```
%ECE 4560 - Homework 11.2
%Caitlyn Caggia

l1 = 1; l2 = 0.5; l3 = 0.25;
ai = [-pi/6; pi/4; -pi/3];
af = [0; -pi/12; pi/4];

%part a
 -----------------------------------------------------------------
gi1 = SE2([0,0],ai(1));
gi2 = SE2([l1,0],ai(2));
gi3 = SE2([l2,0],ai(3));
gi4 = SE2([l3,0],0);
gi = gi1 * gi2 * gi3 * gi4

gf1 = SE2([0,0],af(1));
gf2 = SE2([l1,0],af(2));
gf3 = SE2([l2,0],af(3));
gf4 = SE2([l3,0],0);
gf = gf1 * gf2 * gf3 * gf4

%part b
 -----------------------------------------------------------------
Ti = getTranslation(gi);
Tf = getTranslation(gf);
xi = Ti(1);
yi = Ti(2);
xf = Tf(1);
yf = Tf(2);
thetai = getTheta(gi);
thetaf = getTheta(gf);

givec = [Ti; thetai]
gmvec = givec + (2.5/5.0) * [xf - xi; yf - yi; thetaf - thetai]
gfvec = [Tf; thetaf]

alphaib = inversekin(givec)
alphamb = inversekin(gmvec)
alphafb = inversekin(gfvec)

%part c
 -----------------------------------------------------------------
expi = givec;
expf = gfvec;
z = SE2.log(inv(gi)*gf);

expi = gi * SE2.exp(z*0);
giexp = [getTranslation(expi); getTheta(expi)]
expm = gi * SE2.exp(z*0.5); %halfway point
gmexp = [getTranslation(expm); getTheta(expm)]
expf = gi * SE2.exp(z*5,5);
gfexp = [getTranslation(expf); getTheta(expf)]
```

```
alphaic = inversekin(giexp)
alphamc = inversekin(gmexp)
alphafc = inversekin(gfexp)

%part d
 -----------------------------------------------------------------
figure
tlong = linspace(0,5,100);
pos = zeros(length(tlong), 3);
alphab = zeros(length(tlong), 3);
for i = 1:length(tlong)
    t = tlong(i);
    pos(i,1) = xi + t/5*(xf - xi);
    pos(i,2) = yi + t/5*(yf - yi);
    pos(i,3) = thetai + t/5*(thetaf - thetai);
    alphab(i,:) = inversekin([pos(i,1); pos(i,2); pos(i,3)]);
end
plot(tlong, alphab)
title('part b alphas')
legend('alpha1', 'alpha2', 'alpha3')

figure
alphac = zeros(length(tlong), 3);
for i = 1:length(tlong)
    t = tlong(i);
    exp = expi * SE2.exp(z*t);
    gexp = [getTranslation(exp); getTheta(exp)];
    alphac(i,:) = inversekin([gexp(1); gexp(2); gexp(3)]);
end
plot(tlong, alphac)
title('part c alphas')
legend('alpha1', 'alpha2', 'alpha3')

figure
alphaold = zeros(length(tlong),3);
for i = 1:length(tlong)
    t = tlong(i);
    alphaold(i,1) = [-pi/6 0  pi/50 -pi/375] * [1; t; t^2; t^3];
    alphaold(i,2) = [pi/4 0 -pi/25 2*pi/375] * [1; t; t^2; t^3];
    alphaold(i,3) = [-pi/3 0 7*pi/100 -7*pi/750] * [1; t; t^2; t^3];
end
plot(tlong, alphaold);
title('alphas from HW10.2');
legend('alpha1', 'alpha2', 'alpha3');

%part e
 -----------------------------------------------------------------
figure
hold on
planarR3_display(alphaib)
planarR3_display(alphamb)
planarR3_display(alphafb)
title('part b end effector')
```

```
figure
hold on
planarR3_display(alphaic)
planarR3_display(alphamc)
planarR3_display(alphafc)
title('part c end effector')

figure
hold on
l = [l1; l2; l3];
tlong = linspace(0,5,100);
alpha = zeros(length(tlong), 3);
for i = 1:length(tlong)
    t = tlong(i);
    alpha(i,1) = [-pi/6 0  pi/50 -pi/375] * [1; t; t^2; t^3];
    alpha(i,2) = [pi/4 0 -pi/25 2*pi/375] * [1; t; t^2; t^3];
    alpha(i,3) = [-pi/3 0 7*pi/100 -7*pi/750] * [1; t; t^2; t^3];
end
planarR3_display(alpha(1,:), l); %initial position
planarR3_display(alpha(25,:), l);
planarR3_display(alpha(50,:), l);
planarR3_display(alpha(75,:), l);
planarR3_display(alpha(100,:), l); %final position
title('end effector trajectory from HW 10.2')

%part f
 ------------------------------------------------------------------
disp('The alphas and end effector trajectories do not match between
 (b) and (c).')
disp('Since joint angles generate curved trajectories, the splines
 cannot exactly fit.')
disp('Increasing the number of points on the spline would increase
 accuracy in all cases.')


%function to compute inverse kinematics
 ----------------------------------
function alphas = inversekin(ge)

l1 = 1; l2 = 0.5; l3 = 0.25;

geh = [R(ge(3)) [ge(1); ge(2)]; 0 0 1];
g4 = [l3*cos(ge(3)); l3*sin(ge(3)); ge(3)];
g4h = [R(ge(3)) [l3*cos(ge(3)); l3*sin(ge(3))]; 0 0 1];
gw = geh *inv(g4h);

xw = gw(1,3);
yw = gw(2,3);
thetae = ge(3);

gamma = atan2(yw, xw);
r = sqrt(xw^2 + yw^2);
delta = acos((l1^2 + l2^2 - r^2)/(2*l1*l2)) - pi;
```

