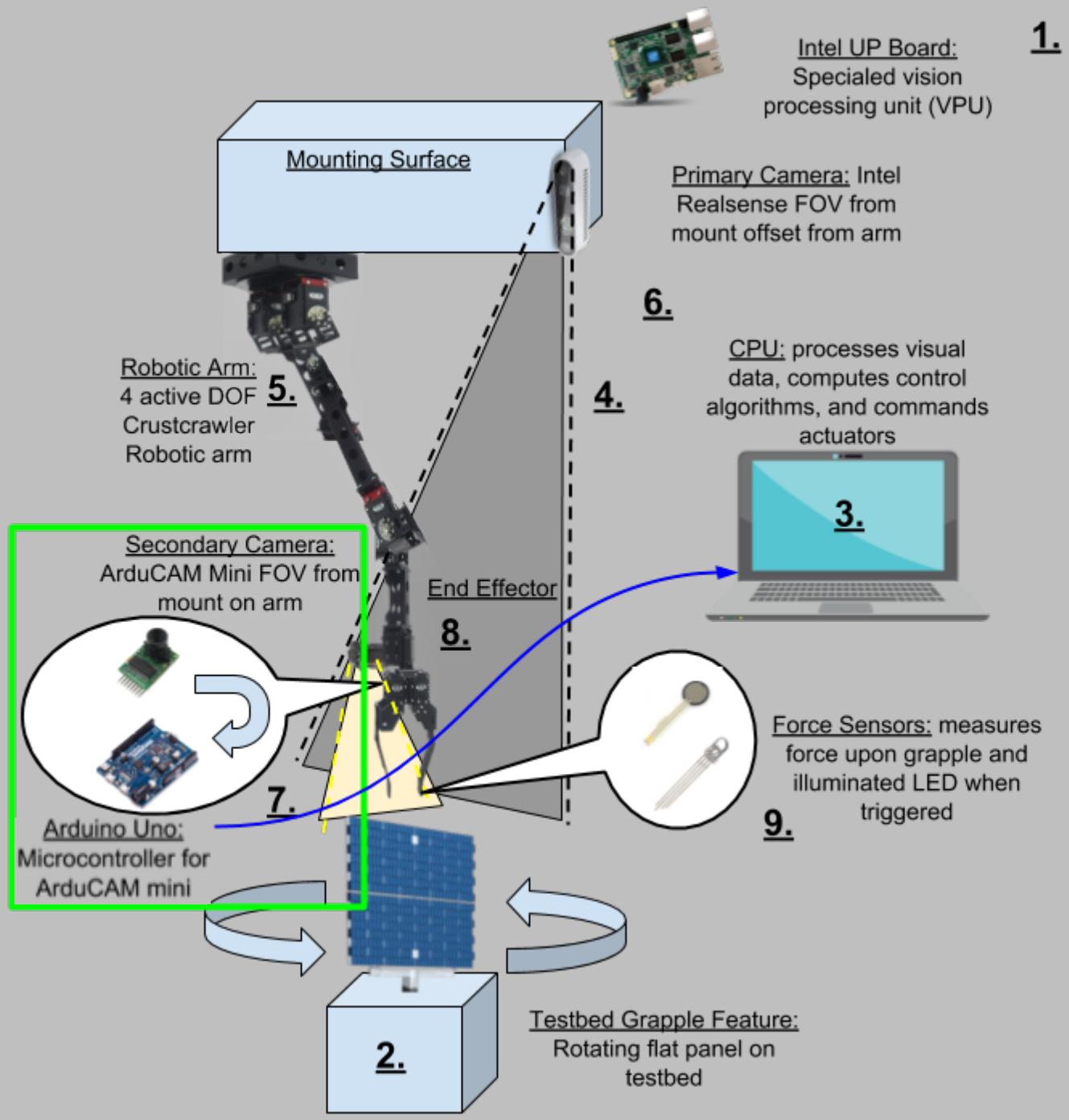

1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED

MEGACLAW
University of Colorado
Senior Design

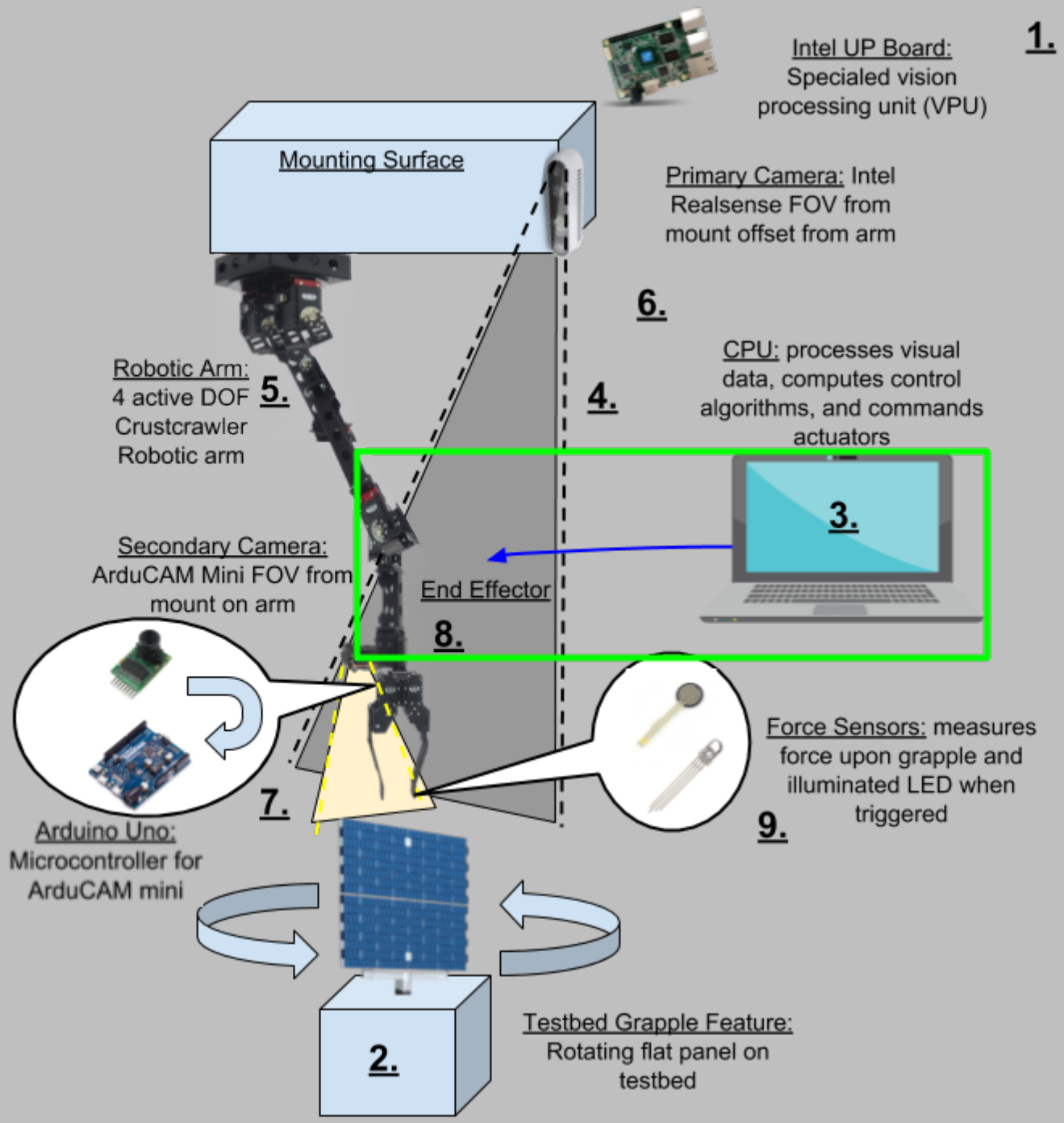



1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED

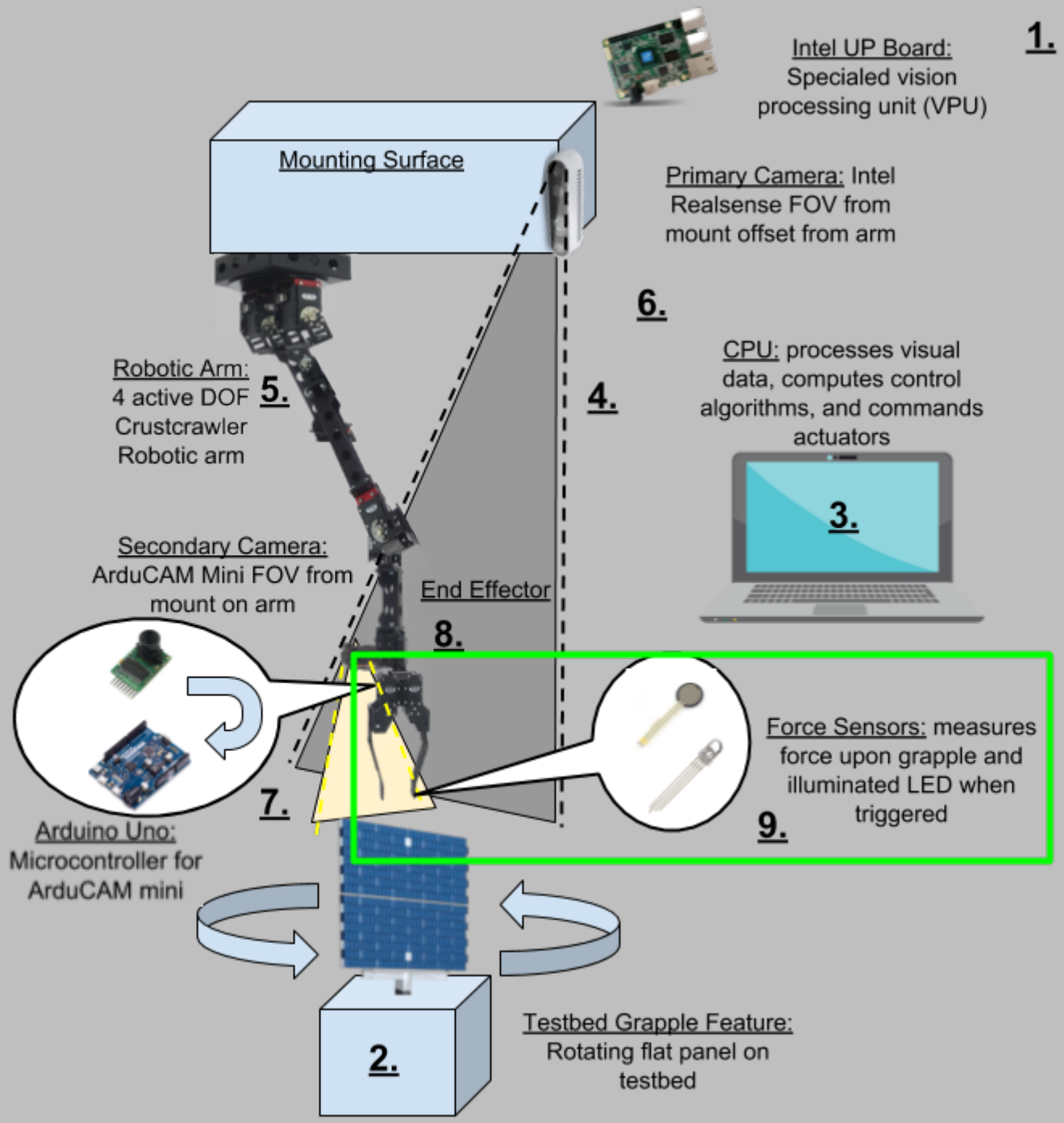
MEGACLAW
University of Colorado
Senior Design




1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED




1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED



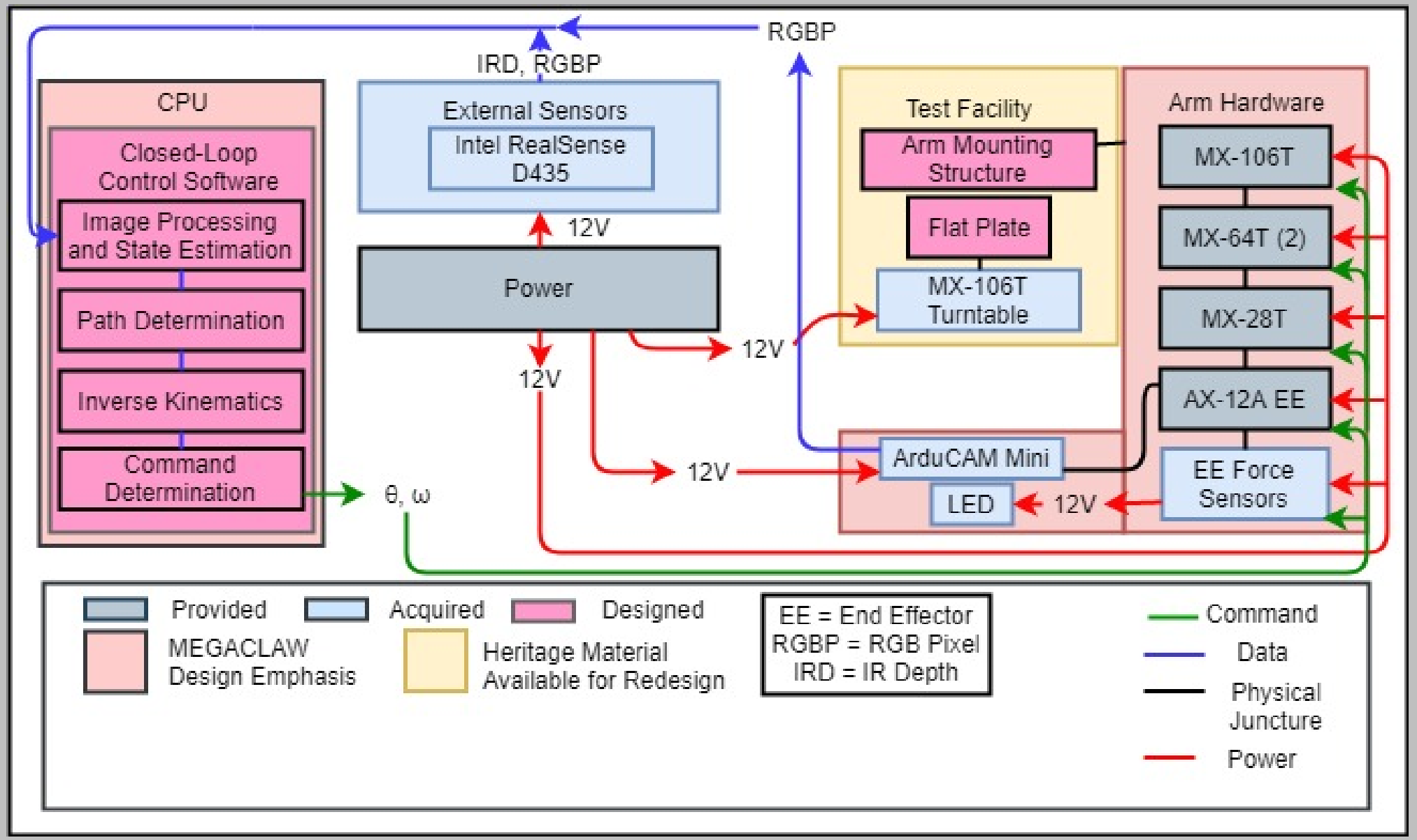

1. Primary sensor searches for pre-defined grappling feature and sends data through VPU to CPU
2. Algorithm identifies state of pre-defined grappling feature
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Visual sensor verifies end effector position is aligned with spin axis
7. Secondary sensor measures rotation of grapple feature and sends data to CPU
8. CPU commands rotation of end effector to match rotation necessary to grapple feature
9. Verifies grapple by triggering force sensor and illuminating LED

MEGACLAW
University of Colorado
Senior Design

# Functional Requirements

| Req. ID | Requirement |
|---|---|
| FR1 | The system shall operate with a closed loop response, based on data received from decoupled sensors |
| FR2 | The end effector's final state shall be equal to that of the grapple point |
| FR3 | The system shall be operable in an Earth-based controlled environment which simulates Low-Earth Orbit |

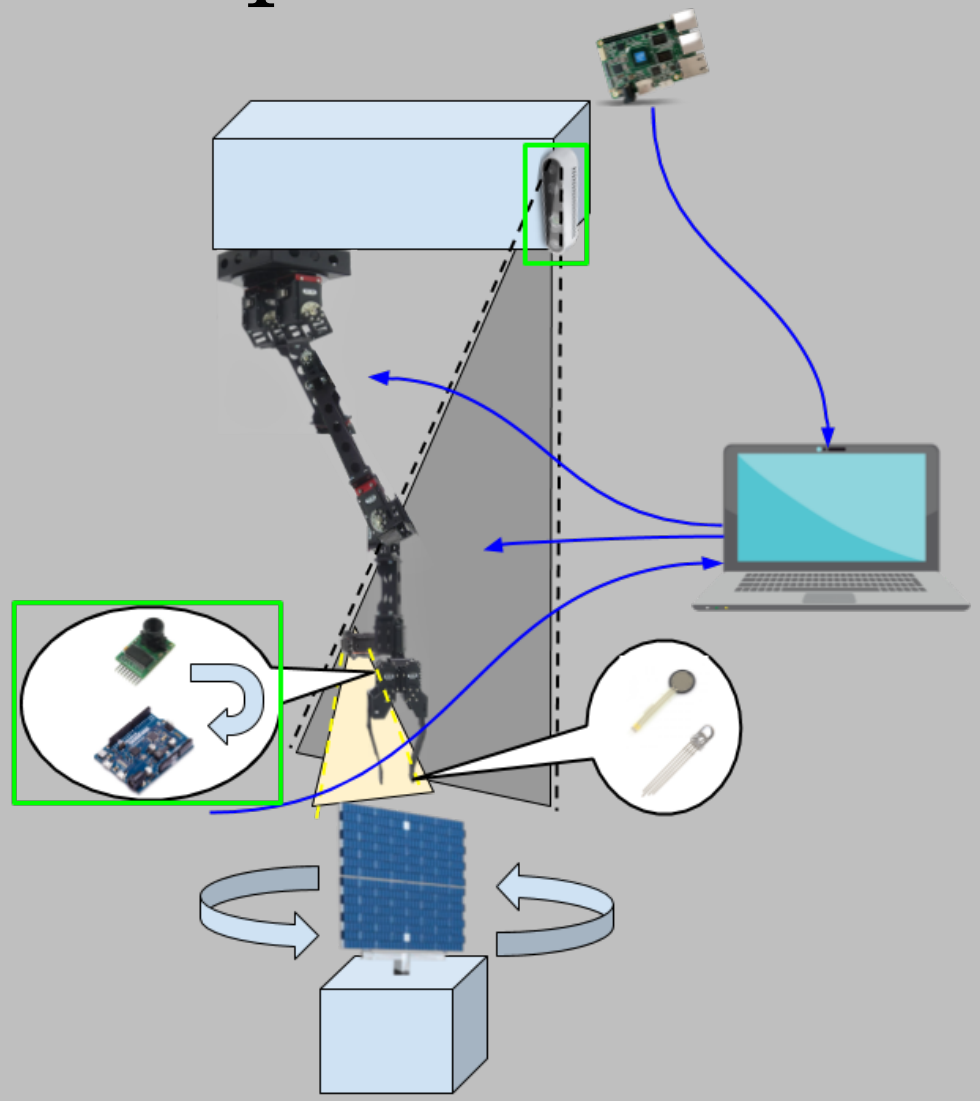
# Functional Block Diagram



SIERRA NEVADA CORPORATION

# Component Baseline Design Overview

# Component Baseline Design- Sensors



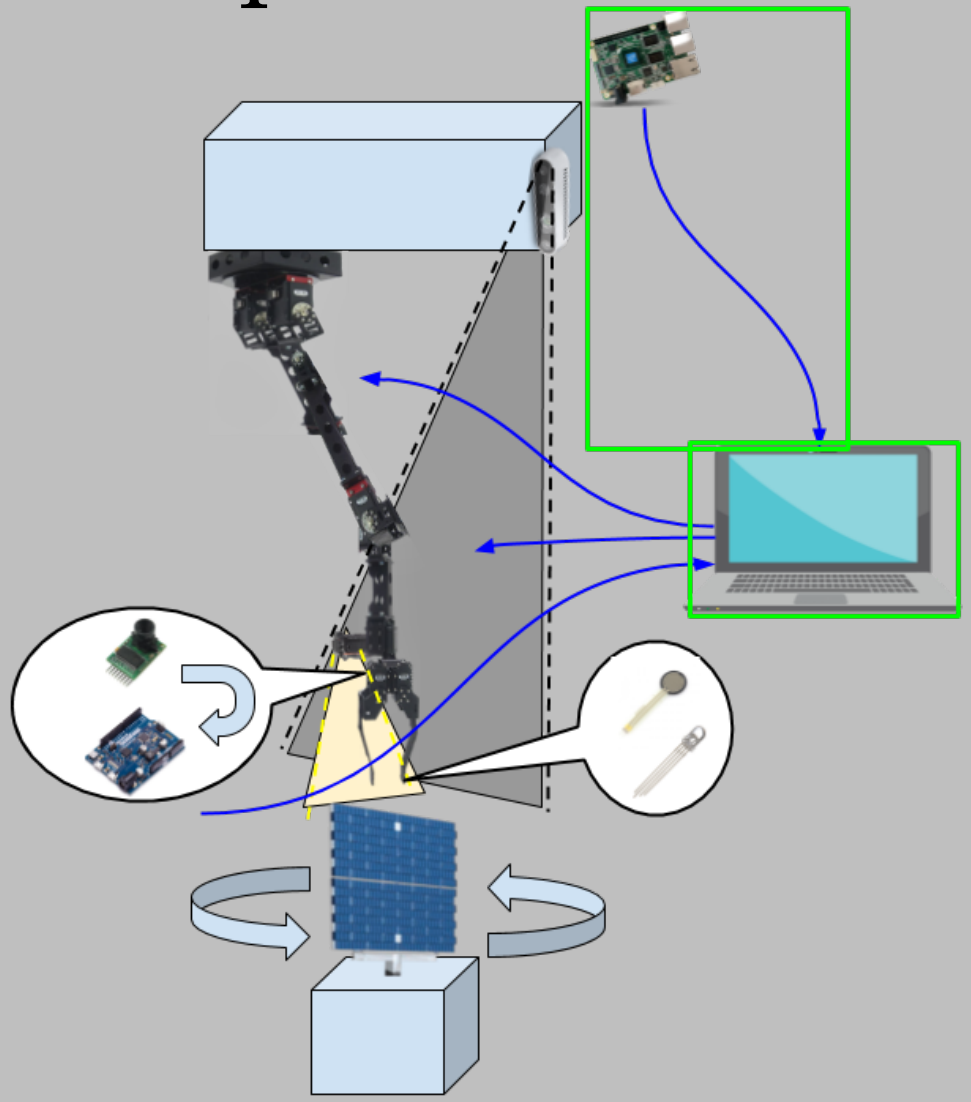
Intel RealSense Depth Camera D435

- The Intel D435 will use an RGB camera to identify the grapple point is within the field of view of 69.4° x 42.5° x 77° (+/- 3°).
- The Intel D435 will use an infrared projector to identify the position of the end effector and grapple point.
- The infrared projector will be used to determine the angular velocity until the grapple point is in the field of view of the ArduCAM mini (69.4° x 42.5°).

