<u>Collaborative Robotics-Final Project</u>

University at Buffalo, The State University of New York
Pavithrun Dhanasekaran - 50308610
Nikhil Arora - 50320282
Rammya Iyer - 50320295
Makarand Mandolkar - 50321880
Vivek Lekkala - 50322187

INTRODUCTION - FANUC CR-4iA

It is the smallest of the collaborative robot range, with six axis arm, and a maximum payload of 4kg. Similar to its fellow collaborative robots, it can handle lightweight tasks that are tedious, highly manual. Its compact nature enables it to perform smaller jobs in areas with limited space requirements. It is possible for it to be wall- or invert-mounted, offering a wider range of motion without interfering with the operator's workspace.

Fig 1. Real life model

Fig 2. Our SolidWorks model.

1. D.H. analysis and Transformation matrices.

DH Parameters-

The kinematic analysis of an n-link manipulator can be extremely complex and the conventions allow us to simplify the analysis significantly. And also, they enable robot engineers to communicate in a single common language. A commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg, or DH convention. In this convention, where the four quantities θ_i, a_i, d_i, α_i are parameters associated with link i and joint i. The four parameters a_i, α_i, d_i, and are generally given the names link length, link twist, link offset, and joint angle, respectively.

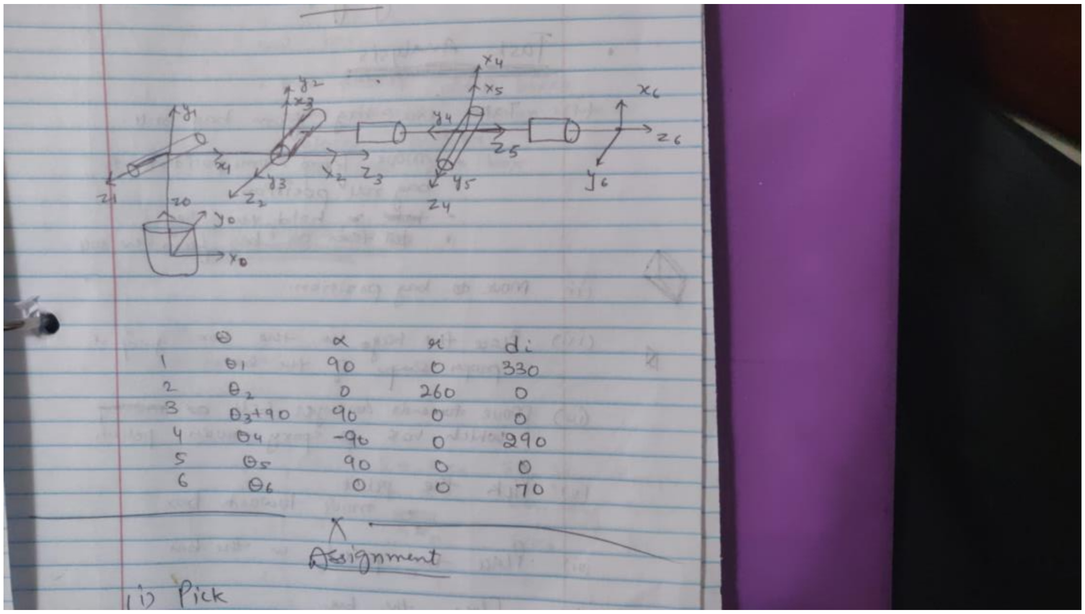

Fig. 3

```
DH =

[        theta1,  90,   0, 330]
[        theta2,   0, 260,   0]
[ theta3 + 90,  90,   0,   0]
[        theta4, -90,   0, 290]
[        theta5,  90,   0,   0]
[        theta6,   0,   0,  70]

>>
```

Fig. 4

Homogenous Matrices:

An arbitrary homogeneous transformation matrix can be characterized by six numbers, three numbers to specify the fourth column of the matrix and three Euler angles to specify the upper left 3 × 3 rotation matrix.

2. Jacobian matrices that relate the C.G of each link.

Jacobian Matrix:

The 'Jacobian Matrix' helps us to relate joint velocities to end-effector velocities. The Jacobian is a matrix that can be thought of as the vector version of the ordinary derivative of a scalar function. The Jacobian is one of the most important quantities in the analysis and control of robot motion. It is helpful in the derivation of the dynamic equations of motion, and in the transformation of forces and torques from the end-effector to the manipulator joints.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

Fig.5

For an n-link manipulator we first derive the Jacobian representing the instantaneous transformation between the n-vector of joint velocities and the 6-vector consisting of the linear and angular velocities of the end-effector. This Jacobian is then a 6 × n matrix. The same approach is used to determine the transformation between the joint velocities and the linear and angular velocity of any point on the manipulator.

Inertia matrix- To calculate the kinetic energy and potential energy of the system, we have to get the moment of inertia of every link from center of gravity. Matrix we get as output is called inertia matrix.

3. Differential equations of motion using Lagrange method

Euler-Lagrange equations allow us to describe the evolution of a mechanical system subject to holonomic constraints. To motivate the Euler-Lagrange approach we begin with a simple derivation of these equations from Newton's Second Law for a one-degree-of-freedom system. We then derive the Euler-Lagrange equations from the principle of virtual work in the general case. In order to determine the Euler-

Lagrange equations in a specific situation, one has to form the Lagrangian of the system, which is the difference between the kinetic energy and the potential energy.

Lagrange = KE – PE

After getting the lagrange equations of motion we can get the differential equation of motion by just putting the values in following equation:

D(q)q'' – C(qq')q' + g(q') = Torque

Where q is a vector, D is mass-inertia matrix, C is Coriall's and centrifugal terms, G is potential energy due to gravity matrix and torque is a torque due to input of forces.

CONCLUSION

Thus, we were successfully able to perform the given engineering analysis tasks. We used SolidWorks to model the robot. MATLAB was the software package we used for our analysis tasks. You can get all the matrices and outputs by running the MATLAB code(For making the code bet ter I have put ; at the end of every function, removing that will print the required matrix.).

REFERENCE

1.http://www.robogrok.com/2-1-3_Jacobian_Matrix.php

2.Robot Modeling and Control (First Edition) Mark W. Spong, Seth Hutchinson, and M. Vidyasagar