MAE C263C Final Project: Catcher Arm

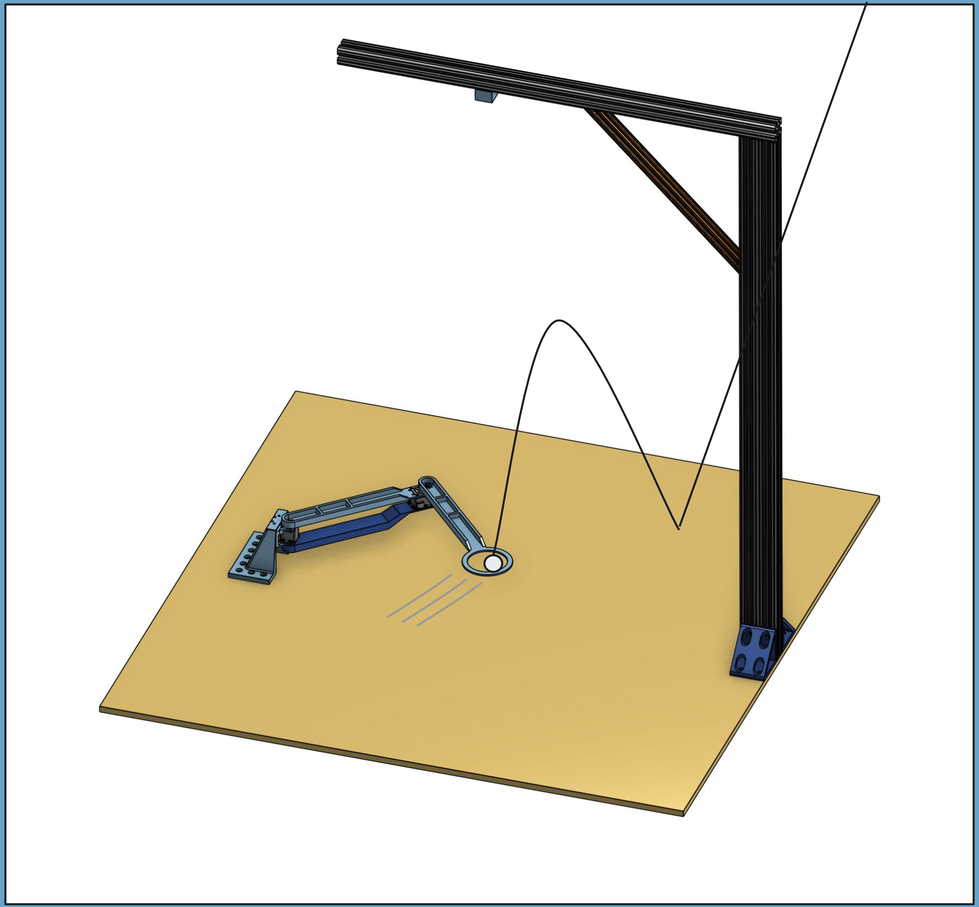
Team 6: Manuel Gonzalez Parra, Justin Lee,
Trevor Oshiro, Eli Whitaker

Overview

Goal: A two-link planar manipulator that can catch a ball in mid air

Motivation: Push these actuators to *need* centralized control through fast, dynamic movements

