

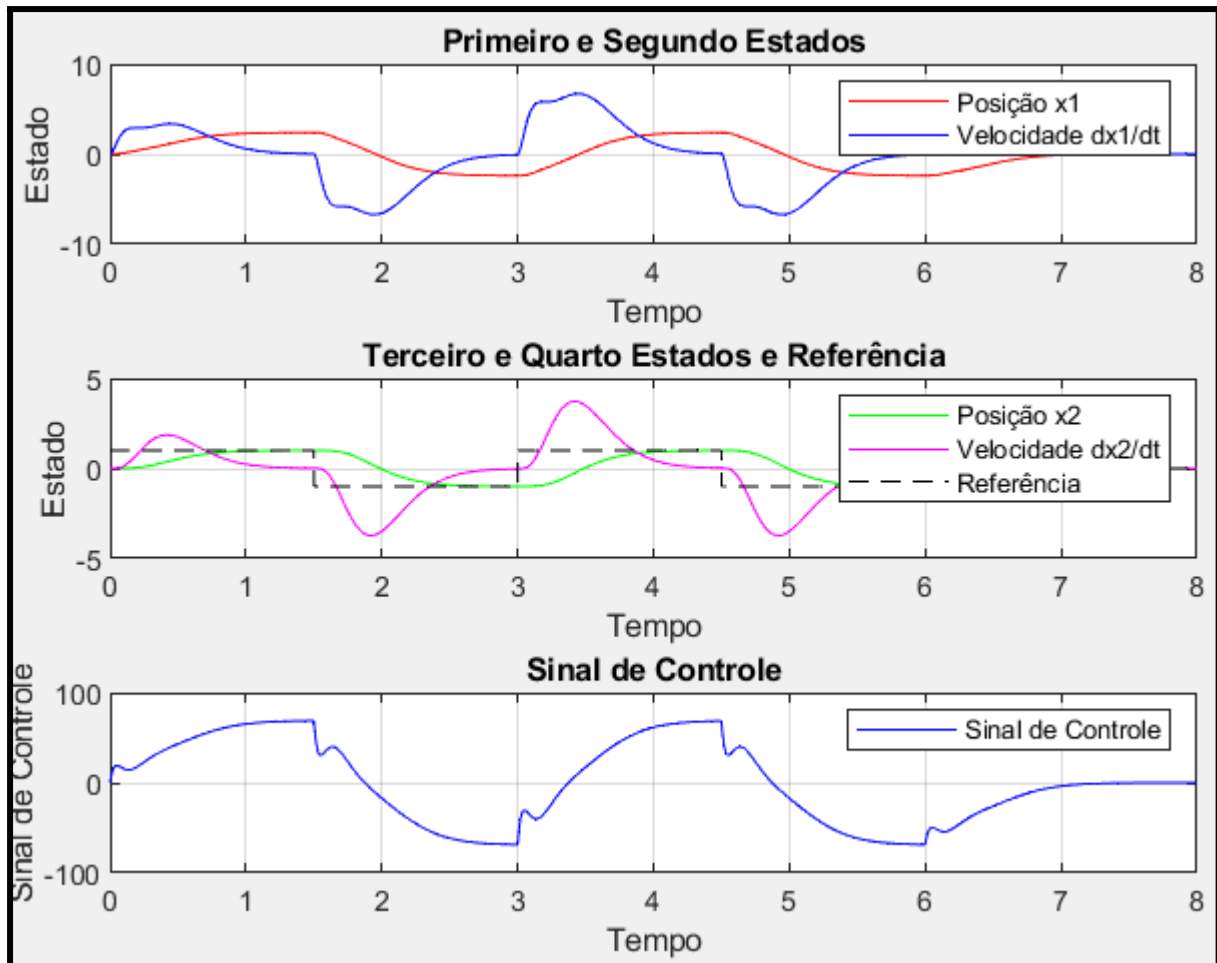
EA721 - Princípios de Controle e Servomecanismo
Turma A

