Uma Introdução ao Ambiente de Programação MATLAB

Alisson Jaques*. Matheus Nogueira*. Sillas Francisco*.

**Universidade do Estado de Minas Gerais*

Abstract: In this article you will find an introduction to the MATLAB environment. Such an introduction consists of solving the MATLAB exercises proposed in Chapter 1 of the Book of Robotics (3rd Edition) by author John J. Craig.

Resumo: Neste artigo encontra-se uma introdução ao ambiente MATLAB. Tal introdução consiste na resolução dos exercícios de MATLAB propostos no Capítulo 1 do Livro de Robótica (3ª Edição) do autor John J. Craig.

Palavras-chaves: MATLAB; vetores; matrizes; robótica.

1. INTRODUÇÃO

Para o estudo da Robótica é fundamental que o aluno tenha um bom entendimento do software MATLAB, pois o mesmo possibilita a manipulação de diversas operações algébricas que, feitas de forma manual, seriam bastante enfadonhas. Sendo assim, no Capítulo 1 do livro Robótica do author John J. Craig é proposto ao aluno um estudo, visando a familiarização com o ambiente do MATLAB (neste capítulo não tem exercícios de programação propostos ao aluno). Este artigo visa responder às questões de MATLAB propostas no livro, dando uma introdução ao ambiente MATLAB.

1.2 MATLAB

MATLAB é uma linguagem de programação criada com o intuito de desenvolver aplicativos de natureza técnica. MATLAB vem de MATRIX LABORATORY e foi originalmente desenvolvido para prover um acesso amigável ao tratamento de vetores e matrizes. Sendo assim, é necessário ter um ótimo entendimento de Álgebra Linear, para melhor aproveitamento da ferramenta.

2. AS PALAVRAS-CHAVES DEMO E HELP

2.1 A PALAVRA RESERVADA DEMO

A palavra reservada demo tem como função fornecer uma demonstração de um comando ou ferramenta no MATLAB. Ao digitarmos *demo* e depois enter no prompt de comando do MATLAB o mesmo retorna uma janela conteúdo uma interface de ajuda ao usuário (conforme FIGURA 1).

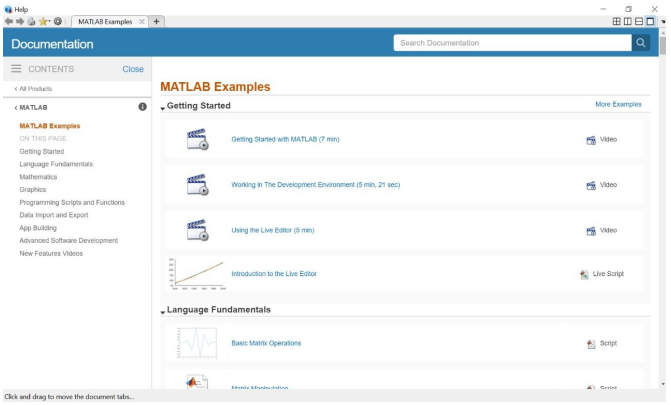

FIGURA 1: Janela criada ao usar o comando *demo* no prompt do MATLAB

É possível também obter informações específicas de comandos em matlab utilizando a palavra *demo*. Se digitarmos, por exemplo, *demo simulink 'simulink control design'* vamos obter como retorno uma interface que demonstra diversos exemplos de design de controle do simulink (conforme FIGURA 2).

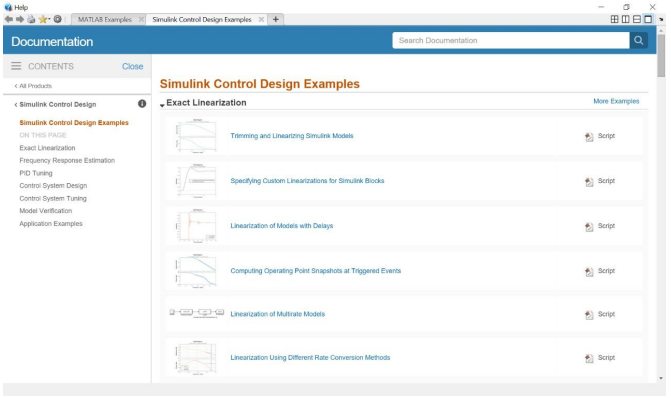

FIGURA 2: Janela criada ao usar o comando *demo simulink 'simulink control design"*