ArduCAM mini

- The ArduCAM mini will use an RGB camera to determine the determine the angular velocity of the grapple point.

# Component Baseline Design- Electronics



Intel Up Board

- Specialized video processing unit does preliminary data processing

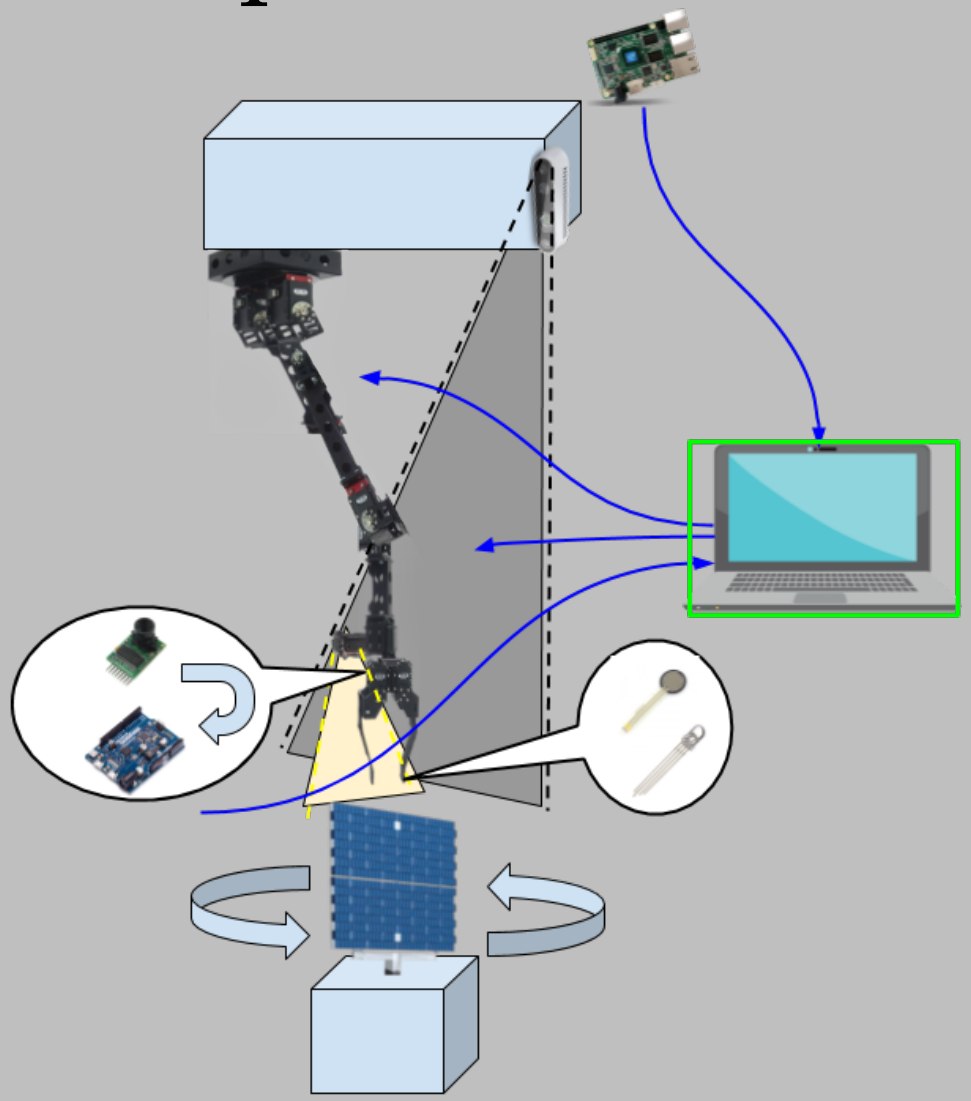
Arduino Uno

- Microprocessor that will process the signal of the ArduCAM mini

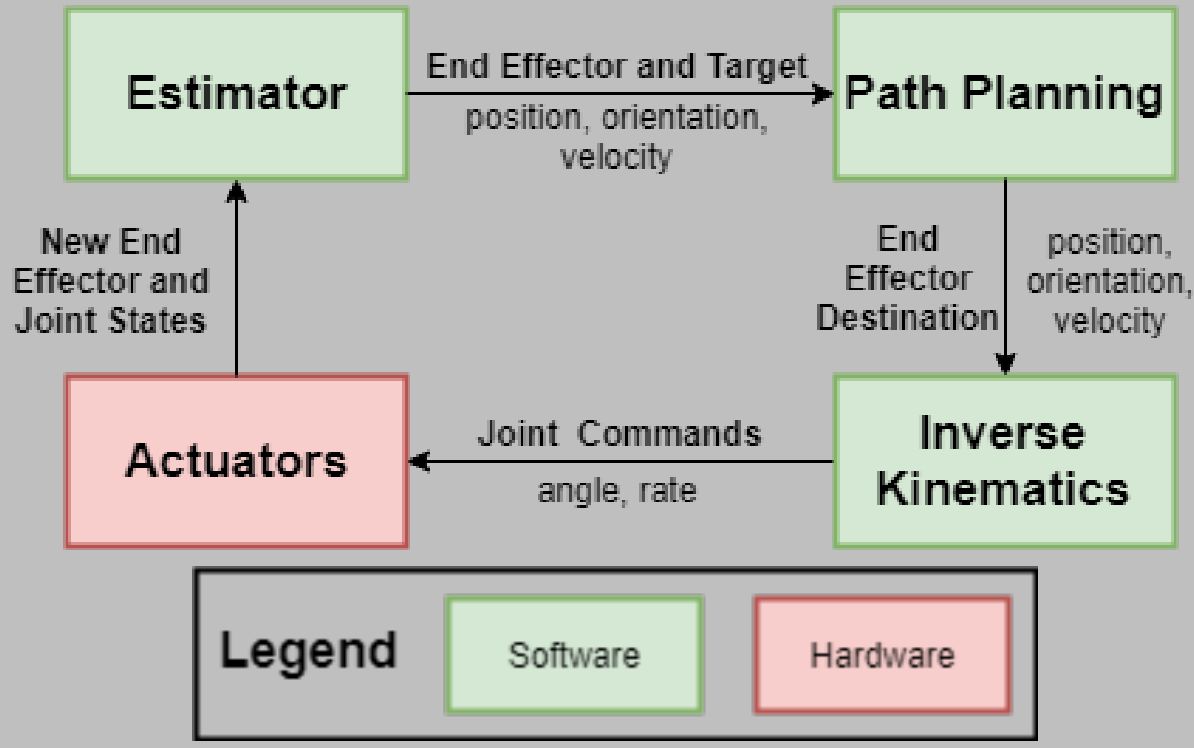
Laptop Computer

- Runs control software and additional image processing

# Component Baseline Design- Software

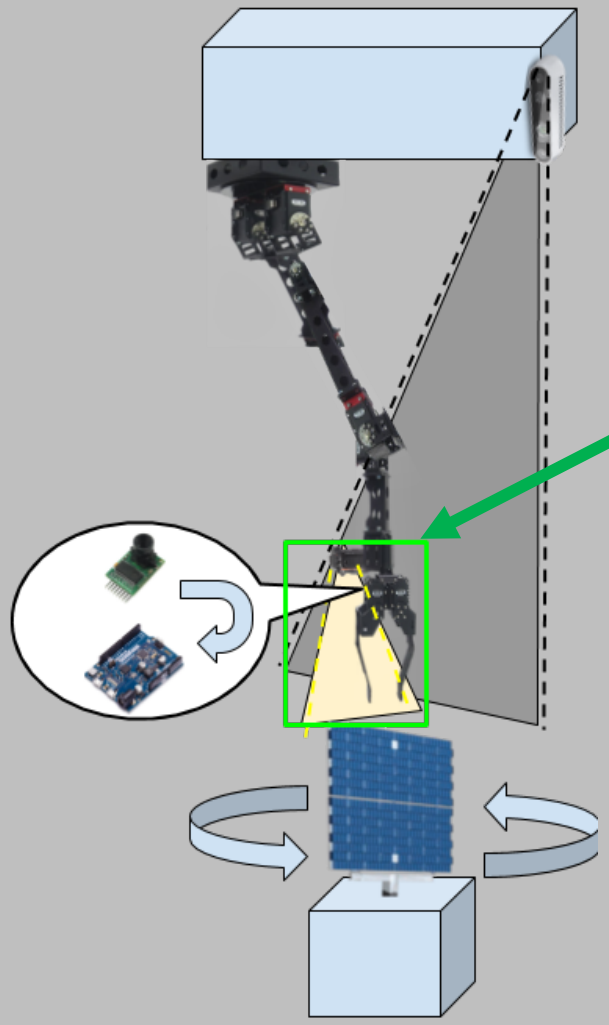


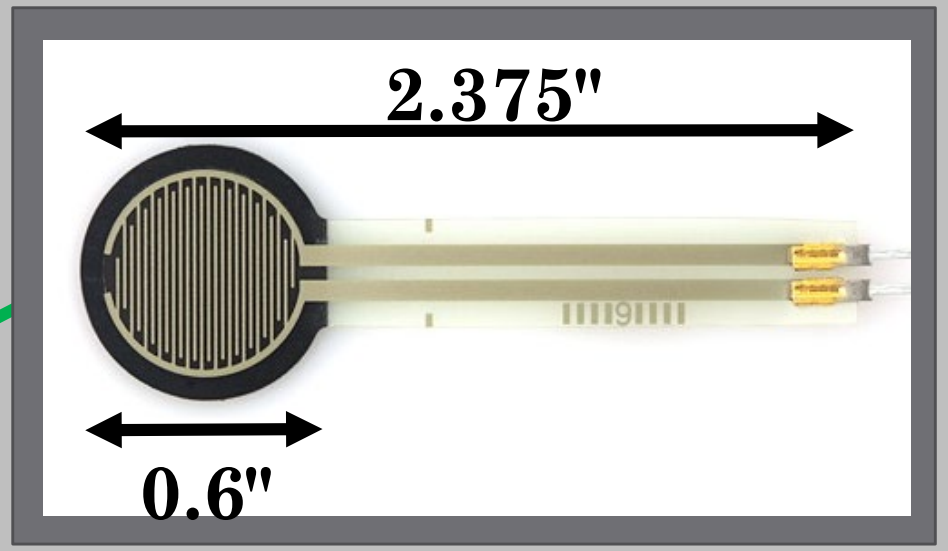
- The control and estimation software will be developed in **MATLAB** and **Simulink**
- **Robotic Operating System (ROS)** will be used to integrate the state estimation and control code



SIERRA NEVADA CORPORATION

# Component Baseline Design- Force Sensor

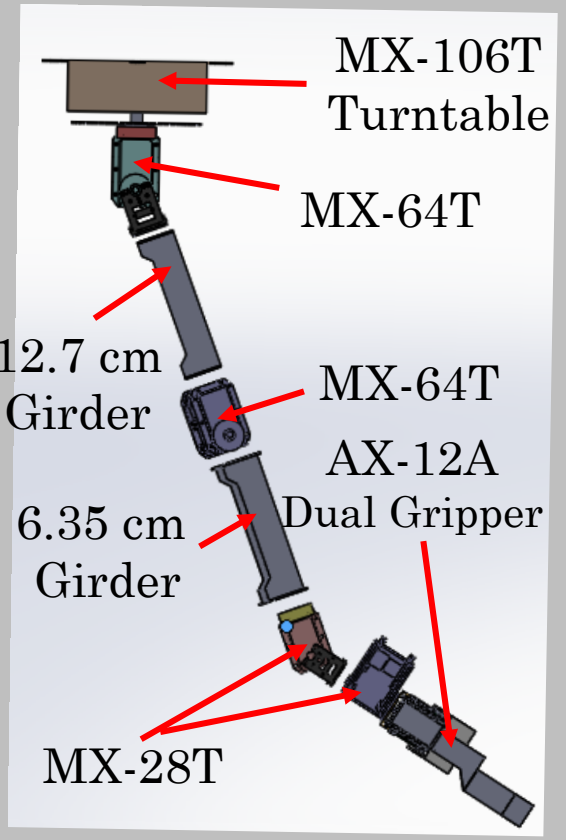





- Pololu 0.6" Force Sensitive Resistor
- 1MΩ resistance when no pressure applied
- ~100 kΩ --> ~200Ω resistance when pressure is applied
- **Minimum force detection: 0.196 N (0.04406 lbf)**

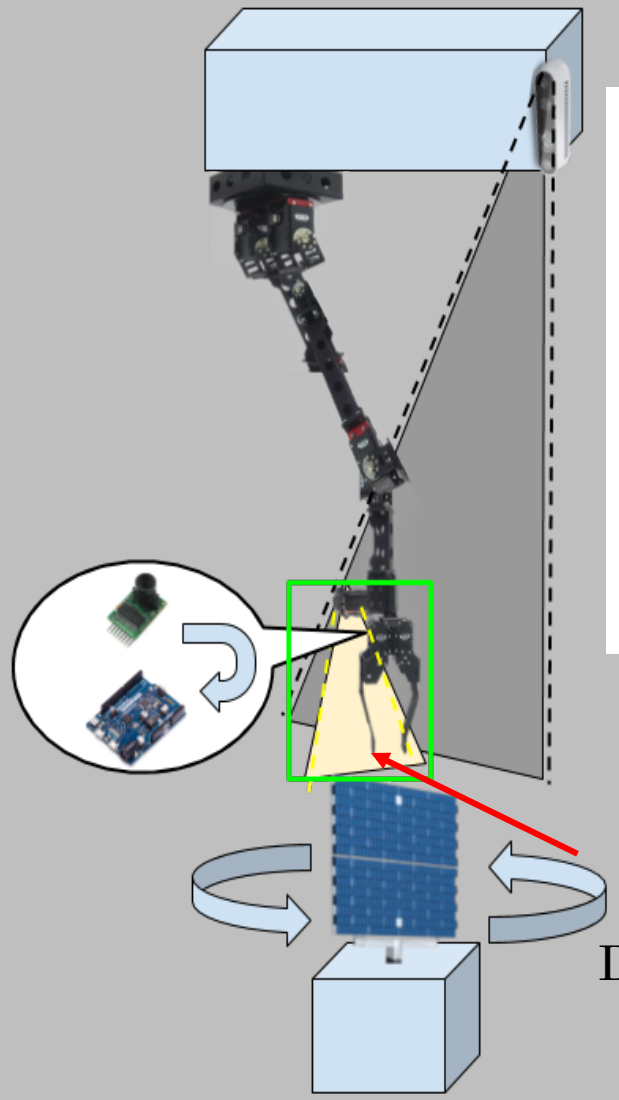
SIERRA NEVADA CORPORATION

# Component Baseline Design- Robotic Arm

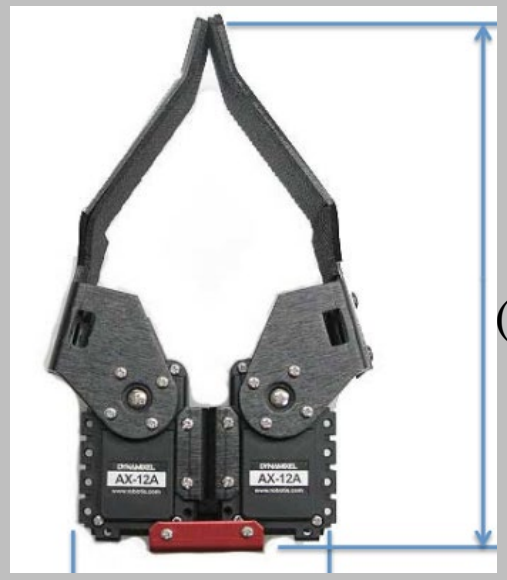



| Dyna-mixel Servo | Stall Torque [Nm] | Minimum Control Angle [°] | No Load Speed [RPM] | Operating Voltage [V] | Operating Temp.[°C] | Length [cm] |
|---|---|---|---|---|---|---|
| MX-106T | 8.4 | 0.088 | 45 | 12 | -5,80 | NA |
| MX-64T | 6 | 0.088 | 63 | 12 | -5,80 | 6.1 |
| MX-28T | 2.5 | 0.088 | 55 | 12 | -5,80 | 5.1 (vertical) 3.55 (horizontal) |

