UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
CENTRO DE PROCESSAMENTO DE DADOS

ÁREA DE APOIO A USUÁRIOS
SUPORTE INFORMACIONAL

PC-MATLAB ver. 3.1-PC

LEA SILVIA MARTINS GONÇALVES

SETEMBRO / 1991
PUBLICAÇÃO 067/93
REIMPRESSÃO

ÍNDICE

01. INTRODUÇÃO

MATLAB é um programa interativo voltado para cálculo numérico, cujo elemento básico de informação é uma matriz que não requer dimensionamento.

O software PC-MATLAB é muito utilizado em cálculo numérico, cálculos científicos principalmente voltado para engenharia. Possui recursos específicos para tratamento de matrizes e facilidade para plotar gráficos.

O PC-MATLAB foi desenvolvido em linguagem C, sendo possível a construção de novas rotinas.

É aconselhado para os usuários que não gostam muito de programar nas linguagens tradicionais como : FORTRAN, PASCAL, C...

02. SISTEMA REQUERIDO

O PC-MATLAB requer o seguinte hardwarwe/software:

. IBM PC, PC/XT, PC/AT ou compatíveis
. no mínimo de 320Kbytes de memória
. versão superior a 2.0 do MS-DOS ou PC-DOS
. coprocessador matemático 8087, 80287 ou 80387
. no mínimo um driver
. 4 disquetes para cópia do software

03. INSTALAÇÃO DO PC-MATLAB

Para instalar o PC-MATLAB em um disco rígido é necessário:

- criar um novo subdiretório chamado PC-MATLAB

C> MD\MATLAB

C> CD\MATLAB

- inserir o DISCO DE PROGRAM #1 e digitar:

C:\MATLAB> A:INSTALL HARD

a partir daqui,a instalação prosseguirá observando-se a colocação dos disquetes e o tipo de hardware que se tem.

- Para instalar a parte das rotinas especiais voltadas para

Sistema de Controle, há necessidade da criação de um subdiretório específico para esta parte:

```
C:\> MD\MATLAB\CONTROL

C:\> CD\MATLAB\CONTROL
```

- Inserir o disquete CONTROL SYSTEM TOOLBOX e copiar todos os arquivos no subdiretório criado.

```
C:\MATLAB\CONTROL> COPY A:*.*
```

Após o término da instalação, é necessário:

- alterar O ARQUIVO AUTOEXEC.BAT :

```
PATH = C:\; C:\MATLAB
```

- alterar o ARQUIVO CONFIG.SYS :

```
FILES = 20
BUFFERS = 20
BREAK = ON
```

Para carregar o software **MATLAB** na memória, basta digitar:

```
C:\MATLAB > MATLAB    <ENTER>
```

A tela inicial do software é:

```
                    < P C - M A T L A B >
   (c) Copyright The MathWorks, Inc. 1984, 1985, 1986, 1987
                    All Rights Reserved
                Version 3.05        1-Apr-87

            HELP, DEMO and INFO are available

>>
```

O prompt padrão dO MATLAB é: >>

04. EDIÇÃO DE UM PROBLEMA

Pode-se introduzir os dados de um problema no **MATLAB** de várias formas.

ENTRADA DE UMA LISTA EXPLÍCITA DE DADOS

É bastante utilizado e é a forma mais simples.

É aconselhável para um conjunto pequeno de dados, pois uma vez errado há necessidade de nova digitação.

CRIAR ARQUIVO .M

Qualquer editor pode ser utilizado para facilitar a edição do problema, como WORD, WORDSTAR, SIDEKICK,etc..., uma vez que o PC-MATLAB não contém um editor embutido, com todos os recursos facilitados de um processador de palavras.

Cria-se o arquivo do tipo .M na forma não formatada, isto é, em ASCII. Os dados no arquivo estarão como uma lista explícita de elementos.

CARREGAR UM ARQUIVO EM ASCII

Utiliza-se o comando load para fazer a leitura do arquivo.

Os dados devem ter comprimentos fixos, os números separados por branco e o fim da linha por **<RETURN>**.

O resultado está em uma variável cujo nome é o nome do arquivo.

05. EXPORTAR DADOS/RESULTADOS PARA FORA DO MATLAB

Para matrizes não muito grandes, usar o comando diary para criar um arquivo listando todas as variáveis nesse arquivo.

Posteriormente, pode-se usar qualquer editor de texto para manipular o arquivo.

Pode-se também salvar as variáveis usando o comando **save**. Sair do MATLAB e usar o comando **translate** para converter arquivos do MATLAB para outros formatos possíveis como: ASCII, **binário, FORTRAN não formatado, DIF (planilha eletrônica),...**

06. LIMITES

Para um limitação de memória de 640 kbytes, limitação essa do MS-DOS tem-se:

- vetor limite é de 8.188 elementos
- matriz quadrada da ordem de 90

07. SIMBOLOGIA - CARACTERES ESPECIAIS

SÍMBOLO	DESCRIÇÃO
=	Comando de atribuição
[ ]	Delimitar elementos de matrizes e vetores
( )	Alterar precedência de expressões aritméticas
.	Ponto decimal
..	Continuação de comando na próxima linha
,	Separa argumentos de funções e elementos de matrizes e vetores
;	Indica o fim de uma linha com a supressão da impressão da resposta
%	Comentário
:	Geração de um vetor enumerado
!	Execução de comandos do Sistema Operacional

08. FORMATO DE SAÍDA

O comando **format** controla o formato de saída, que pode ser: exponencial, decimal e controla também o número de casas decimais significativas.

EXEMPLO

FORMATO	NÚMERO DE DÍGITOS SIGNIFICATIVOS	RESULTADO DE 3/4
format short	5	1.3333
format short e	5	1.3333e+000
format long	16	1.33333333333333
format long e	16	1.333333333333333e+000

09. COMANDOS E VARIÁVEIS

Há duas formas de se trabalhar com o PC-MATLAB: forma interpretada e forma programada.

Na forma interpretada após cada comando que termina teclando-se **ENTER** ou **RETURN**, há uma resposta do PC-MATLAB.

VARIÁVEL = EXPRESSÃO <ENTER>

ou simplesmente

EXPRESSÃO <ENTER>

EXEMPLO

```
2500 / 20
Resulta pelo PC-MATLAB :

ans =
        125
```

```
A=2500/20

Resulta pelo PC-MATLAB :

A=
    125
```

Na forma programada o último caracter da linha de comando é um ponto e vírgula ; e a impressão é suprimida.

EXEMPLO

```
A=[1 2 3; 4 5 6; 7 8 9] ;
B=A' ;
C=A*B ;
```

10. GERAÇÃO DE VETORES

Há possibilidade de gerar números através de intervalos definidos, utilizando um incremento fixo. A sua forma é:

valor_inicial: incremento: valor_final

EXEMPLOS

```
Z= 6:-1:1
Resulta como saída pelo PC-MATLAB:
Z=
   6 5 4 3 2 1
```

```
X= (0.8 : 0.2 : 1.4) ' ;
Y= exp(-X).* sin(X) ;
Z= [X Y]

Resulta como saída pelo PC-MATLAB:

Z =
      0.8000        0.3223
      1.0000        0.3096
      1.2000        0.2807
      1.4000        0.2430
```

11. TRABALHANDO COM MATRIZES

PC-MATLAB trabalha essencialmente com matrizes numéricas retangulares com possibilidade de ter elementos complexos.

Para entrar com dados de uma simples matriz há vários modos:

- lista explícita dos elementos
- gerar uma matris utilizando comandos e funções
- criar arquivos do tipo .M
- carregar dados de arquivos externos

Não há comandos de dimensões nem de tipos de dados.

A forma mais simples de introduzir uma MATRIZ é a lista explícita de dados:

- os elementos são separados por branco ou vírgula
- as linhas são separadas por ;
- o início e o fim por [e]

EXEMPLO

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

Resulta como saída na tela pelo PC-MATLAB :

A =
    1 2 3
    4 5 6
    7 8 9
```

ELEMENTOS DE UMA MATRIZ

Os elementos de um matriz podem ser simples elementos, ou calculados através de qualquer expressões.

EXEMPLO

```
X = [ -1.5    COS(45)     (1 + 5*2)/5^2 ]

Resulta como saída pelo PC-MATLAB:

X =

     -1.5       0.5253    0.4400
```

Para se incluir uma linha dentro de uma matriz, sem ter que redigitá-la basta dar o comando:

```
A = [ 1 2 3 ; 4 5 6  7 8 9 ] ;
A = [ A ; [10 11 12] ]

Resulta como saída pelo PC-MATLAB :

A =
        1  2  3
        4  5  6
        7  8  9
       10 11 12
```

12. OPERAÇÃO COM MATRIZES

Os operadores possíveis das matrizes são:

+	ADIÇÃO
-	SUBTRAÇÃO
*	MULTIPLICAÇÃO
/	DIVISÃO À DIREITA
\	DIVISÃO À ESQUERDA
^	EXPONENCIAÇÃO
'	TRANSPOSTA

TRANSPOSTA

O caracter aspa simples ['] calcula a transposta de uma matriz.

EXEMPLO

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

Resulta como saída pelo PC-MATLAB:

B =
     1  4  7
     2  5  8
     3  6  9
```

ADIÇÃO / SUBTRAÇÃO

EXEMPLO

```
A=[1 2 3;4 5 6;7 8 9];
B=A';
C=A+B

Resulta como saída pelo PC-MATLAB:

C=
    2    6   10
    6   10   14
   10   14   18
```

```
A=[1 2 3;4 5 6;7 8 9];
B=A';
C=B-A

Resulta como saída pelo PC-MATLAB:

C=
    0    2    4
   -2    0    2
   -4   -2    0
```

MULTIPLICAÇÃO

EXEMPLO

```
X=[-1 0 2];
Y=
     [ -2 ]
     [ -1 ]
     [  1 ]

X*Y
Resulta como saída pelo PC-MATLAB:
ans=
     4
```

A multiplicação é possível quando a 2ª dimensão do 1º operando é igual a 1ª dimensão do 2º operando. Quando isso não ocorrer o PC-MATLAB acusa um erro de dimensão.

