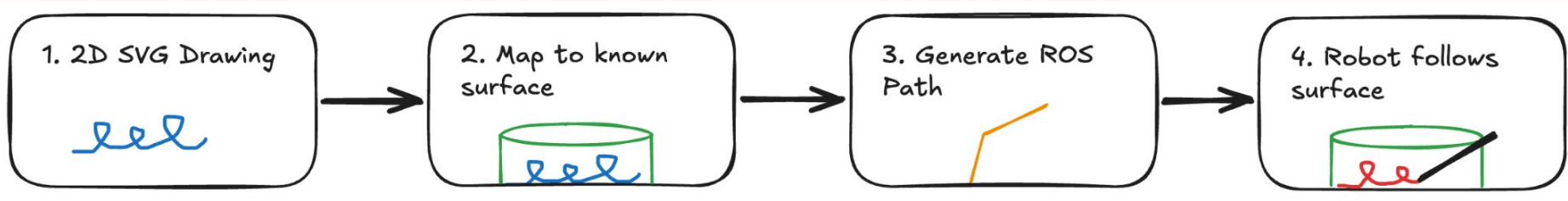
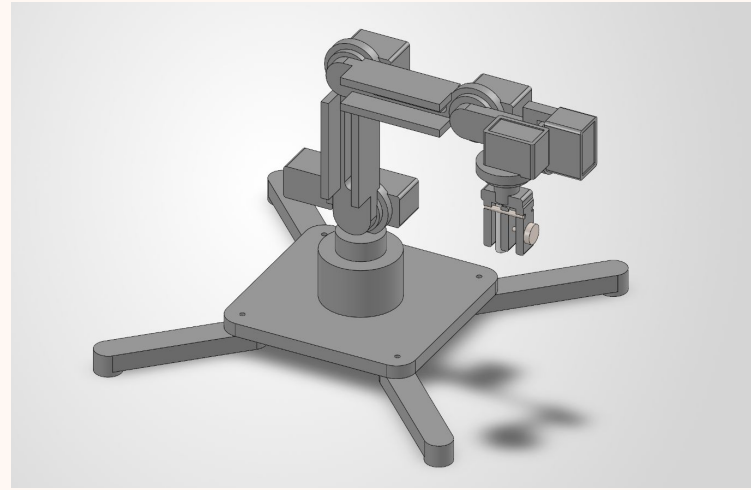


Use Case/Application

- Transfer **digital artwork** onto **curved objects**
- Adapt drawing motion to changing surface orientation
- Support **low-volume, high-variation** custom artwork



Quantitative Use-Case Requirements

Requirement	Target	Justification
Target specifications	One surface; shape, curves, and dimensions known	Keeps the testing surface bounded.
Working area	$0.4 \times 0.4 \times 0.4$ m	Allow space for the target, holder and pen motion.
Drawing surface alignment	Camera + fixture reference; set distance from target	Place the target consistently in a fixed location in front of the robot.
Surface drawing area	Predefined, with front 180 degrees reachable only	Avoid collisions and unreachable back surfaces.



Quantitative Design Requirements

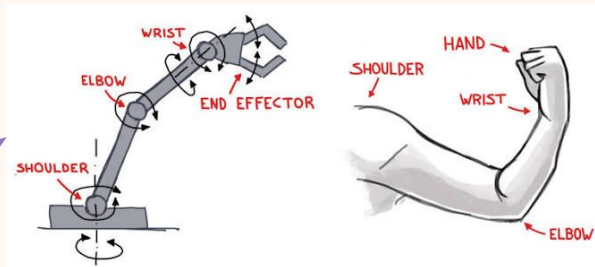
Use-Case Requirements

Target specifications

Working area
(0.4m x 0.4m x 0.4m)

Drawing surface
alignment

Surface drawing area



Design Requirements

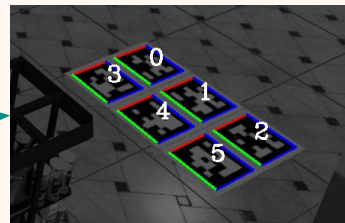
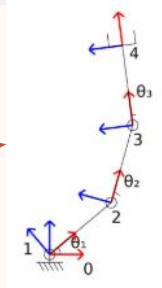
6 Degrees of Freedom (DOF)