# Component Baseline Design- Robotic Arm End Effector



AX-12A
Dual Gripper

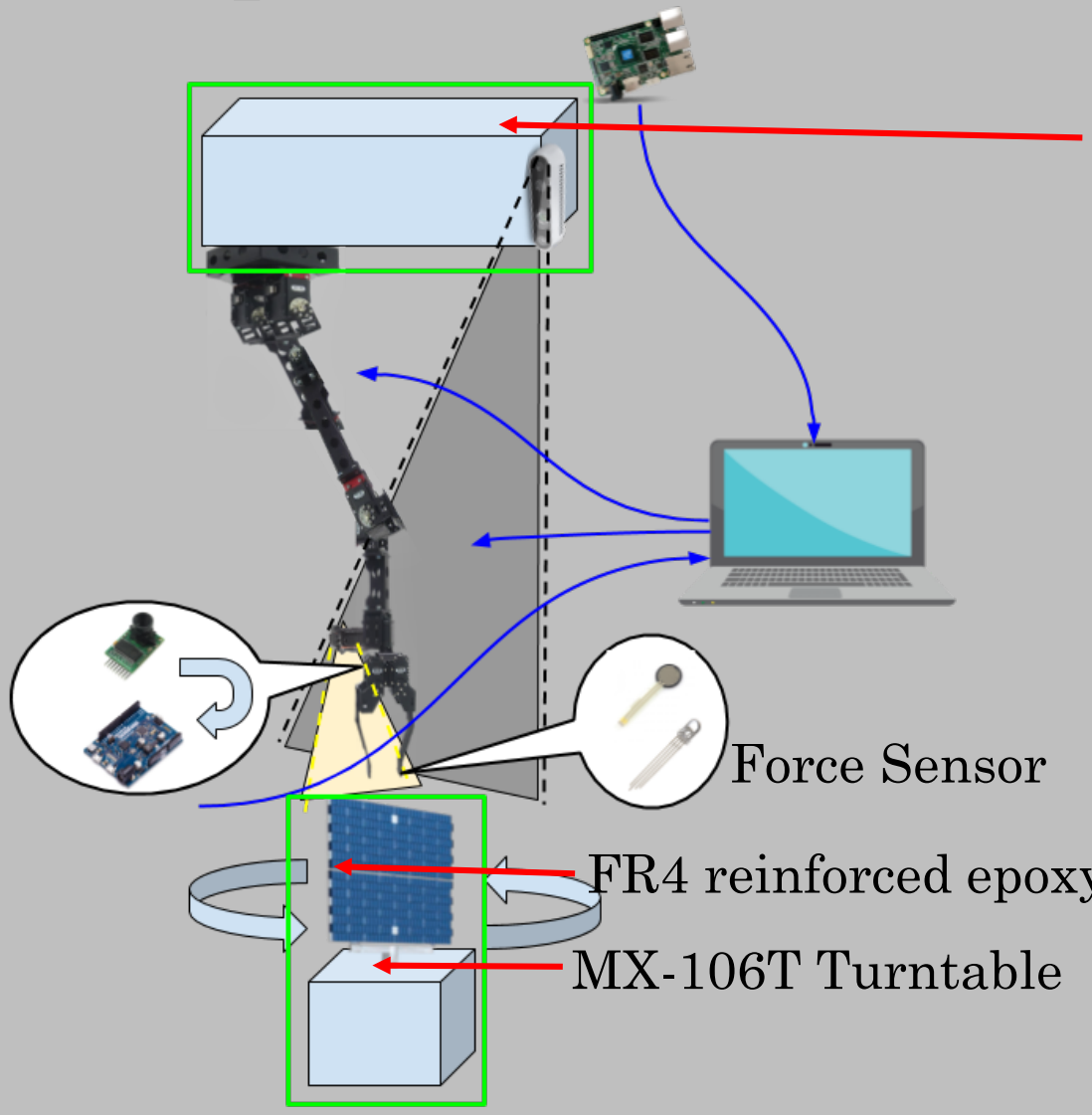


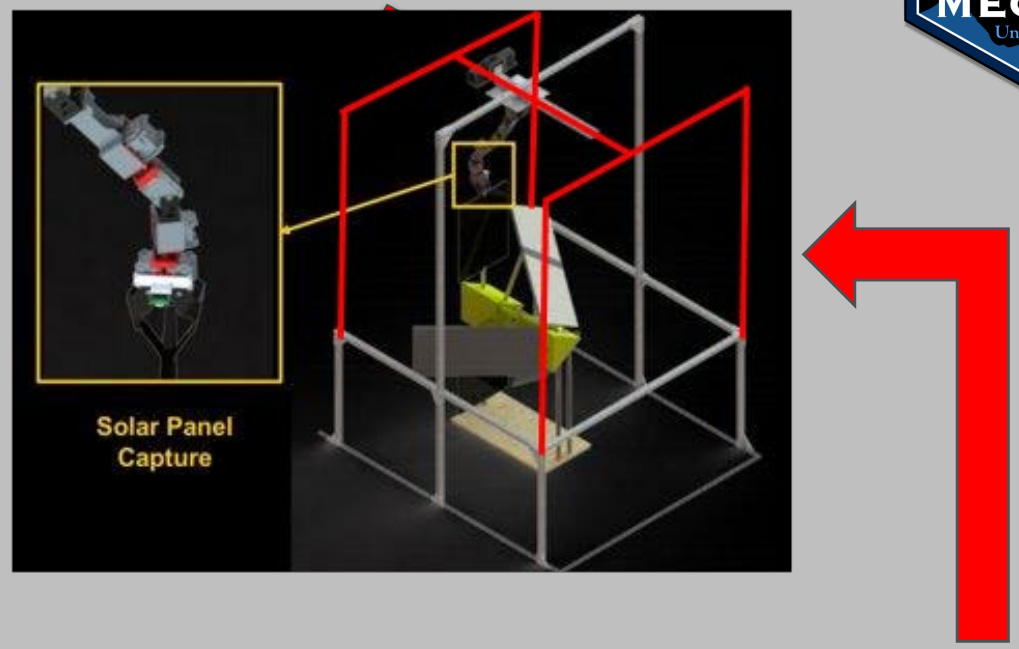
5.97 in.
(15.16 cm)


| Dyna-mixel Servo | Final Max Holding Torque [Nm] | No Load Speed [RPM] | Operating Voltage [V] | Operating Temp.[°C] |
|---|---|---|---|---|
| AX-12A | 1.6 | 59 | 12 | -5,80 |

SIERRA NEVADA CORPORATION

# Component Baseline Design- Test Bed

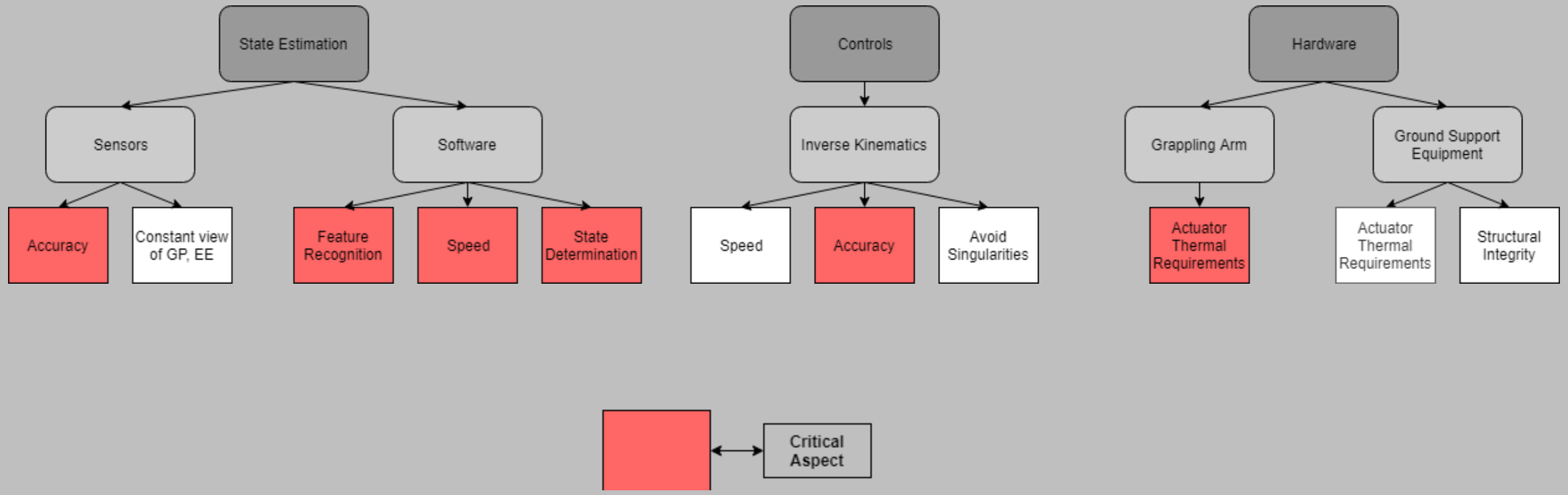



- Mounting structure will build upon heritage KESSLER system while implementing…
  - Reinforced structural support (addressed in feasibility studies)
  - Redesigned target motion system (MX-106T turntable)

MEGACLAW
University of Colorado
Senior Design

# Project Elements - Overview



# Critical Project Element Overview

| CPE | Description |
|---|---|
| **Sensors - Accuracies** | Sensors can measure EE and GP within combined error of 2.008 cm |
| **Software - Feature Recognition** | Software can identify and locate predetermined markings at a set distance |
| **Software - Speed** | Software can run at 0.916 Hz to support closed loop operation |
| **Software - State Determination** | Software can determine EE and GP positions to within a combined 2.95 cm. |
| **Controls - Accuracy** | Compute actuator commands with a 15mm to grapple the target |
| **Actuation of Robotic Arm and Ground Support Equipment** | Commands must be carried out under 255s |

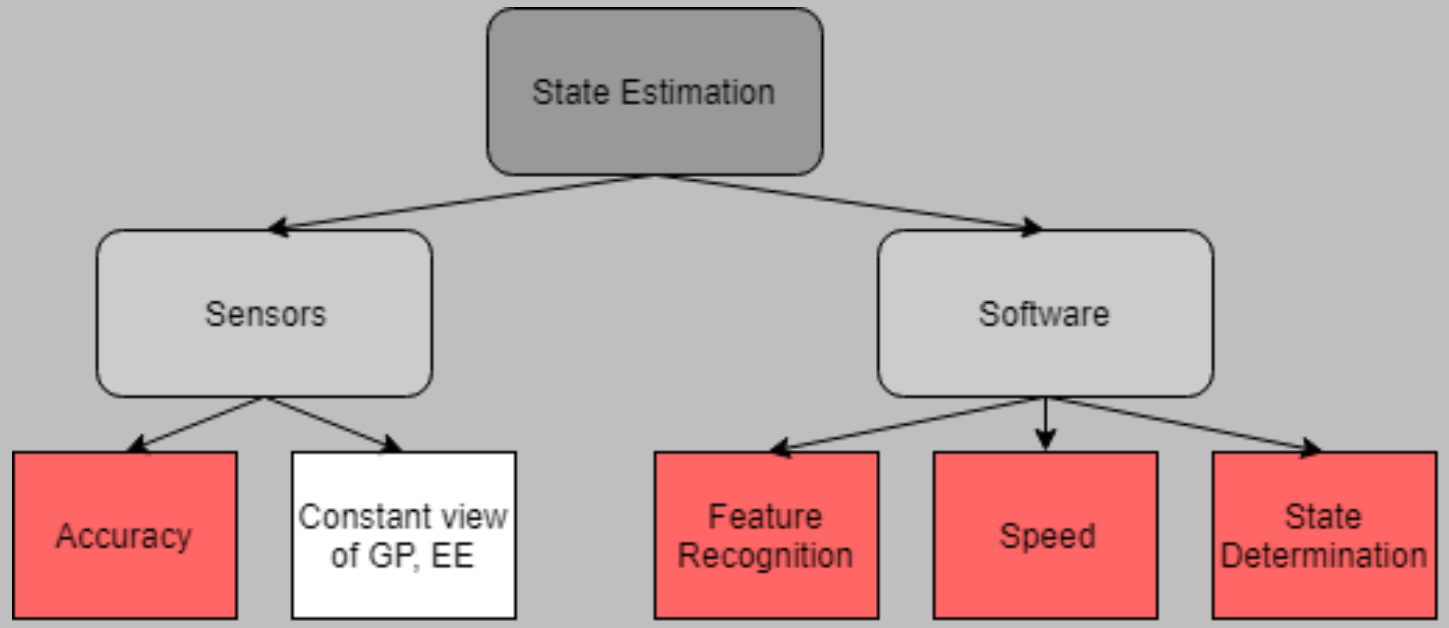
SIERRA NEVADA CORPORATION

# Feasibility Studies

## State Estimation, Controls, and Hardware

SIERRA NEVADA CORPORATION

# CPE 1: State Estimation



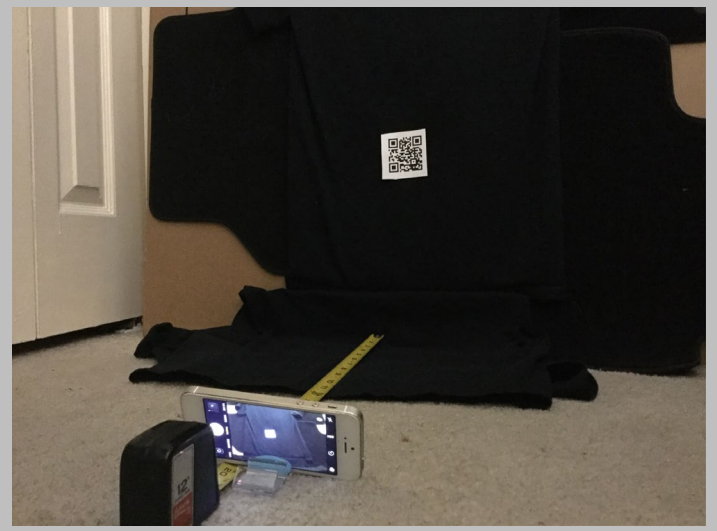

Use MATLAB to estimate the states of the end effector and target

# State Estimation

FR 2 The end effector's final state shall be equal to that of the grapple point

**DR 2.1 Sensors shall determine whether grapple point is within the field of view.**

- The Intel D435's RGB camera will be used to identify the grapple point.
- The grapple point will have fiducial markers allowing an RGB sensor to identify it.
- The experiment set a camera in a position 0.3 m down and 0.5 m away, which is a position representative of the mission.



Experiment Setup

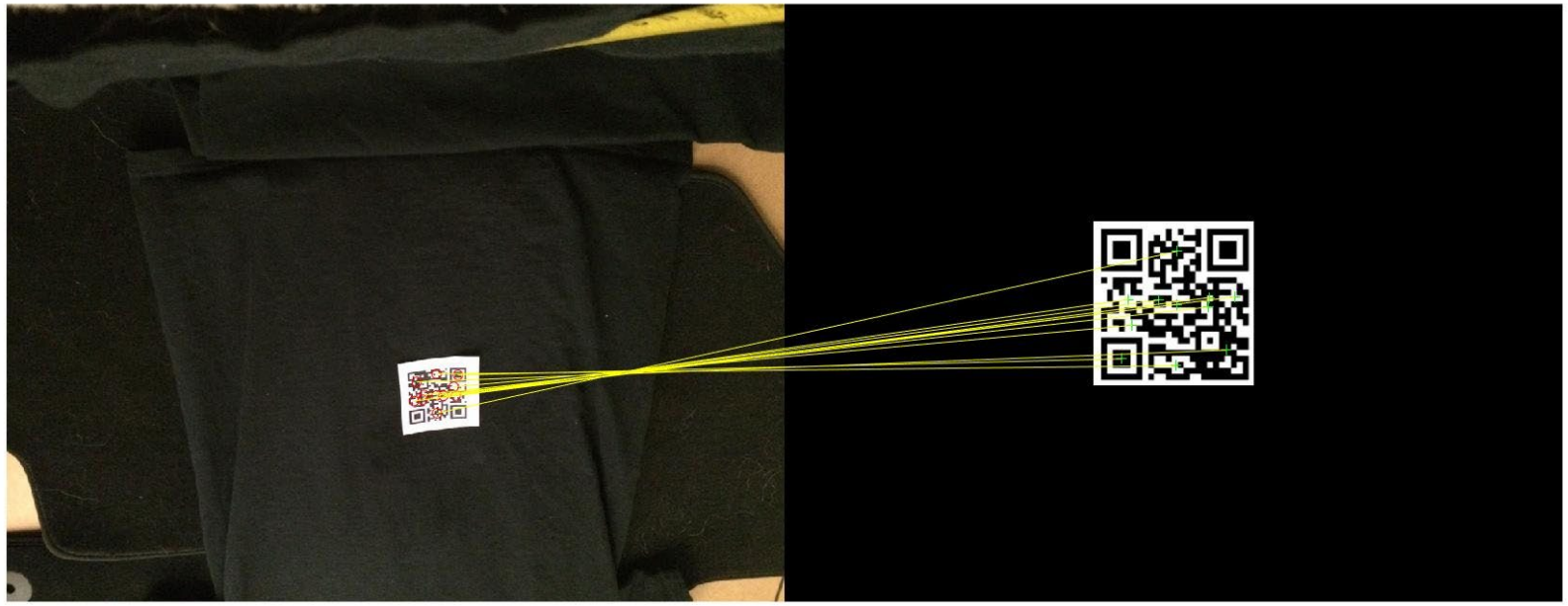
Fiducial marker used as the calibration image.

Features of fiducial marker

# State Estimation

- The calibration image and simulated view are imported into MATLAB. The software identifies features in the calibration image and compares them to features in the simulated view. The matched features are shown below.
- The simulated view has a resolution of 1224x918. This is much lower than the RGB camera on the Intel D435 (1920x1080).

Simulated view



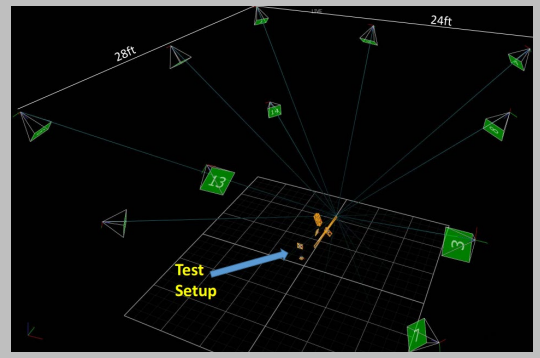
Features found in scene image confirms the object is identified. Therefore, feature recognition is feasible ✓

# State Estimation

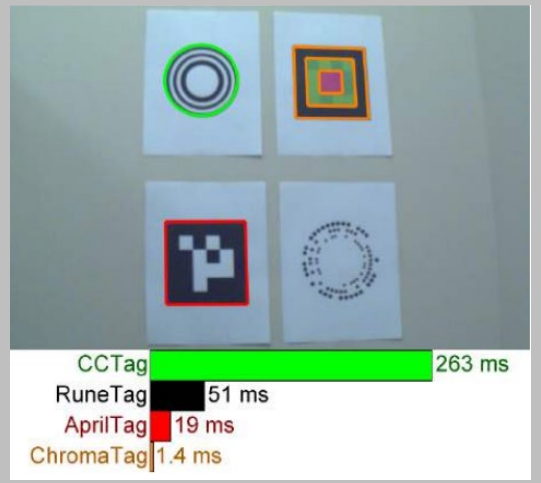
## Feasibility of Achieving Feedback rates

- FR1: The system shall operate with a closed loop response, based on data from decoupled sensors
- **DR 1.1.1: Image processing shall operate at five times the Nyquist frequency of the target motion: 0.916 Hz**
- Using MATLAB's basic feature recognition on fiducial markers: achieved 4.1 Hz (1280x720 resolution image, used by D435)
  - MATLAB's CV Toolbox isn't optimized for localizing specific types of fiducial markers
  - Fiducial markers tend to be designed with specific recognition methods in mind
- CASCADE (2016-2017)
  - Achieved target state determination frequency of 20 Hz using a large enclosure of multiple cameras and reflective spheres on object corners

**4.1 Hz > 0.916 Hz. State estimation at required rate feasible.**




*CASCADE vision system*



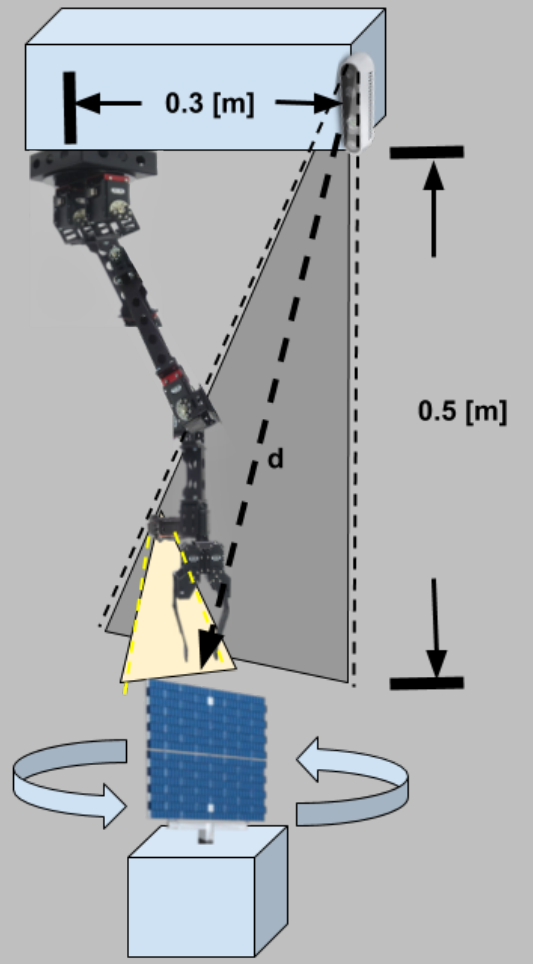

*Results of third-party study on fiducial marker identification*

# State Estimation

FR 2 The end effector's final state shall be equal to that of the grapple point

**DR 2.4.2 Position of EE shall be determined within 4.95cm**

**DR 2.4.3 Position of GP shall be determined within 4.95cm**




Intel documentation states that the accuracy of the IR projector for D400 series cameras is less than one percent of the distance from the object.

The grapple point is located is within 0.5 meters of the base of the robotic arm.

