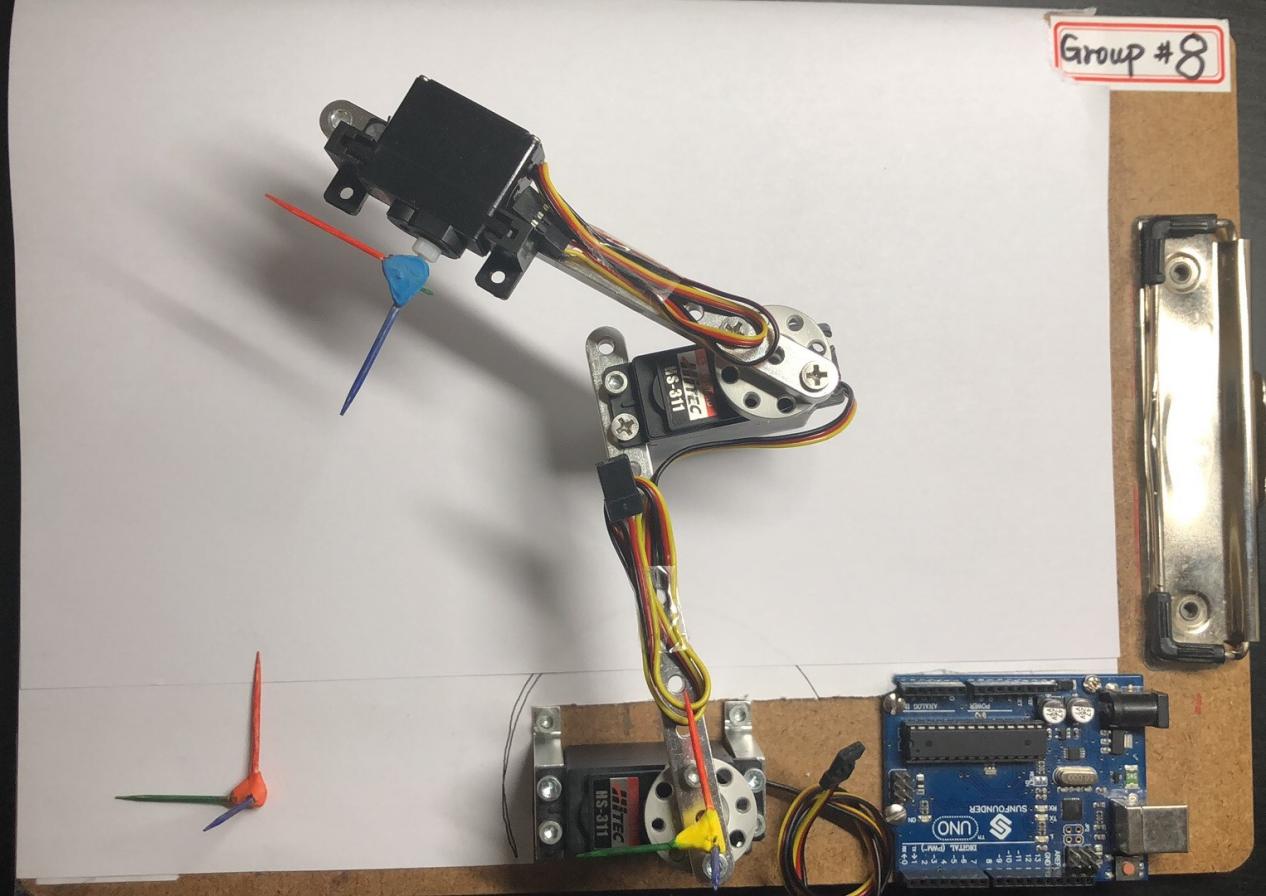
Assignment1-DH parameters and optimization

