Table of Contents

Lista 8.

```
% Mauricio Garcia Di Mase - nUSP:12547152

close all %fecha todas janelas
clear all %limpa memoria
clc %limpa command window
```

Atividade 1

```
A = [3 0 0; 5 4 0; 1 2 3];
B = [0; 2; 5];
Ts = 0.1;

Ad = expm(A*Ts)
Bd = A\(Ad - eye(size(Ad)))*B

[Ad2,Bd2]=c2d(A,B,Ts)

Ad == Ad2
Bd == Bd2

% vemos aqui que os valores das matrizes discretizadas pelos
 diferentes
% métodos são quase idênticas, salvo o segundo elemento da matriz B
```

```
Ad =

    1.3499         0         0
    0.7098    1.4918         0
    0.2048    0.2839    1.3499


Bd =

         0
    0.2459
    0.6084
```

```
Ad2 =

    1.3499         0         0
    0.7098    1.4918         0
    0.2048    0.2839    1.3499


Bd2 =

         0
    0.2459
    0.6084


ans =

  3×3 logical array

   1   1   1
   1   1   1
   1   1   1


ans =

  3×1 logical array

   1
   0
   1
```

Atividade 2

```
close all
clear all
clc

Ad=[0.75 -0.25 4; 2.25 -0.75 0; -0.375 0.125 3.5];

eig(Ad) %obtem autovalores de Ad

if any(abs(eig(Ad))>1) %verifica se existe algum autovalor maior que 1
    disp('Sistema não é assintoticamente estavel!')
else
    disp('Sistema é assintoticamente estavel!')
end

Q = eye(size(Ad))
P = dlyap(Ad,Q) %obtem solucao da eq. de lyapunov
```

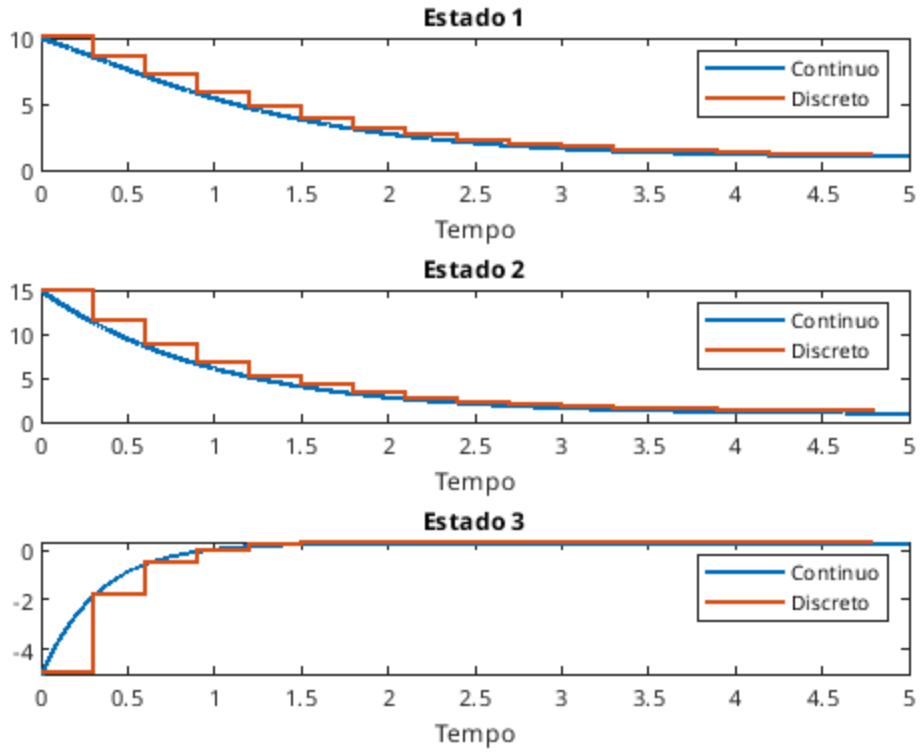
```
if any(eig(P)<0) %verifica se P>0
    disp('Sistema não é assintoticamente estavel!')
else
    disp('Sistema é assintoticamente estavel!')
end
```

```
ans =

    0.0000
    0.5000
    3.0000

Sistema não é assintoticamente estavel!

Q =

     1     0     0
     0     1     0
     0     0     1


P =

   58.0885   76.4531   14.9115
   76.4531   84.6719   28.5469
   14.9115   28.5469    0.9635

Sistema não é assintoticamente estavel!
```

