

EA721 - Princípios de Controle e Servomecanismo
Turma A

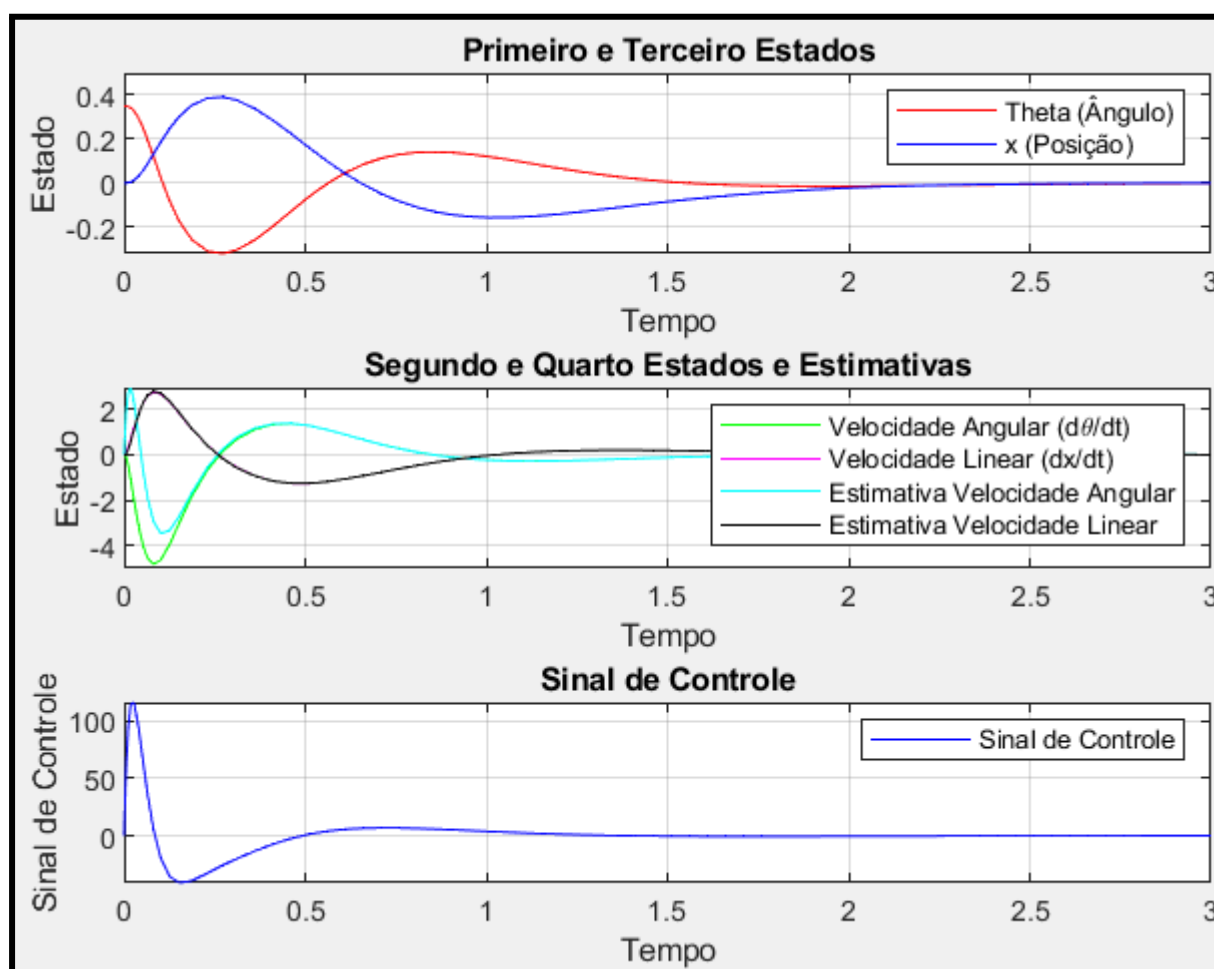
Trabalho Computacional 05

Exercício 1

Thiago Maximo Pavão - 247381

Vinícius Esperança Mantovani - 247395

Comparação dos estados medidos com os estimados pelo observador



Ganhos K e L projetados

```
L =  
  
    1.0e+03 *  
  
    0.0709    -0.0034  
    1.2672    -0.0880  
   -0.0007     0.0691  
   -0.0388     1.1656  
  
K =  
  
   -112.7187   -24.1937   -20.3874   -20.3874
```

