



UnB

Universidade de Brasília

Laboratório de Análise Dinâmica Linear

Experimento 01 - primeira parte

Simulação de Sistemas Dinâmicos Lineares

Visão Geral do Matlab

Por que se analisam sistemas dinâmicos?

- O sistema ainda não existe (projeto).
- Explicar o funcionamento de sistemas já existentes.
- O teste experimental é muito caro ou perigoso.

Como proceder com a simulação?

- Deduzir as equações matemáticas que descrevem o sistema (modelagem).
- Utilizar alguma ferramenta que permita analisar o comportamento do modelo ao longo do tempo (simulador).
- Nosso interesse: simulação utilizando o programa MATLAB.

O programa Matlab

- MATLAB: MATrix LABoratory - Laboratório de Matrizes.
- Programa de simulação matemática capaz de realizar operações matriciais, construir gráficos em duas ou três dimensões, auxiliar no processamento de sinais, além de manipular outras funções especializadas.
- Possui funções de tratamento numérico de alto desempenho permitindo resolver problemas computacionais técnicos de forma eficiente e satisfatória.
- Possui um ambiente interativo, além de permitir a execução de arquivos textos (extensão .m) contendo sequências de instruções definidas pelo usuário.

Janelas do programa Matlab

- Command Window: principal janela do Matlab. Onde os dados e instruções são digitados no prompt '»»'.
- Command History: histórico dos últimos comandos digitados.
- Current Directory: diretório corrente do programa.
- Workspace: janela onde se visualizam dados e variáveis correntes.
- M-file Editor: janela para edição e criação de arquivos do Matlab (.m).

Como obter ajuda no Matlab?

- Quando o comando é conhecido: »» help <comando de interesse>.
- Quando o comando não é conhecido: »» lookfor <palavra chave>.

Operações aritméticas

- O Matlab utiliza operadores padrões:

- ▶ `>> (5 + 9) * 6`
- ▶ `>> 3.14/6 - 1`
- ▶ `>> 4 ^6`

- Funções pré-definidas:

- ▶ `>> 3*sin(pi/4)-6`

Variáveis

- O Matlab permite salvar resultados em variáveis.
- O nome das variáveis pode ser alfanumérico, mas precisa começar com letra.

- ▶ `>> raio=4;`
- ▶ `>> circunferencia=2*pi*raio;`
- ▶ `>> area=pi*raio^2;`

Variáveis - manipulação

- who: Lista os nomes das variáveis.
- whos: Lista o nome e o tipo das variáveis.
- clear: Elimina todas as variáveis da área de trabalho.
- clear+nome da variável: Elimina apenas a variável especificada.
- save: Salva as variáveis em arquivo para futura utilização.
 - ▶ `>> save dados_circulo raio circunferencia area;`
- load: Carrega as variáveis previamente salvas.
 - ▶ `>> load dados_circulo;`

Controle de Fluxo

- Condicionais: if-elseif-else-end.
- Laços: for-end, while-end.

Operadores lógicos

Operador	Descrição
<	Menor que
<=	Menor ou igual a
>	Maior que
>=	Maior ou igual a

Operador	Descrição
==	Igual a
~=	Diferente de
&	E
	OU

Vetores e Matrizes

- Uma matriz/vetor pode ser definido da seguinte forma no Matlab:
 - ▶ `>> A=[0 1 2 3; 4 5 6 7; 8 9 10 11; 12 13 14 15];`
- Ou ainda com alguns comandos específicos:
 - ▶ `>> x=[1:1:10];`
 - ▶ `>> x=linspace(1,10,50);`
 - ▶ `>> A=ones(5);`
 - ▶ `>> B=eye(5);`
 - ▶ `>> C=rand(5);`
- Indexação na forma padrão: `>> A(linha, coluna).`

Vetores e Matrizes - Operações Aritméticas

Operador	Descrição
+	adição (matricial e escalar)
-	subtração (matricial e escalar)
*	multiplicação matricial
/	divisão matricial à direita $\rightsquigarrow A/B = A * B^{-1}$
\	divisão matricial à esquerda $\rightsquigarrow A \setminus B = A^{-1} * B$
^	potenciação matricial $\rightsquigarrow A^2 = A^2 = A * A$
'	transposição matricial
.*	multiplicação elemento por elemento
./	divisão elemento a elemento pela direita
.\	divisão elemento a elemento pela esquerda
.^	potenciação elemento a elemento

