

1.12 Criação e manipulação de interface gráfica

Prof. Dr. Sidney Bruce Shiki

e-mail: bruce@ufscar.br

Prof. Dr. Vitor Ramos Franco

e-mail: vrfranco@ufscar.br

UFSCar – Universidade Federal de São Carlos

DEMec - Departamento de Engenharia Mecânica

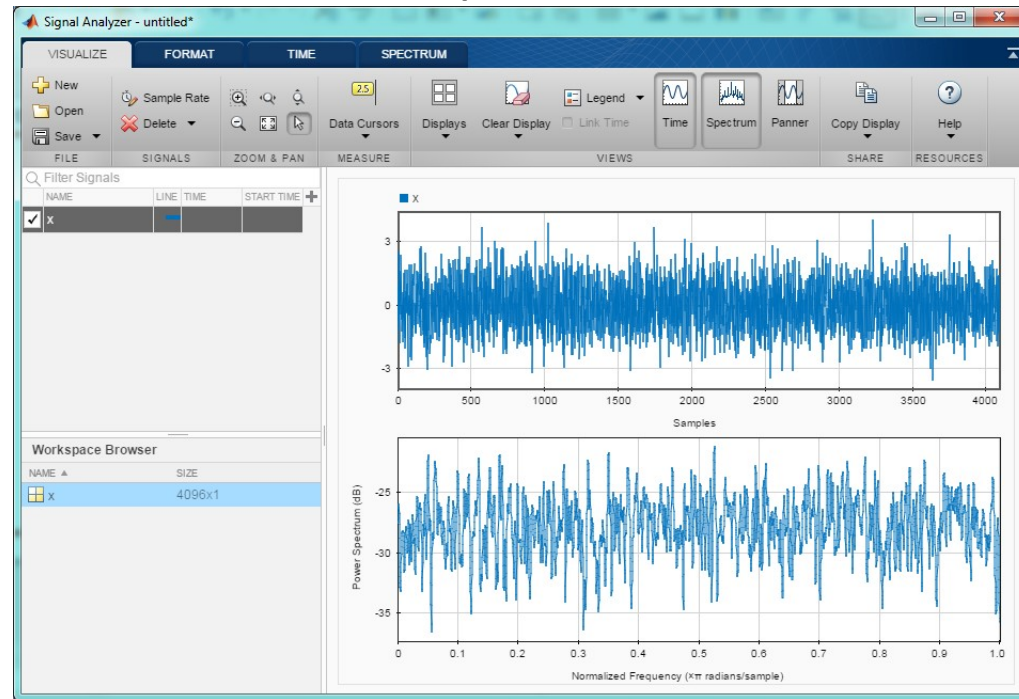
- Introdução
- Comandos para GUI
- MATLAB GUIDE
- Geração de animações simples
- Exercícios

- O MATLAB tradicional programado com *scripts* é adequado para soluções de problemas numéricos;
- No entanto, usuários menos vivenciados com linguagens de programação podem preferir uma interface gráfica (GUI, do inglês *graphical user interface*) para interagir com as rotinas implementadas em MATLAB.

Código: legível apenas para o programador

```
function popupsig_Callback(hObject, eventdata, handles)
sigtype = get(hObject,'Value'); % indica o tipo do sinal
%disp(sigtype);
% Propriedades do sinal gerado
amp = str2num(get(handles.editamp,'String'));
maxtime = str2num(get(handles.edittime,'String'));
tao = str2num(get(handles.edittao,'String'));
f = str2num(get(handles.edittfreq,'String'));
% Propriedades de amostragem do sinal
Fs = str2num(get(handles.editFs,'String'));
t = 0:1/Fs:maxtime;
% Construção do sinal
u = zeros(length(t),1); % inicializa
switch sigtype
case 1 % nada
case 2 % senóide
u = amp*sin(2*pi*f*t);
case 3 % onda quadrada
u = amp*square(2*pi*f*t);
case 4 % dente de serra
u = amp*sawtooth(2*pi*f*t);
case 5 % porta
set(handles.text4,'Visible','on');
set(handles.edittao,'Visible','on');
for k = 1:length(t)
if t(k)<=tao
u(k) = amp;
else
u(k) = 0;
end
end
case 6 % aleatório
u = amp*randn(length(t),1);
case 7 % outros
% Transforma em uma Handle Function a função digitada
funcoutros = inputdlg('Insira uma função (colocar a variável
independente como "t"):');
funcoutros = strcat('funcao = @(t) ',funcoutros);
eval(funcoutros(1)); % joga na workspace a função
for aux = 1:length(t)
u(aux) = funcao(t(aux));
end
case 8
[arquivo,caminho] = uigetfile('*.txt','Carregue o arquivo com o
sinal');
sig = importdata([caminho arquivo]);
u = sig(:,1);
t = linspace(0,(length(u)-1)/Fs,length(u));
end
```

GUI: feita para ser intuitiva e para Usuários interagirem com as rotinas computacionais



- Nesta aula serão abordados:
 - Comando para gerar GUI simples;
 - Ferramenta GUIDE para geração de interfaces gráficas mais complexas;
 - Geração de animações simples em MATLAB para ilustrar resultados de simulações.

