```
clear;

%physical system parameters
L = 55; % (mm) from spine architecture
d = 4;  % (mm) "
disk_diameter = 15; % (mm) diameter of the disk
PWMrange = 500-100;
setmid = 350;

%set frequency of sampling
res_curve = 18;
res_theta = 40;
iterations = 3;

%create theta
theta = linspace(0,2*pi,res_theta);

% Read in coefficients from any known data as a start.
% This example uses coefficients fit to the xz plane data from previous
% trial runs that just used one iteration as a start point for every plane.
% Starting coefficients could theoretically be calculated from the constant
% cuvature model.
fit_coeffs_tab = readtable("x_coeffs_Q1.csv");
fit_coeffs_prev = table2array(fit_coeffs_tab);
fit_coeffs_prev = fit_coeffs_prev(2:6,1);
xfit_coeffs = zeros(5,iterations+1,res_theta);
z_coeffs_tab = readtable("z_coeffs_Q1.csv");
z_coeffs_prev = table2array(z_coeffs_tab);
z_coeffs_prev = z_coeffs_prev(2:6,1);
zfit_coeffs = zeros(5,iterations+1,res_theta);
for i = 1:res_theta
    xfit_coeffs(:,1,i) = fit_coeffs_prev(:,1);
    zfit_coeffs(:,1,i) = z_coeffs_prev(:,1);
end

%Establish serial connection to Arduino
device = serialport("/dev/tty.usbmodem1301",115200)
flush(device);

%Establish serial connection to Aurora
fser = serialport('/dev/
tty.usbserial-120',115200,'DataBits',8,'FlowControl','none','StopBits',1,'Timeout',0.001);

%read in aurora information
TF_aurora_to_model_tab = readtable('TF_aurora_to_model.csv');
TF_aurora_to_model = table2array(TF_aurora_to_model_tab);
spinetip_in_coilspace_tab = readtable('spinetip_in_coilspace.csv');
spinetip_in_coilspace = table2array(spinetip_in_coilspace_tab);

% load spine model for plotting
spine_stl = stlread('dual_helix.STL');
[f_resamp,v_resamp] =
 reducepatch(spine_stl.ConnectivityList,spine_stl.Points,0.1);
```

```
%spine tip in model space
spine_tip_modelspace = [0,0,L]';

%initiate vars for loop
dl = zeros(4,res_curve,iterations,res_theta);
dlplane = zeros(1,res_curve+1,iterations,res_theta);
dl_all_iterations = zeros(1,(res_curve+1)*iterations,res_theta);
spinedata_in_auroraspace = zeros(3,res_curve);
points_record = zeros(7,res_curve,iterations,res_theta);
error = zeros(iterations,res_theta);
error_per_point = zeros(iterations,res_theta);
r = zeros(1,res_curve+1,iterations,res_theta);
r_all_iterations = zeros(1,(res_curve+1)*iterations,res_theta);
z_all_iterations = zeros(1,(res_curve+1)*iterations,res_theta);

x_all_coeffs = zeros(5,iterations+1,res_theta);
z_all_coeffs = zeros(5,iterations+1,res_theta);
x_avg_coeffs = zeros(6,res_theta);
z_avg_coeffs = zeros(6,res_theta);
% x_all_coeffs(1,:) = theta;
% z_all_coeffs(1,:) = theta;
x_avg_coeffs(1,:) = theta;
z_avg_coeffs(1,:) = theta;

for i = 1:res_theta
    for k = 1:iterations

    %establish requests, and resulting cable change
    r_req = linspace(0,30,res_curve);
    pos_req = [r_req.*cos(theta(i)); r_req.*sin(theta(i));
        zfit_coeffs(1,k,i)*r_req.^4 + zfit_coeffs(2,k,i)*r_req.^3
 + zfit_coeffs(3,k,i)*r_req.^2 + zfit_coeffs(4,k,i)*r_req.^1 +
 zfit_coeffs(5,k,i)];


    dlplane(1,2:end,k,i) = xfit_coeffs(1,k)*r_req.^4 +
 xfit_coeffs(2,k)*r_req.^3 + xfit_coeffs(3,k)*r_req.^2 +
 xfit_coeffs(4,k)*r_req.^1 + xfit_coeffs(5,k);
    dl(1,:,k,i) = cos(theta(i))*dlplane(1,2:end,k,i);
    dl(2,:,k,i) = -cos(theta(i))*dlplane(1,2:end,k,i);
    dl(3,:,k,i) = sin(theta(i))*dlplane(1,2:end,k,i);
    dl(4,:,k,i) = -sin(theta(i))*dlplane(1,2:end,k,i);

    %convert to PWM motor inputs
    dtheta = dl(:,:,k,i)/(disk_diameter/2);
    dPWM = -dtheta*PWMrange/pi;
    u = ones(4,res_curve)*setmid;
    u = u+dPWM;

    % %plot
    % figure(1)
    % grid on;
    %
 plot(r_req,u(1,:),'.',r_req,u(2,:),'.',r_req,u(3,:),'.',r_req,u(4,:),'.')
