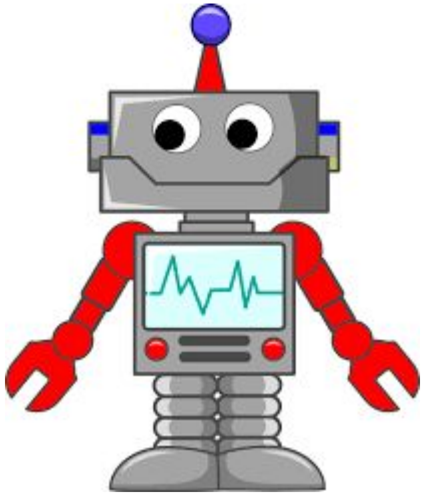
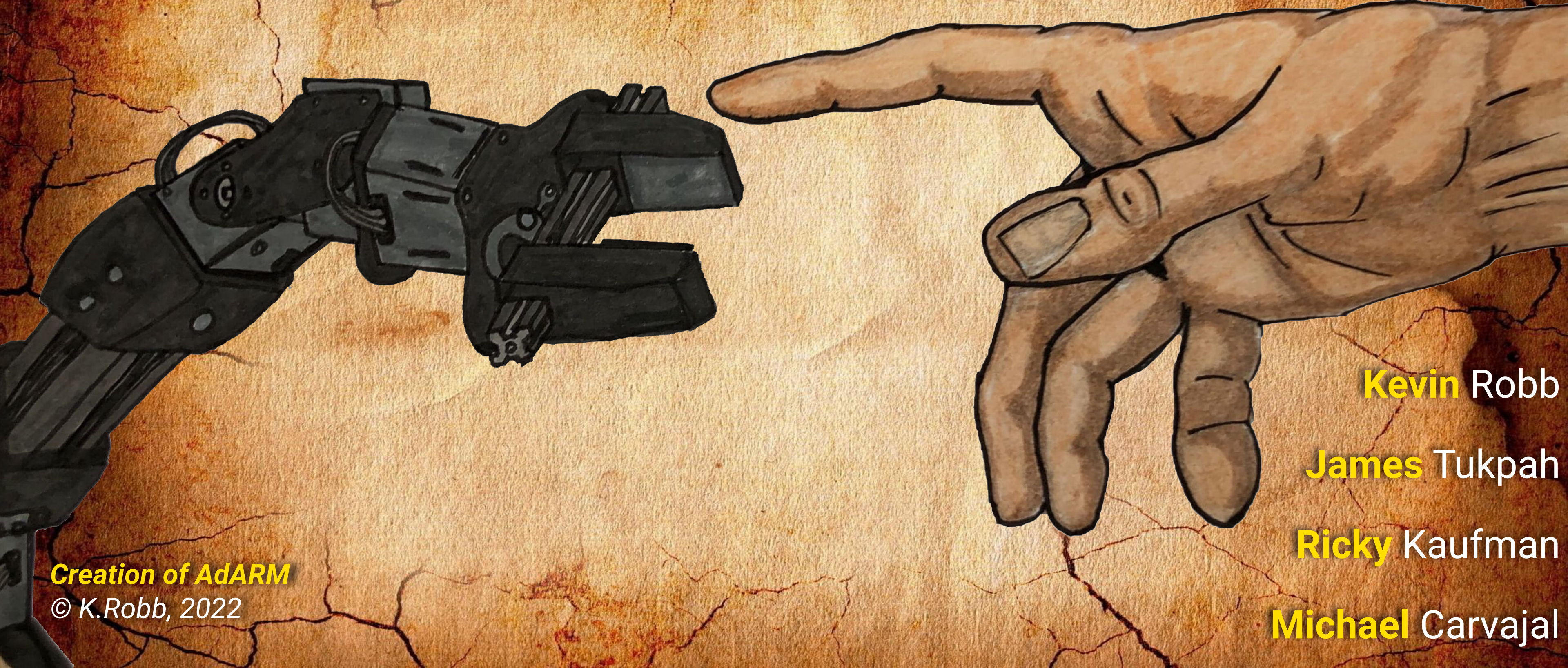
The Presentation Will Be Starting Shortly.

(Please Hold Applause Until The End.)



