
```
% Jesse Wynn HW2 ME 537 Robotics
clc
% clear all
close all
% clear imports

% Problem 2(g) from the hw printout

import +ETS3.*

% link lengths
a1 = 0.5;
a2 = 0.5;
a3 = 0.5;
a4 = 0.25;
a5 = 0.25;
a6 = 0.5;

% define joint min and max
joint_min = -1; % radians
joint_max = 1;
num_positions = 5;

increment = (joint_max - joint_min)/num_positions;

count = 0;

% initialize variables to store data in
x = zeros(1000000, 1);
y = zeros(1000000, 1);
z = zeros(1000000, 1);

% initial joint angles (at the negative extreme
q0 = [-1, -1, -1, -1, -1, -1];

% this way doesn't work
% E3 = Rz('q1') * Tz(a1) * Ry('q2') * Tx(a2) * Ry('q3') * Tx(a3) *
    Rx('q4') * Tx(a4) * Rz('q5') * Tx(a5) * Rx('q6') * Tx(a6);

% so instead I'll use the mdl_puma560 and edit it to match...
%p560.plot(q0)

% step through incremental joint angles to approximate the reachable
    space

% set q = q0
q = q0;

% loop through first joint
for i = 0:num_positions
    q(1) = q(1) + increment;
```

```

% loop through second joint
for j = 0:num_positions
    q(2) = q(2) + increment;

% loop through third joint
for k = 0:num_positions
    q(3) = q(3) + increment;

% loop through fourth joint
for l = 0:num_positions
    if l == 0
        q(4) = q(4);
    else
        q(4) = q(4) + increment;
    end

% loop through 5th joint
for m = 0:num_positions
    if m == 0
        q(5) = q(5);
    else
        q(5) = q(5) + increment;
    end

%           disp(q(5))

% sixth joint is just twisting about the same axis
as 5

% so we won't add another loop

% figure out where the endeffector is using fkine
T = p560.fkine(q);

x_pos = T.t(1);
y_pos = T.t(2);
z_pos = T.t(3);

% increment the counter
count = count + 1;

% add the positions onto growing x vectors
x(count) = x_pos;
y(count) = y_pos;
z(count) = z_pos;

%           p560.plot(q)

%           disp(count)

% reset q(5)
if q(5) == joint_max
    q(5) = joint_min;
end

```

```

        end
        % reset q(4)
        if q(4) == joint_max
            q(4) = joint_min;
        end

        end
        % reset q(3)
        if q(3) == joint_max
            q(3) = joint_min;
        end

        end
        % reset q(2)
        if q(2) == joint_max
            q(2) = joint_min;
        end

        end
        % reset q(1)
        if q(1) == joint_max
            q(1) = joint_min;
        end

    end

    % plots

    % get the data that you want to plot
    x_plot = x(1:count);
    y_plot = y(1:count);
    z_plot = z(1:count);

    figure(1)
    plot3(x_plot, y_plot, z_plot, '.')
    hold on
    xlabel('x position (m)')
    ylabel('y position (m)')
    zlabel('z position (m)')
    title('Robot reachable workspace')
    % p560.plot([0, 0, 0, 0, 0, 0])
    p560.plot(q)

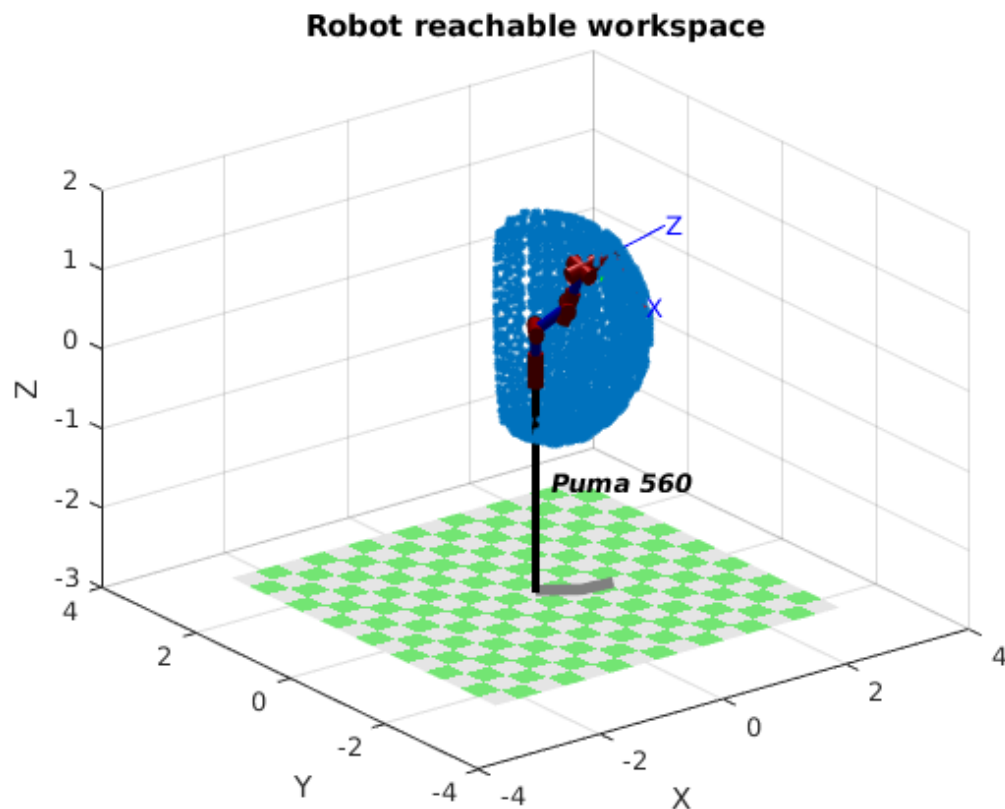
    figure(2)
    plot(x_plot, y_plot, '.')