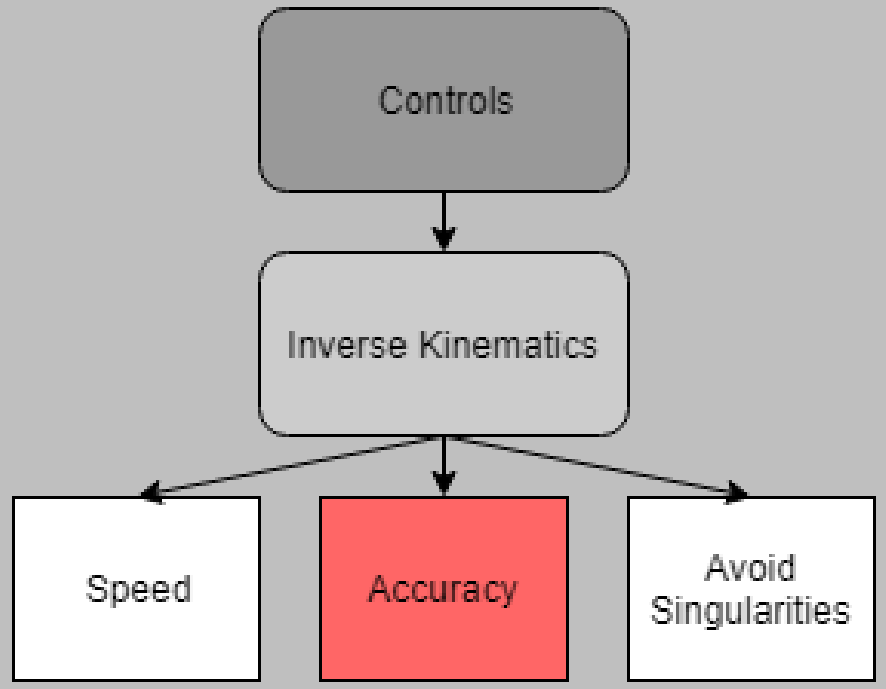
The worst case is parallel to the sensor on the far side. The total distance from the sensor would be 0.8 m. This results in a maximum error or 8 mm.

A more common case would be straight in front of the robotic arm base, shown in the figure. The total distance would be 0.583 m. This results in a maximum error of 5.8 mm.

An error of 8 mm is significantly lower than the required minimum of 4.95 cm, therefore the Intel D435's accuracy of the determined positions is feasible.

SIERRA NEVADA CORPORATION

# CPE 2: Controls Software

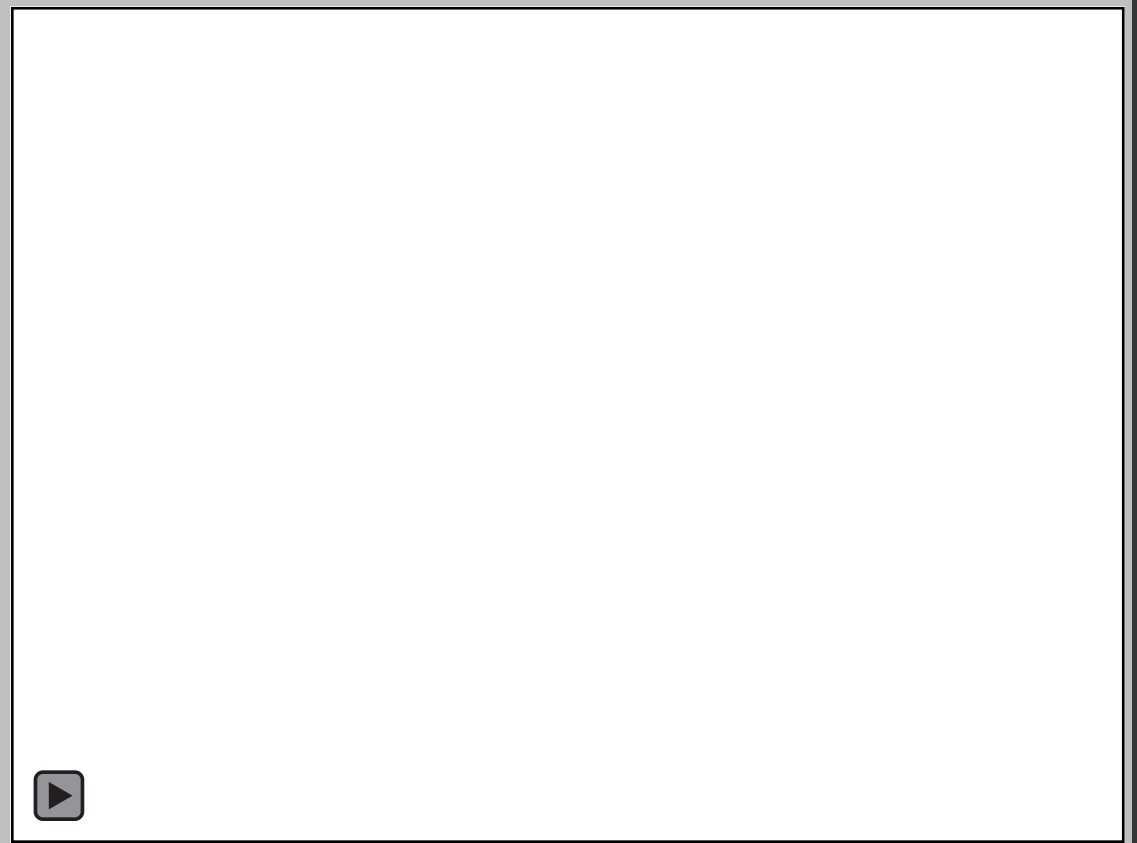



Solve inverse arm kinematics along the path for desired actuator angles and angular velocities

# SOFTWARE: Inverse Kinematics Accuracy Feasibility Analysis

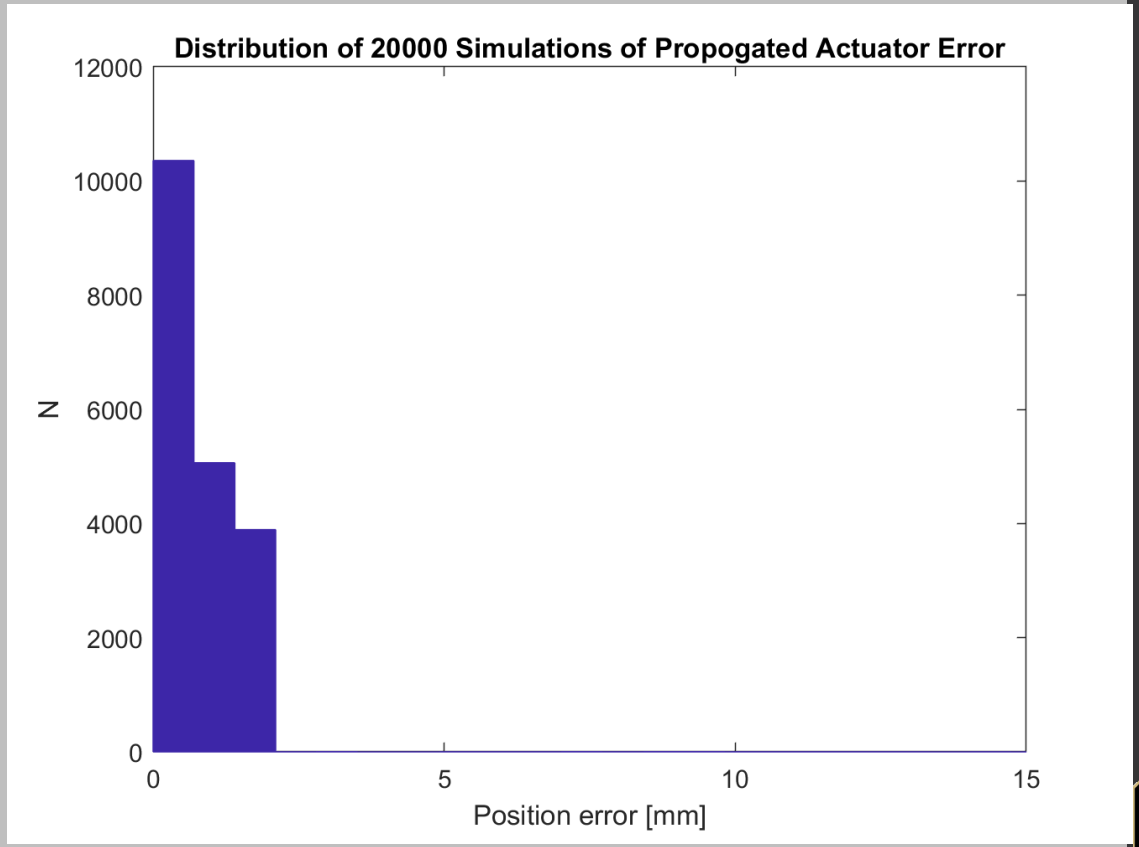
- FR1: The system shall operate with a closed loop response, based on data from decoupled sensors
- **DR 1.1.2 Inverse kinematics shall be solved with 15 mm accuracy**

- Test process: Simulate inverse kinematics for 20000 possible end effector states to get joint positions.
- Solve forward kinematics for the joint positions + maximum actuator error
- Calculate magnitude of difference in end effector states to get error
- **ARM VIDEO GOES HERE (feel free to format however looks best)**



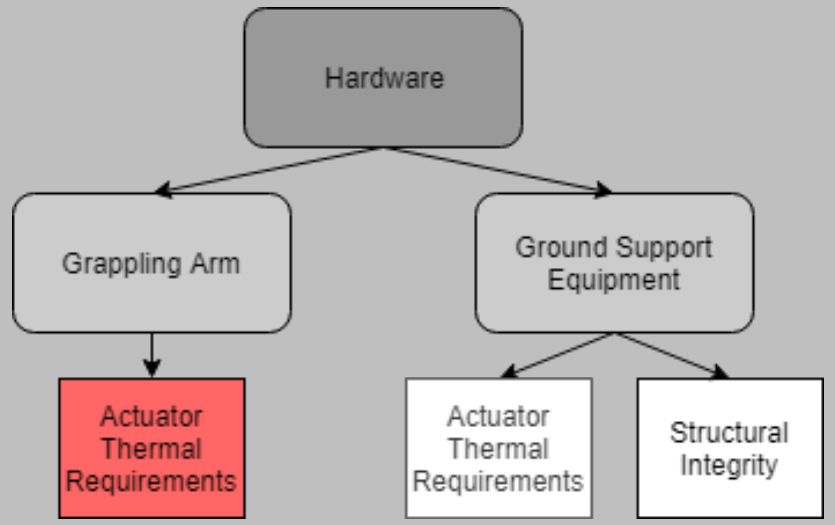
# SOFTWARE: Inverse Kinematics Accuracy Feasibility Results

- **DR 1.1.2 Inverse kinematics shall be solved with 15 mm accuracy**
- **Mean error = 7.519 mm**
- **Max error = 561.508 mm**
- **Percentage of error > 15mm = 3.26%**
- **Feasibility Pending**



SIERRA NEVADA CORPORATION

# CPE 3: Hardware



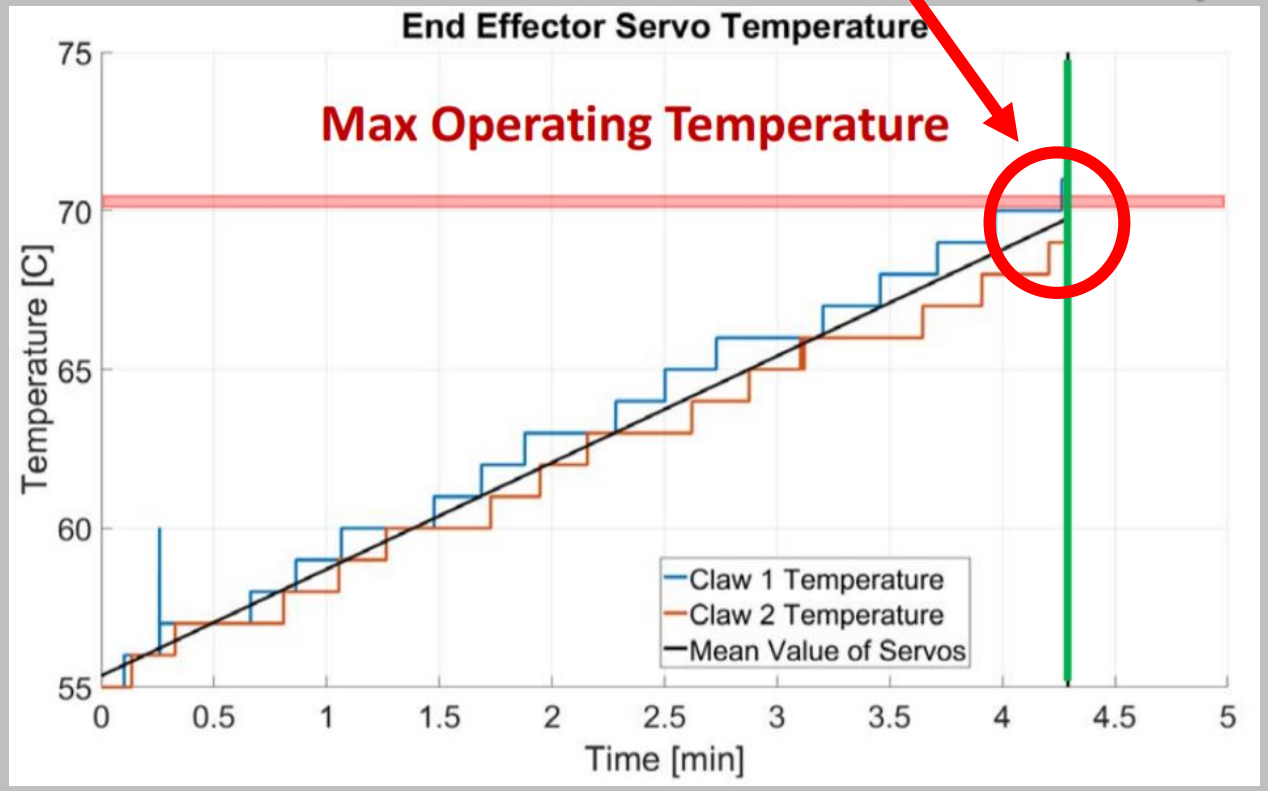

Actuation of servos on arm within thermal requirements

FR3: The system shall be operable in an Earth-based controlled environment which simulates LEO

**DR 3.1.1: Actuators shall not operate for no longer than 255 seconds.**

- Same servos as CASCADE baseline (1 MX-106T, 2 MX-64T, 2 MX-28T)
- Max operating temperature of 70 °C with factor of safety
  - Servos could operate for 4.25 minutes before reaching this threshold.
- Note: CASCADE experiment was performed with horizontally oriented arm
- Orienting the MEGACLAW arm vertically will increase time to reach max operating temperature.

**Max Operating Temp = 70°C**
**Operating time to reach 70°C = 4.25 min**




**End effector servo temperature study conducted by CASCADE.**

# DR 3.1.1: Actuators shall not operate for no longer than 255 seconds.

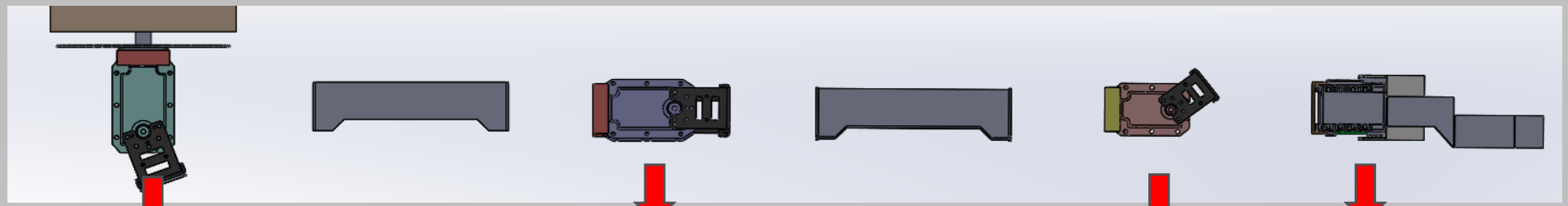
- **DR 1.1:** Control loop shall operate at a frequency of 0.368 Hz.
- 255 seconds for maximum operational time for single servo.
- Worst case scenario that each actuator needs to move 180°.
- **The thermal max operation is proven feasible with DR 1.1**

$$\frac{degree}{sec} = \frac{1.91489^o/comm}{2.71739sec/comm} = 0.704681^o/sec$$

$$0.704681^o/sec < 327.273^o/sec$$

FR3: The system shall be operable in an Earth-based controlled environment which simulates LEO

# 3.1.2: All arm servos shall be able to withstand maximum torque at 90°



$T_{MX64T} = 0.854[Nm]$ $T_{MX64T} = 0.522[Nm]$ $T_{MX28T} = 0.236[Nm]$ $T_{MX28T} = 0.139[Nm]$

| | MX-64T | 12.7cm Girder | MX-64T | 6.35cm Girder | MX-28T (Vertical) | MX-28T (Horizontal) | AX-12A Dual Gripper |
|---|---|---|---|---|---|---|---|
| Mass[kg] | 0.126 | 0.037 | 0.126 | 0.022 | 0.072 | 0.072 | 0.187 |
| Length[m] | 0.061 | 0.127 | 0.061 | 0.0635 | 0.051 | 0.0355 | 0.1516 |
| Stall Torque [Nm] | 6 | | 6 | | 2.5 | 2.5 | |
| Maximum Torque Experienced At 90° [Nm] | 0.854 | | 0.522 | | 0.236 | 0.139 | |
| Factor of Safety | ~7 | | ~11 | | ~10 | ~18 | |

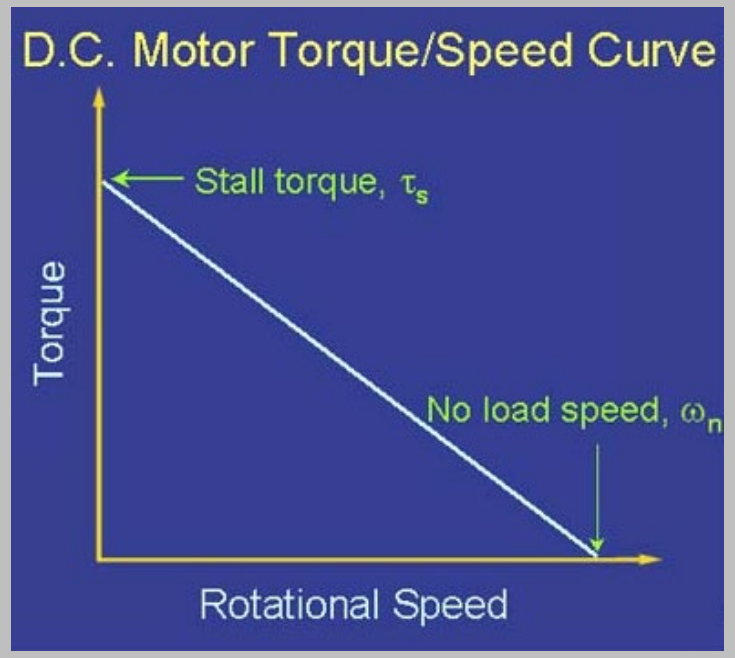
FR1: The system shall operate with a closed loop response, based on data received from decoupled sensors

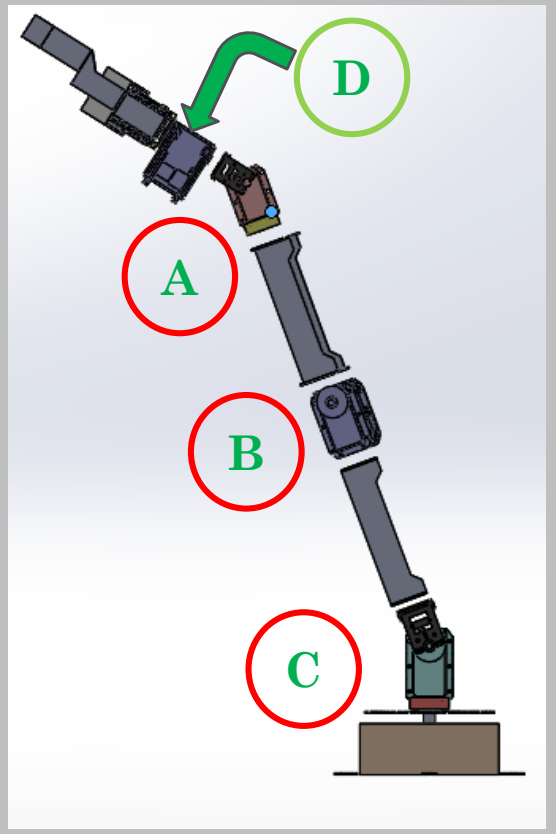
# DR 1.1: The control loop shall operate at a frequency of 0.368 Hz