MichelARMgelo



Creation of AdARM
© K.Robb, 2022

Kevin Robb

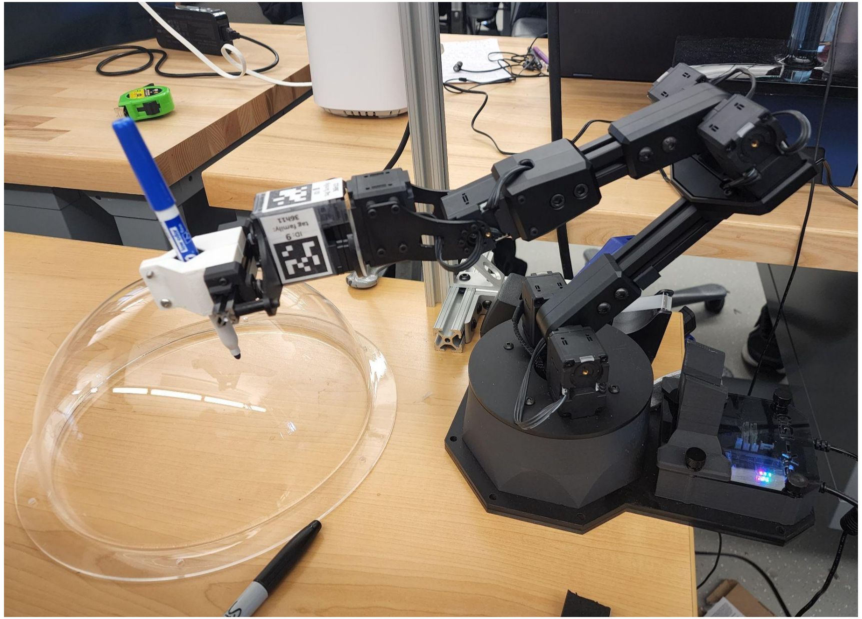
James Tukpah

Ricky Kaufman

Michael Carvajal

The Task

Draw with a robot arm!

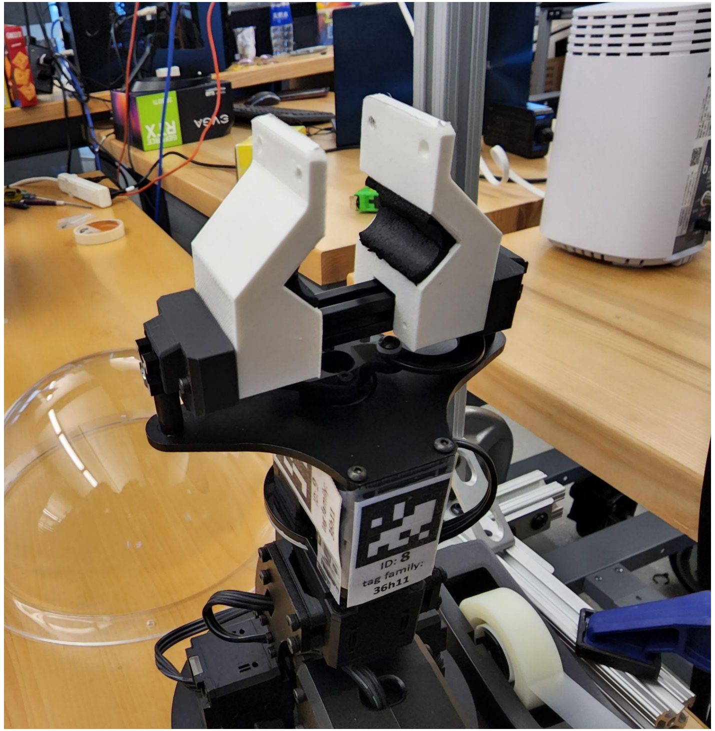


Specific Goals & Progress

- ✅ On flat paper, draw a variety of pre-planned shapes.
- ✅ Draw random shapes within the reachable workspace.
- ✅ On a domed surface, draw smooth arcs.
- ✅ On the interior of a domed surface, draw smooth arcs.
- ❌ Recognize and replicate a drawn path.
- ❌ Write alphanumeric characters.

Components

- Interbotix PincherX 150 Arm & Software Packages
- ROS Noetic or Melodic
- Standard Sharpie or Thin Expo Dry Erase Marker
- Custom 3D-Printed End Effector
- Dry erase board and hemisphere
- RealSense D435 Camera & Overhead Mount
- Printed AprilTags



