

ZAB1051 – Processamento de Sinais em Biossistemas

Profa. Dra. Ana Carolina de Sousa Silva

PROGRAMA DA DISCIPLINA

| DATA | CONTEÚDO |
|----------|---|
| 14/04/21 | Apresentação da disciplina e Introdução ao MATLAB® - parte 1 |
| 21/04/21 | Tiradentes - NÃO HAVERÁ AULA |
| 28/04/21 | Introdução ao MATLAB® - partes 2, 3 e 4 |
| 05/05/21 | Sinais e sistemas de tempo discreto - partes 1 e 2 |
| 12/05/21 | Sinais e sistemas de tempo discreto - partes 3 e 4 |
| 19/05/21 | Sinais e sistemas de tempo discreto - partes 5 e 6 |
| 26/05/21 | Transformada Z - partes 1, 2 e 3 |
| 02/06/21 | Transformada Z - partes 4 e 5 |
| 09/06/21 | AVALIAÇÃO 1 |
| 16/06/21 | Análise no domínio da frequência - partes 1 e 2 |
| 23/06/21 | Análise no domínio da frequência - partes 3 e 4 |
| 30/06/21 | Amostragem de sinais contínuos - partes 1, 2 e 3 |
| 07/07/21 | Análise no domínio transformado de sistemas LTI - partes 1, 2, 3 e 4 |
| 14/07/21 | Filtros - partes 1, 2 e 3 |
| 21/07/21 | Filtros - partes 4, 5 e 6 |
| 28/07/21 | AVALIAÇÃO 2 |

Média final = $0,35 * \text{AVALIAÇÃO 1} + 0,65 * \text{AVALIAÇÃO 2}$

Recuperação: Substituição da menor nota entre a AVALIAÇÃO 1 e AVALIAÇÃO 2

Bibliografia

[1] Hayes, MH. **Processamento Digital de Sinais - Coleção Schaum**

[2] Haykin, S, Veen BV. **Sinais e sistemas**

[3] Ingle, VK; Proakis, JG. **Essentials of digital signal processing using MATLAB**

[4] Proakis, JG; Manolakis, DG. **Digital Signal Processing - Principles, Algorithms and Applications**

[5] **Oppenheim, AV; Schafer, RW. Discrete-time signal processing**

[6] Nalon, JA. **Introdução ao processamento digital de sinais**

[7] Smith, SW. **Digital Signal Processing**

[8] Lyons, RG. **Understanding digital signal processing**

Introdução

Introdução

O que é um Sinal?

Introdução

- Para **todo instante** de tempo é possível **associar uma informação**
 - A variação no tempo dessas informações é chamada de **SINAL**

Sinais

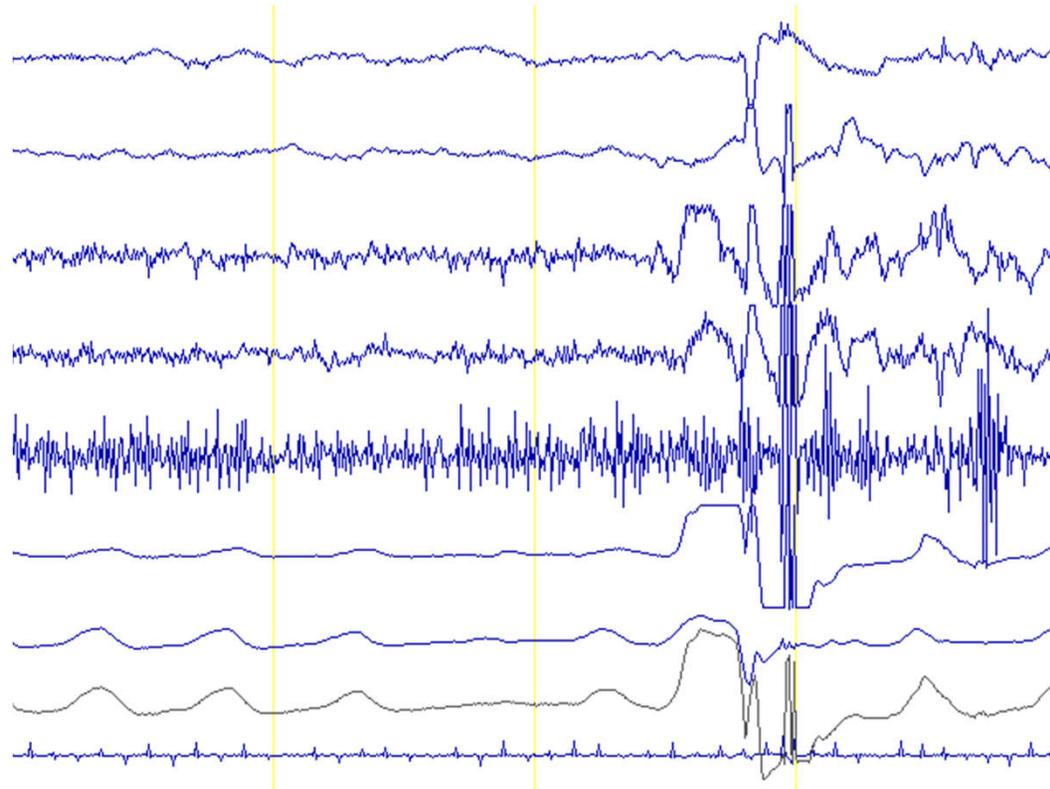
- Como um **parâmetro varia em função de outro**

– contem informação do **estado ou comportamento de um sistema**

- Matematicamente: **função** de uma ou mais variáveis

Sinais

EEG



Sinai's

ECG



Sinais

Sinal elétrico de planta

E.F. Cabral et al. / Computers and Electronics in Agriculture 76 (2011) 1–5

3

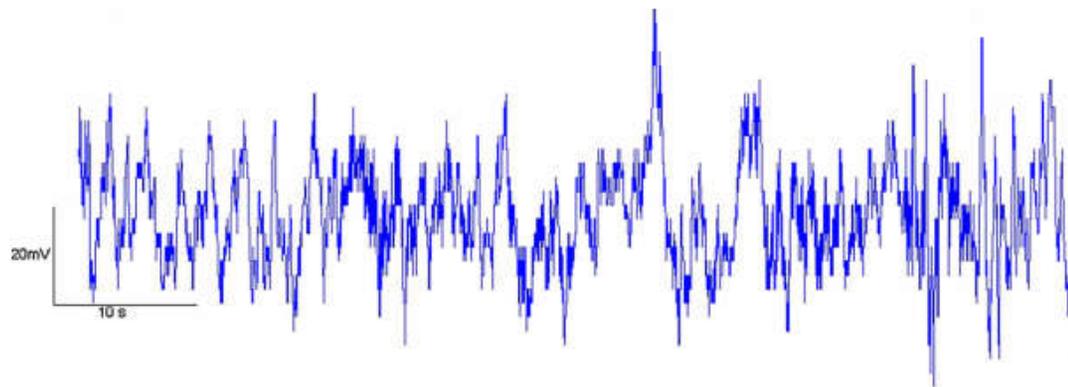
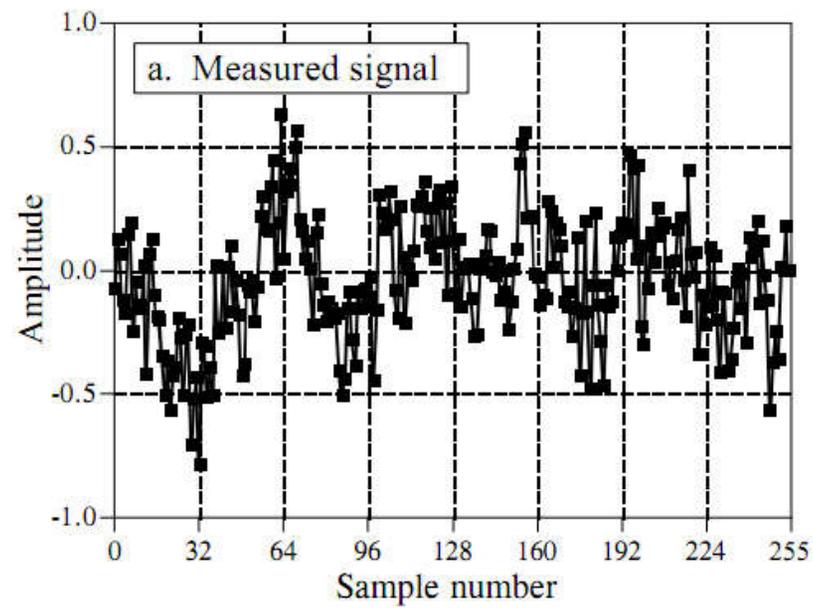


Fig. 2. Diagram representing the electrical model proposed to explain the signal acquired from plant.