Álgebra Linear

Operador	Descrição
$\det(A)$	retorna o determinante da matriz A
$\text{inv}(A)$	retorna a inversa da matriz A
$\text{eig}(A)$	retorna os autovalores da matriz A

Análise de Dados

Operador	Descrição
sum	soma dos elementos de um vetor de entrada
prod	produto dos elementos de um vetor de entrada
mean	média aritmética dos elementos de um vetor de entrada
min e max	valor mínimo e máximo dos elementos de um vetor de entrada
sort	vetor de entrada com elementos ordenados em ordem crescente

Gráfico em Duas Dimensões

- Cria-se um vetor x com as coordenadas do eixo das abscissas;
- Escreve-se a função desejada, a partir do vetor x , definindo um novo vetor y das ordenadas;
- Desenha-se o gráfico: `>> plot(x,y);`
- Exemplo:
 - ▶ `>> x = [0:0.2:10];`
 - ▶ `>> y = x.^2;`
 - ▶ `>> plot(x,y);`

Polinômios

- No Matlab um polinômio é armazenado como um vetor;
- O tamanho do vetor menos um é igual à ordem do polinômio;
- Os elementos do vetor são os coeficientes do polinômio em ordem decrescente do expoente;

Polinômios

- Exemplo: $p(x) = x^3 - 2x^2 - x + 2$
 - ▶ `>> p=[1 -2 -1 2]`
- Valor do polinômio em um ponto x_1 :
 - ▶ `>> y=polyval(p,x1);`
- Raízes do polinômio:
 - ▶ `>> roots(p)`

Exercício 01:

- Obtenha o gráfico do polinômio do exemplo anterior para os valores de x entre -3 e 3 com incremento de 0.1. Estime graficamente o valor das raízes (dica: utilize a figura quadriculada, help grid) e verifique os valores obtidos com o comando **roots**.

Exercício 02:

- Dado a matriz

$$A = \begin{bmatrix} 1 & 4 & -3 \\ 2 & 1 & 5 \\ -2 & 5 & 3 \end{bmatrix}$$

Calcule:

- ▶ Determinante de A;
- ▶ Os autovalores e autovetores de A (dica: `>> help eig`);
- ▶ Inversa de A (atribuir o resultado a uma matriz B).

Em seguida:

- ▶ Verificar o resultado: $A * B = I$ e $B * A = I$;
- ▶ Salvar os dados do exercício em arquivo `ex01_data.mat`.

Exercício 03:

- Considere um motor a pistão em uma aeronave. Durante o seu funcionamento, um sensor de temperatura registrou os seguintes dados:

Tempo (min)	Temperatura (°F)
0	105
0.5	126
1.0	119
1.5	129
2.0	132
2.5	128
3.0	131
3.5	135
4.0	136
4.5	132
5.0	137

Conversão de temperatura

$$t(^{\circ}C) = \frac{t(^{\circ}F) - 32}{1.8}$$

- Determinar o valor máximo, mínimo e médio da temperatura em graus Celsius e plotar a evolução da temperatura no tempo, especificando o eixo da temperatura, do tempo e o título do gráfico.

Função de Transferência de Sistemas Dinâmicos

- Relação entre a transformada de Laplace do sinal de saída $y(t)$ pelo sinal de entrada $u(t)$:

$$H(s) = \frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{u(t)\}} = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}$$

- No Matlab, o comando **tf** cria um objeto função de transferência;
 - ▶ \gg `tf(N,D)`
 - ▶ N polinômio do numerador, D polinômio do denominador.
- Exemplo: Circuito RLC série

$$\frac{V_c(s)}{V(s)} = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

Definir a FT no Matlab para os seguintes parâmetros $R=6\Omega$, $C=0.04F$, $L=1H$.