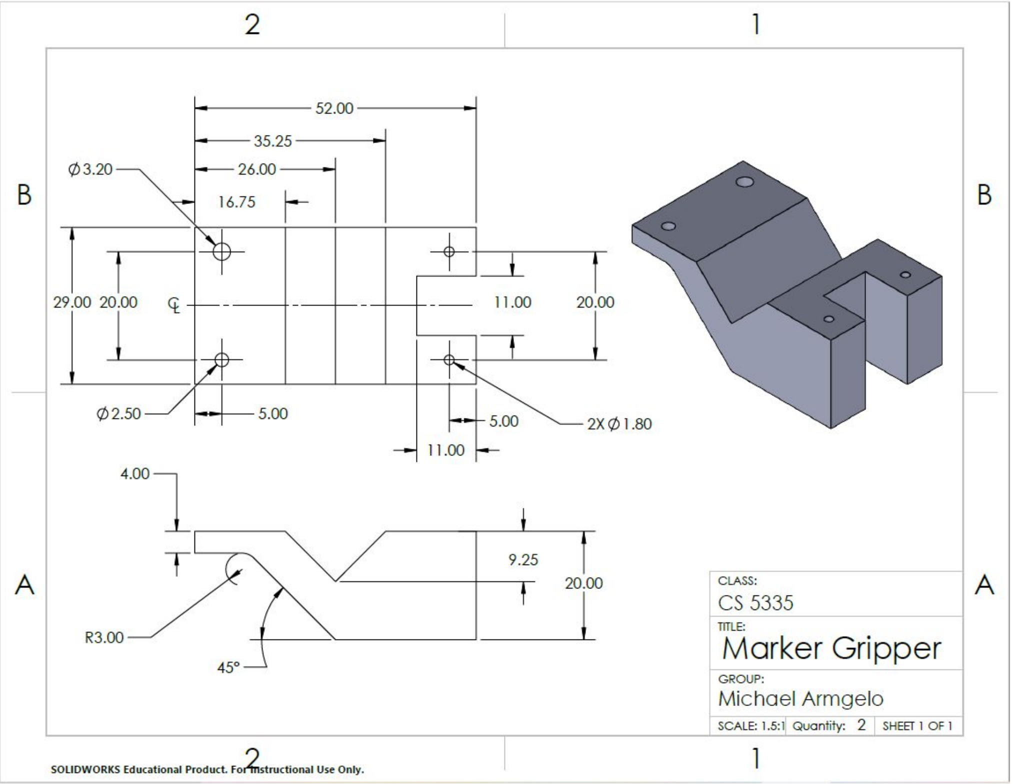
Environment Setup and Assumptions

- Robot arm is clamped to the edge of a table.
- Canvas is centered in front of the robot along its +X axis (at Y=0).
- There are no obstacles above Z=0 within the workspace we define in `constraints.txt`.
- Camera is mounted overhead; camera's FoV can see both the canvas and the arm.
- Geometry of canvas is known ahead of time.

Achievements (Presentation Overview)

- Design of custom 3D-printed pen gripper.
- Setup and manual control of the arm.
- Creation of ROS nodes to control the arm.
- Modification of Interbotix-provided trajectory code.
- AprilTag detection to allow leniency in environment setup.

