Astro-Robotic Garden Harvester (ARGH)

Members: Aurash Filsoof, Kyle Fingerhut, Noah Kendall, Evan Luebbert, Cameron Mitchell, Samuel Oberg, Connor O’Reilly, Dominic Plaia, Collin Rasbid, Evan Shults, Thomas Shepard, Panitnan Yuvanondha

Presenters: Kyle Fingerhut, Noah Kendall, Evan Luebbert, Cameron Mitchell, Connor O’Reilly, Dominic Plaia, Thomas Shepard
Purpose & Objectives

Credit: Sierra Nevada Corporation

Commercial Space Stations

Modernization and Automation

Robotics Tasking

ARGH
Customer Requirements

- Remotely Collect Information
- Remote Station Oversight
- Physical Interaction
- Automation
- Repeatability
- Extensibility

Space Gardening

Credit: Megan McArthur
Credit: Sierra Nevada Corporation
Concept of Operations

1) Task Identification
2) Target Acquisition
3) Orientation
Concept of Operations (cont.)

4) Harvest
5) Store Target
6) Reset & Repeat
Purpose & Objectives

Design Solution

Critical Project Elements

Requirements & Satisfaction

Risks

Verification & Validation

Planning

Concept of Operations

1) Task Identification
2) Target Acquisition
3) Orientation
4) Harvest
5) Store Target
6) Reset & Repeat
Design Solution

Remote Station

Target

ARGH Computer

Camera + Mount

UR Computer

0.88 m

0.07 m

1.2 m

>0.3 m
Design Solution - Arm

- Universal Robotics UR10e provided by Sierra Space
- Great dexterity with minimal error

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Reach</td>
<td>1.3 m</td>
</tr>
<tr>
<td>Mass</td>
<td>33.5 kg</td>
</tr>
<tr>
<td>Pose Repeatability</td>
<td>0.05 mm</td>
</tr>
<tr>
<td>Power Requirements</td>
<td>350 W</td>
</tr>
<tr>
<td>Maximum Payload</td>
<td>12.5 kg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Axis movement</th>
<th>Working range</th>
<th>Maximum speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>± 360°</td>
<td>± 120°/s</td>
</tr>
<tr>
<td>Shoulder</td>
<td>± 360°</td>
<td>± 120°/s</td>
</tr>
<tr>
<td>Elbow</td>
<td>± 360°</td>
<td>± 180°/s</td>
</tr>
<tr>
<td>Wrist 1</td>
<td>± 360°</td>
<td>± 180°/s</td>
</tr>
<tr>
<td>Wrist 2</td>
<td>± 360°</td>
<td>± 180°/s</td>
</tr>
<tr>
<td>Wrist 3</td>
<td>± 360°</td>
<td>± 180°/s</td>
</tr>
</tbody>
</table>
Design Solution - Object Recognition

- Mask R-CNN
- Machine learning image processing software
- Higher precision relative to other softwares
- Longer processing time
Design Solution - End Effector

- Robotiq 2F-85
- Acquired by Sierra Space
- Can be programmed to close at a given distance

<table>
<thead>
<tr>
<th>Mass</th>
<th>0.9 kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Object Diameter</td>
<td>45 mm (Encompassing Enclosure Method)</td>
</tr>
<tr>
<td>Stroke</td>
<td>85 mm</td>
</tr>
<tr>
<td>Communication</td>
<td>Modbus RTU</td>
</tr>
</tbody>
</table>
Design Solution - Sensor

- Intel Realsense D435
- Both RGB and Depth sensing
- Intel software package compatible with ROS, python, matlab and other languages

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>124 mm × 26 mm × 29 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Requirements</td>
<td>3.5 W</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>10 - 90 fps depending on depth requirements</td>
</tr>
<tr>
<td>Accuracy</td>
<td>≤2% (up to 2 Meters and 80% ROI, HD Resolution)</td>
</tr>
</tbody>
</table>
System Block Diagram

- **PWR Source**
  - 115 VAC

- **UR10e (Arm)**
  - 2F-85 COMM
  - 24 VDC

- **Robotiq 2F-85 (Gripper)**
  - 115 VAC

- **UR Computer**
  - PWR
  - UR10e COMM

- **ARGH Computer**
  - PWR
  - WiFi XMTR
  - USB
  - RGB-D

- **Intel RealSense D435**
  - USB-C
  - 5 VDC

- **Remote Station**
  - WiFi RCVR

- **Power Communication**
Critical Project Elements

- **Sensor Positioning**
- **Object Recognition**
- **Robotics/Pathfinding**
Sensor Mounting