Trabalho Computacional 05

Exercício 2

Thiago Maximo Pavão - 247381

Vinícius Esperança Mantovani - 247395



Ganhos de realimentação e integral

$K =$

$1.0e+03 *$

$0.2975 \quad 0.0206 \quad -0.2187 \quad 0.0314 \quad -1.0816$

E portanto, $K = [297,5 \ 20,6 \ -218,7 \ 31,4]$ e $k_i = -1,0816$

Programa MATLAB

```
function rastreamentoMassaMola()
global A B C K sinalU;
%defina aqui os parametros da planta
m1=0.5;
m2=0.4;
k1=20;
k3=21;
k2=15;
c1=1.5;
c2=1.5;
A = [0 1 0 0;
      -(k1+k2)/m1 -c1/m1 k2/m1 0;
      0 0 0 1;
      k2/m2 0 -(k2+k3)/m2 -c2/m2];
B = [0 inv(m1) 0 0]';
C = [0 0 1 0];
Acal = [A zeros(4,1);
        -C 0];
Bcal = [B;0];
%informe a alocação desejada (5 polos - 4 da planta mais o do integrador)
polesCon = [-8+j -8-j -10+2*j -10-2j -12];
K=place(Acal,Bcal,polesCon)
tempoTotal=8;
optOde = odeset('maxStep',0.05,'OutputFcn',@saida);
%condicao inicial nula
x0 = [0;0;0;0;0];
outOde = ode45(@massaMola,[0 tempoTotal],x0,optOde);
graficos = 1; % Para gerar gráficos
if graficos
    figure; % Cria uma nova figura

    % Subplot 1: primeiro e segundo estados (posição e velocidade de x1)
    subplot(3,1,1);
    plot(outOde.x, outOde.y(1,:), 'r', 'DisplayName', 'Posição x1');
    hold on;
    plot(outOde.x, outOde.y(2,:), 'b', 'DisplayName', 'Velocidade dx1/dt');
    title('Primeiro e Segundo Estados');
    xlabel('Tempo');
    ylabel('Estado');
    legend('show');
    grid on;

    % Subplot 2: terceiro e quarto estados (posição e velocidade de x2)
    % Calcular o sinal de referência
    r = zeros(1, length(outOde.x)); % Inicializa o vetor de referência
    for i = 1:length(outOde.x)
        r(i) = referencia_massa_mola(outOde.x(i));
    end

    subplot(3,1,2);
    plot(outOde.x, outOde.y(3,:), 'g', 'DisplayName', 'Posição x2');
```

```

hold on;
plot(outOde.x, outOde.y(4,:), 'm', 'DisplayName', 'Velocidade dx2/dt');
plot(outOde.x, r, 'k--', 'DisplayName', 'Referência');
title('Terceiro e Quarto Estados e Referência');
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;
max(abs(sinalU)) %printa valor de u barra
else
    % Faz o vídeo
    gravarVideo = 1;
    t = outOde.x;
    x1 = outOde.y(1,:);
    dx1 = outOde.y(2,:);
    x2 = outOde.y(3,:);
    dx2 = outOde.y(4,:);
    animateTracking(gravarVideo, t, x1, dx1, x2, dx2);
end
%
function animateTracking(gravarVideo,t,x1,dx1,x2,dx2)
global K;
fig = figure();
ori = [0 0];
tamCar1 = [1 0.5];
tamMolas = [0.7 0.7 0.7];
tamCar2 = [1 0.5];
maxDef=0;
scale = 0.1;
f=1;
while f <= size(x2,2)

    x1_ = scale*x1(f);
    x2_ = scale*x2(f);
    hold on;
    axis equal;
    axis([-1 6 -1 1]);

    desenhaCarros(ori,x1_,x2_,tamCar1,tamCar2,tamMolas);
    text(4.8,0.7,sprintf('x2 = %.2f\n',x2(f)));
    text(4.8,0.4,sprintf('t = %.2f\n',t(f)));
    text(-0.5,-0.9,sprintf('K = [%.1f %.1f %.1f %.1f %.1f]\n',K));
    hold off;
    if gravarVideo

```

```

        F{f} = getframe();
    else
        pause(0.1);
    end
    clf(fig);
    fprintf('frame=%d -> [x1 x2]=[%.3f %.3f]\n',f,x1_,x2_);
    f=f+1;
end
if gravarVideo
    geraVideo('videoMassaMola_saturado.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 desenhaCarros(ori,x1,x2,tamCar1,tamCar2,tamMolas)
%carro 1
p1 = [ori(1)+tamMolas(1)+x1 ori(2)];
p2 = [ori(1)+tamMolas(1)+x1 + tamCar1(1) ori(2)];
p3 = [ori(1)+tamMolas(1)+x1 + tamCar1(1) ori(2)+tamCar1(2)];
p4 = [ori(1)+tamMolas(1)+x1 ori(2)+tamCar1(2)];
pts = [p1;p2;p3;p4];
f=fill(pts(:,1),pts(:,2),'b');
%carro 2
r1 = ori(1)+tamMolas(1)+tamMolas(2)+x1+tamCar1(1)+x2;
r2 = ori(1)+tamMolas(1)+tamMolas(2)+x1+tamCar1(1)+tamCar2(1)+x2;
p1 = [r1 ori(2)];
p2 = [r2 ori(2)];
p3 = [r2 ori(2)+tamCar2(2)];
p4 = [r1 ori(2)+tamCar2(2)];
pts = [p1;p2;p3;p4];
f=fill(pts(:,1),pts(:,2),'b');
%mola 1

