Mechanically Engineered Grappling Arm to Capture Litter and Atmospheric Waste

Critical Design Review


Customer: TJ Sayer - Sierra Nevada Corporation (SNC)

Advisor: Francisco Lopez Jimenez
Project Purpose & Objectives
MEGACLAW is a proof of concept for mitigating debris by capturing then deorbiting debris or performing maintenance on broken spacecraft.

Heritage: CASCADE, KESSLER

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

Primary Camera: Intel D435 FOV from mount offset from arm's base

CPU: processes visual data and control algorithm computations

Testbed Grapple Feature: Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU

Primary Camera: Intel D435 FOV from mount offset from arm's base

CPU: processes visual data and control algorithm computations

AR tag (fiducial marker)

Testbed Grapple Feature: Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel

Primary Camera: Intel D435 FOV from mount offset from arm’s base

CPU: processes visual data and control algorithm computations

Testbed Grapple Feature: Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
Updated CONOPS

Robotic Arm: 4 active DOF
Crustcrawler
Robotic arm

Primary Camera: Intel D435 FOV from mount offset from arm's base

CPU: processes visual data and control algorithm computations

Testbed Grapple Feature:
Flat panel on rotating turntable

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU

Robotic Arm: 4 active DOF
Crustcrawler Robotic arm

Primary Camera: Intel D435 FOV from mount offset from arm's base

CPU: processes visual data and control algorithm computations

Testbed Grapple Feature: Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Intel D435 measures end effector and solar panel spin axis state. Move on if converged

Robotic Arm: 4 active DOF
Crustcrawler
Robotic arm

Primary Camera: Intel D435 FOV from mount offset from arm’s base

CPU: processes visual data and control algorithm computations

Testbed Grapple Feature: Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Intel D435 measures end effector and solar panel spin axis state. Move on if converged
7. Pixy2 finds angle of solar panel spin axis and communicates with CPU through Arduino Uno MEGA

Robotic Arm: 4 active DOF
Crustcrawler Robotic arm
Primary Camera: Intel D435 FOV from mount offset from arm’s base
CPU: processes visual data and control algorithm computations
Secondary Camera: Pixy2 FOV from mount on arm
End Effector
Arduino Uno MEGA: Microcontroller for Pixy2
Testbed Grapple Feature: Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Intel D435 measures end effector and solar panel spin axis state. Move on if converged
7. Pixy2 finds angle of solar panel spin axis and communicates with CPU through Arduino Uno MEGA
8. CPU computes spin rate and commands rotation of end effector to match rotation of solar panel
Updated CONOPS

Primary Camera: Intel D435 FOV from mount offset from arm's base

CPU: processes visual data and control algorithm computations

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Intel D435 measures end effector and solar panel spin axis state. Move on if converged
7. Pixy2 finds angle of solar panel spin axis and communicates with CPU through Arduino Uno MEGA
8. CPU computes spin rate and commands rotation of end effector to match rotation of solar panel
9. Intel D435 verifies rotation alignment. Move on if correctly aligned

Robotic Arm:
4 active DOF
Crustcrawler
Robotic arm

Secondary Camera:
Pixy2 FOV from mount on arm

Arduino Uno MEGA:
Microcontroller for Pixy2

Testbed Grapple Feature:
Flat panel on rotating turntable
Updated CONOPS

1. Intel D435 searches for AR tag on solar panel and sends data to CPU
2. Algorithm identifies state of end effector and solar panel
3. Algorithm solves inverse kinematics for joint velocities
4. CPU transmits commands to arm actuators
5. Robotic arm actuates as commanded by CPU
6. Intel D435 measures end effector and solar panel spin axis state. Move on if converged
7. Pixy2 finds angle of solar panel spin axis and communicates with CPU through Arduino Uno MEGA
8. CPU computes spin rate and commands rotation of end effector to match rotation of solar panel
9. Intel D435 verifies rotation alignment. Move on if correctly aligned
10. Actuate arm forward and close end effector. Verify grapple through binary measurement from force sensors

Primary Camera: Intel D435 FOV from mount offset from arm's base
CPU: processes visual data and control algorithm computations

