Universidade de Brasília

Análise Dinâmica Linear

Experimento I

Matlab

---

25 de Setembro de 2013

Professor	Henrique Cezar Ferreira	
Alunos:		
	Juarez A.S.F	11/0032829

1 Objetivos

Utilizar o Matlab para resolver e analisar um sistema dinâmico linear.

2 Introdução Teórica

Em muitas aplicações de engenharia se faz necessário a solução de sistemas de equações lineares. Algoritmos como a eliminação gaussiana e a regra de Crammer podem ser utilizados para resolver o problema manualmente, mas a medida que o número de incógnitas cresce esse trabalho se torna tedioso. Nessas ocasiões pacotes matemáticos como o Matlab facilitam a solução do problema ao tomarem conta da parte computacional do problema, permitindo que o projetista possa focar em outros aspectos do projeto.

Considere o sistema de n equações lineares e n incógnitas escrito na forma matricial $A \cdot \mathbf{x} = \mathbf{b}$:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

desde que o determinante da matriz dos coeficientes não se anule, a regra de Crammer nos diz que podemos determinar a incógnita x_j fazendo:

$$x_j = \frac{det(A_j)}{det(A)} \tag{1}$$

onde A_j é a matriz dos coeficientes trocando a j-ésima coluna pelo vetor b. Outra maneira de resolver o sistema é calculando a inversa de A e determinar $\mathbf{x}$ pela fórmula:

$$\mathbf{x} = A^{-1} \cdot \mathbf{b} \tag{2}$$

Vejamos agora como sistema lineares aparecem recorrentemente na análise de sistemas dinâmicos lineares. Considere o circuito na figura 1a. Um problema comum é determinar a função de transferência que relaciona a saída v_L com a entrada v(t). Para facilitar vamos procurar essa relação no domínio da frequência s e determinar $H(s) = \frac{V_L(s)}{V(s)}$. A figura 1b mostra o sistema com as grandezas indicadas no domínio da frequência. Aplicamos a Lei de Kirchoff das tensões nas 3 malhas e escrevemos o sistema matricialmente:

$$\begin{pmatrix} R_2 + s(L_3 + L_1) & -L_1 s & -R_2 \\ -L_1 s & R_1 + L_1 s & -R_1 \\ -R_2 & -R_1 & (R_1 + R_2) + L_2 s \end{pmatrix} \cdot \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \begin{pmatrix} 0 \\ V(s) \\ 0 \end{pmatrix} \tag{3}$$

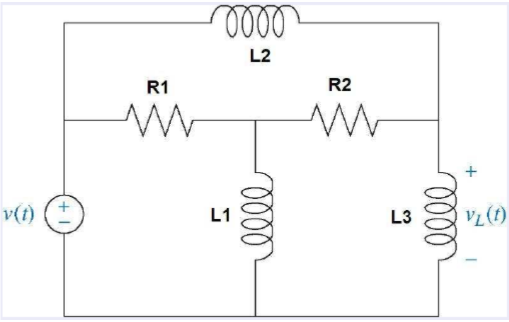
pela regra de Crammer a corrente I_1 pode ser determinada como:

$$I_1(s) = \frac{\begin{vmatrix} 0 & -L_1 s & -R_2 \\ V(s) & R_1 + L_1 s & -R_1 \\ 0 & -R_1 & (R_1 + R_2) + L_2 s \end{vmatrix}}{\begin{vmatrix} R_2 + s(L_3 + L_1) & -L_1 s & -R_2 \\ -L_1 s & R_1 + L_1 s & -R_1 \\ -R_2 & -R_1 & (R_1 + R_2) + L_2 s \end{vmatrix}} \tag{4}$$

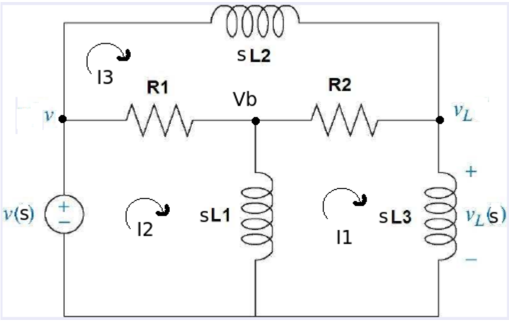
e então V_L pode ser determinado:

$$V_L = sL_3 \cdot I_1 \tag{5}$$

dividindo por V(s) temos a função de transferência procurada. No experimento que segue usaremos o Matlab para resolver esse determinante e agilizar o cálculo da função de transferência.



(a) sistema dinâmico linear elétrico



(b) sistema no domínio da frequência

3 Descrição Experimental

Primeiramente resolvemos o seguinte sistema de equações lineares:

$$\begin{cases} x + y + 2z = 9 \\ 2x + 4y - 3z = 1 \\ 3x + 6y - 5z = 0 \end{cases}$$

o script matlab para resolver esse sistema é apresentado a seguir:

```
%entrando com o sistema na forma Ax = b

    A = [1  1  2;
         2  4  -3;
         3  6  -5];

    b = [9;
         1;
         0;];
disp('Resolvendo um sistema na forma A\b')
    x = A\b

disp('Resolvendo um sistema na forma A^(-1)*b')
    x = A^(-1)*b
```