- Caixa de diálogo para entrada de dados (inputdlg):

```
respostas=inputdlg(prompt,nome,nlinhas,respdefault,options);
```

prompt – Célula com strings dos campos a serem colocados

nome – nome da caixa de diálogo

nlinhas -número de linhas a se preencher na caixa

respdefault – célula com respostas default da caixa

options – opções adicionais da caixa de diálogo

- Exemplo de criação (inputdlg):

```
prompt = {'Campo 1', 'Campo 2', 'Campo 3'};  
name = 'Caixa de diálogo';  
numlin = 1;  
defaultanswer = {'0', '0', '0'};  
options.Resize='on'; % Permite ajustar tamanho  
answer=inputdlg(prompt,name,numlin,defaultanswer  
,options);
```

- Caixa de mensagem (msgbox):

```
msgbox(texto,titulo);
```

texto – texto a ser exibido na caixa e mensagem

titulo – nome da caixa de mensagens

- Exemplo de criação (msgbox):

```
texto = {'Os dias do curso de MATLAB  
são:', 'Terça-feira (18:00-20:00)', 'Quinta-feira  
(18:00-20:00)'};  
titulo = 'MSG';  
msgbox(texto,titulo);
```


- Exercício 1 – Faça uma caixa de diálogo de entrada que receba 5 números. Analise na sua rotina estes 5 números e retorne o maior deles em uma caixa de mensagens com um texto indicando isso. Obs.: a resposta da caixa de diálogo é uma char, para lidar com números converta cada elemento usando o comando `str2num()`.



- Caixa de diálogo para selecionar item em lista de dados (`listdlg`):

```
selecao=listdlg('PromptString',prompt,'ListString',lista);
```

`prompt` – Nome da caixa de diálogo

`lista` – Célula com strings das opções da lista

`selecao` – retorna índice do item selecionado

- Exemplo de criação (`listdlg`):

```
lista = {'Banana','Maçã','Atemóia','Cereja'};  
prompt = 'Selecione uma fruta:';  
selecao=listdlg('PromptString',prompt,'ListString',lista);
```

- Caixa de diálogo para selecionar itens com botão (`questdlg`):

```
selecao=questdlg(questao,titulo,bot1,bot2,bot3,bdefault);
```

`questao` – String com a questão

`titulo` – Título da caixa de diálogo

`bot1, bot2, bot3` – nome dos botões

`bdefault` – botão *default* selecionado

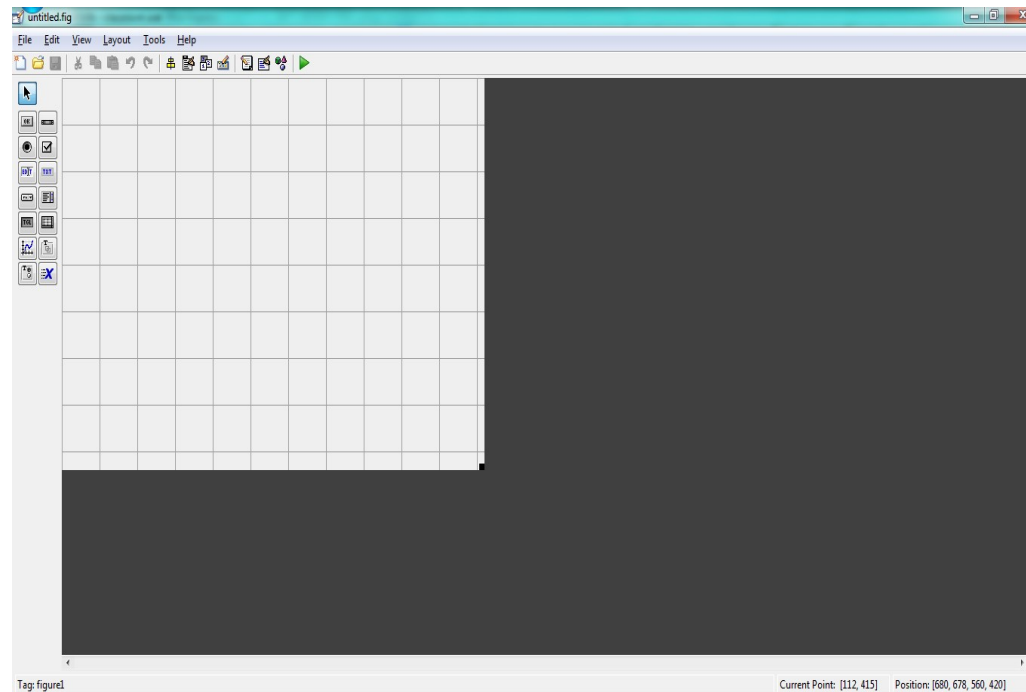
- Caixa de diálogo para selecionar itens com botão (questdlg):

```
selecao=questdlg(questao,titulo,bot1,bot2,bot3,bdefault);
```

