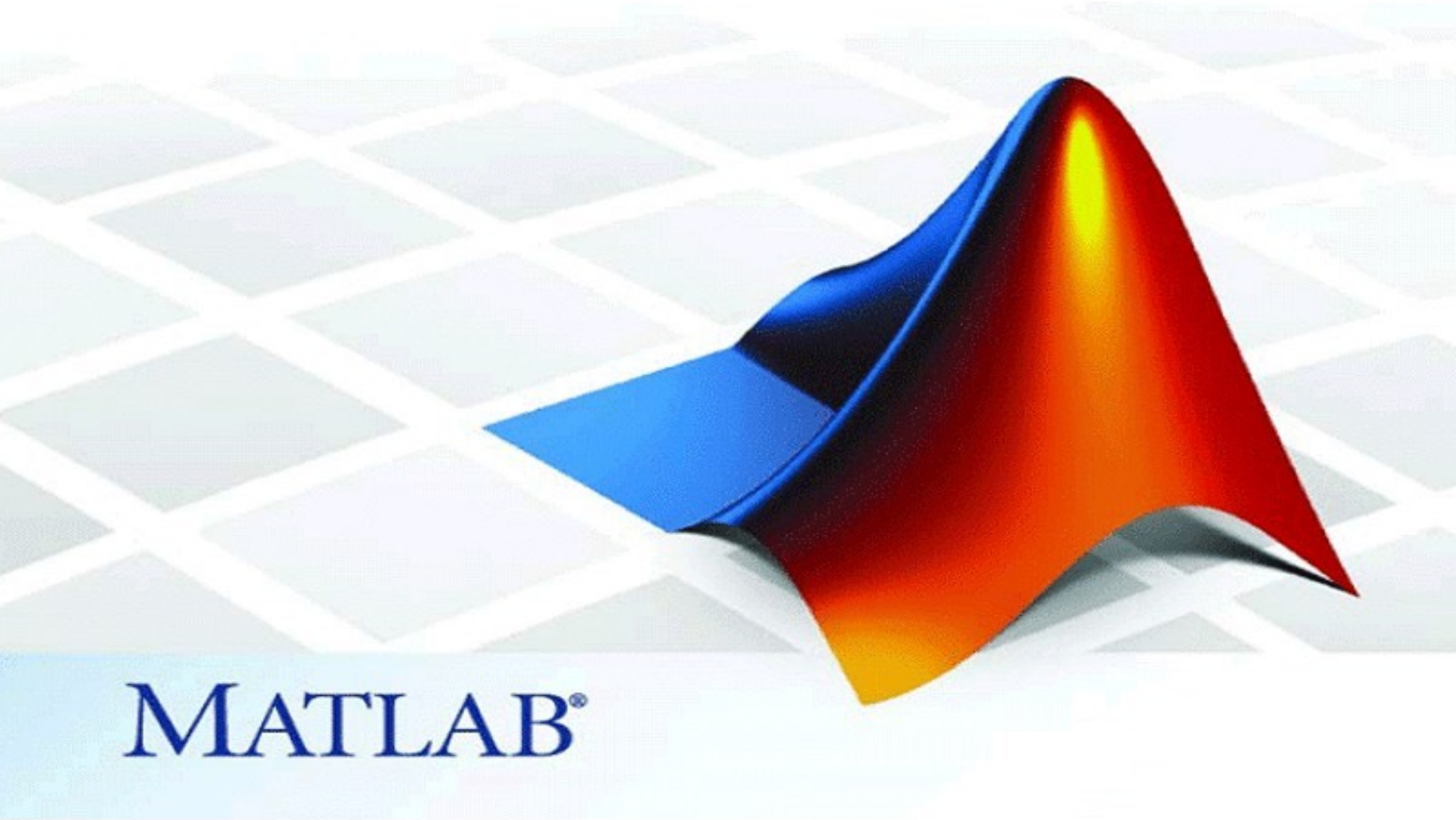
Tutorial MATLAB



Prof. Fernando Passold
Controle Automático

Comandos que serão explorados:

- Operações aritméticas básicas
- plot
- plotyy
- simulink
- tf
- poly
- Criação de vetores/matrizes
- zpk
- Multiplicação de tf's. Paralelismo de tf's.
- feedback
- step
- lsim

Introdução ➠ Op. básicas

- **Operação básica**

Comando de prontidão

```
>>variável=expressão
```

- **Operações aritméticas: +, -, /, *, ^, E**

```
>>z=3+4*i
z =
   3.0000 + 4.0000i

>>Inf
ans =
   Inf

>>0/0
ans =
   NaN
```

FIGURA A.6 Três variáveis predefinidas *i*, *Inf* e *NaN*.

- **Outras:**

```
>>who
Your variables are:
A    M    ans    m    z
```

FIGURA A.7 Usando a função **who** para exibir variáveis.

```
>>whos
```

Name	Size	Bytes	Class	Attributes
A	2x2	32	double	
M	1x2	16	double	
ans	1x1	8	double	
m	1x3	24	double	
z	1x1	16	double	complex

FIGURA A.8 Usando a função **whos** para exibir variáveis.

```
>>clear A
>>who
Your variables are:
M    ans    m    z
```

FIGURA A.9 Removendo a matriz **A** do workspace.

Introdução ➠ Constantes pré-definidas…

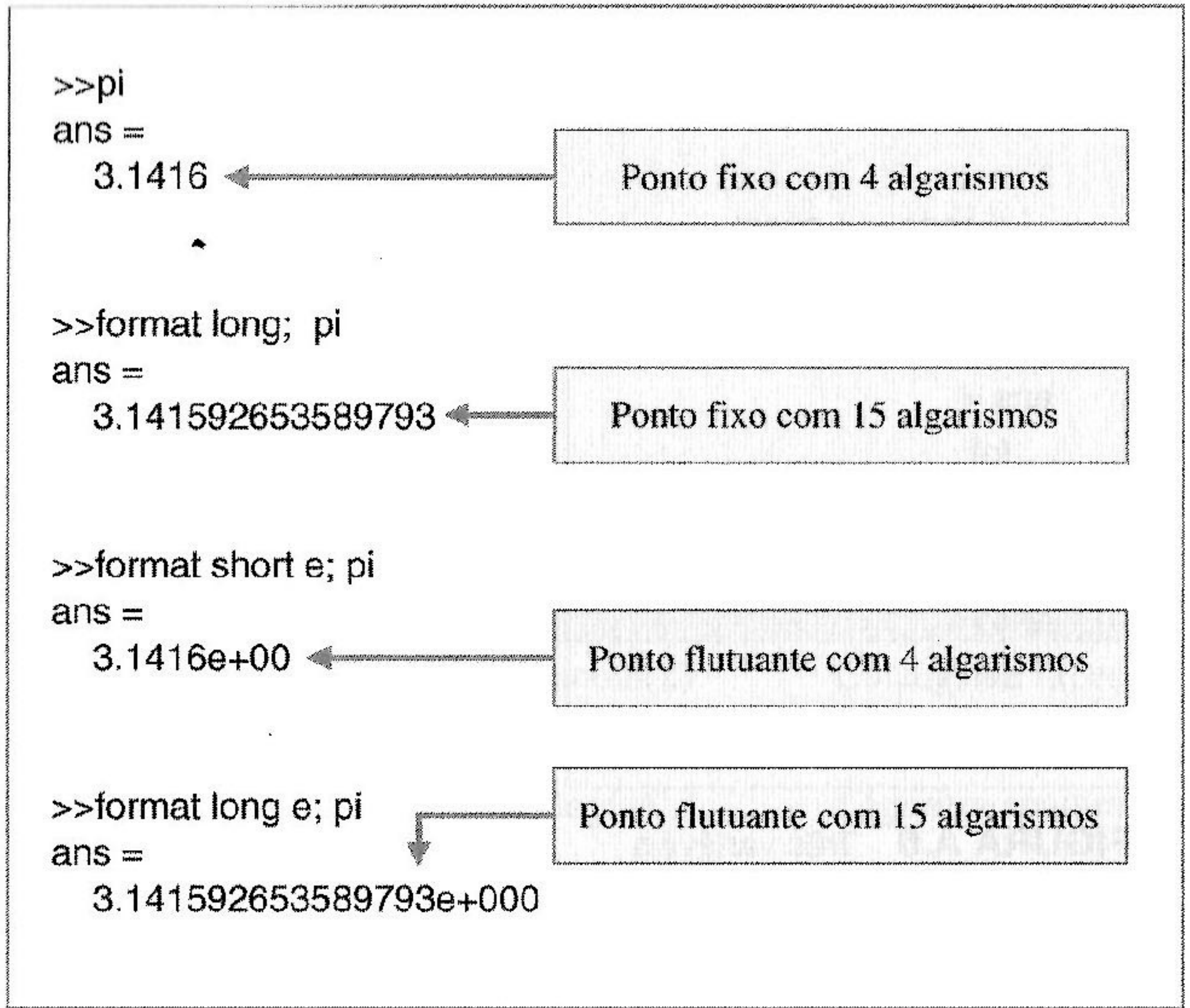


FIGURA A.10 O controle de formato de saída ilustra quatro formas da saída.

```
>>WHO
??? Undefined
function or variable 'WHO'.

>>Who
??? Undefined function or
variable 'Who'.
```

FIGURA A.11 Os nomes de funções são sensíveis à caixa.

Comandos básicos:

- **who**: lista (resumida) de variáveis da atual seção de trabalho. Ex.:

```
>> who
Your variables are:
G     a     ans   b     c     den   ftma  ftmf  num   polo  tout
>>
```

- **whos**: lista (com detalhes) das variáveis atuais:

```
>> whos
  Name       Size            Bytes  Class     Attributes

  G          1x1              1249  tf
  a          1x1                 8  double
  ans        1x1                 8  double
  b          1x1                 8  double
  c          1x1                 8  double
  den        1x5                40  double
  ftma       1x1              1249  tf
  ftmf       1x1              1249  tf
  num        1x4                32  double
  polo       1x1                16  double    complex
  tout      53x1               424  double

>>
```

Outros comandos:

- **clear**: limpa todas as variáveis do ambiente. Ex.:

```
>> clear
>> whos
>>
```

Cuidado com este comando. É irreversível. Usar antes “save"

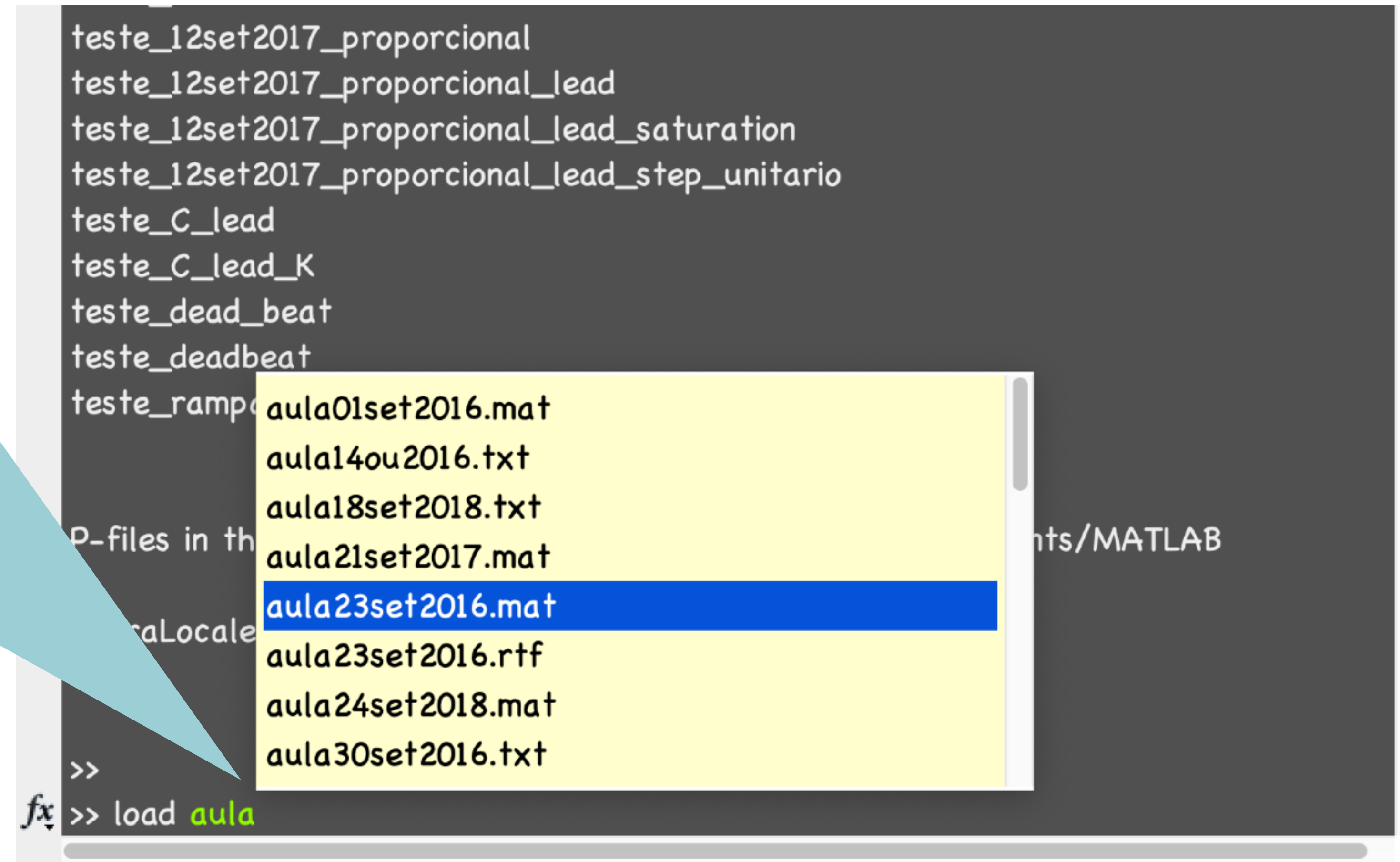
- **pwd**: verifica (na CLI) diretório de trabalho atual. Ex.

```
>> pwd
ans =
    '/Users/fernandopassold/Documents/MATLAB'
>>
```

Outros comandos:

- **load**: carrega variáveis de arquivo *.MAT criado anteriormente com comando “save”. Ex.:

Tecla “TAB" abre menu pull-down




```
>> load aula23set2016.mat
>> who

Your variables are:

BoG          K_c          den          ftma_reduzido  num
C            T            den_c        ftmf           num_c
G            ans          ftma         ftmf_reduzido

>>
```

Outros comandos:

- **what**: lista arquivos compatíveis com MATLAB no diretório atual. Ex.:

```
>> what

MATLAB Code files in the current folder /Users/fernandopassold/Documents/MATLAB

amostrado_quadrado       pattern7seg               show_robot2
amostrado_quadrado2      polos_reais               sintese_maq_sinc_mode3
angular_contrib          print7seg                 sintese_maq_sinc_mode3b
angulos                  print7seg2                square_with_noise
angulos2                 print7seg3                tabela_jk
derivada_sinal_ruidoso   quadrada1                 teste_le_arquivo_texto2
example_9_3              quadrada2                 teste_robot
example_9_4              que03                     teste_sonar
example_9_5              que_senoide_amostrada     teste_sonar1
example_9_6              questao_2c                trab1_que02
gera_ondas               resposta_que2_trab1       trab_1_que_02
palavras7seg             show_robot

MAT-files in the current folder /Users/fernandopassold/Documents/MATLAB

06out2015
16_04_2019_controle3
17out2015
31out2015
ControlSystemDesignerSession
ControlSystemDesignerSession_planta3_2018_2
ControlSystemDesignerSession_planta4_2018_2
PD_design_2016_2
PD_ensaio_P_REC_2016_2
aula21set2017
aula24set2018
controle2_aula09042019
controle3_prova1_2018_1
ensaio_Prova_21jun2017
>>
```

Funções matemáticas

sin(x)	Seno	acoth(x)	Cotangente hiperbólica inversa
sinh(x)	Seno hiperbólico	exp(x)	Exponencial
asin(x)	Arco-seno	log(x)	Logaritmo natural
asinh(x)	Arco-seno hiperbólico	log10(x)	Logaritmo comum (base 10)
cos(x)	Cosseno	log2(x)	Logaritmo na base 2 e separa números reais em mantissa e expoente
cosh(x)	Cosseno hiperbólico	pow2(x)	Potência de base 2 e escala números reais
acos(x)	Arco-cosseno	sqrt(x)	Raiz quadrada
acosh(x)	Arco-cosseno hiperbólico	nextpow2(x)	Próxima potência de 2
tan(x)	Tangente	abs(x)	Valor absoluto
tanh(x)	Tangente hiperbólica	angle(x)	Ângulo de fase
atan(x)	Arco-tangente	complex(x,y)	Constrói dados complexos a partir das partes real e imaginária
atan2(x)	Arco-tangente de quatro quadrantes	conj(x)	Conjugado complexo
atanh(x)	Arco-tangente hiperbólica	imag(x)	Parte imaginária do número complexo
sec(x)	Secante	real(x)	Parte real do número complexo
sech(x)	Secante hiperbólica	unwrap(x)	Ângulo de fase base
asec(x)	Secante inversa	isreal(x)	Verdadeiro para vetores reais
asech(x)	Secante hiperbólica inversa	cplxpair(x)	Ordena números em pares conjugados complexos
csc(x)	Cossecante	fix(x)	Arredonda em direção ao zero
csch(x)	Cossecante hiperbólica	floor(x)	Arredonda em direção a menos infinito
acsc(x)	Cossecante inversa	ceil(x)	Arredonda em direção a mais infinito
acsch(x)	Cossecante hiperbólica inversa	round(x)	Arredonda para o inteiro mais próximo
cot(x)	Cotangente	mod(x,y)	Módulo (resto com sinal após a divisão)
coth(x)	Cotangente hiperbólica	rem(x,y)	Resto após a divisão
acot(x)	Cotangente inversa		

Álgebra linear ➠ Vetores & Matrizes…

- **Declarações de vetores e matrizes:**

```
>>A=[1 2; 4 6]  < enter >

A =
   1   2
   4   6
```

Retorno de carro

```
>>A=[1 2;4 6];
>>
>>A=[1 2;4 6]

A =
   1   2
   4   6
```

Ponto e vírgula suprime a saída.

Sem ponto e vírgula exibe a saída.

```
>>M=[1 2];
>>m=[3 5 7];
```

FIGURA A.5
As variáveis são sensíveis à caixa.