Programa MATLAB

```
function regulaPendulo_alunos()  
global A B C D K L m M ell g sinalU;  
%anguloInicial=20; %padrao  
anguloInicial=42; %pre-instabilidade  
%ajuste dos parametros do pendulo  
m=0.5;  
M=2;  
ell=0.5;  
g=9.81;  
%modelo linearizado na origem  
A = [0 1 0 0;  
      (M+m)*g/(M*ell) 0 0 0;  
      0 0 0 1;  
      -(m*g)/M 0 0 0];  
B = [0;-1/(M*ell);0;inv(M)];  
C = [1 0 0 0;  
      0 0 1 0];  
%defina aqui a alocao do observador  
polesObs = [-30+i -30-i -40+2i -40-2i];  
L=place(A',C',polesObs)'  
%defina aqui a alocao do controlador  
polesCon = [-3+i -3-i -4+2i -4-2i];  
K=place(A,B,polesCon)  
optOde = odeset('maxStep',0.05,'OutputFcn',@saida);  
%defina aqui a condicao inicial  
x0 = [deg2rad(anguloInicial) 0 0 0 0 0 0 0]';  
tempoTotal=3;  
outOde = ode45(@penduloNaoLinear,[0 tempoTotal],x0,optOde);
```

```

graficos = 0; % Para gerar gráficos
if graficos
    % Faz os gráficos
    figure; % Cria uma nova figura

    % Subplot 1: primeiro e terceiro estados
    subplot(3,1,1);
    plot(outOde.x, outOde.y(1,:), 'r', 'DisplayName', 'Theta (Ângulo)');
    hold on;
    plot(outOde.x, outOde.y(3,:), 'b', 'DisplayName', 'x (Posição)');
    title('Primeiro e Terceiro Estados');
    xlabel('Tempo');
    ylabel('Estado');
    legend('show');
    grid on;

    % Subplot 2: segundo e quarto estados, e as estimativas
    subplot(3,1,2);
    plot(outOde.x, outOde.y(2,:), 'g', 'DisplayName', 'Velocidade Angular (d\theta/dt)');
    hold on;
    plot(outOde.x, outOde.y(4,:), 'm', 'DisplayName', 'Velocidade Linear (dx/dt)');
    plot(outOde.x, outOde.y(6,:), 'c', 'DisplayName', 'Estimativa Velocidade Angular');
    plot(outOde.x, outOde.y(8,:), 'k', 'DisplayName', 'Estimativa Velocidade Linear');
    title('Segundo e Quarto Estados e Estimativas');
    xlabel('Tempo');
    ylabel('Estado');
    legend('show');
    grid on;

    % Subplot 3: sinal de controle
    subplot(3,1,3);
    plot(outOde.x, sinalU, 'b', 'DisplayName', 'Sinal de Controle');
    title('Sinal de Controle');
    xlabel('Tempo');
    ylabel('Sinal de Controle');
    legend('show');
    grid on;
else
    % Faz o vídeo
    gravarVideo = 0;
    theta = outOde.y(1,:);
    x = outOde.y(3,:);
    parametros = [0.23 2.4 0.36 9.81];
    tempo = outOde.x;
    animatePendulo(gravarVideo, tempo, x, theta, parametros);
end
%
function dX = penduloNaoLinear(t,x)
global m M ell g A B C L K sat umax;
xs=x(1:4);