Atividade 3

```
close all
clear all
clc

A = [-2 1 0; 0 -1 0; 0 0 -3];
B = [1; 1; 1];

Ts = 0.3;
T = 5;
td = 0:Ts:T;
dt = 0.001;
t = 0:dt:T;

[Ac,Bc] = c2d(A,B,dt); %modelo tempo continuo (aproximacao)
[Ad,Bd] = c2d(A,B,Ts); %modelo em tempo discreto

Nd = numel(td);
Nc = numel(t);
n = size(Ad,1);
x = zeros(n,Nc);
xd = zeros(n,Nd);
```

```
u = ones(Nd,1);
xd(:,1) = [10; 15; -5];

x(:,1) = xd(:,1);

Nr = Ts/dt;
kd = 1;
for k=1:Nc-1
%simula sistema em tempo continuo
    x(:,k+1)=Ac*x(:,k)+Bc*u(kd);
%simula sistema em tempo discreto
if mod(k,Nr)==0 && kd<numel(td)
    xd(:,kd+1)=Ad*xd(:,kd)+Bd*u(kd);
    kd=kd+1;
end

end

figure

subplot(3,1,1)
plot(t,x(1,:),'LineWidth',1.5)
hold on
stairs(td,xd(1,:),'LineWidth',1.5)
xlabel('Tempo')
legend('Continuo','Discreto')
title('Estado 1')

subplot(3,1,2)
plot(t,x(2,:),'LineWidth',1.5)
hold on
stairs(td,xd(2,:),'LineWidth',1.5)
xlabel('Tempo')
legend('Continuo','Discreto')
title('Estado 2')

subplot(3,1,3)
plot(t,x(3,:),'LineWidth',1.5)
hold on
stairs(td,xd(3,:),'LineWidth',1.5)
xlabel('Tempo')
legend('Continuo','Discreto')
title('Estado 3')
```



Atividade 4

```
close all
clear all
clc


M=1;
m=0.1;
l=0.4;
g=9.81;

syms F
syms th th_d th_dd
syms x x_d x_dd

eq1=(M+m)*x_dd-m*l*th_dd*cos(th)+m*l*th_d^2*sin(th)-F;
eq2=l*th_dd-x_dd*cos(th)-g*sin(th);
u=F;
S=solve(eq1==0, eq2==0, x_dd, th_dd)

x_vet=[x;x_d;th;th_d];
x_vet_dot=[x_d; S.x_dd; th_d; S.th_dd];

A=simplify(jacobian(x_vet_dot,x_vet))
```

```
B=simplify(jacobian(x_vet_dot,u))
```

```
S =

  struct with fields:

     x_dd: [1×1 sym]
    th_dd: [1×1 sym]
```

```
A =

[0, 1,

                                                                    0,
                                        0]
[0, 0,                                        (40*th_d^2*cos(th) -
 1962*cos(th)^2 + 981)/(100*(cos(th)^2 - 11)) - (cos(th)*sin(th)*(-
 40*sin(th)*th_d^2 + 1000*F + 981*cos(th)*sin(th)))/(50*(cos(th)^2 -
 11)^2),    -(4*th_d*sin(th))/(5*(sin(th)^2 + 10))]
[0, 0,

                                                                    0,
                                        1]
[0, 0, - (10791*cos(th) - 80*th_d^2*cos(th)^2 - 1000*F*sin(th)
 + 40*th_d^2)/(40*(cos(th)^2 - 11)) - (cos(th)*sin(th)*(-
 40*cos(th)*sin(th)*th_d^2 + 10791*sin(th) + 1000*F*cos(th)))/
(20*(cos(th)^2 - 11)^2), (2*th_d*cos(th)*sin(th))/(cos(th)^2 - 11)]
```

```
B =

                             0
          -10/(cos(th)^2 - 11)
                             0
-(25*cos(th))/(cos(th)^2 - 11)
```