Álgebra linear ➠Atribuições de valores…

- **Declarações de vetores e matrizes:**

$$\mathbf{A} = \begin{bmatrix} 1 & -4j & \sqrt{2} \\ \log(-1) & \operatorname{sen}(\pi/2) & \cos(\pi/3) \\ \operatorname{asen}(0{,}8) & \operatorname{acos}(0{,}8) & \exp(0{,}8) \end{bmatrix}.$$

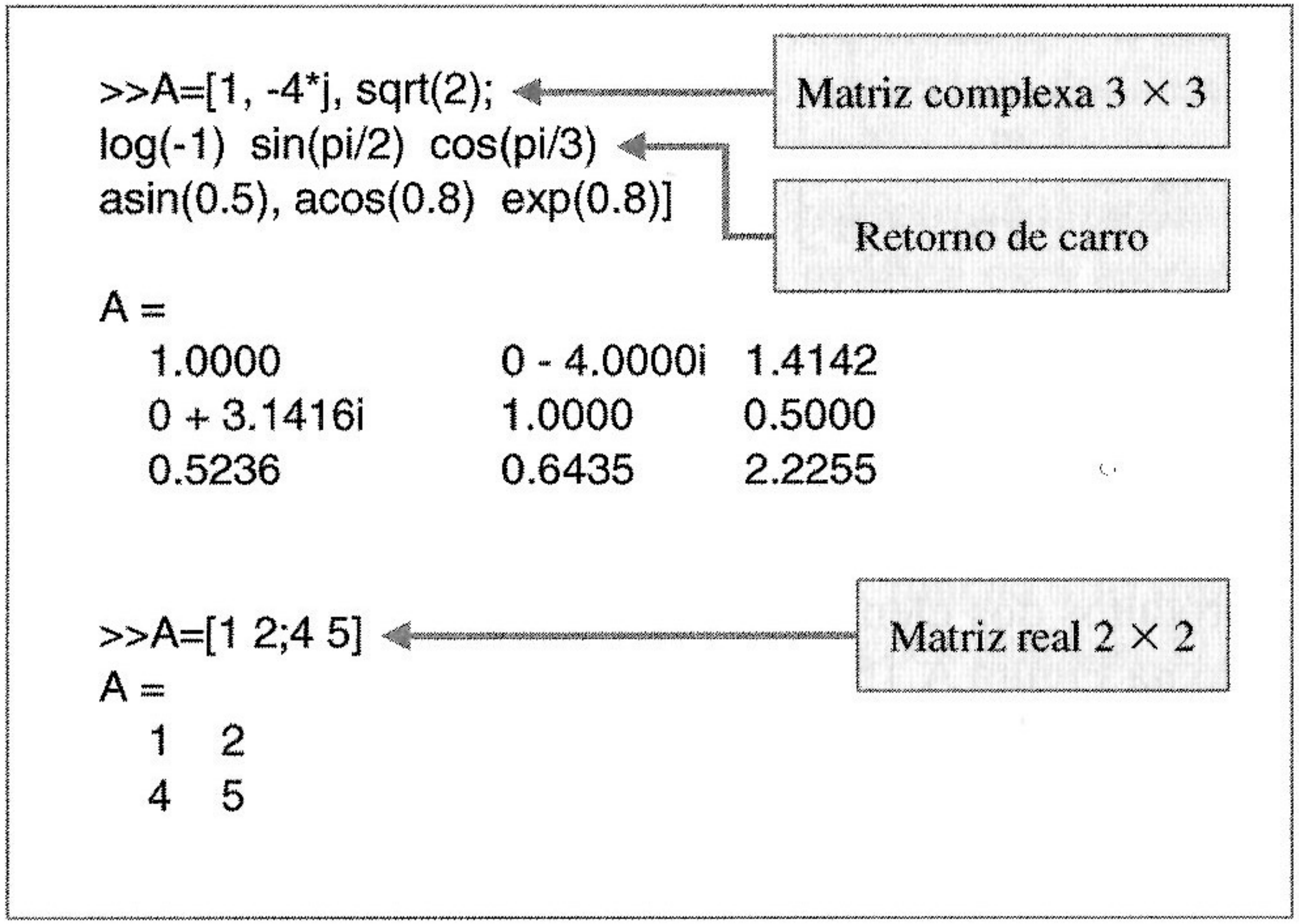


FIGURA A.12
Entrada de matriz complexa e real com ajuste automático de dimensão e de tipo.

Álgebra linear ➠ Operações

- **Operações com vetores e matrizes:**

Note: $\begin{cases} x_{3\times 1}; \\ y_{3\times 1}; \\ x'_{1\times 3}; \end{cases}$

```
>>A=[1 3; 5 9]; B=[4 -7; 10 0];
>>A+B            <- Adição matricial
ans =
   5   -4
  15    9
```

$\leftarrow$ Note: $A_{2\times 2} + B_{2\times 2}$

Note: $b_{2\times 1}$

```
>>b=[1;5];
>>A*b            <- Multiplicação matricial
ans =
   16
   50
```

Note: $A_{2\times 2} \times b_{2\times 1} = ans_{2\times 1}$

```
>>A'             <- Transposição matricial
ans =
   1   5
   3   9
```

FIGURA A.13 Três operações matriciais básicas: adição, multiplicação e transposição.

```
>>x=[5;pi;sin(pi/2)]; y=[exp(-0.5);-13;pi^2];
>>x'*y           <- Produto interno
ans =
  -27.9384

>>x*y'           <- Produto externo
ans =
   3.0327  -65.0000   49.3480
   1.9055  -40.8407   31.0063
   0.6065  -13.0000    9.8696
```

Note: $x_{3\times 1} \times y'_{1\times 3} = ans_{3\times 3}$

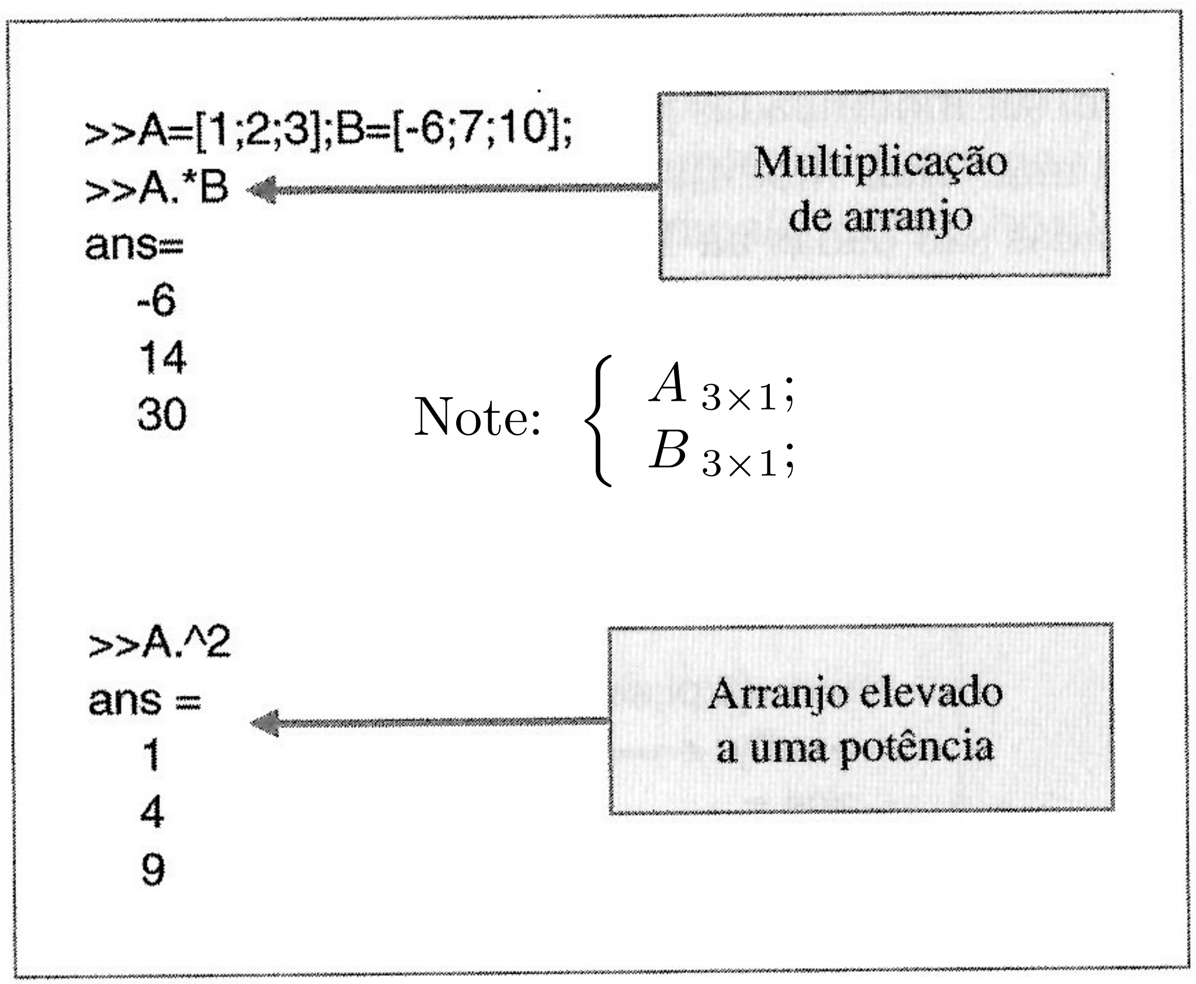
FIGURA A.14 Produtos interno e externo de dois vetores.

Tabela A.3 Operadores de Arranjo Matemáticos

+	Adição
−	Subtração
.*	Multiplicação
./	Divisão
.^	Potenciação

Álgebra linear

- **Operações ponto a ponto com vetores e matrizes:**



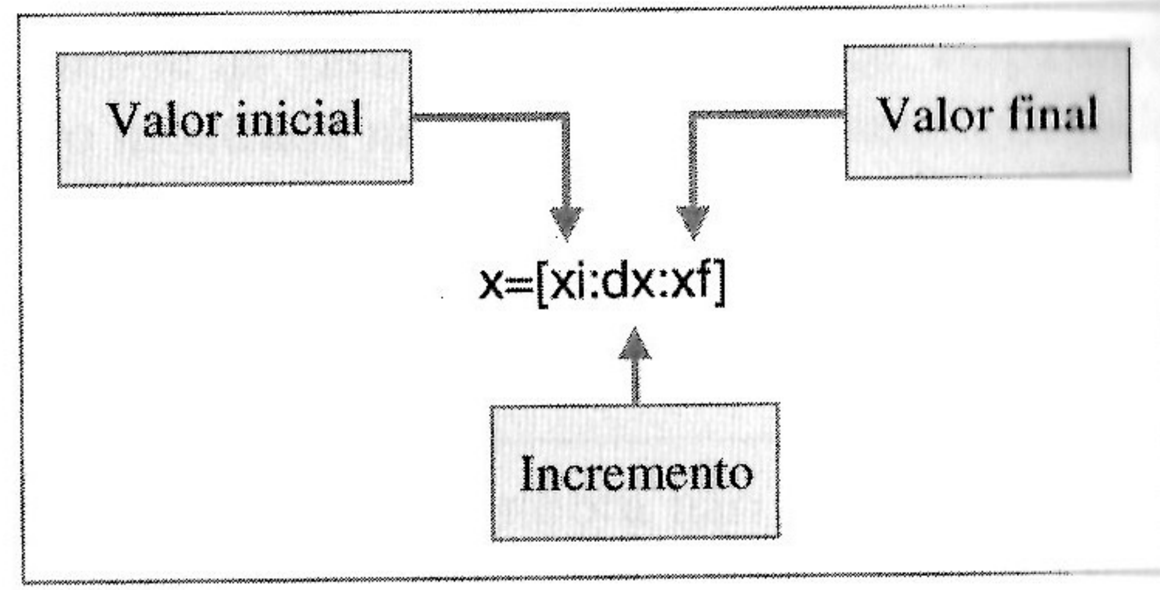
$$= \overbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}}^{A} \boxed{.} * \overbrace{\begin{bmatrix} -6 \\ 7 \\ 10 \end{bmatrix}}^{B}$$

Operação
elemento à
elemento

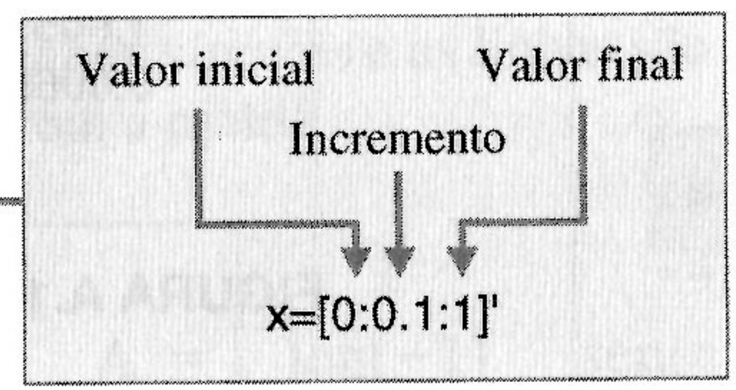
Não é produto matricial!

Álgebra linear

- **Criação de vetores:**




```
>>x=[0:0.1:1]';y=x.*sin(x);
>>[x y]
```




```
ans =
```

0	0
0.1000	0.0100
0.2000	0.0397
0.3000	0.0887
0.4000	0.1558
0.5000	0.2397
0.6000	0.3388
0.7000	0.4510
0.8000	0.5739
0.9000	0.7050
1.0000	0.8415

FIGURA A.17
Gerando vetores usando a notação de dois-pontos.

Sistemas Lineares

Seja:

$$\begin{cases} 3x + y - z = -7 \\ x - 2y + 3z = 3 \\ 4x + y + z = 0 \end{cases}$$

$$A \cdot x = b$$

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

atenção

$$x_{[3\times1]} = \left(A^{-1}\right)_{[3\times3]} \cdot \left(b\right)_{[3\times1]}$$

$$\begin{bmatrix} 3 & 1 & 1 \\ 1 & -2 & 3 \\ 4 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -7 \\ 3 \\ 0 \end{bmatrix}$$

- **Solução usando Matlab:**

```
>> A=[3 1 1; 1 -2 3; 4 1 1]
A =
     3     1     1
     1    -2     3
     4     1     1
>> b=[-7 3 0]'
b =
    -7
     3
     0
```

```
>> aux=inv(A)
aux =
   -1.0000         0    1.0000
    2.2000   -0.2000   -1.6000
    1.8000    0.2000   -1.4000
>> x=aux*b
x =
    7.0000
  -16.0000
  -12.0000
>>
```

ou

```
>> x=A\b
x =
     7
   -16
   -12

>>
```

Op. Trigonómetricas

- **Operações Trigonométricas:**

Atenção: argumento de entrada em **radianos**

```
>> cos(pi/6)
ans =
    0.8660

>> sin(pi/6)
ans =
    0.5000

>>
```

Graus	Radianos	Cos	Sen
30^o	$\frac{30\cdot\pi}{180} = \pi/6$	$\frac{\sqrt{3}}{2} = 0,8660$	$\frac{1}{2} = 0,5$
45^o	$\frac{45\cdot\pi}{180} = \pi/4$	$\frac{\sqrt{2}}{2} = 0,7071$	$\frac{\sqrt{2}}{2} = 0,7071$
60^o	$\frac{60\cdot\pi}{180} = \pi/3$	$\frac{1}{2} = 0,5$	$\frac{\sqrt{3}}{2} = 0,8660$
90^o	$\frac{90\cdot\pi}{180} = \pi/2$	0	1
180^o	π	-1	0

$$180^o \longleftrightarrow \pi$$

$$\text{Deg} \longleftrightarrow \text{Rad}$$

$$Rad = \frac{Deg\cdot\pi}{180}$$

$$Deg = \frac{Rad\cdot 180}{\pi}$$

Gráficos

- "plot"

Tabela A.4 Formatos de Gráficos

plot(x,y)	Traça o gráfico do vetor **x** *versus* o vetor **y**.
semilogx(x,y)	Traça o gráfico do vetor **x** *versus* o vetor **y**. O eixo x é $\log_{10}$; o eixo y é linear.
semilogy(x,y)	Traça o gráfico do vetor **x** *versus* o vetor **y**. O eixo x é linear; o eixo y é $\log_{10}$.
loglog(x,y)	Traça o gráfico do vetor **x** *versus* o vetor **y**. Cria um gráfico com escalas $\log_{10}$ em ambos os eixos.

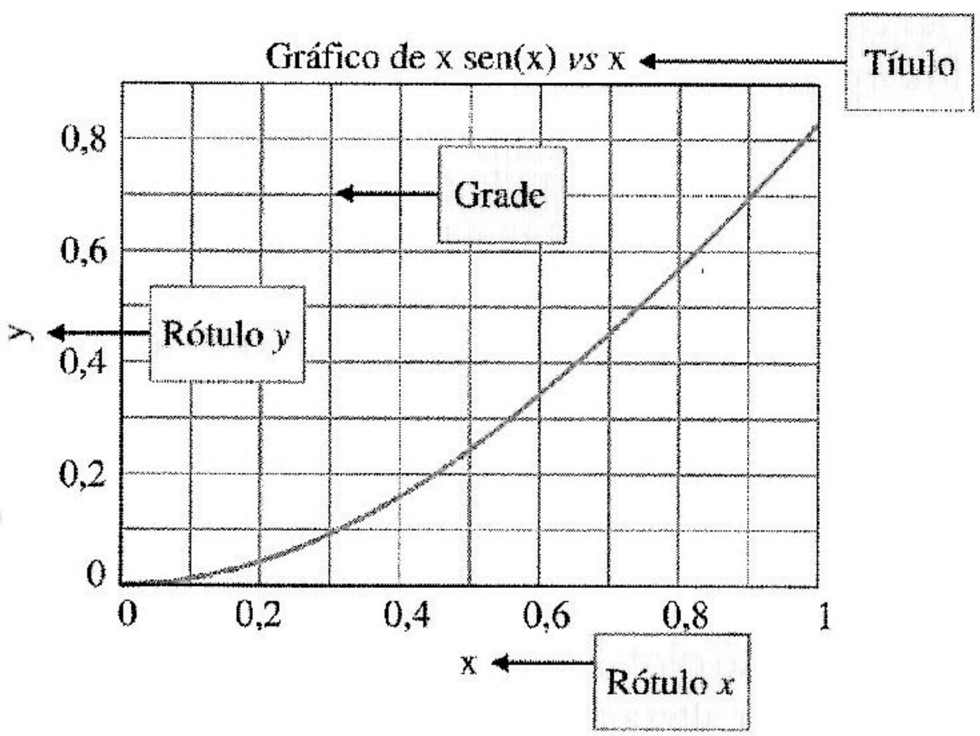
Tabela A.5 Funções para Gráficos Personalizados

title('texto')	Coloca 'texto' no topo do gráfico
legend(cadeia1, cadeia2,...)	Coloca uma legenda no gráfico atual usando cadeias de caracteres especificados como rótulos
xlabel('texto')	Rotula o eixo x com 'texto'
ylabel('texto')	Rotula o eixo y com 'texto'
text(p1,p2, 'texto')	Acrescenta 'texto' na posição (p1, p2), sendo que (p1, p2) estão em unidades do gráfico atual
subplot	Subdivide a janela gráfica
grid on	Acrescenta linhas de grade à figura atual
grid off	Remove linhas de grade da figura atual
grid	Comuta o estado da grade

Gráficos ➠ função plot(.)…

```
>>x=[0:0.1:1]';
>>y=x.*sin(x);
>>plot(x,y)
>>title('Gráfico de x sen(x) vs x')
>>xlabel('x')
>>ylabel('y')
>>grid on
```

(a)



(b)

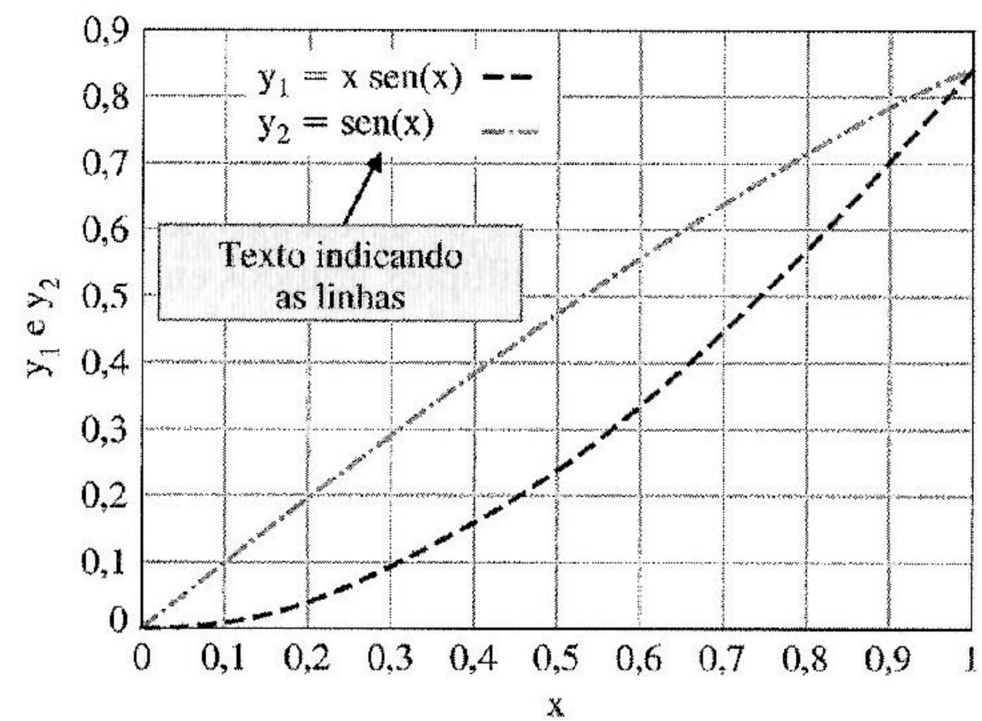
FIGURA A.18 (a) Comandos MATLAB. (b) Um gráfico *x-y* básico de x sen(*x*) versus *x*.

