Joint Trajectory Method:

rostopic pub /arm\_controller/command trajectory\_msgs/JointTrajectory '{joint\_names: ["first\_joint\_base","second\_joint\_enclosure\_joint", "second\_joint\_base", “third\_joint\_base”], points: [{positions: [-5.5, 7.5, 1.02, 2, 3], time\_from\_start: [1,0]}]}' -1

JointPoisitonController:

rostopic pub /arm\_controller/command std\_msgs/Float64 "data: -0.707"

rostopic pub /arm\_controller/command  std\_msgs/Float64 1.0

- second\_joint\_enclosure\_joint

- second\_joint\_base

- third\_joint\_base

- grabber\_fixture\_left\_joint

- grabber\_fixture\_right\_joint

type: "position\_controllers/JointGroupPositionController"

joints:

- left\_gripper\_joint

- right\_gripper\_joint

sudo apt-get install ros-noetic-ros-control ros-noetic-joint-state-controller ros-noetic-effort-controllers ros-noetic-position-controllers ros-noetic-velocity-controllers ros-noetic-ros-controllers ros-noetic-gazebo-ros ros-noetic-gazebo-ros-control

def arm(value):

if value == 1:

perform\_trajectory(arm\_controller, arm\_joints, arm\_low\_position)

elif value == 2:

perform\_trajectory(arm\_controller, arm\_joints, arm\_medium\_position)

elif value == 3:

perform\_trajectory(arm\_controller, arm\_joints, arm\_high\_position)

def arm(value):

if value == 1:

perform\_trajectory(arm\_controller, arm\_joints, arm\_low\_position)

elif value == 2:

perform\_trajectory(arm\_controller, arm\_joints, arm\_medium\_position)

elif value == 3:

perform\_trajectory(arm\_controller, arm\_joints, arm\_high\_position)

cv2.putText(frame, "Box Identified", (x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 1.0, (0, 255, 0))

area\_str = str(area)

area\_output = "Area: " + area\_str

cv2.putText(frame, area\_output, (x, y - 50), cv2.FONT\_HERSHEY\_SIMPLEX, 1.0, (0, 255, 0))