Robotic Arm:
- 4 active DOF
- Crustcrawler
- Robotic arm

Secondary Camera:
- Pixy2 FOV from mount on arm

Arduino Uno MEGA:
- Microcontroller for Pixy2

Testbed Grapple Feature:
- Flat panel on rotating turntable

Force Sensors:
- Measures force upon grapple and communicates with Arduino Uno MEGA
Proposed Design
Functional Block Diagram
Functional Block Diagram
Functional Block Diagram
Control Block Diagrams
## Hardware Description – Arm and Actuators

### Actuator Specifications

<table>
<thead>
<tr>
<th>Mass [kg]</th>
<th>106T Dual (Vertical) Actuator</th>
<th>12.7cm Girder</th>
<th>MX-64T Actuator</th>
<th>6.35cm Girder</th>
<th>MX-28T (Vertical) Actuator</th>
<th>MX-28T (Horizontal) Actuator</th>
<th>AX-12A Dual Gripper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length [m]</td>
<td>0.93</td>
<td>0.037</td>
<td>0.126</td>
<td>0.022</td>
<td>0.072</td>
<td>0.072</td>
<td>0.187</td>
</tr>
<tr>
<td>Stall Torque [Nm]</td>
<td>16</td>
<td>6</td>
<td>2.5</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factor of Safety</td>
<td>~19</td>
<td>~11</td>
<td>~10</td>
<td>~18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hardware Description – End Effector

- **Ball Joint Grippers**
  - Allows 10° error with gripping position

- **AC4598 Slip Ring**
  - End Effector needs to be able to rotate indefinitely

- **Pololu 0.6" Force Sensitive Resistor**
  - Confirms a successful grapple
Hardware Description - Test Bed

- Improving KESSLER test bed
- Added Crossbars to counteract torque from arm
Hardware Description: Solar Panel

- 11.8in x 7.9in x 0.5in
- 2 fiducial markers on edges for position determination
- 3 color center for rotation
Primary Vision System Block Diagram

- Mounting Surface
- Cube Satellite
  - Depth Data Output from IR: (x,y,z)
- Intel D435
  - Color Data Output from visual: (R,G,B)
  - Identify specific AR tag
- Determine AR tag position
  - Determine Orientation and Position of AR tag
    - Determined

Legend:
- Hardware Connection
- IR Feedback
- IR Data
- Visual Feedback
- Visual Data
- Data Flow
- AR tag
Vision System: Intel RealSense D435 & Pixy2

**Intel RealSense D435**

Provides: Color video
- (1920x1080) at 30fps
- 69.4° x 42.5°

3D metric position
- <1% error
- 85.2° x 58°

**Pixy2**

Provides: Pixel position and angle of solar panel
- 60° x 40°
- 10 grams
Primary Vision System: AR Tags

Fiducial markers proposed at PDR: Colored Spheres

**Drawbacks:**
- Prohibitively slow (~0.5 Hz)
- Sensitive to variable lighting
- Number of distinct colors limits available number of markers (~6)

New design solution: AR tags

**Advantages:**
- Fast (~20 Hz)
- Reliable in variable lighting and orientation
- Up to 250 unique IDs
- Four corners give orientation of tag
Hardware Description – End Effector Tags
Secondary Vision System: Colored Markers

- **Pixy2** uses color signatures to track objects
- **Pixy2** has a FOV of $40^\circ \times 60^\circ$, and can track up to 7 color signatures
- **Solar Panel** is blue, with 3 colored markers placed in the center
- Two or more color signatures side by side are used to determine the angle of the signatures

![Top View of Grapple Point](image)

*not to scale*
Project Elements

Controls
- Inverse Kinematics
  - Accuracy
  - Speed
  - Avoid Singularities

Hardware
- Ground Support Equipment
  - Actuator Thermal Requirements
  - Structural Integrity
- Grappling Arm
  - Actuator Thermal Requirements

State Estimation
- Software
  - Feature Recognition
  - Speed
  - State Determination
- Sensors
  - Accuracy
  - Constant view at GP & EE

Critical Elements
- Non-Critical Elements
## Critical Project Elements

<table>
<thead>
<tr>
<th>CPE</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sensors/Controls - Accuracies</strong></td>
<td>Error tolerance is very small, multiple components contribute</td>
</tr>
<tr>
<td><strong>Software - Speed</strong></td>
<td>Image processing and 3D modeling are time-intensive processes; minimum GP angular velocity sampling rate</td>
</tr>
<tr>
<td><strong>Actuation of Robotic Arm and Ground Support Equipment</strong></td>
<td>Any damage to components in the project can cause week-long delays</td>
</tr>
</tbody>
</table>
Design Requirements
Control Process