Tipo de Linha p/gráfico	Caracteres (string)
Linha contínua (default)	–
Linha tracejada	––
Linha pontilhada	:
Linha com traços e	–.

```
>>  x=[0:0.1:1]';
>>  y1=x.*sen(x); y2=sen(x);
>> plot(x,y1,'--',x,y2,'-.')
>> text(0.1,0.85,'y_1 = x sen(x) ---')
>> text(0.1,0.80,'y_2 = sen(x) .\_.\_')
>> xlabel('x'), ylabel('y_1 and y_2'), grid on
```

Linha tracejada para y1
Linha com traços e pontos para y2

(a)



(b)

FIGURA A.19 (a) Comandos MATLAB. (b) Um gráfico *x-y* básico com múltiplas linhas.

Gráficos ➠ função **subplot**(.)...

Uma figura com 4 gráficos no formato de uma matriz 2 x 2

1	2
3	4

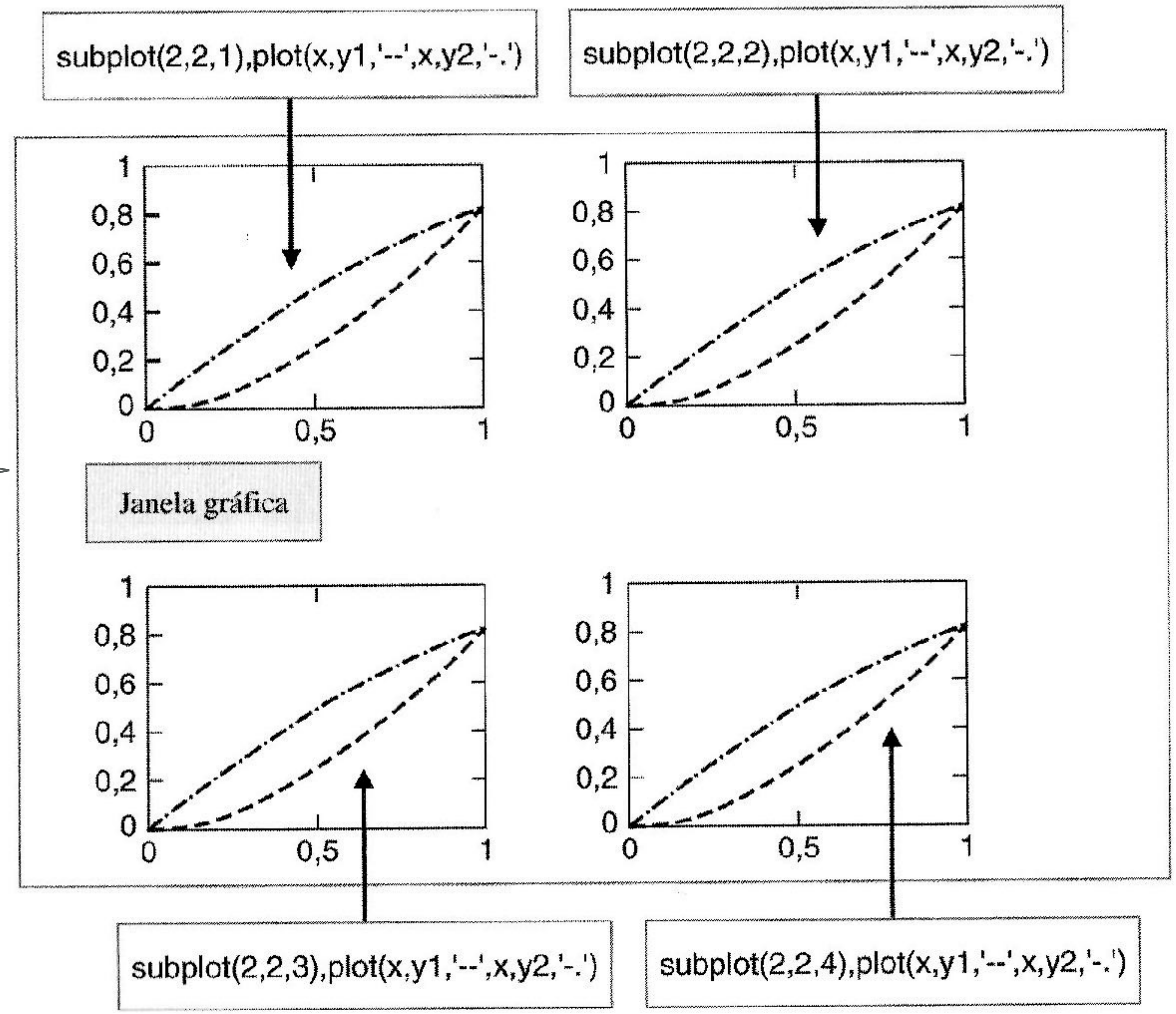
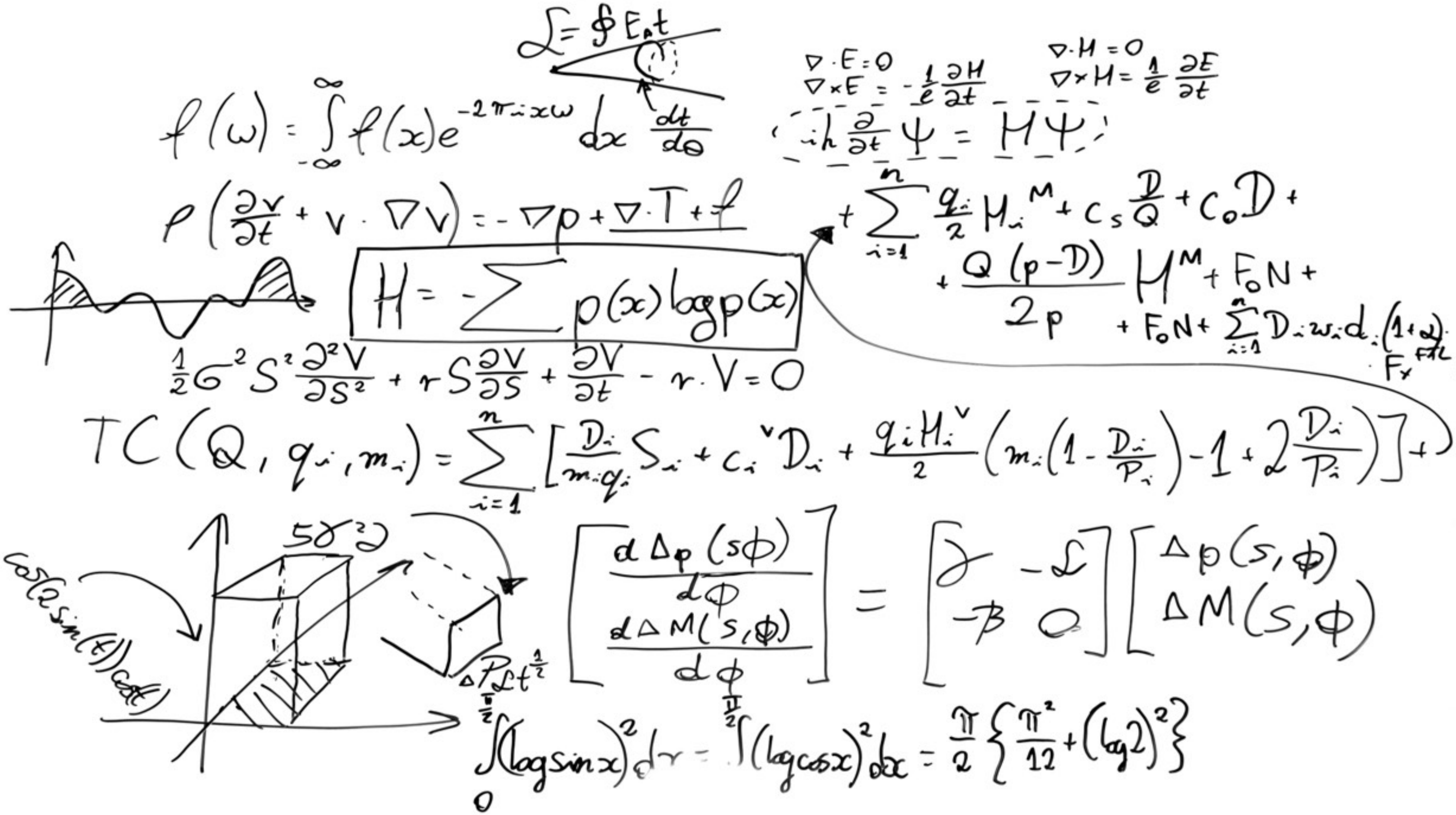


FIGURA A.20
Usando subplot para criar uma partição 2 × 2 da janela gráfica.

LaTeX !?



LaTeX !?

- **Melhorando informações nos gráficos (ou arquivo MarkDown)**

1 Greek and Hebrew letters

α	`\alpha`	κ	`\kappa`	ψ	`\psi`	$\digamma$	`\digamma`	Δ	`\Delta`	Θ	`\Theta`
β	`\beta`	λ	`\lambda`	ρ	`\rho`	ε	`\varepsilon`	Γ	`\Gamma`	Υ	`\Upsilon`
χ	`\chi`	μ	`\mu`	σ	`\sigma`	$\varkappa$	`\varkappa`	Λ	`\Lambda`	Ξ	`\Xi`
δ	`\delta`	ν	`\nu`	τ	`\tau`	φ	`\varphi`	Ω	`\Omega`		
ϵ	`\epsilon`	o	`o`	θ	`\theta`	ϖ	`\varpi`	Φ	`\Phi`	$\aleph$	`\aleph`
η	`\eta`	ω	`\omega`	υ	`\upsilon`	ϱ	`\varrho`	Π	`\Pi`	$\beth$	`\beth`
γ	`\gamma`	ϕ	`\phi`	ξ	`\xi`	ς	`\varsigma`	Ψ	`\Psi`	$\daleth$	`\daleth`
ι	`\iota`	π	`\pi`	ζ	`\zeta`	ϑ	`\vartheta`	Σ	`\Sigma`	$\gimel$	`\gimel`

2 LaTeX math constructs

$\frac{abc}{xyz}$	`\frac{abc}{xyz}`	$\overline{abc}$	`\overline{abc}`	$\overrightarrow{abc}$	`\overrightarrow{abc}`
f'	`f'`	$\underline{abc}$	`\underline{abc}`	$\overleftarrow{abc}$	`\overleftarrow{abc}`
$\sqrt{abc}$	`\sqrt{abc}`	$\widehat{abc}$	`\widehat{abc}`	$\overbrace{abc}$	`\overbrace{abc}`
$\sqrt[n]{abc}$	`\sqrt[n]{abc}`	$\widetilde{abc}$	`\widetilde{abc}`	$\underbrace{abc}$	`\underbrace{abc}`

3 Delimiters

<table>
<tr><td>$\vert$</td><td>|</td><td>$\{$</td><td>\{</td><td>$\lfloor$</td><td>\lfloor</td><td>$/$</td><td>/</td><td>$\Uparrow$</td><td>\Uparrow</td><td>$\llcorner$</td><td>\llcorner</td></tr>
<tr><td>$\vert$</td><td>\vert</td><td>$\}$</td><td>\}</td><td>$\rfloor$</td><td>\rfloor</td><td>$\backslash$</td><td>\backslash</td><td>$\uparrow$</td><td>\uparrow</td><td>$\lrcorner$</td><td>\lrcorner</td></tr>
<tr><td>$\Vert$</td><td>\|</td><td>$\langle$</td><td>\langle</td><td>$\lceil$</td><td>\lceil</td><td>$[$</td><td>[</td><td>$\Downarrow$</td><td>\Downarrow</td><td>$\ulcorner$</td><td>\ulcorner</td></tr>
<tr><td>$\Vert$</td><td>\Vert</td><td>$\rangle$</td><td>\rangle</td><td>$\rceil$</td><td>\rceil</td><td>$]$</td><td>]</td><td>$\downarrow$</td><td>\downarrow</td><td>$\urcorner$</td><td>\urcorner</td></tr>
</table>

Use the pair `\left`s_1 and `\right`s_2 to match height of delimiters s_1 and s_2 to the height of their contents, e.g.,
`\left|` *expr* `\right|` `\left\{` *expr* `\right\}` `\left\Vert` *expr* `\right.`

LaTeX !?

- **Melhorando informações nos gráficos (ou arquivo MarkDown)**

α	\alpha	β	\beta	γ	\gamma
δ	\delta	ϵ	\epsilon	ζ	\zeta
η	\eta	θ	\theta	ι	\iota
κ	\kappa	λ	\lambda	μ	\mu
ν	\nu	ξ	\xi	o	o
π	\pi	ρ	\rho	σ	\sigma
τ	\tau	υ	\upsilon	ϕ	\phi
χ	\chi	ψ	\psi	ω	\omega
Γ	\Gamma	Δ	\Delta	Θ	\Theta
Λ	\Lambda	Ξ	\Xi	Π	\Pi
Σ	\Sigma	Υ	\Upsilon	Φ	\Phi
Ψ	\Psi	Ω	\Omega		

Relations

$\equiv$	\equiv
$\doteq$	\doteq
$\cong$	\cong
$\simeq$	\simeq
$\approx$	\approx
$\sim$	\sim
$\propto$	\propto
$\neq$	\neq
$\leq$	\leq
$\geq$	\geq
$\ll$	\ll
$\gg$	\gg
$\rightarrow$	\rightarrow
$\Rightarrow$	\Rightarrow
$\Leftrightarrow$	\Leftrightarrow
$\in$	\in

Operators

$\pm$	\pm
$\mp$	\mp
$\times$	\times
$\cdot$	\cdot
$\oint$	\oint
$\int$	\int
$\sum$	\sum

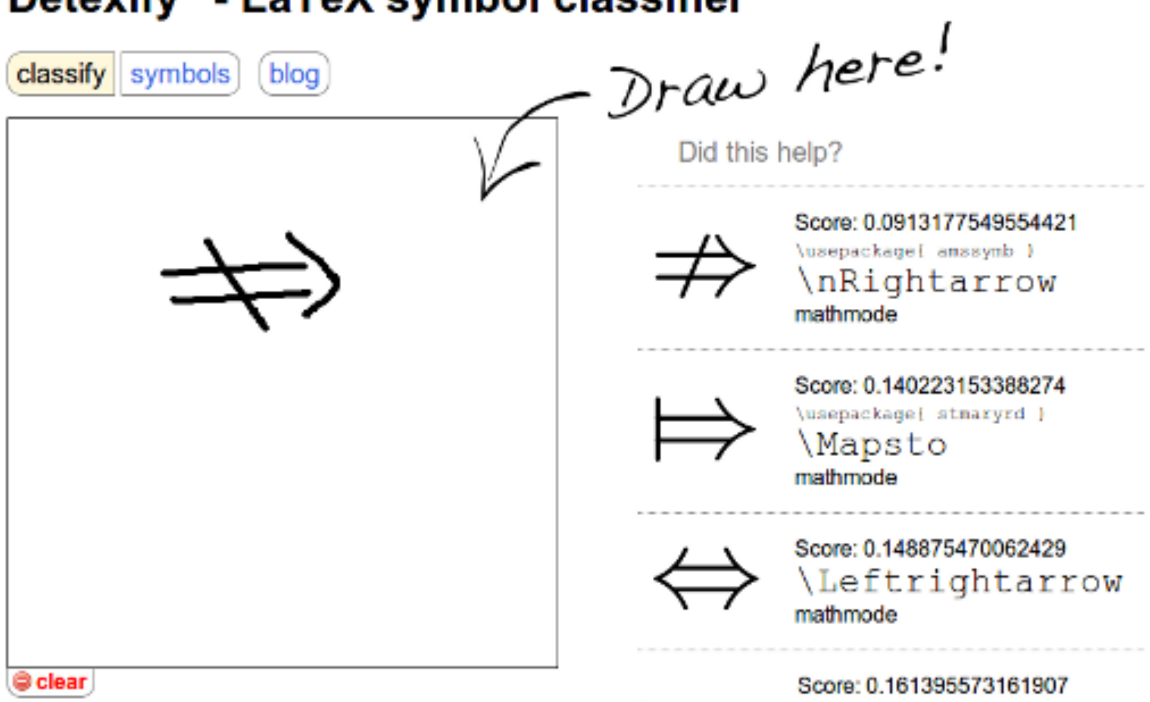
Decorations

$\sqrt{x}$	\sqrt{x}
$\tilde{x}$	\tilde{x}
$\dot{x}$	\dot{x}
$\ddot{x}$	\ddot{x}
$\hat{x}$	\hat{x}
$\bar{x}$	\bar{x}
$x^\dagger$	x^\dagger
x°	x^\circ
$\mathbb{R}$	\mathbb{R}

Symbols

∞	\infty
∂	\partial
$\perp$	\perp
$\parallel$	\parallel
∇	\nabla
$\Box$	\Box
$\hbar$	\hbar
$\cdots$	\cdots
$\forall$	\forall
$\exists$	\exists
$\uparrow$	\uparrow
$\downarrow$	\downarrow

Detexify² - LaTeX symbol classifier



http://detexify.kirelabs.org/classify.html

LaTeX !?