DIVISÃO

EXEMPLO

```
A=[1 2 3;4 5 6;7 8 9];
B=A';
X=A\B
Resulta como saída pelo PC-MATLAB:
X=
       -0.2334    -0.3333    -1.1344
        0.4667    -2.3333    -3.7311
        0.1000     3.0000     5.1989
```

13. OPERAÇÃO COM CONJUNTOS

```
.+ ADIÇÃO
.- SUBTRAÇÃO
.* MULTIPLICAÇÃO
./ DIVISÃO À DIREITA
.\ DIVISÃO À ESQUERDA
.^ EXPONENCIAÇÃO
.' TRANSPOSTA
```

ADIÇÃO

EXEMPLO

```
x=[1 2 3];
y=[4 5 6];
z=x.+y
Resulta como saída pelo PC-MATLAB:
z=
     5    7    9
```

SUBTRAÇÃO

EXEMPLO

```
x=[1 2 3];
y=[4 5 6];
z=x.-y
Resulta como saída pelo PC-MATLAB:
z=
    -3    -3   -3
```

MULTIPLICAÇÃO

EXEMPLO

```
x=[1 2 3];
y=[4 5 6];
z=x.*y
Resulta como saída pelo PC-MATLAB:
z=
     4    10    18
```

EXPONENCIAÇÃO

EXEMPLO

```
x=[1 2 3];
y=[4 5 6];
z=x.^y
Resulta como saída pelo PC-MATLAB:
z=
     1    32    729
```

```
x=[1 2 3];
z=x.^2
Resulta como saída pelo PC-MATLAB:
z=
     1     4     9
```

14. NÚMEROS COMPLEXOS

A unidade imaginária i é definida formalmente como o número que elevado ao quadrado resulta -1.

A introdução da unidade imaginária conduz à generalização do conceito de número, aos números complexos, que desempenham um papel muito importante na álgebra e na análise e admitem algumas interpretações concretas em alguns problemas da geometria e da física.

Para se trabalhar com números complexos no PC-MATLAB, há necessidade de se gerar a unidade imaginária da seguinte forma:

```
i = sqrt (-1)
```

Uma vez definida a unidade imaginária é só usá-la.

EXEMPLOS

```
i= sqrt(-1) ;
Z= 3 + 4*i
```

Resulta como saída pelo PC-MATLAB:

```
Z=
   3.000 + 4.000i
```

```
i= sqrt(-1) ;
A= [1 2; 3 4] + i*[5 6; 7 8]
```

Resulta como saída pelo PC-MATLAB:

```
A=
    1.000 + 5.000i     2.000 + 6.000i
    3.000 + 7.000i     4.000 + 8.000i
```

15. OPERAÇÃO RELACIONAL E LÓGICA

Os operadores relacionais são:

```
<    menor que
<=   menor ou igual
>    maior que
>=   maior ou igual
==   igual
~=   diferente
```

EXEMPLO

```
2+2 ~= 4
Resulta pelo PC-MATLAB:
ans=
     0
2+2 == 4
Resulta pelo PC-MATLAB:
ans=
     1
```

Os operadore lógicos são:

```
&   AND
|   OR
~   NOT
```

16. ROTINAS ESPECIAIS PARA PROGRAMAÇÃO

OBSERVAÇÃO

Todas as funções/comandos embutidos do **PC-MATLAB** devem ser escritos em letras minúsculas.

ROTINAS	DESCRIÇÃO
if <cond> elseif end	Comando executável condicional
if <cond> else end	Comando executável condicional

for end	Gerar um loop enumerável
while <cond> end	Gerar um loop enquanto uma condição for verdadeira
break	Sair para fora de um loop for-end ou while-end
return	Retornar de uma função (arquivo .M)
pause	Parada até se apertar qualquer tecla do teclado

EXEMPLO

```
m=3;
n=3;
for i=1:m
     for j=1:n
          A(i,j)=1/(i+j-1);
     end
end
A

Tem-se como resposta do PC-MATLAB:

A=
     1.0000   0.5000   0.3333
     0.5000   0.3333   0.2500
     0.3333   0.2500   0.2000
```

```
n=1;
while prod(1:n) < 1.e100
     n=n+1 ;
end
n

Tem-se como resposta do PC-MATLAB:

n=   70
```

```
n=11;
if n < 0
     A=abs(n)
elseif round(n/2) == 0
     A=2*n
else
     A=3*n
end
A
```

Tem-se como resposta do PC-MATLAB:

```
A=
    33
```

17. FUNÇÕES MATEMÁTICAS ELEMENTARES

FUNÇÃO	DESCRIÇÃO
abs	Retorna o valor absoluto ou a magnitude de um número complexo
angle	Angulo do número complexo, que é seu argumento
sqrt	Raíz quadrada
real	Parte real de um número complexo
imag	Parte imaginária de um número complexo
conj	Conjugado de uma matriz complexa
round	Arredondamento de um número para o inteiro mais próximo
fix	Arredondamento de um número para o inteiro mais próximo até zero
floor	Arredondamento de um número para o inteiro mais próximo até infinito negativo
ceil	Arredondamento de um número para o inteiro mais próximo até infinito positivo
sign	Retorna para cada elemento da matriz, 1 quando o elemento > 0; 0 elemento = 0 e -1 elemento < 0

rem	Resto de uma divisão
sin	Seno
cos	Cosseno
tan	Tangente
asin	Arco-seno
acos	Arco-cosseno
atan	Arco-tangente
atan2	Arco-tangente para o quarto quadrante
sinh	Seno hiperbólico
cosh	Cosseno hiperbólico
tanh	Tangente hiperbólica
exp	Exponencial com base e
log	Logaritmo natural
log10	Logaritmo decimal
bessel	Função de Bessel
gamma	Função Gama
rat	Aproximação racional

18. TRAÇANDO GRÁFICOS

FORMA BÁSICA

Dado um vetor Y, plot (Y) produz um gráfico linear dos elementos de Y contra o índice dos elementos de Y.

EXEMPLO

```
Y = [0  .5  .8  .1  .7  .2] ;
plot (Y)
```

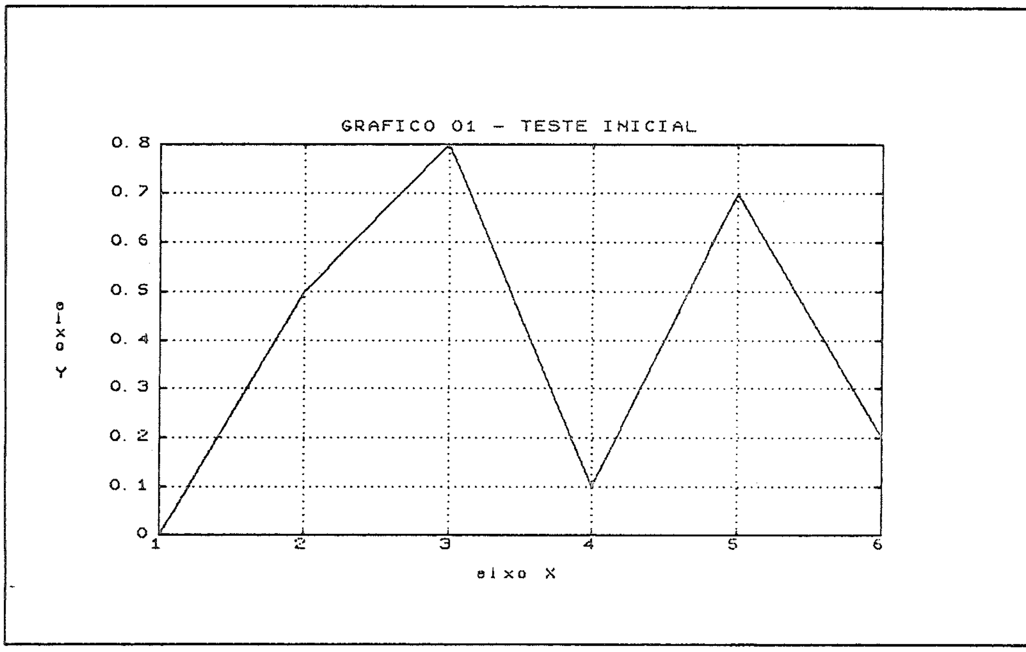
O gráfico será desenhado em uma tela gráfica.

A escala é feita automaticamente e o eixo X e Y também.

Pode-se melhorar a saída do gráfico dando títulos e nomes às ordenadas.

```
title ('Grafico 01 - Teste Inicial')
xlabel('eixo X')
ylabel('eixo Y')
grid
```

A tela final do gráfico gerado é:



Quando for necessária a impressão do gráfico, digitar o seguinte comando:

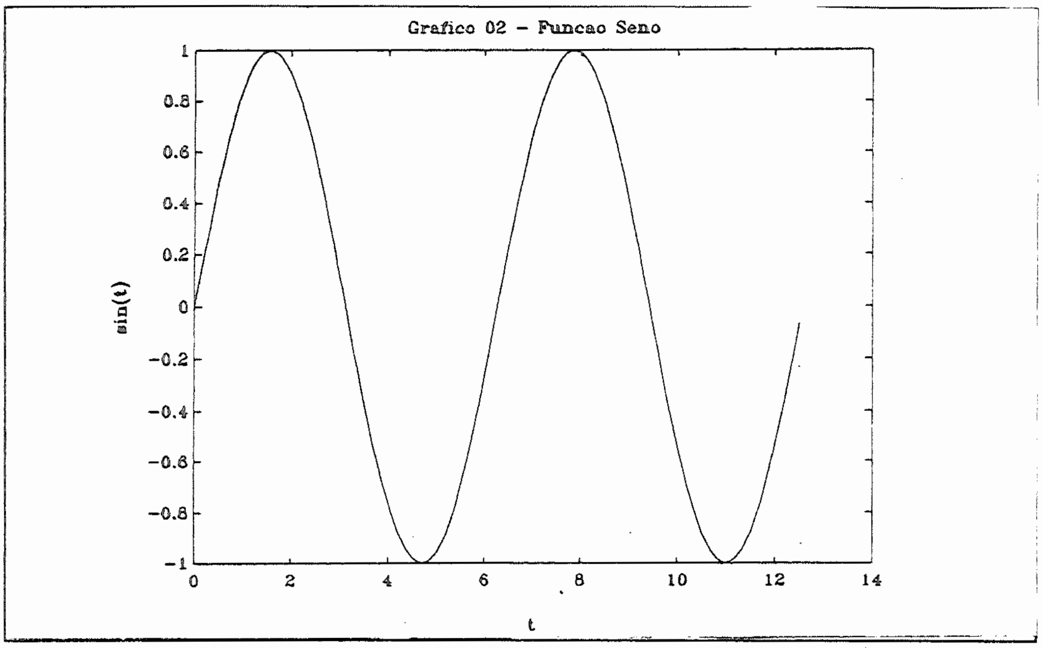
```
print
```

Dado 2 vetores X e Y, pode-se traçar os elementos de X contra os elementos de Y, dando o comando plot (X,Y).