Arm Max Reach
Radius (≥ 0.50 m)

Camera Pose Error
(≤ 2 mm)

End-Effector
Accuracy (≤ 2 mm)

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial \theta}{\partial \theta_1} & \frac{\partial \theta}{\partial \theta_2} & \frac{\partial \theta}{\partial \theta_3} \end{bmatrix} = \begin{bmatrix} -s_1 - s_{12} - s_{123} & -s_{12} - s_{123} & -s_{123} \\ c_1 + c_{12} + c_{123} & c_{12} + c_{123} & c_{123} \\ 1 & 1 & 1 \end{bmatrix}$$



Solution Approach

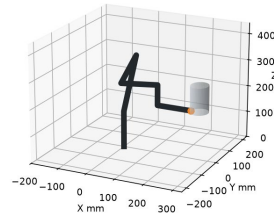
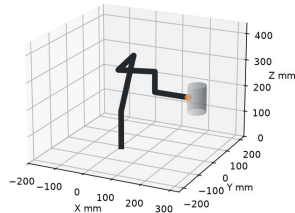
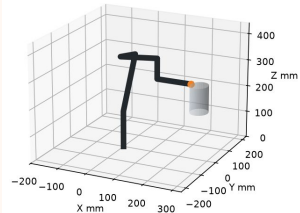
Design Decision	Requirement	Justification
Use a known non-planar surface	Accurately transfer a 2D path onto a curved object	Provides predictable environment for robot & easier programming
Use a 6-DOF arm	Control pen position + direction over curvature	A 3-DoF system moves XYZ, but can't orient pen around varying surface orientations
Restrict drawing to the reachable front region	Complete the drawing without unreachable/colliding poses	Keeps the required workspace compatible with arm's reachable poses

MuJoCo edge-cylinder vertical stroke • fixed base • gravity + pen load

0 s • tip error 0.00 mm

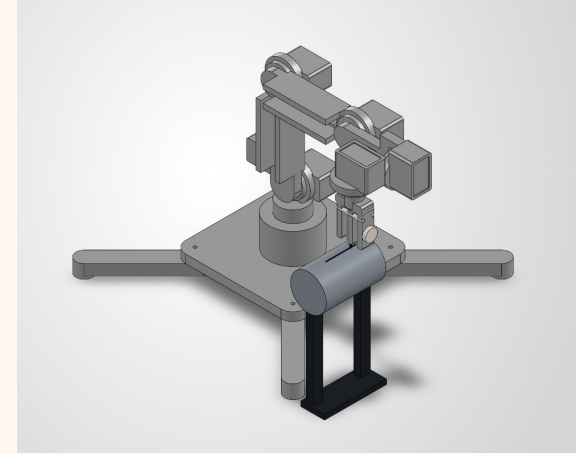
10 s • tip error 0.50 mm

20 s • tip error 0.50 mm



Solution Approach Pt. 2

Design Decision	Requirement	Justification
SVG (Scalable Vector Graphics) input	Accept digital line drawings and convert them into robot trajectories	SVGs are ordered vectors, be easily mapped onto coordinates
Use camera + AprilTags	Align the digital surface model with the physical object	Need repeatable coordinate frames so digital path lines up to physical object
Use a fixed marker rack location	Reliably support multiple colors	Fixed known pickup locations simplify reliable color changes



System Specification



User's Interface

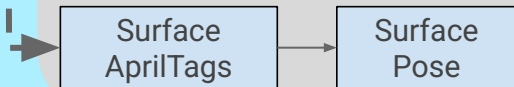
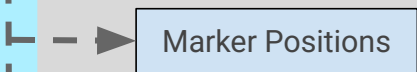