2.1 A PALAVRA RESERVADA HELP

Como o próprio nome indica a palavra reservada help fornece ajuda para a compreensão dos diversos conceitos, comandos

e funções do MATLAB. Se digitarmos *help* no prompt de comando e, em seguida, enter surgirá uma lista contendo informações de todos os comandos presentes do MATLAB (conforme FIGURA 3).

```
>> help
HELP topics:

matlab\datafun                  - Data analysis and Fourier transf
matlab\datatypes                - Data types and structures.
matlab\elfun                    - Elementary math functions.
matlab\elmat                    - Elementary matrices and matrix m
matlab\funfun                   - Function functions and ODE solve
matlab\general                  - General purpose commands.
matlab\iofun                    - File input and output.
matlab\lang                     - Programming language constructs.
matlab\matfun                   - Matrix functions - numerical lin
matlab\ops                      - Operators and special characters
matlab\polyfun                  - Interpolation and polynomials.
matlab\randfun                  - Random matrices and random strea
matlab\sparfun                  - Sparse matrices.
matlab\specfun                  - Specialized math functions.
matlab\strfun                   - Character strings.
matlab\timefun                  - Time and dates.
matlabhdlcoder\matlabhdlcoder   - (No table of contents file)
matlabxl\matlabxl               - (No table of contents file)
matlab\demos                    - Examples.
matlab\graph2d                  - Two dimensional graphs.
matlab\graph3d                  - Three dimensional graphs.
matlab\graphics                 - Handle Graphics.
```

FIGURA 3: Lista de ajuda que aparece ao digitarmos *help* no terminal do MATLAB

É possível também obter ajuda específica no MATLAB, bastando para isso inserir a palavra help seguida de um comando que se queira obter ajuda. Por exemplo, se digitarmos help delete aparecerá no prompt orientações sobre o comando delete, como descrito na FIGURA 4.

```
>> help delete
 delete Delete file or graphics object.
    delete file_name  deletes the named file from disk.  Wildcard
    may be used.  For example, delete *.p deletes all P-files fro
    current directory.

    Use the functional form of delete, such as delete('file') whe
    file name is stored in a string.

    delete checks the status of the RECYCLE option to determine w
    the file should be moved to the recycle bin on PC and Macinto
    moved to a temporary folder on Unix, or deleted.

    delete(H) deletes the graphics object with handle H. If the o
    is a window, the window is closed and deleted without confirm

    See also recycle.

    Reference page for delete
    Other functions named delete
```

FIGURA 4: Obtemos uma ajuda sobre o comando delete

3. CRIANDO, EDITANDO, SALVANDO E DEPURANDO ARQUIVOS

3.1 CRIANDO ARQUIVOS

Para criar um arquivo no MATLAB, use a barra de menus acesse File->New->Script. ou aperte as teclas “Ctrl + N”. O documento será salvo no formato “Nome do Arquivo”.m que será compilado na Comand Window.

3.2 EDITANDO ARQUIVOS

Uma vez que o arquivo esteja aberto, pode-se editá-lo à vontade. Todavia, é importante tomar cuidado ao se editar arquivos, pois se os mesmos não forem salvos o conteúdo editado poderá ser perdido.

3.3 SALVANDO ARQUIVO

Ao criar ou editar um arquivo o mesmo pode ser salvo ao ir na guia File na barra de menus

3.4 DEPURANDO ARQUIVOS

A depuração permite que o programador acompanhe determinada parte do código linha a linha, visualizando todos os passos que o algoritmo segue para facilitar a correção de erros. Para depurar seu código faça: Clique no número referente a linha a partir da qual quer ver passo a passo, aparecerá uma bolinha vermelha (ver FIGURA 5) depois clique em “Debug” e então em “Run”.

