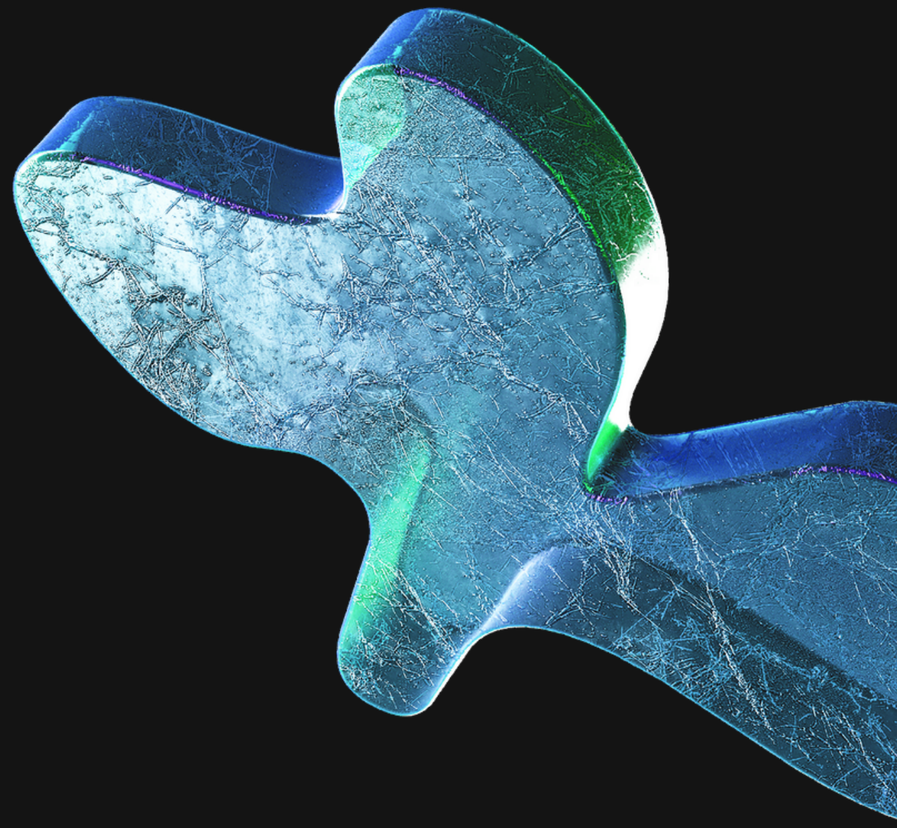
ENPM662-0101:
Introduction to
Robot Modeling-
Fall 2021

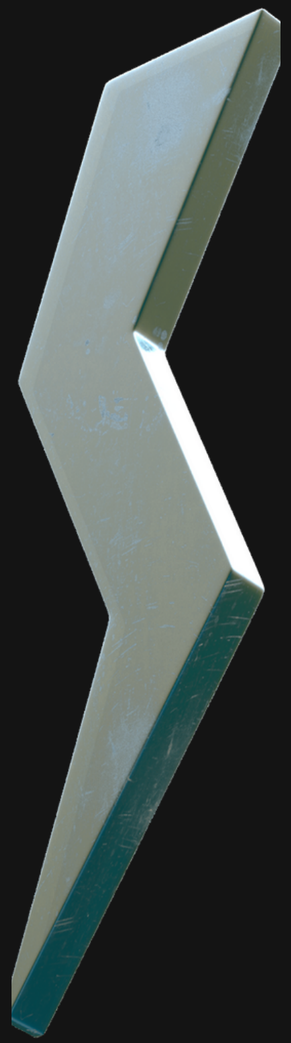


Farming in Space using Baxter

Final Project Presentation

- NEHA SAINI

rethink robotics™



Baxter Intera3 is a research robot.

