HW #9

Submitted by Jesse Austin Stringfellow, Due Dec. 3, 2019

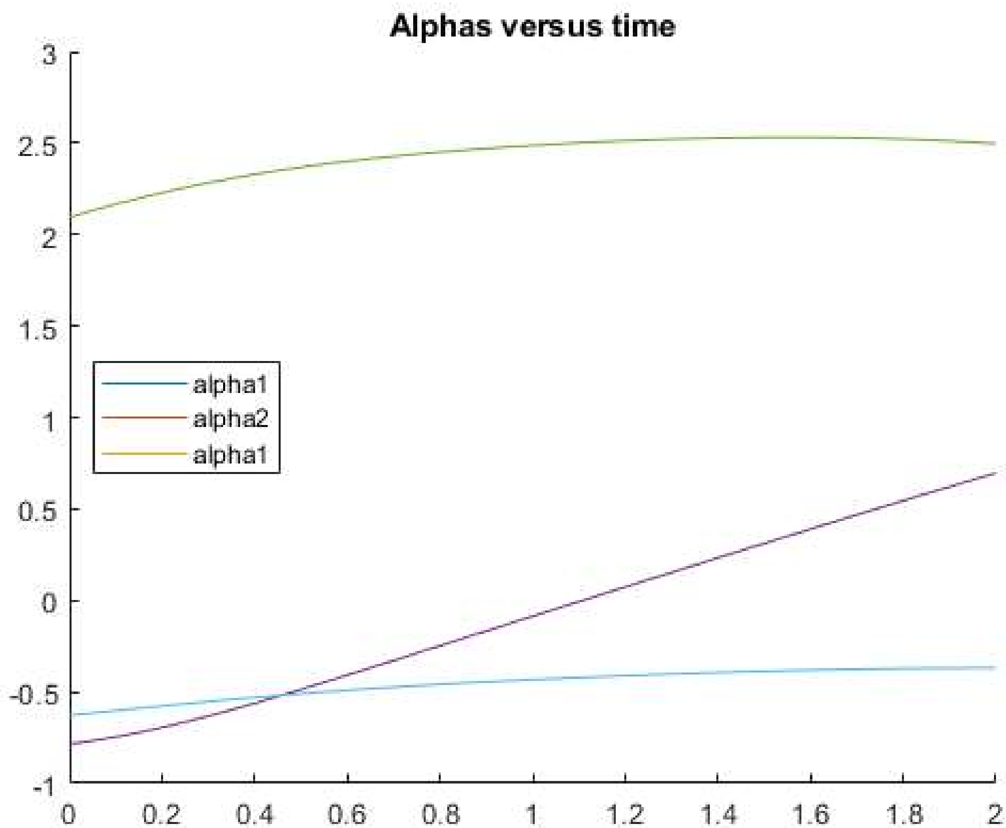
Contents

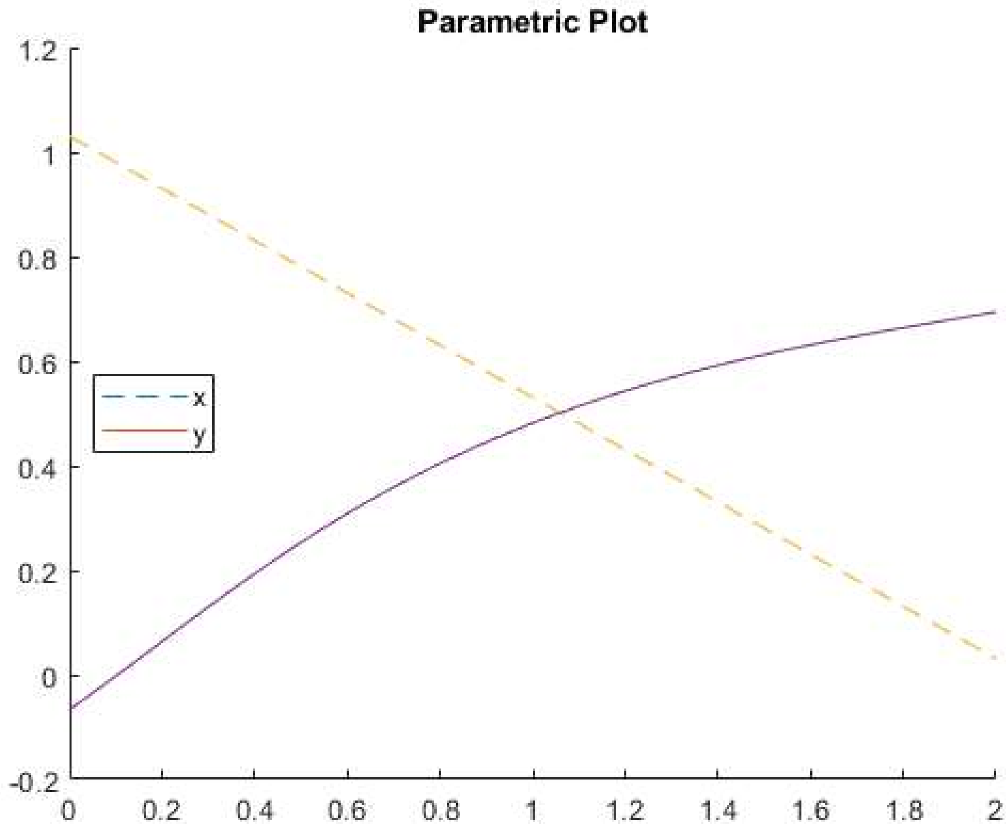
Problem #1

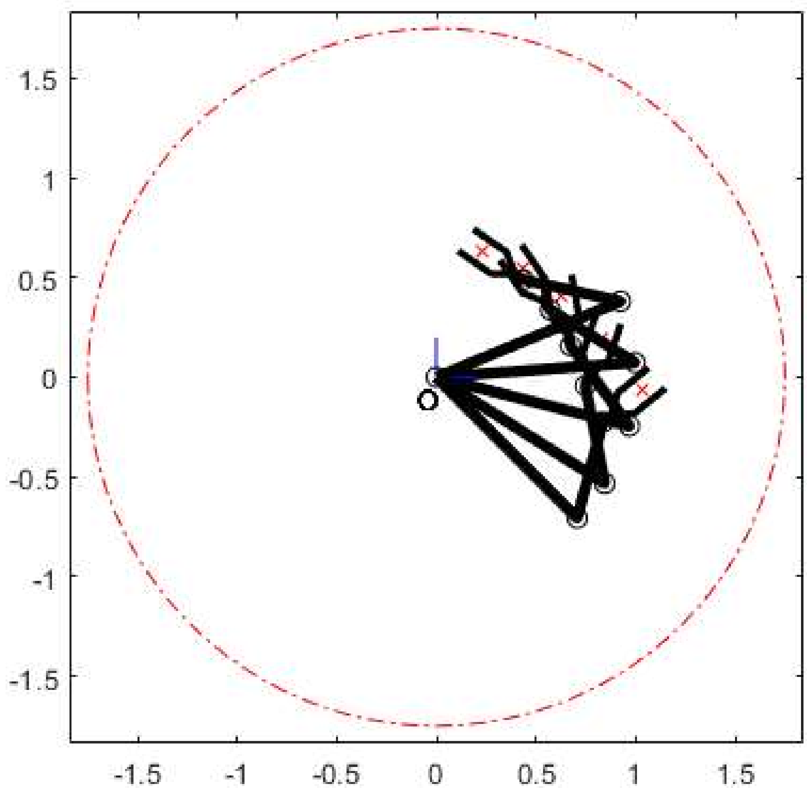
```
alpha = [-pi/4 2*pi/3 -pi/5];
l1 = 1;
l2 = 1/2;
l3 = 1/4;
links=[1;1/2;1/4];
tspan = [0,2];
[tsol, asol] = ode45(@rrDiffEq, tspan, alpha);
figure(1);
hold on;
plot(tsol,asol)
title('Alphas versus time')
legend({'alpha1','alpha2','alpha1'},'Location','west')
hold off;
alpha1Cnt = numel(asol(:,1)); % Space allocation
x = zeros(1,alpha1Cnt);
y = zeros(1,alpha1Cnt);
j = 1;
a1=links(1);
a2=links(2);
a3=links(3);

for i=1:alpha1Cnt
    psol1=asol(i,1);
    psol2=asol(i,2);
    psol3=asol(i,3);
      x(j) = a1*cos(psol1)+ a2*cos(psol1+psol2) + a3*cos(psol1+psol2+psol3);
      y(j) = a1*sin(psol1)+ a2*sin(psol1+psol2) + a3*sin(psol1+psol2+psol3);
      j = j+1;
end
figure(2);
hold on;
tsteps = linspace(tspan(1),tspan(2),length(x));
plot(tsteps,x,'--');
plot(tsteps,y,'-');
title('Parametric Plot')
legend({'x','y'},'Location','west')
hold off;
figure(3);
tsteps = linspace(tspan(1),tspan(2),10);
```

```
i = 1;
for t=1:length(asol)/8
psol1=asol(i,1);
psol2=asol(i,2);
psol3=asol(i,3);
planarR3_display([psol1; psol2; psol3], [links(1); links(2); links(3)], 1/5)
hold on
i = i+8; % Allows for plotting only some of the solutions
end
```