```

```

xo=x(5:end);
u=-K*xo;
%dinamica da planta
dXs(1,1) = xs(2);
dXs(2,1) = (u*cos(xs(1))-(m+M)*g*sin(xs(1))+m*ell*cos(xs(1))*sin(xs(1))*(xs(2))^2)/(m*ell*(cos(xs(1)))^2 - (m+M)*ell);
dXs(3,1) = xs(4);
dXs(4,1) = ((u+m*ell*sin(xs(1))*(xs(2))^2 - m*g*cos(xs(1))*sin(xs(1)))/(m+M-m*(cos(xs(1)))^2));
%dinamica do observador
dXo = (A - L*C)*xo+L*C*xs+B*u;
dX = [dXs; dXo];
%
function status = saida(t, y, flag)
global sinalU K;
status = 0;
if strcmp(flag, 'init')
    % Inicialização
    u=-K*y(5:8,end);
    sinalU = [u];
elseif isempty(flag)
    % Atualização
    u=-K*y(5:8,end);
    sinalU = [sinalU; u];
elseif strcmp(flag, 'done')
    % Finalização
end
%
function animatePendulo(gravarVideo,tempo,x,theta,parametros)
fig = figure(2);
%dimCarro [largura altura raioRoda];
dimCarro = [3 1.25 0.3];
dimHaste = 5*parametros(3);
raioHaste=0.2;
f=1;
while f <= length(tempo)

    valorTheta = theta(f);
    valorX = x(f);

    hold on;
    axis equal;
    axis([-4 4 -1 5]);

    desenhaCarro(dimCarro,valorX);
    desenhaHaste([valorX dimCarro(2)],valorTheta,dimHaste,raioHaste);

    hold off;
    if gravarVideo
        F{f} = getframe();
    else
        pause(0.1);
    end
end

```

```

    end
    clf(fig);
    fprintf('frame=%d -> [theta x]=[%.3f %.3f]\n',f,valorTheta,valorX);
    f=f+1;
end
if gravarVideo
    geraVideo('videoPendulo_caso2.avi',F);
end
close(fig);
%
function geraVideo(nome,F)
writerObj = VideoWriter(nome,'Motion JPEG AVI');
open(writerObj);
minL=1e10;
minC=1e10;
for f=1:size(F,2)
    if size(F{f}.cdata,1) < minL
        minL = size(F{f}.cdata,1);
    end
    if size(F{f}.cdata,2) < minC
        minC = size(F{f}.cdata,2);
    end
end
for f=1:size(F,2)
    F2.colormap = [];
    for i=1:3
        F2.cdata(1:minL,1:minC,i) =F{f}.cdata(1:minL,1:minC,i);
    end
    writeVideo(writerObj,F2);
end
close(writerObj);
%
function desenhaCarro(dimCarro,x)
%dimCarro = [tamX tamY];
tX=dimCarro(1);
tY=dimCarro(2);
raioRoda = dimCarro(3);
%desenha corpo principal do carro
pts = [x-tX/2 0;
        x+tX/2 0;
        x+tX/2 tY;
        x-tX/2 tY];
f=fill(pts(:,1),pts(:,2),'k');
set(f,'FaceColor',[0 0 1],'FaceAlpha',0.75);
%desenha roda esquerda
meioCirculo([x-0.7*(tX/2) 0],1.33*raioRoda,[0 0 0.7]);
circulo([x-0.7*(tX/2) 0],raioRoda,[0 0 0]);
%desenha roda direita
meioCirculo([x+0.7*(tX/2) 0],1.33*raioRoda,[0 0 0.7]);
circulo([x+0.7*(tX/2) 0],raioRoda,[0 0 0]);
%desenha o solo
f=line([-4 4],[-raioRoda -raioRoda]);
set(f,'Color',[0 0 0]);

```

```

para=1;
%
function desenhaHaste(origem,theta,ell,raioMassa)
dir=[ell*sin(theta) ell*cos(theta)];
dir=dir./norm(dir);
ort(2)=1;
ort(1)=-(dir(2)*ort(2))/dir(1);
ort=ort./norm(ort);
%da origem da haste
p1 = [origem(1)-0.0*ell*sin(theta) origem(2)-0.0*ell*cos(theta)];
p2 = [origem(1)+ell*sin(theta) origem(2)+ell*cos(theta)];
sizeHaste = 0.03;
p2a=p2-sizeHaste*ort;
p2b=p2+sizeHaste*ort;
p1a=p1+sizeHaste*ort;
p1b=p1-sizeHaste*ort;
pts = [p2a;p2b;p1a;p1b];
f=fill(pts(:,1),pts(:,2),'b');
%circulo no final
circulo(p2,raioMassa,[1 0 0]);
%
function circulo(centro,raio,cor)
th = 0:0.1:2*pi;
x = raio*cos(th) + centro(1);
y = raio*sin(th) + centro(2);
f=fill(x,y,'k');
set(f,'FaceColor',cor);
%
function meioCirculo(centro,raio,cor)
th = 0:0.1:pi;
x = raio*cos(th) + centro(1);
y = raio*sin(th) + centro(2);
f=fill(x,y,'k');
set(f,'FaceColor',cor);

```