Applications: Sports, juggling, projectile motion tracking, entertainment

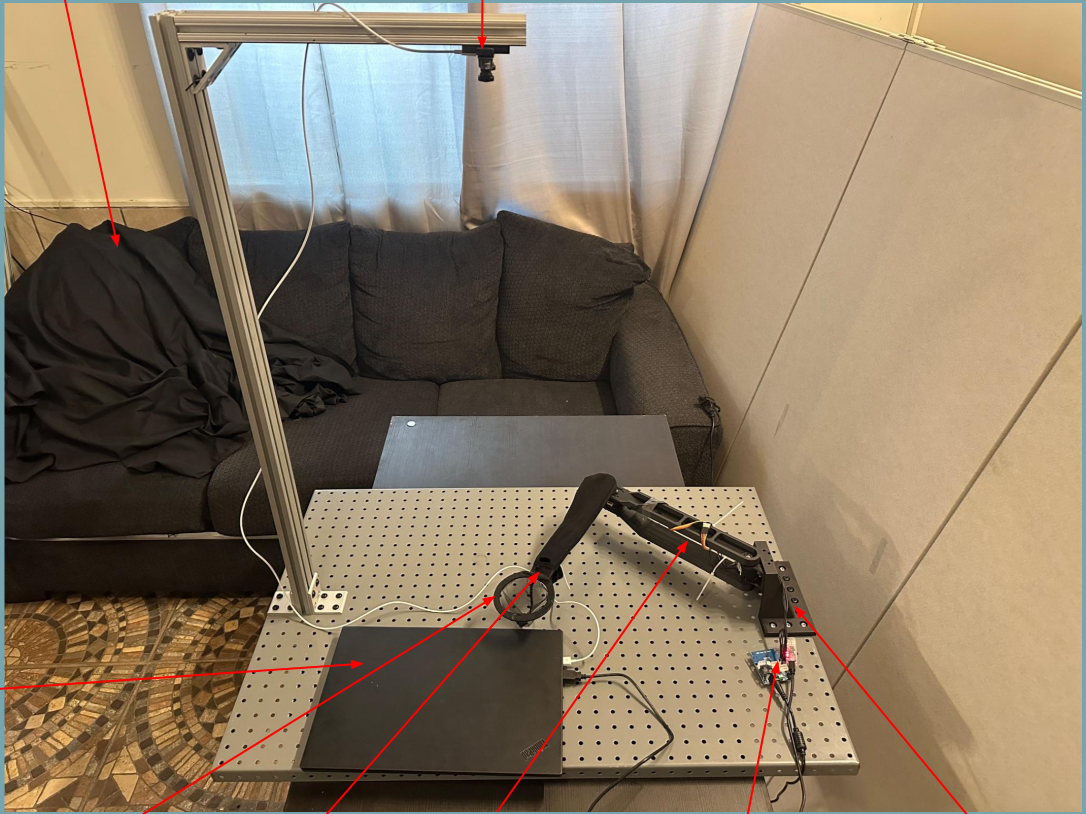


Final Configuration and Results

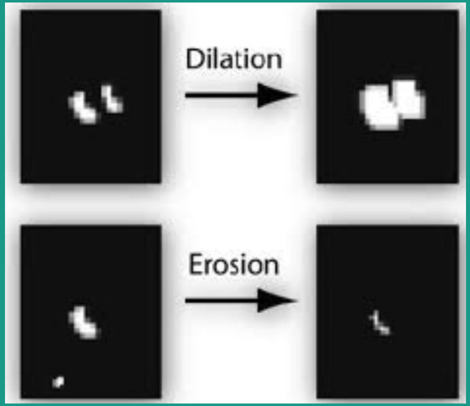
- 2 subsystems
  - Computer Vision used to estimate landing spot
    - Continuously updating target endpoint as opposed to tracking current position
  - Joint Space Robust Controller used to move arm to target
    - Position/velocity based on motor sensors
  - ~75% catch rate

Design Considerations

- Rigid connections between arm and camera for consistent coordinate systems
- Black arm and panel to improve detection of white ball
- Camera mounted high to minimize lens distortion
- Long links needed to be light yet rigid



Methods (Computer Vision)



A human-computer collaborative workflow for the acquisition and analysis by Reda et al.