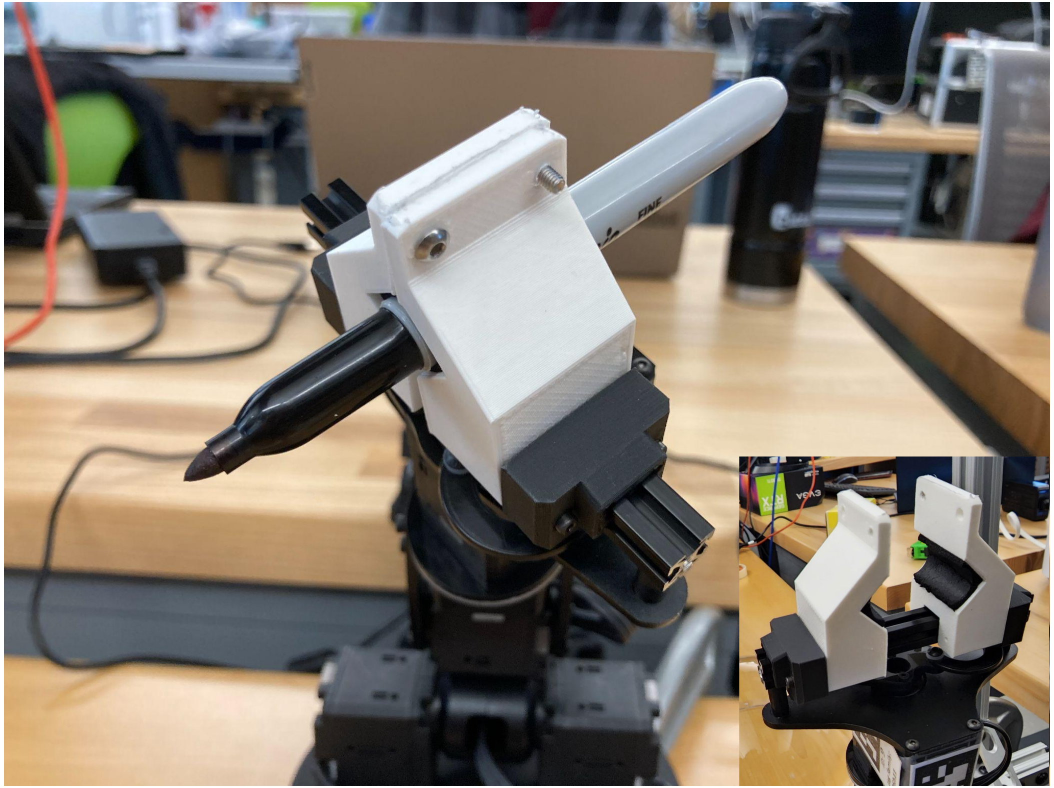
Gripper Design



Pen Position

- Designed to be mounted to the carriage pieces of the pincher end effector.
- Width must accommodate the minimum and maximum distances between the carriages.
- Major axis is perpendicular to the rotation axis of the arm's wrist (the last joint in kinematic chain)
- The pen's axis is close to supporting linear rail to minimize the moment arm of the reaction forces (from writing) that exert a torque on the wrist joint

Gripper Design



Pen Compatibility

- V-groove cutouts in each gripper self-center the cylindrical shape of a pen’s body and guarantee two points of contact per side of the pen
- Screw holes added to far-end of gripper to clamp marker even tighter
- Foam padding adapts to diameter of the marker

Arm Setup & Manual Control

Our README at **https://github.com/jtukpah/MichelARMgelo** contains full setup details.

- Interbotix workspace contains functions to set EE pose directly.
- Custom python scripts:
    - Control EE pose from command line, either one point at a time or with a series of points to form a longer trajectory.
    - Read trajectory from a file to command robot.

https://www.youtube.com/watch?v=hnT60kQ3MnY

ROS Architecture

Trajectory Processing Node

- Create a trajectory and publish [X, Y, Z, R, P].
- Mode is user-specified.

Control Node

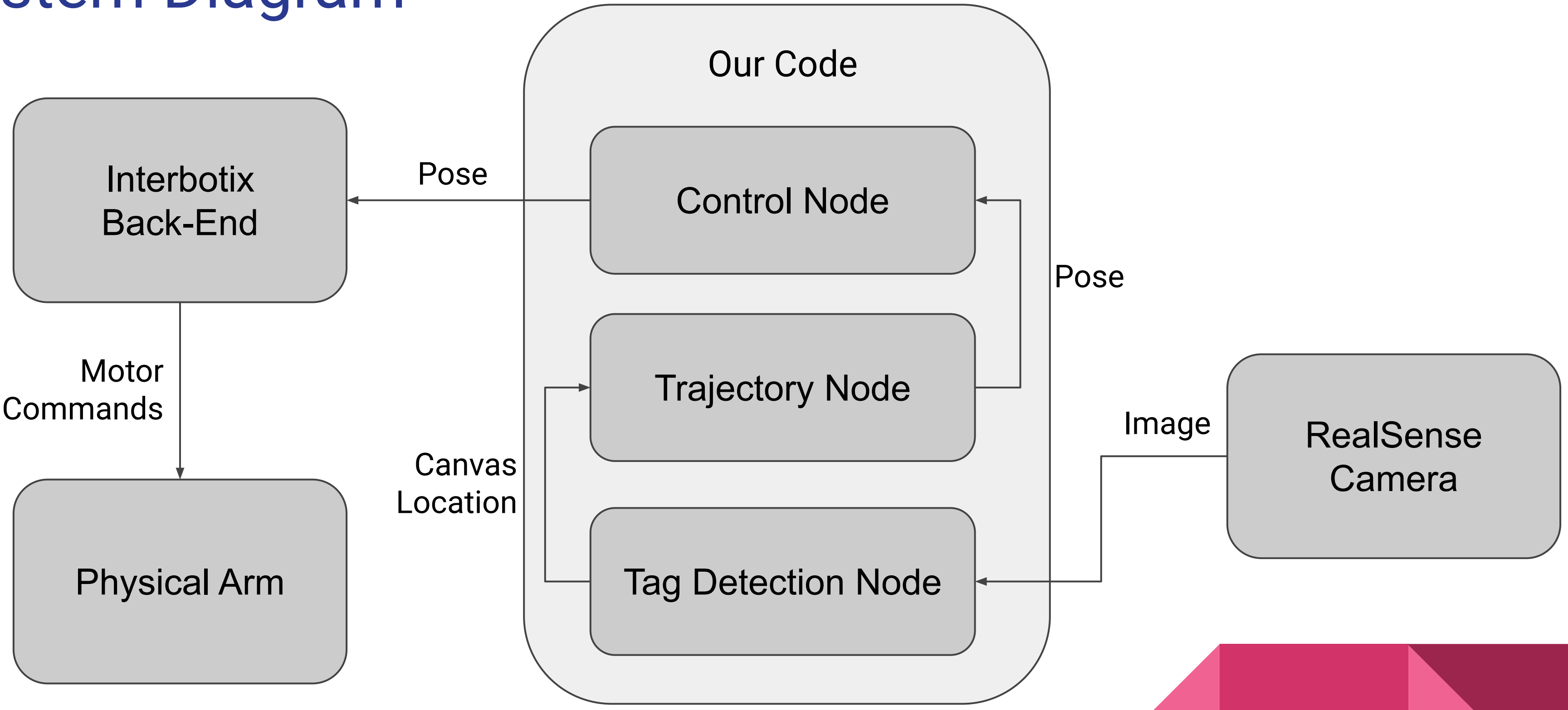
- Subscribe to trajectory points.
- Limit trajectories within constraints.
- Command the arm.