    xlabel('x position (m)')
    ylabel('y position (m)')
    title('Robot reachable workspace xy plane')

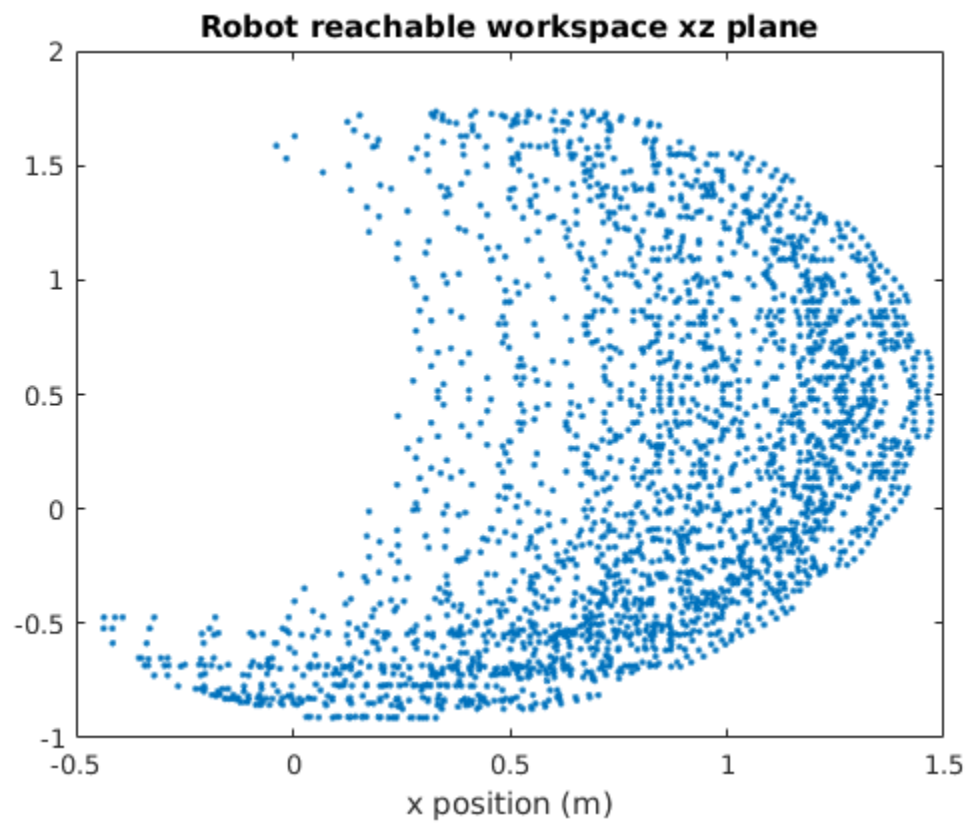
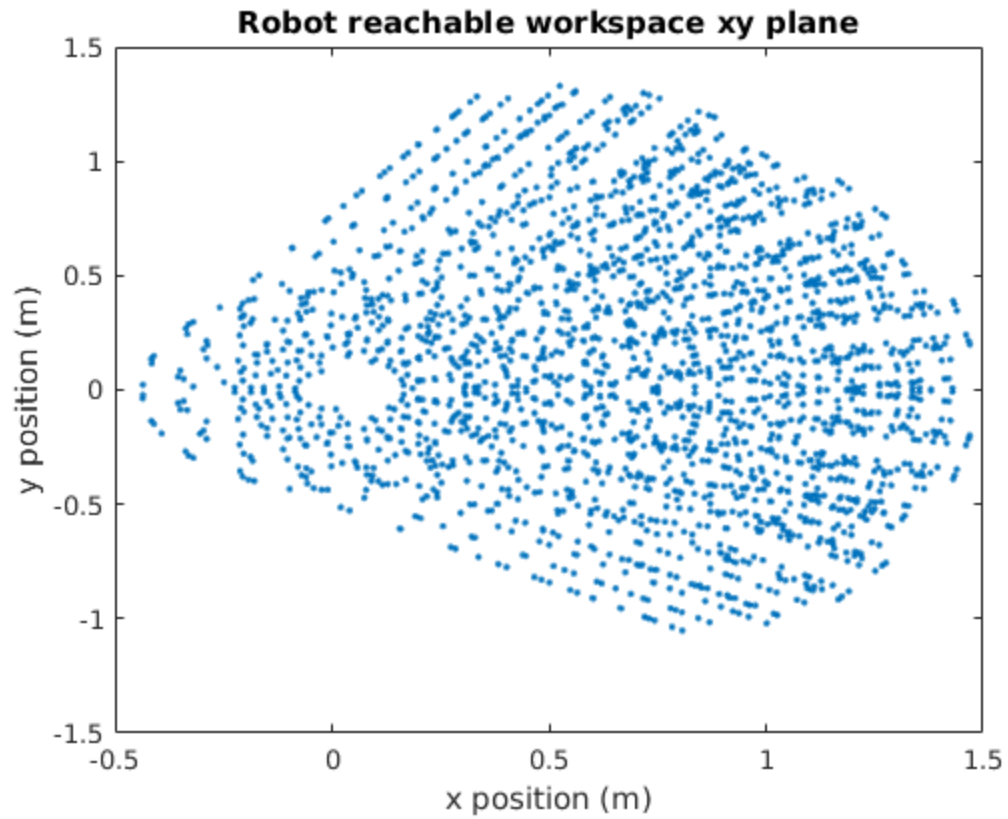
    figure(3)
    plot(x_plot, z_plot, '.')
    xlabel('x position (m)')
    zlabel('z position (m)')
    title('Robot reachable workspace xz plane')