Stage 1:
Coarse Alignment

Stage 2:
Fine Adjustment

Stage 3:
Rotation Matching

Stage 4:
Grapple
## Design Requirements: Stage 1

### Stage 1: Coarse Alignment

![Side View Diagram](image)

<table>
<thead>
<tr>
<th>Stage</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Theta</th>
<th>Derived From</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Coarse Alignment</td>
<td>+18.2 [cm]</td>
<td>+18.2 [cm]</td>
<td>+5.1 [cm]</td>
<td>N/A</td>
<td>Pixy2 FOV {60°,40°}</td>
</tr>
<tr>
<td>2: Fine Adjustment</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+5.1 [cm]</td>
<td>N/A</td>
<td>CASCADE Heritage</td>
</tr>
<tr>
<td>3: Rotation Matching</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+5.1 [cm]</td>
<td>± 10°</td>
<td>Ball Joint Limitations</td>
</tr>
<tr>
<td>4: Grapple</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+2.5 [cm]</td>
<td>± 10°</td>
<td>Grip Depth</td>
</tr>
</tbody>
</table>
Design Requirements: Stage 2

Stage 2: Fine Alignment

<table>
<thead>
<tr>
<th>Stage</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Theta</th>
<th>Derived From</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Coarse Alignment</td>
<td>+18.2 [cm]</td>
<td>+18.2 [cm]</td>
<td>+5.1 [cm]</td>
<td>N/A</td>
<td>Pixy2 FOV {60°, 40°}</td>
</tr>
<tr>
<td>2: Fine Adjustment</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+5.1 [cm]</td>
<td>N/A</td>
<td>CASCADE Heritage</td>
</tr>
<tr>
<td>3: Rotation Matching</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+5.1 [cm]</td>
<td>± 10°</td>
<td>Ball Joint Limitations</td>
</tr>
<tr>
<td>4: Grapple</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+2.5 [cm]</td>
<td>± 10°</td>
<td>Grip Depth</td>
</tr>
</tbody>
</table>
## Design Requirements: Stage 3

### Stage 3: Rotation Matching

![Diagram showing rotation matching](image)

<table>
<thead>
<tr>
<th>Stage</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Theta</th>
<th>Derived From</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Coarse Alignment</td>
<td>+18.2</td>
<td>+18.2</td>
<td>+5.1</td>
<td>N/A</td>
<td>Pixy2 FOV {60°, 40°}</td>
</tr>
<tr>
<td>2: Fine Adjustment</td>
<td>+1.5</td>
<td>+1.5</td>
<td>+5.1</td>
<td>N/A</td>
<td>CASCADE Heritage</td>
</tr>
<tr>
<td>3: Rotation Matching</td>
<td>+1.5</td>
<td>+1.5</td>
<td>+5.1</td>
<td>± 10°</td>
<td>Ball Joint Limitations</td>
</tr>
<tr>
<td>4: Grapple</td>
<td>+1.5</td>
<td>+1.5</td>
<td>+2.5</td>
<td>± 10°</td>
<td>Grip Depth</td>
</tr>
</tbody>
</table>

\[\omega_{\text{Claw}} = 0\]
\[\omega_{\text{SolarPanel}} = 33 \text{ [deg/s]}\]

\[\omega_{\text{SolarPanel}} = \omega_{\text{Claw}}\]
Design Requirements: Stage 4

<table>
<thead>
<tr>
<th>Stage</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Theta</th>
<th>Derived From</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Coarse Alignment</td>
<td>+18.2 [cm]</td>
<td>+18.2 [cm]</td>
<td>+5.1 [cm]</td>
<td>N/A</td>
<td>Pixy2 FOV {60°,40°}</td>
</tr>
<tr>
<td>2: Fine Adjustment</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+5.1 [cm]</td>
<td>N/A</td>
<td>CASCADE Heritage</td>
</tr>
<tr>
<td>3: Rotation Matching</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+5.1 [cm]</td>
<td>± 10°</td>
<td>Ball Joint Limitations</td>
</tr>
<tr>
<td>4: Grapple</td>
<td>+1.5 [cm]</td>
<td>+1.5 [cm]</td>
<td>+2.5 [cm]</td>
<td>± 10°</td>
<td>Grip Depth</td>
</tr>
</tbody>
</table>
Satisfying Design Requirements: End Effector State Accuracy