```

```

xa = ori(1);
ya = ori(2)+tamCar1(2)/2;
xb = ori(1)+tamMolas(1)+x1;
yb = ori(2)+tamCar1(2)/2;
ne = 5;
a = tamMolas(1);
ro = 0.1;
[xs,ys] = spring(xa,ya,xb,yb,ne,a,ro);
plot(xs,ys,'LineWidth',2,'Color','r')
%mola 2
xa = ori(1)+tamMolas(1)+tamCar1(1)+x1;
ya = ori(2)+tamCar1(2)/2;
xb = ori(1)+tamMolas(1)+tamMolas(2)+x1+tamCar1(1)+x2;
yb = ori(2)+tamCar1(2)/2;
ne = 5;
a = tamMolas(2);
[xs,ys] = spring(xa,ya,xb,yb,ne,a,ro);
plot(xs,ys,'LineWidth',2,'Color','r')
%mola 3
if tamMolas(3) ~= 0
    xa = ori(1)+tamMolas(1)+tamCar1(1)+x1 +tamMolas(2)+tamCar2(1)+x2;
    ya = ori(2)+tamCar1(2)/2;
    %xb = ori(1)+tamMolas(1)+tamCar1(1)+x1
    +tamMolas(2)+tamCar2(1)+x2+tamMolas(3);
    xb=ori(1)+tamMolas(1)+tamCar1(1)+tamMolas(2)+tamCar2(1)+tamMolas(3);
    yb = ori(2)+tamCar1(2)/2;
    ne = 5;
    a = tamMolas(2);
    [xs,ys] = spring(xa,ya,xb,yb,ne,a,ro);
    plot(xs,ys,'LineWidth',2,'Color','r')

    %fixacao da direita
    r=ori(1)+tamMolas(1)+tamCar1(1)+tamMolas(2)+tamCar2(1)+tamMolas(3);
    p1 = [r ori(2)];
    p2 = [r+0.1 ori(2)];
    p3 = [r+0.1 ori(2)+0.7];
    p4 = [r ori(2)+0.7];
    pts = [p1;p2;p3;p4];
    f=fill(pts(:,1),pts(:,2),'k');

end
%fixacao da esquerda
p1 = [ori(1)-0.1 ori(2)];
p2 = [ori(1) ori(2)];
p3 = [ori(1) ori(2)+0.7];
p4 = [ori(1)-0.1 ori(2)+0.7];
pts = [p1;p2;p3;p4];
f=fill(pts(:,1),pts(:,2),'k');
%pisso
p1 = [ori(1)-0.1 ori(2)];
p2 = [ori(1)+tamMolas(1)+tamCar1(1)+tamMolas(2)+tamCar2(1)+tamMolas(3),
ori(2)];
plot([p1(1) p2(1)],[p1(2) p2(2)],'LineWidth',0.5,'Color','k');

```

```

%
function dX = massaMola(t,x)
global A B C K;
r = referencia_massa_mola(t);
%informe a lei de controle (-K*x)
u=-K*x;
u_barra=68.8221;
u_sat=u_barra/2;
if u > u_sat
    u = u_sat;
elseif u < -u_sat
    u = -u_sat;
end

Acal = [A zeros(4,1);
        -C 0];
Bcal = [B;0];
dX = Acal*x + Bcal*u + [zeros(4,1);1]*r;

```

```

%
function r = referencia_massa_mola(t)
g=1.5;
if t < g
    r=1;
elseif t < 2*g
    r=-1;
elseif t < 3*g
    r=1;
elseif t < 4*g
    r=-1;
else
    r=0;
end

```

```

%
function status = saida(t, y, flag)
global sinalU K;
status = 0;
if strcmp(flag, 'init')
    % Inicialização
    u=-K*y(:,end);
    sinalU = [u];
elseif isempty(flag)
    % Atualização
    u=-K*y(:,end);
    sinalU = [sinalU; u];
elseif strcmp(flag, 'done')
    % Finalização
end

```