MATLAB programs to generate the Linear Pendulum: Phase Diagram and the Adjoint and Control Variable Diagram

```
function [  ] = linear_pendulum(  )
%Written by Elmer G. Wiens June 2019
%Linear pendulum
clear

%Set the starting values -5.5 < v1 < 5.5; -5.5 < v2 < 5.5

disp('Starting Values')
v1= 4 % -6 %4
v2= 4 % 0 %4


%Save starting Values
tempv1 = v1;
tempv2 = v2;

mang(1) = 0;
myv1(1) = v1;
myv2(1) = v2;
myu(1) = 0;

d = 7;
%plot path
figure(1)
hold on;
nxax = -d - 2;pxax = d + 2;nyax = -d - 1;pyax = d + 1;
axis([nxax pxax nyax pyax])
axis square
plot(zeros(100),linspace(nyax,pyax))
plot(linspace(nxax,pxax),zeros(100))

title('Linear Pendulum: Phase Diagram', 'FontSize',15)
xlabel('x1', 'FontSize',17)
ylabel('x2', 'FontSize',17)


plot(v1,v2,'y*', 'LineWidth',4)
x = linspace(0,2,40);
for i=1:length(x)
   y(i) = -circles(x(i),1,1);
end

for k=0:2:5
   plot(x+k,y,'r', 'LineWidth', 3);
   plot(-x-k,-y,'r', 'LineWidth', 3)
end

for d = 3:2:7
   x = linspace(-d + 1, d + 1,40);
   for i=1:length(x)
      y(i) = circles(x(i),1,d);
```

```
   end
   plot(x,-y,'g', 'LineWidth', 3)

   x = linspace(-d - 1, d - 1,40);
   for i=1:length(x)
      y(i) = circles(x(i),-1,d);
   end
   plot(x,y,'k', 'LineWidth', 3);
end
totang=0;
loop = 1;
maxloop = 5;
rad = 10;

while (loop <= maxloop) & (rad >= .1)
ang1=0; ang2=0; %//clear angles
plot(v1,v2,'y*', 'LineWidth', 4)
if (v2 >= 0)
   c = -1;
   for k =1:2:7
      lt = sqrt((v1 + k)^2 + v2^2);
      if (lt < .999)
         c = 1;
         break
      end
   end
else
   c = 1;
   for k =1:2:7
      lt = sqrt((v1 - k)^2 + v2^2);
      if (lt <= .999)
         c = -1;
         break
      end
   end
end


d = sqrt((v1-c)^2 + v2^2);
if (c == -1)
   if(v2 >= 0)
      x = linspace(v1, d + c, 40);
      for i=1:length(x)
         y(i) = circles(x(i),c,d);
      end
      plot(x,y,'b', 'LineWidth', 3);
      tv1 = d + c; tv2 = -0;
      if ((v1 - c) == 0)
         ang1 = pi/2;
      else
         tang1 = (v2/(v1 - c));
         ang1 = atan(tang1);
      end
   end
   a = 1;
   for i = 0:d+1
```

```
      if(2*i >= d+c)
         a = 2*i-1;
         break
      end
   end
   %a
   if (a ~= c)

      xint = .5*(d^2 + a^2 -(c^2 + 1))/(a-c);
      yint = sqrt(1 - (xint - a)^2);
      if (a > 0)
         xt = d + c;
         if(v2 < 0)
            xt = v1;
         end
         x = linspace(xint, xt, 40);
         for i=1:length(x)
            y(i) = -circles(x(i),c,d);
         end
         plot(x,y,'m', 'LineWidth', 3);
         tv1 = xint; tv2 = -yint;
         if ((tv1 - c) == 0)
            ang2 = pi/2;
         else
            tang2 = (tv2/(tv1 - c));
            ang2 = atan(tang2);
         end
         if (tv2 == 0)
            tv2 = -.00001;
         end

      end
   end

end




if (c == 1)
   if(v2 <= 0)
      x = linspace(- d + c, v1, 40);
      for i=1:length(x)
         y(i) = -circles(x(i),c,d);
      end
      plot(x,y,'b', 'LineWidth', 3);
      tv1 = - d + c; tv2 = +0;
      if ((v1 - c) == 0)
         ang1 = pi/2;
      else
         tang1 = (v2/(v1 - c));
         ang1 = atan(tang1);
      end
   end
```

```
   a = 1;
   for i = 0:d+1