Actual System

Simulated System
Satisfying Design Requirements – End Effector State Accuracy

DR 2.4  Control algorithm shall command EE within convergence tolerance of 1.45 cm

- Created and solved inverse kinematics (IK) using exact SolidWorks model of arm
- Simulated error from actuators between IK and forward kinematics, then error of sensor measurement
- Set convergence tolerance of arm to 1.45 [cm] in z-position (can set for any of the states)
- Satisfied tolerance, <1.45 [cm] (and therefore the design requirement), every time

**Requirement Satisfied**
Visual Processing – Accuracy of Position Determination

Why?
Must be accurate enough for end effector to grab grapple point.

Results
• Position error between known position and primary sensor $\approx 2 - 5 \text{ mm}$
  
  **Requirement Satisfied**

• Position error between two known positions $\approx 4 - 8 \text{ mm}$

| DR 2.4 | Sensors shall measure position of EE and GP within combined error of 1.45 cm |
Visual Processing – Accuracy of Orientation Determination

DR 2.5 | The spin axis offset during grapple must be no more than 10°

AR tag set as close as possible to 50° away from Intel D435
• Mean measurement: 52.19°
• $\sigma = 0.21°$
• Some error likely due to setup

Mean error: 2.19°

Requirement Satisfied
**Visual Processing – Visibility**

<table>
<thead>
<tr>
<th>DR 2.1</th>
<th>Sensors shall determine whether grapple point is within the field of view.</th>
</tr>
</thead>
<tbody>
<tr>
<td>15°</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td>45°</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>60°</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>75°</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Visibility of AR tag's at variant conditions:

- **Distances:**
  - 0.4 m, 0.5 m, 0.6 m, 0.7 m

- **Angles:**
  - 15°, 45°, 60°, 75°

66% in view at 0.6m, 60°

Requirement Satisfied
Visual Processing – Visual Demo
Secondary Vision System – Pixy2 Angle Determination

<table>
<thead>
<tr>
<th>DR 3.4</th>
<th>GP shall rotate at a rate of $\omega_{gp} = 0.578 \text{ rad/s (or } \sim 33 , ^\circ/\text{s)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR 2.5.1</td>
<td>The GP spin rate determined by the secondary sensor shall have an error $\pm 0.33 \text{ deg/s } (1% \text{ error})$</td>
</tr>
</tbody>
</table>

Why?
From the time the wrist is spun up to final grapple, the offset angle can be no more than 10°

Results:
Testing at 200 deg/s provided an error of 0.368%

Requirement Satisfied

200 deg/s rotation
Spin Axis Alignment

- Intel D435 will **scan the fiducial markers** and create point cloud.

- **Find centroid** and normal vector out of centroid.

- **Move arm** to align with estimated spin axis (normal vector).

- **Switch over** to Pixy2 for fine alignment and **spin rate determination**.

**Calculated Centroid Location**
Spin Axis Alignment

MATLAB simulation incorporating error in measurements.

Path of AR tags on solar panel

Why?
The ball joint grippers only allow for an addition 10° of movement

- Measured X,Y,Z position from AR tags and Intel
- Black circle is true path of the AR tags on the GP

DR 2.5 The spin axis offset during grapple must be no more than 10°
Spin Axis Alignment

MATLAB simulation incorporating error in measurements.

Results: Simulation provided a mean spin axis offset 0.68° from actual

Requirement Satisfied

Mean calculated from 2000 simulations each run over a 30s interval
Project Risks
**Risks – Before Mitigation**

1. Bad gains
2. Wires become tangled during actuation of arm
3. Coordinate frame transformation accuracy
4. Actuators overheating
5. Camera not identifying AR tags in view
6. Arm is commanded into a damaging configuration

<table>
<thead>
<tr>
<th></th>
<th>Negligible</th>
<th>Minor</th>
<th>Moderate</th>
<th>Significant</th>
<th>Severe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Likely</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Likely</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible</td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unlikely</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Very Unlikely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Risks – Mitigation Methods