```

```
    %Enter motor values
    motorvalue = [setmid setmid setmid setmid]';

    %Write to device and read response
    write(device,motorvalue,"uint16")
    count = size(motorvalue,1);
    response = read(device,count,"uint16")

    %array to record actual positions
    xstar = zeros(length(u),10);

    %now do same for all u sets
    for j = 1:length(u)
        motorvalue = u(:,j)
        j
        k
        i
        write(device,motorvalue,"uint16")
        count = size(motorvalue,1);
        response = read(device,count,"uint16")
        pause(1)

        pkt = [];
        while(isempty(pkt))
            pkt = getAuroraPacket(fser,0.1);
        end
        xstar(j,:) = pkt;
    end

    %return to neutral
    motorvalue = [setmid setmid setmid setmid]';
    write(device,motorvalue,"uint16")
    count = size(motorvalue,1);
    response = read(device,count,"uint16")

    for coil_idx = 1:size(xstar,1)
        q = xstar(coil_idx,3:6);
        t = xstar(coil_idx,7:9);
        spinedata_in_auroraspace(:,coil_idx) = (t +
quatrotate(q,spinetip_in_coilspace'))';
    end

    spinedata_in_modelspace =
hTF(spinedata_in_auroraspace,TF_aurora_to_model,0);
    points_record(1:3,:,k,i) = spinedata_in_modelspace;
    points_record(4:7,:,k,i) = u;

    figure(2)

plot3(spinedata_in_modelspace(1,:),spinedata_in_modelspace(2,:),spinedata_in_modelspace(3
    hold on; grid on; axis equal;

patch('Vertices',v_resamp,'Faces',f_resamp,'EdgeColor','k','FaceColor',"#0072BD",'LineWid
```

```
    xlabel('x')
    ylabel('y')
    zlabel('z')

    r(1,2:end,k,i) =
 sqrt(spinedata_in_modelspace(1,:).^2+spinedata_in_modelspace(2,:).^2);

    for a = 1:res_curve
        if dlplane(1,a,k,i) > 0
            dlplane(1,a,k,i) = -dlplane(1,a,k,i);
        end
    end

    %add known (0,0) point for curve fitting
    r(1,:,k,i) = [0 r(1,2:end,k,i)];
    dlplane(1,:,k,i) = [0 dlplane(1,2:end,k,i)];
    zforfit = [L spinedata_in_modelspace(3,:)];

    [xfit_motor1, gofx] = fit(r(1,:,k,i)',dlplane(1,:,k,i)','poly4')
    [zfit_motor1, gofz] = fit(r(1,:,k,i)',zforfit','poly4')

    xfit_coeffs(:,k+1,i) = [xfit_motor1.p1; xfit_motor1.p2; xfit_motor1.p3;
 xfit_motor1.p4; xfit_motor1.p5];
    z_coeffs(:,k+1,i) = [zfit_motor1.p1; zfit_motor1.p2; zfit_motor1.p3;
 zfit_motor1.p4; zfit_motor1.p5];

    %aggregate data for averaging
    dl_all_iterations(1,(k-1)*(res_curve+1)+1:k*(res_curve+1),i) =
 dlplane(1,:,k,i);
    r_all_iterations(1,(k-1)*(res_curve+1)+1:k*(res_curve+1),i) = r(1,:,k,i);
    z_all_iterations(1,(k-1)*(res_curve+1)+1:k*(res_curve+1),i) = zforfit;

    figure(3)
    plot(xfit_motor1,r(1,:,k,i),dlplane(1,:,k,i))
    grid on;
    xlabel('r')
    ylabel('dlplane')
    title('In-Plane Defined by Theta')

    end
    x_all_coeffs(:,:,i) = xfit_coeffs(:,:,i);
    z_all_coeffs(:,:,i) = z_coeffs(:,:,i);

    [xfit_motor1, gofx] =
 fit(r_all_iterations(1,:,i)',dl_all_iterations(1,:,i)','poly4')
    [zfit_motor1, gofz] =
 fit(r_all_iterations(1,:,i)',z_all_iterations(1,:,i)','poly4')

    x_avg_coeffs(2:6,i)  = [xfit_motor1.p1; xfit_motor1.p2; xfit_motor1.p3;
 xfit_motor1.p4; xfit_motor1.p5];
    z_avg_coeffs(2:6,i) = [zfit_motor1.p1; zfit_motor1.p2; zfit_motor1.p3;
 zfit_motor1.p4; zfit_motor1.p5];
end
```

```
writematrix(x_all_coeffs,'x_all_coeffs.csv')
writematrix(z_all_coeffs,'z_all_coeffs.csv')
writematrix(points_record,'points_record.csv')
writematrix(dl_all_iterations,'dl_all_iterations.csv')
writematrix(x_avg_coeffs,'x_avg_coeffs.csv')
writematrix(z_avg_coeffs,'z_avg_coeffs.csv')
```

```
Error using serialport
Unable to connect to the serialport device at port '/dev/tty.usbmodem1301'.
Verify that a device is connected to the port, the port is not in use, and all
serialport input arguments and parameter values are supported by the device.
See <a href="matlab: helpview(fullfile(docroot,'matlab','helptargets.map'),
'serialport_connectError')">related documentation</a> for troubleshooting
steps.

Error in iterative_fitting (line 40)
device = serialport("/dev/tty.usbmodem1301",115200)
```

Published with MATLAB® R2023a