- Worst case scenario: Actuators A, B and C must rotate 180° (π [rad]) upon each update of the control loop (every 2.7 s)
- Analyzing the actuator no load speed specified by the manufacturer (5.76 rad/s)
  an actuator under no load can rotate 180° in 0.55s

$$\frac{\omega_{ABC} = 5.76\ \frac{rad}{s}}{\pi\ rad} = 0.55\ s < 2.7\ \text{s}$$

- As shown in the previous feasibility study, stall torque is not approached for any of the actuators therefore assuming actuators are operating at a no load speed is a reasonable assumption



FR1: The system shall operate with a closed loop response, based on data received from decoupled sensors

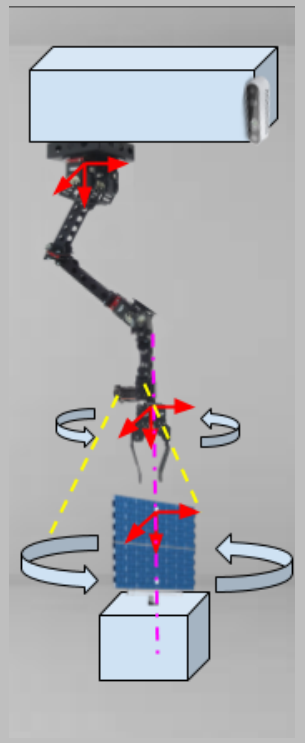
**DR 1.1: The control loop shall operate at a frequency of 0.368 Hz**

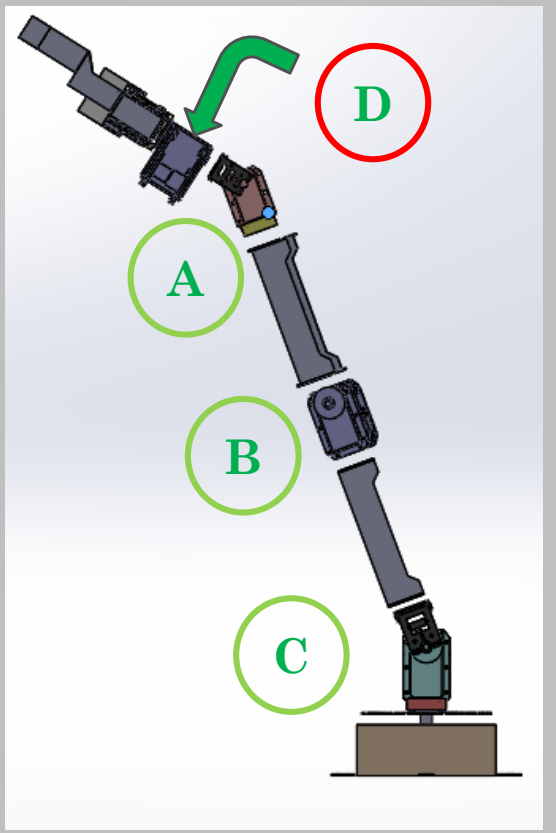
- Worst case scenario: Actuator D must rotate at the same rate as target object, $\omega_{GP} = 0.578\frac{rad}{s}$
- Analyzing the actuator no load speed specified by the manufacturer, $\omega_{EE} = 6.597\frac{rad}{s}$ the design is seen to be feasible

$\omega_{GP} < \omega_{EE(actuator\ D)}$ FS ~ 11 ✓

$\omega_{EE} = 6.597\frac{rad}{s}$

$\omega_{GP} = 0.578\frac{rad}{s}$





SIERRA NEVADA CORPORATION

# Summary & Strategy

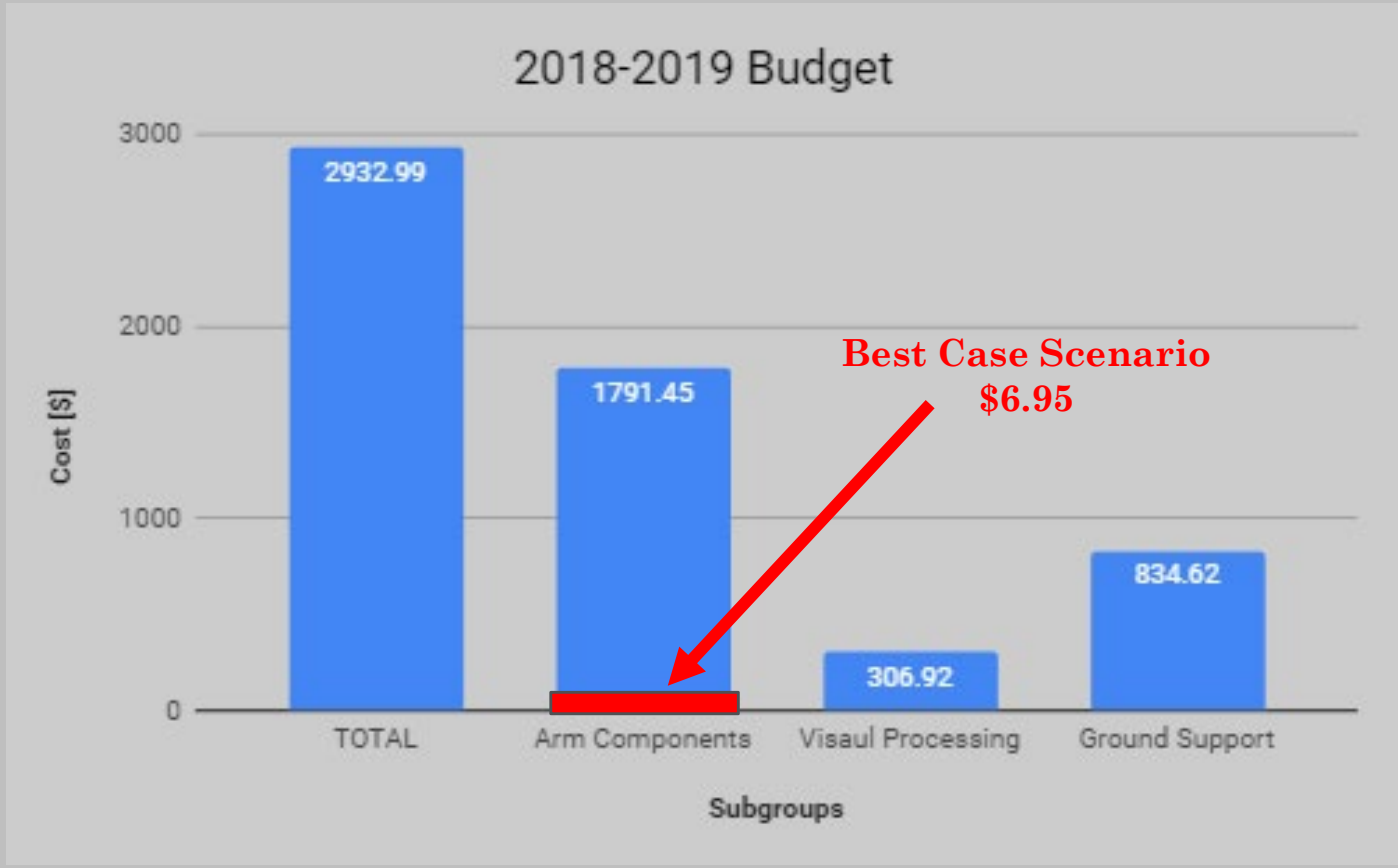
| CPE | Solution |
|---|---|
| Sensors - Accuracies | Feasible – Intel specifications(8mm) are significantly lower than the required accuracy (4.95cm) |
| Software - Feature Recognition | Feasible – An image with much lower resolution (1224x918) than the Intel D435 (1920x1080) can identify fiducial markers. |
| Software - Speed | Feasible – Testing image processing code shows that it meets time requirement |
| Controls - Accuracy | Feasibility Pending – Verify that inaccurate states are impossible by limiting possible joint angles |
| Actuation of Robotic Arm and Ground Support Equipment | Feasible – Imposing a 255s time limit on servo actuation prevents thermal failure |

SIERRA NEVADA CORPORATION

# Testing Facilities

- **VISION Lab:** Testing Environment
- **ITLL 150:** Testing Environment
- **SNC Shop Floor:** Disassembly of arm

# Financial Feasibility




- Starting budget: $5,000
- Worst-case scenario:
  - Total expenses: $2,932.99
  - Total with 10% margin: $3,226.39
  - Remaining budget: $1773.71
- Best-case scenario:
  - From worst-case, heritage hardware and components provided by SNC saves $1784.50 in arm components and $639.00 in ground support equipment.

SIERRA NEVADA CORPORATION

# Financial Feasibility

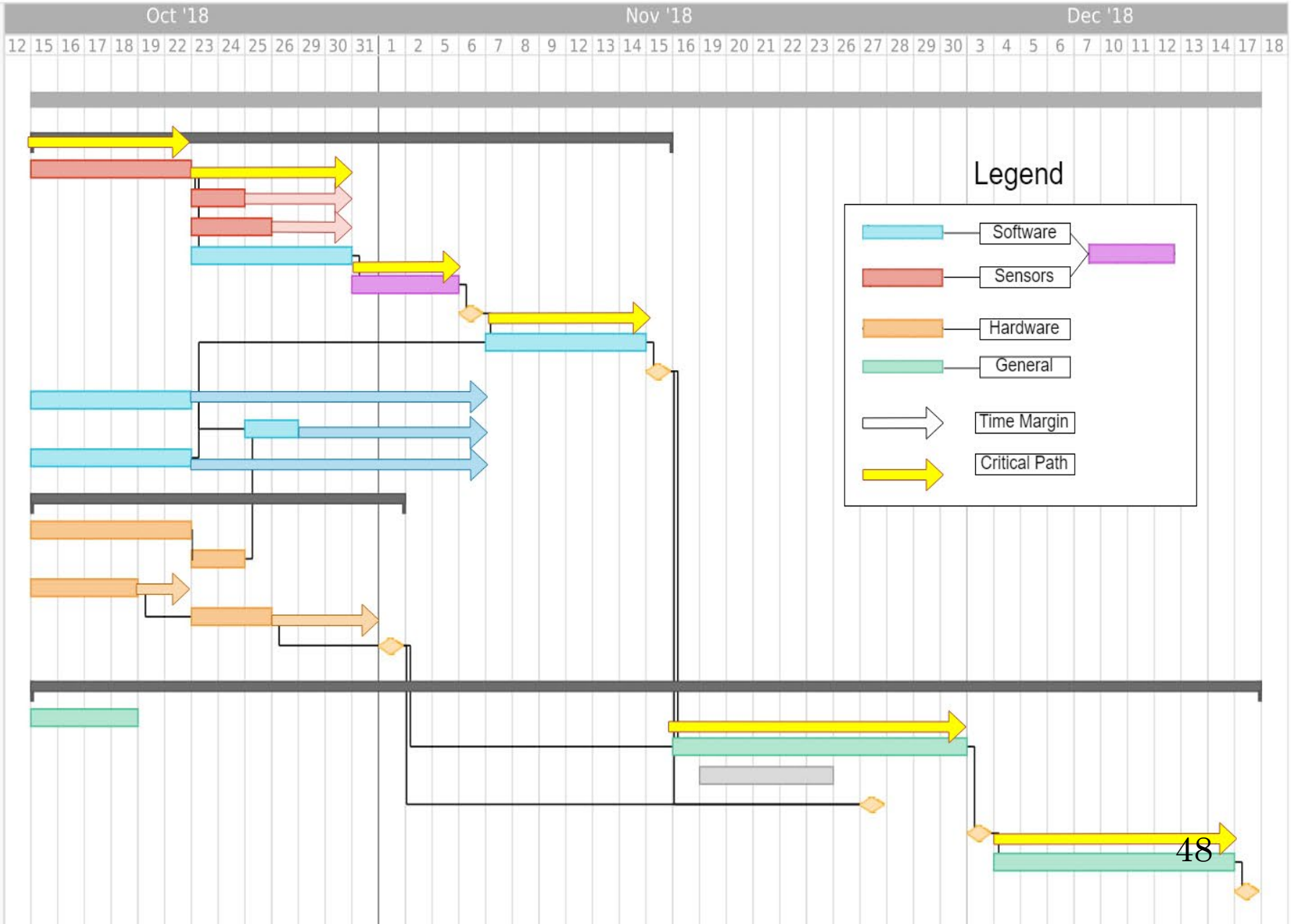
| Item | Quantity | Total Price |
|---|---|---|
| Arducam Mini Module | 1 | $15.99 |
| Arduino UNO | 1 | $22.00 |
| RGB tri-color LED | 1 | $8.95 |
| UP board | 1 | $79.99 |
| 1-½ x 96" Zinc-plated slotted angle rod | 230" | $74.50 |
| Dynamixel MX-64T servo | 2 | $599.80 |
| Dynamixel MX-106T servo | 1 | $493.90 |
| MX106trntbl | 1 | $639.00 |
| Dynamixel AX-12A servo | 1 | $189.00 |
| Dynamixel MX-28T servo | 1 | $439.80 |
| 12.7 cm Girder | 1 | $34.00 |
| 6.35 cm Girder | 1 | $28.00 |
| Intel RealSense D435 | 1 | $179.99 |

*Quantities in blue correspond to heritage hardware that will likely be reused.*

SIERRA NEVADA CORPORATION

# Gantt Chart

| | | |
|---|---|---|
| **MEGACLAW** | **0h** | **0%** |
| **Sensors, Software** | **0h** | **0%** |
| Procure D435, ArduMini | 0 | 0% |
| Set Up RealSense SDK Software | 0 | 0% |
| Data Reception Testing | 0 | 0% |
| Write Fiducial Marker Code | 0 | 0% |
| Feature Recognition Testing | 0 | 0% |
| Feature Recognition Design Review | 0 | 0% |
| Develop State Determination Softwa... | 0 | 0% |
| State Determination Software Design... | 0 | 0% |
| Inverse Kinematics Code Developme... | 0 | 0% |
| Inverse Kinematics Testing | 0 | 0% |
| Evaluate / Define Software Architect... | 0 | 0% |
| **Hardware** | **0h** | **0%** |
| Acquire / Disassemble Hardware (fr... | 0 | 0% |
| Test Heritage Actuators | 0 | 0% |
| Target CubeSat Actuator TS | 0 | 0% |
| Finalize Ground Station Equipment D... | 0 | 0% |
| Ground Station Equipment Design R... | 0 | 0% |
| **General / Logistics** | **0h** | **0%** |
| PDR Feedback Review | 0 | 0% |
| Prepare CDR Presentation | 0 | 0% |
| Fall Break | 0 | 0% |
| Internal Progress Report (Nov) | 0 | 0% |
| Critical Design Review (Due) | 0 | 0% |
| Prepare Fall Final Report | 0 | 0% |
| Fall Final Report Due | 0 | 0% |



SIERRA NEVADA CORPORATION

# Next Steps

| | |
|---|---|
| 1. | Gather CrustCrawler components from SNC |
| 2. | Finish software accuracy and feasibility studies |
| 3. | Finalize ground support equipment design |
| 4. | Select rotating actuator for target CubeSat model (new baseline component) |
| 5. | Conduct risk management study |
| 6. | More in depth modeling of kinematics and state estimation |
| 7. | Develop electrical/power model |

SIERRA NEVADA CORPORATION

# Acknowledgements

Special thanks to...

- Francisco Lopez Jimenez
- Trudy Schwartz
- Bobby Hodgkinson
- Matt Rhodes
- Marcus Holzinger
- TJ Sayer
- Christine Reilly
- Ian Cooke
- Dale Lawrence
- Christoffer Heckman
- Charlie Labonde

# References


- Barinka, Lukas. “Inverse Kinematics - Basic Methods.” *Inverse Kinematics - Basic Methods*, 2012, old.cescg.org/CESCG-2002/LBarinka/paper.pdf.
- Schaal, Stefan. “Jacobian Methods for Inverse Kinematics and Planning.” *Jacobian Methods for Inverse Kinematics and Planning*, 2015, Jacobian Methods for Inverse Kinematics and Planning.
- Corke, Peter I. *Visual Control of Robots: High-Performance Visual Servoing*. Research Studies Press, 1997.
- Siciliano, Bruno, and Oussama Khatib. *Springer Handbook of Robotics*. Springer, 2016.
- *CrustCrawler Robotics*. 2018, www.crustcrawler.com/.
- “Computer Vision System Toolbox.” *MATLAB & Simulink*, www.mathworks.com/products/computer-vision.html.
- *"Computer Vision with MATLAB" Webinar Demo Files - File Exchange - MATLAB Central*, www.mathworks.com/matlabcentral/fileexchange/40079-january-2013-computer-vision-with-matlab-webinar-demo-files.

# References


- “ArduCAM Mini Released.” *Arduino Based Camera*, 8 July 2017, www.arducam.com/arducam-mini-released/.
- “Intel® RealSense™ Depth Camera D435.” *Intel*, https://click.intel.com/intelr-realsensetm-depth-camera-d435.html
- “Depth Resolution of Intel® RealSense™ Depth Camera D435 and Intel®...” *Intel*, www.intel.com/content/www/us/en/support/articles/000026260/emerging-technologies/intel-realsense-technology.html.
- "Intel® RealSense™ Depth Camera D400-Series" Datasheet https://www.mouser.com/pdfdocs/Intel_D400_Series_Datasheet.pdf
- Gross, R J. “Robotis Dynamixel Actuators in LabVIEW.” *Robotis Dynamixel Actuators in LabVIEW - National Instruments*, 13 Jan. 2011, www.ni.com/white-paper/12557/en/.
- "Intel® RealSense D400 Series Product Family Datasheet", *Intel,* https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf


# Questions?



# Back-up Slides

# Control Loop Frequency Derivation

## Assumptions:

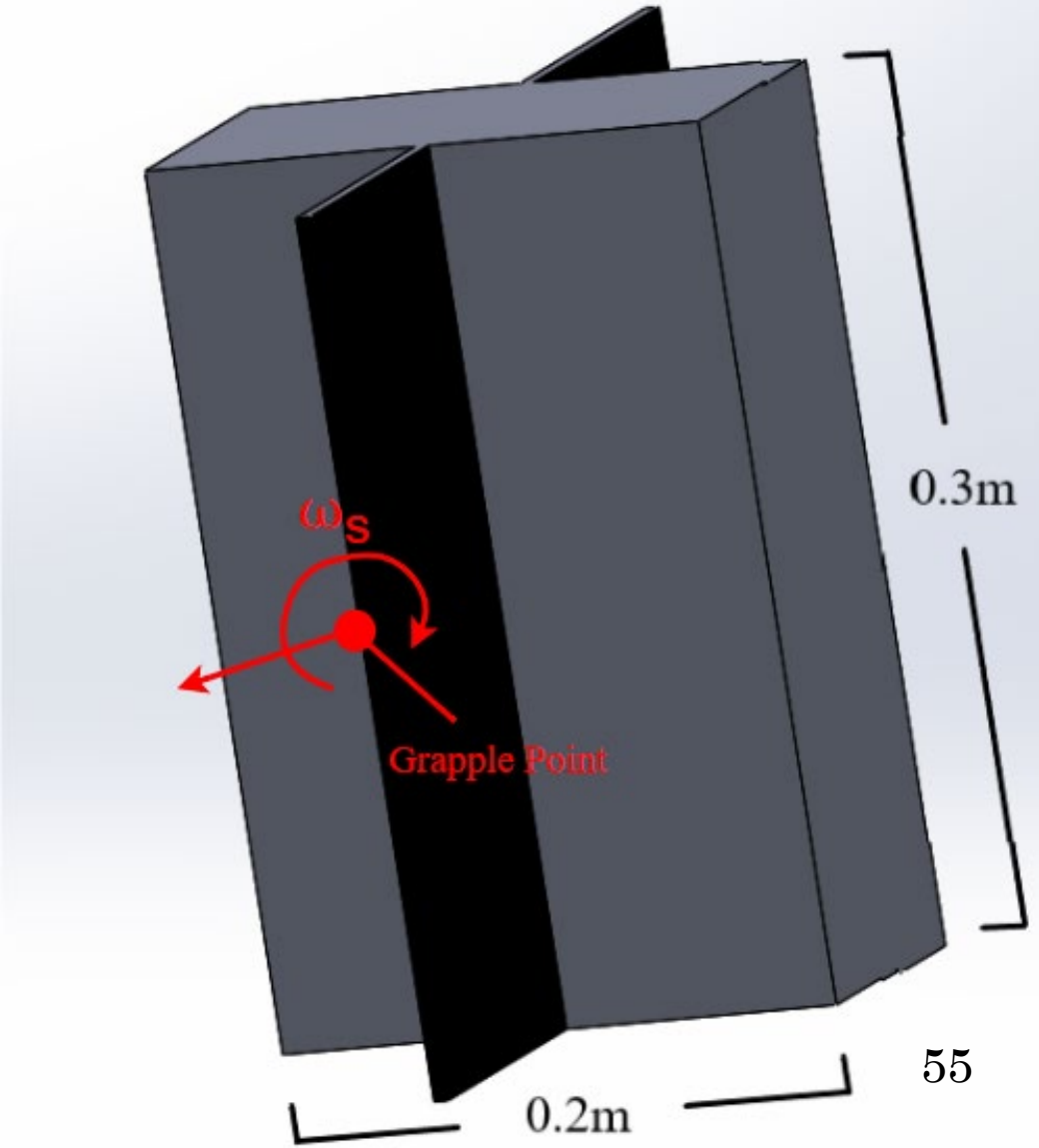
- Host spacecraft has no initial angular momentum
- Reaction wheel begins at rest, in line with GP's axis of rotation
- Reaction wheel model used is Blue Canyon RWP050 ($(I\omega)_{wheel,max} = 0.05Nms$)
- Target object can be modeled as a 6U CubeSat with constant density
- GP can be modeled as a flat plate (simulating solar panel)

## Solution:

$$I_s\omega_s \leq (I\omega)_{wheel,max} \therefore \omega_s = \frac{0.05Nms}{I_s}$$

$$where\ I_s = \frac{m(L^2+W^2)}{12}$$

As the target object is a 6U CubeSat, dimensions used are $m = 8kg$, $L = 0.3m$, $W = 0.2m$, yielding $I_s = 0.8645kg-m^2$. Thus, angular velocity $\boxed{\omega_s \approx 0.5784\frac{rad}{s}}$. The maximum angular distance between the (optimal and worst feasible) end effector and grapple point orientations is $\Delta\theta = 90° = \frac{\pi}{2}rad$, and the time between said distances is $\boxed{\Delta t = \frac{\Delta\theta}{\omega_s} \approx 2.72s.}$



# Time to Despin Target Derivation

## Assumptions:

- Despin time should match control loop refresh rate computation
- Reaction wheel angular acceleration is constant ($\frac{d\alpha_s}{dt} = 0$).

