Laboratório 4

SEL0359 - Controle Digital - 2° Semestre de 2024

Prof. Marcos R. Fernandes

Modelagem e Identificação de Sistemas

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 diferentes métodos de discretização para modelagem de sistemas dinâmicos em tempo discreto.

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

1 Atividade 1

Nessa atividade, o objetivo é comparar diferentes métodos discretização.

Para isso, considere o seguinte exemplo:

$$G(s) = \frac{s+3}{(s+1)(s+2)}$$

A função de transferência equivalente em tempo discreto pode ser computado usando o Matlab através da função **c2d** para os métodos **degrau-invariante** (zoh), **impulso-invariante** (impulse), **mapeamento casado de pólos e zeros** (matched), método **Bilinear** (tustin) além de outros. (Dica: consulte a documentação do matlab sobre discretização)

https://www.mathworks.com/help/control/ug/continuous-discrete-conversion-methods.html

Por exemplo, o código a seguir obtém a FT em tempo discreto usando o método degrau-invariante, impulso-invariante, mapeamento casado de pólos e zeros e método de tustin:

[1] Ultima atualização: 12/09/2024

```
close all %fecha todas janelas
clear all %limpa memoria
clc %limpa command window
%%
num=[1 3];
den=conv([1 1],[1 2]);
G=tf(num,den); %FT em tempo continuo
T=0.1;%tempo de amostragem
Gd1=c2d(G,T,'zoh') %obtem a FT discreta usando metodo degrau-invariante
Gd2=c2d(G,T,'impulse') %obtem a FT discreta usando metodo impulso-invariante
Gd3=c2d(G,T,'matched') %obtem a FT discreta usando metodo map. casado ...
    polos/zeros
Gd4=c2d(G,T,'tustin') %obtem a FT discreta usando metodo bilinear (tustin)
```

Após obter a FT discretizada, pode-se comparar a resposta em frequência resultante de cada método de discretização usando o comando **bode**:

```
figure
bode(G,Gd1)
title('ZOH')
figure
bode(G,Gd2)
title('Impulse')
figure
bode(G,Gd3)
title('Matched')
figure
bode(G,Gd4)
title('tustin')
```

Seguindo a mesma lógica, encontre a FT discreta usando a função **c2d** e compare a resposta em frequência para os casos:

1. $G(s) = \frac{s}{(s+a)^2}$, em que a é uma constante de sua escolha.
2. $G(s) = \frac{s^2+s+1}{s^3+2s^2+3s+2}$
3. $G(s) = \frac{s}{s^4+3s^2+2s+1}$

O que pode-se concluir através da análise da resposta em frequência para os diferentes métodos de discretização?

2 Atividade 2

Nessa atividade, o objetivo é usar o recurso de cálculo simbólico do Matlab para obter as FT equivalentes em tempo discreto. Para isso, considere o seguinte exemplo:

$$G(s) = \frac{a}{s+a}$$

Para o caso do método de Euler-Backward de discretização, pode-se implementar conforme código de exemplo a seguir:

```
close all %fecha todas janelas
clear all %limpa memoria
clc %limpa command window
%%
syms s z T %define variaveis simbolicas
G=a/(s+a); %monta GT em tempo continuo
Gd=subs(G,'s',(z-1)/(z*T)) %obtem FT discreta via subs
pretty(Gd) %mostra FT de forma "bonita"
```

Seguindo a mesma lógica, encontre a FT em tempo discreto usando os métodos de Euler-Forward e Euler-Backward dos seguintes casos:

1. $G(s) = \frac{1}{s^2}$
2. $G(s) = \frac{s+2}{(s+1)(s+5)}$
3. $G(s) = \frac{s+3}{s^2(s+1)(s-9)}$