- Sensor D435
- 21° incline
- Front box for gripper to grab

Railing
Slider
Sensor Mounting
Sensor Positioning

Multiple Sensor Positions

- Position A
- Position B

Distance:
- 1.35 m
- .9 m
- ~1 m (max)
Sensor Positioning

Cart Mounted
Multiple locations to increase object detection and path optimization
Live Image Feedback
Object Recognition

“Are you able to detect tomatoes and separate them from their surroundings”

Two Viable Network Choices

**Selected Network**

**MaskRCNN**

Mean Average Precision: 0.84

**YOLACT**

Mean Average Precision: 0.30
Robotics

Core Elements:

- Self Collision Avoidance
- Pathing
- Object Collision Avoidance
- Practical Agility
Robotics (Self Collision Avoidance)

“Will a desired position cause the robot to intersect/damage itself?”

Generated from UR10e .URDF file
- ‘Universal Robotic Description Format’

‘Monte Carlo’ Mesh Generation
- MoveIt! Setup

URDF Representation
Generated Collision Mesh
What is the fastest\textsuperscript{1} way to get from position A to position B?\textsuperscript{2}

MoveIt! w/ integrated ‘RRTConnect’ algorithm for inverse kinematics

- Constrained by:
  - Joint limits
  - Self-Collision Mesh

- ‘Special Case’ Pathing Requirements:
  - Sensor Repositioning: Linear Pathing
  - ‘Complex’ Harvest Positions: Cartesian End-Effector Pathing

\textsuperscript{1}Most efficient, ie least possible movements
\textsuperscript{2}‘Rapidly-exploring Random Trees’
“Will the desired path lead to intersection between collision mesh and objects in the environment?”

- If no:
  - Execute movement
- If yes:
  - Execute until defined ‘collision threshold’
  - Re-plan movement path
  - Check for new collisions along new path
  - Repeat

‘Collision’ -> Any intersection of Robot’s collision mesh with objects in environment
Robotics (Practical Agility)

‘Is the robot physically capable/sufficiently dextrous to complete the required motions?’

- Harvesting
  - Move end effector to directed position and rotate along base axis
- Sensor Positioning
  - Move to three preset positions and repeat
# Primary Project Risks

<table>
<thead>
<tr>
<th>ARGH RISK</th>
<th>Negligible (1)</th>
<th>Minor (2)</th>
<th>Serious (4)</th>
<th>Critical (7)</th>
<th>Catastrophic (11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improbable (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Occasional (3)</td>
<td></td>
<td></td>
<td>Incorrect Object Detection</td>
<td>Robot Collision</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sensor Calibration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Liquid Damage</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Machine Learning Failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probable (4)</td>
<td></td>
<td></td>
<td></td>
<td>Critical (7)</td>
<td></td>
</tr>
<tr>
<td>Frequent (5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sensor Mounting Error

- Many sources of error
  - Mounting
  - Pointing/Rolling
  - Depth

- Transform between frames
  - Arm/Mounting Frame
  - Camera Cartesian Pointing
  - Camera Cylindrical Pointing

- Final Errors:
  - In-Plane Radial Error
  - Out-of-Plane Error

- Resulting in total 3D radial error
Sensor - Orthogonal Plane Error

Mounting Frame Error

- Along the rail (y-axis)
  - Additional error due to arm inaccuracy, sliding
- Across the rail (x-axis)
- Up and down from rail (z-axis)

Rotate 21 degrees into pointing frame.
Sensor - Orthogonal Plane Error

Errors

- Rolling
  - Due to misalignment of grid
  - Proportional to distance from centerline
- Pointing
  - Due to pitch and yaw
  - Proportional to distance from camera
Sensor - Out-of-Plane Error

Depth Sensor

- Due to intrinsic properties of camera
- Error proportional to depth
Sensor - Total Error

Cylindrical Coordinates

- Radial In-Plane Error
- Pointing Axis Error
Limitations

- Plant appears larger in frame when closer or larger
- Camera has a maximum FOV
- Camera has a minimum distance to closest point on plant
Robotics

Pointing Accuracy (End Effector)
- Repeat a commanded pose
- Measure end effector variance

Gripper Capability Assessment
- Twisting Capability
- Gripper Grasp Repeatability

Tomato Interaction Reliability
- Vine Interactions
- Extensive Tomato Grasp Tests
- ‘Imperfect’ Gripper Positioning
Fall Planning