- Exemplo de criação (questdlg):

```
resposta = questdlg('Qual sua cor favorita?', ...  
                    'Pergunta', ...  
                    'Vermelho', 'Verde', 'Azul', 'Verde');
```

- O MATLAB possui uma ferramenta de desenvolvimento de interfaces gráficas simples chamada guide;
- Para chamar a mesma digite `guide` na área de comandos;





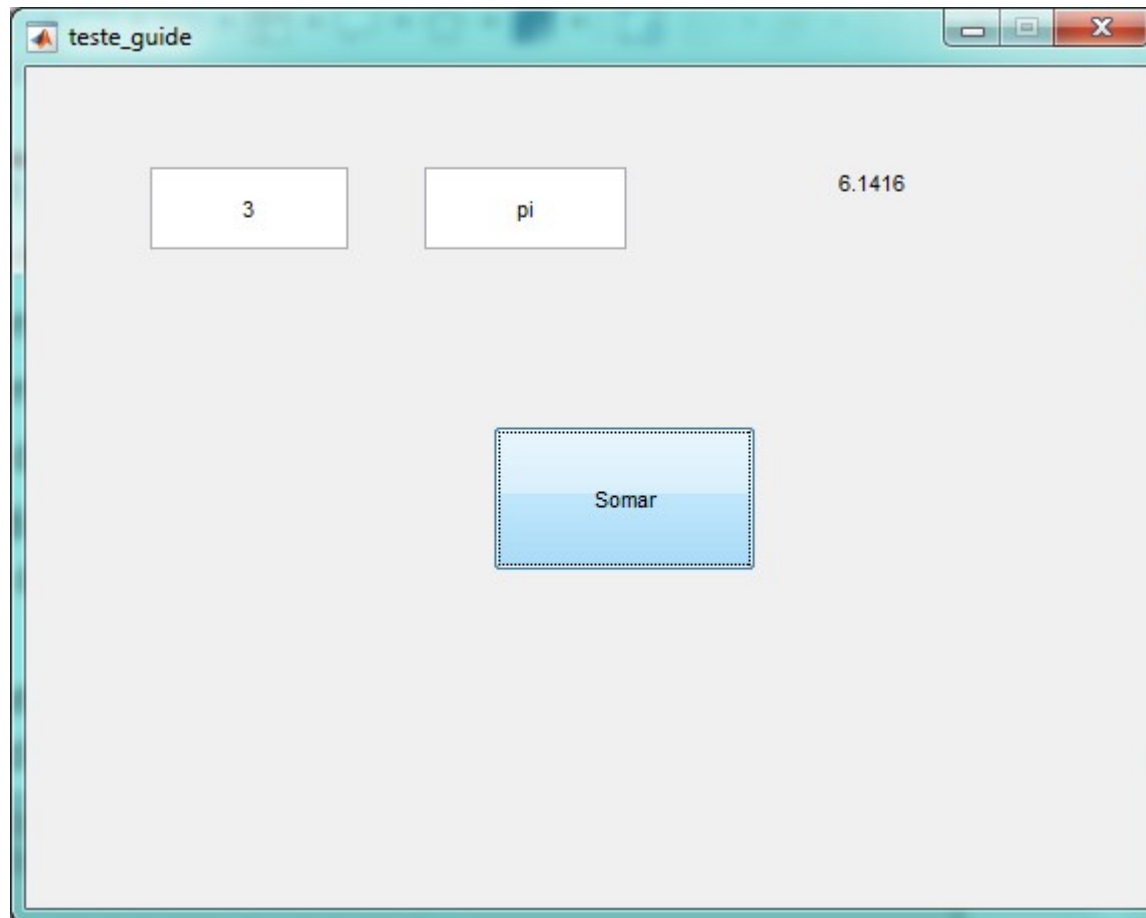
- A forma de programação do guide é um pouco diferente da forma estruturada empregada nos scripts;
- As rotinas escritas para o guide são orientadas a eventos;
- Quando você clica em um botão ele responde, em outros instantes o programa fica em espera.