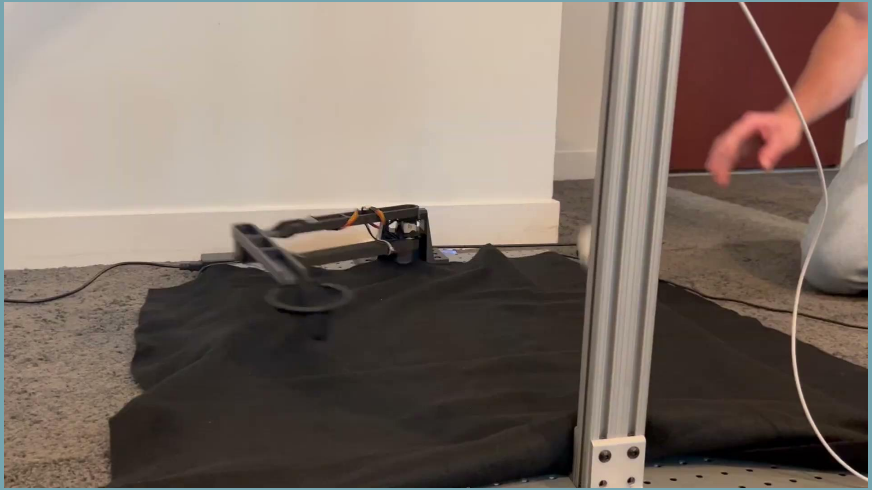
- Centroid of Ball Estimator
    - Binary Threshold
    - Morphological Opening
        - Erosion
        - Dilation
        - Opening
    - Contour Extraction (Suzuki-Abe Algorithm)
        - Finds connected white-pixel boundaries
    - Select Largest Contour
    - Spatial Moments
        - Zeroth Moment
        - First Moments
        - Centroid
- Area Calculation
    - Green's Theorem
- Height estimation
- Projectile Kinematics
    - Filters

Controllers

- First functional system achieved using decentralized joint control
- Implemented and fine-tuned four controller types to increase task success rate
- Each controller contributed unique advantages toward optimizing performance

Controller Type	Pros	Cons
Decentralized Joint Control	• More responsive • Simple to implement and debug	• Jerky movements • Sensitive to noise
Joint space Inverse Dynamics Control	• More responsive relative to robust control • Smaller transient compare do decentralized control	• Computationally intensive • Sensitive to model mismatch or noise
Operational space Inverse Dynamics Control	• Reduces computational load by working in task space • Smaller transient compare do decentralized control	• Difficult gain tuning due to geometry dependence • Highly sensitive to singularities in Jacobian • Prone to cascading errors from inaccurate kinematics
Joint space Robust Control	• Smooth and fluid joint motion • More tolerant to minor disturbances	• Sensitive to jittering • Tends to produce slower settling times