Processor: 3rd Gen Intel Core i7-3770 Processor (8MB, 3.4GHz) w/HD4000 Graphics

It has two arms with 7 DOF each

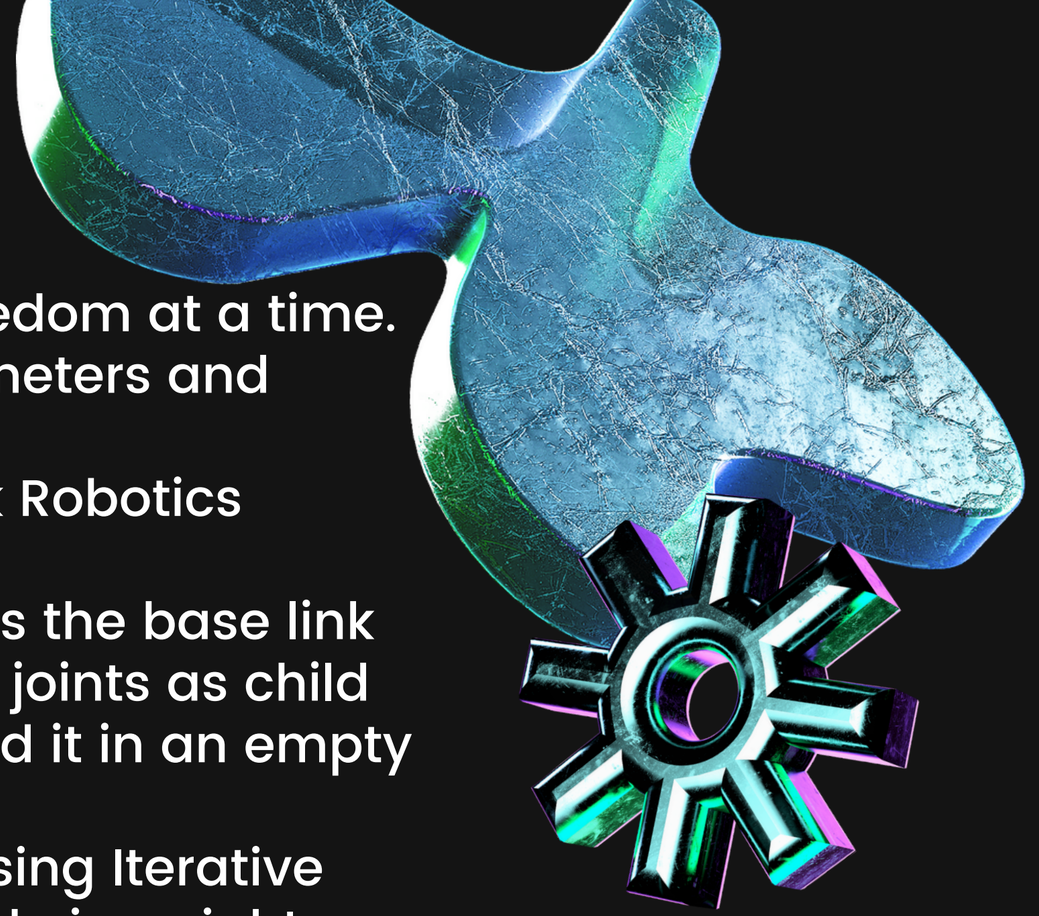
2-Shoulder Joints
2-Elbow Joints
3-Wrist Joints

A humanoid robot robot which is used in manufacturing currently and is widespreadly used for research purposes.

Motivation

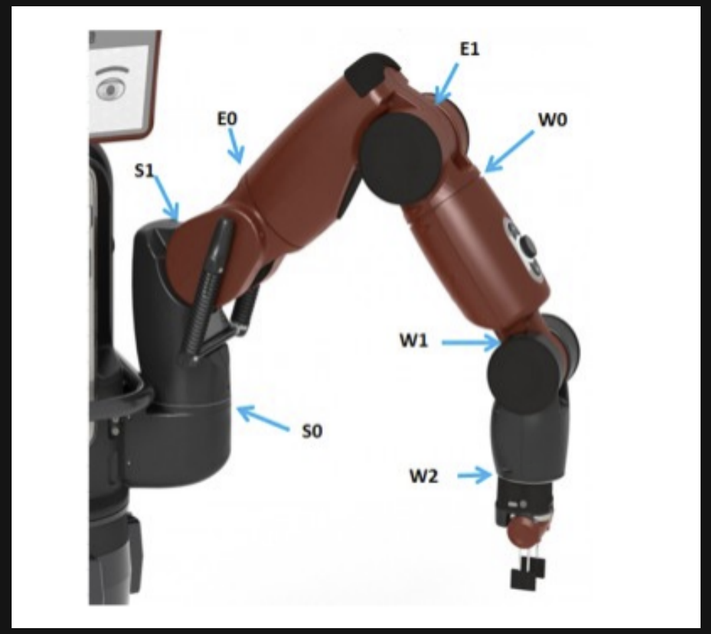
- The astronauts who specifically go into deep space missions for longer duration invest a lot of time in farming due to which they are not able to focus on other significantly important tasks. So in order for a better time management we can employ the robots as space farmers.
- Also in order to send food supplies on a regular basis to ISS, a lot of fuel and time is required with high risk of failure which in turn leads to loss of resources. So creating a habitat in outer space is need of the hour.
- The use of farming robots will not only help in space farming but will also help with farming on earth in the conditions where it is difficult for humans to operate.