You are here.
Spring Planning
Planning
Planning

Winter Break
Planning

Status Column

- Up/Next: 4
- Working on it: 9
- Done: 15
- Future steps: 30
Cost Plan

Provided Items, Cost = $0.00
- UR-10e Robotic Arm
- Intel Realsense D435 sensor
- Robotiq 2F-85 gripper

Manufacturing Workshop Training, Cost = $40.00
- 2 team members require manufacturing training
- Max Cost of $40.00 if a second training required

PC component requirements, Cost = $3,000.00
- Max Cost $3000.00
- Storage: SSD M.2, 256 GB
- RAM: DDR4, 16 GB
- CPU: Modern, Intel i7
- GPU: CUDA Capable, 8/12 GB
- Motherboard: USB-C & Ethernet capable
ARGH Budget

Provided Funds: + $5000.00

Pilot Damage Deposit: - $200.00  = $4800.00
Training: - $40.00            = $4760.00
PC: - $3000.00               = $1760.00
USB-c Cable: - $30.00        = $1730.00
Remote desktop software: - $500.00 annually = $1230.00
Provided technology: - $0.00  = $1230.00

Remaining Funds: + $5000.00

All values are max potential cost, Max projected cost = $3670.00
Safety Risk
Backup Slides
MaskRCNN Network (Requirements & Satisfaction)
Foam Gripper

Model
- Force is limited by friction and internal tomato maximum
- Forces as function of deformation
- Deformation is commanded by gripper diameter

Assumptions
- Small cross section of actual continuous contact
- Internal forces in tomato and foam are the same
- Parameters for tomato were chosen based on specific data from research papers

Reality
- Contact changes with compression (Positive)
- Parameters of tomato may change with variety and ripeness

Furthermore, gravity is present
The variable $f_0/E_f$ represents the thickness to the Young’s Modulus ratio of the foam.

The system can be treated as a tomato which is wider: Total deformation is absorbed nonlinearly by the foam.

Weaker foam requires more foam to produce the minimum required force for friction, but also makes up for more of the total deformation.
Foam Gripper

As the foam gets weaker, or conversely thicker, the minimum and maximum allowable grip diameter separates, allowing the inherent error to produce safe results.

This results in a minimum ratio which produces acceptable results based on the system parameters used.
The total deformation of the tomato can be commanded through the gripper diameter by following 3 separate rules:

- Matching the minimum force for necessary friction
- Matching the average force for average deformation allowable
- Matching the maximum force for maximum deformation

Thus, a selection of a mixture between the low range and mid range would be desirable.
Foam Gripper

Several other parameters alter the bounds by which deformation is allowable: Mass, Diameter, and Area of Contact. Area produces some of the best effects by drastically reducing the minimum force required to produce the necessary friction. Diameter allows for more deformation, but there are inherent limitations as the gripper diameter must exceed the tomato and two slices of foam in dimensions. Mass inversely impacts the range, restricting it by requiring more friction to maintain grip. This is ignorable in space due to low gravity.

![Graphs showing deformation of tomato vs mass, diameter, and area of contact.](image-url)
Deformation of foam can cause cell collapse at only 10% deformation, this is impossible to avoid based on the model and tomato diameter.

Instead, deformation total of 60% (within plateau region) for 15mm of foam.

Thus a desirable $f_0/E_f$ ratio is no more than 2, but given the worst case parameters used, should not be below 1.

The resulting $E_f$ ranges from 7.5-15 MPa. Lower $E_f$'s are also allowable as contacting area should increase with deformation.
Sensor Mounting (Dimension)
Sensor Mounting (Different Views)

Back View

Front View

Top View
# Robotics Backup

## Movement

<table>
<thead>
<tr>
<th>Pose Repeatability per ISO 9283</th>
<th>± 0.05 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Axis movement</strong></td>
<td><strong>Working range</strong></td>
</tr>
<tr>
<td>Base</td>
<td>± 360°</td>
</tr>
<tr>
<td>Shoulder</td>
<td>± 360°</td>
</tr>
<tr>
<td>Elbow</td>
<td>± 360°</td>
</tr>
<tr>
<td>Wrist 1</td>
<td>± 360°</td>
</tr>
<tr>
<td>Wrist 2</td>
<td>± 360°</td>
</tr>
<tr>
<td>Wrist 3</td>
<td>± 360°</td>
</tr>
<tr>
<td><strong>Typical TCP speed</strong></td>
<td>1 m/s (39.4 in/s)</td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td>500 Hz Control frequency</td>
</tr>
<tr>
<td>Modbus TCP</td>
<td></td>
</tr>
<tr>
<td>PROFINET</td>
<td></td>
</tr>
<tr>
<td>Ethernet/IP</td>
<td></td>
</tr>
<tr>
<td>USB 2.0, USB 3.0</td>
<td></td>
</tr>
<tr>
<td><strong>Power source</strong></td>
<td>100-240VAC, 47-440Hz</td>
</tr>
</tbody>
</table>