EXEMPLO

```
t = 0: .1: 4*pi ;
Y = sin(t) ;
plot (t,Y)
title('Grafico 02 - Função Seno')
xlabe('t')
ylabel('sin(t)')
print
```

A impressão do gráfico é:



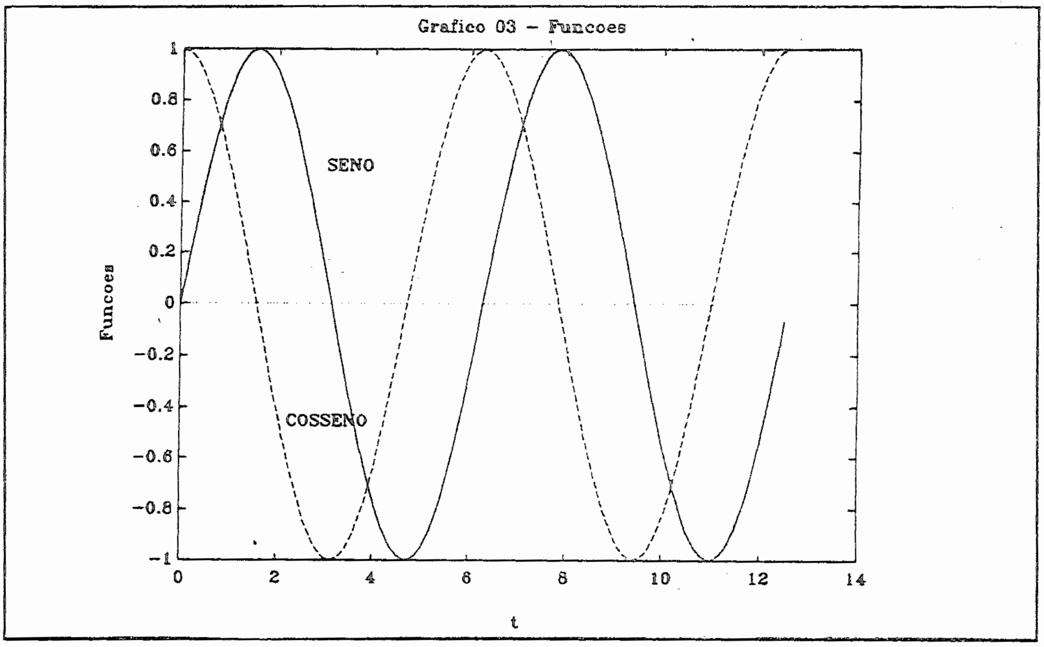
MÚLTIPLAS LINHAS

Pode-se traçar várias linhas num mesmo gráfico, bastando ter os vetores para isso.

plot (X_1, Y_1, X_2, Y_2, ..., X_n, Y_n)

EXEMPLO

```
t = 0: .1: 4*pi ;
Y = sin(t) ;
Z= cos(t);
W= 0*t;
plot(t,Y,t,Z,t,W)
xlabe('t')
ylabel('Funções')
title('Grafico 03 - Funções ')
text(3,0.,5,'SENO')
text(2.2,-0.5,'COSSENO')
```



Pode-se alterar o tipo de linhas, o tipo de ponto e a cor do gráfico.

TIPO DE LINHA		TIPO DE PONTO		CORES	
sólida	———	ponto	.	vermelha	r
tracejada	---------	mais	+	cinza	g
pontos	.....	asterisco	*	azul	b
traço-ponto	-.-.-	círculo	●	branca	w
		marca X	x	invisível	i

EXEMPLO

```
plot (X,Y,'+g') ;
```

Desenhar um gráfico XY cinza e com marca de ponto +.

DIFERENTES SISTEMAS DE COORDENADAS

plot Traçar gráfico linear XY

loglog	Traçar gráfico com escala $\log_{10} x \log_{10}$
semilogx	Traçar gráfico com eixo X linear e eixo Y $\log_{10}$
semilogy	Traçar gráfico com eixo X $\log_{10}$ e eixo Y linear
polar	Traçar gráfico usando coordenadas polares

GRÁFICO DE SUPERFÍCIE

mesh (Z) cria uma perspectiva em 3 dimensões e plota os elementos da matriz Z.

Um gráfico de superfície é definido pelos pontos de coordenadas Z sobre um "grid" retangular no plano XY.

Gráficos de superfícies são usados para visualizar matrizes muito grandes que não são fáceis de imprimí-las na forma numérica.

19. IMPRIMINDO GRÁFICOS

Após gerar um gráfico na tela gráfica, pode-se imprimir esse gráfico em impressora (comum,laser,...) ou até mesmo em ploter.

Há algumas formas de imprimí-los:

- . meta nome_arquivo
- . print
- . prtsc

META NOME-ARQUIVO

É gerado um arquivo com o gráfico em alta resolução, para posterior processamento, com o nome especificado e extensão .MET .

Após ser criado o arquivo **NOME.MET** é necessário utilizar o utilitário **GPP** (no ambiente do DOS), a fim de fazer a adaptação do arquivo com o dispositivo de saída existente (ploter, impressora matricial, impressora laser, ...).

GPP FILENAME /Ddispositivo de saída [/F /P /OP /OL /CS /CC]

onde

/F { é um argumento opcional, que pode ser usado para selecionar um novo nome, ou nova orientação na criação do arquivo (.EPS, por exemplo) pelo GPP.

/P é um argumento opcional que causa uma pausa quando há vários gráficos no arquivo.

/OP /OL são argumentos opcionais que selecionam a orientação com que o arquivo será impresso: horizontal ("portrait") ou vertical ("landscape"). A orientação horizontal que é a de melhor resolução é a default.

/CS /CC são argumentos que selecionam a qualidade do caracter texto utilizado:
CS-conjunto de caracteres de qualidade simples-linha única
CC-conjunto de caracteres de qualidade complexas-multi linhas

DISPOSITIVOS DE SAÍDA		EXTENSÃO GERADA
epsd	EPSON - qualidade rascunho	eps
epsf	EPSON - qualidade final	eps
jet	HP LASER JET PLUS (300 dpi)	jet
jet150	HP LASER JET PLUS (150 dpi)	jet
hpgl	Ploter compatível com HP	hpg

OBS: Após gerado o arquivo **NOME_ARQUIVO.MET** e utilizar o utilitário GPP (utilitário este que se encontra no subdiretório do MATLAB), tem-se um arquivo pronto para imprimir, em qualquer dispositivo de saída escolhido.
No ambiente do sistema operacional pode-se dar os comandos:

```
PRINT NOME_ARQUIVO.EXT
COPY NOME_ARQUIVO.EXT PRN
TYPE NOME_ARQUIVO.EXT>PRN
```

Pode-se também importar esses gráficos gerados após a utilização do utilitário GPP, por qualquer editor de texto: word, wordperfect, chiwriter,...

EXEMPLO

Cria-se o seguinte exemplo dentro do PC-MATLAB

```
t=0:.1:4*pi ;
y=sin(t) ;
plot(y)
meta EXEMPLO
quit
```

Passou a existir o arquivo EXEMPLO.MET.

Após sair do MATLAB, já no ambiente do DOS, chamar o utilitário GPP, para uma impressora do padrão EPSON - rascunho :

```
GPP EXEMPLO /DEPSD
```

Passou a existir o arquivo EXEMPLO.EPS.

Com esse arquivo qualquer comando de impressão do DOS pode-se imprimir o gráfico gerado anteriormente, como os comandos a seguir:

```
PRINT EXEMPLO.EPS
```

ou

```
COPY EXEMPLO.EPS PRN
```

ou

```
TYPE EXEMPLO.EPS >PRN
```

PRINT

Envia uma cópia em alta resolução do último plot feito na tela gráfica. Há problema de limitação de memória, sendo que pode-se não conseguir memória suficiente para impressão de um gráfico muito complexo.

PRTSC

Inicia uma cópia da tela gráfica. É semelhante a apertar as teclas <SHIFT> + <PRT SC>, quando o gráfico desejado estiver na tela gráfica.

OBS: Para teclar <SHIFT> <PRT-SC> na tela gráfica é necessário dar antes o comando do DOS GRAPHICS.

20. FUNÇÕES DE TELA

COMANDOS	DESCRIÇÃO
format	Alterar o formato dos dados na tela (short,long)
disp	Mostrar a matriz ou texto
fprintf	Imprimir um número formatado
clc	Limpar a tela de comando
home	Mover o cursor para o topo da tela
echo	Ativar/desativar as exibições de linhas individuais durante a execução de um arquivo .M.