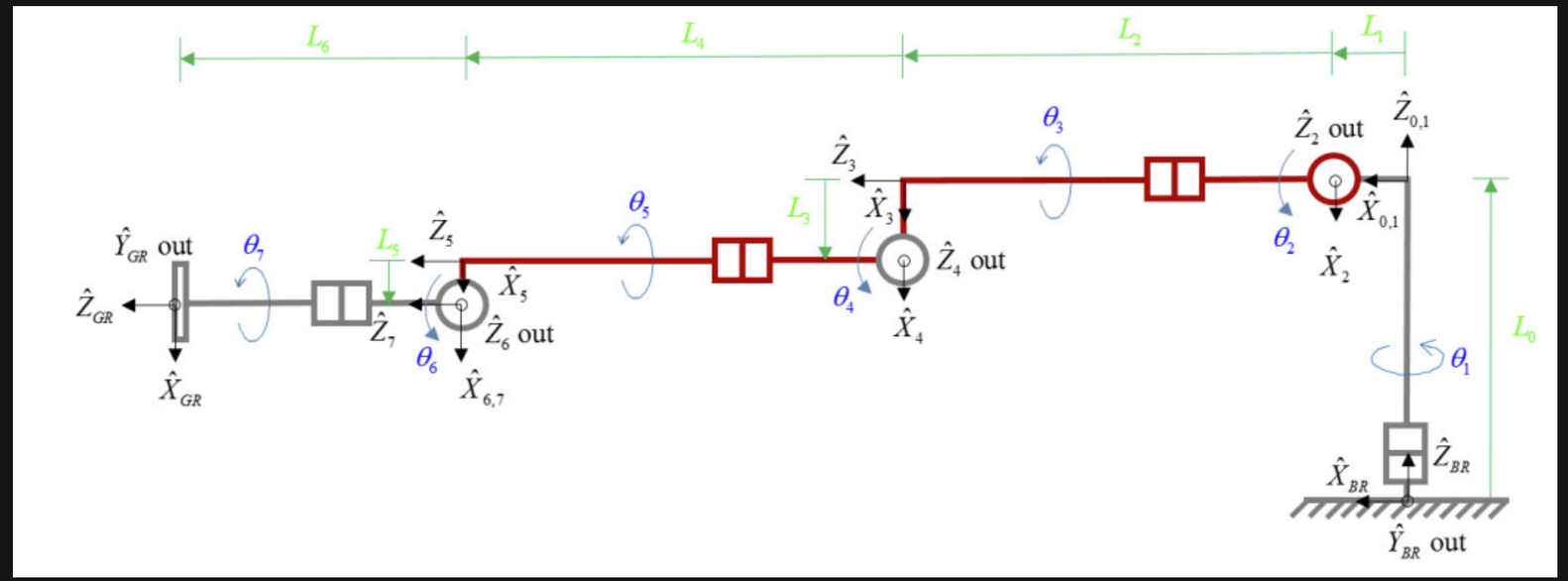
Process Followed

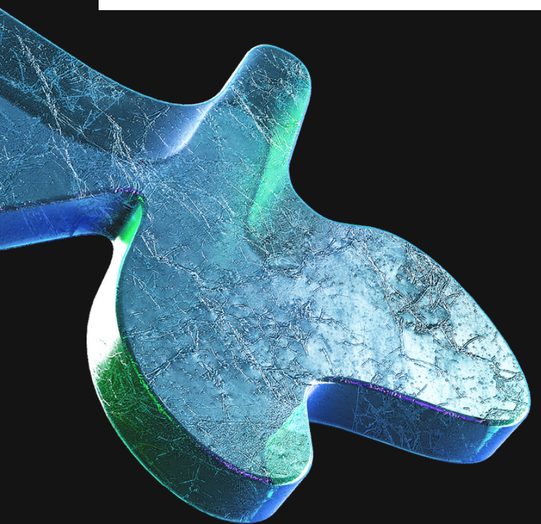


- I considered the right arm utilizing its 6 degree of freedom at a time.
- Then I used Craig's Convention to derive its DH parameters and defining the workspace.
- Using the lengths and joint limits provided by Rethink Robotics I further built a CAD model using solidworks.
- After which I exported the urdf file assigning parent as the base link and the 6 links as child link and further assigning the joints as child links of the 6 links; hence forming a tree and launched it in an empty world in Gazebo.
- Futhere I wrote the forward and inverse kinematics using Iterative Jacobian Method and tested the validation of the code in a right angle configuration in python(Jupyter).
- I tried to interface the joints with position based interfacing.
- Using the joint controllers for creting the .yaml file and storing values.
- After creating the launch file I ran it in gazebo and rviz.
- Finally I tried to complete one action of water the plants utilizing the teleop.