<table>
<thead>
<tr>
<th>Risks</th>
<th>Mitigation Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Bad gains</td>
<td>Spend time tuning gains, trial and error (iteration)</td>
</tr>
<tr>
<td>2. Wires become tangled during actuation of arm</td>
<td>Slip ring implementation</td>
</tr>
<tr>
<td>3. Coordinate frame transformation accuracy</td>
<td>Utilization of ROS for transformation and VICON for measurement</td>
</tr>
<tr>
<td>4. Actuators overheating</td>
<td>255s completion time requirement, vertical arm orientation, shortened arm</td>
</tr>
<tr>
<td>5. Camera not identifying AR tags in view</td>
<td>Two camera system</td>
</tr>
<tr>
<td>6. Arm is commanded into a damaging configuration</td>
<td>Emergency power cut option and set limitations in software</td>
</tr>
</tbody>
</table>
### Risks – After Mitigation

1. **Bad gains**

2. **Wires become tangled during actuation of arm**

3. **Coordinate frame transformation accuracy**

4. **Actuators overheating**

5. **Camera not identifying AR tags in view**

6. **Arm is commanded into a damaging configuration**

<table>
<thead>
<tr>
<th></th>
<th>Negligible</th>
<th>Minor</th>
<th>Moderate</th>
<th>Significant</th>
<th>Severe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Likely</strong></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Likely</strong></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><strong>Possible</strong></td>
<td></td>
<td></td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td><strong>Unlikely</strong></td>
<td>2</td>
<td></td>
<td>4,3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td><strong>Very Unlikely</strong></td>
<td>1</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Verification & Validation – Controls

**Requirement:**
- 2.4.4 The position of EE shall satisfy a control tolerance of 1.5 cm

**Solution:**
- Use an external reference to confirm that when the control algorithm indicates convergence, it is within this requirement

**Verification & Validation Test:**
- Utilize CU RECUV's VICON camera system with 8 reflective spheres to track the end effector's position within ~ 1 [mm]
- Mimic CASCADE test setup (vertical)
Verification & Validation – Visual Accuracy

**Requirement:**
- 2.4.2 The position of EE shall be determined within 1.45 cm
- 2.4.3 The position of GP shall be determined within 1.45 cm

**Solution:**
- 2 camera system
- Calibration of D435 and Pixy2 cameras for accuracy

**Verification & Validation Test:**
- The VICON will be used to determine the actual position of a stationary arm to within 1 mm, and this measurement will be compared against the measurement of the D435 primary camera
Visual Processing – Accuracy Confirmation of Spin Rate

The solar panel will be rotated using a turntable motor that will be commanded to 33 deg/s. The actuator will return position and time data, which will be used to calculate the true spin rate. This truth data will be compared to the Pixy2's calculated spin rate.

Actuator truth spin rate, accurate within 0.1°

Pixy2 tested at 3" from GP
Visual Processing – Coordinate transfer

**Requirement:**
- 2.4.5 Coordinate frames shall be defined in software with a maximum error of ±0.1 cm and ±0.5° from their actual locations

**Solution:**
- Measure coordinate frame locations with ±0.1 cm and ±0.5° accuracy

**Verification & Validation Test:**
- Utilize CU RECUV's VICON camera system with reflective spheres located on the coordinate frames of interest
- This provides measurement accuracy of ~0.1cm
- Angular accuracy dependent on sphere locations, ±0.5° can be achieved
Project Planning
Work Breakdown Structure

Deliverables
- PDD
- CDD
- PDR
- CDR
- FFR
- MSR
- TRR
- SFR
- PFR
- A1AA

Management / Logistics
- Organization Chart
- Schedule (Gantt Chart)
- WBS
- Budget
- Cost Plan
- Test Plan
- Risk Matrices
- Order Parts

Systems Engineering
- Objectives
- Requirement Breakdown Structure
- FBD
- CONOPS
- Coordinate Frame Definitions
- Full Interface Development

Visual Processing
- Sensor Selection
- Sensor Acquisition
- Software Interface
- Data Acquisition
- Develop State Determination Algorithm
- Determine Relative State of Fiducials
- Determine Relative State of Arm

Controls
- Inverse Kinematics Model of Arm
- Control Loop Design
- Control Loop Simulation
- Data Handling Software
- Actuation Software