Transformada de Laplace

- Definir as variáveis simbólicas:
 - ▶ `>> syms t s;`
- Definir a função no domínio do tempo:
 - ▶ `>> f=2*exp(-t)-2*exp(-2*t);`
- Aplicar a transformada de Laplace:
 - ▶ `>> F=laplace(f);`
- Transformada inversa de Laplace:
 - ▶ `>> f=ilaplace(F);`

Expansão em Frações Parciais

$$H(s) = \frac{Y(s)}{U(s)} = \frac{r(1)}{(s-p(1))} + \frac{r(2)}{(s-p(2))} + \dots + \frac{r(n)}{(s-p(n))} + k(s)$$

Expansão em Frações Parciais

- No Matlab:
 - ▶ `>> [r,p,k] = residue(N,D);`
- Exemplo:

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

Funções e *Scripts*

- Tudo que foi apresentado até aqui pode ser automatizado!
- Funções e *scripts*: sequências de comandos do Matlab armazenado em arquivo texto com extensão '.m':
 - ▶ *Scripts*: simplesmente executam sequências de comandos;
 - ▶ Funções: aceitam parâmetros de entrada.
- Cabeçalho para funções:
`function [Saída_1, ... , Saída_n] = nome_da_funcao(Entrada_n , ... ,Entrada_m)`

Exercícios para o relatório

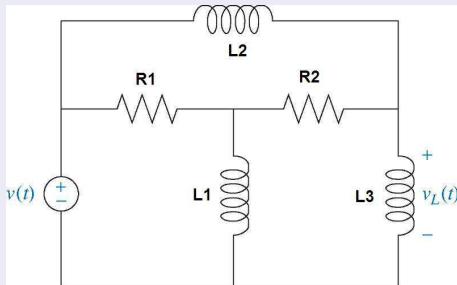
- 1) Escrever um algoritmo em arquivo '.m' para resolver o seguinte sistema de equações lineares:

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

e apresentar o código desenvolvido, juntamente com o resultado obtido após a execução do mesmo, impresso.

Exercícios para o relatório

2) Para o seguinte circuito elétrico



a) Obter a função de transferência $H(s) = V_L(s)/V(s)$, apresentando todo o desenvolvimento da modelagem.

Exercícios para o relatório

Sugestão:

- Deduzir as equações no domínio 's' pela lei das tensões;
- Escrever na forma matricial ($Ax=b$);
- Calcular a corrente no indutor 3 ($I_3(s)$) pela regra de Cramer utilizando o toolbox de matemática simbólica no Matlab;
- Simplificar o objeto simbólico: $\gg I_3 = \text{simplify}(I_3)$;
- Substituir $I_3(s)$ na expressão de $V_L(s)$ para obter $H(s)$;
- Para melhor visualização de $H(s)$ utilizar os comandos *collect* e *pretty*:
 - $\gg H = \text{collect}(H)$ (agrupa termos semelhantes, help collect)
 - $\gg \text{pretty}(H)$ (melhora a visualização, help pretty)

- b) Uma vez obtido a expressão de $H(s)$, criar uma função no Matlab que receba como parâmetros de entrada os valores das indutâncias e resistências e forneça a respectiva função de transferência (objeto função de transferência). Avalie a sua função para os seguintes casos

1º: $R_1 = R_2 = 1\Omega$ e $L_1 = L_2 = L_3 = 1H$;

2º: $R_1 = 6\Omega$, $R_2 = 1\Omega$, $L_1 = 7H$, $L_2 = 5H$ e $L_3 = 1H$

Exercícios para o relatório

apresentando o código desenvolvido, juntamente com os resultados obtidos após a execução do mesmo, impresso.

- 3) Apresentar o script desenvolvido em sala de aula para o cálculo da resposta ao degrau de circuito RLC, juntamente com os resultados obtidos.

Bibliografia

- Nise, N. S., Engenharia de Sistemas de Controle. 5ª Edição, LTC, 2009.
- Apostila de Introdução ao MATLAB[®], Universidade Federal Fluminense, Centro Tecnológico, Escola de Engenharia, <http://www.telecom.uff.br/pet/petws/downloads/apostilas/MATLAB.pdf>.
- Robinson, T.; Kambouchev, N., 16.06/16.07 Matlab/Simulink Tutorial, Massachusetts Institute of Technology, <http://dSPACE.mit.edu/bitstream/handle/1721.1/60691/16-07-fall-2004/contents/study-materials/matlabtut.pdf>.