Tag Detection Node

- Subscribe to AprilTag detections on both canvas and arm.
- Compute pose of end-effector in global frame.
- Set geometry of spherical canvas relative to arm frame.

https://www.youtube.com/watch?v=hnT60kQ3MnY

System Diagram



Creating Smooth Trajectories

Until now, all trajectories have been series of points, and the arm stops at each point, so motion is jerky and slow.

For smooth motion, we can use the built-in `set_ee_cartesian_trajectory` function, which takes a relative motion in [X, Z, R, P] and interpolates to move the end-effector (tool frame) in a straight line.

To draw smooth arcs on the sphere, we made a modified version of the function which takes into account the geometry of the surface, and allows us to draw on the convex dome.

With a slight change to the interpolation equations, we can also draw on the interior of a domed surface.

https://www.youtube.com/watch?v=hnT60kQ3MnY

AprilTag Detection & Pose Correction

We use the overhead camera to detect AprilTags on both the canvas and the arm.

This lets us determine the position of the sphere's center relative to the arm, so our code can work with more than one exact setup environment.

Arm performance is sloppy, especially at high X, so we can use AprilTags on EE to give more reliable localization than dead reckoning with our commands, and send small relative corrections to maintain the desired trajectory.

Roadblocks

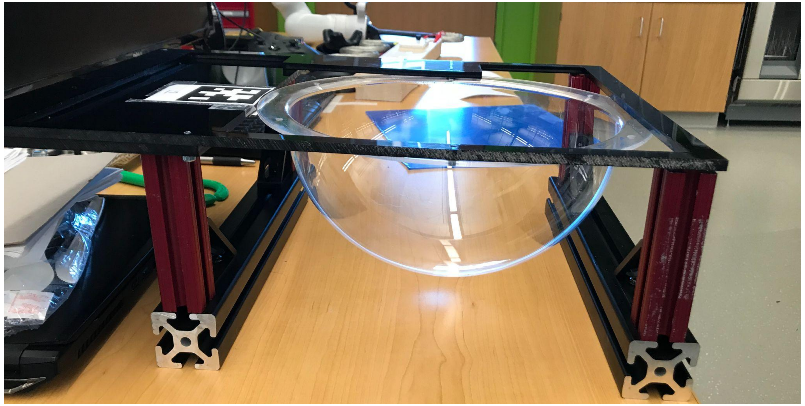
- Hard to control smooth motion without doing all the inverse kinematics ourselves.

- 5DOF arm has low dexterity in pitch and—*when working within the constraints of the provided set_ee_cartesian_trajectory function*—<u>none</u> in yaw, limiting potential drawing surfaces/areas. This means for smooth motions the only way to draw in the Y axis is by using the roll.

- Reachable flat workspace is small, since arm cannot intersect with itself, and reaching too far over-torques the shoulder servos, causing the arm to collapse and require a power cycle. Arm performance could potentially benefit from working with a slanted (or even vertical) drawing surface, rather than the horizontal tabletop we tested on.

Future Work

- Switch to using joint-specific trajectories for smoother (non point-based) motion that we have full control over. This will require doing the inverse kinematics on our end instead of letting interbotix handle it.
- Create smooth motions for more complex paths than arcs and lines, such as alphanumeric characters, and make a more robust way of non-manually generating such trajectories.
- Process an image and automatically create a trajectory for the robot to draw it.



Questions?