Laboratório 8

SEL0359 - Controle Digital - 2° Semestre de 2024

Prof. Marcos R. Fernandes

Espaço de estados - parte 1

Entrega:[1] entregue um arquivo PDF com os gráficos e código fonte utilizado para cada questão além da resposta para as perguntas dos enunciados. Não esqueça de colocar título na figura para identificar o que cada figura representa, nome dos eixos, e legenda.

O objetivo dessa aula de laboratório é explorar recursos computacionais do Matlab para análise de modelos de espaço de estados em tempo discreto.

Dica: Confira sempre o help do matlab para cada comando antes de utiliza-lo!

- expm: calcula exponencial matricial
- c2d: converte modelo tempo contínuo em tempo discreto;
- eye: constrói matriz identidade;
- dlyap: calcula solucao da equação de Lyapunov em tempo discreto;

1 Atividade 1

Nessa atividade, o objetivo explorar os recursos computacionais do Matlab para obter modelos em espaço de estados equivalentes em tempo discreto.

Para isso, considere o modelo em espaço de estados em tempo contínuo a seguir

$$\dot{x}(t) = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 3 \end{bmatrix} u(t)$$

O modelo em espaço de estados equivalente em tempo discreto é dado por

$$x_{k+1} = A_d x_k + B_d u_k$$

em que

$$A_d = e^{AT_s} \tag{1}$$

$$B_d = \int_0^{T_s} e^{A\eta} d\eta B \tag{2}$$

[1] Última atualização: 17/10/2024

no qual T_s é o tempo de amostragem. No exemplo acima, a matriz A é não-singular, portanto, existe A^{-1} e pode-se obter B_d na forma

$$B_d = A^{-1}(e^{AT_s} - I)B.$$

O modelo equivalente em tempo discreto pode ser obtido aplicando as equações acima ou através da função do Matlab **c2d**.

O código Matlab a seguir obtém o modelo discretizado equivalente, para $T_s = 0.1s$, usando as equações acima e também com a função **c2d**.

```
close all
clear all
clc
%%
A=[1 1;...
   0 2];
B=[1;...
   3];
%%
Ts=0.1;
%%
Ad=expm(A*Ts)
Bd=A\(Ad-eye(size(Ad)))*B
%%
[Ad2,Bd2]=c2d(A,B,Ts)
```

Usando a mesma lógica, obtenha o modelo em tempo discreto, com $T_s = 0.5s$, para o sistema:

$$\dot{x}(t) = \begin{bmatrix} 3 & 0 & 0 \\ 5 & 4 & 0 \\ 1 & 2 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \\ 5 \end{bmatrix} u(t)$$

Compare os resultados das equações de discretização apresentadas acima com os resultados da função **c2d**.

2 Atividade 2

Nessa atividade, o objetivo explorar os recursos computacionais do Matlab para avaliar a estabilidade assintótica de modelos em espaço de estados em tempo discreto.

Para isso, considere o seguinte sistema como exemplo:

$$x_{k+1} = \underbrace{\begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}}_{A_d} x_k$$

Um modelo em espaço de estados em tempo discreto é assintoticamente estável se os autovalores da matriz de dinâmica A_d estiverem dentro do circulo unitário:

$$|\lambda\{A_d\}| < 1.$$

Outra forma de avaliar a estabilidade do sistema é através da equação de Lyapunov. Se existir $P \in \mathbb{R}^{n\times n}$ tal que

$$A_d^{\intercal} P A_d - P = -Q, \quad Q > 0, P > 0$$

então o sistema $x_{k+1} = A_d x_k$ é assintoticamente estável. A matriz $Q > 0$ é arbitrária.

No Matlab, pode-se utilizar o comando **eig** para obter os autovalores de A_d e a função **dlyap** para encontrar a solução da equação de Lyapunov.

O código Matlab a seguir avalia a estabilidade do sistema.

```
close all
clear all
clc
%%
Ad=[1 -2;...
    2 1];
Bd=[1;...
    2];
%%
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!')
else
    disp('Sistema e 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 nao e assintoticamente estavel!')
else
    disp('Sistema e assintoticamente estavel!')
end
```

Usando a mesma lógica, avalie a estabilidade assintótica do sistema:

$$x_{k+1} = \underbrace{\begin{bmatrix} 0.75 & -0.25 & 4 \\ 2.25 & -0.75 & 0 \\ -0.375 & 0.125 & 3.5 \end{bmatrix}}_{A_d} x_k$$

3 Atividade 3

Nessa atividade, o objetivo explorar os recursos computacionais do Matlab para simular modelos em espaço de estados em tempo discreto.