External Computer

Surface Dimensions

Scale and transpose 2d points onto the surface in 3d points

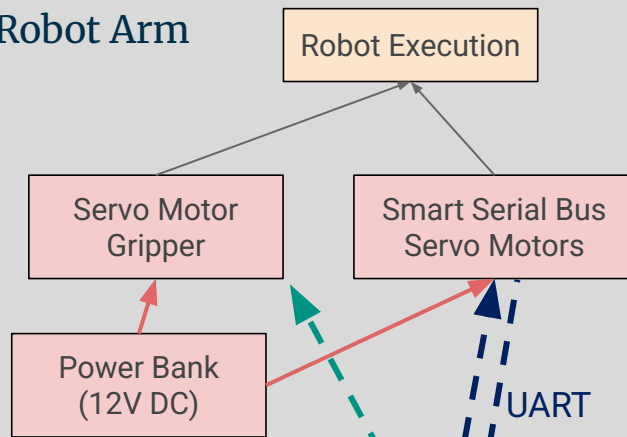
Robot Arm Path Construction



Catmull Rom Splines + Cartesian Path Interpolation

HTTPS GET/POST

Robot Arm



Jetson Orin Nano

PWM

ROS2_control_node: Motor control algorithms

Robot Control System

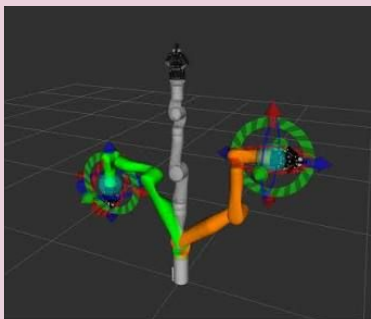
MoveIt 2: Inverse Kinematics + Motion Planning

RVIZ: Robot Simulation and Visualization

Implementation Plan

Inspiration and Tools:
URE3 Robot Arm

RVIZ 2 and MoveIt 2



Parts to Buy / Use:
Smart Serial Bus Servo



Jetson Orin Nano

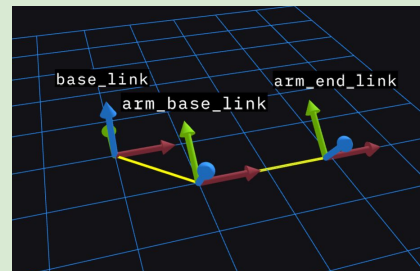


Intel
RealSense
Depth Camera

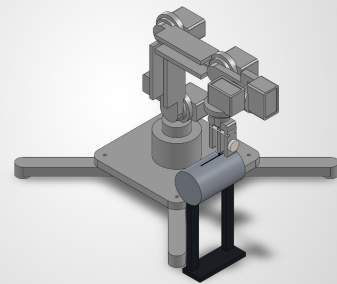
Designing and Assembling:
Designing and 3D Printing
the Full Robot Arm