ponto de operação

```
x=0;
x_d=0;
th=0;
th_d=0;
u=0;
F=u
A0 = simplify(subs(A));
B0 = simplify(subs(B));
A0 = double(A0)
B0 = double(B0)

%Discretização e verificação de estabilidade de Lyapunov
```

```
Ts = 0.1;

[Ad Bd] = c2d(A0,B0, Ts);

eig(Ad) %obtem autovalores de Ad
if any(abs(eig(Ad))>=1) %verifica se existe algum autovalor maior que
 1
    disp('Sistema nao e assintoticamente estavel nesse ponto de
 operação!')
else
    disp('Sistema e assintoticamente estavel nesse ponto de
 operação!')
end
```

```
F =

     0


A0 =

         0    1.0000         0         0
         0         0    0.9810         0
         0         0         0    1.0000
         0         0   26.9775         0


B0 =

         0
    1.0000
         0
    2.5000


ans =

    1.0000
    1.0000
    1.6810
    0.5949

Sistema nao e assintoticamente estavel nesse ponto de operação!
```

segundo ponto de operação

```
x=-1;
x_d=0;
th=pi/2;
th_d=0;
u=0;
F=u;
```

```
A1 = simplify(subs(A));
B1 = simplify(subs(B));
A1 = double(A1)
B1 = double(B1)

%Discretização e verificação de estabilidade de Lyapunov
Ts = 0.1;

[Ad Bd] = c2d(A1,B1, Ts);

eig(Ad) %obtem autovalores de Ad
if any(abs(eig(Ad))>=1) %verifica se existe algum autovalor maior que
 1
    disp('Sistema nao e assintoticamente estavel nesse ponto de
 operação!')
else
    disp('Sistema e assintoticamente estavel nesse ponto de
 operação!')
end
```

```
A1 =

         0    1.0000         0         0
         0         0   -0.8918         0
         0         0         0    1.0000
         0         0         0         0


B1 =

         0
    0.9091
         0
         0


ans =

     1
     1
     1
     1

Sistema nao e assintoticamente estavel nesse ponto de operação!
```

Atividade 5

```
close all
clear all
clc

syms m M l F g
```

```
syms th th_d th_dd
syms x x_d x_dd

eq1=(M+m)*x_dd-m*l*th_dd*cos(th)+m*l*th_d^2*sin(th)-F;
eq2=l*th_dd-x_dd*cos(th)-g*sin(th);
u=F;
S=solve(eq1==0, eq2==0, x_dd, th_dd)

x_vet=[x;x_d;th;th_d];
x_vet_dot=[x_d; S.x_dd; th_d; S.th_dd];

A=simplify(jacobian(x_vet_dot,x_vet))
B=simplify(jacobian(x_vet_dot,u))

% ponto de operacao
x=0;
x_d=0;
th=0;
th_d=0;
u=0;

M=1;
m=0.1;
l=0.4;
g=9.81;
A0=double(simplify(subs(A))) %matriz dinamica
B0=double(simplify(subs(B))) %matriz de entrada

Ts=0.1; %tempo de amostragem
[Ad,Bd]=c2d(A0,B0,Ts)
K=dlqr(Ad,Bd,eye(4),1); %Ganho otimo de realimentacao de estados

T=20; %tempo de simulacao
dt=0.0001;
t=0:dt:T;
Nc=numel(t);
td=0:Ts:T;
Nd=numel(td);

x=zeros(4,Nc);
u=zeros(Nd,1);

Nr=Ts/dt;
kd=1;
fig = figure;
fig.Position = [0 0 1200 500];


S =

  struct with fields:

     x_dd: [1×1 sym]
    th_dd: [1×1 sym]
```