Frame Assignment for 7 DOF Right Arm

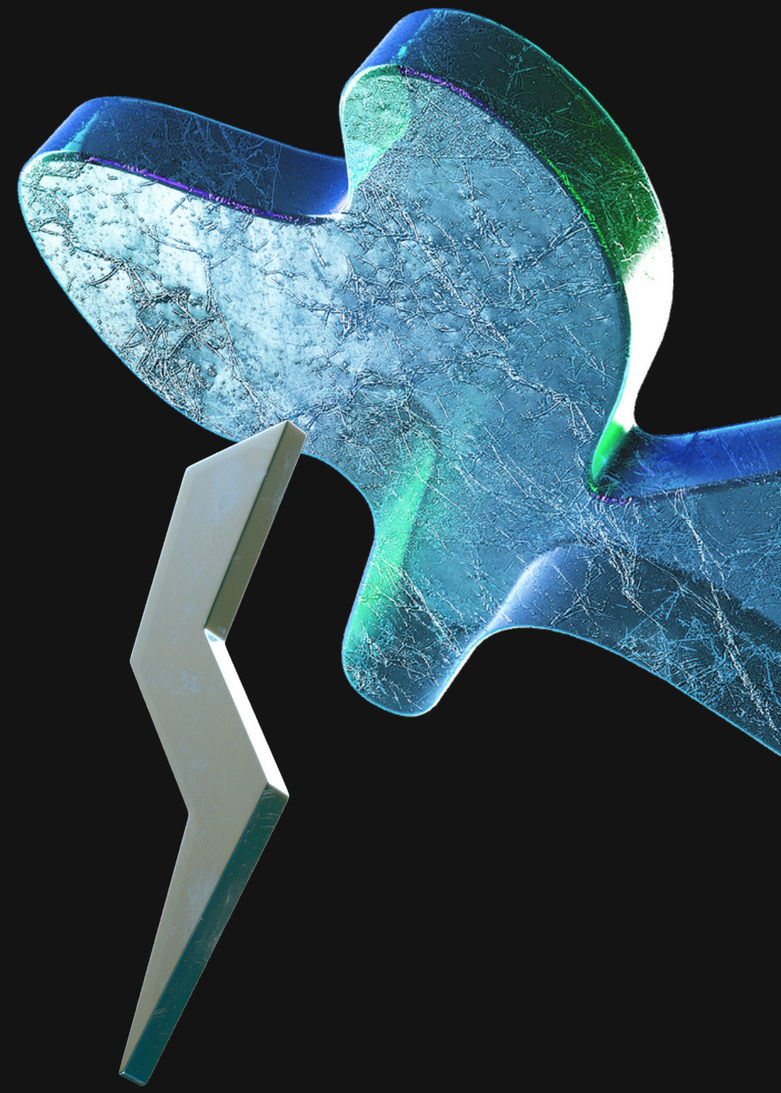






DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	L_1	0	$\theta_2 + 90^\circ$
3	90°	0	L_2	θ_3
4	-90°	L_3	0	θ_4
5	90°	0	L_4	θ_5
6	-90°	L_5	0	θ_6
7	90°	0	0	θ_7



Forward Kinematics Snap

TRANSFORMATION MATRICES

```
In [78]: print(T01)
         print(T12)
         print(T23)
         print(T34)
         print(T45)
         print(T56)

Matrix([[cos(theta1), -sin(theta1), 0, 0], [sin(theta1), cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])
Matrix([[cos(theta2), -sin(theta2), 0, A1], [0, 0, 1, 0], [-sin(theta2), -cos(theta2), 0, 0], [0, 0, 0, 1]])
Matrix([[cos(theta3), -sin(theta3), 0, 0], [0, 0, -1, -D3], [sin(theta3), cos(theta3), 0, 0], [0, 0, 0, 1]])
Matrix([[cos(theta4), -sin(theta4), 0, A3], [0, 0, 1, 0], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]])
Matrix([[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, -D5], [sin(theta5), cos(theta5), 0, 0], [0, 0, 0, 1]])
Matrix([[cos(theta6), -sin(theta6), 0, A5], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0, 0, 0, 1]])

In [79]: T02 = T01 * T12
         T03 = T01 * T12 * T23
         T04 = T01 * T12 * T23 * T34
         T05 = T01 * T12 * T23 * T34 * T45
         T06 = T01 * T12 * T23 * T34 * T45 * T56
         print(sympify(T02))
         print(sympify(T03))
         print(sympify(T04))
         print(sympify(T05))
         print(sympify(T06))

Matrix([[cos(theta1)*cos(theta2), -sin(theta2)*cos(theta1), -sin(theta1), A1*cos(theta1)], [sin(theta1)*cos(theta2),
-sin(theta1)*sin(theta2), cos(theta1), A1*sin(theta1)], [-sin(theta2), -cos(theta2), 0, 0], [0, 0, 0, 1]])
```