Hardware & Testing
- Heritage Hardware Acquisition
- Grasping Arm SolidWorks
- Construct Grasping Arm
- Test Heritage Hardware
- Slip Ring Selection
- Full Test Plan
- EE Gimbals & Sensor Mount (Design, Print)
- Truth Data Determination (Vicon Test)
- Software / Hardware Integration
# Work Plan (Gantt Chart)

## MEGACLAW CDR

### Critical Design
- Write FFR
- CDR Feedback Review
- Fall Final Report
- Final Week
- Winter Break
- Order Components (Deliver Before Jan 14)

### Manufacturing, Software Development
- End Effector Gimbal
- GSE Structure
- Sensor Mounts
- Electrical Component Inspection
- Mechanical Component Testing
- Baud Rate Analysis
- Sensor Data Handling, Command Actuation &
  Manufacturing Status Review

### Component Testing
- Subsystem Testing, Integration
- (Full) System Testing

### Project Conclusion
- SFR Work
- Spring Final Review
- SPR Feedback Review
- Project Final Report

## KEY

- Yellow: Full Team, Logistics
- Grey: Breaks
- Brown: Hardware
- Purple: Electronics
- Blue: Multiple Sub-teams
- Green: Controls Software
- Red: Milestone
- Orange: Critical Path
- White: Margin

### Timeline
- December 2019
- January 2019
- February 2019
- March 2019
- April 2019

---

62
Cost Plan

Project risk will be bought down by purchasing:

• An extra Intel D435 ($179)
• An extra Pixy2 ($60)
• 2 extra Pololu force sensors ($12)
• Printing material for 2 extra end effector gimbals
## Test Plan

<table>
<thead>
<tr>
<th>Phase</th>
<th>Location (Facility)</th>
<th>Special Amenities</th>
<th>(Reservation) Feasibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic and Mechanical Component Testing</td>
<td>ASEN Senior Projects Room</td>
<td>Storage, Floor and Table Space (for test bed and components, respectively)</td>
<td>Through ASEN 4018/28</td>
</tr>
<tr>
<td>(1/14 – 1/21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subsystem Testing</td>
<td>ASEN Senior Projects Room</td>
<td>(as above)</td>
<td>(as above)</td>
</tr>
<tr>
<td>(1/14 – 3/4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VISIONS Lab (RECUV)</td>
<td>VICON motion capture system</td>
<td>CASCADE, KESSLER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Testing</td>
<td>(as above)</td>
<td>(as above)</td>
<td>(as above)</td>
</tr>
<tr>
<td>(3/5 - 4/22)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?
Back-Up Slides
Project Purpose & Objectives
Project Statement

MEGACLAW shall use a robotic arm equipped with an end effector to capture a grapple point, 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.

End Effector (EE)  Grapple Point (GP)
Primary Vision System Block Diagram

Legend:
- : Hardware Connection
- : IR Feedback
- : Visual Feedback
- : Visual Data
- : Data Flow

AR tag
Primary Vision System: Binary AR tags

Advantages:
- Fast (20-30 Hz) even with many tags
- Reliable recognition at:
  - High viewing angles
  - Wide variety of lighting conditions
- Unique IDs provided by binary codification (white pattern)
- Four corners allows for orientation estimation

Implementation used: AprilTags
- Available through OpenCV/ArUco
Primary Vision System: AR tag Algorithm

Adaptive Thresholding
• Converts input image to black-and-white

Boundary Segmentation
• Identifies contours of AR tags

Perspective Removal
• Detected tag is reshaped into a square

Bit Extraction
• Marker is divided into grid
Primary Vision System: Position/Orientation

**Orientation** provided by OpenCV library:
- Found by measuring perspective distortion of each detected tag

**Cartesian position:**
- Applying pixel position of tag to depth image to find linear distance
- Using Intel-provided functionality (along with calibration parameters) to reconstruct Cartesian position of tag
Primary Vision System: State Determination

Target state determination:
• Prior to grapple, tags on target will be visible for a full revolution

End effector state determination:
• System will have foreknowledge of geometric relationships between $n$ visible tags and the end effector center and rotation axis
• By constructing $n$ transformation matrices, $n$ estimates of the end effector's state can be obtained
• The best estimate of the true state will be the average of the $n$ states
Visual Processing Test Setup (Mock-Up)