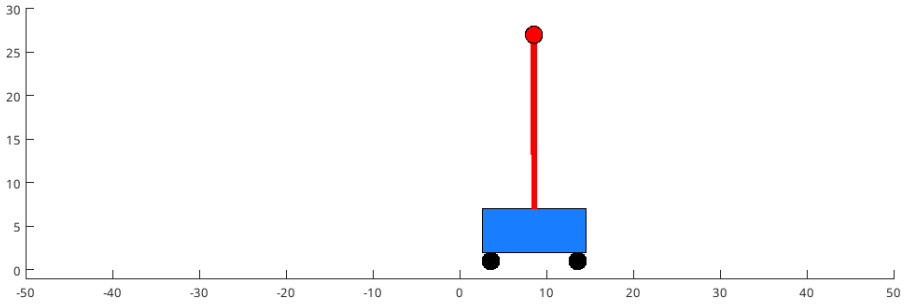
```
A =

[0, 1,


                                    0,
          0]
[0, 0,
                - (m*(g - 2*g*cos(th)^2 + l*th_d^2*cos(th)))/(-
 m*cos(th)^2 + M + m) - (2*m*cos(th)*sin(th)*(- l*m*sin(th)*th_d^2
 + F + g*m*cos(th)*sin(th)))/(- m*cos(th)^2 + M + m)^2,       -
(2*l*m*th_d*sin(th))/(m*sin(th)^2 + M)]
[0, 0,


                                    0,
          1]
[0, 0, - (2*(l*m*(2*cos(th)^2 - 1)*th_d^2 + F*sin(th) -
 g*m*cos(th) - M*g*cos(th)))/(l*(2*M + m - m*(2*cos(th)^2 - 1)))
 - (2*m*cos(th)*sin(th)*(- l*m*cos(th)*sin(th)*th_d^2 + F*cos(th)
 + g*m*sin(th) + M*g*sin(th)))/(l*(- m*cos(th)^2 + M + m)^2), -
(2*m*th_d*sin(2*th))/(2*M + m - m*cos(2*th))]


B =

                                     0
          1/(- m*cos(th)^2 + M + m)
                                     0
cos(th)/(l*(- m*cos(th)^2 + M + m))


A0 =

         0    1.0000         0         0
         0         0    0.9810         0
         0         0         0    1.0000
         0         0   26.9775         0


B0 =

         0
    1.0000
         0
    2.5000


Ad =

    1.0000    0.1000    0.0050    0.0002
         0    1.0000    0.1026    0.0050
```

```
         0         0    1.1379    0.1046
         0         0    2.8207    1.1379


Bd =

    0.0050
    0.1004
    0.0128
    0.2614
```

Parametros PID

```
Kp=12;
Kd=2;
Ki=0.1;

Ti=Kp/Ki;
Td=Kd/Kp;

erro_anterior=0;
ui=0;
H=[1 0 0 0;...
   0 0 1 0];
%%condicao inicial
x(:,1)=[0 0 deg2rad(10) 0]';
pendulo=plotPendulo(x(:,1),fig);
for k=1:Nc-1
    %%simula modelo nao-linear pendulo em tempo continuo
    x(:,k+1)=pendulo_model(x(:,k),u(kd),dt,M,m,l);
    if (mod(k,Nr)==0 || k==1) && kd<=numel(td)
        %%simula controlador em tempo discreto
        if k~=1
            kd=kd+1;
        end
        y=H*x(:,k);
```

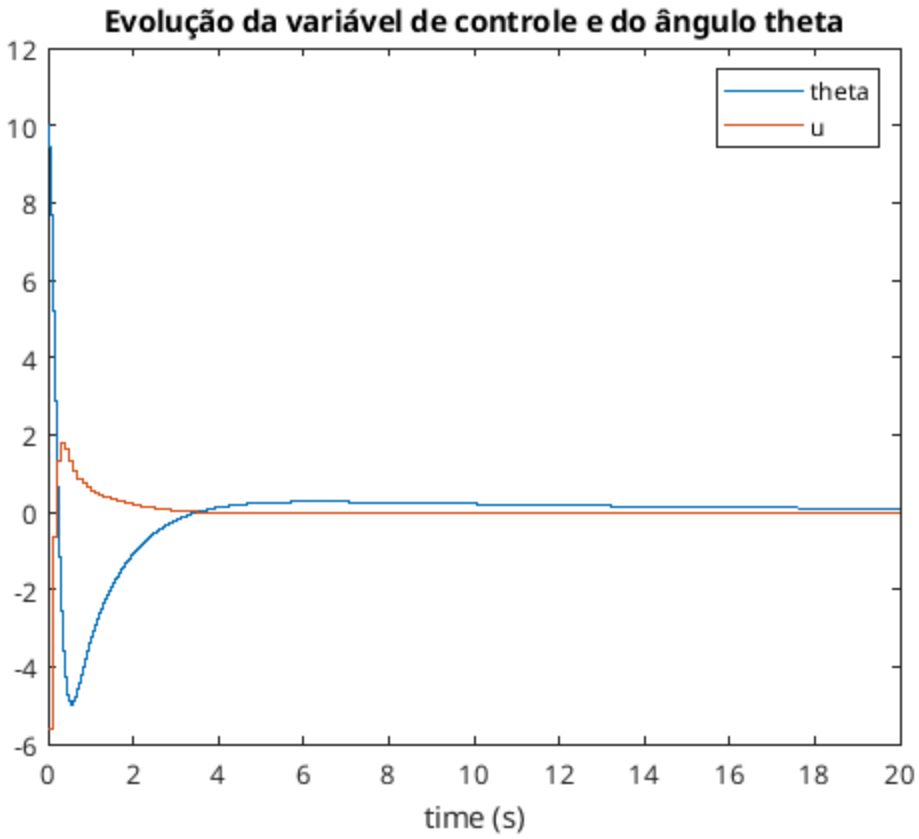
```
        px=y(1); %posicao na direcao x
        theta=y(2); %posicao angular
        %%calcula sinal de erro
        erro=0-theta; %erro de posicao angular
        %%Controle PID
        up=Kp*erro; %proporcional
        ud=Kp*Td/Ts*(erro-erro_anterior); %derivativo (euler-backward)
        ui=ui+Kp*Ts/Ti*erro; %integrativo (euler-backward)
        u(kd)=up+ud+ui;
        erro_anterior=erro;
        %%LQR
        %u(kd)=-K*x(:,k);
        %%atualiza frame da animacao
        updatePendulo(x(:,k),pendulo);
        pause(0.05)
    end
end
```

