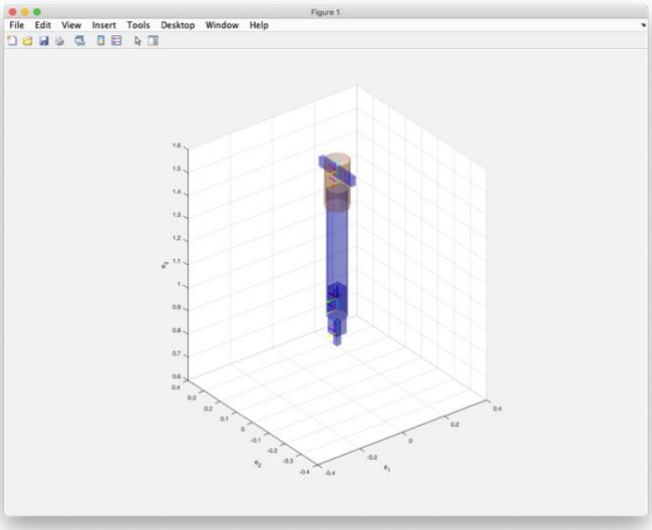
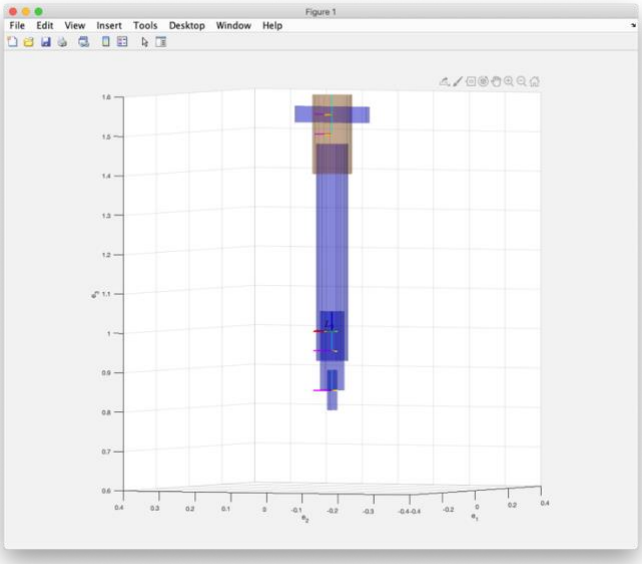
makeMyRobot

-Nayan Man Singh Pradhan

This folder contains multiple files. The file I made is “makeMyRobot.m”.

My robot looks like this:





This robot is a simple robot that makes screwing and unscrewing screws with square holes (like shown below) easier. The robot can be kept on the square holes and turned on. When turned on, cylinder part of the robot rotates, rotating everything below it too. This mechanism unscrews/screws the screw. The robot contains a cap (yellow cylinder on top) and a handle (blue box on the cap) that is fixed. These structures are used to hold the robot and do not rotate, while the bottom cylinder (along with everything below) rotates.

In my code, I made several Links (named Link0, link1, link2, link5 and link6). These links (structures) are variations of two structures: box and cylinder with different dimensions. A screenshot of a link (link0) is provided below:

```
%% Link-0: Base-link -> parent

trf_link0_E= make_transform([0, 0, 1], 0, 0, pi/2, trf_E_axes);
plot_axes(trf_link0_E, 'L_0', false, axis_length);

trf_viz_link0= make_transform([0, 0, 0.5], 0, 0, 0, trf_link0_E);
length0= 0.2; radius0= 0.05;
h(1)= link_cylinder(radius0, length0, trf_viz_link0, [0.823529 0.411765 0.117647]);
plot_axes(trf_viz_link0, ' ', true, axis_length); % V_0
```

This is the main parent link. Other links are made similarly in the file.

An example of a fixed joint (joint 1) and a revolute joint (joint 4) is provided below:

```
%% Joint 1
trf_joint1_link0= make_transform([0, 0, 0], 0, 0, 0, trf_link0_E);
make_child(trf_joint1_link0, trf_viz_link1);
```

```
%% Joint 4

j1_rot_axis_j2= [0,0,1]';
j1_rot_angle= 0; % [-pi/2, pi/2]

trf_joint1_link6= make_transform([0, 0, 0], 0, 0, pi/90, trf_link0_E);
trf_link1_joint6= make_transform_revolute(j1_rot_axis_j2, j1_rot_angle, trf_joint1_link6);
plot_axes(trf_link1_joint6, ' ', false, axis_length);
make_child(trf_link1_joint6, trf_viz_link5);
```

Joint 1 is just a fixed joint. There is no revolution in this joint, therefore the structure is simple. Other fixed joints also have similar sturctrues.

For Joint 4, my revolution is done on the z-axis, as I want my cylinder to spin around the z-axis (in order to screw/unscrew the screw), therefore my rotation matrix is **j1_rot_axis-j2 = [0, 0, 1].** Other joints are made similarly.

Finally, in order to show the animation, I modified the provided code. I have added a screenshot of the modified code below:

```
%% Animation: All joints together.
q_init= 0.5*ones(4,1); % This leads to all joints being at 0.
%
for i= 1:20
    q_next= rand(4,1);
    % rand() gives uniformly distributed random numbers in the interval [0,1]

    for t=0:0.02:1
        q= q_init + t*(q_next - q_init);
        q1= (pi/2)*(2*q(1) - 1);
        q2= (pi/2)*(2*q(2) - 1);
        q3= (pi/2)*(2*q(3) - 1);

        set(handle_axes, 'XLim', [-0.4,0.4], 'YLim', [-0.4,0.4], 'ZLim', [0.6,1.6]);
        trf_q1= makehgtform('axisrotate', j1_rot_axis_j1, q1);
        set(trf_joint1_link5, 'Matrix', trf_q1);

        set(handle_axes, 'XLim', [-0.4,0.4], 'YLim', [-0.4,0.4], 'ZLim', [0.6,1.6]);
        trf_q2= makehgtform('axisrotate', j1_rot_axis_j2, q1);
        set(trf_joint1_link6, 'Matrix', trf_q1);

        set(handle_axes, 'XLim', [-0.4,0.4], 'YLim', [-0.4,0.4], 'ZLim', [0.6,1.6]);
        trf_q3= makehgtform('axisrotate', j1_rot_axis_j2, q1);
        set(trf_link1_joint3, 'Matrix', trf_q1);

        drawnow;
        pause(0.005);

     end

    q_init= q_next;

end
```

I have adjusted the XYX axes in order to fit my robot. The animation is simply done by two for loops. This is the animation for all joints at the same time.