- **Para descobrir código/ comando LaTeX online:**
  - http://detexify.kirelabs.org/

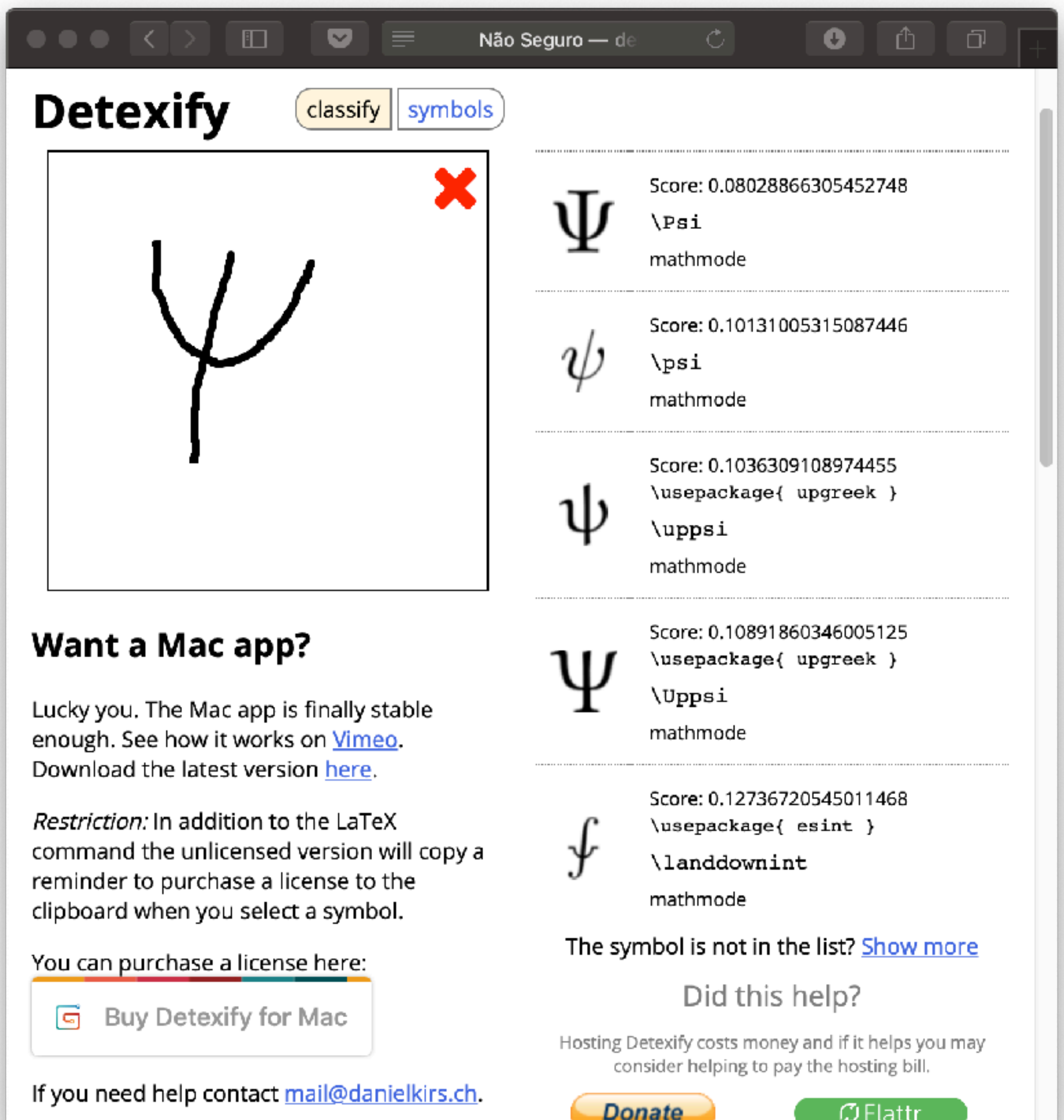
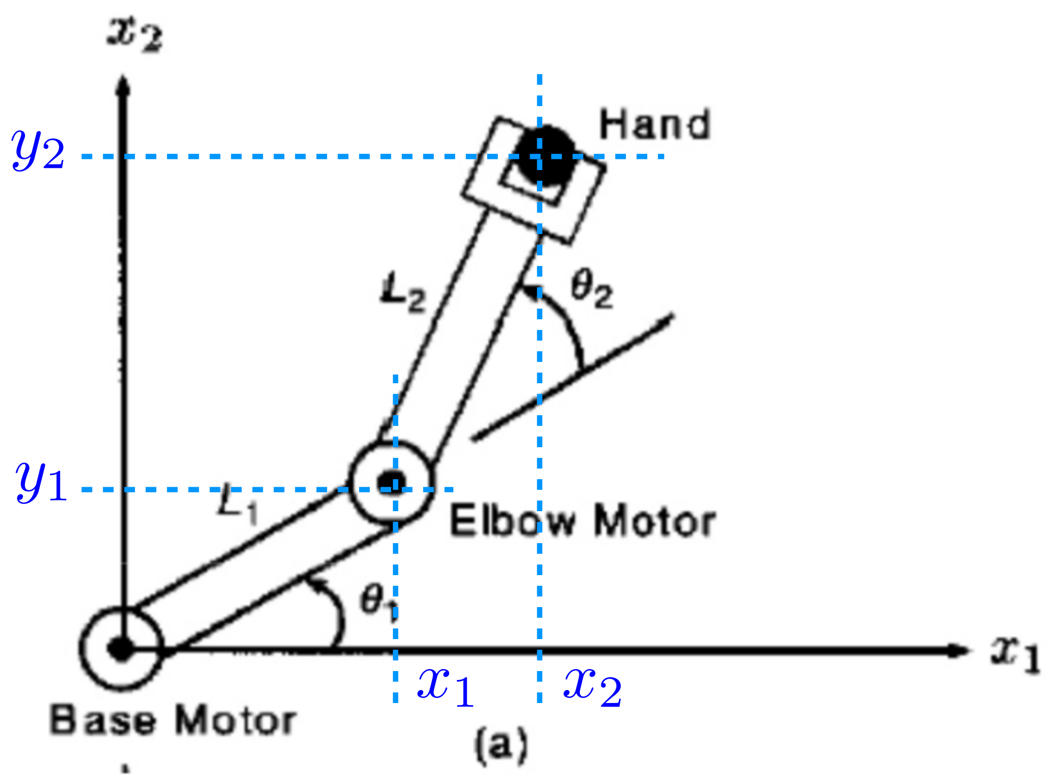


Tabela A.7 Símbolos TeX e Caracteres Matemáticos

Sequência de caracteres	Símbolo	Sequência de caracteres	Símbolo	Sequência de caracteres	Símbolo
\alpha	α	\upsilon	υ	\sim	$\sim$
\beta	β	\phi	φ	\leq	$\leq$
\gamma	γ	\chi	χ	\infty	∞
\delta	δ	\psi	ψ	\clubsuit	$\clubsuit$
\epsilon	ϵ	\omega	ω	\diamondsuit	$\blacklozenge$
\zeta	ζ	\Gamma	Γ	\heartsuit	$\heartsuit$
\eta	η	\Delta	Δ	\spadesuit	$\spadesuit$
\theta	θ	\Theta	Θ	\leftrightarrow	$\leftrightarrow$
\vartheta	ϑ	\Lambda	Λ	\leftarrow	$\leftarrow$
\iota	ι	\Xi	Ξ	\uparrow	$\uparrow$
\kappa	κ	\Pi	Π	\rightarrow	$\rightarrow$
\lambda	λ	\Sigma	Σ	\downarrow	$\downarrow$
\mu	μ	\Upsilon	Υ	\circ	$\circ$
\nu	ν	\Phi	Φ	\pm	$\pm$
\xi	ξ	\Psi	Ψ	\geq	$\geq$
\pi	π	\Omega	Ω	\propto	$\propto$
\rho	ρ	\forall	$\forall$	\partial	∂
\sigma	σ	\exists	$\exists$	\bullet	$\bullet$
\varsigma	ς	\ni	$\ni$	\div	$\div$
\tau	τ	\cong	$\cong$	\neq	$\neq$
\equiv	$\equiv$	\approx	$\approx$	\aleph	$\aleph$
\Im	$\Im$	\Re	$\Re$	\wp	$\wp$
\otimes	$\otimes$	\oplus	$\oplus$	\oslash	$\oslash$
\cap	$\cap$	\cup	$\cup$	\supseteq	$\supseteq$
\supset	$\supset$	\subseteq	$\subseteq$	\subset	$\subset$
\int	$\int$	\in	$\in$	\o	$^{\circ}$

Matlab ➠ Capacidades Gráficas

- **Exemplo1: plotar braços robô 2 DOF:**



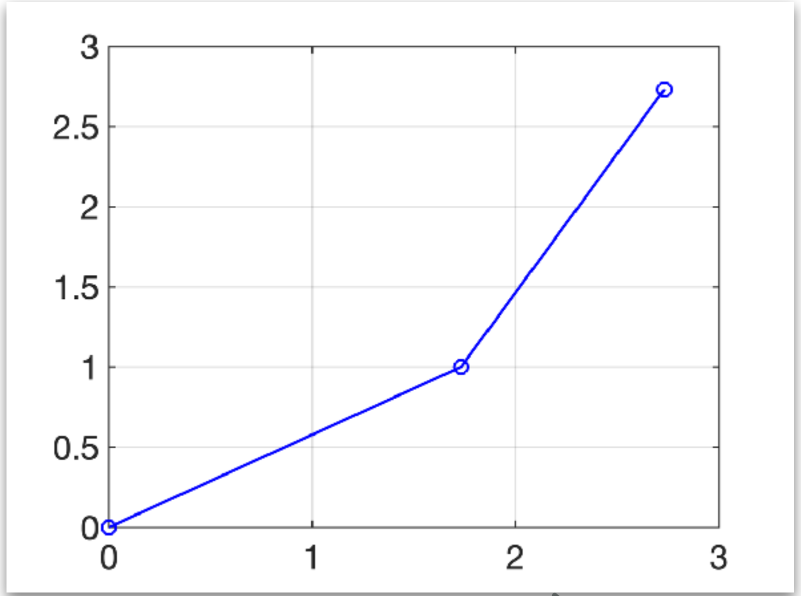
$$x_2 = L_1 \cdot \cos(\theta_1) + L_2 \cdot \cos(\theta_1 + \theta_2)$$

$$y_2 = L_1 \cdot \sin(\theta_1) + L_2 \cdot \sin(\theta_1 + \theta_2)$$

$$L_1 = 2; L_2 = 2;$$

$$\theta_1 = 30^o; \theta_2 = 60^o;$$

No Matlab:




```
L1=2; L2=2;
theta1=30; theta2=30;
% Convertendo angulos de graus p/radianos
theta_deg=[theta1 theta2];
theta=theta_deg*pi/180;
% calculando posições finais Link1
x1=L1*cos(theta(1));
y1=L1*sin(theta(1));
% calculando posições finais Link2
x2=x1+L2*cos(theta(1)+theta(2));
y2=y1+L2*sin(theta(1)+theta(2));
% Montandos vetor como pontos à plotar
x=[0 x1 x2];
y=[0 y1 y2];
% plotando os braços do robô
plot(x, y, 'bo-')
```

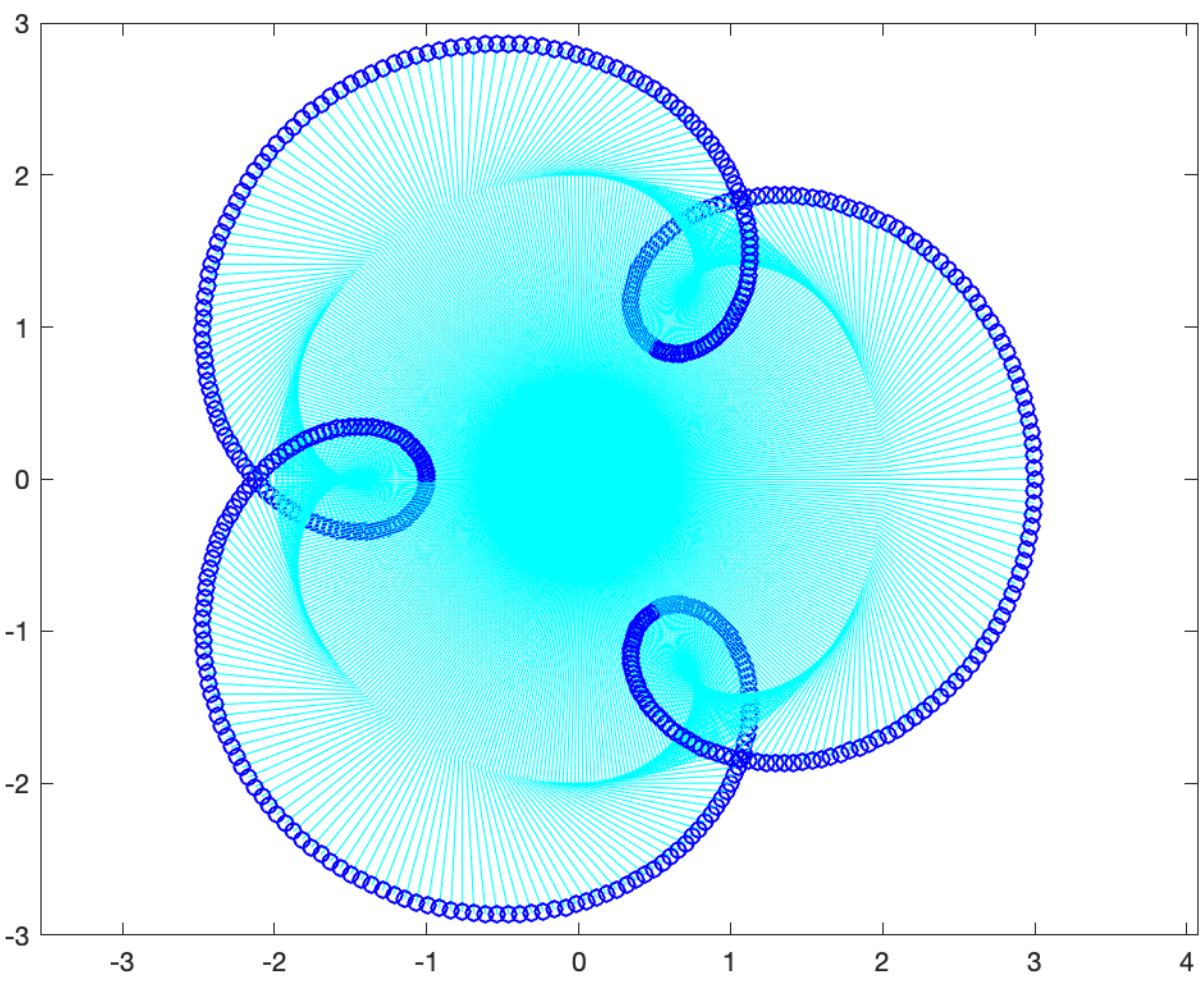
Note

```
>> [x' y']
ans =
         0         0
    1.7321    1.0000
    2.7321    2.7321
>>
```

Matlab ➟ Capacidades Gráficas

- **Exemplo₂: “brincando" com robô 2 DOF:**

```
r1=2; % raio do primeiro link/braço
r2=1; % raio do segundo link

vel1=15;        % velocidade angular de r1 em graus/seg
w1=vel1*pi/180; % velocidade angular de r1 em rad/seg

vel2=45;        % velocidade angular de r2 em graus/seg
w2=vel2*pi/180; % velocidade angular de r2 em rad/seg

T=0.05; % periodo de tempo usado como amostragem
t=0; t2=0; i=0;
t_fim=40; % periodo final de simulação

figure; % cria nova janela gráfica
t_amostra=1; % amostrar a cada quanto tempo (em segundos)

clear x1 y1 x2 y2 % limpando resquícios de rodada anterior
while t<=t_fim
   i=i+1; %incrementando contador de pontos para Matlab (indices de vetores iniciam em 1)
   th1=w1*t;
   x1=r1*cos(th1);
   y1=r1*sin(th1);
   % calculando ponto para segunda parte do braço
   th2=w2*t;
   x2=x1+r2*cos(th1+th2);
   y2=y1+r2*sin(th1+th2);
   % plotando gráfico
   x=[0 x1 x2];
   y=[0 y1 y2];
   plot(x,y, 'c-', x2,y2,'bo');
   if i==1
      hold on % manter mesma janela gráfica para póximos plots
   end
   axis([-3 3 -3 3]);
   % controlando se é necessário pause
   if t2>=t_amostra
      pause(1)
      t2=0;
   end
   % atualizando variáveis para próxima rodada
   t=t+T;
   t2=t2+T;
end
```



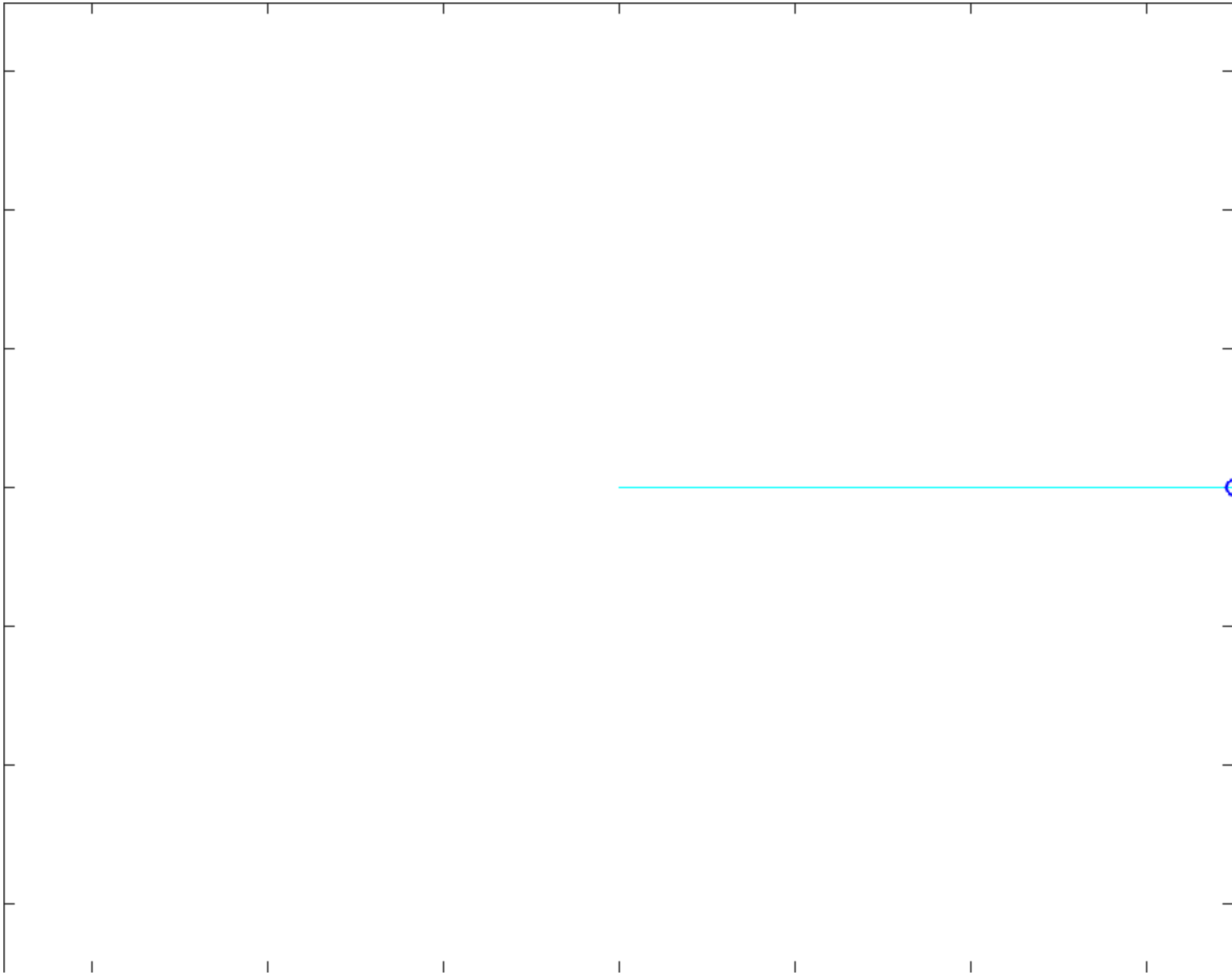
Matlab ➠ Capacidades Gráficas

- **Exemplo$_2$: “brincando" com robô 2 DOF:**

Alteradas algumas linhas de código:

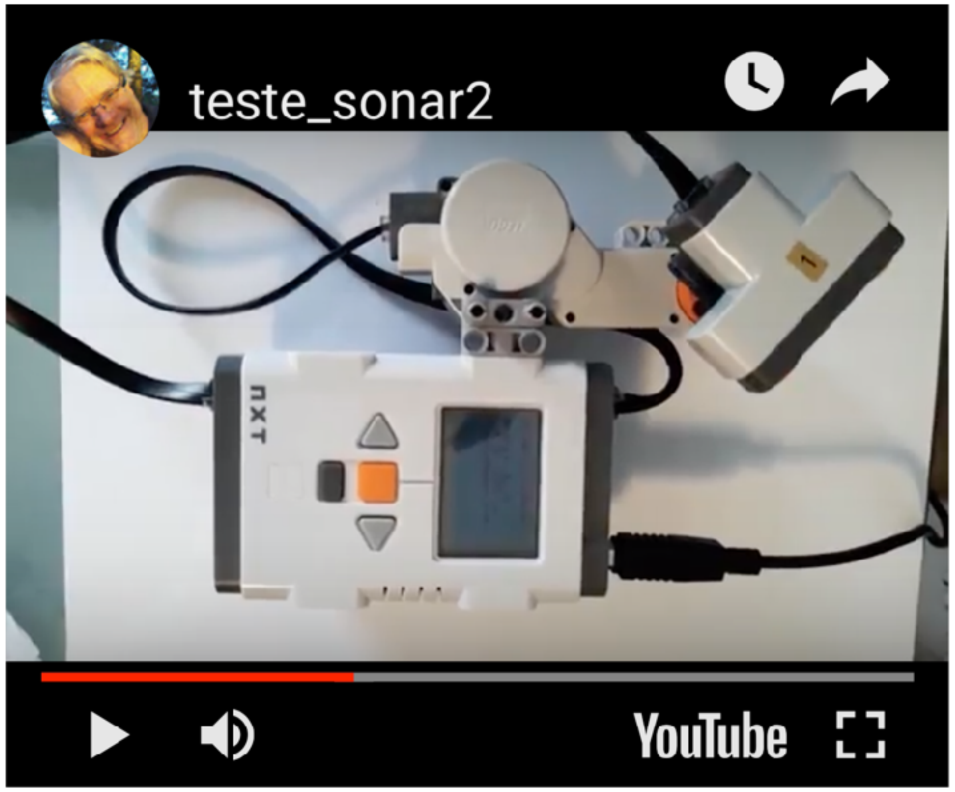
r2=1.5;

vel2=300; % 300^o/segundo



Capacidades Gráficas

- **Exemplo$_2$: Diagrama polar sensor ultra-sônico de distâncias**



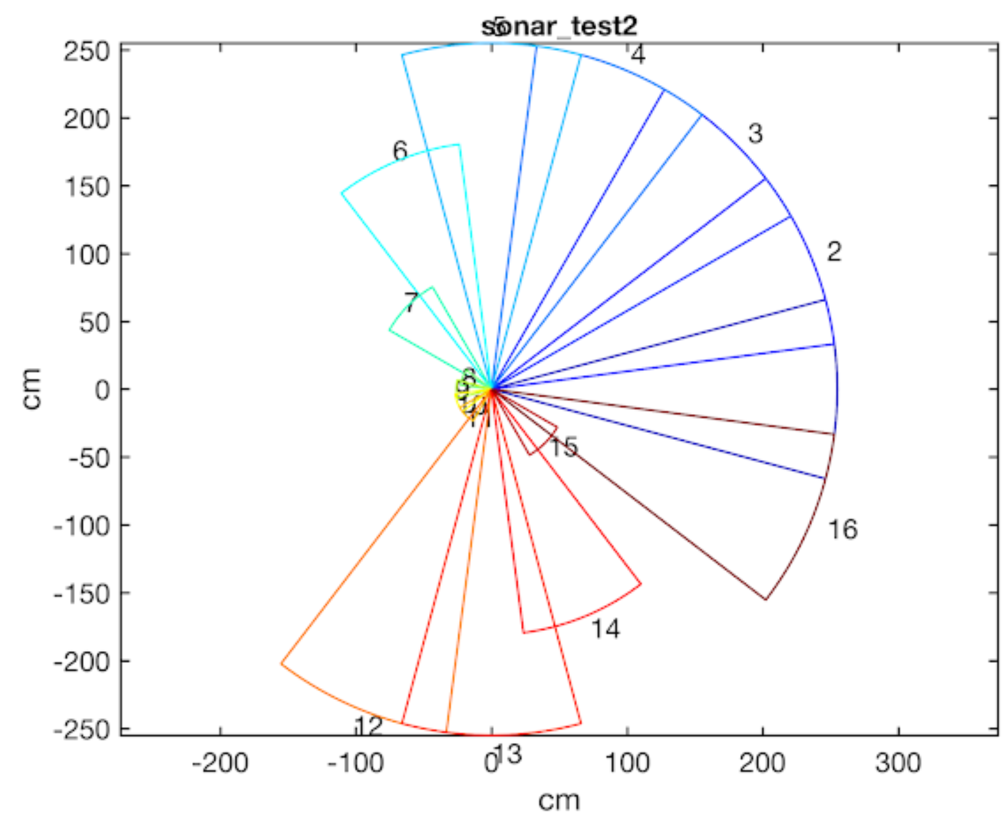
Vídeo: https://youtu.be/Ft8S7VEC0mQ

Arquivo "dados.txt" gerado pelo NXT:

---- dados.txt ---

```
255
255
255
255
255
255
255
23
255
255
26
27
27
255
255
181
----
```

Interpretação gráfica usando Matlab:

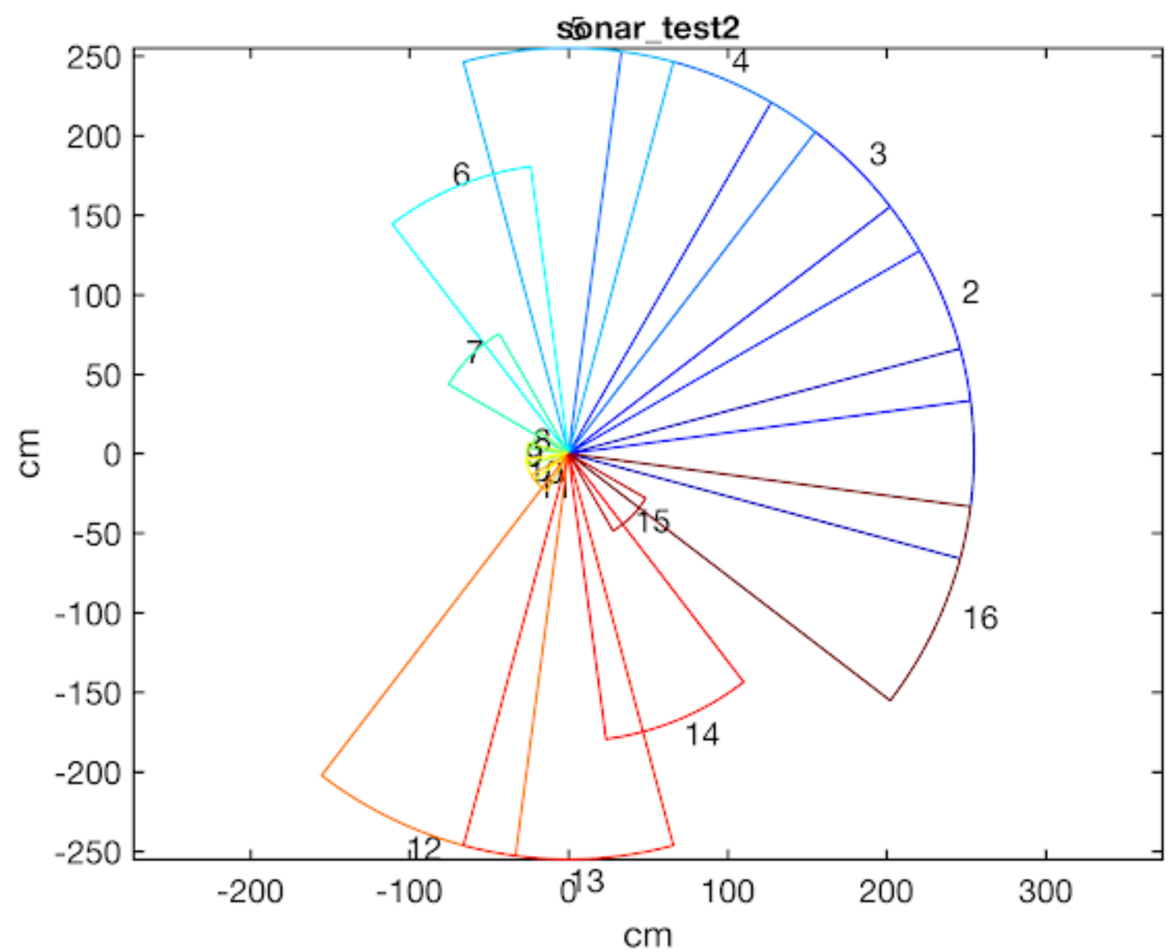


áficas

ensor ultra-sônico de distâncias

```
% teste_sonar2
% Fernando Passold 21/09/2018
%   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16
d=[255 255 255 255 255 182  87  23  26  27  27 255 255 181  56 255];
u=length(d); % caso o tamanho do vetor varie
step=2*pi/u;
angle=0;
range=15; % leque de abertura;
range_rad=(pi*range)/180;
step_r=pi/180; % passo para gerar o plot -15^o ? +15^o
%%%
step_color=64/u; % colormap<=>cmap:64x3
color=step_color;
cmap=colormap(jet); % winter
for index=1:u
    cont=2;
    range_sup=angle+range_rad;
    range_inf=angle-range_rad;
    x(1)=0; y(1)=0;
    for theta=range_inf:step_r:range_sup
        x(cont)=d(index)*cos(theta);
        y(cont)=d(index)*sin(theta);
        cont=cont+1;
    end
    msg=num2str(index);
    x_msg=d(index)*1.05*cos(angle);
    y_msg=d(index)*1.05*sin(angle);
    text(x_msg, y_msg, msg);
    x(cont)=0; y(cont)=0;
    plot(x,y,'Color',cmap(color,1:3));
    if index==1
        hold on; % para pr?ximos gr?ficos
    end
    angle=angle+step;
    color=color+step_color;
end
hold off
axis equal
title('sonar\_test2')
```

No Matlab:



“Programas" ➠ Scripts (arquivos .m)

- **Exemplo:**

```
>>alpha=50;
>>plotdata
```

plotdata.m

```
% Esta é uma sequência de instruções para representar
% graficamente a função y=sen(alpha*t).
% O valor de alpha deve existir no workspace antes de
% se chamar a sequência de instruções.
%
t=[0:0.01:1];
y=sin(alpha*t);
plot(t,y)
xlabel('Tempo (s)')
ylabel('y(t) = sen(\alpha  t )')
grid on
```

FIGURA A.21 Uma sequência de instruções simples para representar graficamente a função $y(t) = \text{sen } \alpha t$.

```
>>help plotdata

   Esta é uma sequência de instruções para representar
   graficamente a função y=sen(alpha*t).

   O valor de alpha deve existir no workspace antes
   de se chamar a sequência de instruções.
```

FIGURA A.22 Usando a função **help**.

Matlab ➠ Uso em Controle

- **Exemplo:**

```
>>y0=0.15;
>>wn=sqrt(2);
>>zeta=1/(2*sqrt(2));
>>t=[0:0.1:10];
>>unforced
```

ω_n

ζ

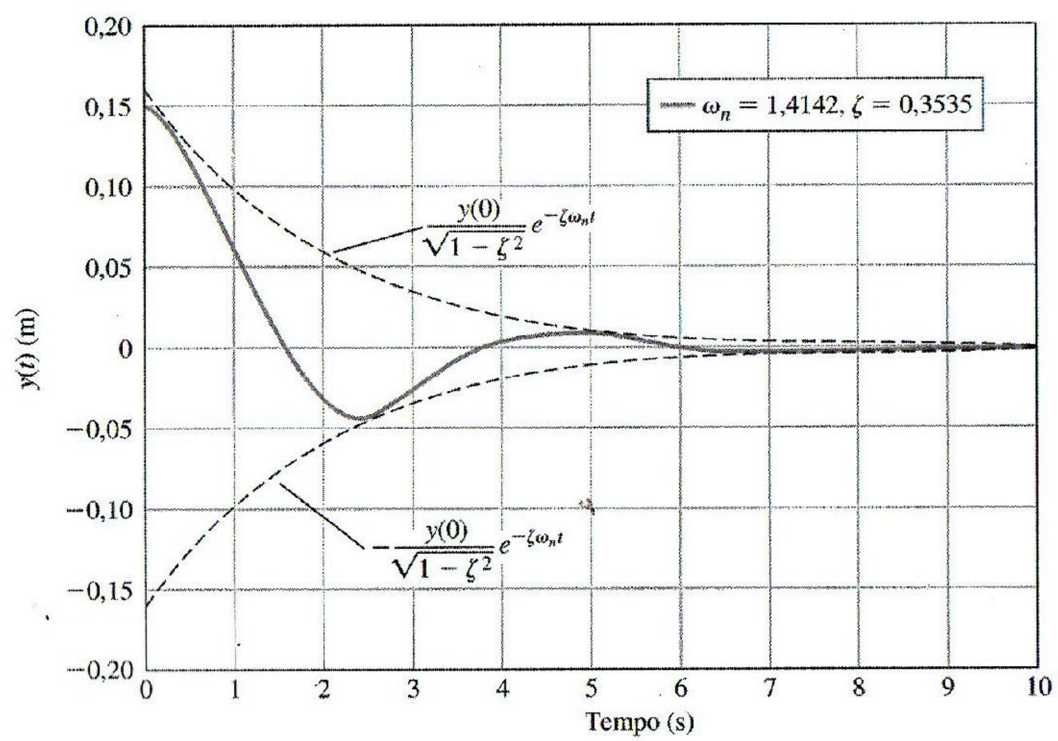
unforced.m

```
%Calcula a Resposta Livre para uma Condição Inicial
%
c=(y0/sqrt(1-zeta^2));
y=c*exp(-zeta*wn*t).*sin(wn*sqrt(1-zeta^2)*t+acos(zeta));
%
bu=c*exp(-zeta*wn*t);bl=-bu;
%
plot(t,y,t,bu,'--',t,bl,'--'), grid
xlabel('Tempo (s)'), ylabel('y(t) (m)')
legend(['\omega_n=',num2str(wn),'   \zeta=',num2str(zeta)])
```

$y(0)/\sqrt{1-\zeta^2}$

envoltória $e^{-\zeta\omega_n t}$

FIGURA 2.50 Sequência de instruções para analisar o sistema massa–mola–amortecedor.



Matlab ➟ Uso em Controle

- **Exemplo:**

Ingressando polinômios
↓
Calculando raízes (`root`)
↓
Montando polinômios (`poly`)
↓
Multiplicando polinômios (`conv`)
↓
Substituindo valores em polinômios (`polyval`)

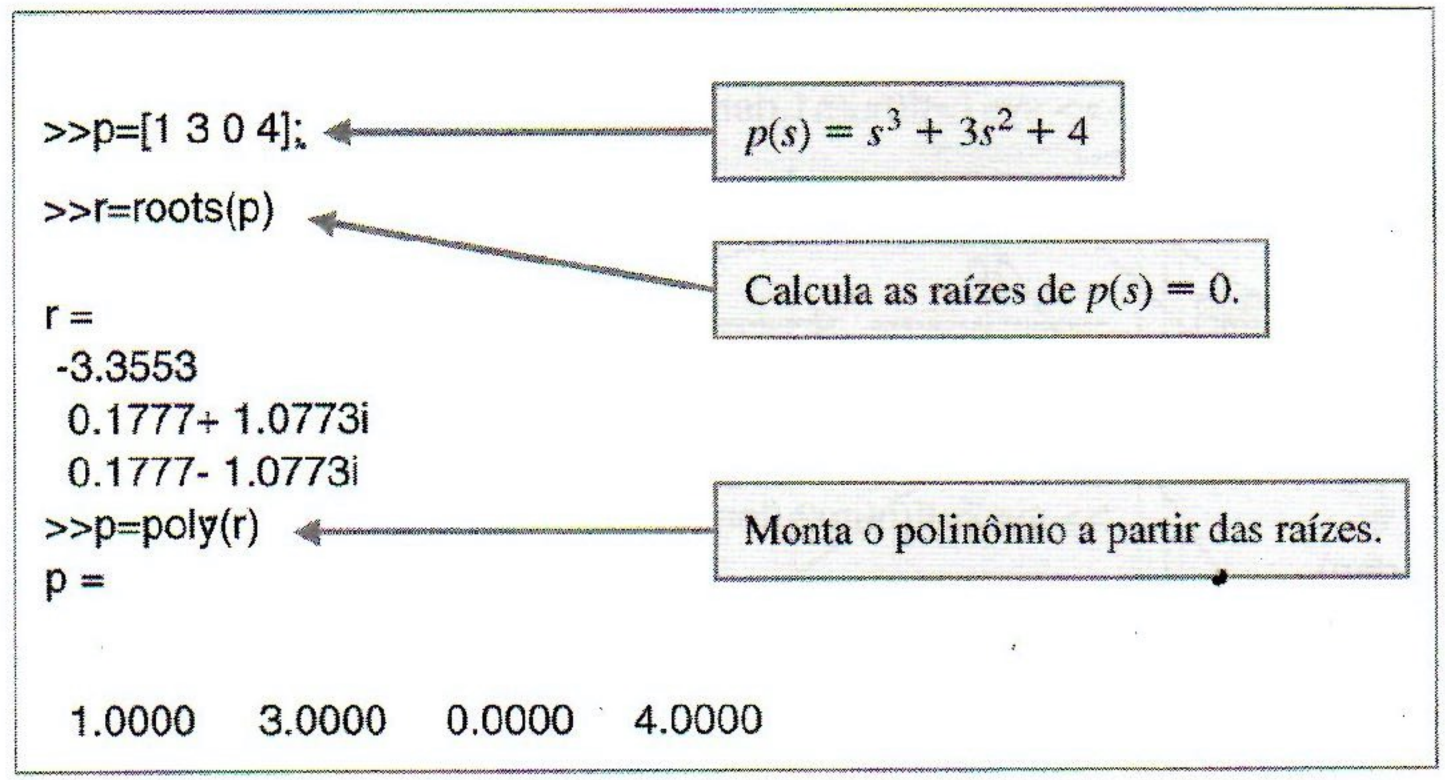


FIGURA 2.52 Definindo o polinômio $p(s) = s^3 + 3s^2 + 4$ e calculando suas raízes.

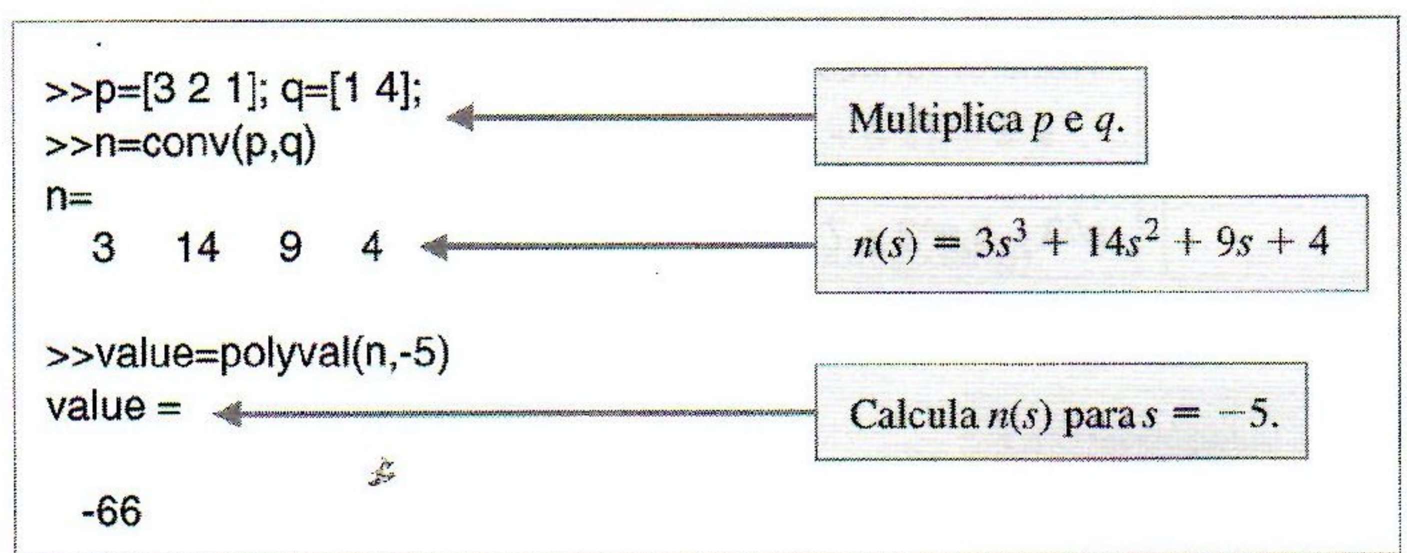


FIGURA 2.53 Usando **conv** e **polyval** para multiplicar e calcular o valor do polinômio $(3s^2 + 2s + 1)(s + 4)$.