1. Set up the robot as below figure showing



X

Z

Y

X

Y

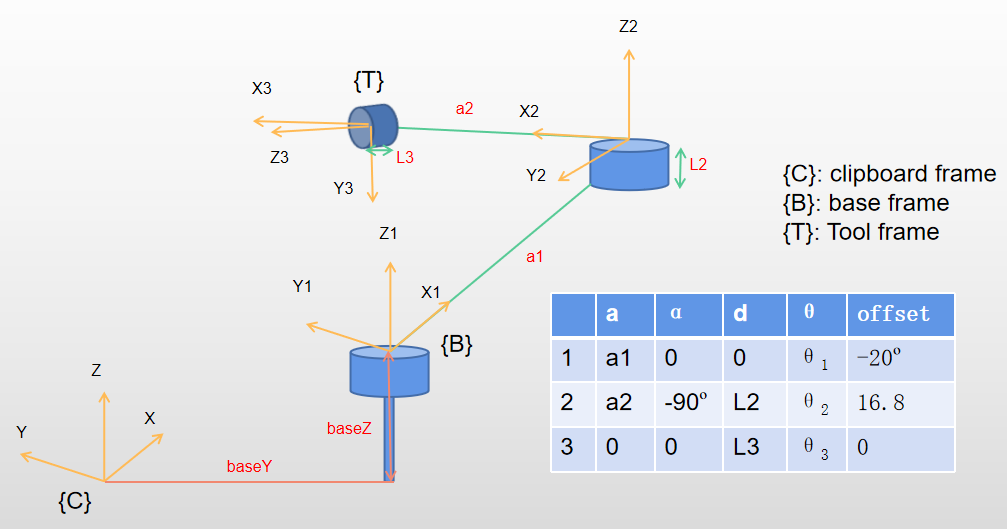
Z

Y

X

Z

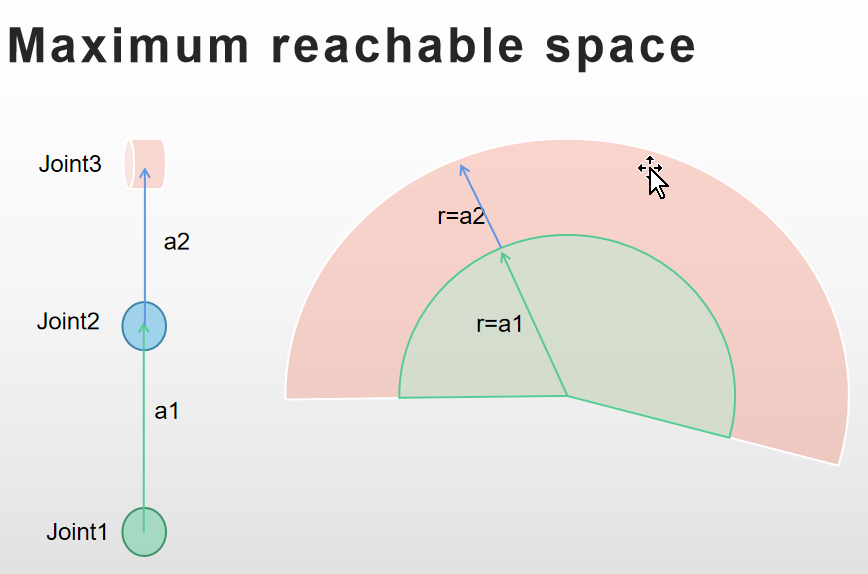
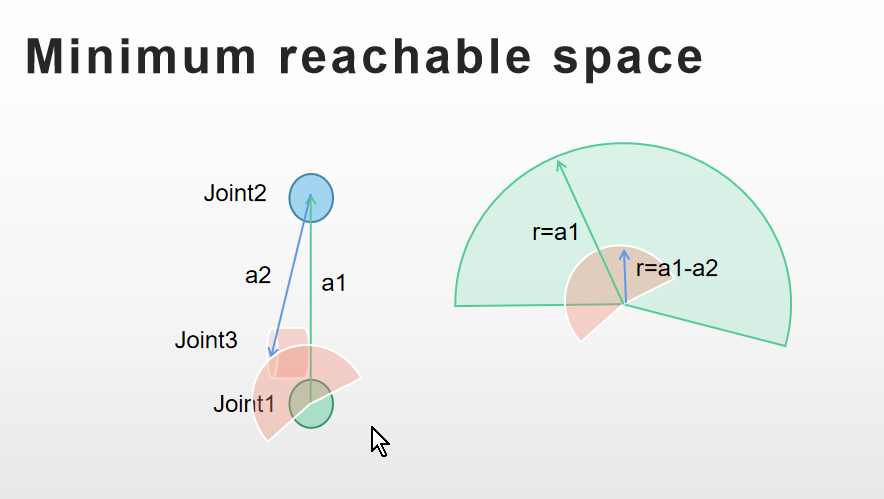
Schematic