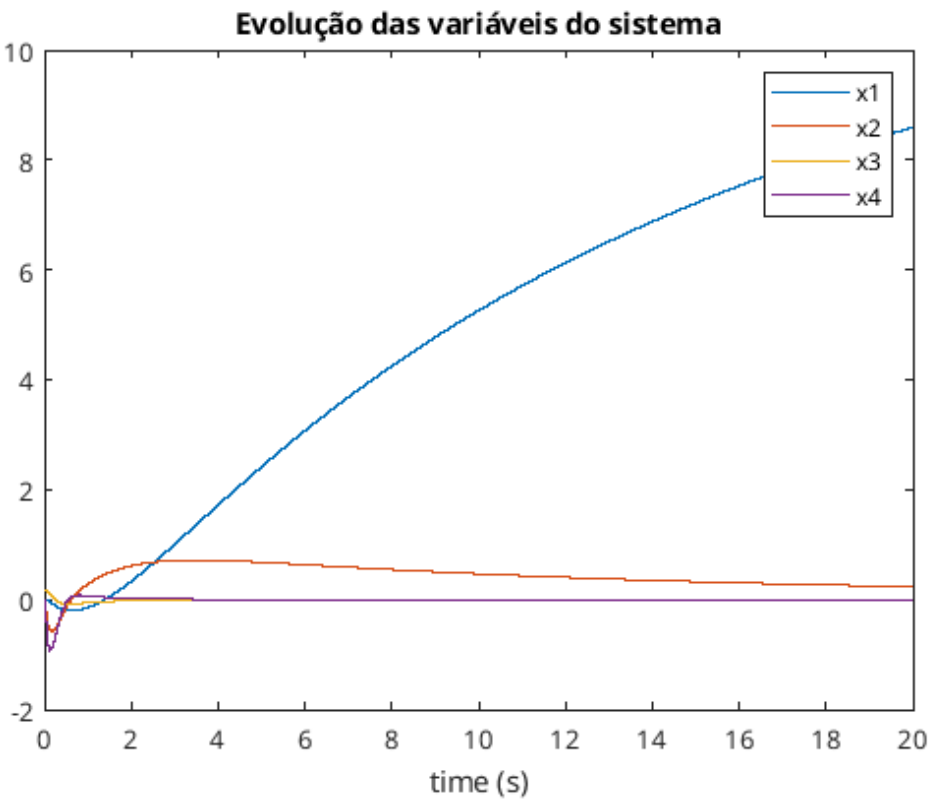



```
disp('Sistema de controle pôde ser sintonizado com os seguintes
 valores:')
Kp
Ti
Td
figure
plot(t,rad2deg(x(3,:)))
hold on
stairs(td,u)
legend('theta','u')
xlabel('time (s)')
title('Evolução da variável de controle e do ângulo theta')
figure
plot(t,x')
legend('x1','x2','x3','x4')
xlabel('time (s)')
title('Evolução das variáveis do sistema')
```

Sistema de controle pôde ser sintonizado com os seguintes valores:

```
Kp =

    12

Ti =

   120

Td =

    0.1667
```





Published with MATLAB® R2021a