Matlab ➟ Uso em Controle

- **Exemplo:**

Ingressando função transferência (`tf`)
↓
Cascateando funções transferência (blocos em paralelo) (**+**)

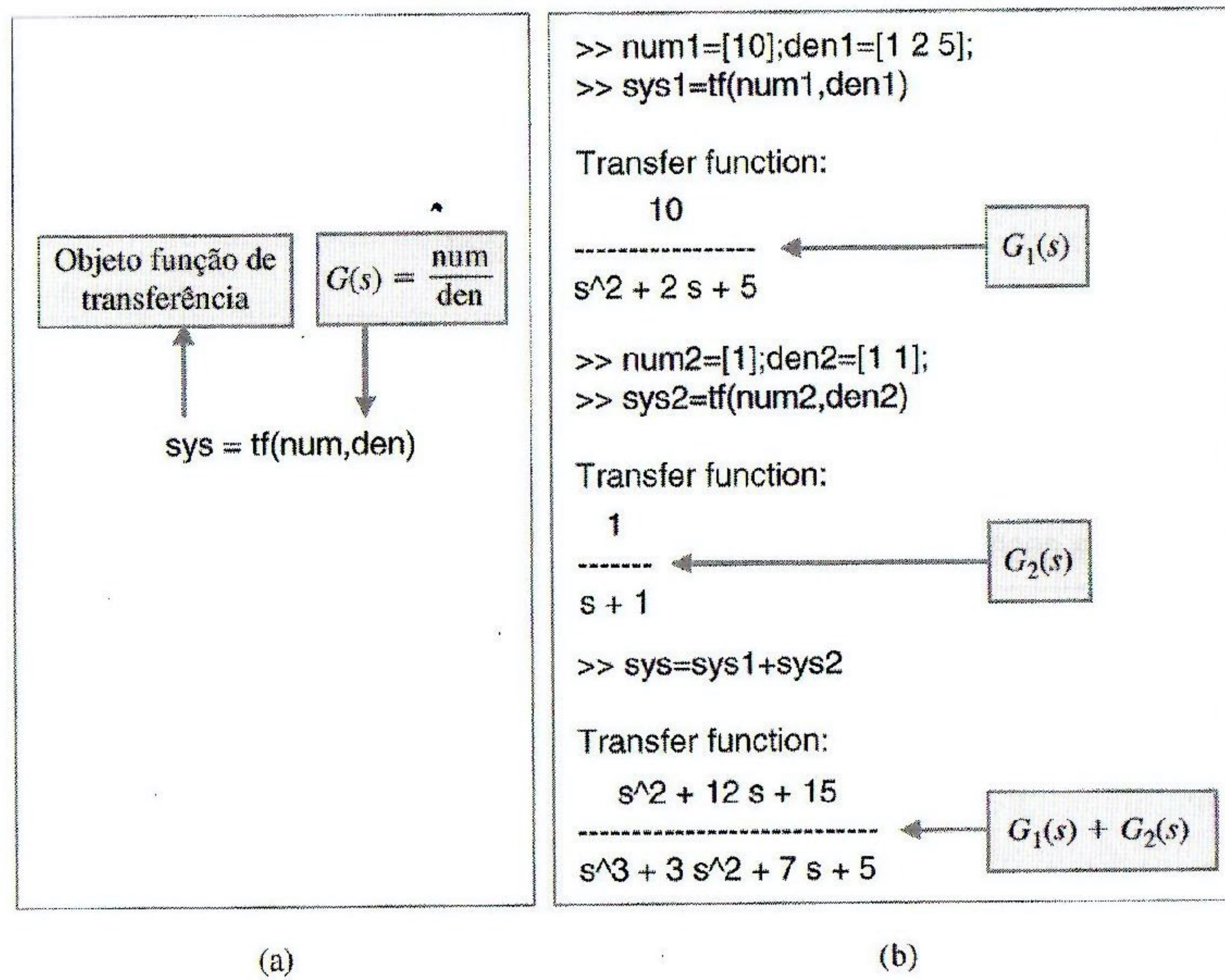


FIGURA 2.54
(a) A função **tf**.
(b) Usando a função **tf** para criar objetos função de transferência e somando-os usando o operador "+".

Matlab ➠ Uso em Controle

- **Exemplo:**

Determinando polos de uma “tf” (`pole`)
↓
Encontrando os zeros de uma "tf" (`zero`)
↓
Gráfico do RL com os polos e zeros (`pzmap`)

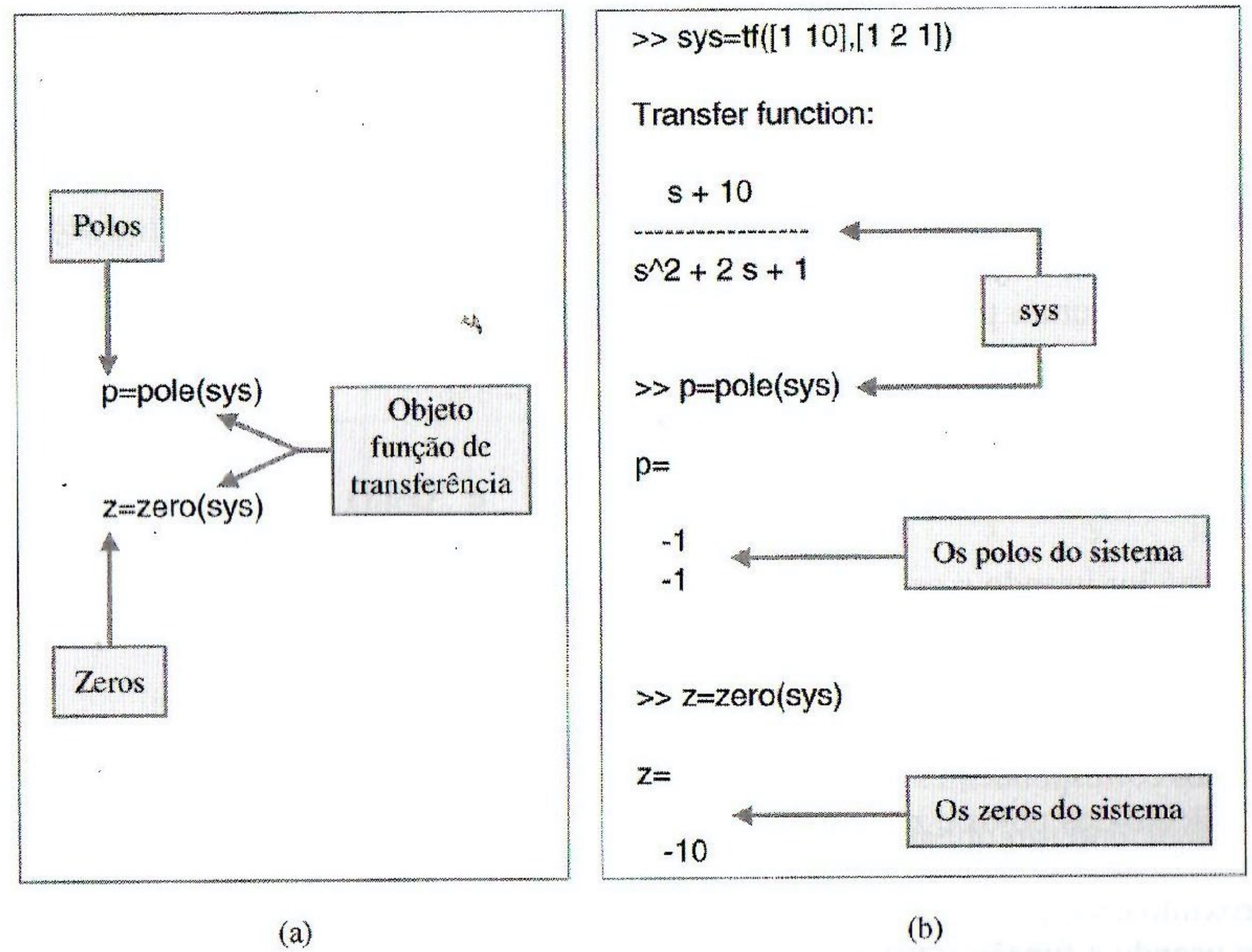


FIGURA 2.55
(a) As funções **pole** e **zero**. (b) Usando as funções **pole** e **zero** para calcular as posições dos polos e zeros de um sistema linear.

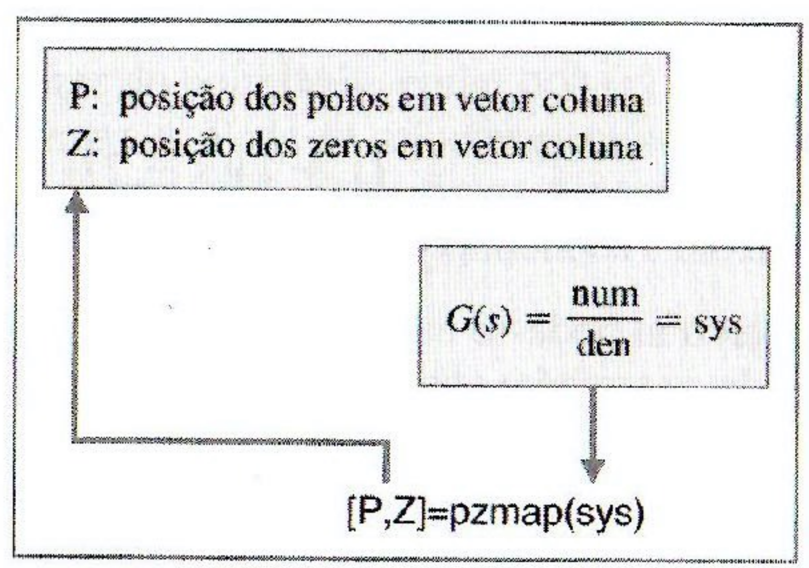
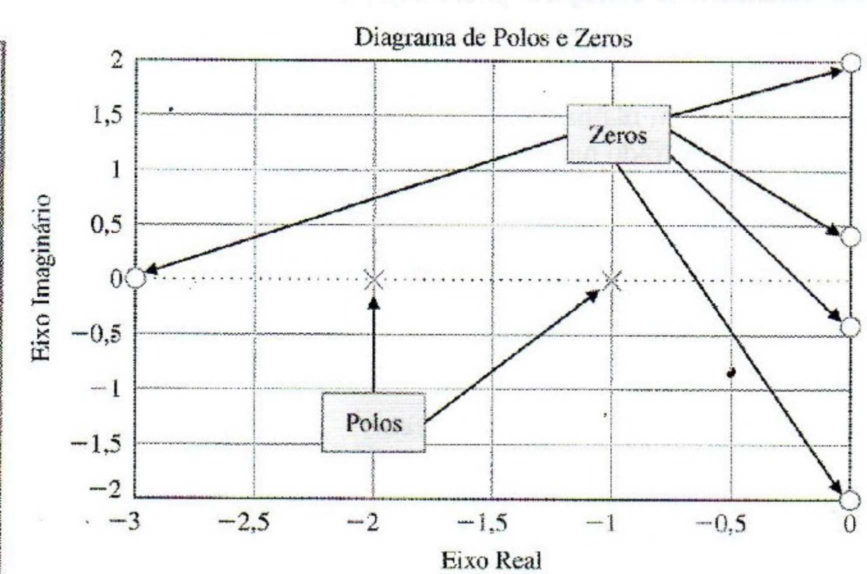


FIGURA 2.56
A função **pzmap**.



Matlab ➠ Uso em Controle

- **Exemplo:**

Determinando polos de uma “tf” (**`pole`**)

↓

Encontrando os zeros de uma "tf" (**`zero`**)

↓

Gráfico do RL com os polos e zeros (**`pzmap`**)

↓

Formato canônico (**`zpk`**)

```
>> G=tf( [6 0 1], [1 3 3 1] )
G =
        6 s^2 + 1
  ---------------------
  s^3 + 3 s^2 + 3 s + 1
Continuous-time transfer function.

>> zpk(G)
ans =
  6 (s^2 + 0.1667)
  ----------------
      (s+1)^3
Continuous-time zero/pole/gain model.

>>
```

```
>>numg=[6 0 1]; deng=[1 3 3 1];sysg=tf(numg,deng);
>>z=zero(sysg)

z =
    0 + 0.4082i
    0 - 0.4082i

>>p=pole(sysg)

p =
    -1.0000
    -1.0000 + 0.0000i
    -1.0000 -  0.0000i

>>n1=[1 1]; n2=[1 2]; d1=[1 2*i]; d2=[1 -2*i]; d3=[1 3];
>>numh=conv(n1,n2); denh=conv(d1,conv(d2,d3));
>>sysh=tf(numh,denh)
Transfer function:
     s^2 + 3 s + 2
----------------------
s^3 + 3 s^2 + 4 s + 12

>>sys=sysg/sysh
Transfer function:
6 s^5 + 18 s^4 + 25 s^3 + 75 s^2 + 4 s +12
------------------------------------------
  s^5 + 6 s^4 + 14 s^3 + 16 s^2 + 9 s + 2

>>pzmap(sys)
```

Calcula os polos e zeros de $G(s)$

Expande $H(s)$

$H(s)$

$\frac{G(s)}{H(s)} = \text{sys}$

Diagrama de polos e zeros

FIGURA 2.58
Exemplo de função de transferência para $G(s)$ e $H(s)$.

Matlab ➠ Uso em Controle

- **Aritmética de blocos:**

Blocos em série (`series`)

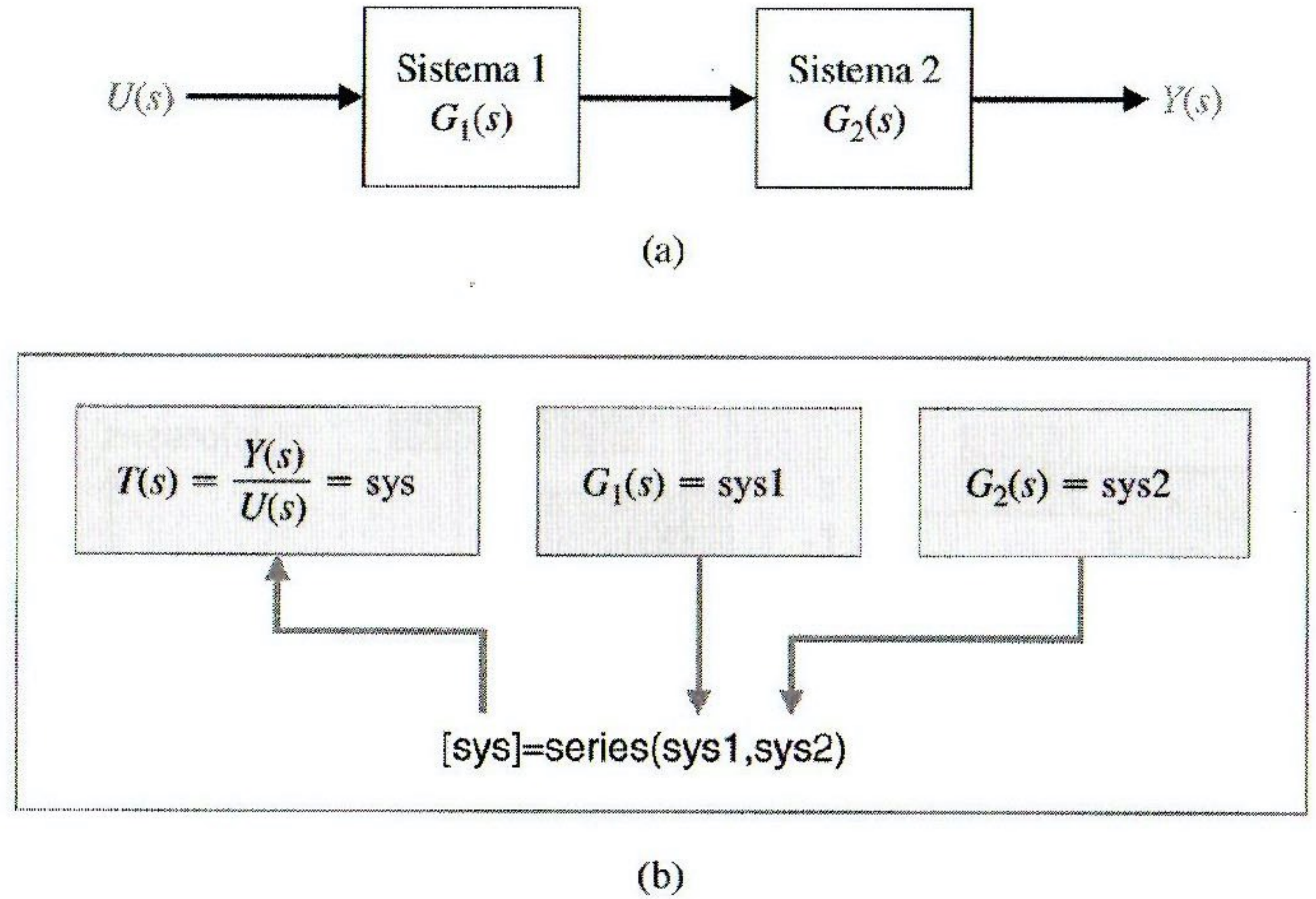


FIGURA 2.60
(a) Diagrama de blocos. (b) A função **series**.

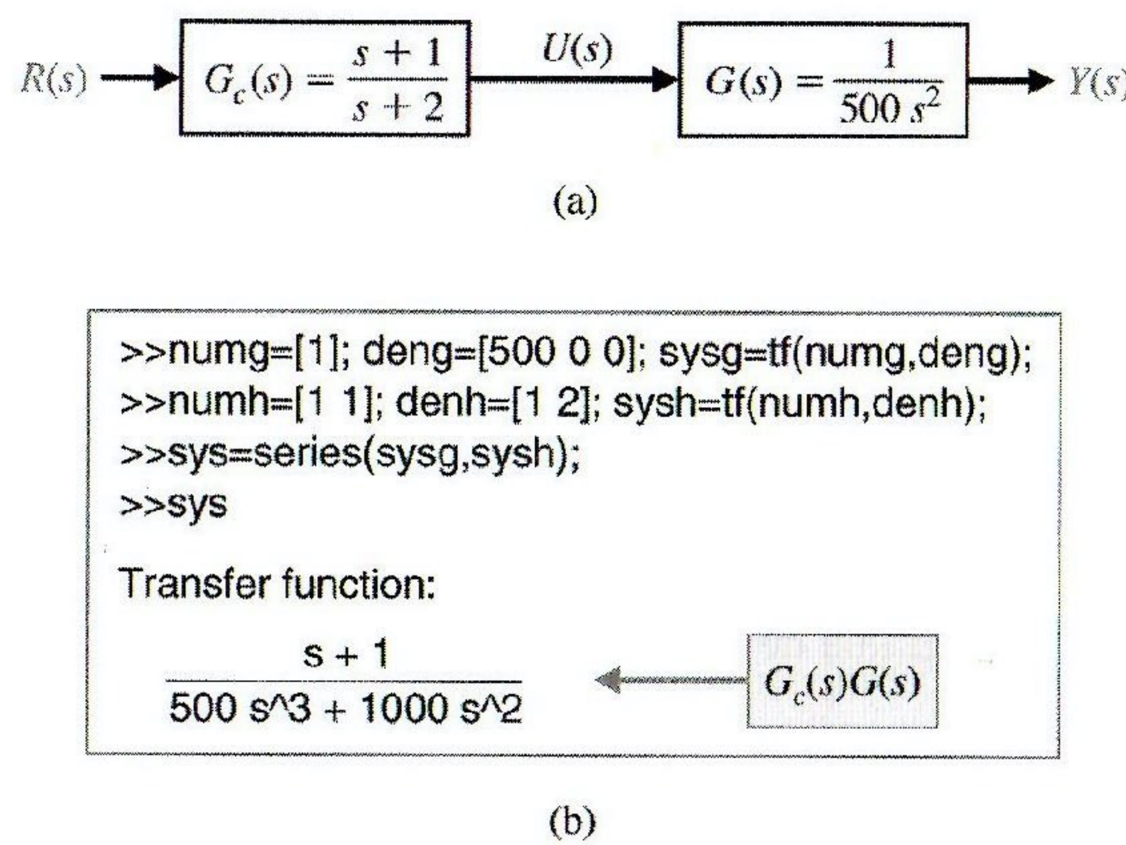


FIGURA 2.61
Uso da função **series**.