Controller Examples

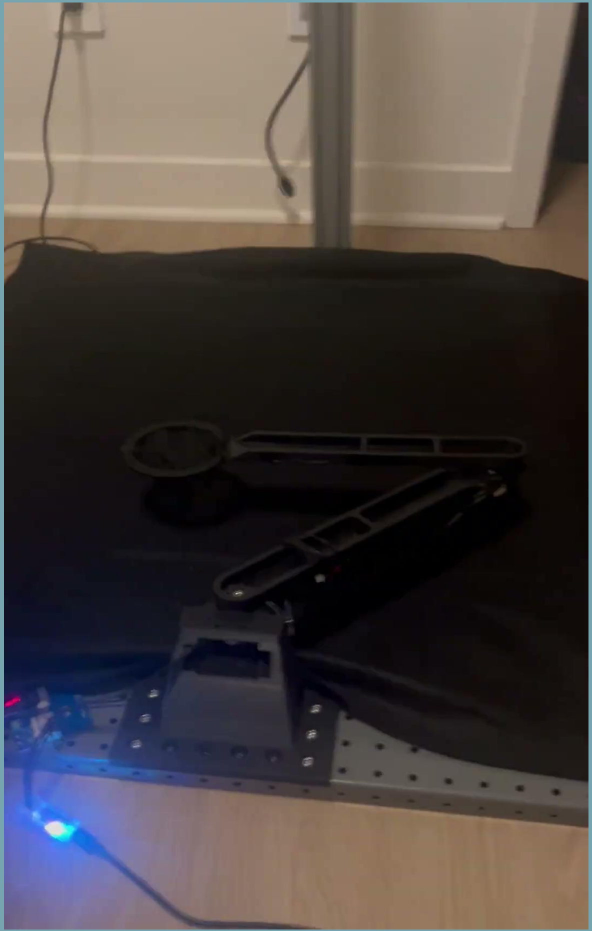


Decentralized PID control

Operational Space Inverse Dynamics

Controller Examples

Joint Space Inverse Dynamics

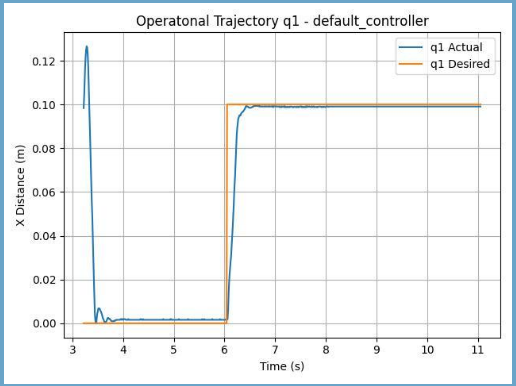


Robust Joint Space Control

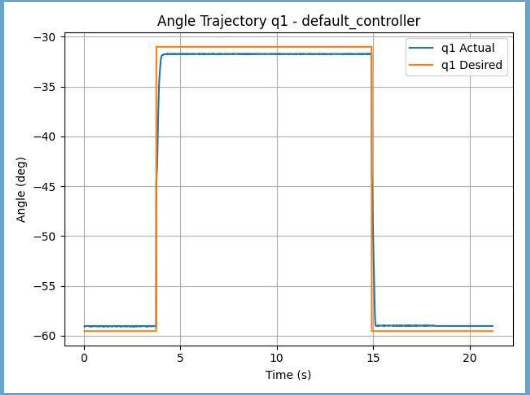
Controller Comparison

- Joint Inverse Dynamics and Decentralized PID best practical performance
- Robust more smooth and oscillation closer to the end point
- Operational Inverse Dynamics sensitive in tuning

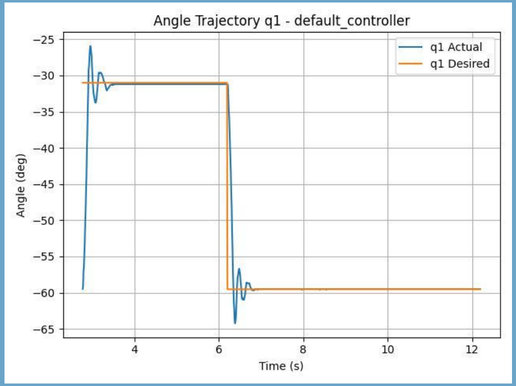
Operational space Inverse Dynamics Control



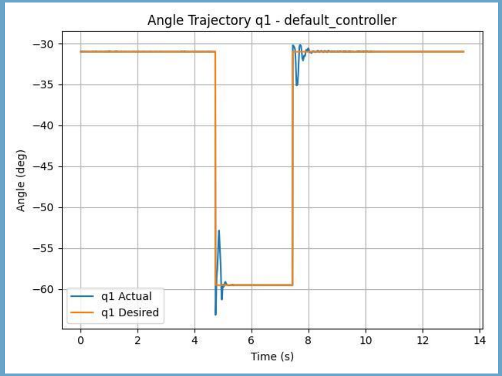
Joint space Inverse Dynamics Control



Decentralized Joint Control



Joint Space Robust Control



Final Results

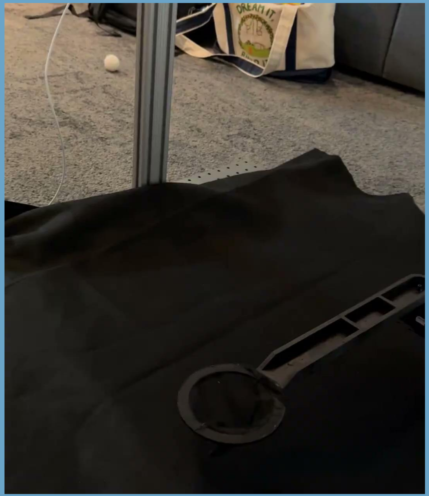
Successes

- 75% catch rate using final robust controller
- System successfully implemented with full ROS2 integration
- Accurate centroid detection and task generation

Controllers

In order of Pure Qualitative Performance:

1) Robust Control
2) Decentralized Control
3) Joint space Inverse Dynamics Control
4) Operational Space Inverse Dynamics Control



Failures

- Lack of sufficient tuning in operational space
- No joint safety limits
- Camera vision conditions and joint position noise reduced precision

Improvements

- Improve system architecture and streamline gain tuning process
- Add joint limit constraints or emergency stops to prevent damage from overshoot
- Controlled lighting environment (mounted light, reflective taping around ball, infrared camera)