---

## Diagram

[Diagram showing the connection between a Robot and ROS-Driver]

- **Robot** 
  - Primary Interface 
  - Secondary Interface 
  - RTDE Interface 
    - External Control 
    - RS485 Daemon 
    - Request 
    - ScriptSender 
    - URScript 
  - Controllers 
    - speed_scaling 
    - tcp_force 
    - joint_states 
    - joint_control 
  - IO Interface 
    - Read / Write

- **ROS-Driver** 
  - Initialization 
    - calibration_check 
  - Custom Scripts 
    - URScript snippets 
  - RS485 Device
## Project Risk Analysis

### Probability

<table>
<thead>
<tr>
<th>Probability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improbable</td>
<td>Very unlikely to occur during the life of the device</td>
</tr>
<tr>
<td>Remote</td>
<td>Unlikely in common use</td>
</tr>
<tr>
<td>Occasional</td>
<td>Occurs infrequently but may occur under certain identifiable conditions</td>
</tr>
<tr>
<td>Probable</td>
<td>Expected to occur with some regularity; Feedback and analysis can verify predictability of occurrence</td>
</tr>
<tr>
<td>Frequent</td>
<td>Occurs on a regular basis</td>
</tr>
</tbody>
</table>

### Severity

<table>
<thead>
<tr>
<th>Severity</th>
<th>Negligible</th>
<th>Minor</th>
<th>Serious</th>
<th>Critical</th>
<th>Catastrophic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improbable</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Remote</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>Occasional</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>Probable</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>Frequent</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>35</td>
</tr>
</tbody>
</table>

- **Negligible**: Nuisance w/ little or no potential to result in damage to robotic device or environment. Human safety unaffected by device.
- **Minor**: Annoyance. Potential reduction in device accuracy. Not expected to result in any damage to plant specimen or device. Human safety unaffected by device.
- **Serious**: Partial system functionality loss, device accuracy reduced. Potential damage to test specimen or device. Human safety unaffected by device.
- **Critical**: Subsystem failure, device functionality greatly reduced. Imminent damage to test environment (specimen and/or housing bay). Potential for human contact injury.
- **Catastrophic**: System failure. Imminent damage to device and test environment. Potential for serious human injury.
Project Risk Analysis

- Collision between robot and environment
  - Under defined bounding box
  - Change of environment
- Collision between different parts of robot
- Collision between arm and sensor
- Power loss / electrical failure
  - Cord cut
  - Cord unplugged
  - Environment power loss
- Sensor calibration error
- Incorrect object detection
  - Non-detection
  - Incorrectly determines ripeness
  - Incorrectly identifies tomato
- Gripper calibration
- Liquid damage to robot
  - Water drippage
  - Busted tomato
- Machine learning doesn't work
Project Risk Mitigation

- Robot collision
  - Place boundary zone with lockout procedures.
  - Conduct daily human visual inspection
  - Adjust bounding box to place obstructions around sensor to prevent unintended collisions but allow a single, pre-trained path for sensor movement
  - On sensor movement, test with dummy sensor to prevent damage, and check calibration at specified time intervals
- Machine learning failure
  - Risk was mitigated by working with 2 different machine learning programs. Both now work, so no longer a risk.
- Incorrect object detection
  - Ripeness is as soon as red is visible. Therefore, only pick tomatoes that have a mask generated from at least 50% red (Up to change)
  - Continuous training until desired accuracy is achieved
- Liquid damage
  - Waterproof seal of some form on joints; Possible option: Use 2 circular shells at each joint (one just smaller than other), to prevent liquid from getting in joints. Do a bi-weekly inspection to ensure no spillage gets in.
<table>
<thead>
<tr>
<th>Name</th>
<th>October 2021</th>
<th>November 2021</th>
<th>December 2021</th>
<th>January 2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connor O'Reilly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noah Kendall</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dominic Pila</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evan Luebert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evan Shultz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Awanah Filipow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thomas Shepherd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parinath Yuvanon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kyle Fingherutz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collins Rashid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samuel Oberg</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cameron Mitchell</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>