      if(2*i > d+c)
         a = 2*i-1;
         break
      end
   end
%   d  a   c
%   a = -a + 2

   if (a ~= c)
      xint = .5*(d^2 + a^2 -(c^2 + 1))/(a-c);
      yint = sqrt(1 - (xint - a)^2);
      a = -a + 2;
      xint = -xint + 2;
      if (a < 0)
         xt = -d + c;
         if(v2 > 0)
            xt = v1;
         end
         x = linspace(xt, xint, 40);
         for i=1:length(x)
            y(i) = circles(x(i),c,d);
         end
         plot(x,y, 'm', 'LineWidth', 3);
         tv1 = xint; tv2 = yint;
         if ((tv1 - c) == 0)
            ang2 = pi/2;
         else
            tang2 = (tv2/(tv1 - c));
            ang2 = atan(tang2);
         end
      end
   end
end

%v1 v2 ang1 ang2
if (c == -1) & (v2 >= 0)&(ang1 < 0)
   ang1 = pi + ang1;
end
if (c == -1) & (v2 == 0)&(ang1 == 0)&((v1==-2)|(v1==-4)|(v1==-6))
   ang1 = pi;
end
if (c == 1) & (v2 <= 0)&(ang1 < 0)
   ang1 = pi + ang1;
end


ang3 = ang1 + abs(ang2);
if (loop == 1)
   myang = ang3;
end
totang = totang + ang3;

v1 = tv1; v2 = tv2;
```

```
rad = sqrt(v1^2 + v2^2);

if (loop == 1)
   g = c;
end

myu(loop) = c;

loop = loop + 1;

mang(loop) = totang;
myv1(loop) = tv1;
myv2(loop) = tv2;

end %while end loop

toploop = loop -1;


disp(' Switching Times, Switching Points, Control Variable')
fprintf('%6s %6s %6s %6s\n',' Time ','  X1  ', '  X2  ', '   u  ');

for i = 1:toploop
    fprintf('%6.2f %6.2f %6.2f %6.2f\n', mang(i), myv1(i), myv2(i), myu(i));
end

fprintf('%6.2f %6.2f %6.2f %6.2f\n', mang(loop), 0, 0, myu(toploop));

disp(' ')

charv1 = num2str(tempv1);
charv2 = num2str(tempv2);
charcom= ', ' ;
charbl = '(';
charbr = ')';
charv = [charbl charv1 charcom charv2 charbr];
text(tempv1 -.1, tempv2 + .4, charv, 'FontSize',17)



txtx1 = 4.2;
txtx2 = 7.5;


for i = 1 : toploop
    mngl = round(100*mang(i)) / 100;
    mngr = round(100*mang(i+1))/100;
    chartl = num2str(mngl);
    chartr = num2str(mngr);
    charu = num2str(myu(i));
    charlit = [chartl ' < t < ' chartr ': u(t) = ' charu];
    text(txtx1, txtx2, charlit, 'FontSize', 12)
    txtx2 = txtx2 - 0.7;
```

```
end



syms t

pm1 = g*cos(myang - t);
pm2 = g*sin(myang - t) ;

s = linspace(0, totang, 100);

for i=1:length(s)
   q1(i) = double(subs(pm1, t, s(i)));
   q2(i) = double(subs(pm2, t, s(i)));
   q3(i) = sign(q2(i));
end
q1=double(q1);
q2=double(q2);
q3=double(q3);




figure(2)
hold on;
maxx = max(s)+1;
maxy = 2*max(max(q2)); %2*max(max(q2) + 4);

nxax = 0;pxax = maxx;nyax = -maxy;pyax = maxy;

axis([nxax pxax nyax pyax])
plot(zeros(100),linspace(nyax,pyax))
plot(linspace(nxax,pxax),zeros(100))

title('Linear Pendulum: Adjoint and Control Variables ', 'FontSize',15)
xlabel('t', 'FontSize',17)
ylabel('p1(t) red, p2(t) blue, u(t) yellow', 'FontSize',17)

s = s(1:length(s));
plot(s,q1,'r', 'LineWidth', 3);
plot(s,q2,'b', 'LineWidth', 3);
plot(s,q3,'y', 'LineWidth', 3);

line([totang totang], [q3(length(s)), 0], 'Color','yellow','LineWidth',3)

disp('Terminal Values of the Adjoint Variables')
pT1 = double(subs(pm1, t, totang))
pT2 = double(subs(pm2, t, totang))

return
end
```

```
function y = circles(x, a, r)
%Circle
y = sqrt(r^2 - (x - a)^2);

return
```