Custom ROS2
Software Stack



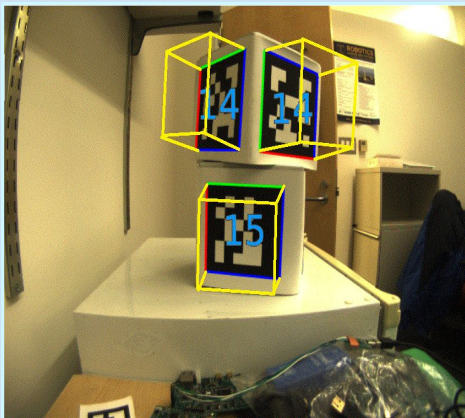
AprilTag Pose
Detection and CV



Test, Verification, and Validation

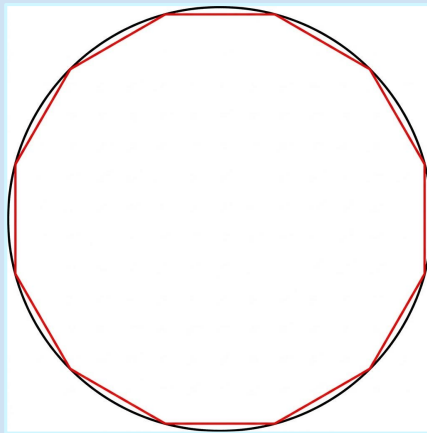
Surface Pose Detection

- AprilTags + OpenCV
- Absolute error ≤ 2 mm



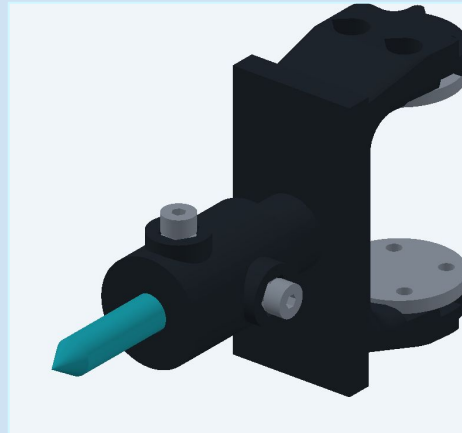
3D Position Accuracy

- Abs. position error ≤ 2 mm
- Drawing accuracy: printed vs drawn pattern ≤ 2 mm



Pen Selection & Pickup

- Test pen holder reliability
- 90% pickup accuracy



Risks and Unknowns

**Surface Pose
Detection Failure**

**Marker Selection &
Pickup Failure**

**3D Position Accuracy
Failure**

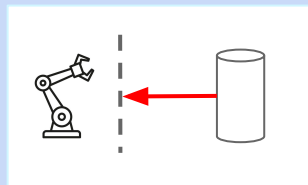
Train YOLOv26
detection model
on AprilTags



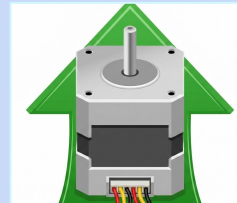
Use depth
functionality on
camera for detection



Fix location of object
to optimal area on
workspace



Upgrade motor
strength/control



Schedule & Demonstration Milestones

1	Leonardo Da Pinchy Project Schedule														
2	September 13–December 5, 2026 Known dimensions; fixed front section first; indexed rotation is a stretch goal.														
3	Task	Owner	Done?	9/13–9/19	9/20–9/26	9/27–10/3	10/4–10/10	10/11–10/17	10/18–10/24	10/25–10/31	11/1–11/7	11/8–11/14	11/15–11/21	11/22–11/28	11/29–12/5
4	Robot Arm Physical Design														
5	Select motor joint types	All													
6	Calculate torque, workspace area, and joint limits	All													
7	Create general parts list (motors, screws, bearings, Jetson nano, wires, power supply, cameras)	All													
8	ITERATION 1: Design mechanical arm in SolidWorks; assign joint physical limits and workspace	Aaron Tang													
9	Create model surfaces (plane, curve, cylinder, half-pipe)	Aaron Tang													
10	Run simulations of arm (SolidWorks?) to outline workspace and verify calculated torques needed	Aaron Tang													
11	3D print ITERATION 1 parts	All													
12	ITERATION 1 assembly	Bryan Hong													
13	Configure power supply, configure microcontroller (if we buy one)	Bryan Hong													
14	Wire up smart bus motors and configure	Bryan Hong													
15	Run physical tests on motors and ITERATION 1 robot assembly	Jonathan Waller													
16	Create framework of ROS2 packages and robot software architecture	Jonathan Waller													
17	Use OpenCV model for tracking april tags	Bryan Hong													
18	ITERATION 2: Update mechanical arm in SolidWorks; assign joint physical limits and workspace	Aaron Tang													
19	Rerun simulations of arm (SolidWorks?) to verify ITERATION 2 fits specs	Aaron Tang													
20	Update assembly of robot to ITERATION 2	Bryan Hong													
21	Create ROS2 link tree for robot arm and create simulation in vis2	Jonathan Waller													
22	Create robot inverse kinematics and path planning to run in simulation	Jonathan Waller													
23	Website creation for file upload, queue dashboard, web whiteboard	Aaron Tang													
24	Create reliable and high fidelity 2d path planner	Aaron Tang													
25	Map 2d paths to 3d space (6DOF per point) on measured model surfaces (surface / projection)	Aaron Tang													
26	Run physical tests on motors and ITERATION 2 robot assembly	All													
27	ITERATION 3: Update mechanical arm in SolidWorks; assign joint physical limits and workspace	Aaron Tang													
28	Rerun simulations of arm (SolidWorks?) to verify ITERATION 3 fits specs	Aaron Tang													
29	Update assembly of robot to ITERATION 3	Bryan Hong													
30	Test tags for locating arm location in 3D space connected to arm coordinate frame	Jonathan Waller													
31	Run physical tests and verification on finalized robot	All													
32	BUFFER SPACE - integration	All													
33	Jonathan Waller: embedded / hardware. Bryan Hong: circuits. Aaron Tang: software / modeling. All: shared work.														
35															