Sinais

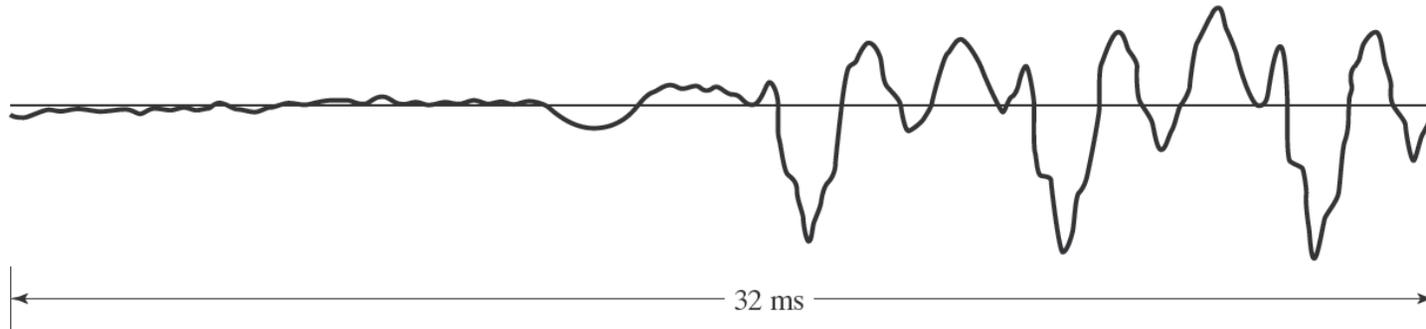
Sinal adquirido por um microfone posicionado na água

Time Domain

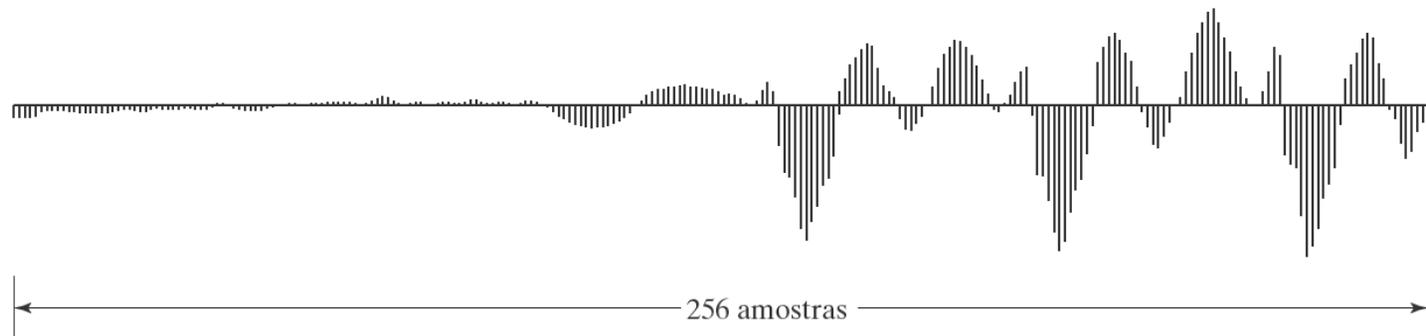


Sinais

- Segmento de um sinal de voz em tempo contínuo $x_a(t)$.



- Sequência de amostras do sinal de voz.



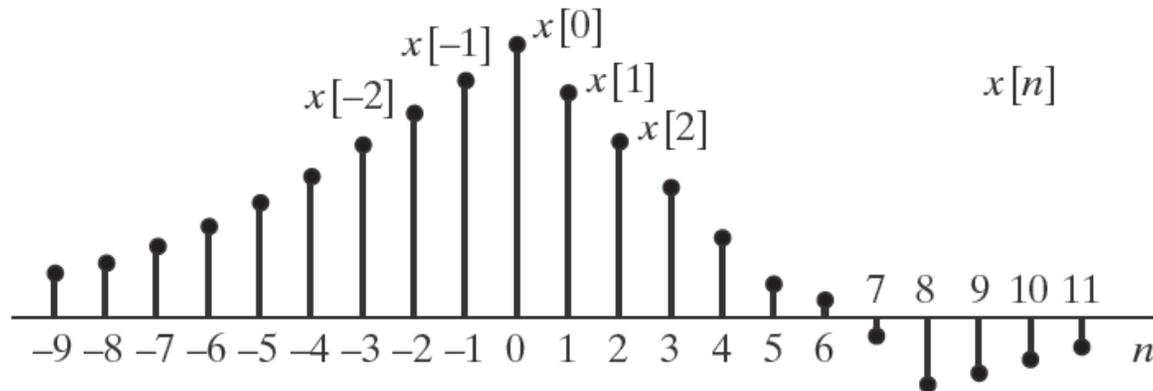
Sinais

- Variável independente pode ser contínua ou discreta
 - **Contínua** → Sinais analógicos
 - **Discreta** → Sinais discretos
- **PDS** lida com a transformação de sinais que são **discretos** no tempo e na amplitude

Sinais digitais

- Obtidos de 2 formas:
 - Amostragem e quantização de um sinal analógico
 - Gerados digitalmente por algum dispositivo

Representação gráfica de um sinal de tempo discreto.



Introdução

O que são Sinais?



O que é Processamento Digital de Sinais (PDS)?

O que são Sistemas Lineares?

Introdução

- Processamento Digital de Sinais (PDS) é uma das **ferramentas mais poderosas** em ciências e engenharia
- Tem **aplicações em diversas áreas**
 - Cada uma dessas áreas tem seus algoritmos e técnicas especializadas
 - entretenimento, comunicações, exploração do espaço, medicina, arqueologia e geofísica, apenas para citar algumas

Processamento Digital de Sinais (PDS)

Como o sinal foi gerado
Características básicas do sinal

Sinal



Que tipo de informação desejo acessar

Informações

Como
extrair?

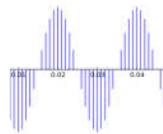


Processamento

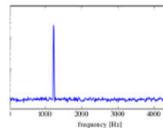
Quais técnicas utilizar

Processamento Digital de Sinais (PDS)

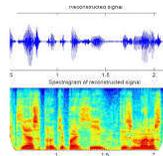
Muitos sinais presentes na natureza tem comportamento oscilatório