## Solution

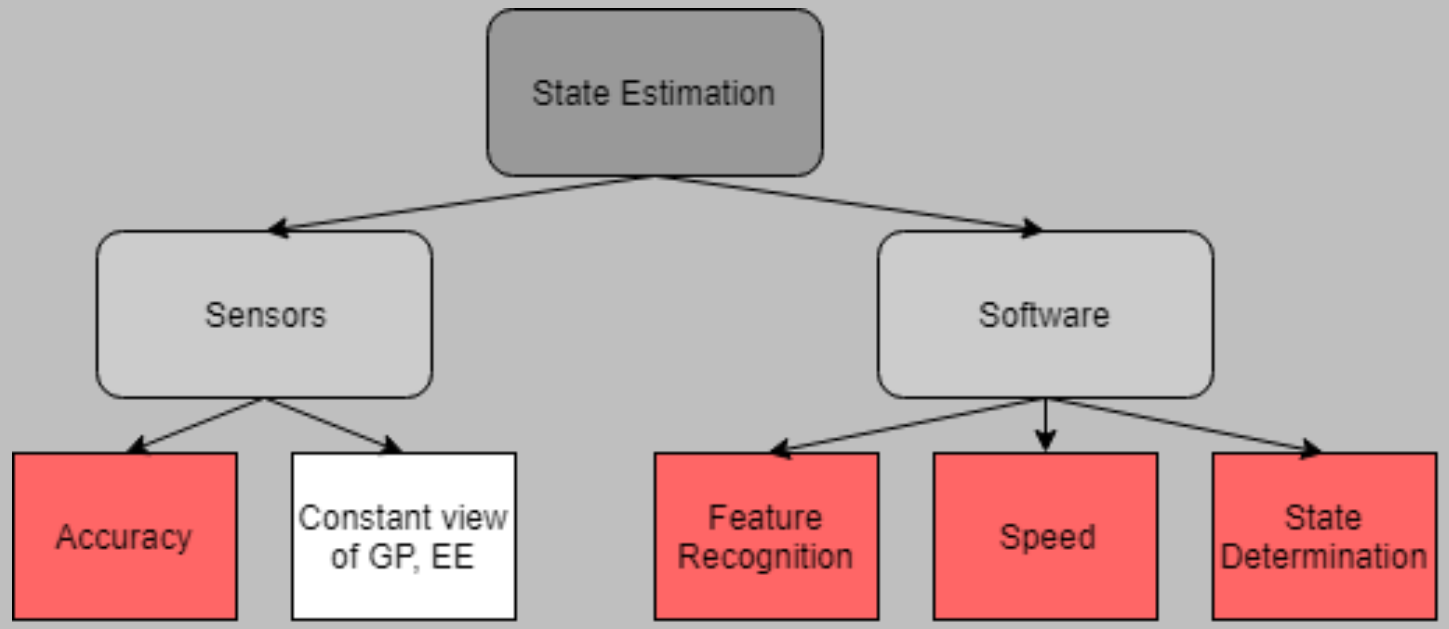
Note, the maximum torque of the reaction wheel is:

$$M_{RW} = 0.007Nm = I_s\alpha_s \therefore I_s \approx 0.08645kg - m^2$$

Rearranging, $\alpha_s \approx 0.08097\frac{rad}{s^2}$. Since angular acceleration is assumed constant, $\Delta\omega_s = \alpha_s\Delta t$, so $\Delta t_{max} = \frac{\omega_{s,max}}{\alpha} \therefore \omega_{s,max} \approx 0.578\frac{rad}{s}$. As such, $\boxed{\Delta t_{max} \approx 7.14s}$

SIERRA NEVADA CORPORATION

# CPE 1: State Estimation




Use MATLAB to estimate the states of the end effector and target

FR2: The end effector's final state shall be equal to that of the grapple point

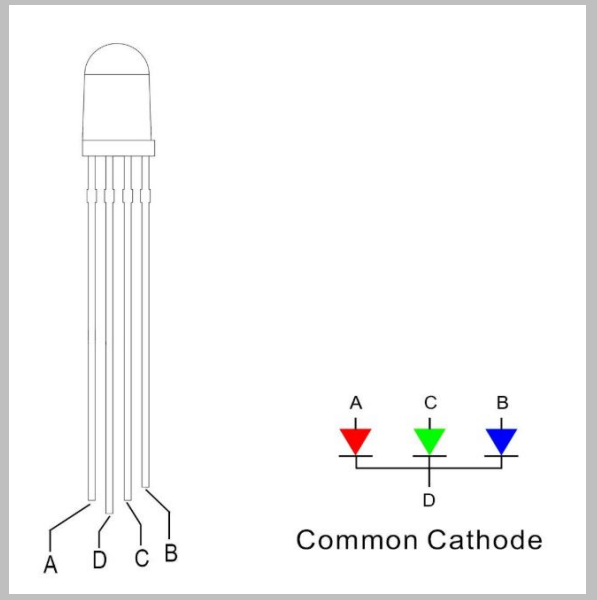
# DR 2.4.2/3: Position of grapple point and end effector shall be determined within 4.95cm

- Force sensor is going to determine the final grip state.
- Minimum force determined by manufacturer is 0.196 N (0.04406 lbf)
- Force will be used to induce a voltage to power an EDGELEC RGB tri-color LED connected to end effector. (DC 6-13V)

$$F_{lbf} = \frac{F_{voltage} - 12.20}{-0.0065}$$

$$F_{voltage} = -0.0065 * F_{lbf} + 12.20$$

**Voltage = 12.1997 V** ✓



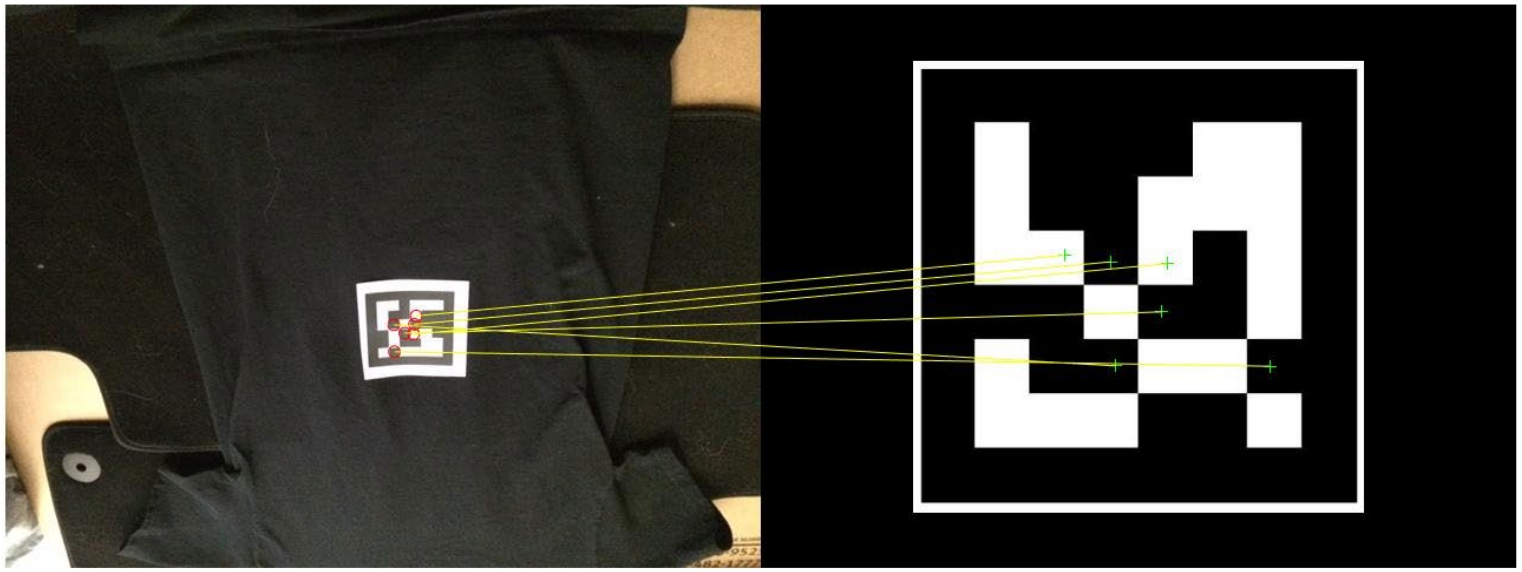
# Sensors

- The system will use two sensors in order to increase the FOV and accuracy
  - The primary sensor will be located offset from the base of the arm at a distance comparable to that of the length of a 6U CubeSat
    - The primary sensor is an *Intel RealSense Depth D435* camera
      - This uses an RGB visual sensor and an IR sensor
  - The secondary sensor will be located on the robotic arm, just behind the rotating grappling end-effector
    - The secondary sensor  is an *ArduCAM Mini 2MP OV2640*
      - This relies solely on an RGB visual sensor

# State Estimation

- An even lower resolution photo can identify the fiducial marker. The fiducial marker for this experiment is larger than the one used for the 1225x918 resolution experiment.
- The simulated image has a resolution of 640x480. This is much lower than the RGB camera on the Intel D435 (1920x1080).

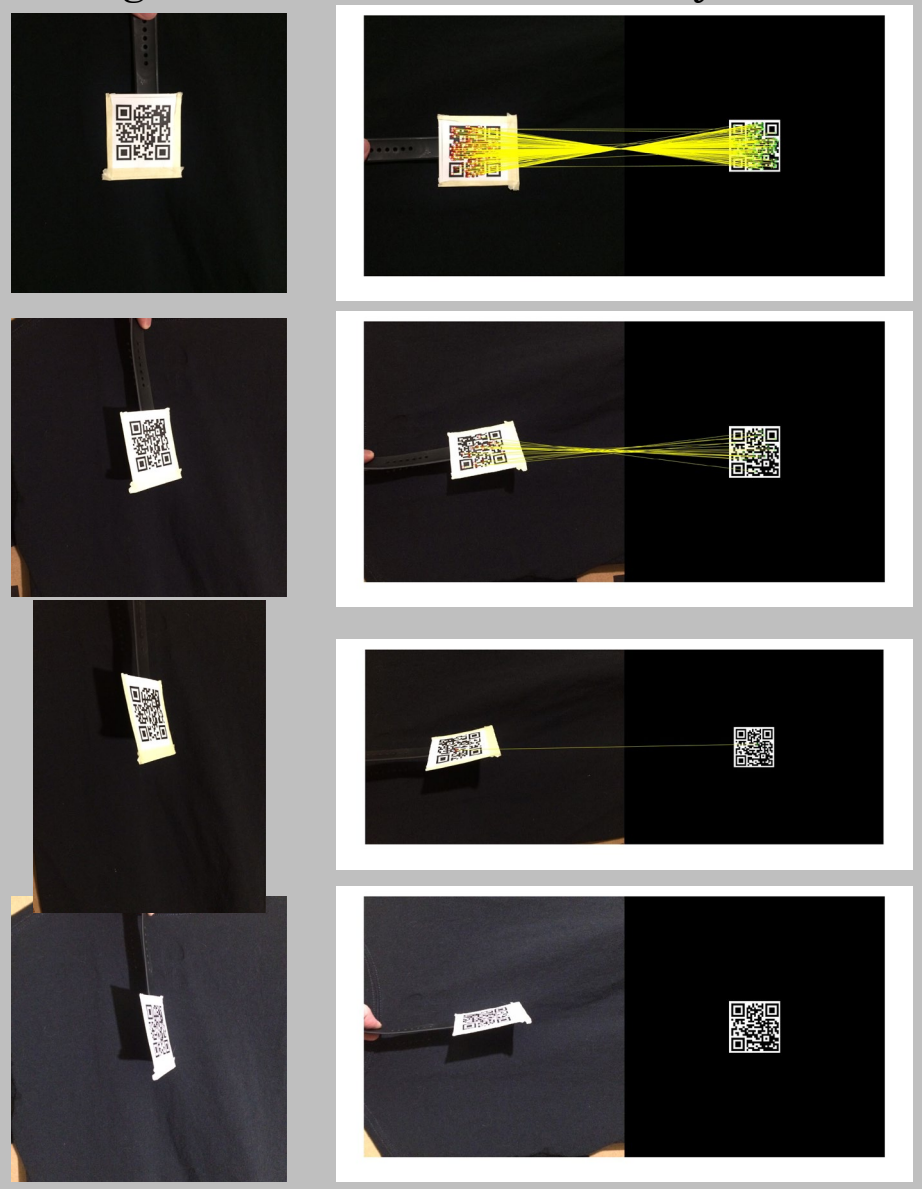
Simulated view



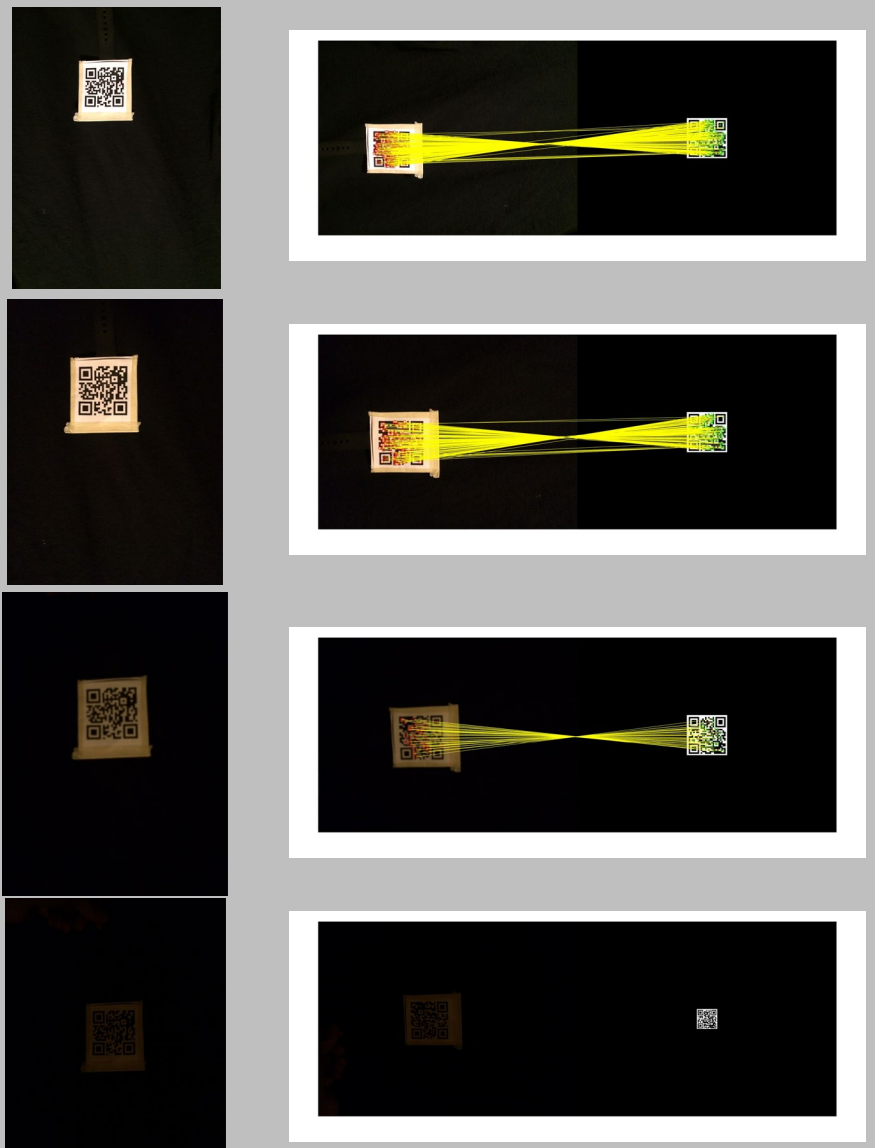
Features found in scene image confirms the object is identified. Therefore, feature recognition is feasible

# State Estimation

- Angle of fiducial maker study



- Lighting effcets on fatucial markers



# Dimensional Uncertainty in RealSense D435 Readings

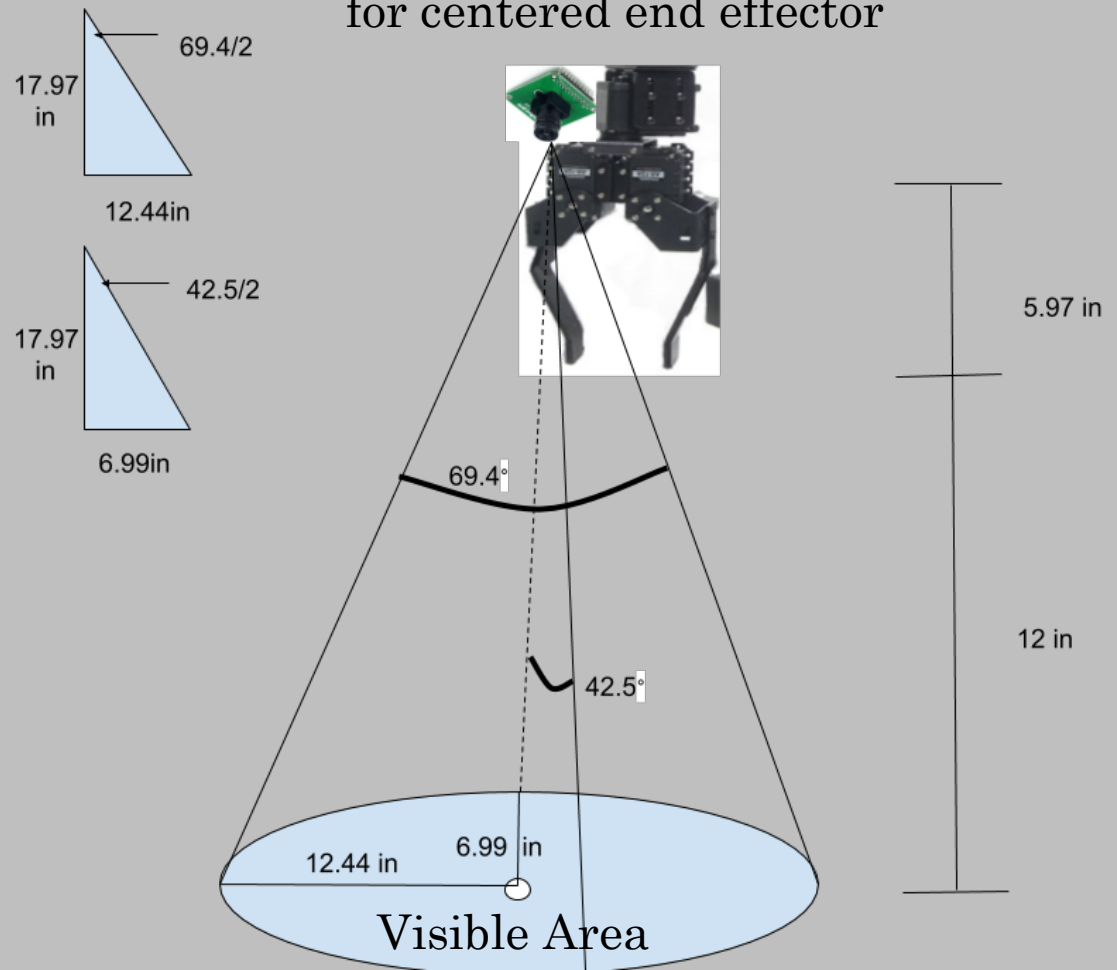

- Maximum RGB resolution: 1920x1080; take $W$ as the width of the image.
  - RGB field-of-view: 91.2°
- Maximum depth sensor resolution: 1280x720, uncertainty in measurements <2%
- Assumption: distance from camera to target: 0.5 m (arm length is 0.551 m)
- Width of a pixel: $\Delta\theta = 91.2\pi/(180 * W) \quad [rad]$
- Assume small angles to find lateral/vertical uncertainty: $\delta_{lat} = 0.5 * 91.2/W \quad [m]$
- What is the uncertainty in depth? $\delta_{depth} = 0.5 * 0.02 = 0.01 \quad [m]$
- What is the total magnitude of this uncertainty?