TRANSLATION JACOBIAN

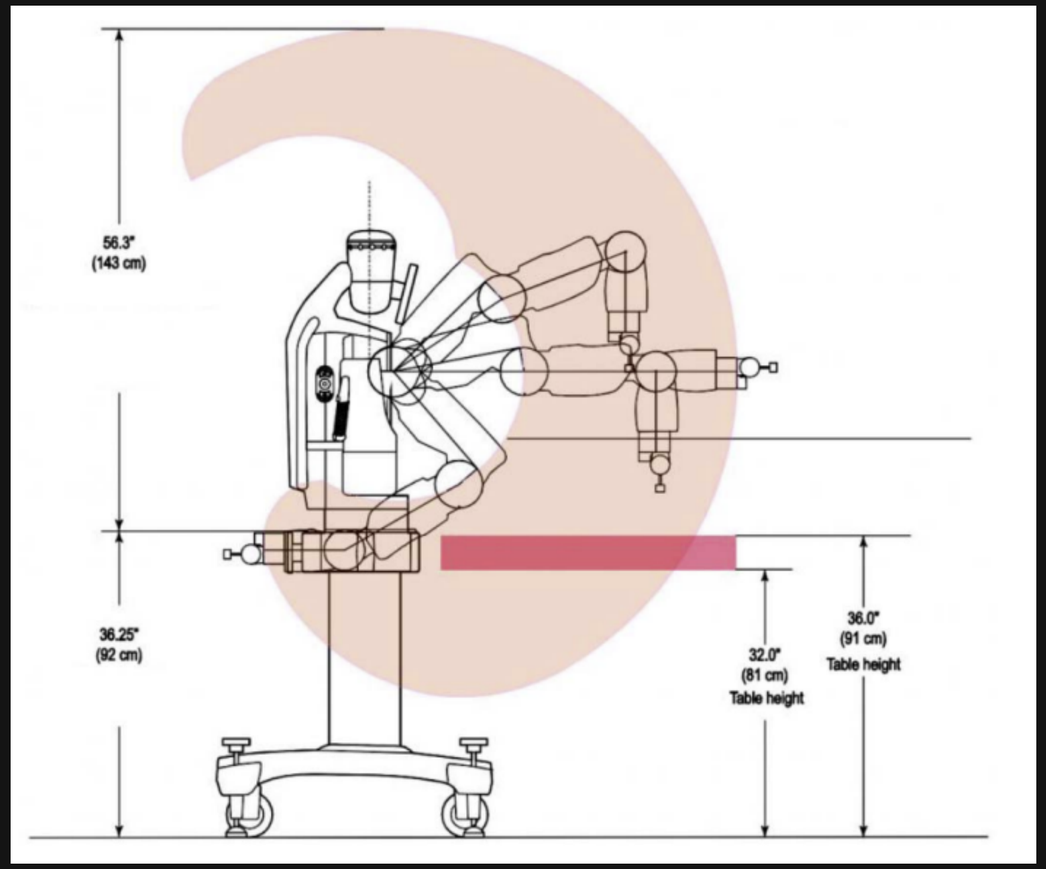
```
J_rot = R01.col_insert(1,R02).col_insert(1,R03).col_insert(1,R04).col_insert(1,R05).col_insert(1,R06)
print(J_rot)

Matrix([[0, -((-sin(theta1)*sin(theta3) + cos(theta1)*cos(theta2)*cos(theta3))*cos(theta4) - sin(theta2)*sin(theta4)
*cos(theta1))*sin(theta5) + (-sin(theta1)*cos(theta3) - sin(theta3)*cos(theta1)*cos(theta2))*cos(theta5), (-sin(thet
a1)*sin(theta3) + cos(theta1)*cos(theta2)*cos(theta3))*sin(theta4) + sin(theta2)*cos(theta1)*cos(theta4), -sin(theta
1)*cos(theta3) - sin(theta3)*cos(theta1)*cos(theta2), sin(theta2)*cos(theta1), -sin(theta1)], [0, -((sin(theta1)*cos
(theta2)*cos(theta3) + sin(theta3)*cos(theta1))*cos(theta4) - sin(theta1)*sin(theta2)*sin(theta4))*sin(theta5) + (-s
in(theta1)*sin(theta3)*cos(theta2) + cos(theta1)*cos(theta3))*cos(theta5), (sin(theta1)*cos(theta2)*cos(theta3) + si
n(theta3)*cos(theta1))*sin(theta4) + sin(theta1)*sin(theta2)*cos(theta4), -sin(theta1)*sin(theta3)*cos(theta2) + cos
(theta1)*cos(theta3), sin(theta1)*sin(theta2), cos(theta1)], [1, -(-sin(theta2)*cos(theta3)*cos(theta4) - sin(theta
4)*cos(theta2))*sin(theta5) + sin(theta2)*sin(theta3)*cos(theta5), -sin(theta2)*sin(theta4)*cos(theta3) + cos(theta
2)*cos(theta4), sin(theta2)*sin(theta3), cos(theta2), 0]])
```