Tempo

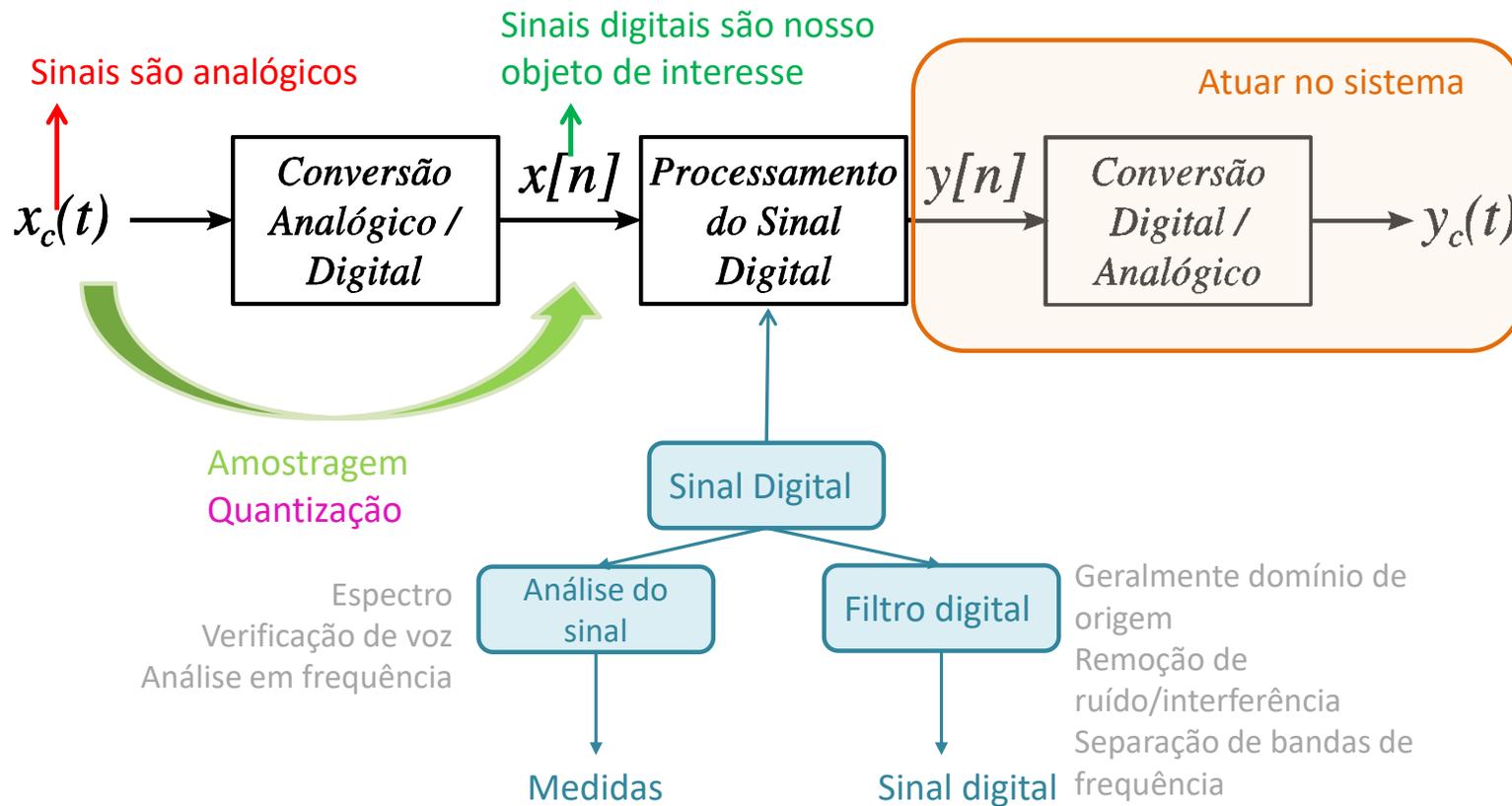


Frequência

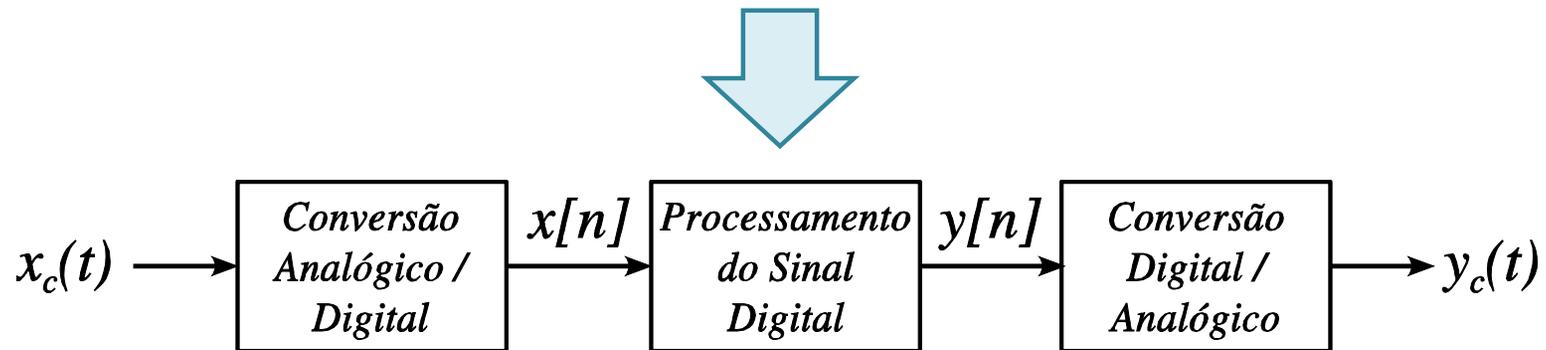


Tempo-frequência

Processamento Digital de Sinais (PDS)



Processamento Digital de Sinais (PDS)

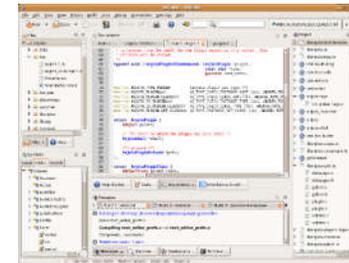
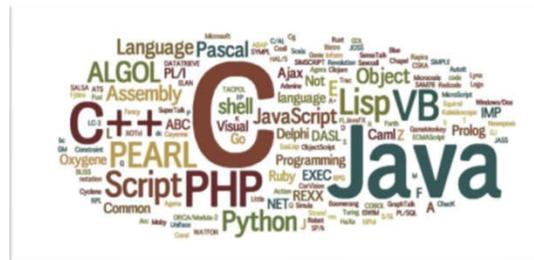


Técnicas de processamento

Algoritmos

SISTEMAS LTI

$x[n] \rightarrow$ SISTEMA $\rightarrow y[n]$



Processamento Digital de Sinais (PDS)



Técnicas de processamento

Algoritmos

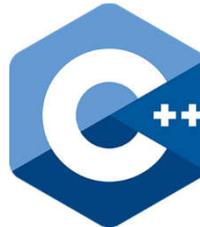
SISTEMAS LTI

$x[n]$ → SISTEMA → $y[n]$



Processamento Digital de Sinais (PDS)

- PDS envolve principalmente algoritmos implementados em hardware ou **software**
 - Linguagens de Programação



Introdução

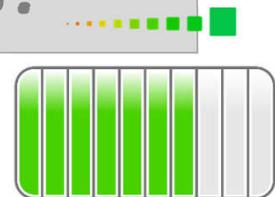
O que são Sinais?