3 Atividade 3

Nessa atividade, o objetivo é comparar o método de discretização impulso-invariante e degrau-invariante. Para isso, considere a FT a seguir como exemplo:

$$G(s) = \frac{s+3}{s^2+s+1}.$$

Usando o seguinte código, obtém-se a resposta ao degrau e ao impulso tanto em tempo contínuo quanto em tempo discreto:

```
close all
clear all
clc
G=tf([1 3],[1 1 1]); %FT em tempo continuo
dt=0.001; %amostragem continuo
```

```
t=0:dt:10; %vetor de tempo continuo
[yu,t]=step(G,t); %resposta ao degrau em tempo continuo
[yi,t]=impulse(G,t); %resposta ao impulso em tempo continuo
T=0.5;%tempo de amostragem discreta
Gd1=c2d(G,T,'zoh') %obtem a FT discreta usando metodo degrau-invariante
Gd2=c2d(G,T,'impulse') %obtem a FT discreta usando metodo impulso-invariante
td=0:T:10; %vetor de tempo discreto
[ydu1,td]=step(Gd1,td); %resposta ao degrau
[ydu2,td]=step(Gd2,td); %resposta ao degrau
[ydi1,td]=impulse(Gd1,td); %resposta ao impulso
[ydi2,td]=impulse(Gd2,td); %resposta ao impulso
figure
plot(t,yu,'LineWidth',1.5)
hold on
stairs(td,ydu1,'LineWidth',1.5)
stairs(td,ydu2,'LineWidth',1.5)
figure
plot(t,yi,'LineWidth',1.5)
hold on
stairs(td,ydi1,'LineWidth',1.5)
stairs(td,ydi2,'LineWidth',1.5)
```

Seguindo a mesma lógica, compare os métodos de discretização degrau-invariante e impulso-invariante para a seguinte função de transferência com $T = 0.1s$ e $t \in [0, 5]$:

$$G(s) = \frac{(s+2)(s+1)}{(s+5)(s+3)(s+7)}$$

O que pode-se observar em relação as respostas temporais dos dois métodos de discretização para as entradas em degrau e impulso?

4 Atividade 4

O objetivo dessa prática é obter a equação à diferenças para uma dada equação diferencial fazendo discretização usando os recursos do Matlab.

Para isso, considere uma equação de primeira ordem na forma

$$\frac{dy}{dt} + y = u(t).$$

A função de transferência associada a essa EDO é

$$G(s) = \frac{1}{s+1}$$

Pode-se obter a FT equivalente em tempo discreto usando o método de degrau-invariante no Matlab através do seguinte código:

```
close all %fecha todas janelas
clear all %limpa memoria
clc %limpa command window
%%
G=tf(1,[1 1]); %FT em tempo continuo
T=0.1;%tempo de amostragem
Gd1=c2d(G,T,'zoh') %obtem a FT discreta usando metodo degrau-invariante
```

O que resulta em

$$G_d(z) = \frac{0.09516}{z - 0.9048} = \frac{0.09516z^{-1}}{1 - 0.9048z^{-1}}$$

Portanto, como $G_d(z) = \frac{Y_d(z)}{U_d(z)}$, tem-se a equação à diferenças associada dada por

$$y[k] = 0.9048y[k-1] + 0.09516u[k-1]$$

Além do mais, essa equação à diferenças pode ser implementada usando estrutura de repetição **for** conforme exemplo a seguir:

```
td=0:T:5-T;
y=zeros(size(td));
u=ones(size(td));
for k=2:50
    y(k)=0.9048*y(k-1) + 0.09516*u(k-1);
end
figure
stairs(td,u,'LineWidth',1.5)
hold on
stairs(td,y,'LineWidth',1.5)
```

Seguindo a mesma lógica, encontre as equações à diferenças associadas as seguintes EDO e plote às 50 primeiras amostras em tempo discreto. Considere $T = 0.1s$.

1. $2\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 5y = \frac{du}{dt} + 2u, \quad y(0) = 0, \frac{dy(0)}{dt} = 0.$
2. $\frac{d^2y}{dt^2} + y = \frac{du}{dt}, \quad y(0) = 1, \frac{dy(0)}{dt} = 0.$

5 Atividade 5

Considere o circuito RC apresentado na Figura 1 cuja representação em diagrama de blocos é apresentada na Figura 2.