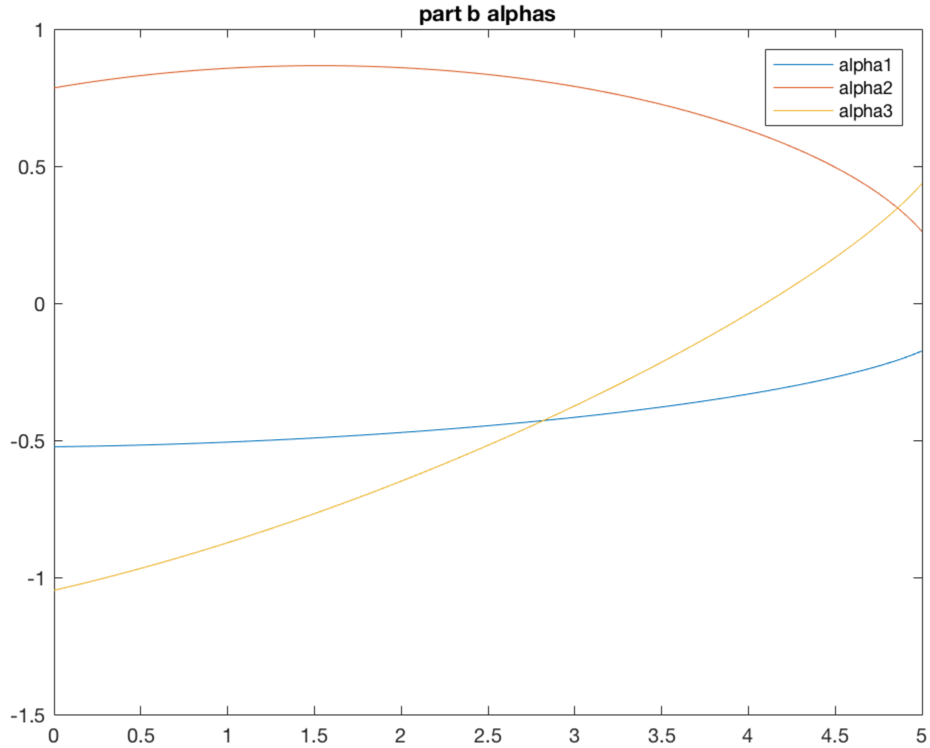
```
alpha = acos((l1^2 + l2^2 - r^2)/(2*l1*l2));
beta = acos((l1^2 + r^2 - l2^2)/(2*l1*r));

a1 = gamma - beta;
a2 = pi - alpha;
a3 = thetae - a1 - a2;

% all angles should fall in the range [-pi, pi]
if (a3 > pi)
    a3 = a3 - 2*pi;
elseif (a3 < -pi)
    a3 = a3 + 2*pi;
end

alphas = [a1; a2; a3];

end
```