- Vamos fazer um programa simples no guide;
- Insira dois elementos do tipo Edit text na interface;
- Coloque um elemento do tipo Static text;
- Coloque um elemento tipo Push Button;
- Salve o arquivo do guide;
- Clique o botão direito do mouse em cada elemento para verificar as tags de cada um desses elementos em Property inspector;
- Clique com o botão direito no Push Button adicionado e clique em View Callbacks>Callback, depois insira o código:

```
num1 = str2num(get(handles.edit1, 'String'));  
num2 = str2num(get(handles.edit2, 'String'));  
num3 = num1+num2;  
set(handles.text2, 'String', num2str(num3));
```

- Se executou corretamente as tarefas, você gerou uma calculadora básica:



Geração de animações simples

- Em diversos casos, ao rodar simulações ou análises numéricas, verifica-se a necessidade de usar recursos gráficos mais avançados;
- Em engenharia quando deseja-se mostrar movimento de estruturas e máquinas isso pode ser útil;

Geração de animações simples

- O comando `drawnow` pode ser usado para gerar simples animações;
- O mesmo é usado para plotar quadro-a-quadro uma animação (de modo similar ao que você pode fazer com um bloco de folhas de papel e caneta).

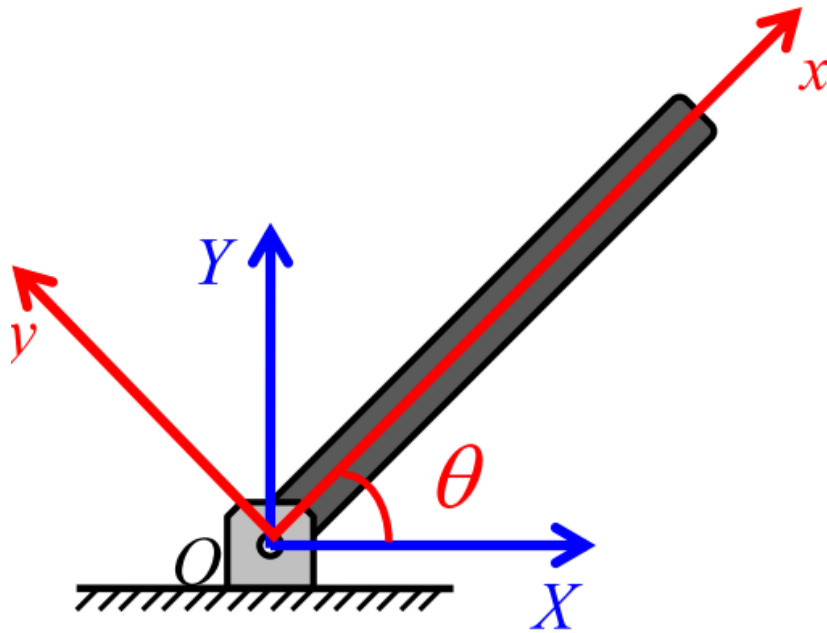


- Exemplo de criação:

```
t = 0:0.1:22;      % Tempo
x = 20*t;          % Coordenada horizontal
y = 100*t-0.5*9.8*t.^2; % Coordenada vertical

% Animacao
for k = 1:length(t)
    plot(x(1:k),y(1:k),'b','linewidth',2);hold on;
    plot(x(k),y(k),'bo','markersize',10);
    text(50,500,['Tempo =',num2str(t(k))]);
    xlim([0 500]);ylim([-200 600]);
    grid on;
    drawnow;
    hold off;
end
```

- Exercício 02 – Resolva a equação diferencial abaixo e faça uma animação do movimento do elo rotativo em torno do seu ponto de rotação.



$$I\ddot{\theta} + mg \frac{L}{2} \cos \theta = T$$

$$I = 0.2147 \quad [\text{kg} \cdot \text{m}^2]$$

$$m = 1.0064 \quad [\text{kg}]$$

$$L = 0.8 \quad [\text{m}]$$

$$T = 2.8 \quad [\text{N} \cdot \text{m}]$$

$$g = 9.8 \quad [\text{m/s}^2]$$

$$\theta(t = 0) = -\pi / 2 \quad [\text{rad}]$$

$$\dot{\theta}(t = 0) = 0 \quad [\text{rad/s}]$$

Perguntas ?