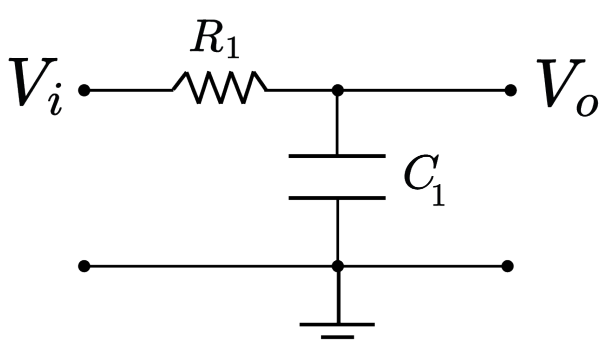

Figure 1: Circuito RC.

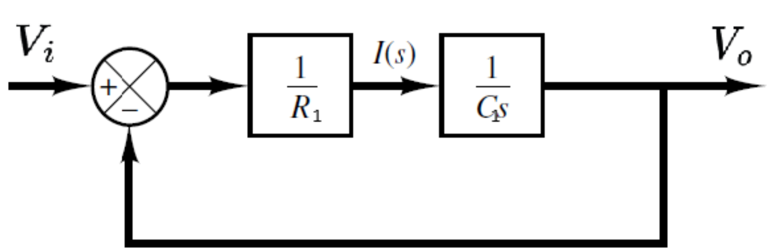

Figure 2: Circuito RC em diagrama de blocos.

1. Obtenha a função de transferência para o circuito RC $G(s) = \frac{V_o(s)}{V_i(s)}$;
2. Discretize a função de transferência $G(s)$ usando o método de Tustin com período de amostragem $T = 0.01s$ e $R = 1$ e $C = 1$;
3. Compare a resposta ao degrau do sistema contínuo e discreto;
4. Compare a resposta ao impulso do sistema contínuo e discreto;
5. Compare a resposta em frequência do sistema contínuo e discreto;
6. Usando FT discretizada $G_d(z)$ obtenha a respectiva equação à diferenças;
7. Usando a equação à diferenças simule uma entrada PRBS (exemplo de sinal PRBS está disponível no e-disciplina);

6 Atividade 6 (Opcional)

O objetivo dessa atividade é mostrar como fazer a identificação paramétrica de um sistema dado um conjunto de dados de entrada e saída.

Por exemplo, considere o sistema do exercício anterior (circuito RC). A equação à diferenças tem o seguinte formato

$$y[k] = \theta_1 y[k-1] + \theta_2 u[k-1] + \theta_3 u[k]$$

que pode ser colocada de forma vetorial resultando em

$$y[k] = \begin{bmatrix} y[k-1] & u[k-1] & u[k] \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Portanto, dado um conjunto de entradas $\{u[1], u[2], \ldots, u[N]\}$ e saídas $\{y[1], y[2], \ldots, y[N]\}$, pode-se construir a representação matricial dos sinais de entrada e sáida conforme a seguir

$$\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[N] \end{bmatrix} = \begin{bmatrix} y[0] & u[0] & u[1] \\ y[1] & u[1] & u[2] \\ \vdots & \vdots & \vdots \\ y[N-1] & u[N-1] & u[N] \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$
$$\Rightarrow y = H\theta + \varepsilon$$

Assim, supondo $rank(H) = n$, pode-se aplicar um Estimador de Mínimos Erros Quadráticos

(EMQ) para obter a estimativa dos parâmetros do modelo:

$$\hat{\theta} = (H^\intercal H)^{-1} y$$

Em Matlab, pode-se implementar essa estratégia de identificação conforme o código de exemplo a seguir:

```
close all %fecha todas janelas
clear all %limpa memoria
clc %limpa command window
load('ident_RC.mat');
figure
stairs(t,u) %visualiza sinal de entrada (PRBS)
title('Entrada')
figure
stairs(t,y) %visualiza sinal de saida (resposta ao PRBS)
title('Saida')
%% constroi matriz H
H=[0 y(1:end-1)';...
    0 u(1:end-1)';...
    u']';
rank(H) %verifica rank de H
theta=(H'*H)\H'*y %obtem estimativa dos parametros do modelo
```

Dessa forma, o modelo identificado é dado por

$$y[k] = \hat{\theta}_1 y[k-1] + \hat{\theta}_2 u[k-1] + \hat{\theta}_3 u[k]$$

O arquivo **identRC.mat** está disponível no e-disciplina.