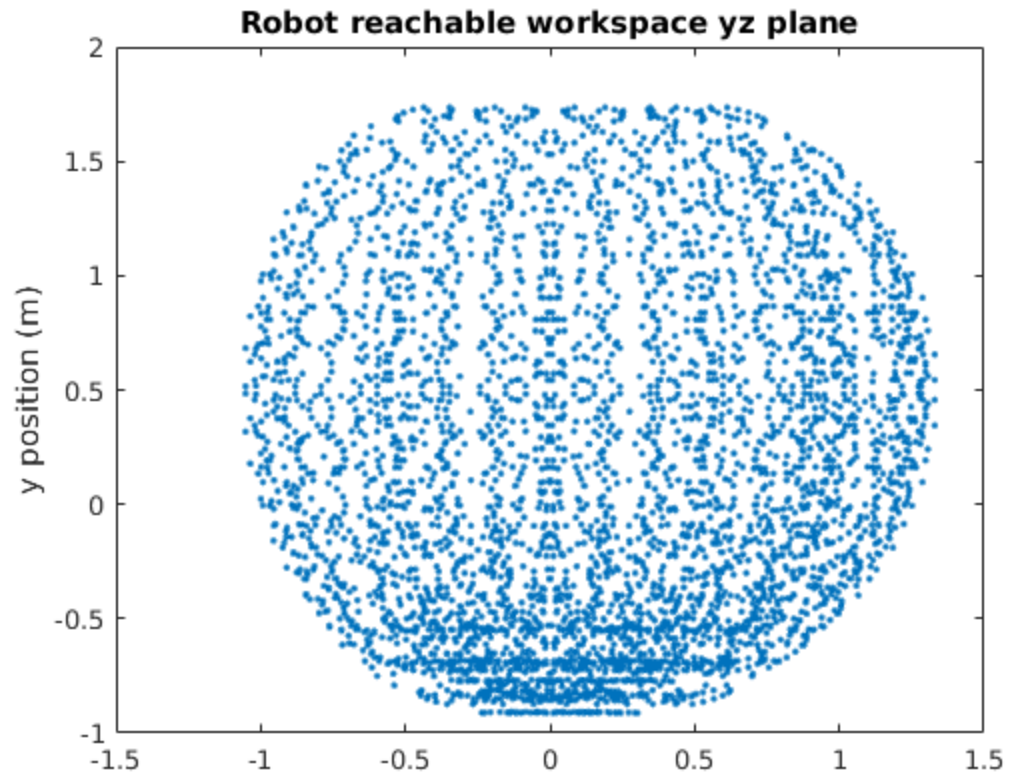
```

```
figure(4)
plot(y_plot, z_plot, '.')
ylabel('y position (m)')
zlabel('z position (m)')
title('Robot reachable workspace yz plane')
```

*Warning: floor tiles too small, making them 2.000000 x bigger - change the size
or disable them
Warning: arrow option requires arrow3 from FileExchange*







Published with MATLAB® R2017a