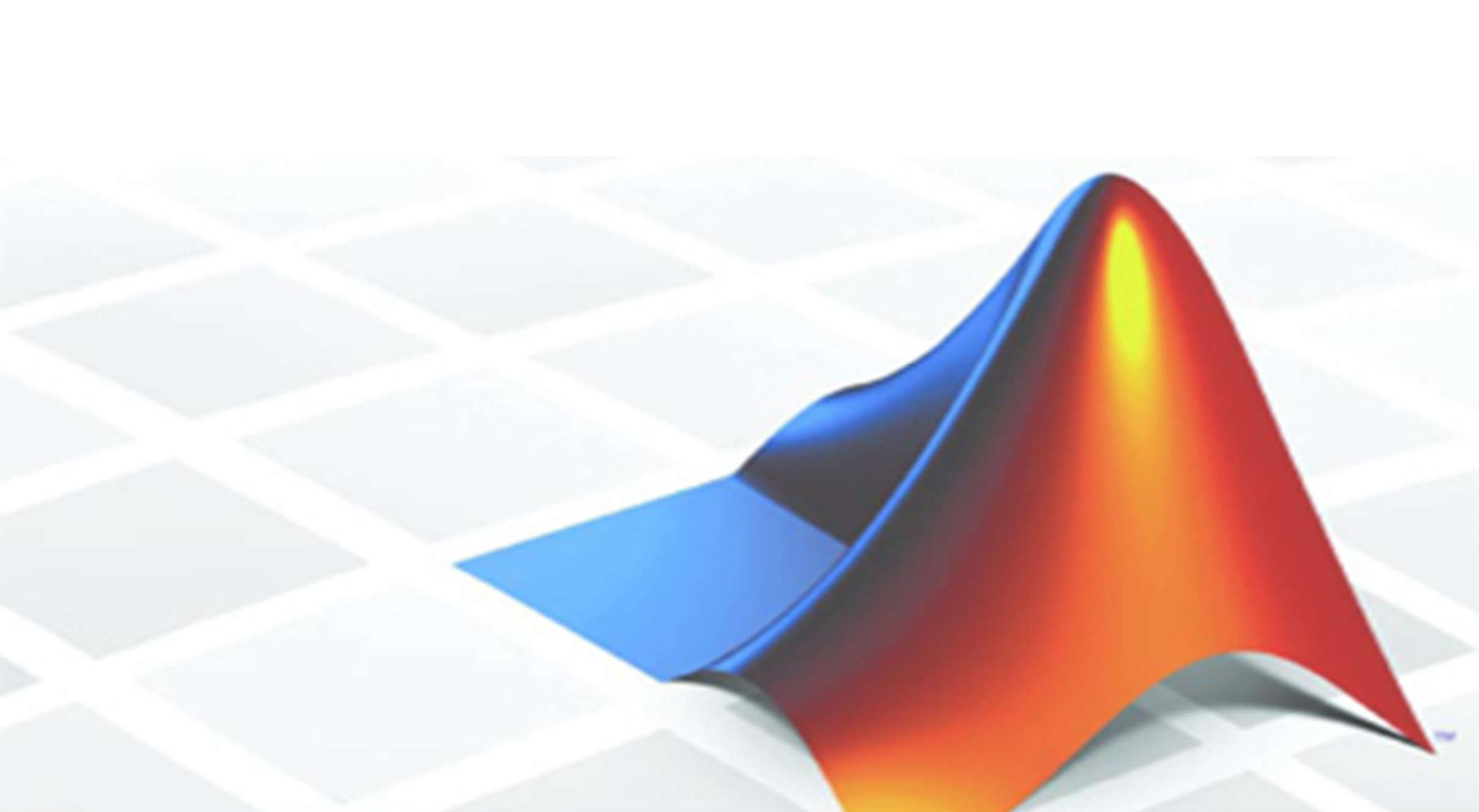


O que é Processamento Digital de Sinais (PDS)?



O que são Sistemas Lineares?





MATLAB®

MATLAB

Sistema interativo, baseado em matrizes de uso científico e de engenharia para computação numérica e visualização

MATrixLABoratory

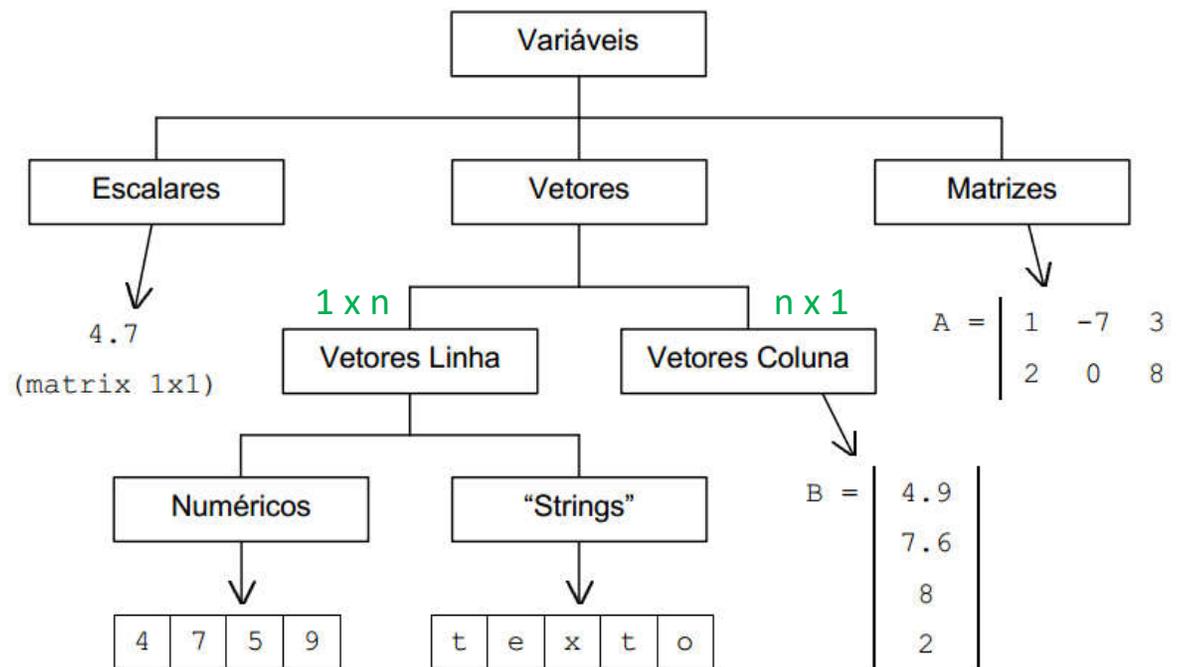
Seus **elementos básicos são matrizes** que não requerem dimensionamento. Ele permite implementar e resolver **problemas matemáticos muito mais rápida e eficientemente** que através de outras linguagens como C, Basic, Pascal ou Fortran.



Possui uma família de aplicativos específicos (**toolboxes**), que são coleções de funções usadas para resolver determinados problemas tais como: otimização, manipulação algébrica, redes neurais, **processamento de sinais**, simulação de sistemas dinâmicos, entre outros.

Variáveis

O MATLAB trabalha essencialmente com **um tipo de variável**: uma **matriz** contendo números, complexos ou não.



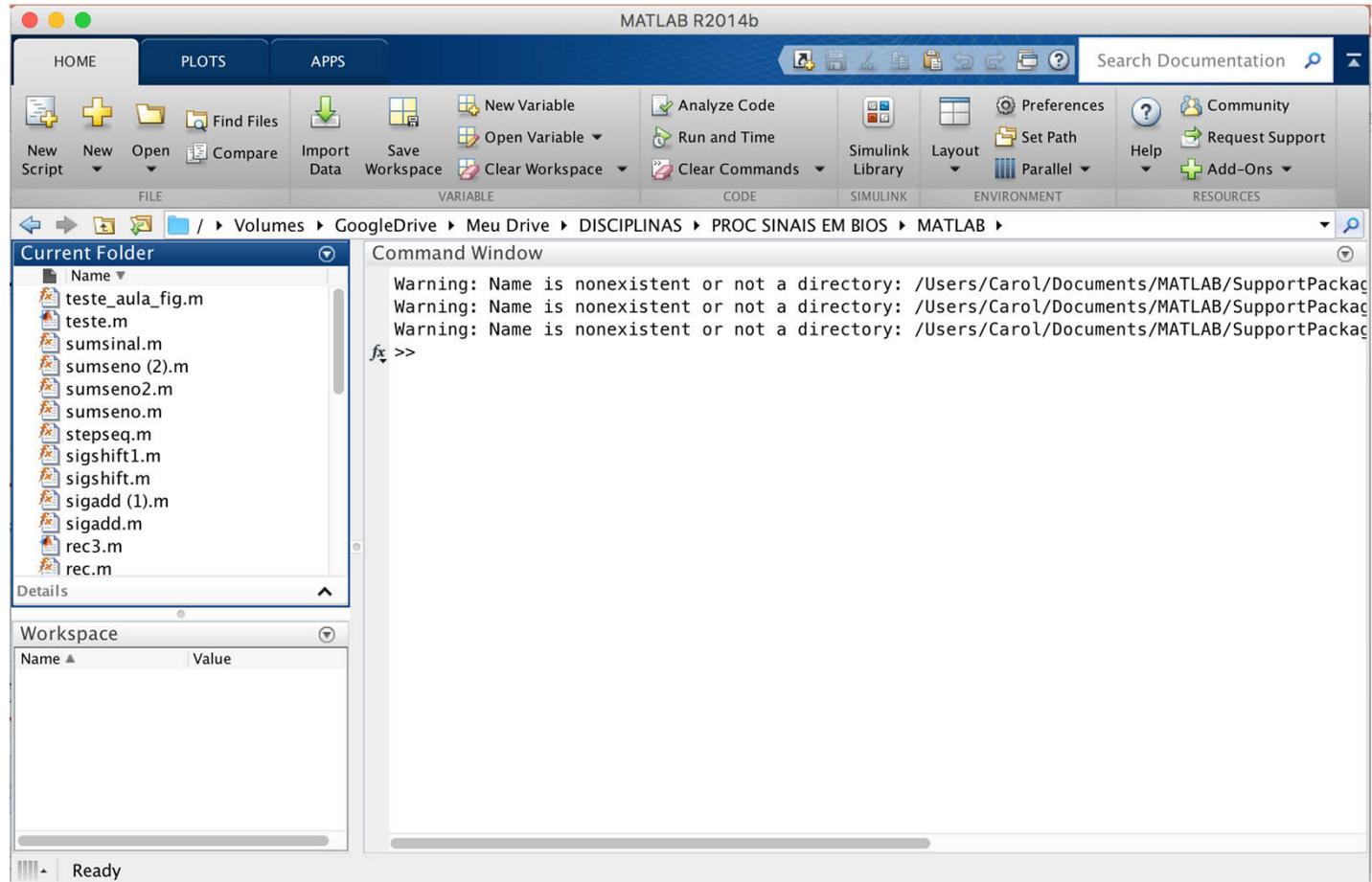
Variáveis permanentes

Existem algumas variáveis que são intrínsecas ao MATLAB e que não podem ser apagadas. Algumas são interessantes:

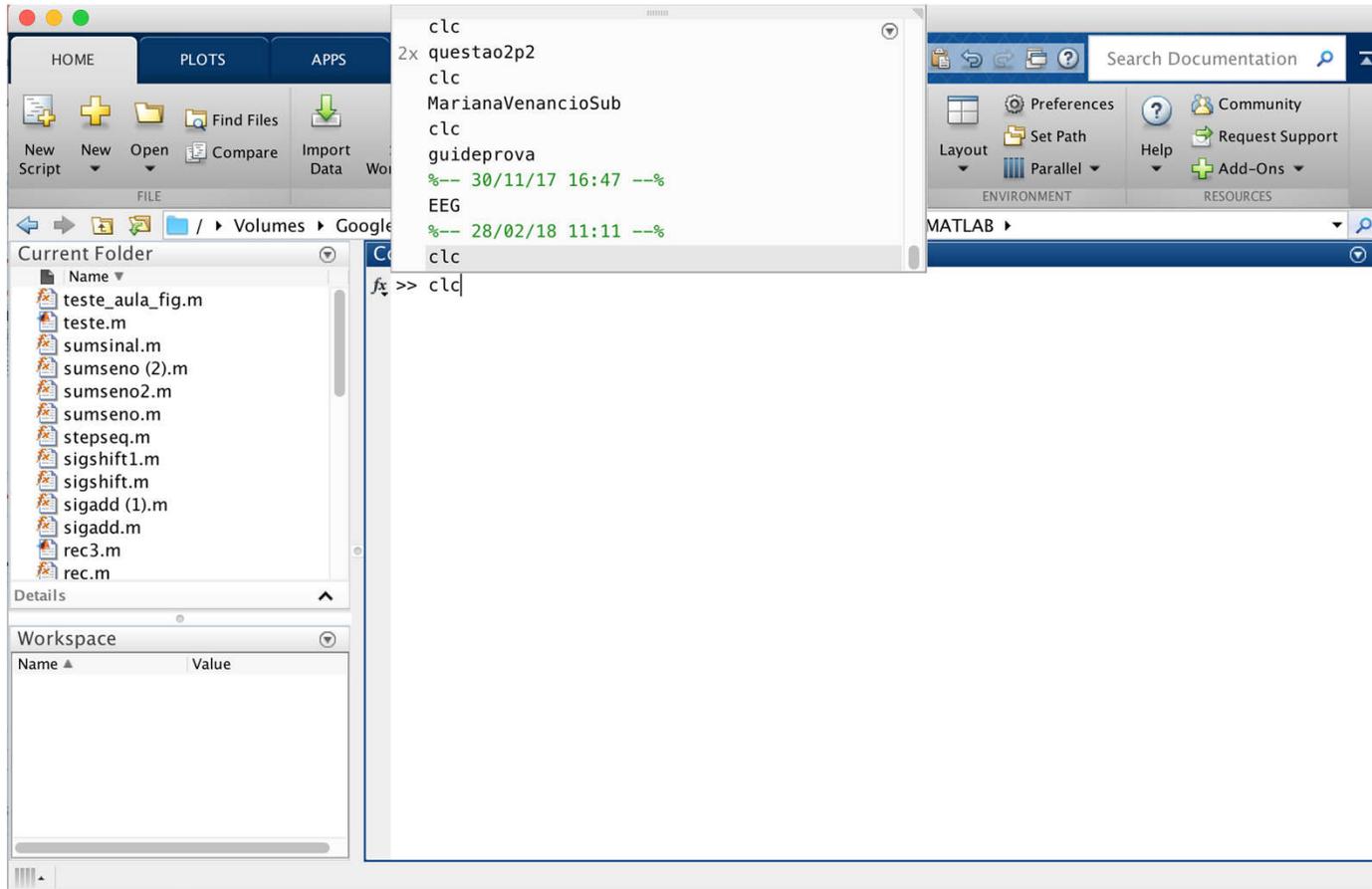
| | | | |
|----------------------|--|-----------------------|------------------------------------|
| <code>ans</code> | Resposta mais recente, que não foi atribuída a nenhuma variável. | <code>flops</code> | Contador de operações matemáticas. |
| <code>eps</code> | Precisão da máquina. | <code>NaN</code> | Not a Number (indeterminação) |
| <code>realmax</code> | Maior número de ponto flutuante. | <code>inf</code> | Infinito. |
| <code>realmin</code> | Menor número de ponto flutuante. | <code>computer</code> | Tipo de computador. |
| <code>pi</code> | 3,14159265358979 | <code>why</code> | Resposta sucinta. |
| <code>i, j</code> | Unidade imaginária | <code>version</code> | Versão do MATLAB. |

MATLAB®

Command window



Histórico de comandos



>> comando;

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

$A = ([1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]);$ $A = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$

Guarda os elementos na variável A

MA

MATLAB 7.8.0 (R2009a)

File Edit Debug Desktop Window Help

Current Directory: D:\Users\User\Documents\MATLAB

Shortcuts How to Add What's New

Current Directory

Command Window

```
>> A=[1 2 3;4 5 6;7 8 9];  
>> A=[1 2 3;4 5 6;7 8 9]
```

A =

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

f_x >>

Workspace

| Name | Value | Min | Max |
|------|---------------------|-----|-----|
| A | [1,2,3;4,5,6;7,8,9] | 1 | 9 |

Workspace

| Name | Value | Min | Max |
|------|---------------------|-----|-----|
| A | [1,2,3;4,5,6;7,8,9] | 1 | 9 |

```
>> A=[1 2 3;4 5 6;7 8 9];  
>> A=[1 2 3;4 5 6;7 8 9]
```

A =

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

f_x >>

```
13/07/11 10:53 --%  
a=[-12 6 -3;-4 2 -3;-18 9 -6]  
det(a)  
b=[2 0 1 0 1;0 2 0 1 2;1 0 1 0 1;0 1 0 1 1]  
rref(b)  
b=[2 0 1 0 1;0 2 0 1 2;1 0 1 0 0;0 0 1 0 1 1]  
rref(b)  
10/08/11 14:50 --%  
x=[0 1 0 1 0 1 0 1 0 1];  
complexq2(x,5,1,0)  
complexq2(x',5,1,0)  
02/03/12 15:25 --%  
help full  
padding  
02/05/12 11:11 --%  
A=[1 2 3;4 5 6;7 8 9];  
A=[1 2 3;4 5 6;7 8 9]
```

>> a=3 **Escalar**

>> x=[1,2,3] ou x=[1 2 3] **Vetor linha**

>> y=[1;2;3] **Vetor coluna**

>> A=[1,2,3;4,5,6] ou A=[1 2 3;4 5 6] **Matriz**

Exercicio 1

Testem as seguintes funções

- zeros(M,N)
- ones (M,N)
- eye(N)

Exercicio 2

Digite a seguinte seqüência de comandos e veja o que acontece com as variáveis na workspace

```
>> pi  
>> b=2/3  
>> 2/3
```

ans

Resposta mais recente, que não foi atribuída a nenhuma variável.