Considere o sistema em tempo contínuo dado por

$$\dot{x}(t) = \begin{bmatrix} -2 & 1 \\ 0 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

O seguinte código em Matlab simula esse sistema para $T_s = 0.2s$ com entrada degrau unitário e compara com o sistema em tempo contínuo (amostragem bem pequena).

```
close all
clear all
clc
%%
A=[-2 1;...
   0 -1];
B=[1;...
   1];
%%
Ts=0.2;
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;-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(2,1,1)
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(2,1,2)
plot(t,x(2,:),'LineWidth',1.5)
hold on
stairs(td,xd(2,:),'LineWidth',1.5)
xlabel('Tempo')
legend('Continuo','Discreto')
title('Estado 2')
```

Seguindo a mesma lógica simule o sistema a seguir com entrada degrau unitário e $T_s = 0.3s$. Considere $x(0) = [10\ 15\ \ -5]^\intercal$.

$$\dot{x}(t) = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -3 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u(t)$$

4 Atividade 4

Nessa atividade, o objetivo explorar os recursos simbólico do Matlab para obter modelos linearizados para plantas não-lineares em tempo discreto.

Nesse exemplo, será utilizado o modelo de um pêndulo invertido conforme ilustrado na Figura 1. As EDO que descrevem o comportamento dessa planta são dadas a seguir:

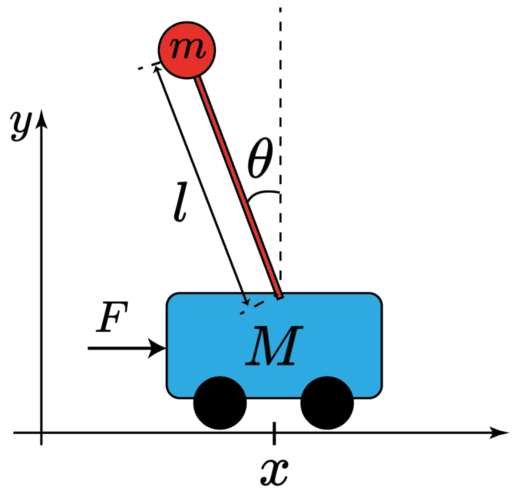

Figure 1: Pêndulo invertido.

$$(M + m)\ddot{x} - ml\ddot{\theta}\cos(\theta) + ml\dot{\theta}^2\sin(\theta) = F, \tag{3}$$

$$l\ddot{\theta} - \ddot{x}\cos(\theta) - g\sin(\theta) = 0, \tag{4}$$

em que o vetor de estados é dado por

$$x = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}, \quad \begin{cases} x \rightarrow \text{posição} \\ \dot{x} \rightarrow \text{velocidade} \\ \theta \rightarrow \text{posição angular} \\ \dot{\theta} \rightarrow \text{velocidade angular} \end{cases}$$

e o sinal de entrada é a força que atua na plataforma móvel

$$u = F.$$

Suponha que esteja disponível medições da posição x e da posição angular θ. Assim, a equação de saída desse sistema é dada por

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x.$$

Adotando,

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad u_0 = 0,$$

como ponto de operação do sistema (posição vertical com velocidade zero), busca-se um modelo linearizado na forma

$$\dot{x} = A(x_0, u_0)x + B(x_0, u_0)u,$$

válido para regiões próximas do ponto (x_0, u_0).

O código Matlab a seguir obtém as matrizes $A(x_0, u_0)$ e $B(x_0, u_0)$ do modelo linearizado para esse sistema.

```
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;
A0=simplify(subs(A))
B0=simplify(subs(B))
%%
```

```
M=1;
m=0.1;
l=0.4;
g=9.81;
A0=double(simplify(subs(A)))
B0=double(simplify(subs(B)))
```

Após obter o modelo linearizado, pode-se utilizar o comando **c2d** para obter o equivalente em tempo discreto.

Usando a lógica do código acima, obtenha as matrizes do modelo linearizado para o pêndulo invertido em tempo discreto com $T_s = 0.1s$ nos seguintes pontos de operação:

$$x_0 = \begin{bmatrix} -1 \\ 0 \\ \pi \\ 0 \end{bmatrix}, u_0 = 0$$

e

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, u_0 = 0$$

Em seguida, avalie a estabilidade assintótica do modelo linearizado, usando a equação de Lyapunov, nos dois pontos de operação. Considere

$$\begin{cases} M = 1 \\ m = 0.1 \\ l = 0.4 \\ g = 9.81 \end{cases}$$

O sistema linearizado é assintoticamente estável nos dois pontos de operação?

5 Atividade 5

Usando o modelo da atividade 4 do pêndulo invertido, faça a sintonia de um controlador do tipo PID para controlar o pêndulo invertido na posição vertical com $\theta = 0$.

Use o código de exemplo disponível no e-disciplina (arquivo **controle-pendulo-invertido.zip**).

No arquivo zip terão 4 scripts. Um para o controle, outro para o modelo do sistema e duas funções auxilares para gerar uma animação do pêndulo invertido. Lembre de deixar os quatro arquivos na mesma pasta para o Matlab.

Após obter sua melhor sintonia do controle PID, compare com o desempenho do controle LQR (descomente a linha 84).

Explique com suas palavras quais diferenças podem ser observadas entre os dois tipos de controladores.