Matlab ➠ Uso em Controle

- **Aritmética de blocos:**

Blocos em paralelo (`parallel`)

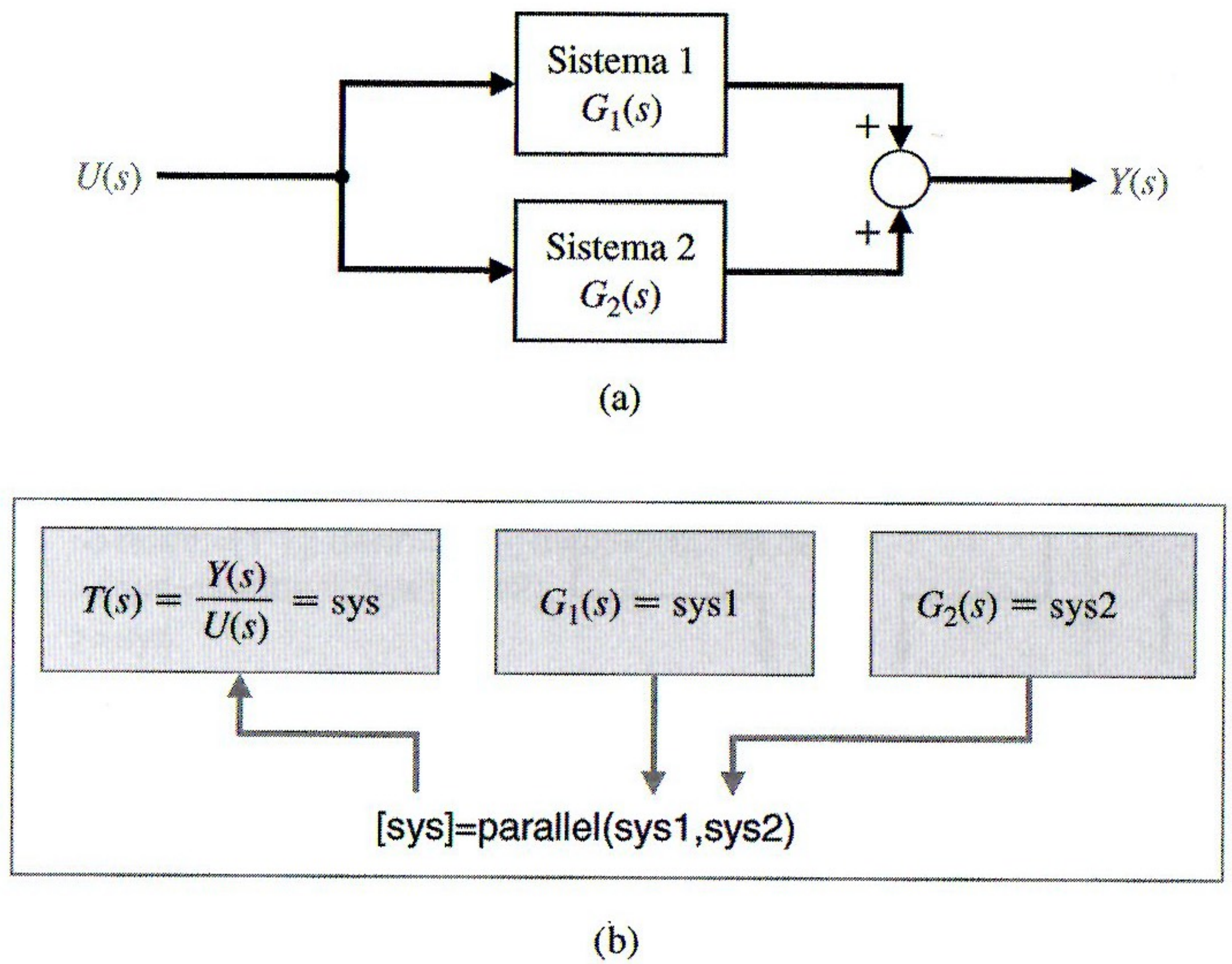
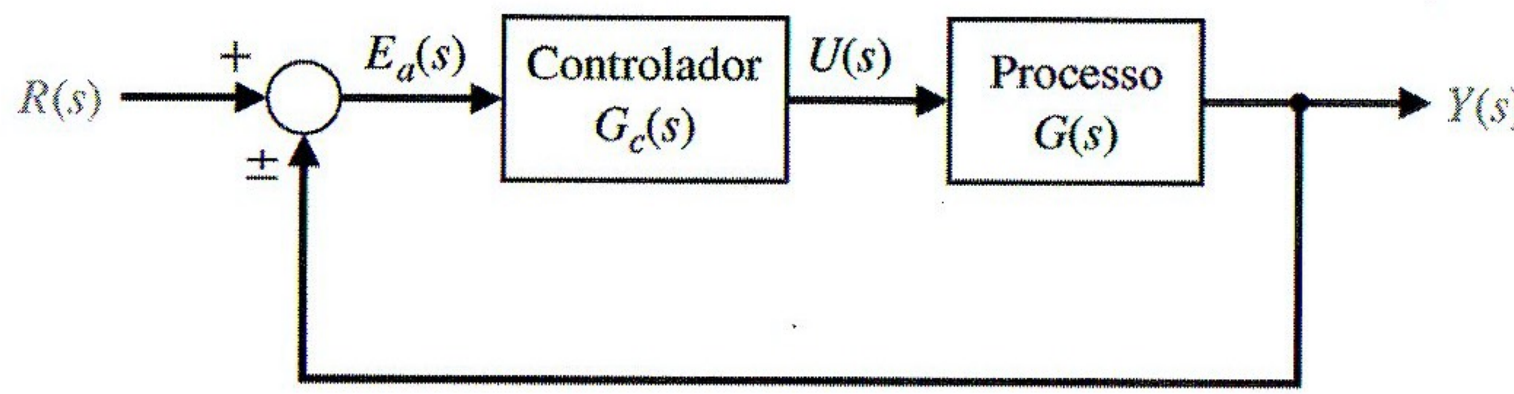


FIGURA 2.62
(a) Diagrama de blocos. (b) A função **parallel**.

Matlab ➠ Uso em Controle

- **Aritmética de blocos:**

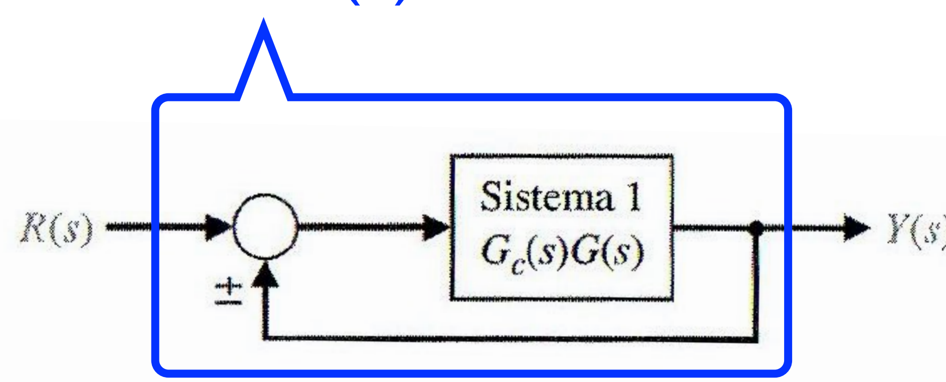


Fechando uma malha de controle (**feedback**)

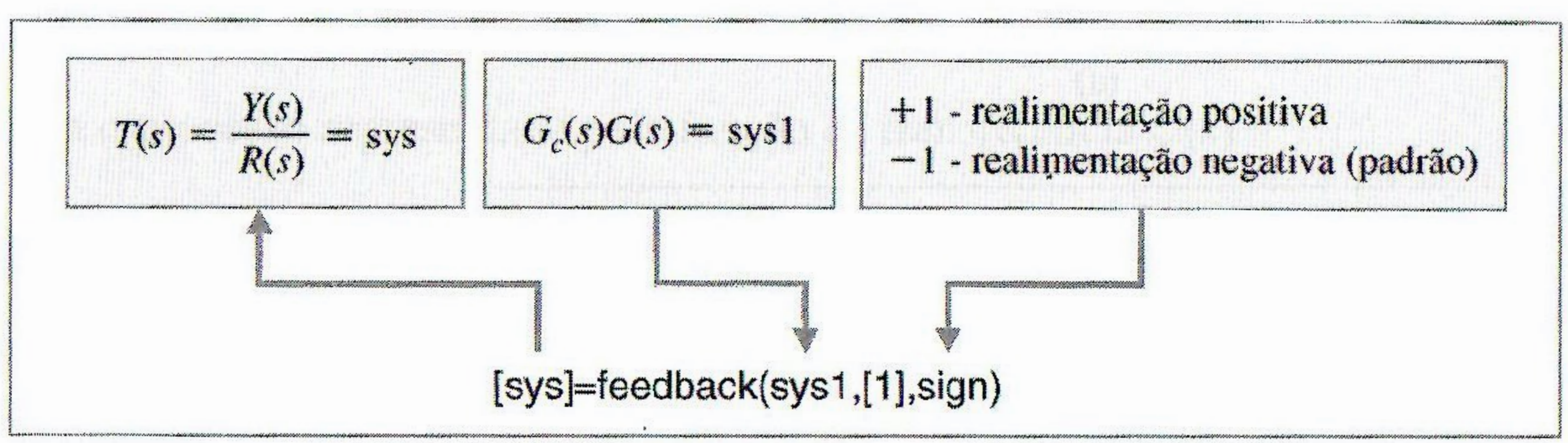
FTMF(s) ➨ `feedback()`

$$Y(s) = R(s) \cdot FTMF(s)$$

$$FTMF(s) = \frac{Y(s)}{R(s)}$$



(a)



(b)

FIGURA 2.64
(a) Diagrama de blocos. (b) A função **feedback** com realimentação unitária.

Matlab ➠ Uso em Controle

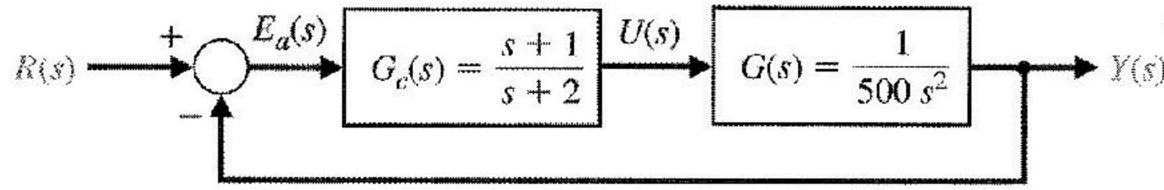
- **Aritmética de blocos:**

Fechando uma malha de controle (**feedback**)

Fechando uma malha de controle (**feedback**)

$$Y(s) = R(s) \cdot FTMF(s)$$

$$FTMF(s) = \frac{Y(s)}{R(s)}$$



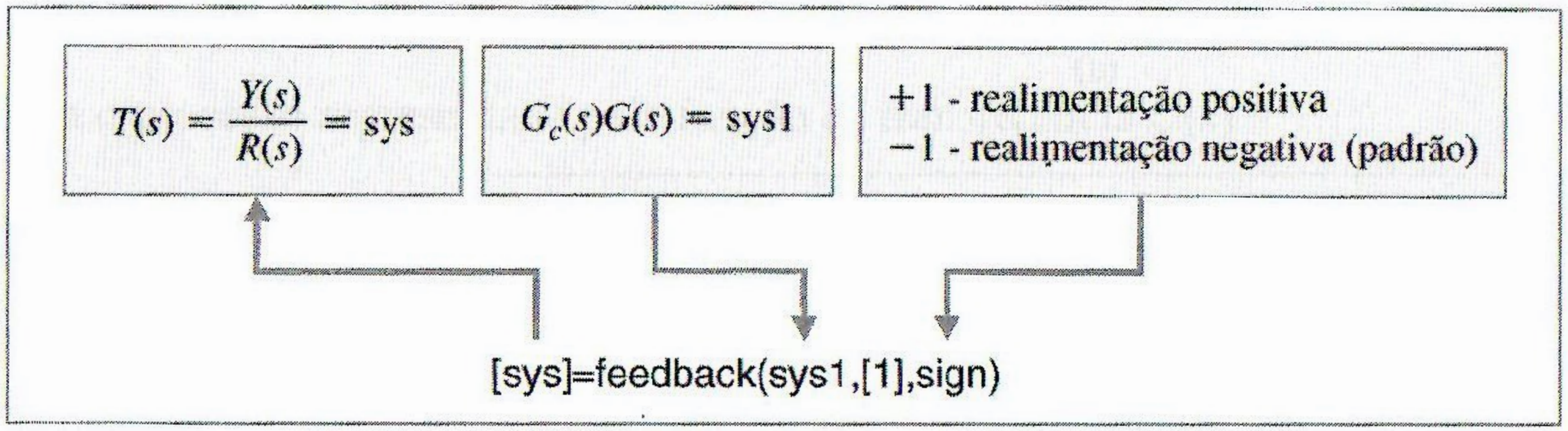
(a)

```
>>numg=[1]; deng=[500 0 0]; sys1=tf(numg,deng);
>>numc=[1 1]; denc=[1 2]; sys2=tf(numc,denc);
>>sys3=series(sys1,sys2);
>>sys=feedback(sys3,[1])

Transfer function:
            s + 1
-------------------------------
500 s^3 + 1000 s^2 + s + 1
```

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}$$

(b)

FIGURA 2.66 (a) Diagrama de blocos. (b) Utilização da função **feedback**.



(b)

FIGURA 2.64 (a) Diagrama de blocos. (b) A função **feedback** com realimentação unitária.

Matlab ➠ Uso em Controle

- **Aritmética de blocos:**

$$FTMF(s) = \frac{Y(s)}{R(s)}$$

$$FTMF(s) = \frac{G(s)}{1 + H(s) \cdot G(s)}$$

Fechando uma malha de controle (**`feedback`**)

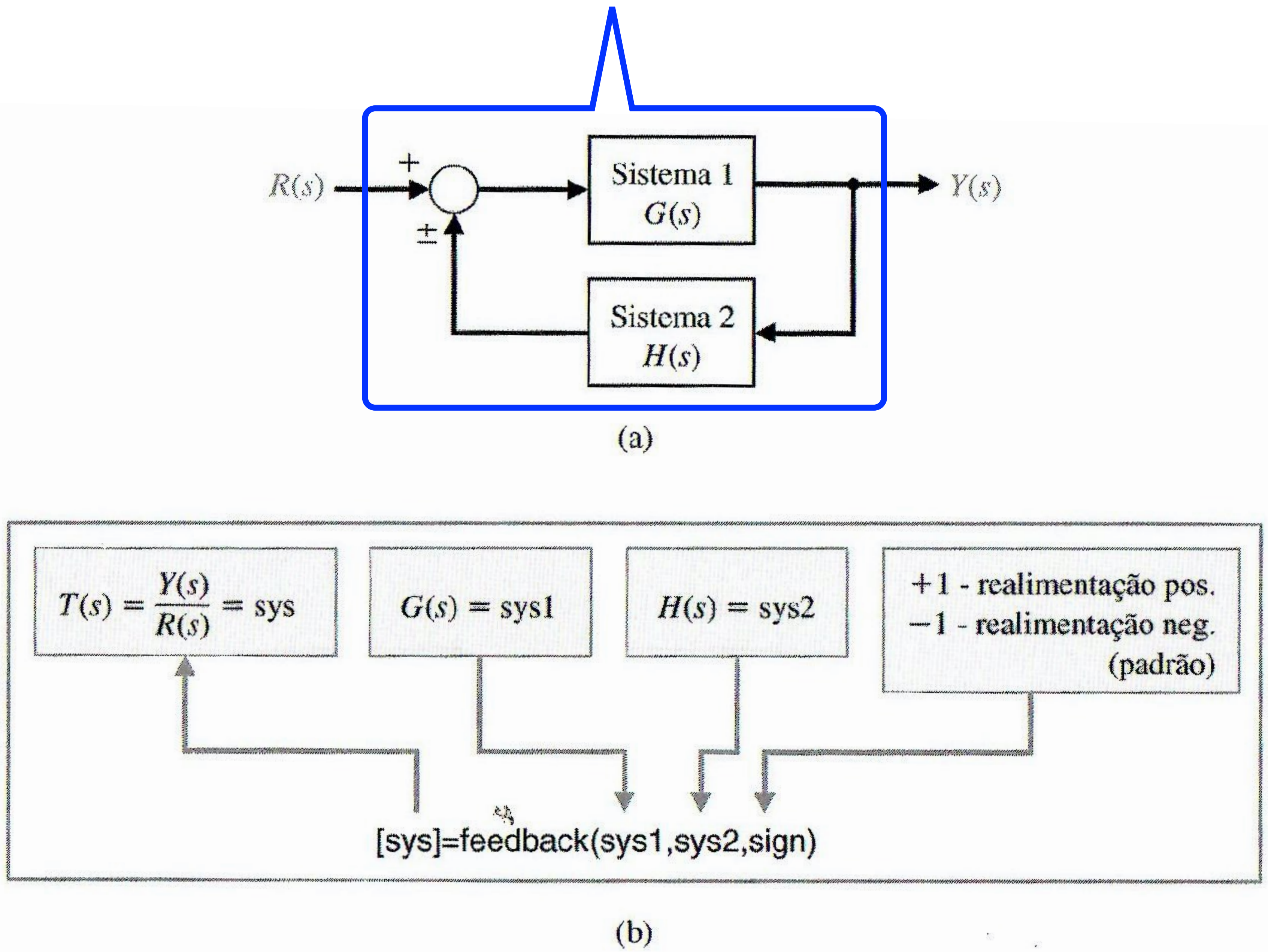
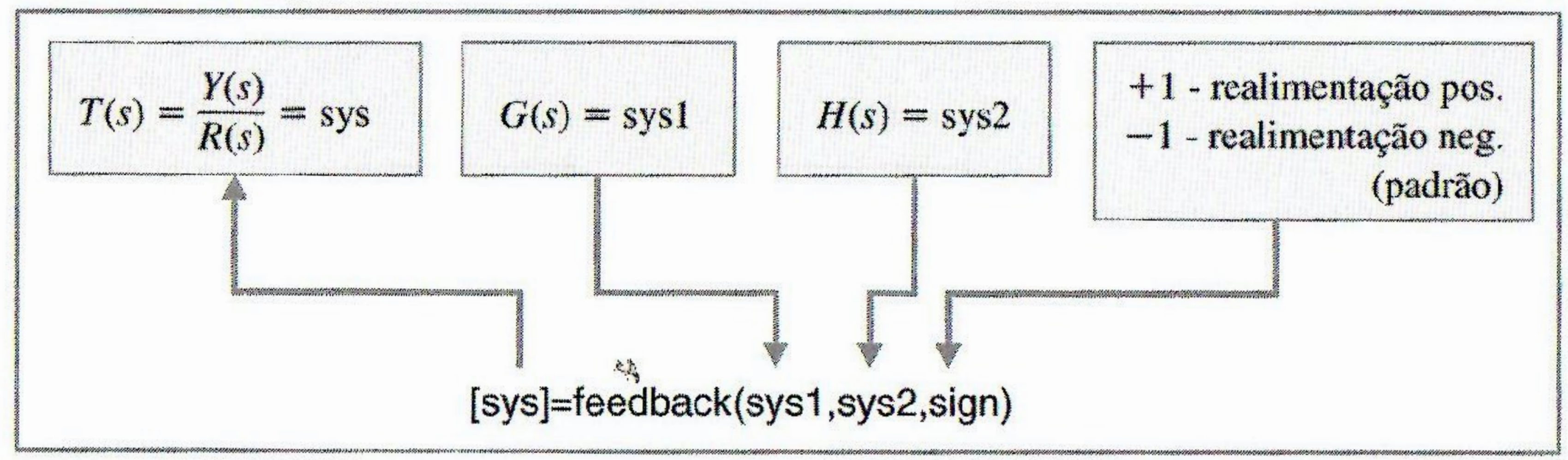


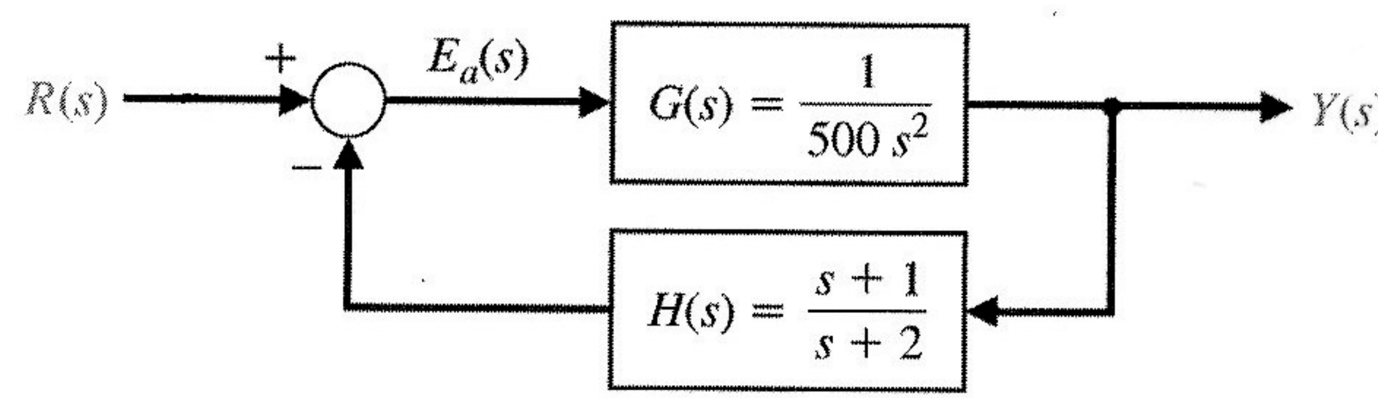
FIGURA 2.65
(a) Diagrama de blocos. (b) A função **feedback**.