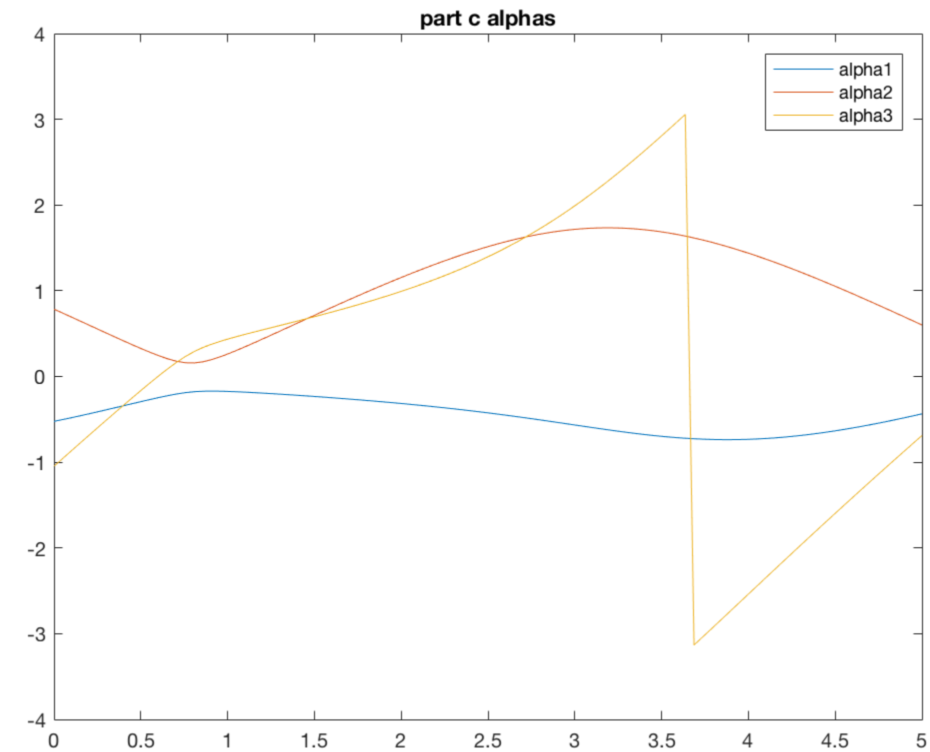
```
    0.7071    0.7071    1.5258
   -0.7071    0.7071   -0.5474
         0         0    1.0000

    0.8660   -0.5000    1.6995
    0.5000    0.8660   -0.0044
         0         0    1.0000


givec =

    1.5258
   -0.5474
   -0.7854


gmvec =

    1.6126
   -0.2759
   -0.1309


gfvec =

    1.6995
   -0.0044
    0.5236


alphaib =

   -0.5236
    0.7854
   -1.0472
```