21. FUNÇÕES PARA JANELA GRÁFICA

COMANDOS	DESCRIÇÃO
plot	Traçar gráfico linear XY
loglog	Traçar gráfico com escala $\log_{10}$x $\log_{10}$
semilogx	Traçar gráfico com eixo X linear e eixo Y $\log_{10}$
semilogy	Traçar gráfico com eixo X $\log_{10}$ e eixo Y linear
polar	Traçar gráfico usando coordenadas polares
mesh	Traçar gráfico de superfícies
bar	Desenhar barras
grid	Criar o grid do gráfico
title	Título do gráfico
xlabel	Título do eixo X
ylabel	Título do eixo Y
text	Nomes dos pontos de dados de um gráfico

axis	Criar uma escala manual dos eixos
hold	Parar a plotagem em uma tela
shg	Mostrar a janela gráfica
clg	Limpar a janela gráfica
subplot	Dividir a janela gráfica em várias sub-janelas
print	Enviar gráfico para impressora
prtsc	Enviar tela gráfica para impressora =SHIFT PRTSC
meta	Gerar uma arquivo com o gráfico criado (.MET)

22. FUNÇÕES PARA TRATAMENTO DE ARQUIVOS

COMANDOS	DESCRIÇÃO
chdir	Mudar o diretório corrente
delete	Deletar um arquivo
diary	Salvar a seção de trabalho em arquivo de disco
dir	Mostrar o diretório
load	Carregar variáveis do disco
save	Salvar variáveis no disco
translate	Mudar o tipo de dado (MATLAB,binário,ASCII,DIF)
type	Listar o conteúdo de um arquivo ou função
what	Mostra os nomes dos arquivos .M do disco

23. FUNÇÕES LÓGICAS E RELACIONAIS

FUNÇÕES	DESCRIÇÃO
any	Condição lógica

all	Condição lógica
find	Encontrar índices de elementos ≠ 0 em um vetor
isnan	Detectar NaNs
finite	Retornar 1 quando os elementos do vetor são finitos e 0 quando são infinitos
isempty	Retornar 1 se a matriz é vazia, 0 caso contrário

24. UTILITÁRIOS PARA MATRIZES

FUNÇÕES	DESCRIÇÃO
diag	Matriz diagonal
eye	Matriz identidade
ones	Matriz de constantes
zeros	Matriz de zeros
rand	Matriz randômica
logspace	Gerar logaritmamente um vetor
magic	A somatória de todas as linhas/colunas é igual
tril	Parte inferior de uma matriz triangular
triu	Parte superior de uma matriz triangular
toeplitz	Matriz Toeplitz
rsf2csf	Converter matriz real Schur para complexa
cond	Número condicional de uma matriz
det	Determinante de uma matriz
norm	Norma de uma matriz
rank	Ordem de uma matriz
rcond	Número condicional de uma matriz

25. FUNÇÕES DE DECOMPOSIÇÃO E FATORIZAÇÃO

FUNÇÕES	DESCRIÇÃO
chol	Fatorização de Cholesky
eig	Autovalor e autovetor
hess	Forma de Hessenberg
inv	Inversa de uma matriz
lu	Fatores da eliminação de Gauss
null	Espaço nulo
orth	Ortogonalização de uma matriz
pinv	Pseudo inversão de uma matriz
qr	Decomposição ortogonal-triangular
qz	Algoritmo QZ
rref	Redução escalar de linha usando eliminação Gauss-Jordan com pivotamento parcial
schur	Decomposição de Schur
svd	Decomposição de valor singular

26. FUNÇÕES POLINOMIAIS

FUNÇÕES	DESCRIÇÃO
poly	Características polinomiais
roots	Raízes
polyval	Avaliação polinomial
polyvalm	Avaliação da matriz polinomial
conv	Multiplicação entre dois polinômios
deconv	Divisão entre dois polinômios

residue	Expansão em frações parciais
polyfit	Aproximação de uma curva por um polinômio

27. FUNÇÕES PARA ANÁLISE DE DADOS

FUNÇÕES	DESCRIÇÃO
max	Valor máximo
min	Valor mínimo
mean	Média
std	Desvio padrão
median	Mediana
sort	Ordenar
sum	Soma dos elementos
prod	Produto dos elementos
cumsum	Soma acumulada dos elementos
cumprod	Produto acumulado dos elementos
diff	Derivada aproximada
hist	Histograma
table1	Retornar uma linha linearmente interpolada - TAB
corr	Matriz de correlação
cov	Matriz de covariância
spline	Interpolação cúbica Spline

28. FUNÇÕES PARA PROCESSAMENTO DE SINAIS

FUNÇÕES	DESCRIÇÃO
abs	Magnitude - valor absoluto
conv	Convolução
conv2	Convolução 2D
cov	Covariância
deconv	Deconvolução
dft	Transformada discreta de Fourier
fft	Raiz quadrada da transformada rápida de Fourier
fft2	Transformada discreta de Fourier 2D
idft	Transformada inversa
ifft	Inversa da transformada discreta de Fourier
ifft2	Inversa da transformada discreta de Fourier 2D
fftshift	Rearranjos das saídas FFT p/ centro do espectro
filter	Implementação direta de filtro
freqz	Tempo de resposta - frequência digital
freqs	Tempo de resposta - frequência analítica
xcorr	Função correlação Cross
xcorr2	Função correlação Cross 2D

EXEMPLOS

```
A=[1 2 3;7 8 9;5 3 2];
det(A)

ans =
     -6

rank(A)

ans =
     3
```

```
cond(A)
ans =
      111.9999

inv(A)
ans =
      1.8333   -0.8333    1.0000
     -5.1667    2.1667   -2.0000
      3.1667   -1.1667    1.0000

eig(A)
ans =
      13.2328
      -2.4202
       0.1873

p=poly(A)
p=
      1.0000   -11.0000   -30.0000   6.0000

roots(p)
ans =
      13.2328
      -2.4202
       0.1873
```

29. FACILIDADE DO HELP

O HELP dentro do MATLAB é muito útil. Pode-se ter HELP geral ou HELP de uma função específica.

EXEMPLO

HELP

intro	<	chol	end	global	macro	qz	sprintf
help	>	clc	eps	grid	magic	rand	sqrt
demo	=	clear	error	hess	max	rat	startup
[	&	clg	eval	hold	memory	rcond	string
]	|	clock	exist	home	mesh	real	subplot
(	~	conj	exit	ident	meta	relop	sum
)	abs	contour	exp	if	min	rem	svd
.	all	cos	expm	imag	nan	return	tan
,	ans	cumprod	eye	inf	nargin	round	text
;	any	cumsum	fft	input	norm	save	title

%	acos	delete	filter	inv	ones	schur	type
!	asin	det	find	isnan	pack	script	what
:	atan	diag	finite	keyboard	pause	semilogx	while
'	atan2	diary	fix	load	pi	semilogy	who
+	axis	dir	floor	log	plot	setstr	xlabel
-	balance	disp	flops	loglog	polar	shg	ylabel
*	break	echo	for	logop	prod	sign	zeros
\	casesen	eig	format	ltifr	prtsc	sin	
/	ceil	else	fprintf	ltitr	qr	size	
^	chdir	elseif	function	lu	quit	sort	

Para qualquer das funções existentes é possível pedir uma descrição dela. Por exemplo, informações sobre a função sort:

```
HELP sort
```

```
SORT(X) ordena cada coluna de X na ordem ascendente.
[Y,I] = SORT (X) também retorna matriz I contendo os índices
usados na ordenação. Se X é um vetor, Y = X(I). Quando X é
matriz complexa, os elementos são ordenados pelo ABS(X).
```

30. TUTORIAL

Existe um exemplo chamado DEMO que é uma introdução ao MATLAB, dividido em todos os ítens principais do MATLAB.

Os comandos básicos e principais são descritos na forma de exemplos.

31. SISTEMA DE CONTROLE TOOLBOX

MATLAB trabalha somente com uma forma de objeto que é uma matriz numérica retangular com possibilidade de elementos complexos.

O SISTEMA DE CONTROLE TOOLBOX pode se usado com modelos de sistema lineares e invariantes no tempo (LTI) .

Alguns tipos de modelos de sistema LTI são:

- Espaço de Estado
- Ganho de Polos e Zeros
- Função Transferência
- Frações Parciais
- Tempo contínuo
- Tempo discreto

ESPAÇO DE ESTADO

Um sistema de equações diferenciais LTI pode também ser representado como um conjunto de equações diferenciais de 1ª ordem.

As equações podem ser escritas:

$\dot{x} = Ax + Bu$

$y = Cx + Du$

onde

u - vetor de nu entradas de controle
x - vetor de ns elementos de estado
y - vetor de ny saídas

Sistema de Espaço de Estado são facilmente representados no MATLAB:

A,B,C e D são matrizes e recebem tratamento individual.

EXEMPLO

Um sistema de 2ª ordem cosistindo de um par de polos com:

frequência natural W_n=1.5

fator de amortecimento ξ=0.2

Para entrar este sistema na forma de Espaço de Estado digitar:

```
Wn=1.5;
z=0.2;
a=[  0       1
   -Wn^2  -2*z*Wn ];
b=[  0
   -Wn^2 ];
c=[1 0 ];
d=0;
```

A representação de Espaço de Estado é o modelo mais natural em MATLAB.

Para sistemas MIMO (multi-input-multi-output) a representação Espaço de Estado é uma forma muito conveniente para trabalhar.

FUNÇÃO TRANSFERÊNCIA

Uma representação equivalente a Espaço de Estado é a Função de Transferência pela Transformada de LAPLACE, descrita:

$$Y(s) = H(s)\ U(s)$$

$$H(s) = C(sI-A)^{-1}B + D$$

onde

H(s) requer matriz tri-dimensional na maioria dos casos

As dimensões de H(s) são ny linhas, por nu colunas por ns+1 de profundidade, onde ns+1 é o número de coeficientes polinomiais.

Devido as variáveis no MATLAB serem de 2 dimensões nós limitamos a representação desse sistema em SIMO, com uma entrada única de u:

$$H(s) = \frac{N(s)}{q(s)} = \frac{N(1)s^{nn-1} + N(2)s^{nn-2} + \ldots + N(nn)}{q(1)s^{nq-1} + q(2)s^{nq-2} + \ldots + q(nq)}$$

onde:

q - vetor linha usado para conter os coeficientes do denominador em potência decrescente de s.

N - matriz que contém os coeficientes do numerador com tantas linhas quantas forem as saídas do vetor y.

EXEMPLO

Considerando o sistema SIMO:

$$H(s) = \frac{\begin{bmatrix} 3s + 2 \\ s^3 + 2s + 5 \end{bmatrix}}{3s^3 + 5s^2 + 2s + 1}$$

Tem-se como entrada para o MATLAB:

```
num=[ 0 0 3 2
      1 0 2 5 ];
den=[ 3 5 2 1 ];
```

GANHO DE POLOS E ZEROS

Uma Função de Transferência pode ser expressa em fatores ou na forma de Ganho de Polos e Zeros, a qual para um sistema SIMO em MATLAB é:

$$H(s) = \frac{Z(s)}{p(s)} = k\ \frac{(s+Z(1))\ (s+Z(2))\ \ldots\ (s+Z(3))}{(s+p(1))\ (s+p(2))\ \ldots\ (s+p(3))}$$

de uma única entrada.

No MATLAB existe uma convenção que raízes polinomiais são armazenadas em vetores colunas, cujo vetor linha contém os coeficientes polinomiais.

Portanto numa forma fatorada, o vetor coluna **p** contém a localização do polo do denominador da função de transferência.