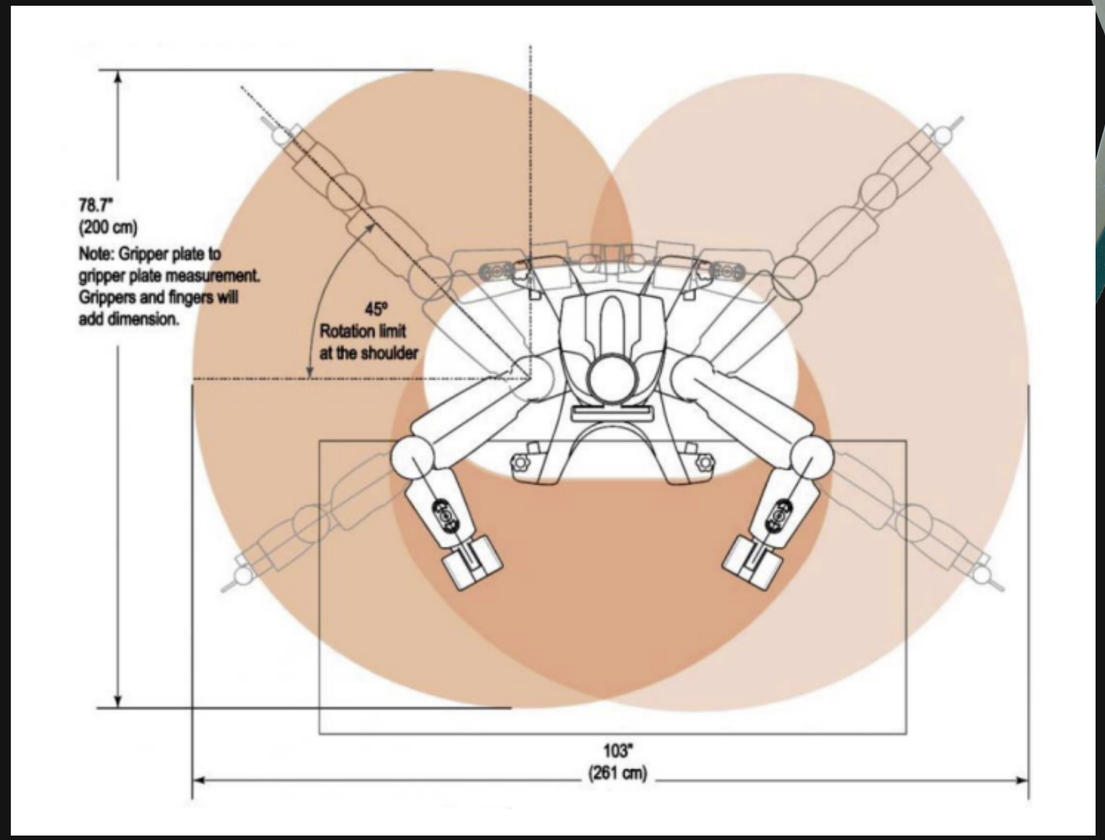
ROTATION JACOBIAN

```
J_trans = J1.col_insert(1,J2).col_insert(1,J3).col_insert(1,J4).col_insert(1,J5).col_insert(1,J6)
print(J_trans)
```