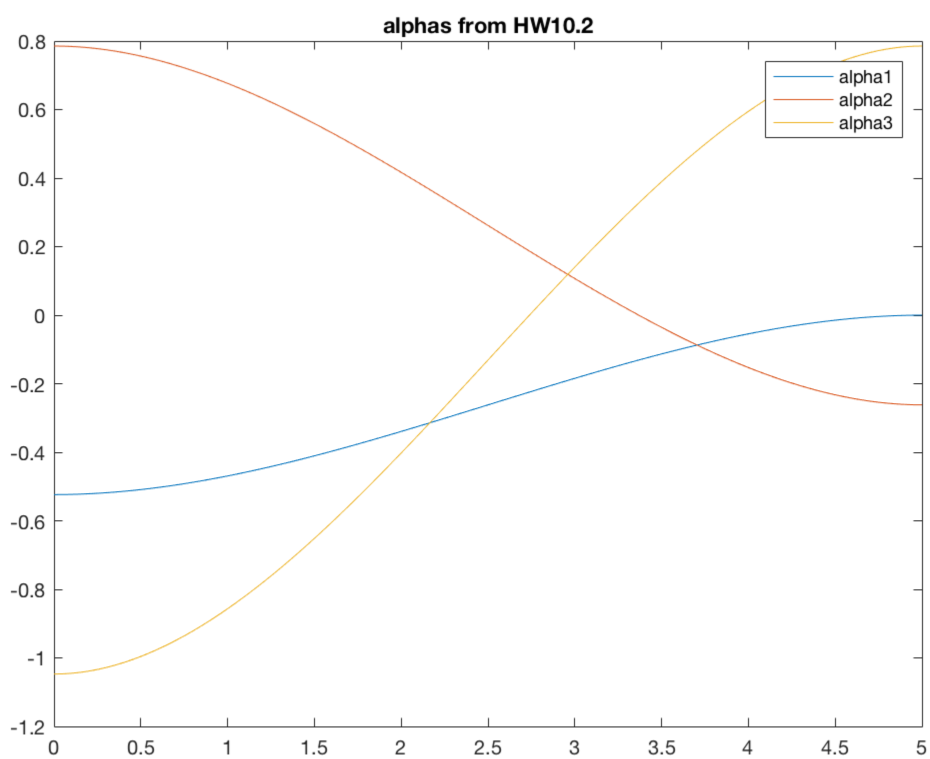
```
alphamb =

   -0.4469
    0.8343
   -0.5183


alphafb =

   -0.1741
    0.2618
    0.4359


giexp =

    1.5258
   -0.5474
   -0.7854


gmexp =

    1.7048
   -0.3054
   -0.1309


gfexp =

    1.6995
   -0.0044
    0.5236


alphaic =

   -0.5236
    0.7854
   -1.0472


alphamc =

   -0.2937
    0.3271
   -0.1643


alphafc =

   -0.1741
    0.2618
```