Parametric Plot
x
y
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


1.5
1
0.5
0
-0.5
-1
-1.5
O
-1.5
-1
-0.5
0
0.5
1
1.5

Problem 2

```
%a & b)
%Shows the process by which one finds the inverse kinematics while
%computing it
l1 = 1; l2 = 1/2; l3 = 1/4;
gestar = SE2([1.556; .7288],.7854);
g4 = SE2([l3;0],0);
gwstar = gestar*inv(g4);
xw = gwstar.M(1,3);
yw = gwstar.M(2,3);
y = atan2(yw,xw);
r = sqrt((xw)^2+(yw)^2);
delta = acos((l1^2+l2^2-r^2)/(2*l1*l2));
beta = acos((l1^2+r^2-l2^2)/(2*l1*r));
a1 = y+beta;
a2 = delta - pi;
a3 = .7854-a1 - a2;
alphas1 = [a1;a2;a3]
a1_2 = y-beta;
a2_2 = pi - delta;
a3_2 = .7854-a1_2 - a2_2;
alphas2 = [a1_2;a2_2;a3_2]
```

```
alphas1 =

    0.4785
   -0.2944
    0.6012


alphas2 =

    0.2829
    0.2944
    0.2081
```

Problem 3

HW9 - Problem 3

```
dt = 0.01;
% Robot's initial position
x0 = [-1.2; 0];
% Goal position
xg = [1.2; 0];
% Obstacle position
ObsPos = 2;          % <==== SWITCH to "2" for point b) <========
switch ObsPos
    case 1
        xo = [0.2; 0.3];
    case 2
        xo = [0.2; 0];
end
% Coefficients for goal potential
```