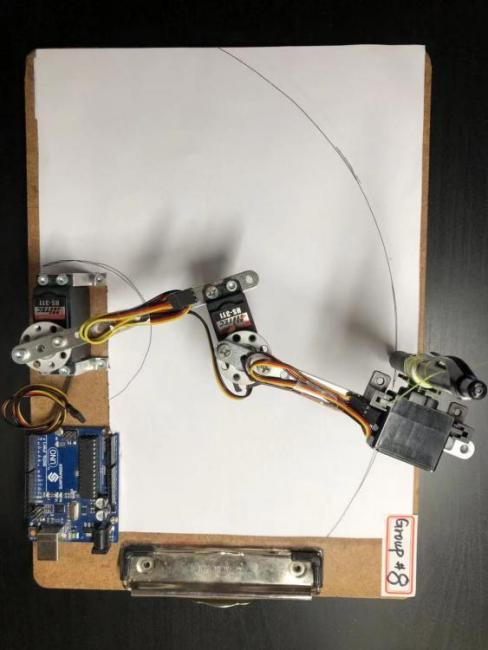
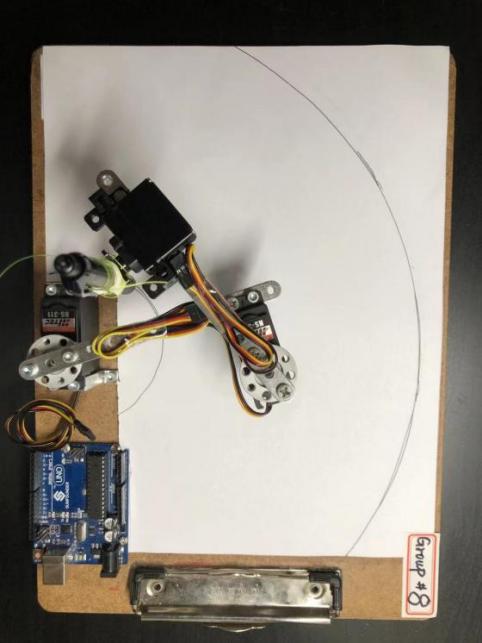