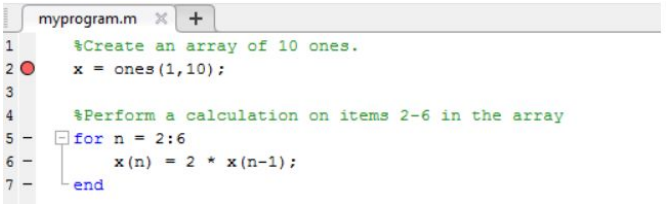

FIGURA 5: Selecionando a linha

Na linha em que o algoritmo está aparecerá uma seta verde como na imagem a seguir, então aperte F10 para passar para a linha de baixo e repita até o fim do mesmo. Para sair do modo de depuração clique em “Debug” e “Exit Debugguer Mode” (FIGURA 6).

FIGURA 6: Seta com conteúdo sendo depurado

4. CRIANDO ARRANJOS E EXPLORANDO AS FUNÇÕES DE ÁLGEBRA LINEAR

4.1 MATRIZES E VETORES (ARRANJOS)

Matrizes e vetores são os principais tipos de dados na linguagem de programação do MATLAB, pois são as ferramentas fundamentais para o estudo da Álgebra Linear. A seguir, mostra-se como manipular esses conceitos no MATLAB.

4.1.1 Vetor

A sintaxe de criação de vetores é muito simples. Se quisermos, por exemplo, criar o vetor v = (a,b,c), onde a, b e c são número reais, basta digitarmos o seguinte comando:

v = [a,b,c]; ou v = [a b c]

No matlab, assim como em outras linguagens de programação, existem diversos métodos que manipulam vetores (como métodos para achar a quantidade de linhas ou colunas por exemplo).

4.1.2 Matriz

Uma matriz em MATLAB é formada por um conjunto de vetores linhas separados pelo símbolo “;”. Se quisermos, por exemplo, criar a matriz *m = [a b c /d e f]* basta digitarmos o seguinte comando:

m = [a b c; d e f]

Existem diversos comandos em MATLAB que possibilitam a manipulação de dados de matrizes (assim como os mencionados para vetores).

4.2 PRODUTO ESCALAR

O produto escalar entre dois vetores pode ser dado pela função *dot* do MATLAB. Se tivermos um vetor A e outro B, seu produto escalar poderá ser obtido pelo seguinte comando:

prodEscalar = dot(A,B)

No comando acima, a variável prodEscalar armazena o produto escalar dos vetores A e B. Podemos também multiplicar um vetor qualquer por um escalar em MATLAB, bastando para isso multiplicar um escalar por um vetor, como o seguinte códito abaixo:

escalarVetor = a*vetor1

O mesmo vale para divisão, soma e subtração por um escalar.

4.3 PRODUTO VETORIAL

Podemos obter o produto vetorial entre dois vetores utilizando a palavra reservada *cross*. A FIGURA 7 demonstra usos do comando cross.

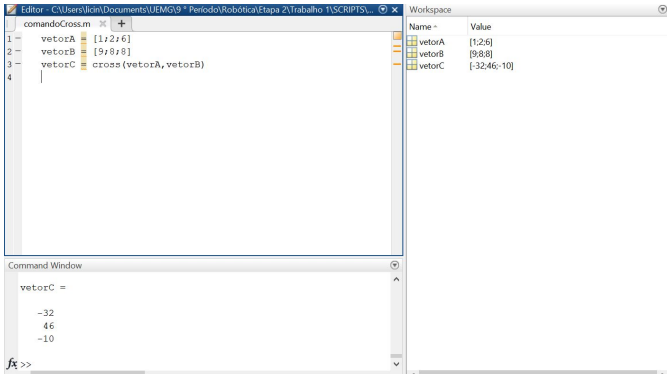

FIGURA 7: Uso do comando *dot*

4.4 TRANSPOSIÇÃO

Obtemos a transposta de uma matriz em MATLAB inserindo aspas simples (‘) na frente do identificador da matriz. No exemplo abaixo a variável *T* recebe a matriz transposta de *A*.

T = A’

4.5 DETERMINANTES

Obtemos o determinante de uma matriz A utilizando o comando det(matriz). Como, exemplo, temos:

determinanteB = det(matrizB)

No código acima, a variável *determinanteB* recebe o valor do determinante da matriz B.