Exercicio 3

Digite os seguintes comandos e verifique se há diferença para o MATLAB do uso do i ou do j como numero imaginário

```
>> c=2+3*i  
>> d=2+3*j
```

MATLAB®

Operadores

| | | | |
|----|------------------------|----|-----------------------------|
| = | atribuição | == | igualdade |
| + | adição | - | subtração |
| * | multiplicação | .* | Multiplicação ponto a ponto |
| ^ | potência | .^ | Potencia ponto a ponto |
| / | divisão | ./ | Divisão ponto a ponto |
| <> | Operadores relacionais | && | E lógico (AND) |
| | OU lógico (OR) | ~ | NÃO lógico (NOT) |
| ' | transposição | .' | Transposição pontual |

O MATLAB possui uma vasta gama de funções matemáticas elementares, com seno (\sin), tangente (\tan), logaritmo (\log_{10}), etc. Por exemplo, para calcular o seno de 5 e guardar na variável x:

MA

Outros comando importantes

```
save (matlab.mat)
```

```
save nome_do_arquivo
```

```
save nome_do_arquivo nome_da_variável
```

```
load nome_do_arquivo
```

```
clear
```

```
clear nome_da_variável
```

Workspace

```
clc
```

Command window

Exercicio 4

- 1) Limpe a janela de comando
- 2) Limpe a workspace
- 3) Adicione as seguintes variáveis à workspace
 - a) `a=1;`
 - b) `b=2;`
 - c) `c=3;`
- 4) Salve todas as variáveis para o arquivo *abc.mat*
- 5) salve a variável `b` para o arquivo *incompleto.mat*
- 6) Limpe a workspace
- 7) Carregue o arquivo *abc.mat*
- 8) Limpe a workspace
- 9) Carregue o arquivo *incompleto.mat*

```
save (matlab.mat)
```

```
save nome_do_arquivo
```

```
save nome_do_arquivo nome_da_variavel
```

```
load nome_do_arquivo
```

```
clear
```

```
clear nome_da_variavel
```

Workspace

```
clc
```

Command window

MATLAB®

Comandos mais importantes

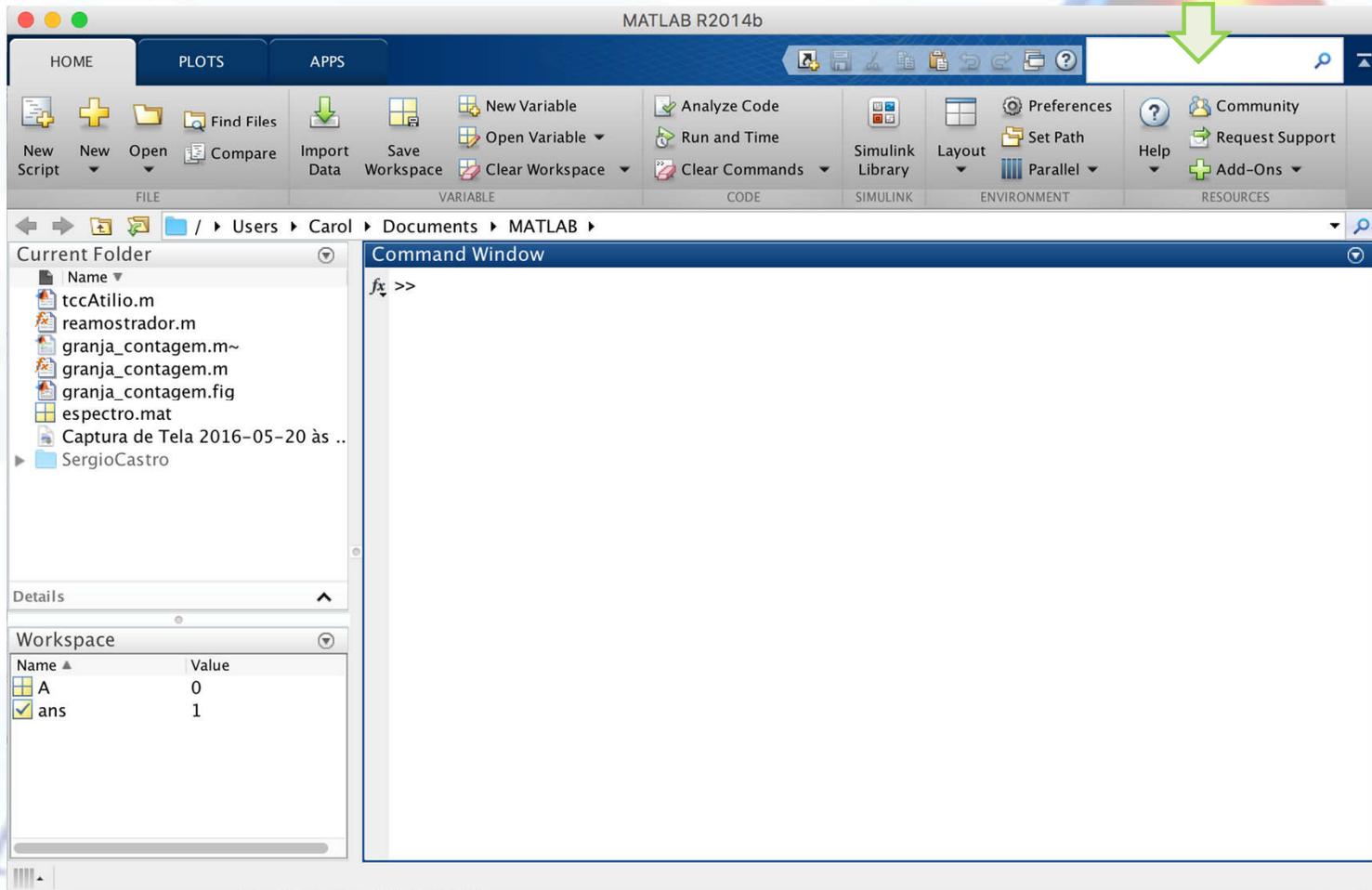
HELP

LOOKFOR

Exercicio 5

>> lookfor max

>> help who





MATLAB 7.6.0 (R2009a)

File Edit Debug Desktop Window Help

Current Directory: D:\Users\User\Documents\MATLAB

Shortcuts How to Add What's New

Current Directory Documents MATLAB

Command Window

```

>> who
Name Value Min Max
a 15 15 15
y -0.9589 -0.9589 -0.9589

```

Workspace

Command History

```

>> realsim
>> 1/3
>> %Podosome set numeroc complex
>> b=3+2i

```

who

List variables in workspace

who lists in alphabetical order all variables in the currently active workspace.

who lists in alphabetical order all variables in the currently active workspace along with their sizes and types. It also reports the totals for sizes.

Note If who or whos is executed within a nested function, the MATLAB software lists the variables in the workspace of that function and in the workspaces of all functions containing that function. See the Remarks section, below.

who ... VAR1 VAR2 restricts the display to the variables specified. The wildcard character '*' can be used to display variables that match a pattern. For instance, who A* finds all variables in the current workspace that start with A.

who -REGEXP PAT1 PAT2 can be used to display all variables matching the specified patterns using regular expressions. For more information on using regular expressions, type "doc regexp" at the command prompt.

Use the functional form of WHO, such as WHO('-file',FILE,V1,V2), when the filename or variable names are stored in strings.

S = WHO(...) returns a structure array containing the names of the variables in the workspace of there is an output.

Examples for pattern

```

who a*
who -regexp "b"
who -file fname

```

See also whos, clear

Overloaded methods: simulink.who

Reference page in Help

doc who



Help

Current Directory: D:\Users\User\Documents\MATLAB

Command Window

```

>> who
Name Value Min Max
a 15 15 15
y -0.9589 -0.9589 -0.9589

```

who

List variables in workspace

who lists in alphabetical order all variables in the currently active workspace.

who lists in alphabetical order all variables in the currently active workspace along with their sizes and types. It also reports the totals for sizes.