Os zeros do numerador são armazenados na coluna de uma matriz **Z** com tantas colunas quantas forem as do vetor saída **y**.

Os ganhos para cada numerador da função de transferência estão no vetor coluna K. Para sistema SISO, K é um escalar.

As funções **poly** e **root** convertem-se entre si.

EXEMPLO

```
p= [1 3 5 2];
r=roots(p)
Tem-se como resposta pelo MATLAB:
r=
      -1.2267    +1.4677i
      -1.2267    -1.4677i
      -0.5466
pp=poly(r)
Tem-se como resposta pelo MATLAB:
pp=
      1.0000   3.0000   5.0000   2.0000
```

Para um sistema SIMO, tem-se:

$$H(s)= \frac{Z(s)}{p(s)} = \frac{\begin{bmatrix} 3(s+12) \\ 4(s+1)(s+2) \end{bmatrix}}{(s+3)(s+4)(s+5)}$$

A entrada pelo MATLAB é:

```
k=[ 3; 4 ];
Z=[-12  -1
   inf  -2];
p=[-3
   -4
   -5];
```

FRAÇÕES PARCIAIS

Uma Função de Transferência pode também ser expressa como uma Fração Parcial numa forma expandida ou reduzida, a qual para um sistema SISO é:

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

onde:

p - vetor coluna que contém os polos

r - vetor coluna que contém os resíduos correspondentes aos ganhos em p

k - vetor linha que contém uma parte imprópria da Função de Transferência

Função de Transferência são convertidas para/da Fração Parcial expandindo a forma, usando a função **residue**.

TEMPO DISCRETO

Sistemas invariantes no tempo (LTI) são representados no MATLAB da mesma forma que os Sistemas de Tempo Contínuo:

. Espaço de Estado
. Ganho de Polos e Zeros
. Forma de residue

Um sistema de equações diferenciais LTI pode também ser representado como um conjunto de equações de 1ª ordem.

Na forma de matriz ou de Espaço de Estado, as equações podem ser escritas:

$$x[n+1] = Ax[n] + Bu[n]$$
$$y[n] = Cx[n] + Du[n]$$

onde:

u - vetor de entradas de controle
x - vetor de estado
y - vetor de saída

Uma representação equivalente é a Função de Transferência transformada-Z, descrita abaixo:

$$Y(z) = H(z)\,U(z)$$
$$H(z) = C(zI - A)^{-1}\,B + D$$

ou na forma SIMO, para uma entrada única de u:

$$H(z) = \frac{N(z)}{q(z)} = \frac{N(1) + N(2)z^{-1} + \ldots + N(nn)z^{-(nn-1)}}{q(1) + q(2)z^{-1} + \ldots + q(nq)z^{-(nq-1)}}$$

onde:

q - vetor que contém os coeficientes do denominador em potência decrescente de 1/z

N - matriz que contém os coeficientes do numerador com tantas linhas quantas forem as saídas y

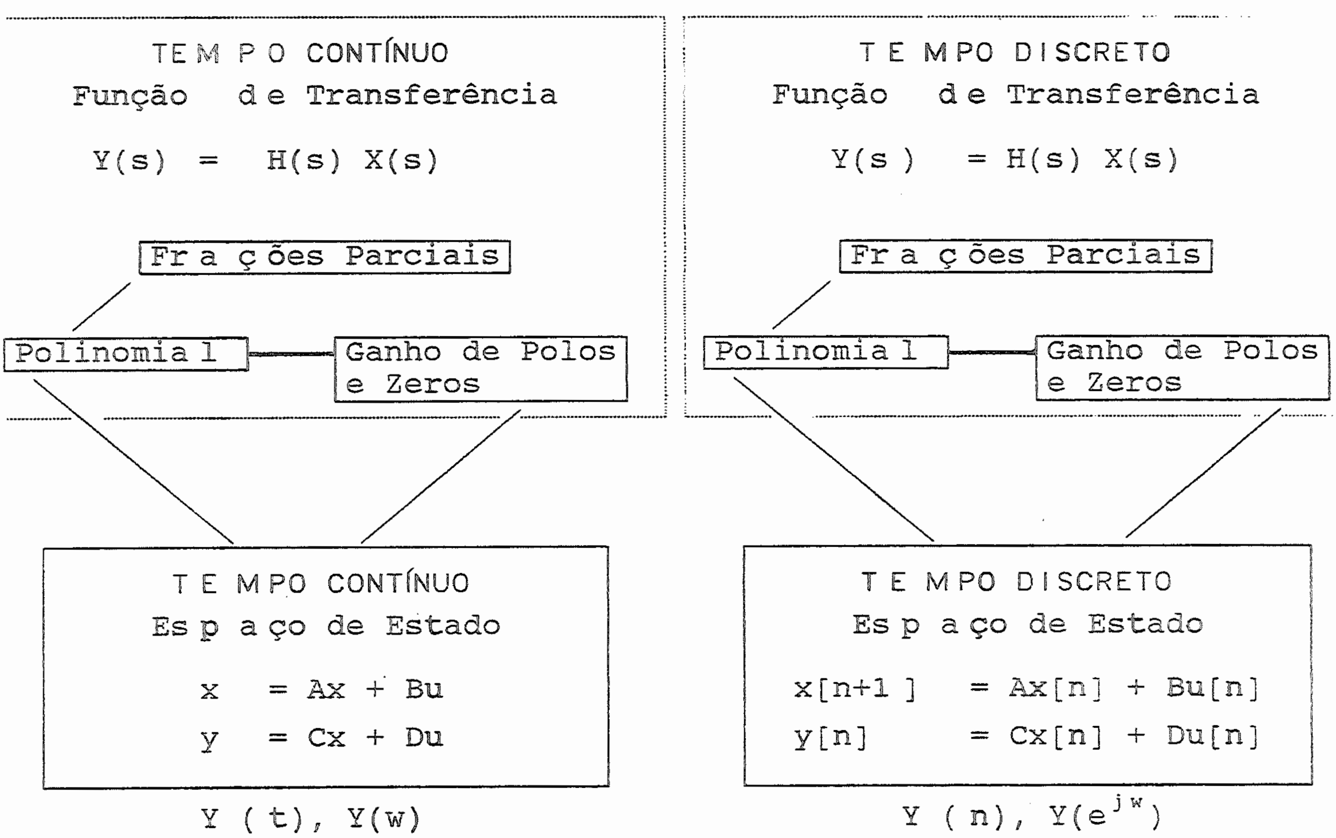
Se o sistema é multi-saídas e alguns dos numeradores são de ordem menor que os seguintes, coloca-se zero no numerador ausente.

A forma fatorada de Ganho de Polos e Zeros é:

$$H(z) = \frac{Z(z)}{p(z)} = k \, \frac{(z^{-1}+ Z(1))(z^{-1}+ Z(2))\dots(z^{-1}+ Z(n))}{(z^{-1}+ p(1))(z^{-1}+ p(2))\dots(z^{-1}+ p(n))}$$

CONVERSÃO DE SISTEMAS

REPRESENTAÇÃO DE SISTEMAS



31.1 FUNÇÕES PARA CONSTRUÇÃO DE MODELOS

FUNÇÃO	DESCRIÇÃO
append	Criar um sistema dinâmico
connect	Interconecção de sistema
parallel	Conecção a um sistema paralelo
series	Conecção a um sistema de séries
ord2	Gerar A,B,C,D para um sistema de 2ª ordem

31.2 FUNÇÕES PARA CONVERSÃO DE MODELOS

Um conjunto de funções segundo modelo LTI podem ser convertidas em várias representações:

FUNÇÃO	DESCRIÇÃO
ss2tf	Espaço de Estado para Função de Transferência
ss2zp	Espaço de Estado para Polo Zero
tf2ss	Função de Transferência para Espaço de Estado
tf2zp	Função de Transferência para Polo Zero
zp2tf	Polo Zero para Espaço de Estado
zp2ss	Polo Zero para Função de Transferência
residue	Função de transferência para residuo
c2d	Contínuo para discreto
d2c	Discreto para contínuo

As várias representações podem ser obtidas:

```
[num,den] = ss2tf(a,b,c,d,iu)        [z,p,k]   = tf2zp(num,den)
[z,p,k]   = sstzp(a,b,c,d,iu)        [a,b,c,d] = zp2ss(z,p,k)
[a,b,c,d] = tf2ss(num,den)           [num,den] = zp2tf(z,p,k)
```

```
[r,p,k]     = residue(num,den)        [ad,bd]  = c2d(a,b,Ts)
[num,den] = residue(r,p,k)            [a,b]    = d2c(ad,bd,Ts)
```

31.3 FUNÇÕES PARA REALIZAÇÃO DE MODELOS

FUNÇÃO	DESCRIÇÃO
ctrbf	Controlabilidade na forma de escada
obsvf	Observabilidade na forma de escada
minreal	Realização mínima e cancelamento Polo Zero
balreal	Realização balanceada
modreal	Redução da ordem de um modelo
dbalreal	Realização balanceada discreta
dmodreal	Redução da ordem de um modelo discreto

31.4 PROPRIEDADES DE UM MODELO

FUNÇÃO	DESCRIÇÃO
damp	Fator de amortecimento e frequência natural
gram	Controlabilidade e observabilidade de "Gramians"
dgram	Controlabilidade e observabilidade discreta de "Gramians"
ctrb	Matriz de controlabilidade
obsv	Matriz de observabilidade
tzero	Zero de transmissão

31.5 FUNÇÕES PARA RESPOSTA NO TEMPO

FUNÇÃO	DESCRIÇÃO
impulse	Resposta a função impulso
step	Resposta a função degrau
lsim	Simulação contínua a entradas arbitrárias
dimpulse	Resposta a amostra unitária no intervalo tempo
dstep	Resposta ao degrau no intervalo de tempo
dlsim	Simulação discreta a entradas arbitrárias
filter	SISO - simulação por transformada Z

31.6 FUNÇÕES PARA RESPOSTA DE FREQUÊNCIA

FUNÇÃO	DESCRIÇÃO
bode	Diagrama Bode
nyquist	Diagrama Nyquist
dbode	Diagrama discreto Bode
freqz	Resposta de frequência por transformada Z
freqs	Resposta de frequência por transformada Laplace