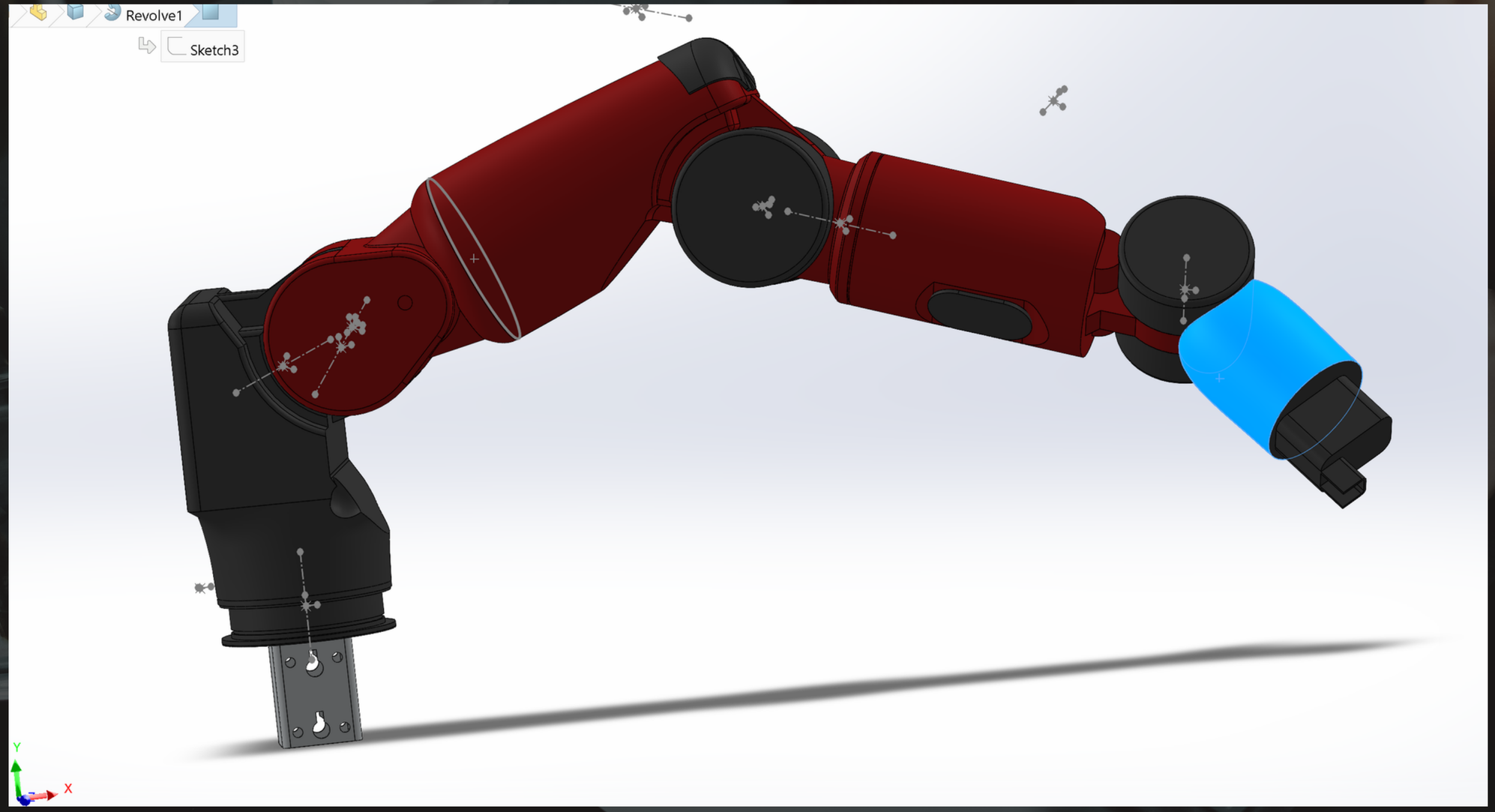
```
Matrix([[-A1*sin(theta1) + A3*(-sin(theta1)*cos(theta2)*cos(theta3) - sin(theta3)*cos(theta1)) + A5*(((-sin(theta1)*
cos(theta2)*cos(theta3) - sin(theta3)*cos(theta1))*cos(theta4) + sin(theta1)*sin(theta2)*sin(theta4))*cos(theta5) +
(sin(theta1)*sin(theta3)*cos(theta2) - cos(theta1)*cos(theta3))*sin(theta5)) - D3*sin(theta1)*sin(theta2) - D5*((sin
(theta1)*cos(theta2)*cos(theta3) + sin(theta3)*cos(theta1))*sin(theta4) + sin(theta1)*sin(theta2)*cos(theta4)), 0, A
5*(-((-sin(theta1)*sin(theta3) + cos(theta1)*cos(theta2)*cos(theta3))*cos(theta4) - sin(theta2)*sin(theta4)*cos(thet
a1))*sin(theta5) + (-sin(theta1)*cos(theta3) - sin(theta3)*cos(theta1)*cos(theta2))*cos(theta5)), A5*(-(-sin(theta1)
*sin(theta3) + cos(theta1)*cos(theta2)*cos(theta3))*sin(theta4) - sin(theta2)*cos(theta1)*cos(theta4))*cos(theta5) -
D5*((sin(theta1)*sin(theta3) - cos(theta1)*cos(theta2)*cos(theta3))*cos(theta4) + sin(theta2)*sin(theta4)*cos(theta
1)), A3*(-sin(theta1)*cos(theta3) - sin(theta3)*cos(theta1)*cos(theta2)) + A5*((sin(theta1)*sin(theta3) - cos(theta
1)*cos(theta2)*cos(theta3))*sin(theta5) + (-sin(theta1)*cos(theta3) - sin(theta3)*cos(theta1)*cos(theta2))*cos(theta
4)*cos(theta5)) - D5*(sin(theta1)*cos(theta3) + sin(theta3)*cos(theta1)*cos(theta2))*sin(theta4), -A3*sin(theta2)*co
s(theta1)*cos(theta3) + A5*((-sin(theta2)*cos(theta1)*cos(theta3)*cos(theta4) - sin(theta4)*cos(theta1)*cos(theta2))
*cos(theta5) + sin(theta2)*sin(theta3)*sin(theta5)*cos(theta1)) + D3*cos(theta1)*cos(theta2) - D5*(sin(theta2)*sin(t
heta4)*cos(theta1)*cos(theta3) - cos(theta1)*cos(theta2)*cos(theta4))], [A1*cos(theta1) + A3*(-sin(theta1)*sin(theta
3) + cos(theta1)*cos(theta2)*cos(theta3)) + A5*(((-sin(theta1)*sin(theta3) + cos(theta1)*cos(theta2)*cos(theta3))*co
s(theta4) - sin(theta2)*sin(theta4)*cos(theta1))*cos(theta5) + (-sin(theta1)*cos(theta3) - sin(theta3)*cos(theta1)*c
os(theta2))*sin(theta5)) + D3*sin(theta2)*cos(theta1) - D5*((sin(theta1)*sin(theta3) - cos(theta1)*cos(theta2)*cos(t