4.6 INVERSAS

Obtemos a inversa de uma matriz através do comando inv(matriz), conforme exemplo abaixo:

inversaA = inv(matrizA)

No código acima, a variável *inversaA* recebe a matriz inversa de A.

4.7 SOLUÇÃO DE EQUAÇÕES LINEARES

Existem diversos comandos em MATLAB que permitem a solução de Sistemas Lineares. Aqui daremos um exemplo básico, considerando que a matriz gerada pelas equações será quadrada, onde utilizaremos o comando reservado *linsolve(argumento1,argumento2)*. Este comando resolve o sistema linear $Ax = b$ usando a fatoração LU caso a matriz seja quadrada. Na FIGURA 8 encontra um exemplo de uso da função linsolve.

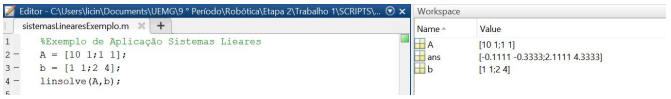

FIGURA 8: Exemplo do comando linsolve

5. ESTRUTURA E CICLOS LÓGICOS EM MATLAB

5.1 ESTRUTURAS

As estruturas são um tipo especial de matriz utilizada para armazenar dados de tipos diferentes. Elas diferem de uma lista por possuir nomes que referenciam o tipo de dado. A FIGURA 9 mostra como são declaradas as estruturas em MATLAB.

```
>> resultado.alunos='maria';
>> resultado.notas=[9 8.5];
>> resultado
resultado =
    alunos: 'maria'
     notas: [9 8.50000000000000]
>> resultado(2).alunos='joao';
>> resultado(2).notas=[10 7];
>> resultado
resultado =
1x2 struct array with fields:
    alunos
    notas
>> resultado.alunos
ans =
maria
ans =
joao
>>
```

FIGURA 9: Declarando estruturas em MATLAB

5.2 CICLOS LÓGICOS

Os ciclos lógicos em MATLAB funcionam exatamente como as funções de laços de repetição das linguagens de programação tradicionais e são muito semelhantes aos

comandos que se usa na linguagem C. Os principais comandos de ciclos são o *while* e o *for* cujas sintaxes são:

while expressãoVerdadeira

instruções

end

for index=expressão

instruções

end

Esses comandos são essenciais para execução de instruções repetidas, mediante critério de parada e muito úteis para a criação de algoritmos que lidam com Álgebra Linear em geral.

6. SUBPROGRAMAS E FUNÇÕES

As funções em MATLAB são como sub-rotinas no FORTRAN, procedimentos em PASCAL e as funções em C. São arquivos .m que podem ser utilizados de duas formas.

1. Para realizar uma tarefa frequente como calcular o valor de uma função matemática para vários valores de uma variável..
2. Como uma sub-rotina dentro de outra rotina. Uma função começa sempre com o seguinte cabeçalho

function [variáveis de saída] = Nome_da_Função (variáveis de entrada)

Todas as variáveis usadas temporariamente são locais, logo, após a execução da função elas são removidas do espaço de trabalho.

7. COMENTÁRIOS E ABAS

7.1 COMENTÁRIOS

Para adicionar comentários no seu arquivo basta utilizar o símbolo "%". Para comentários de múltiplas linha usa-se o símbolo "{%" sozinho na linha para abrir o escopo do comentário e "%}" também sozinho em outra linha para fechar e terminar o mesmo.

7.2 ABAS

Abas são diferentes janelas, onde cada janela corresponde a um script escrito em MATLAB. Sua função é a de facilitar o desenvolvimento de programas pelo programador. Pois com elas o programador não precisa fechar uma aplicação que está mexendo para consultar outra e vice versa.

8. SOBRE O SITE DA MATHWORKS.

O site da MathWorks (www.mathworks.com) é o site oficial das ferramentas MATLAB e SIMULINK. Foi desenvolvido com o propósito de facilitar o dia a dia dos pesquisadores. Além de uma descrição detalhada das ferramentas mencionadas acima o mesmo fornece as mais diversas informações e ajudas sobre as ferramentas incorporadas nos nestes sistemas.