Note If who or whos is executed within a nested function, the MATLAB software lists the variables in the workspace of that function and in the workspaces of all functions containing that function. See the Remarks section, below.

who ... VAR1 VAR2 restricts the display to the variables specified. The wildcard character '*' can be used to display variables that match a

MATLAB

Operações com matrizes

Multiplicação por um escalar

$ab \Rightarrow a*b$ Escalar

$a\mathbf{x} \Rightarrow a*\mathbf{x}$ Vetor

$a\mathbf{X} \Rightarrow a*\mathbf{X}$ Matriz

Multiplicação vetor-vetor

Pode ser tanto um escalar quanto uma matriz. $\mathbf{x}(N \times 1)$ e $\mathbf{y}(1 \times M)$

$\mathbf{xy} \Rightarrow \mathbf{x}*\mathbf{y}$

$M, N \neq 1$ Vetor ou Matriz

$M=N=1$ Escalar

Operações com matrizes

Multiplicação matriz-vetor

$A(N \times M)$ e $x(M \times 1)$

$$y = A * x$$

$y(N \times 1)$ Vetor coluna

Multiplicação matriz-matriz

$A(N \times M)$ e $B(M \times P)$

$$C = A * B$$

$C(N \times P)$ Matriz

Operações com “arrays”

Multiplicação

$x.*y$ Array 1D

$X.*Y$ Array 2D

Exponenciação

$$a.^x \equiv \begin{bmatrix} a^{x_1} \\ a^{x_2} \\ \vdots \\ a^{x_N} \end{bmatrix}$$

$$a.^X \equiv \begin{bmatrix} a^{x_{11}} & a^{x_{12}} & \dots & a^{x_{1M}} \\ a^{x_{21}} & a^{x_{22}} & \dots & a^{x_{2M}} \\ \vdots & \vdots & \ddots & \vdots \\ a^{x_{N1}} & a^{x_{N2}} & \dots & a^{x_{NM}} \end{bmatrix}$$

Exercicio 6

1. Crie a matriz A

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

2. Atribua o elemento $A(2,2)$ à variável b e o elemento $A(3,3)$ à variável c.

3. Multiplique b e c e atribua a d

Elementos de uma matriz

Exercicio 7

1. Multiplique A por 2 e atribua a B

Multipliação por escalar

Exercicio 8

1. Crie um vetor coluna $x = (1 \ 1 \ 2)$
2. Obtenha $A * x$
3. Obtenha $A * A$
4. Obtenha $A . * A$
5. Obtenha $A.^2$
6. Obtenha $x * x'$

Gerando vetores

Operador `inicio:fim` e `inicio:passo:fim`

```
» x = 1:8
```

```
» x = 1:1.5:8
```

Exercicio 9

1. Observe o que acontece gerando os vetores acima

Util para gerar gráficos, tabelas e vetores com espaçamento conhecido

Exemplo

```
» x=0:0.2:3;  
» y=exp(-x) + sin(x);
```

Exercício 10

1. Faça um gráfico de y

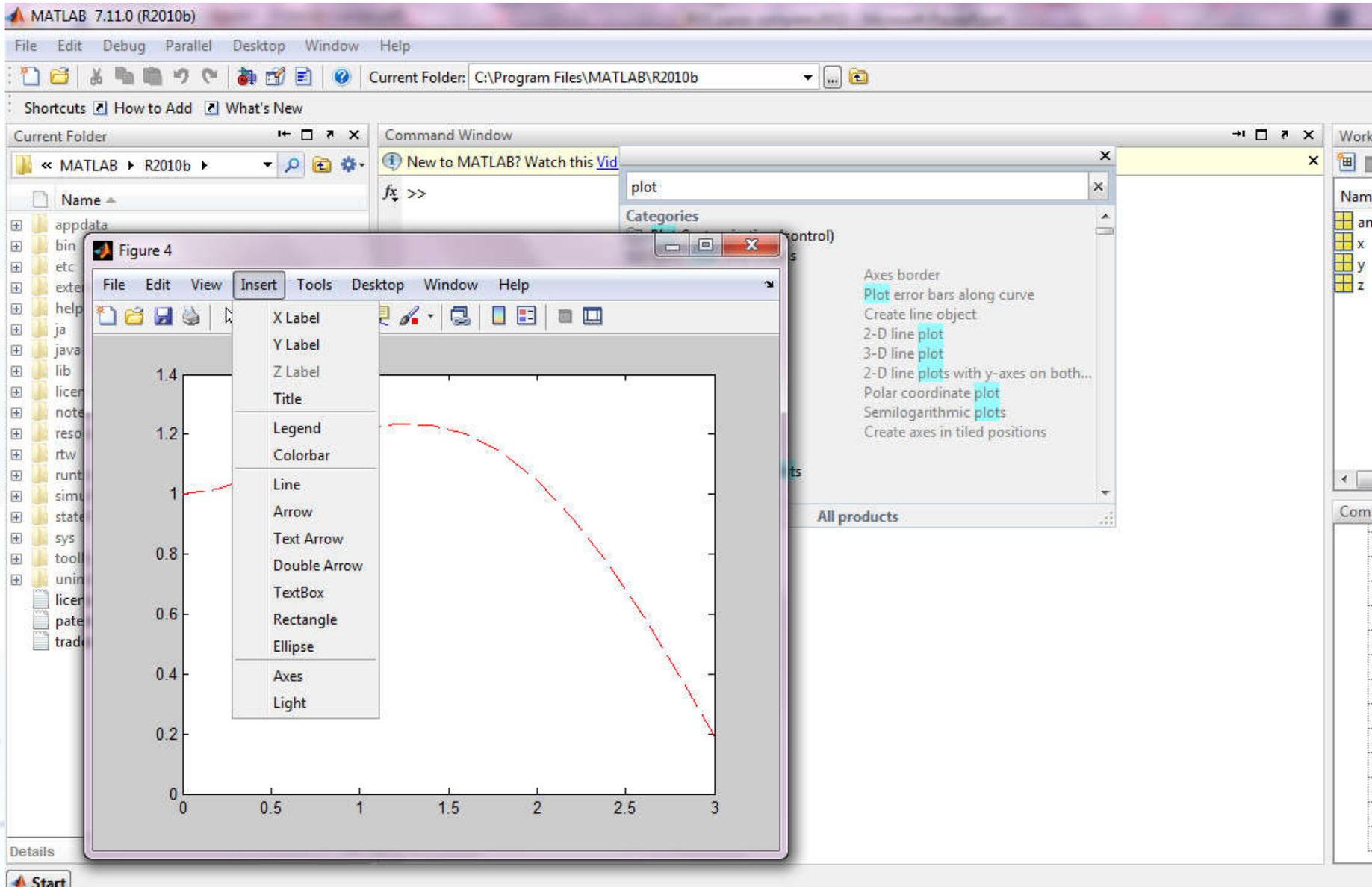
Comando plot

Gráficos

| | | | | | |
|---|------------|----|-----------|----|-------------|
| y | amarelo | w | branco | + | cruz |
| m | roxo | k | preto | - | sólida |
| c | azul claro | -- | tracejada | * | estrela |
| r | vermelho | . | ponto | : | pontilhada |
| g | verde | o | círculo | -. | traço ponto |
| b | azul | x | x | | |

```
>> figure(2);plot(x,y,'r')  
>> figure(3);plot(x,y,'--')  
>> figure(4);plot(x,y,'--r')
```

MATLAB®



Gráficos

| | | | |
|-----------------------|-------------------------------|-----------------------|--------------------------------|
| <code>title</code> | título do gráfico | <code>xlabel</code> | nome do eixo x |
| <code>text</code> | escreve no local especificado | <code>ylabel</code> | nome do eixo y |
| <code>gtext</code> | escreve texto no usando mouse | <code>grid</code> | desenha linhas de grade |
| <code>semilogx</code> | gráfico mono-log em x | <code>semilogy</code> | gráfico mono-log em y |
| <code>loglog</code> | gráfico di-log | <code>axis</code> | intervalo dos eixos no gráfico |

MATLAB®

Submatrizes

É possível construir matrizes maiores a partir de matrizes menores.

```
» A=[1 2 3;4 5 6;7 8 9];
```

```
» r=[13 32 5];
```

```
» A=[A;r]
```

```
C =
```

| | | |
|----|----|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 13 | 32 | 5 |

MATLAB®

Submatrizes

Se é possível “juntar” matrizes, também é possível “extrair” algumas partes.