Matlab ➠ Uso em Controle

- **Aritmética de blocos:**

Fechando uma malha de controle (**feedback**)





(a)

$$FTMF(s) = \frac{Y(s)}{R(s)}$$

$$FTMF(s) = \frac{G(s)}{1 + H(s) \cdot G(s)}$$

```
>>numg=[1]; deng=[500 0 0]; sys1=tf(numg,deng);
>>numh=[1 1]; denh=[1 2]; sys2=tf(numh,denh);
>>sys=feedback(sys1,sys2);
>>sys

Transfer function:
           s + 2
---------------------------
500 s^3 + 1000 s^2 + s + 1
```

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Matlab ➠ Uso em Controle

- **Simulações:**

Entrada
Degrau
(`step`)

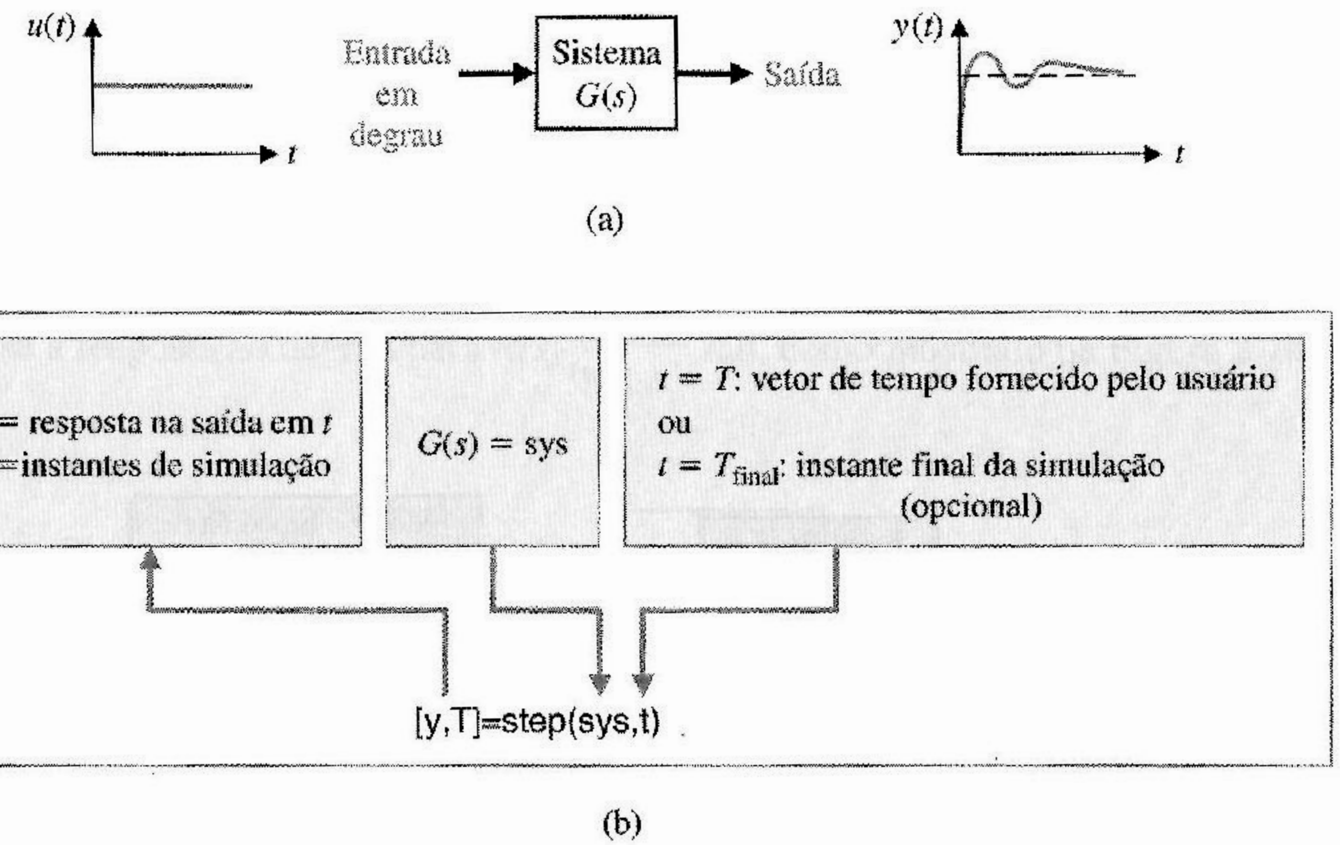
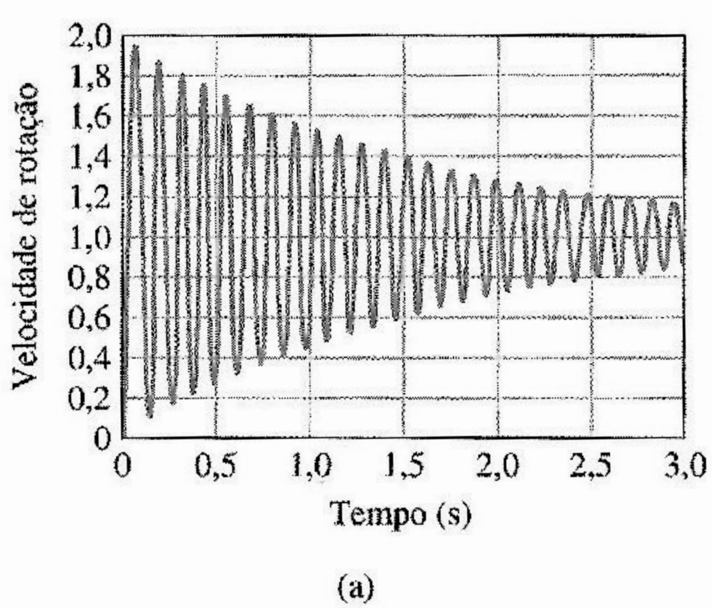


FIGURA 2.73
A função **step**.




```
% Esta sequência de instruções calcula a
% resposta ao degrau da velocidade de rotação
% do motor de tração
%
num=[5400]; den=[2 2.5 5402]; sys=tf(num,den);
t=[0:0.005:3];
[y,t]=step(sys,t);
plot(t,y),grid
xlabel('Tempo (s)')
ylabel('Velocidade de rotação')
```

(b)

FIGURA 2.74
(a) Resposta ao degrau da velocidade de rotação do motor de tração. (b) Sequência de instruções.

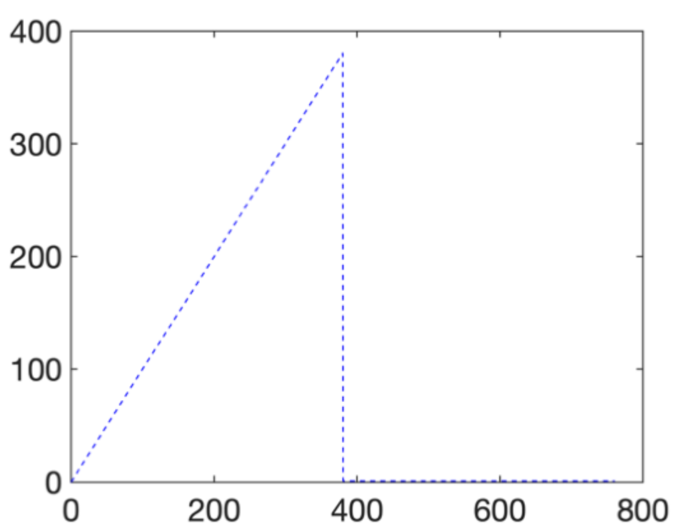
Matlab ➠ Uso em Controle

- **Simulações:**

Simulando Outras entradas

↓

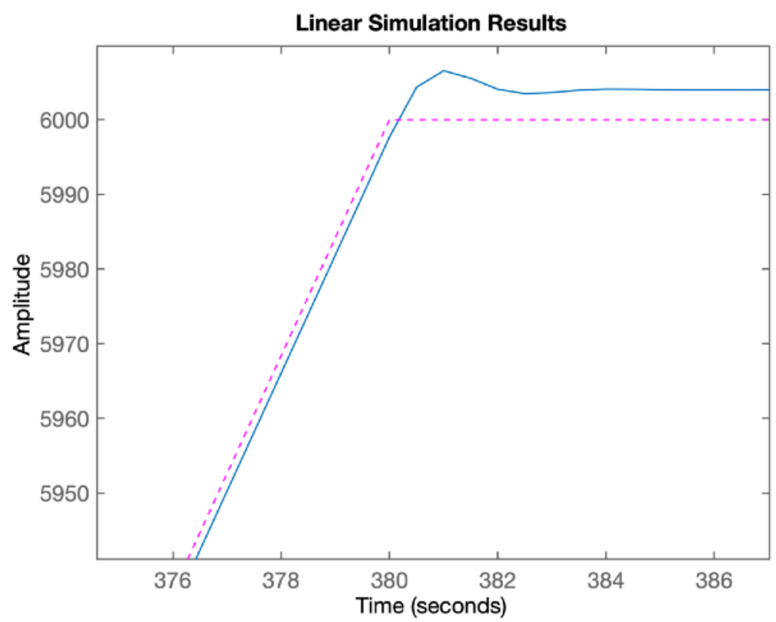
Criando sinal de entrada, referência

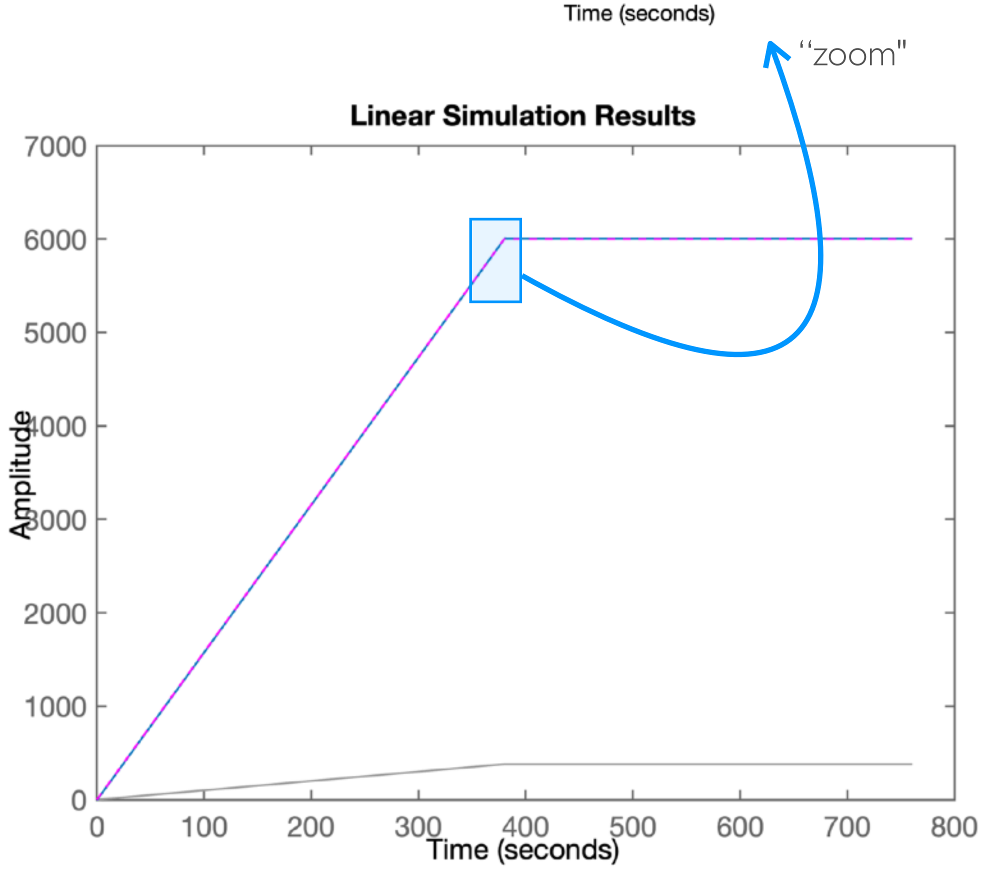



```
>> % Simulando rampa partida motor trifásico
>> y=[0:0.5:380]'; % variação tensão, incremento de 05, Volts
>> t=[0:0.5:380]'; % eixo do tempo, relacionado com y=f(t)
>> whos
  Name         Size                Bytes  Class     Attribute
  t          761x1                  6088  double
  y          761x1                  6088  double
>> % Note que nas linhas acima criamos a rampa
>> % Falta acrescentar o "degrau"
>> % Note a função ones()
>> ones(2,2)
ans =
     1     1
     1     1
>> % Acrescentando o degrau depois da rampa
>> y=[y; ones(761,1)]; % Note: vetor linha
>> size(y)
ans =
        1522           1
>> % Alguns cálculos preliminares
>> 380*2 % tempo final da onda
ans =
   760
>> 760/.5 % qtdade de pontos presentes no vetor t
ans =
        1520
>> 761/.5
ans =
        1522
>> t=[0: 0.5: 761]';
>> whos
  Name          Size                Bytes  Class     Attributes
  t          1523x1                 12184  double
  y          1522x1                 12176  double
>> % Ops, vetor t maior do que deveria, uma amostra à menos:
>> t=[0: 0.5: 760.5]';
>> plot(t, y, 'b--')
>> % Ops.: faltou multiplicar "ones" por 380 Volts
```

Matlab ➠ Uso em Controle

- **Simulacões:**






```
>> % Corrigindo a parte final do sinal anterior
>> y=[0:0.5:380]'; % variação tensão, incremento de 05, Volts
>> t=[0:0.5:380]'; % eixo do tempo, relacionado com y=f(t)
>> % Acrescentando o degrau depois da rampa
>> y=[y; 380*ones(761,1)]; % Nonte: vetor linha
>> t=[0: 0.5: 760.5]';
>> % Mostrando as primeiras 10 amostras [t y]
>> [t(1:10) y(1:10)]
ans =
         0         0
    0.5000    0.5000
    1.0000    1.0000
    1.5000    1.5000
    2.0000    2.0000
    2.5000    2.5000
    3.0000    3.0000
    3.5000    3.5000
    4.0000    4.0000
    4.5000    4.5000
>> plot(t, y, 'b--')
>> % Acrescentando o modelo de um motor
>> G = tf( 79, [1 2 5] );
>> zpk(G)
ans =
        79
  --------------
  (s^2 + 2s + 5)
Continuous-time zero/pole/gain model.
>> figure; lsim(G, y, t) % simula/apresenta resposta
>> FatorEscala=6000/380;
>> y_aux=FatorEscala*y; % Assim max(y_aux)=6000 RPM
>> hold on % sobrepões próximos gráficos no anterior
>> plot(t,y_aux,'m--'); % acrescenta referencia tracejada
>>
```

Criando sinal de entrada, referência
↓
Simulando (**lsim**)

Exemplo

- **PID analógico:**

$$G_c(s) = \frac{V_0(s)}{V_1(s)} = \frac{R_4R_2(R_1C_1s + 1)(R_2C_2s + 1)}{R_3R_1(R_2C_2s)}$$

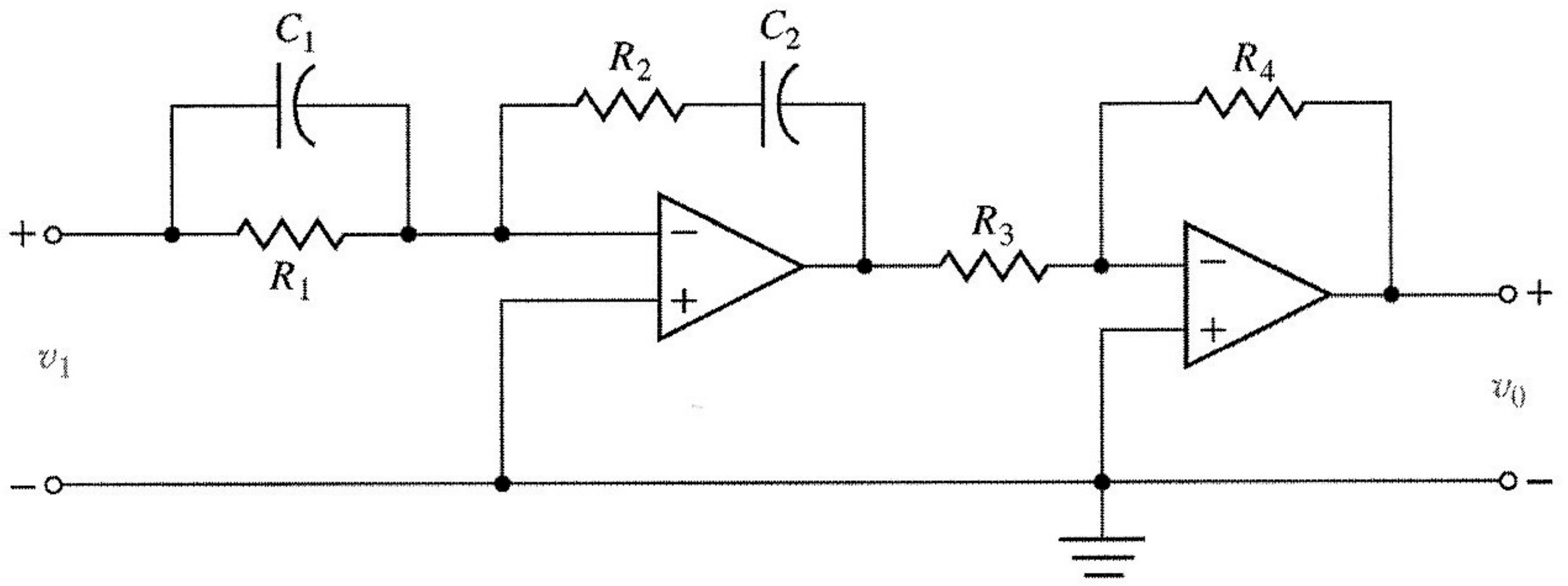


FIGURA 12.50 Circuito com amplificador operacional usado para controlador PID.

Ref. Bib.: DORF, Richard C. **Sistemas de controle modernos.** 13. Rio de Janeiro LTC 2018 1 recurso online ISBN 9788521635147.

DORF, Richard C.; BISHOP, Robert H. **Sistemas de controle modernos.** 12. ed. Rio de Janeiro: LTC, 2013. xx, 814 p. ISBN 9788521619956.