heta3))*sin(theta4) - sin(theta2)*cos(theta1)*cos(theta4)), 0, A5*(-((sin(theta1)*cos(theta2)*cos(theta3) + sin(thet
a3)*cos(theta1))*cos(theta4) - sin(theta1)*sin(theta2)*sin(theta4))*sin(theta5) + (-sin(theta1)*sin(theta3)*cos(thet
a2) + cos(theta1)*cos(theta3))*cos(theta5)), A5*(-(sin(theta1)*cos(theta2)*cos(theta3) + sin(theta3)*cos(theta1))*si
n(theta4) - sin(theta1)*sin(theta2)*cos(theta4))*cos(theta5) - D5*((-sin(theta1)*cos(theta2)*cos(theta3) - sin(theta
3)*cos(theta1))*cos(theta4) + sin(theta1)*sin(theta2)*sin(theta4)), A3*(-sin(theta1)*sin(theta3)*cos(theta2) + cos(t
heta1)*cos(theta3)) + A5*((-sin(theta1)*sin(theta3)*cos(theta2) + cos(theta1)*cos(theta3))*cos(theta4)*cos(theta5) +
(-sin(theta1)*cos(theta2)*cos(theta3) - sin(theta3)*cos(theta1))*sin(theta5)) - D5*(sin(theta1)*sin(theta3)*cos(thet
a2) - cos(theta1)*cos(theta3))*sin(theta4), -A3*sin(theta1)*sin(theta2)*cos(theta3) + A5*((-sin(theta1)*sin(theta2)*
cos(theta3)*cos(theta4) - sin(theta1)*sin(theta4)*cos(theta2))*cos(theta5) + sin(theta1)*sin(theta2)*sin(theta3)*sin
(theta5)) + D3*sin(theta1)*cos(theta2) - D5*(sin(theta1)*sin(theta2)*sin(theta4)*cos(theta3) - sin(theta1)*cos(theta
2)*cos(theta4))], [0, 0, A5*(-(-sin(theta2)*cos(theta3)*cos(theta4) - sin(theta4)*cos(theta2))*sin(theta5) + sin(the
ta2)*sin(theta3)*cos(theta5)), A5*(sin(theta2)*sin(theta4)*cos(theta3) - cos(theta2)*cos(theta4))*cos(theta5) - D5*
(sin(theta2)*cos(theta3)*cos(theta4) + sin(theta4)*cos(theta2)), A3*sin(theta2)*sin(theta3) + A5*(sin(theta2)*sin(th
eta3)*cos(theta4)*cos(theta5) + sin(theta2)*sin(theta5)*cos(theta3)) + D5*sin(theta2)*sin(theta3)*sin(theta4), -A3*c
os(theta2)*cos(theta3) + A5*((sin(theta2)*sin(theta4) - cos(theta2)*cos(theta3)*cos(theta4))*cos(theta5) + sin(theta
3)*sin(theta5)*cos(theta2)) - D3*sin(theta2) - D5*(sin(theta2)*cos(theta4) + sin(theta4)*cos(theta2)*cos(theta3))]])
```

WOKRSPACE ANALYSIS





CAD MODEL OF ARM



THANK YOU

QUESTION?