

Exercicio

Desenvolva um script no MATLAB que calcule a Matriz de Transição de estados de um circuito RLC serie, bem com a tensão no capacitor e a corrente no indutor.

```

close all
clear all
clc

fprintf('Este programa visa calcular a matriz de transição de estados de
um\n sistema RLC série, assim como a tensão no capacitor e a corrente no
indutor.\n\n')

prompt = 'Qual o valor do Resistor?';
R = input(prompt)
prompt = 'Qual o valor do Indutor?';
L = input(prompt)
prompt = 'Qual o valor do Capacitor?';
C = input(prompt)
prompt = 'Qual o valor inicial da corrente no indutor?';
I11 = input(prompt)
prompt = 'Qual o valor inicial da tensão no capacitor?';
Vc1 = input(prompt)
prompt = 'Qual o valor da fonte de tensão?';
Vs = input(prompt)

```

Recebe, do usuário, os valores dos componentes eletrônicos e das condições iniciais variáveis de estado

%gera as matrizes de estados REPRESENTAÇÃO NO ESPAÇO DE ESTADOS

A=[-R/L -1/L; 1/C 0]

B=[Vs/L; 0]

C=[0 1]

D=[0]

%condição inicial, assume os valores das variáveis de estado

X0=[I11; Vc1]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} Vs/L \\ 0 \end{bmatrix} u(t)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u(t)$$

define a condição inicial de acordo c/ as variáveis de estado

$$x_1 = I_{11} \quad x_2 = V_{c1}$$

%calcula o autovalor da matriz A

lambda = eig(A)

$$\lambda_1 \text{ e } \lambda_2, \det[A - \lambda I] = 0$$

%declara os coeficientes da matriz de transição e^At

syms a0 a1

%declara t como variavel tempo, tau, e coeficiente em relação a tau

syms t tau a0tau altau

%compara os lambdas

if (lambda(1) ~= lambda(2))

→ definirá como o cálculo será feito de acordo com os valores de lambda. Para lambdas ≠

S = solve(exp(lambda(1)*t)==a0+a1*lambda(1),...
exp(lambda(2)*t)==a0+a1*lambda(2), a0, a1);

$$a_0 + a_1 \lambda_1 + a_2 \lambda_1^2 + \dots + a_{n-1} \lambda_1^{n-1} = e^{\lambda_1 t}$$

$$\begin{cases} a_0 + a_1 \lambda_1 = e^{\lambda_1 t} \\ a_0 + a_1 \lambda_2 = e^{\lambda_2 t} \end{cases}$$

Coefficiente_a0 = S.a0;

Coefficiente_a1 = S.a1;

Stau = solve(exp(lambda(1)*(t-tau))==a0tau+altau*lambda(1),...
exp(lambda(2)*(t-tau))==a0tau+altau*lambda(2), a0tau, altau);

Coefficiente_a0tau = Stau.a0tau;

Coefficiente_altau = Stau.altau;

SOLVE (expressão 1, expressão 2, var1, var2)

↳ solução de equações algébricas

$$a_0 + a_1 \lambda + a_2 \lambda^2 + \dots + a_{n-1} \lambda^{n-1} = e^{\lambda t}$$

$$\begin{cases} a_0 + a_1 \lambda_1 = e^{\lambda_1 t} \\ \frac{d}{d\lambda_1} (a_0 + a_1 \lambda_1) = \frac{d}{d\lambda_1} (e^{\lambda_1 t}) \end{cases} \rightarrow \begin{cases} a_0 + a_1 \lambda_1 = e^{\lambda_1 t} \\ a_1 = t e^{\lambda_1 t} \end{cases}, \lambda_1 =$$

else (PARA LAMBDA IGUAIS)

```
S= solve(exp(lambda(1)*t)==a0+a1*lambda(1), t*exp(lambda(2)*t)==a1, a0, a1)
S.a0;
S.a1;
```

```
Stau= solve(exp(lambda(1)*(t-tau))==a0tau+ altau*lambda(1), ...
t*exp(lambda(2)*(t-tau))==altau, a0tau, altau);
Stau.a0tau;
Stau.altau;
```

$$e^{At} = a_0 I + a_1 A + a_2 A^2 + \dots + a_{n-1} A^{n-1}$$

end

$$e^{At} = a_0 I + a_1 A$$

```
Matriz_Transicao = [S.a0 0; 0 S.a0] + (S.a1*A) → COMUM TANTO PARA λ1 = λ2 E λ1 ≠ λ2
```

```
Termo1= Matriz_Transicao*X0;
```

```
Matriz_Transicao_Tau = [Stau.a0tau 0; 0 Stau.altau] + (Stau.altau*A)
```

```
Termo2= int(Matriz_Transicao_Tau*B, tau, 0, t);
```

```
Solucao_Final = Termo1+Termo2; → É a solução da Eq. de Estados com Condições Iniciais (Resposta Temporal)
```

```
%Valores de tensao e corrente no capacitor
Tensao_no_Capacitor = Solucao_Final(2)
Corrente_no_Indutor = Solucao_Final(1)
```

$$x(t) = \underbrace{e^{At} x_0}_{\text{Termo 1}} + \underbrace{\int_0^t e^{A(t-\tau)} b u(\tau) d\tau}_{\text{Termo 2}}$$

```
%curva da tensao no capacitor
t=0:0.01:10;
figure(1)
plot(t, double(subs(Solucao_Final(2), t)), 'g');
xlabel('tempo em segundos')
ylabel('Tensão de saída no capacitor')
title('Tensão do Capacitor x Tempo')
grid
figure(2)
plot(t, double(subs(Solucao_Final(1), t)), 'r');
xlabel('tempo em segundos')
ylabel('Corrente do Indutor')
title('Corrente no Indutor x Tempo')
grid
```

int (Função, variável a ser integ., 0, t)