Exercício 11

1. Construa as seguintes matrizes e selecione as partes indicadas

$$E = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 14 & 83 & 23 & 0 \\ \hline \end{array}$$

$$E = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 14 & 83 & 23 & 0 \\ \hline \end{array}$$

Selecione a 3ª linha toda (Sugestão: use o comando “:”)

$$A = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ \hline 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ \hline 14 & 83 & 23 & 0 \\ \hline \end{array}$$

Pegar os elementos da 2ª linha e 3ª e 4ª
colunas:

```
» A=[1 2 3 4; 5 6 7 8;9 10 11 12;14 83 23 0]
```

```
A =
```

```

     1     2     3     4
     5     6     7     8
     9    10    11    12
    14    83    23     0
```

```
» A(2,[3 4])
```

```
ans =
```

```

     7     8
```

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 14 & 83 & 23 & 0 \end{pmatrix}$$

```
» A([1 2 3],[2 3])
```

```
ans =  
     2     3  
     6     7  
    10    11
```

ou ainda um comando equivalente:

```
» A(1:3,2:3)
```

```
ans =  
     2     3  
     6     7  
    10    11
```

```
» A(3,:)
```

```
ans =  
     9    10    11    12
```

Operadores relacionais

Existem seis operadores relacionais no MATLAB. São eles:

| Símbolo | Operador |
|---------|--------------------|
| < | menor que |
| <= | menor ou igual que |
| > | maior que |
| >= | maior ou igual que |
| == | igual |
| ~= | não igual |

O resultado da comparação é 1 se verdadeiro e 0 se falso. por exemplo:

```
» 2 + 2 == 4
```

```
ans =
```

```
1
```

MATLAB

Operadores relacionais

No caso de comparação com matrizes, o resultado será uma matriz de 0 e 1:

```
» x=[2 3 4;5 2 7;9 2 7]
```

```
x =
```

| | | |
|---|---|---|
| 2 | 3 | 4 |
| 5 | 2 | 7 |
| 9 | 2 | 7 |

```
» x>4
```

```
ans =
```

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 0 | 1 |

MATLAB®

Operadores relacionais

Exercicio 12

1. Considere a matriz E criada anteriormente. Atribua a um vetor s todos os elementos maiores que 5

$$E = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ \hline 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ \hline 14 & 83 & 23 & 0 \\ \hline \end{array}$$

Operadores relacionais

```
>> E=[1 2 3 4;5 6 7 8;9 10 11 12;14 83 23 0]
```

```
E =
```

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 14 | 83 | 23 | 0 |

```
>> s=E(E>5)
```

```
s =
```

| |
|----|
| 9 |
| 14 |
| 6 |
| 10 |
| 83 |
| 7 |
| 11 |
| 23 |
| 8 |
| 12 |

```
>> x=E>5
```

Possibilidade para salvar a matriz e não um vetor

```
x =
```

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |

```
>> s=E.*x
```

```
s =
```

| | | | |
|----|----|----|----|
| 0 | 0 | 0 | 0 |
| 0 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 14 | 83 | 23 | 0 |

Outros operadores que podem ser úteis

Eis abaixo alguns exemplos de funções básicas no MATLAB. Qualquer dúvida em como usá-las basta usar o comando **help**,

| | | | |
|--------------------|-------------------|--------------------|-------------------------------------|
| <code>exp</code> | <code>e</code> | <code>poly</code> | polinômio característico |
| <code>log</code> | logaritmo natural | <code>det</code> | determinante |
| <code>log10</code> | logaritmo base 10 | <code>abs</code> | módulo |
| <code>find</code> | índice de matriz | <code>sqrt</code> | raiz quadrada |
| <code>max</code> | máximo valor | <code>real</code> | parte real de número complexo |
| <code>min</code> | mínimo valor | <code>imag</code> | parte imaginária de número complexo |
| <code>mean</code> | média aritmética | <code>conj</code> | conjunto de número complexo |
| <code>std</code> | desvio padrão | <code>round</code> | arredondar |

MATLAB

Polinômios

| | | | |
|---------|-----------------------------|---------|--------------|
| roots | raízes do polinômio | deconv | divisão |
| polyval | valor do polinômio no ponto | polyder | derivadas |
| conv | multiplicação | polyfit | melhor curva |

$$p=x^3-6x^2-72x-27$$

```
>> %  
>> %  
>> %  
>> p=[1 -6 -72 -27];  
>> roots(p)
```

```
ans =  
  
12.1229  
-5.7345  
-0.3884
```

MATLAB®

Exercicio 13

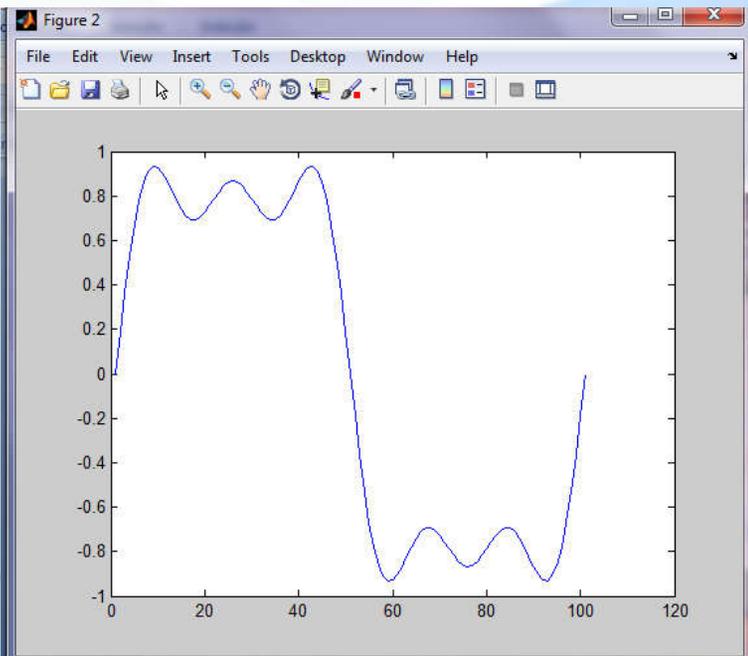
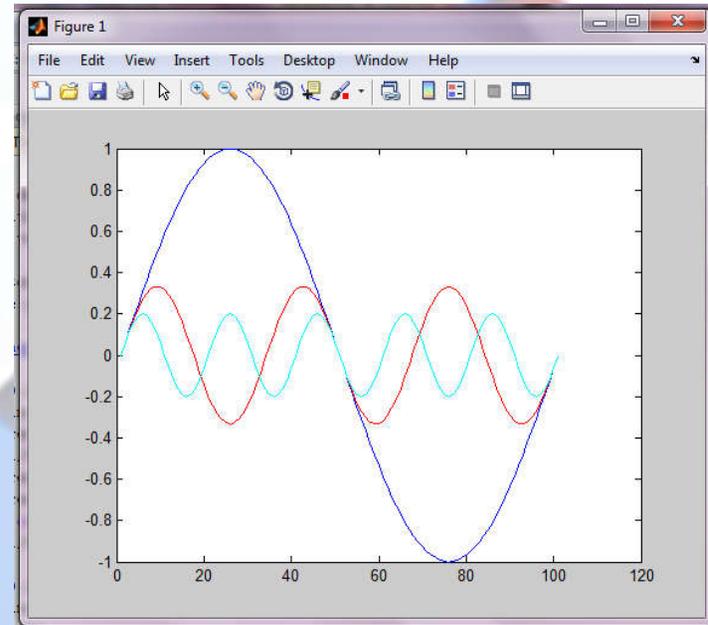
Considere a soma de senoides abaixo

$$x(t) = \sin(2\pi t) + \frac{1}{3} \sin(6\pi t) + \frac{1}{5} \sin(10\pi t)$$

usando o MATLAB queremos gerar “amostras” de $x(t)$ no intervalo 0:0.01:1

MATLAB®

```
>> t=0:0.01:1;  
>> s1=sin(2*pi*t);  
>> plot(s1)  
>> s2=(1/3)*sin(6*pi*t);  
>> hold on;plot(s2,'r')  
>> s3=(1/5)*sin(10*pi*t);  
>> hold on;plot(s3,'c')
```



```
>> sn=s1+s2+s3;  
>> figure(2);plot(t,sn)
```