$$Uncertainty = \sqrt{0.01^2 + 2\left(\frac{45.6\pi}{180W}\right)^2} \quad [m]$$

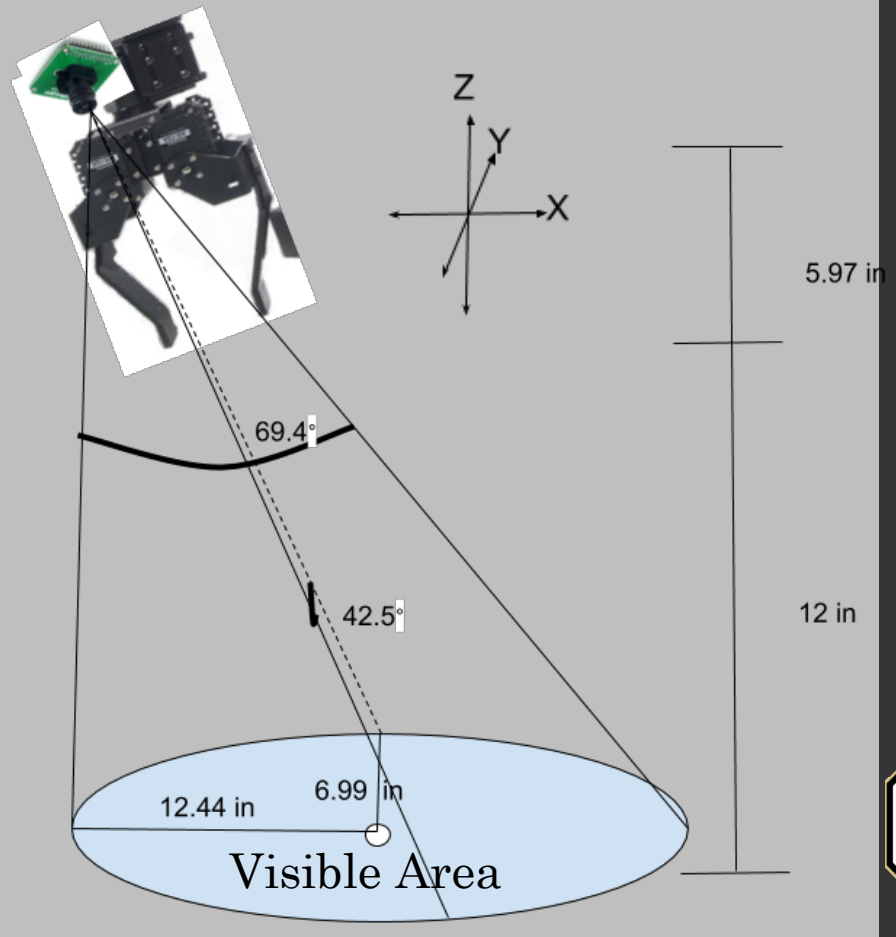
# SENSOR: Constant Vision Feasibility

- DR 2.1.1, 2.2.1, 2.3.1 For every measurement, there must be an unobstructed view of the grapple point

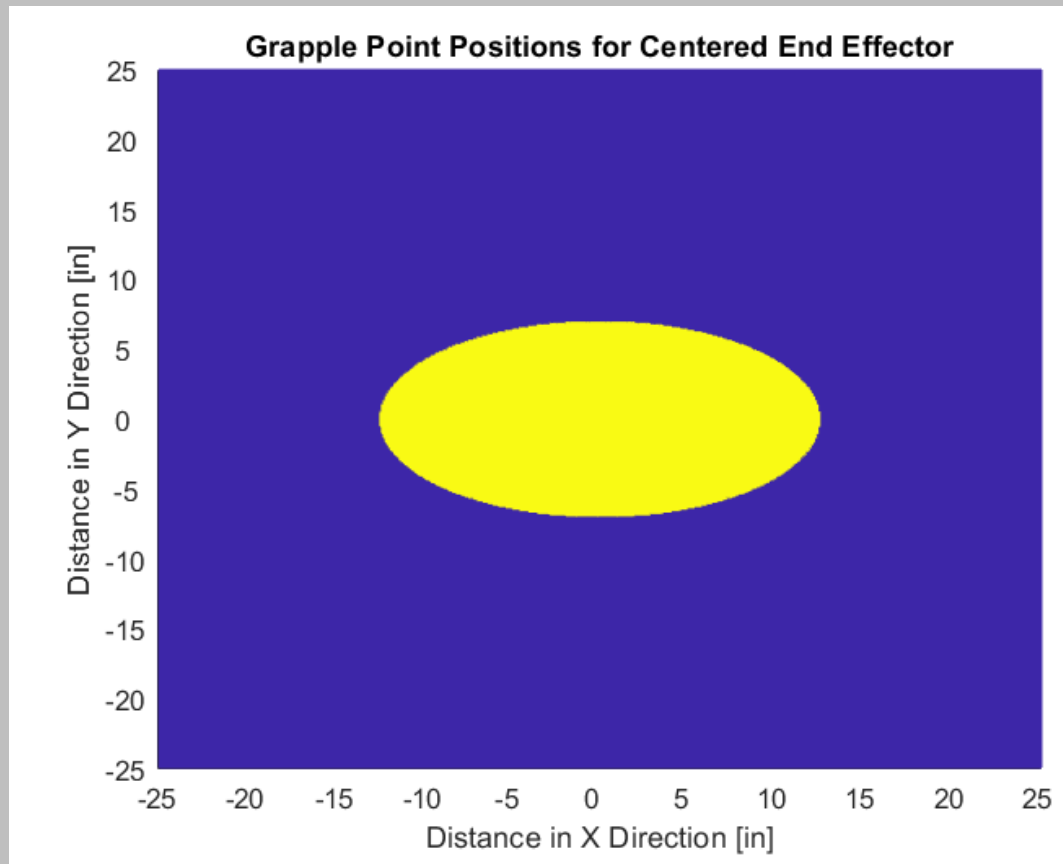
Grapple point positions for centered end effector

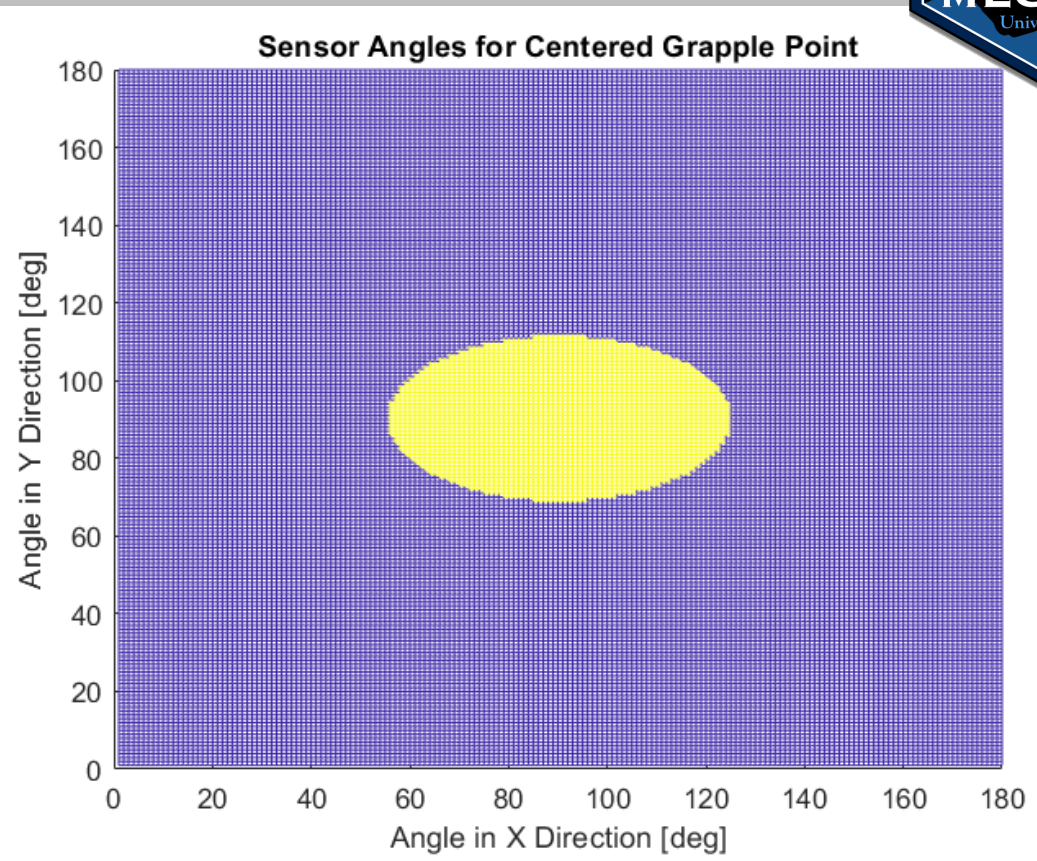



Sensor angle for centered grapple point



# SENSOR: Constant Vision Feasibility

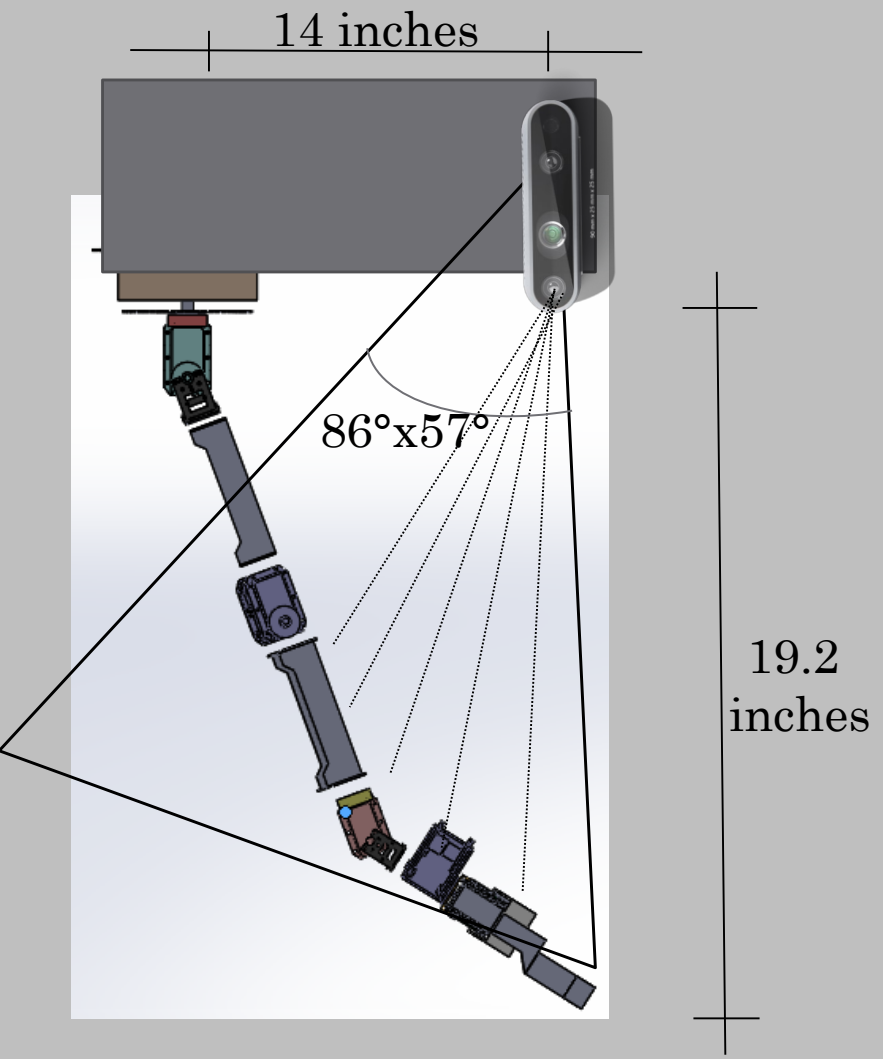


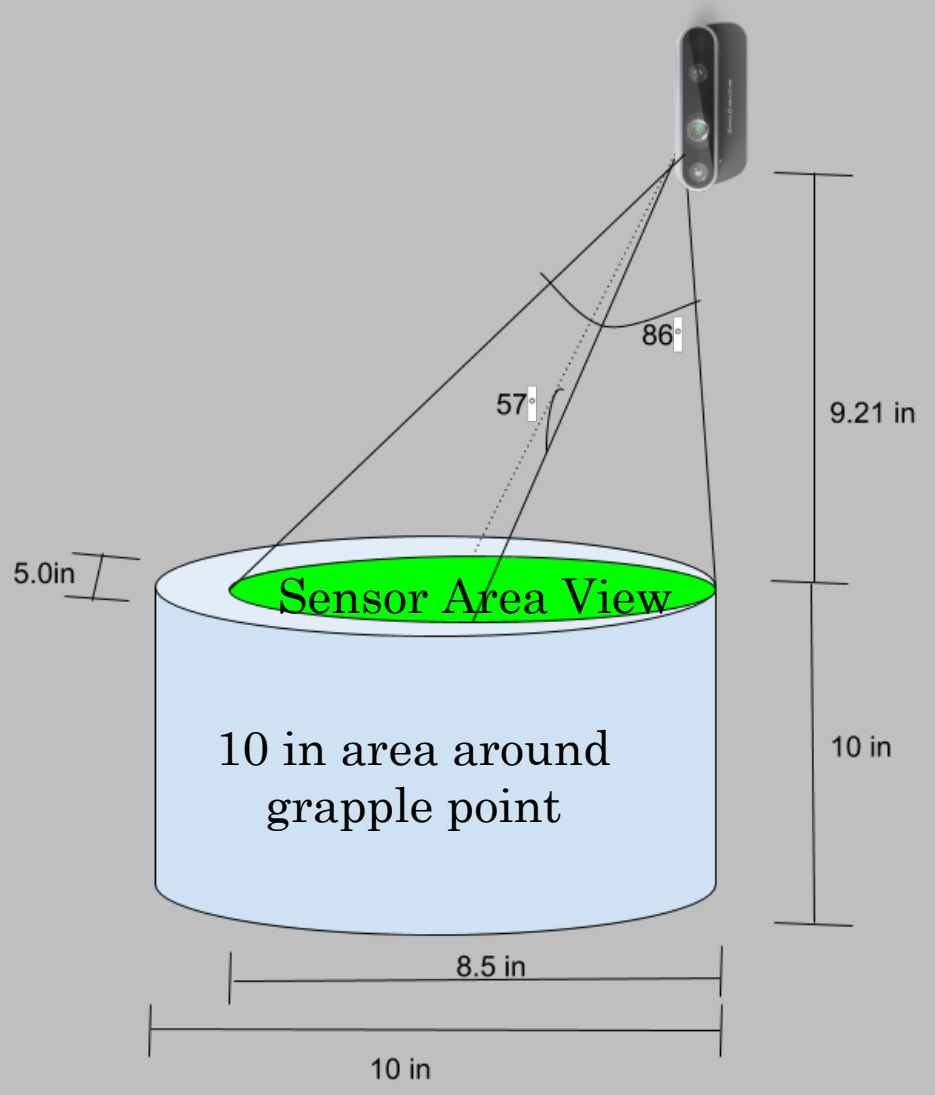




Grapple point positions for centered end effector

# SENSOR: Constant Vision Feasibility

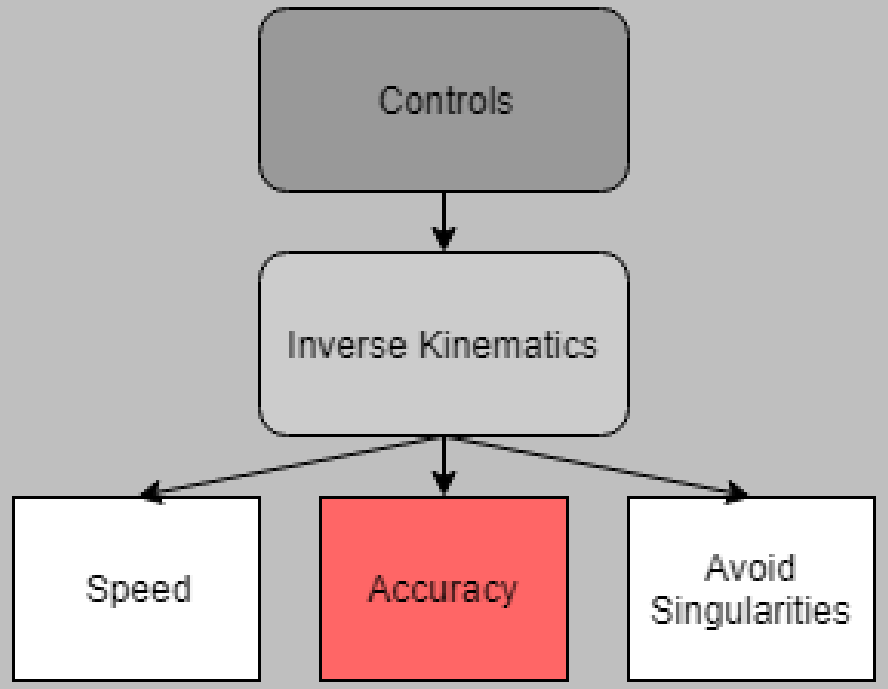
- DR 2.#For every measurement there must be an unobstructed view of the end effector




The sensor covers 85% of the 10 in grapple point "area"

# CPE 2: Controls Software

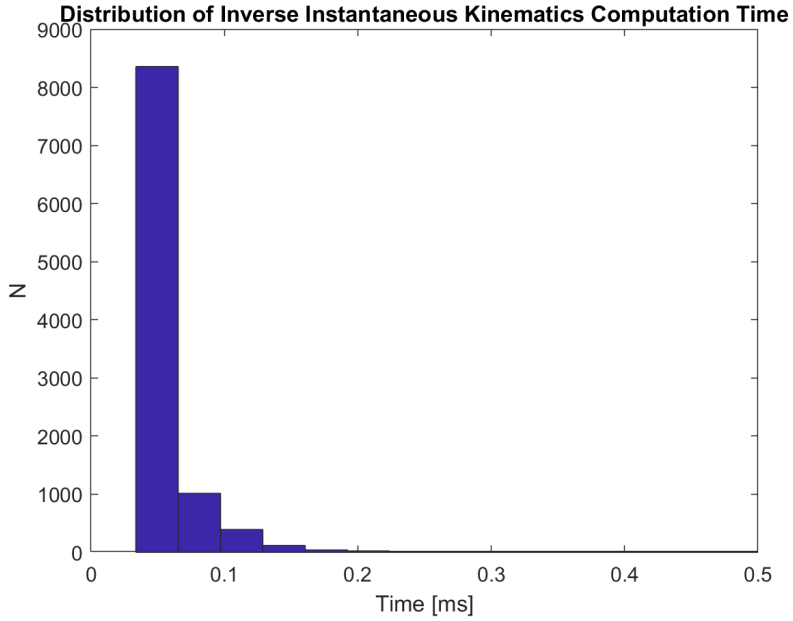


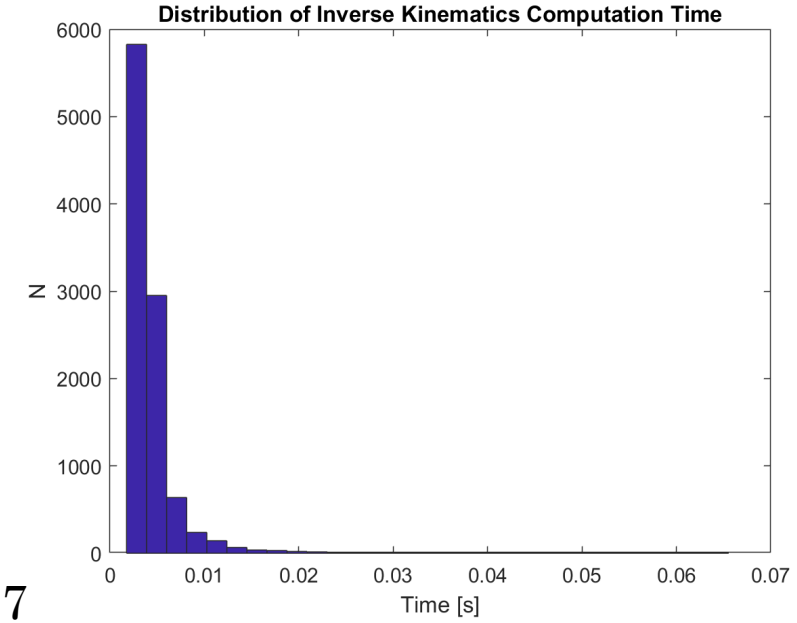

Solve inverse arm kinematics along the path for desired actuator angles and angular velocities