- Two AR tags on each top end of the GP
  - Used by intel to determine spin axis

- Three colored markers centered on top of GP
  - Used by PIXY 2 to determine spine rate and fine tune axis alignment

- Additional AR tags will be placed on the tip of the claw for Intel to ensure a 90 degree alignment with the claw for successful grapple
Spin Axis Alignment – Vision Area

Diagram showing:
- A cone with angles 60° and 40°
- Dimensions: 10in height, 8in width, 0.5in depth
- A circle with diameter 11.6in
- A rectangle with sides 7.3in and 11.6in
Results of simulating the grapple point estimation accuracy 2000 times with a sample time of 30 seconds
Results of simulating the grapple point estimation accuracy 2000 times with a sample time of 60 seconds
Inside the Control Block

- Inside the control block, apply proportional control to the error and add to desire position to "trick" actuator PID control

- Assume small angle adjustments (by limiting distance between commanded points) to simplify quaternion control to linear:
  - \([q_1, q_2, q_3, q_4]' = [\phi/2, \theta/2, \psi/2, 1]'\)
Other Processes

- Inverse Kinematics: Takes commanded end effector state \([x, y, z, q1, q2, q3, q4]\) and returns joint angles. Quaternion eliminates singularities.

- Built-In Actuator PID Control: Takes a set of joint angles and sends to actuators. Actuators respond according to their supplied PID control.

- Intel, Pixy2: Senses the actual state of the end effector, returns to the algorithm.
CPE Overview
Overview of CPE's and Functionality

CPE 1: State Estimation
CPE 2: Controls
CPE 3: Hardware
Design Requirements
Software Rate Requirement

| Software - Rate | Software shall run at 0.916 Hz to support closed loop operation |

The rate of 0.916 Hz is for the entire control loop. This means that all operations within the closed loop must be performed in 1.092 seconds.

- Some control loop processes can be achieved simultaneously.
- The control loop rate becomes a primary concern during stages 3 and 4 due to the stages having the most complicated controls.
Visual Processing - Speed

- Reading in position – Intel 0.067 seconds
- Reading in position – Pixy 0.003 seconds
- Communication with ROS – Negligible
- Calculating center point – 0.015 seconds
- Visual Processing total time – 0.082 seconds
- Overall time requirement – 1.092 seconds
Visual Processing – Visibility

Visibility of AR tag's at variant conditions:

- Distances: 0.4 m, 0.5 m, 0.6 m, 0.7 m
- Angles: 15°, 45°, 60°, 75°
## Visual Processing – Visibility

<table>
<thead>
<tr>
<th></th>
<th>15°</th>
<th>45°</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7 (m)</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>0.6 (m)</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>0.5 (m)</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>0.4 (m)</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>
## Visual Processing – Visibility

<table>
<thead>
<tr>
<th></th>
<th>15°</th>
<th>45°</th>
<th>60°</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0.7 (m)</strong></td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
</tr>
<tr>
<td><strong>0.6 (m)</strong></td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
</tr>
<tr>
<td><strong>0.5 (m)</strong></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
</tr>
<tr>
<td><strong>0.4 (m)</strong></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
</tbody>
</table>
## Visual Processing – Visibility

<table>
<thead>
<tr>
<th>Distance (m)</th>
<th>15°</th>
<th>45°</th>
<th>60°</th>
<th>75°</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>0.6</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>0.5</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>0.4</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>
• Overall: All states oscillate about 0
• q4 about 1
• z-state converges to the points given
### Marker Location on GP and EE

- **State of end effector shall be measured using a decoupled sensor**
- **State of grapple point shall be measured using a decoupled sensor**

- Two AR tags on each top edge of the GP
  - Used by Intel D435
- Three colored markers centered on top of GP
  - Used by Pixy2

[Diagram showing AR tags on a GP]
• Spin rate of record player is 33.333 rev/min
  - $33.333 \times 360^\circ / 60\text{s} = 200^\circ/\text{s}$

• Averaging 5 calculations,
  - estimated spin rate = $200.736^\circ/\text{s}$

• Error = $0.736 / 200 \times 100\%$
  = 0.368%
Verification & Validation
Control Loop Accuracy
Control Loop Accuracy – VICON Test