1. Explain how the joints and robot base were setup to maximize the range

Consider on XY plane and

Maximum range is when Z1 Z2 and Y3 are in the same plane

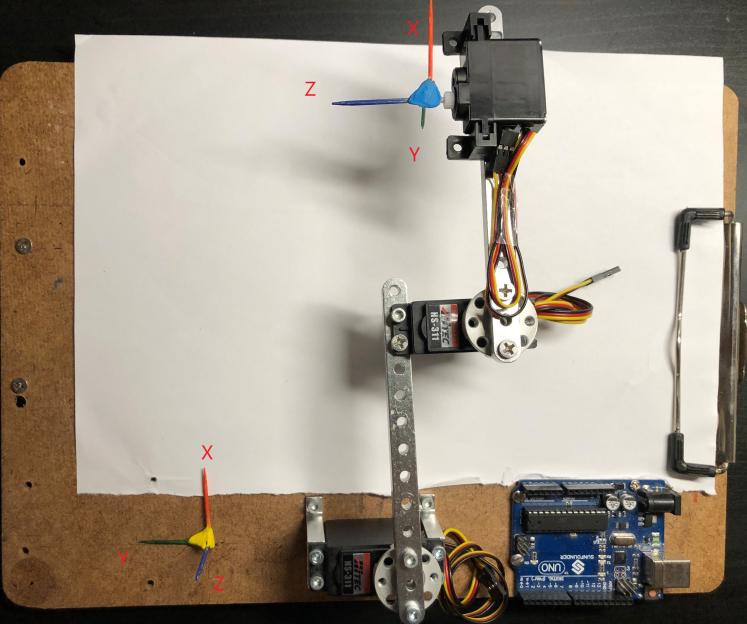
 

Max range

Min range

1. Using toothpicks or stir straws to place a coordinate system at the end-effector and one at the clipboard somewhere.

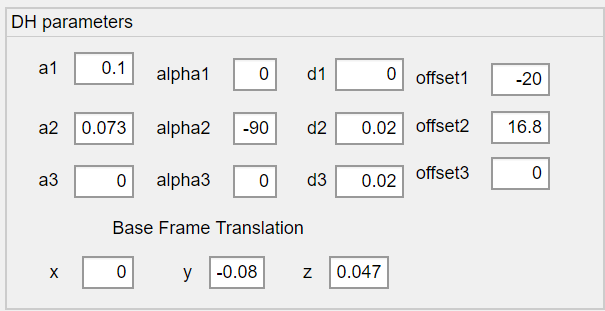


1. Measure the DH parameters and build the graphical model of the robot in the Robotics toolkit.

Measured DH parameters:

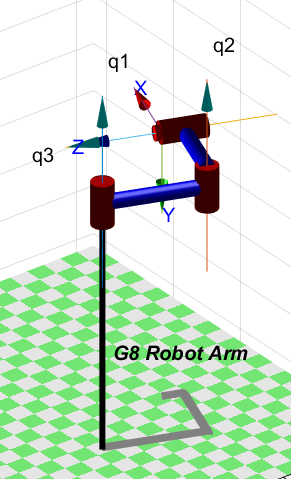
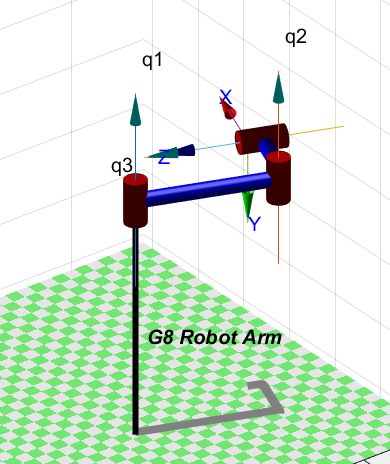
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | alpha | d | offset | QlimMin | QlimMax |
| Joint1 | 0.1 | 0 | 0.047 | -19 | -90 | 120 |
| Joint2 | 0.073 | -90 | 0.02 | 16.8 | -120 | 90 |
| Joint3 | 0 | 0 | 0.02 | 0 | 0 | 135 |

* 1. Create a section of the GUI to input the DH parameters of robot.



* 1. Explain with pictures all the DH-parameters used.

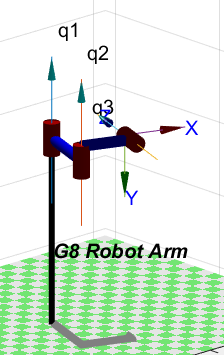
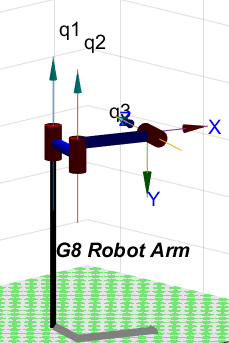
a1=0.1 a1=0.2