# SOFTWARE: Inverse Kinematics Timing Feasibility

- **DR 1.1.2 Inverse kinematics shall be solved in under 2.72 seconds**

| Kinematics | Mean [s] | Max [s] |
|---|---|---|
| Position | 0.0044 | 0.0655 |
| Velocity | 0.0512E-3 | 0.0032 |
| **Total** | **0.0045** | **0.0687** |

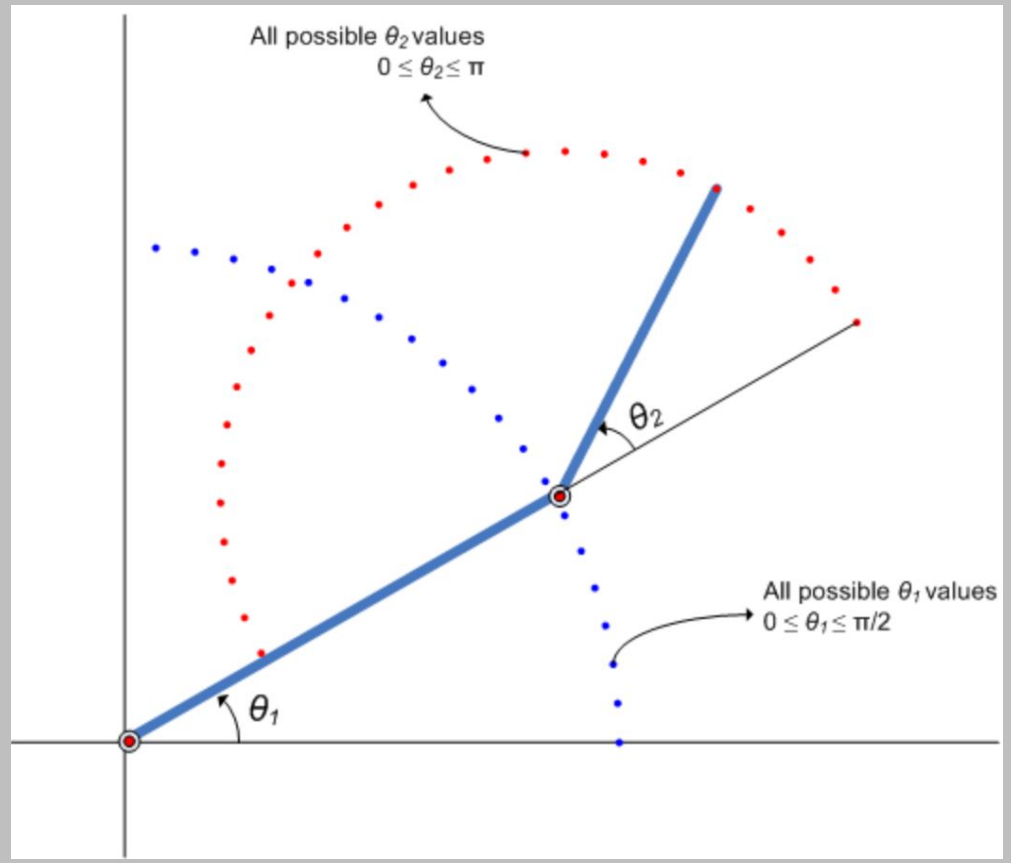


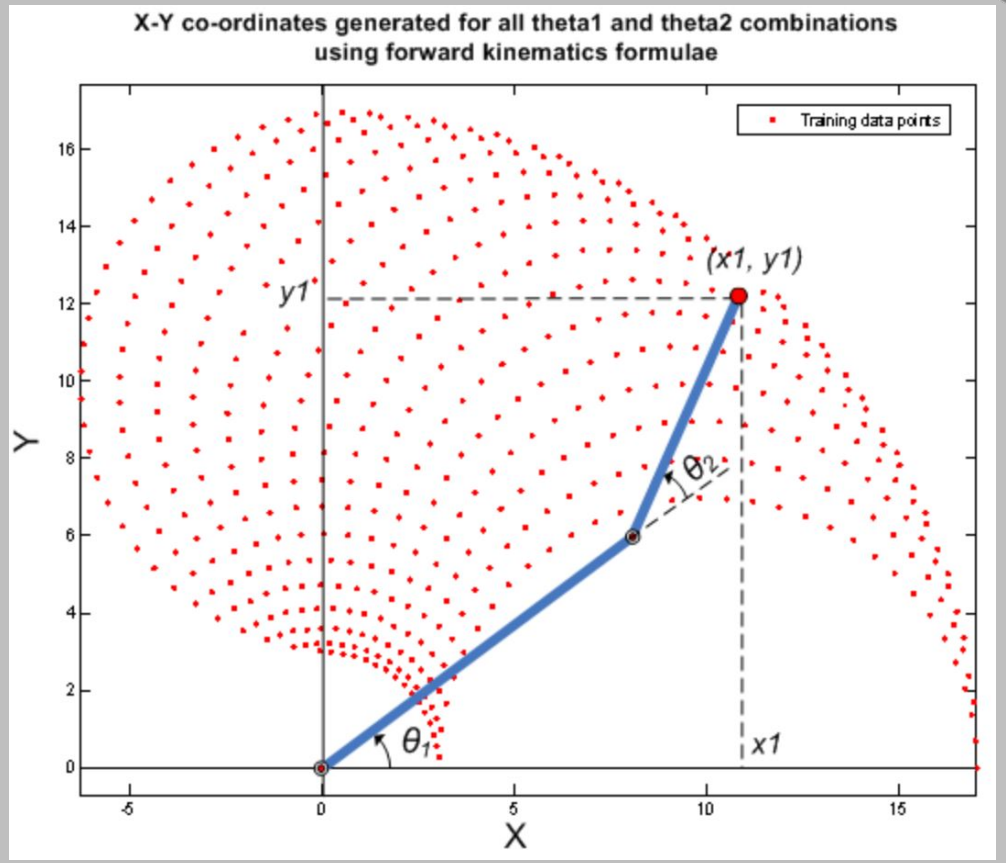



# SOFTWARE: Inverse Kinematics Singularity Avoidance


- FR1: The system shall operate with a closed loop response, based on data from decoupled sensors
- **DR1.2: Inverse kinematics solution shall avoid singularities**
- Singularities can and will be avoided by using quaternions to specify joint states

# S: Inverse Kinematics Feasibility



# Servo Command Protocol (TTL)


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

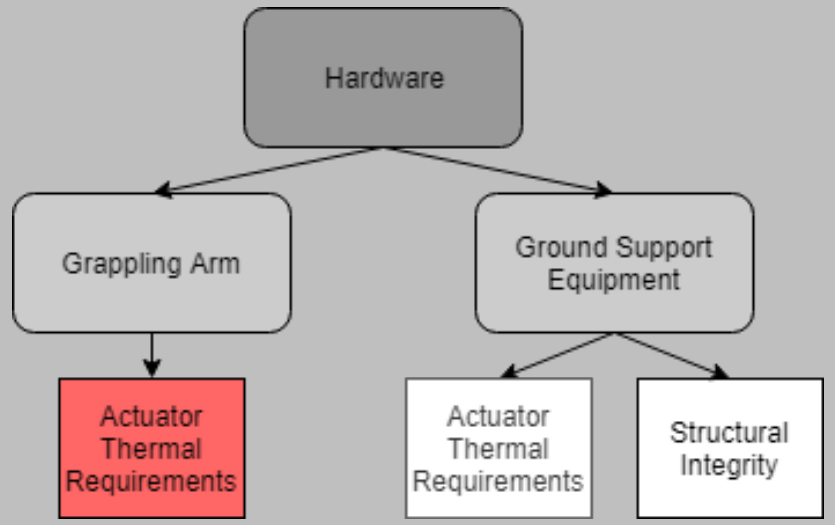
START: (Hex 0XFF 0XFF) The double FF initializes communication between the COM and the Dynamixels.
ALL: Broadcast ID (Hex-0XFE
) to all Dynamixels, disables return of status packets
T.LEN: Total Length, Uses the formula (NxL)+N+4, where N is number of Dynamixels, L is the length of the Control Table 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)

**Changing the SYNC and INSTRUCTION bits will result in different commands**
The dynamixel_motor ROS package will be used to write these commands



SIERRA NEVADA CORPORATION

# CPE 3: Hardware




Actuation of servos on arm within thermal requirements

Project Overview | Component Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Summary & Strategy

FR3: The system shall be operable in an Earth-based controlled environment which simulates LEO

**DR 3.1.1: Actuators shall not operate for no longer than 255 seconds.**

$$0.368Hz = 2.71739sec$$

$$4.25min = 255sec$$

$$\#_{Commands} = \frac{255sec}{2.71739sec/comm} \cong 94$$

$$\frac{degree}{command} = \frac{180^o}{94} = 1.91489^o/command$$

$$\frac{degree}{sec} = \frac{1.91489^o/comm}{2.71739sec/comm} = \boxed{0.704681^o/sec}$$

$$0.704681^o/sec < 327.273^o/sec$$ ✓

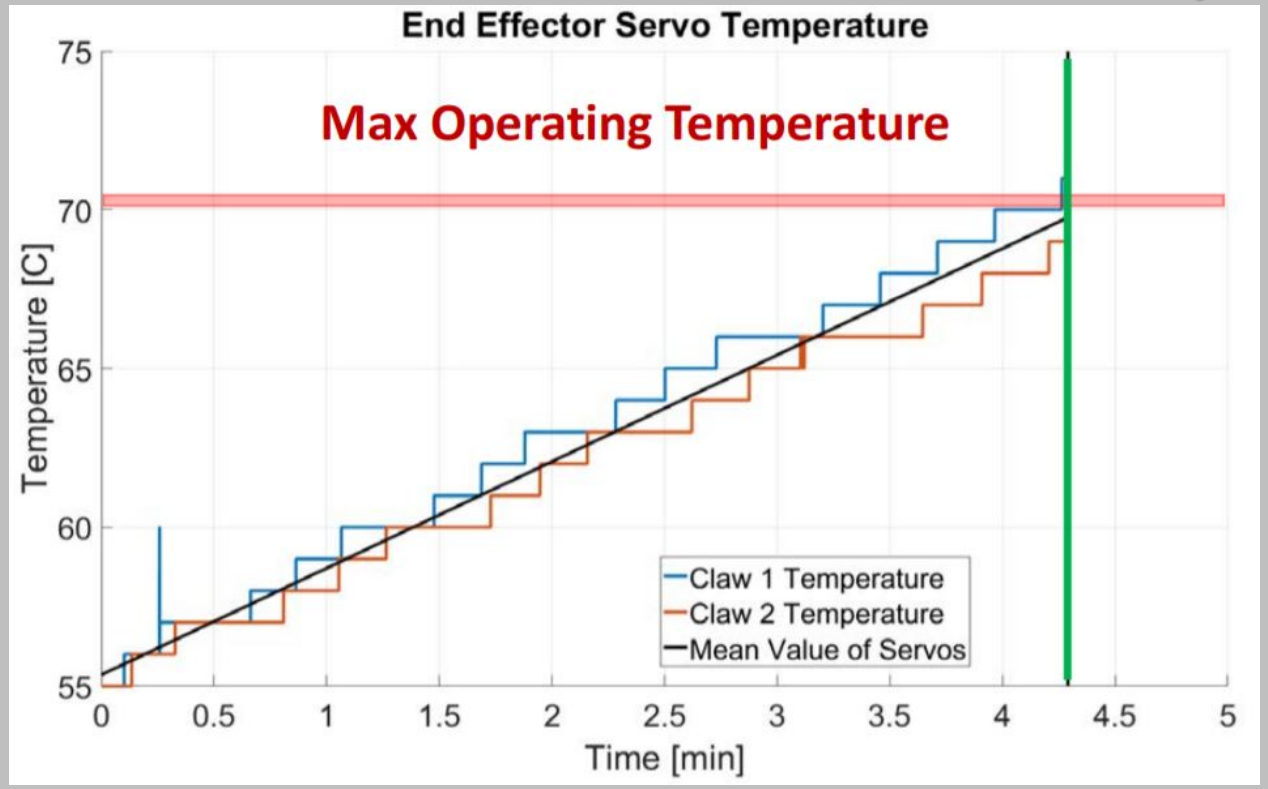
# Thermal Analysis of Target Actuation

FR3: The system shall be operable in an Earth-based controlled environment which simulates LEO

**DR 3.1.4:** Target (flat plate) shall rotate at a speed of Wgp = 0.578 rad/s

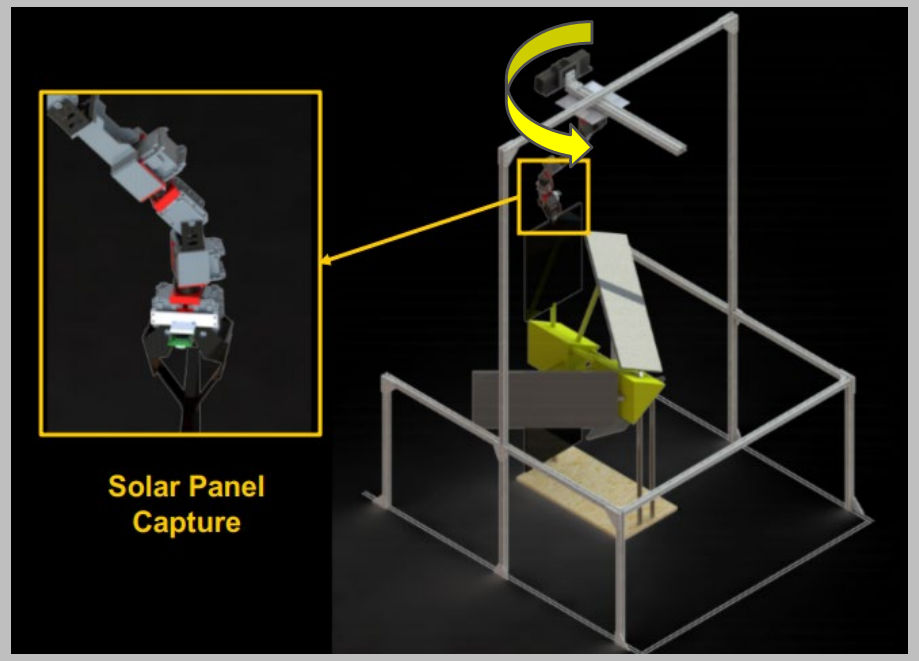
**Note: Other options for target motion actuation are being considered (open to suggestion)**

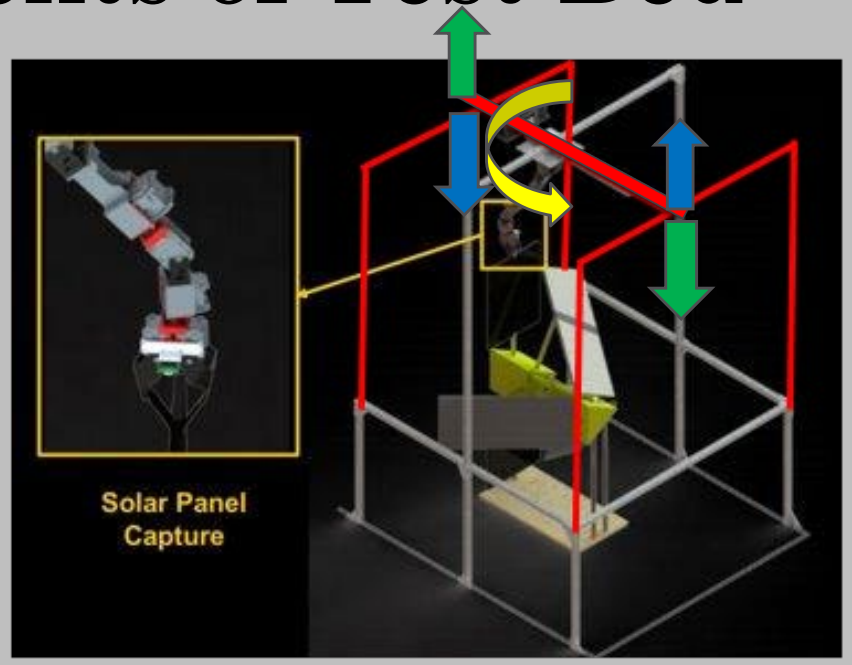
- Target motion will be actuated using an MX-106T dynamixel servo (no load speed = 4.712 rad/s)
- Max operating temperature of 70 °C with factor of safety
  - Servo can operate for 4.25 minutes before reaching this threshold.
- Note: CASCADE experiment was performed with horizontally oriented arm
- Orienting the MEGACLAW arm vertically will increase time to reach max operating temperature.




**End effector servo temperature study conducted by CASCADE.**

# Structural Enhancements of Test Bed



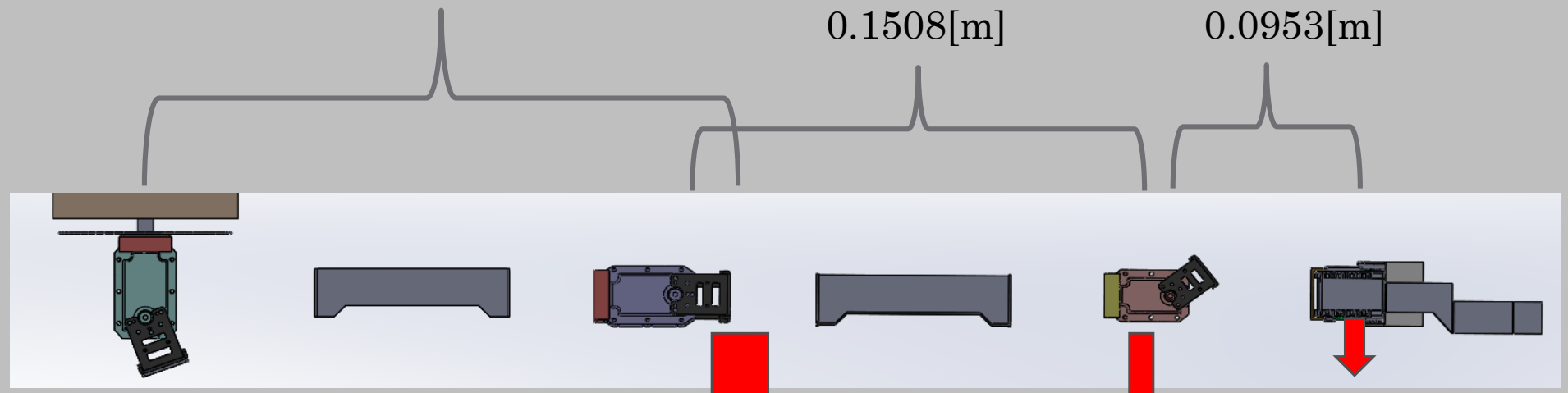

Torque From Arm

Reaction Force from Arm Torque

Reaction Force from Supports

Structural supports counter reaction forces from the torque created by the arm displacement

# DR 3.1.2: All arm servos shall be able to withstand maximum torque at 90°




$$T_{MX-28T} = 0.259[kg]*g[\frac{m}{s^2}]*0.0935[m]$$

$$T_{MX-64T} = 0.353[kg]*g[\frac{m}{s^2}]*0.1508[m]$$

$$T_{MX-64T} = 0.516[kg]*g[\frac{m}{s^2}]*0.169[m]$$