9. SIMULINK E SYMBOLIC TOOLBOX PARA MATLAB

9.1 SIMULINK

Simulink é um ambiente de diagrama de blocos para simulação e projeto baseado em modelo de sistemas de engenharia integrados e de múltiplos domínios. Foi desenvolvido para dar mais praticidade ao usuário no que diz respeito à rotinas de programação, o objetivo é que o usuário preocupe-se menos com detalhes técnicos de linguagens de programação, como C e JAVA por exemplo, utilizando uma linguagem mais intuitiva aplicada à matemática.

9.2 SYMBOLIC TOOLBOX

Symbolic Math Toolbox é uma pacote de ferramentas que fornece funções para resolver, plotar e manipular equações matemáticas simbólicas. Com ele é possível criar, executar e compartilhar código matemático simbólico usando o MATLAB. Esta caixa de ferramentas fornece funções nas mais diversas áreas da matemática, como cálculo, álgebra linear, equações diferenciais algébricas ordinárias etc.

10. ROBOTICS TOOLBOX PARA MATLAB

Robotics Toolbox é um sistema que fornece ferramentas e algoritmos usados para testar e projetar manipuladores robóticos. Esse sistema inclui algoritmos para gerar trajetórias, verificar colisões para manipuladores e robôs humanóides. Também inclui algoritmos para mapear, localizar, planejar caminhos e controlar movimentos de robôs móveis.

4. CONCLUSÃO

Neste artigo fizemos uma introdução ao ambiente MATLAB. Passando pelos conceitos básicos da ferramenta e explorando sua utilidade para a álgebra linear. Assim obtivemos uma base fundamental para podermos avançar no estudo da robótica, onde criaremos scripts em MATLAB que resolvem situações problemas envolvendo Sistemas de Coordenadas e Manipuladores Robóticos.

AGRADECIMENTOS

Agradecemos ao professor Ângelo pela disposição e empenho na disciplina e aos colegas de classe, sempre disponíveis para troca de ideias.

REFERÊNCIAS

Craig, John J. "Robótica. 3ª edição." (2012).

O que é MATLAB. Disponível em: <http://w3.impa.br/~zubelli/tutorial/node1.html>. Acesso em: 20 nov. 2020.

MathWorks. Disponível em: <www.mathworks.com>. Acesso em: 20 nov. 2020.

MATLAB Avançado. Disponível em: <http://mtm.ufsc.br/~melissa/arquivos/matlabpet/aula_01.pdf >. Acesso em: 20 nov. 2020.

TutorialDeMatlab. Disponível em: <https://www.di.ubi.pt/~cbarrico/Disciplinas/ComputacaoCie

ntifica/Downloads/TutorialdeMatlab.pdf>. Acesso em: 20 nov. 2020.

Apostila MATLAB para engenharia Disponível em: <http://ftp.demec.ufpr.br/disciplinas/TMEC078/curso%20Matlab/referencia/6_MatLab_para_Engenharia.pdf> Acesso em: 21 nov. 2020.

MATLAB_12-Funções Disponível em: <https://edisciplinas.usp.br/pluginfile.php/95376/mod_resource/content/1/MATLAB_12-Funcoes.pdf> Acesso em: 21 nov. 2020.

Mini Curso Introdução ao MATLAB Disponível em: <https://www.ufjf.br/getproducao/files/2013/05/Apostila-Mini-Curso-MATLAB-GET-EP1.pdf> Acesso em: 20 nov. 2020.

MATLAB Avançado Disponível em <http://mtm.ufsc.br/~melissa/arquivos/matlabpet/aula_02.pdf> Acesso em: 20 nov. 2020.

Dicas de utilização do MATLAB Disponível em : <http://www2.peq.coppe.ufrj.br/Pessoal/Professores/Arge/COQ897/Matlab/Apostila_Matlab_Andrea.pdf> Acesso em: 20 nov. 2020.

Noções Básicas de Programação em MATLAB Disponível em: <https://www.ufsm.br/app/uploads/sites/783/2020/02/Apostila_Matlab.pdf> Acesso em: 20 nov. 2020.