a1=0.2

a1=0.1

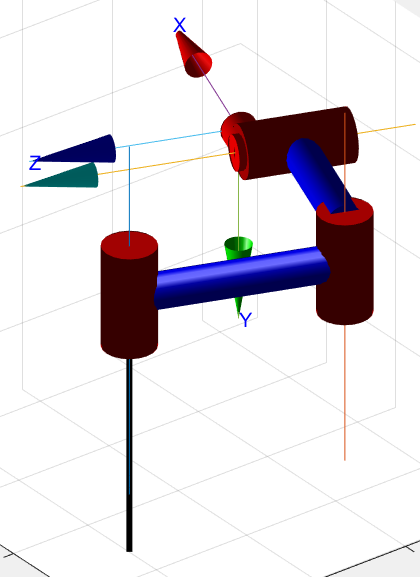
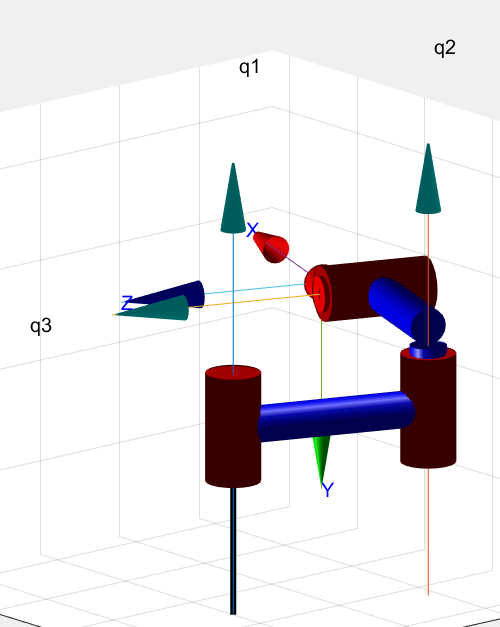
a2 = 0.073 a2 = 0.14

a2=0.14

a2=0.073

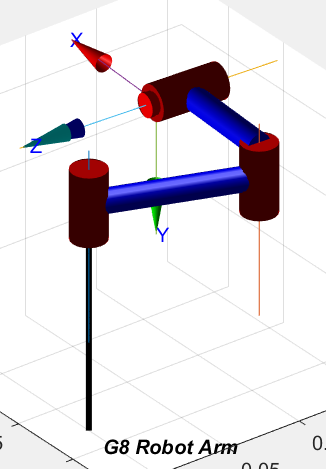
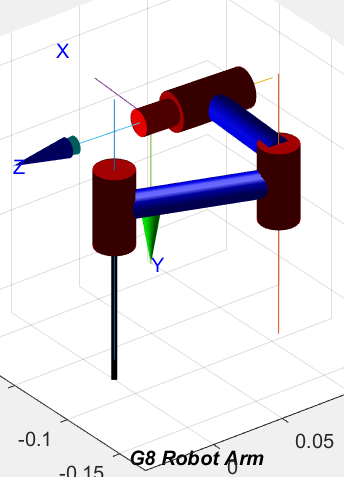
d2 = 0.02 d2 = 0.04

d2=0.02

d2=0.04

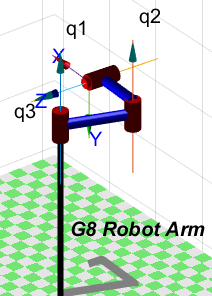
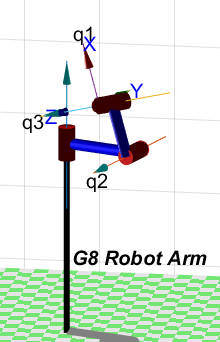
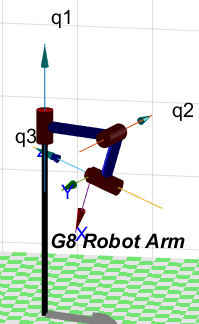
d3 = 0.2 d3 = 0.4