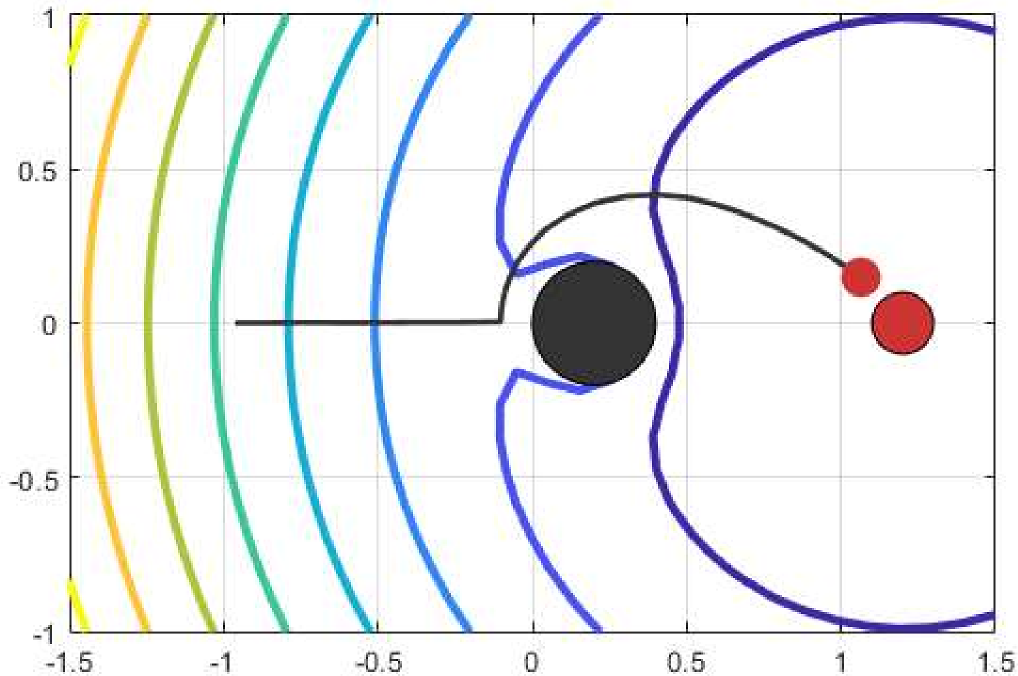
```
Kg = 100*[0.1,0;0,0.1];
% Coefficients for obstacle potential
ko = .36;
% Potential Functions
Vg = @(x,y)(0.5*([x;y]-xg)'*Kg*([x;y]-xg));
Vo = @(xo, x,y)( ko*0.5/( ([x;y]-xo)'*([x;y]-xo) ) );
% Open plot
figure, set(gcf, 'color', 'white'), hold on
% Plot potential
nx = 30;
ny = 20;
xgrid = linspace(-1.5,1.5,nx);
ygrid = linspace(-1,1,ny);
[xx,yy] = meshgrid(xgrid , ygrid);
zz = zeros(ny,nx);
for i = 1:nx
    for j = 1:ny
        zz(j,i) = Vg(xgrid(i),ygrid(j)) + Vo(xo,xgrid(i),ygrid(j));
    end
end
[M,c] = contour(xx,yy,zz);
c.LineWidth = 3;
% Plot robot and environment features
trPl = plot(x0(1), x0(2), '-', 'color',[0.2,0.2,0.2], 'linewidth',2); % trajectory
rb_pl = plot(x0(1), x0(2), 'o', 'markersize',14, 'markerfacecolor',[0.8,0.2,0.2], 'markeredge
color','none'); % robot
patch(xo(1)+0.2*cos(0:0.1:2*pi),xo(2)+0.2*sin(0:0.1:2*pi),[0.2,0.2,0.2]); % obstacle
patch(xg(1)+0.1*cos(0:0.1:2*pi),xg(2)+0.1*sin(0:0.1:2*pi),[0.8,0.2,0.2]); % goal
grid on, box on
axis([-1.5 1.5 -1 1]), axis equal
% Set robot's initial conditions
xt = x0(1);
yt = x0(2);
traj = [xt; yt];    % collect trajectory points (for plot purposes only)
k = 0;
while ( [xt;yt]-xg )'*( [xt;yt]-xg ) > 0.05
    k = k+1;

    % COMPLETE !!!!
    Fg = Kg*(xg-[xt;yt]);
    Fo = ko*([xt;yt]-xo)/(([xt;yt]-xo)'*( [xt;yt]-xo )).^2;

    F = Fg +Fo;

    % Update Single Integrator dynamics
    xt = xt + dt*F(1);
    yt = yt + dt*F(2)+.001*randn(1); % Added random number to yt so robot ...
    %...can move around obstacle


    % Update plot
    set(rb_pl, 'xdata', xt, 'ydata', yt)
    traj(:,k) = [xt; yt];
    set(trPl,'xdata',traj(1,:),'ydata',traj(2,:))
    pause(0.01)
    if k > 3000; break; end
end
```



Problem 3

Part b)The robot does not reach the goal. It runs into the object because it is perfectly centered with the robot. Because of this the robot's y value is eseentially in equilibrium and is not able to move around the object. If one adds a small random component to the yt value, it allows the robot to break loose from equilibrium and make it around the obstacle.

Functions

```
function mJ = pJacob(alpha)
linklen=[1;1/2;1/4];
s1 = sin(alpha(1));
c1 = cos(alpha(1));
s12 = sin(alpha(1)+alpha(2));
c12 = cos(alpha(1)+alpha(2));
s13 = sin(alpha(1)+alpha(2)+alpha(3));
c13 = cos(alpha(1)+alpha(2)+alpha(3));
mJ = [-linklen(2)* s12-linklen(1)*s1 - linklen(3)*s13,  -linklen(2)*s12 - linklen(3)*s13  , -
linklen(3)*s13; ...
       linklen(2)* c12+linklen(1)*c1 - linklen(3)*c13,   linklen(2)*c12 + linklen(3)*c13  ,
linklen(3)*c13 ];
end

function alphadot = rrDiffEq(t,alpha)
vdes = [-.5;.5];
J = pJacob(alpha);
alphadot = pinv(J)*vdes;
end
```