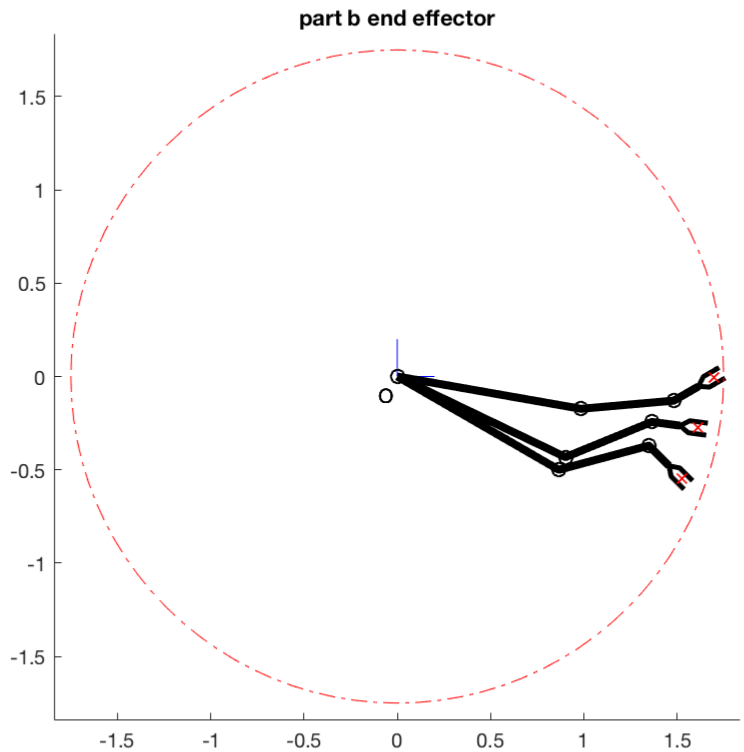
0.4359

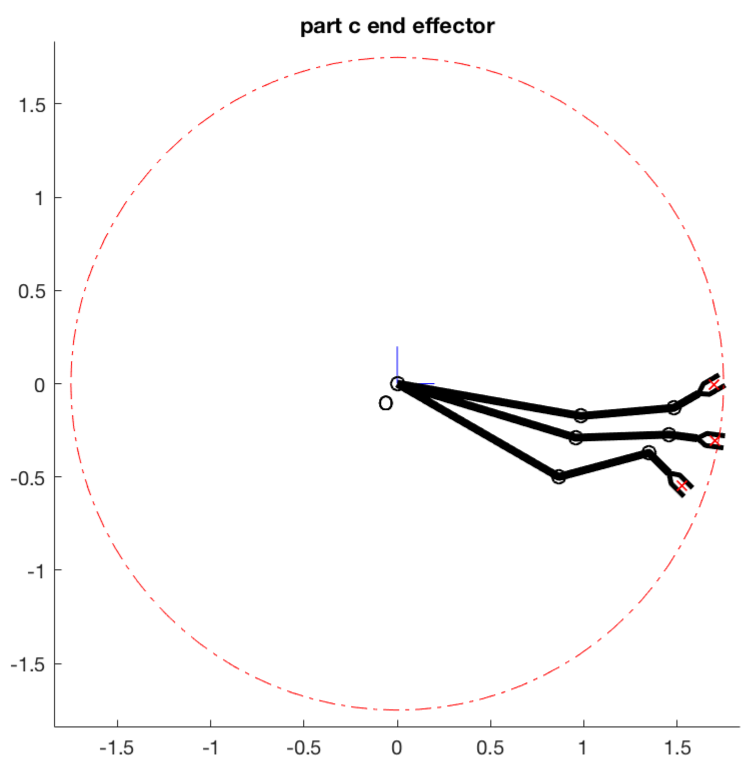
The alphas and end effector trajectories do not match between (b) and (c).
Since joint angles generate curved trajectories, the splines cannot exactly fit.
Increasing the number of points on the spline would increase accuracy in all cases.

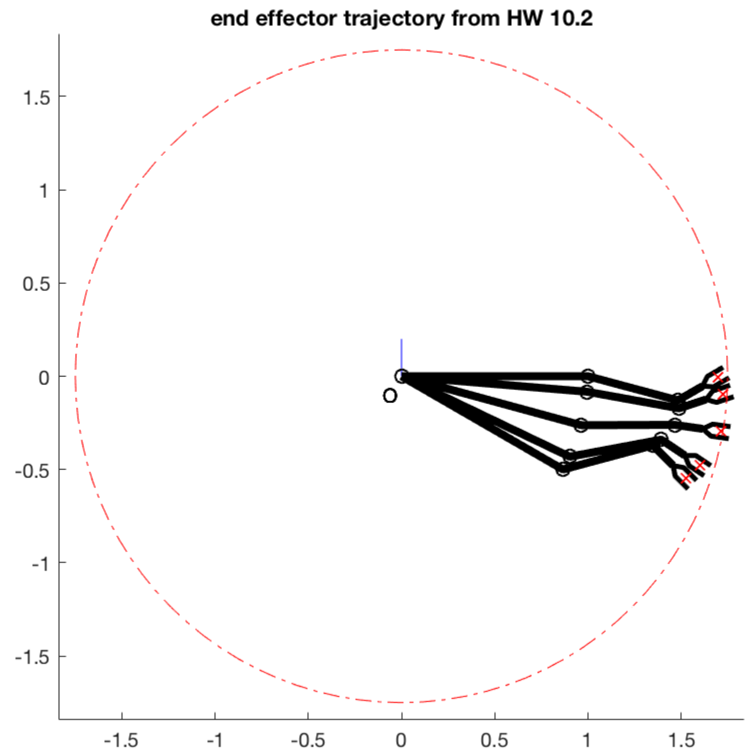



part c alphas
alpha1
alpha2
alpha3
4
3
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


alphas from HW10.2
alpha1
alpha2
alpha3
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


part b end effector
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
O


part c end effector
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
O



Published with MATLAB® R2017a