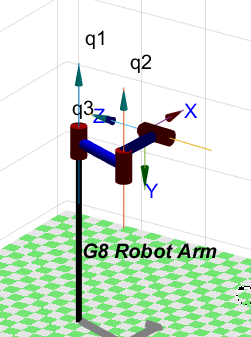
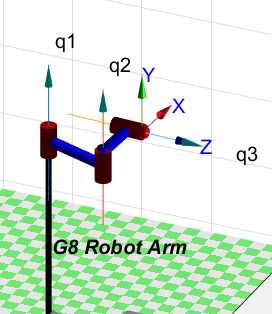
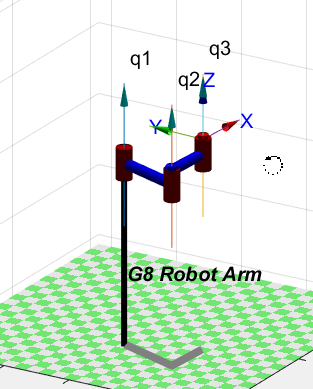
d3 = 0.4

d3 = 0.2

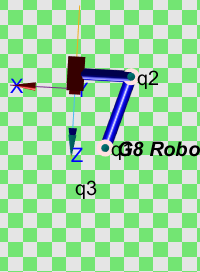
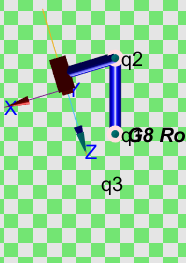
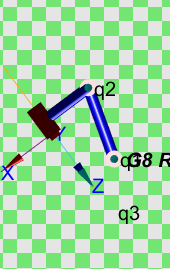
alpha1 = 0 alpha1 = 90 alpha21 = -90

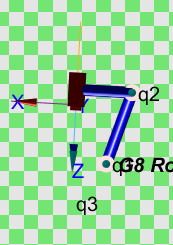
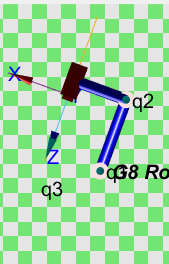
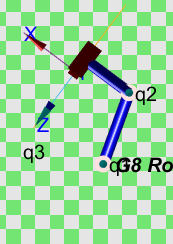
alpha2 = -90 alpha2 = 90 alpha2 = 0

offset1 = -20º offset1 = 0º offset1 = 20º

offset2 = 16.8º offset2 = 0º offset1 = -16.8º

* 1. Explain each RTB command used to build the graphical model.

Revolute construct a revolute joint+link using standard DH

SerialLink create a serial-link arm-type robot. Each link and joint in the chain is described by a Link-class object using DH parameters

Plot plot robot arm in the figure with joint angles

* 1. Insert the joint limits (as measured from the real robot) into the model and visually verify that the robot looks and moves like the actual robot.

1. Build in an interface for your robot that allows you to move each of the joints (by a slider) in the graphical user interface and show its movements via forward kinematics.
2. Also create the ability to position the end-effector by small increments using the arrow keys of your keyboard. The position of the end-effector should be displayed in the GUI.
3. In at least 5 positions (pick easy joint angles, like 0, 90, -90 etc.), predict where the EE of the

robot will be with respect to the clipboard coordinate system (use the robots base transform in

your model to move the base) and verify the distance computation with actual measurements.

Place the actual hardware measurements in a file with each line being the joint angles and then

the distance measured. Explain any error that you observe. Add GUI elements to make this easy to do.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Joint1 | Joint2 | Joint3 | Measured | Calculated |
| Pose1\_000 | 0 | 0 | 0 | 0.212 | 0.2047 |
| Pose2\_090 | 0 | 90 | 0 | 0.115 | 0.1095 |
| Pose3\_0-90 | 0 | -90 | 0 | 0.233 | 0.2273 |
| Pose4\_900 | 90 | 0 | 0 | 0.118 | 0.1119 |
| Pose5\_990 | 90 | 90 | 0 | 0.087 | 0.07868 |
| Pose6\_9-90 | 90 | -90 | 0 | 0.136 | 0.1296 |
| Pose7\_-900 | -90 | 0 | 0 | 0.264 | 0.2576 |
| Pose8\_-990 | -90 | 90 | 0 | 0.175 | 0.1759 |
| Pose9\_-9-90 | -90 | -90 | 0 | 0.231 | 0.2292 |



1. Write an objective function and run a minimization function (fminsearch)

Optimization result:

