**Project 1:**

Joint Variables:

• Joint Position A = [-106.12, -162.47, 135.49, 205.71, 15.19, 0.01]

• Joint Position B = [ -45.48, -183.35,120.62, 238.42, -69.27, 0.01]

**Part A:**

Do you encounter any problems? Do you have any initial hypotheses why some motions are

possible, and some are not? Write about what could have impacted what you observed.

|  |
| --- |
| % % TCP Host and Port settings % host = '127.0.0.1'; % THIS IP ADDRESS MUST BE USED FOR THE VIRTUAL BOX VM % host = '192.168.230.128'; % THIS IP ADDRESS MUST BE USED FOR THE VMWARE clear all; host = '192.168.0.100'; % THIS IP ADDRESS MUST BE USED FOR THE REAL ROBOT port = 30003; % Calling the constructor of rtde to setup tcp connection rtde = rtde(host,port);  % Setting home home = [-588.53, -133.30, 371.91, 2.2214, -2.2214, 0.00];  jointPosA = deg2rad([106.12, -162.47, 135.49, 205.71, 15.19, 0.01]); jointPosB = deg2rad([45.48, -183.35,120.62, 238.42, -69.27, 0.01]);  disp("Part A: ") rtde.movel(home);pause(4) disp("Move second") % rtde.movel(jointPosA, "joint");pause(4); disp("Move third") rtde.movel(jointPosB, "joint");pause(4); rtde.drawPath(tcp\_pose);  rtde.close; |

A screenshot of a computer

Description automatically generated

**Part B:**

Part 1:

Joint Position A = [-106.12, -162.47, 135.49, 205.71, 15.19, 0.01]

We are interested in the all the joints, which has a position in degrees. We convert these joints to radians to get:

The DH parameters for the joints are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| = 0 | = -0.425 | = -0.3922 | = 0 | = 0 | = 0 |
|  | = 0 | = 0 | = 0.1333 | = 0.0997 | = 0.0996 |
| = | = | = | = | = | = |

The general form of the transformation matrix using Denavit-Hartenberg parameters is:

Substituting the parameters for all the joints into this formula, we get:

After substitution and simplification of this, we get:

represents the transformation from the base frame to the end effector. The position of the end effector in the base frame can be found in the top-right column of this matrix. The orientation of the end effector can be found in the top-left 3x3 submatrix.

The position of the end effector in the base frame can be found in the top-right column of the transformation matrix and the orientation of the end effector can be found in the top-left 3x3 submatrix of the transformation matrix.

|  |
| --- |
| %% Part 2 =================================================================  % Import the Robotics Toolbox  clear all  startup\_rvc;  % Define the DH parameters  % [ theta d a alpha]  L(1) = Link([ 0 0.1625 0 pi/2], 'standard');  L(2) = Link([ 0 0 -0.425 0], 'standard');  L(3) = Link([ 0 0 -0.3922 0], 'standard');  L(4) = Link([ 0 0.1333 0 pi/2], 'standard');  L(5) = Link([ 0 0.0997 0 -pi/2], 'standard');  L(6) = Link([ 0 0.0996 0 0], 'standard');  % Create the robot  robot = SerialLink(L, 'name', 'UR5e');  % Display the robot model  robot.display(); |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| Part 3 =================================================================  % Define the joint angles for Joint Variable A  % Convert the joint angles from degrees to radians  jointAnglesA = deg2rad([-106.12, -162.47, 135.49, 205.71, 15.19, 0.01]);  % Compute the forward kinematics  F = robot.fkine(jointAnglesA);  % Compare the matrices  isEqual = isequal(T, F);  % Display the result  disp(isEqual); |