Vamos agora resolver a equação 4 da introdução para determinar a função de transferência. O código é apresentado a seguir:

```
%vamos salvar a saida de texto
    diary('exercicio2-output')
    diary on

% variaveis simbolicas
    syms L1 L2 L3 R1 R2 V s t


%entrando com a matriz dos coeficientes
    A = [ s*(L3+L1)+R2 -L1*s -R2;
            -L1*s L1*s+R1 -R1;
            -R2 -R1 L2*s+R1+R2]

%entrando com a matriz Aj
%a primeira coluna(correspondente a I1) e substituida
    Aj = A;
    Aj(:,1) = [0;
               V;
               0]

 %calcula I1 e simplifica a expressao encontrada
     i1 = det(Aj)/det(A);
     i1 = simplify(i1)

 %calculamos a funcao de transferencia
     VL = i1*L3*s
     H = VL/V

 %agrupa termos semelhantes e mostra na tela
  %...de modo 'bonito'
     H = collect(H)
     pretty(H)

     diary off
```

Por último, criamos uma rotina que irá receber como parâmetros os valores dos componentes do circuito e retornar a função de transferência encontrada. O código é mostrado a seguir. A rotina é semelhante ao código anterior, por isso omitimos os comentários.

```
  function H = circuitSolver(R1, R2, L1, L2, L3)

diary('exercicio2b-output');
diary on;

                syms V s t;
                A = [ s*(L3+L1)+R2 -L1*s -R2;
                         -L1*s L1*s+R1 -R1;
                         -R2 -R1 L2*s+R1+R2];
                Aj = A;
                Aj(:,1) = [0;
                          V;
                          0];

                 i1 = det(Aj)/det(A);
                 i1 = simplify(i1);
                 VL = i1*L3*s;
                 H = VL/V;
                 H = collect(H);
 diary off;
```

A função acima é salva em um arquivo 'circuitSolver.m'. O código a seguir testa o funcionamento da rotina para algumas entradas:

```
      H = circuitSolver(1,1,1,1,1)
      H = circuitSolver(6,1,7,5,1)
```

4 Resultados

A saída dos códigos descritos é apresentada a seguir:

- **resolvendo o sistema**

```
A =

     1     1     2
     2     4    -3
     3     6    -5


b =

     9
     1
     0

Resolvendo um sistema na forma A\b

x =

    1.0000
    2.0000
    3.0000

Resolvendo um sistema na forma A^(-1)*b

x =

    1.0000
    2.0000
    3.0000
```

- **achando a função de transferência**

```
A =

[ R2 + s*(L1 + L3),      -L1*s,              -R2]
[            -L1*s, R1 + L1*s,              -R1]
[              -R2,       -R1, R1 + R2 + L2*s]
```

```
Aj =

[ 0,      -L1*s,              -R2]
[ V, R1 + L1*s,               -R1]
[ 0,        -R1, R1 + R2 + L2*s]

H =
                  2
((L1 L2 L3) s  + (L1 L3 R1 + L1 L3 R2) s + L3 R1 R2) /

                      2
   ((L1 L2 L3) s  + (L1 L2 R1 + L1 L2 R2 + L1 L3 R1 + L1 L3 R2
   + L2 L3 R1) s + L2 R1 R2 + L3 R1 R2)
```

Utilizando a função 'latex' do Matlab podemos obter o código LateX para a expressão acima. Temos :

$$H = \frac{(\mathrm{L1}\,\mathrm{L2}\,\mathrm{L3})\ s^2 + (\mathrm{L1}\,\mathrm{L3}\,\mathrm{R1} + \mathrm{L1}\,\mathrm{L3}\,\mathrm{R2})\ s + \mathrm{L3}\,\mathrm{R1}\,\mathrm{R2}}{(\mathrm{L1}\,\mathrm{L2}\,\mathrm{L3})\ s^2 + (\mathrm{L1}\,\mathrm{L2}\,\mathrm{R1} + \mathrm{L1}\,\mathrm{L2}\,\mathrm{R2} + \mathrm{L1}\,\mathrm{L3}\,\mathrm{R1} + \mathrm{L1}\,\mathrm{L3}\,\mathrm{R2} + \mathrm{L2}\,\mathrm{L3}\,\mathrm{R1})\ s + (\mathrm{L2}\,\mathrm{R1}\,\mathrm{R2} + \mathrm{L3}\,\mathrm{R1}\,\mathrm{R2})}$$

- **testando a rotina**

```
>> H = circuitSolver(1,1,1,1,1)

H =

(s^2 + 2*s + 1)/(s^2 + 5*s + 2)

>>
>> H = circuitSolver(6, 1, 7, 5, 1)

H =

(35*s^2 + 49*s + 6)/(35*s^2 + 324*s + 36)
```

5 Discussões e Conclusões

Na primeira etapa vimos como resolver rapidamente sistema lineares com a notação matricial do matlab. Em seguida vimos como transformar um circuito em um sistema matricial a ser resolvido. Na terceira etapa rescrevemos o script anterior, mas na forma de uma função que reebe de entrada os parâmetros do ciruito. As vantagens da codificação de rotinas para o matlab se torna evidente quando um mesmo circuito precisa ser resolvido várias vezes. Nesse caso, podemos apenas mudar alguns parâmetros de entrada de uma rotina e obter a nova resposta. O engenheiro então resolve o problema apenas uma vez e deixa para o computador a tarefa tediosa dos cálculos repetitivos.

Referências

[1] Nise, N.S. *Engenharia de Sistemas de Controle* 5ª ed. LTC, 2009.

[2] Ogata, K. *Moder Control Engeeniring* 5ª ed. Pearson, 2010.