31.7 FUNÇÕES PARA SELEÇÃO DE GANHO

FUNÇÃO	DESCRIÇÃO
lqr	Projeto de um regulador linear-quadrático
lqe	Projeto de um processador linear-quadrático
dlqr	Projeto de um regulador linear-quadrático discreto

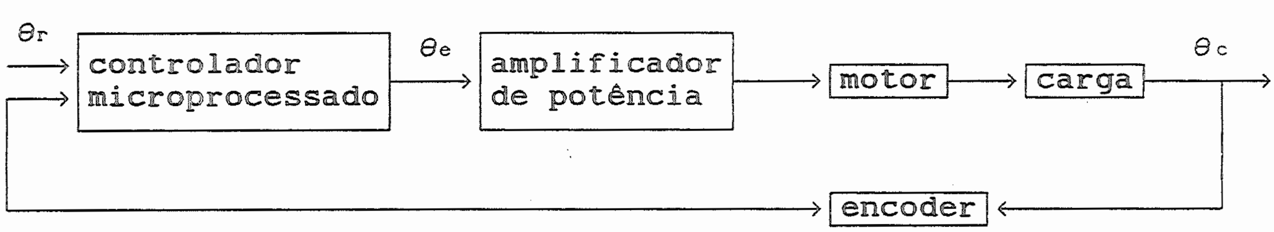
dlqe	Projeto de um processador linear-quadrático discreto
margin	Margem de ganho e margem de fase
place	Definição de polos
rlocus	Lugar geométrico das raízes

31.8 UTILITÁRIOS

FUNÇÃO	DESCRIÇÃO
lyap	Equação Lyapunov
dlyap	Equação discreta Lyapunov
fixphase	Desempacotar a fase para plotagem BODE
abcdcheck	Verificação de consistência de um conjunto (A,B,C,D)
nargcheck	Verificação de números dos arquivos do tipo .M

31.9 EXEMPLO COMPLETO

Considere o seguinte sistema de controle de posição de uma impressora:



A função de transferência malha fechada, é dada por:

$$\frac{\theta_c(s)}{\theta_r(s)} = \frac{K_s.K.\ K_i}{R_a.B.\ s\ (1 + \tau_a.s)\ (1 + \tau.s) + K_b.K_i.s + K_s.K.\ K_i}$$

A função de transferência malha aberta, é dada por:

$$G(s) = \frac{\theta_c(s)}{\theta_e(s)} = \frac{K_s.K.\ K_i}{R_a.B.\ s\ (1 + \tau_a.s)\ (1 + \tau.s) + K_b.K_i.s}$$

onde:

$\tau_a = L_a \ / \ R_a \cong 0$ $\qquad$ $\tau = J \ / \ B = 0.025$ seg

$B = 0.06$ oz-in-seg $\qquad$ $K_s = 1$ v/rad

$R_a = 5$ ohms $\qquad$ $K_i = 3$ oz-in/A $\qquad$ $K_b = 0.02126$ v/rad/seg

Pede-se:

a) Simular a resposta do sistema ao degrau unitário e rampa unitária para K = 1, 2.94 e 100. Traçar o lugar das raízes.

b) Traçar o diagrama de BODE

c) Traçar o diagrama de Nyquist

d) Adicionar um compensador PID e refazer osd ítens a,b,c. A função de transferência malha aberta é:

$$G(s) = \frac{400\ (\ s^2.\ K_d + s.\ K_p + K_i\ )}{s^2\ (\ s + 48.5\ )}$$

onde:

$K_d = 1.0$ $\qquad$ $K_p = 100.0$ $\qquad$ $K_i = 5.0$

Resolução do problema

a) Após as substituições tem-se as seguintes funções de transferências :

$$\frac{\theta_c(s)}{\theta_r(s)} = \frac{400 * K}{s^2 + 48.5\ s + 400 * K} \quad \text{(malha fechada)}$$

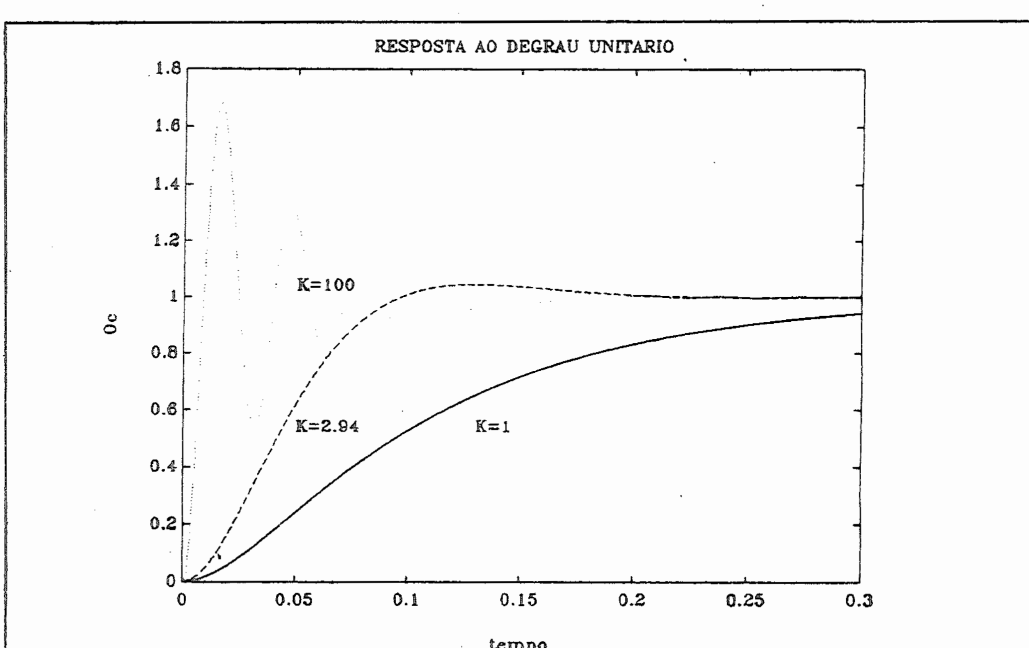
$$G(s) = \frac{\theta_c(s)}{\theta_e(s)} = \frac{400 * K}{s^2 + 48.5\ s} \quad \text{(malha aberta)}$$

a1) RESPOSTA AO DEGRAU UNITÁRIO

Para um intervalo de tempo (0,0.3) entrar com os dados no MATLAB:

```
t = 0:0.0005:0.3 ;
num1 = 400 ;
num2 = 400*2.94 ;
num3 = 400 * 100 ;
den1 = [1 48.5 num1] ;
den2 = [1 48.5 num2] ;
den3 = [1 48.5 num3] ;
z1 = step(num1,den1,t) ;
z2 = step(num2,den2,t) ;
z3 = step(num3,dem3,t) ;
plot(t,z1,t,z2,t,z3)
text(0.13,0.5,'K=1')
text(0.05,0.5,'K=2.94')
text(0.05,1.0,'K=100')
title('"RESPOSTA AO DEGRAU UNITARIO')
xlabel('tempo')
ylabel('θc')
meta g01
```

O gráfico gerado pelo MATLAB é:

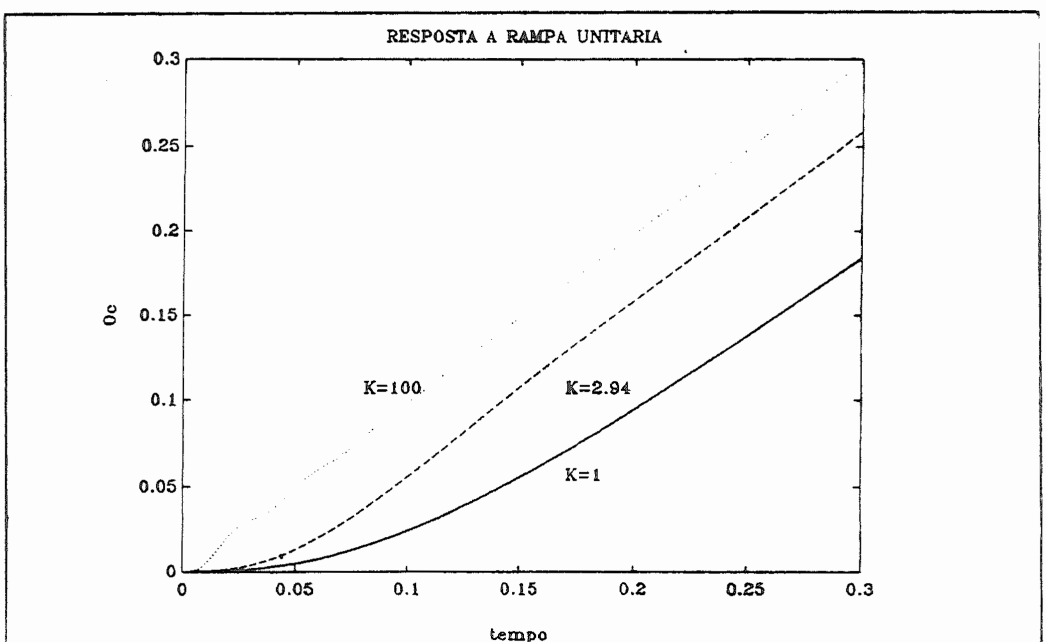


a2) RESPOSTA A RAMPA UNITARIA

Para um intervalo de tempo (0,0.3) entrar com os dados no MATLAB:

```
t = 0:0.0005:0.3 ;
num1 = 400 ;
num2 = 400*2.94 ;
num3 = 400 * 100 ;
den1 = [1 48.5 num1] ;
den2 = [1 48.5 num2] ;
den3 = [1 48.5 num3] ;
u = t ;
z1 = lsim(num1,den1,u,t) ;
z2 = lsim(num2,den2,u,t) ;
z3 = lsim(num3,den3,u,t) ;
plot(t,z1,t,z2,t,z3)
text(0.17,0.05,'K=1')
text(0.17,0.1,'K=2.94')
text(0.08,0.1,'K=100')
title('RESPOSTA A RAMPA UNITARIA')
xlabel('tempo')
ylabel('θc')
meta g02
```

O gráfico gerado pelo MATLAB é:

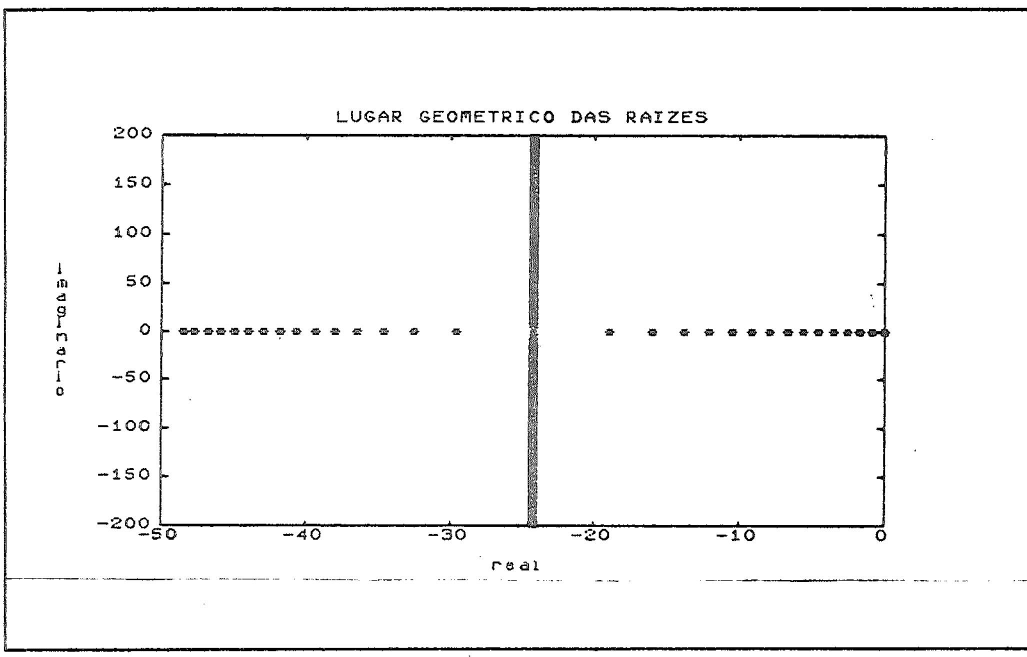


a3) LUGAR GEOMÉTRICO DAS RAÍZES

Para um ganho desejado (0,100) entrar com os dados no MATLAB:

```
K = 0:0.1:100 ;
num  = 400 ;
den  = [1 48.5 0] ;
r = rlocus(num,den,k) ;
plot(r,'*')
title('LUGAR GEOMETRICO DAS RAIZES')
xlabel('real')
ylabel('imaginario')
<shift> <prt sc>
```

O gráfico gerado pelo MATLAB é:

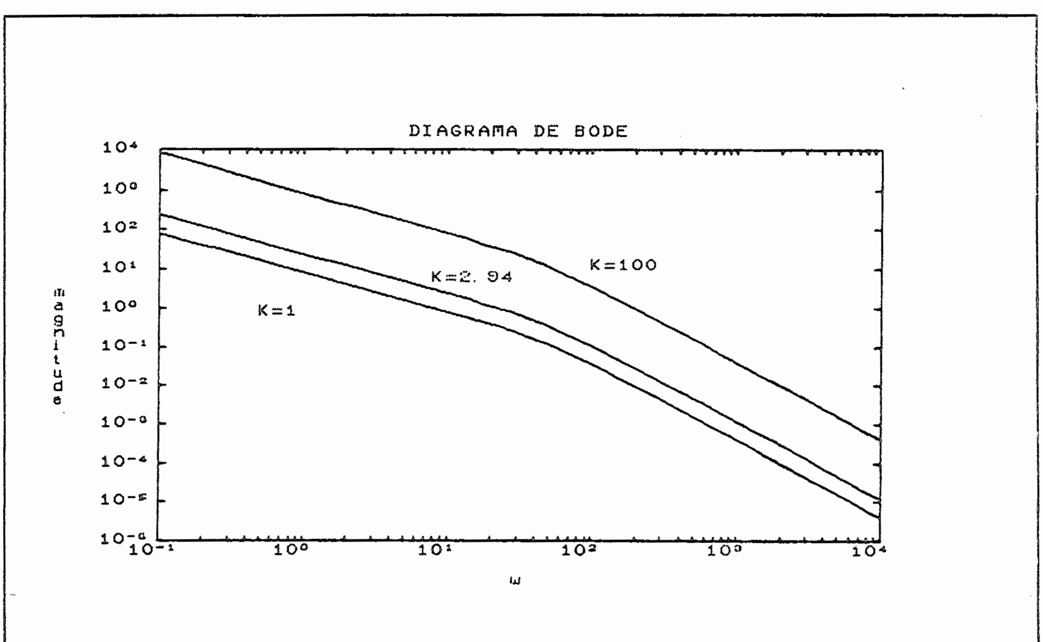


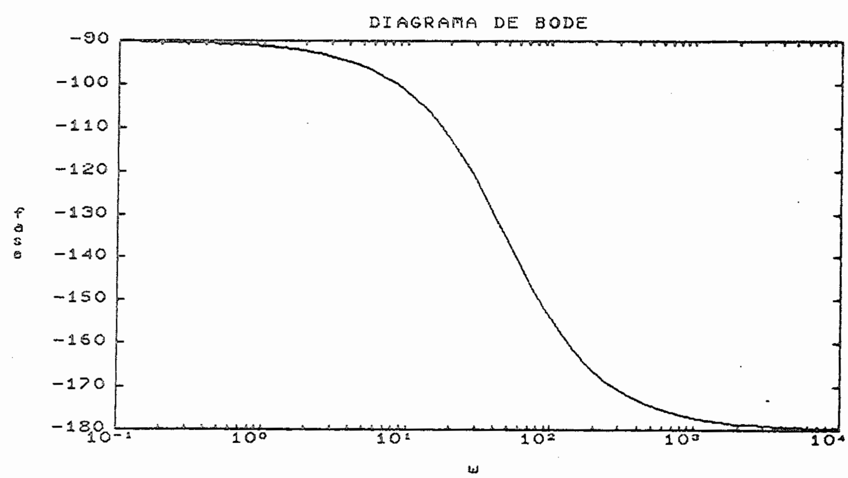
b) **TRAÇAR O DIAGRAMA DE BODE**

Para um ganho desejado $(10^{-1},10^{4})$, entrar com os dados no MATLAB:

```
w = logspace(-1,4)
num1 = 400 ;
num2 = 400*2.94 ;
num3 = 400*100 ;
den  = [1 48.5 0] ;
[mag1,fase1] = bode(num1,den,w) ;
[mag2,fase2] = bode(num2,den,w) ;
[mag3,fase3] = bode(num3,den,w) ;
loglog(w,mag1,w,mag2,w,mag3)
title('DIAGRAMA DE BODE')
xlabel('w')
ylabel('magnitude')
text(0.5,0.5,'K=1')
text(8,4,'K=2.94')
text(100,8,'K=100')
meta g04a
semilogx(mag1,fase1)
title('DIAGRAMA DE BODE')
xlabel('w')
ylabel('fase')
meta g04b
```

Os gráficos gerados pelo MATLAB são:



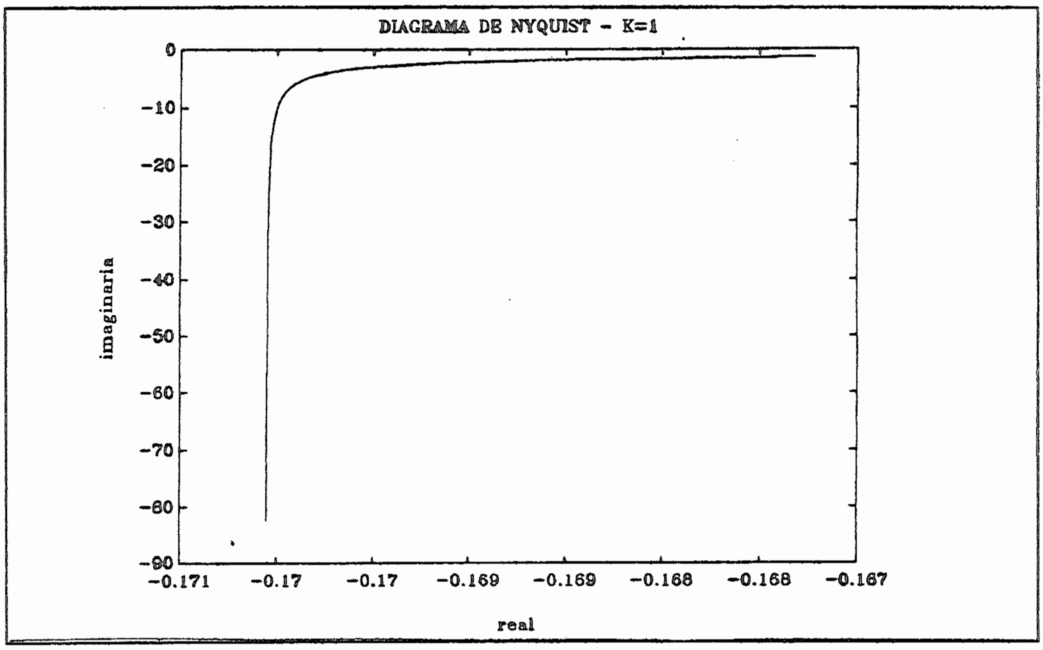


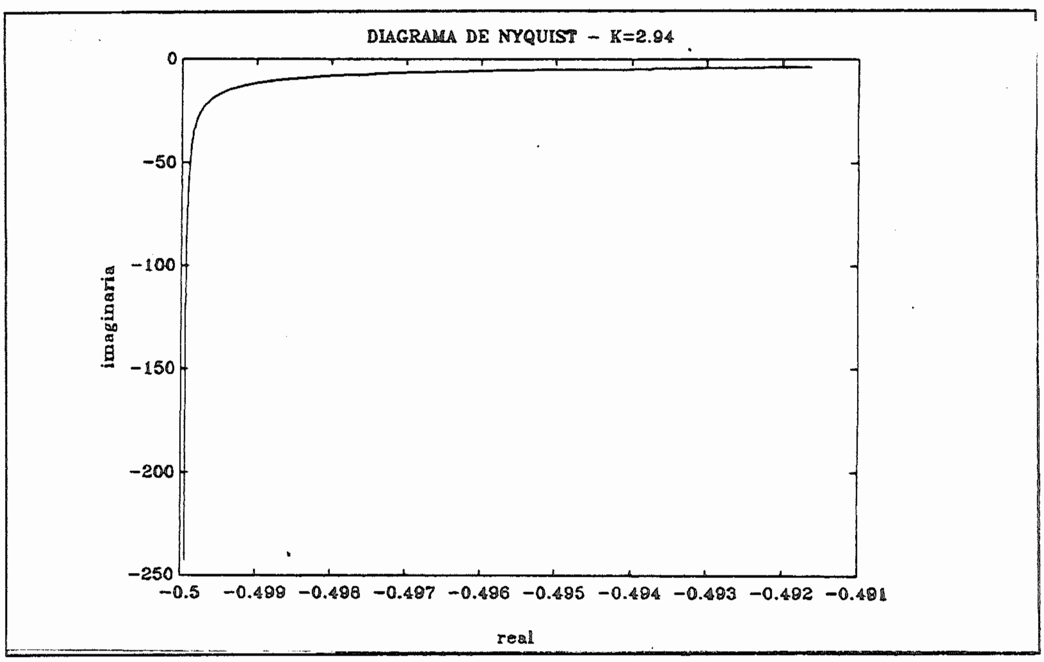
c) TRAÇAR O DIAGRAMA DE NYQUIST

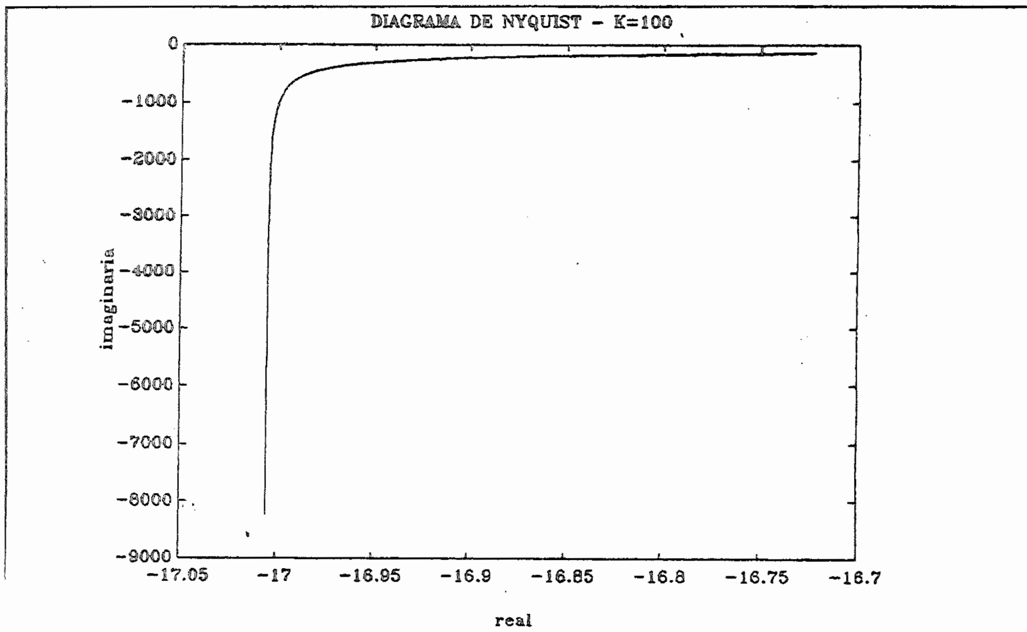
Para um ganho desejado $(10^{-1}, 10^{0.8})$, entrar com os dados no MATLAB:

```
w = logspace(-1,0.8);
num1 = 400 ;
num2 = 400*2.94 ;
num3 = 400*100 ;
den  = [1 48.5 0] ;
[re1,im1] = nyquist(num1,den,w) ;
[re2,im2] = nyquist(num2,den,w) ;
[re3,im3] = nyquist(num3,den,w) ;
plot(re1,im1)
title('DIAGRAMA DE NYQUIST - K=1')
xlabel('real')
ylabel('imaginario')
meta g05a
plot(re2,im2)
title('DIAGRAMA DE NYQUIST - K=2.94')
xlabel('real')
ylabel('imaginario')
meta g05b
plot(re3,im3)
title('DIAGRAMA DE NYQUIST - K=100')
xlabel('real')
ylabel('imaginario')
meta g05c
```

Os gráficos gerados pelo MATLAB são:







d) ADICIONANDO UM COMPENSADOR PID AS EQUAÇÕES DE TRANSFERÊNCIA SÃO:

$$\frac{\theta_c(s)}{\theta_r(s)} = \frac{400\ s^2 + 40000\ s + 2000}{s^3 + 448.5\ s^2 + 40000\ s + 2000} \quad \text{(malha fechada)}$$

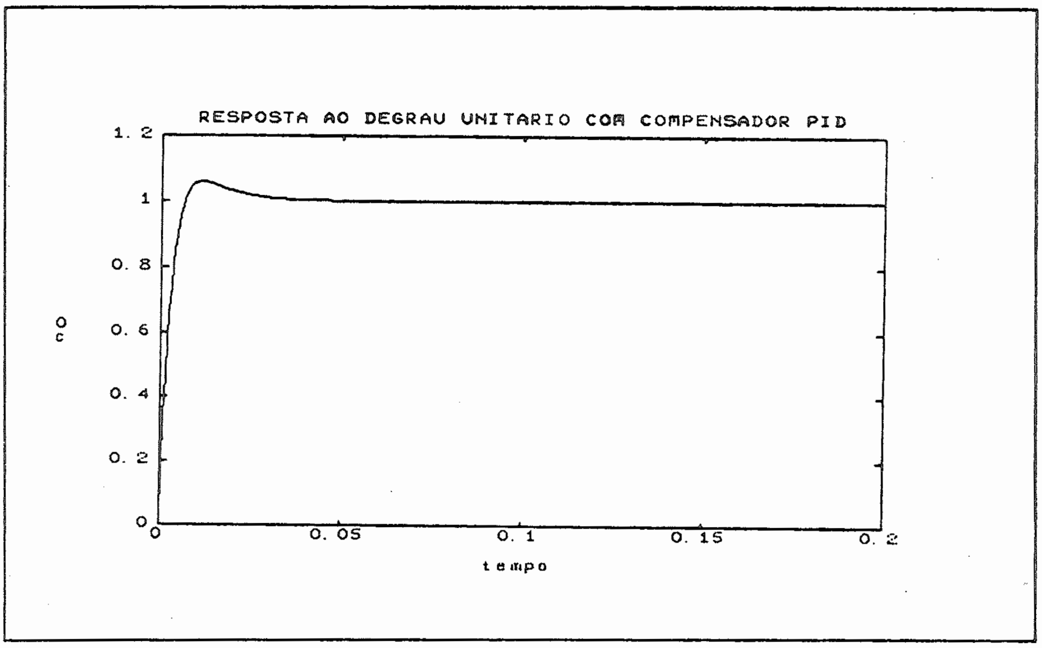
$$G(s) = \frac{\theta_c(s)}{\theta_e(s)} = \frac{400\ s^2 + 40000\ s + 2000}{s^3 + \ 48.5\ s} \quad \text{(malha aberta)}$$

d1) RESPOSTA AO DEGRAU UNITÁRIO

Para um intervalo de tempo (0,0.2) entrar com os dados no MATLAB:

```
t = 0:0.0005:0.2 ;
num  = [400 40000 2000] ;
den  = [1 448.5 40000 2000] ;
z  = step(num,den,t) ;
plot(t,z)
title('RESPOSTA AO DEGRAU UNITARIO COM COMPENSADOR PID')
xlabel('tempo')
ylabel('θc')
meta g06
```

O gráfico gerado pelo MATLAB é:

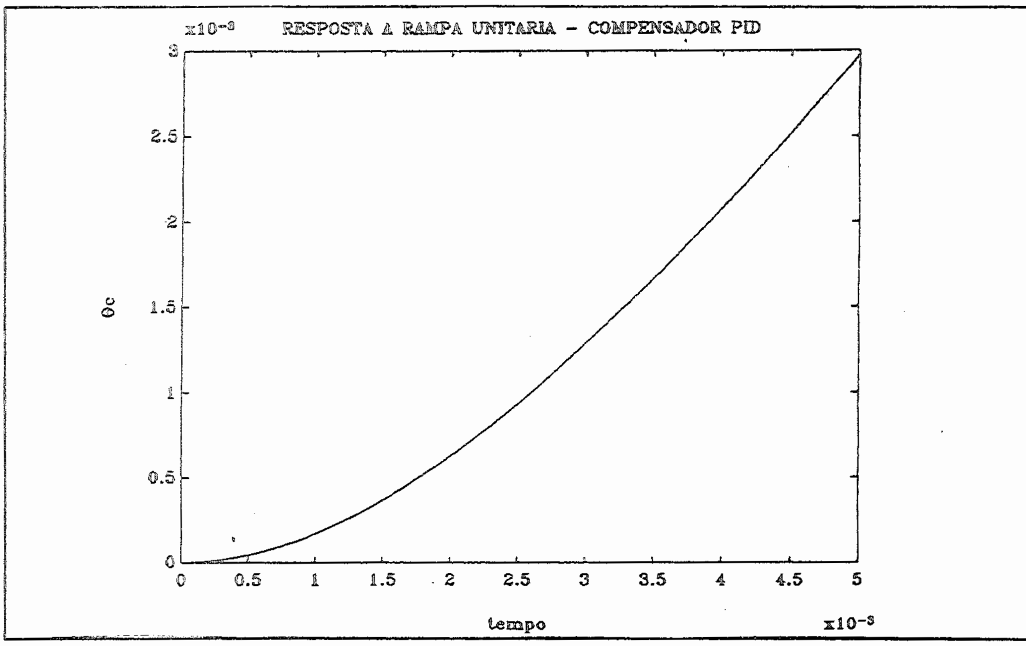


d2) **RESPOSTA A RAMPA UNITARIA**

Para um intervalo de tempo (0,0.005) entrar com os dados no MATLAB:

```
t = 0:0.00005:0.005 ;
num  = [400 40000 2000] ;
den  = [1 448.5 40000 2000] ;
u = t ;
z  = lsim(num,den,u,t) ;
plot(t,z)
title('RESPOSTA A RAMPA UNITARIA COM COMPENSADOR PID')
xlabel('tempo')
ylabel('θc')
meta g07
```

O gráfico gerado pelo MATLAB é:

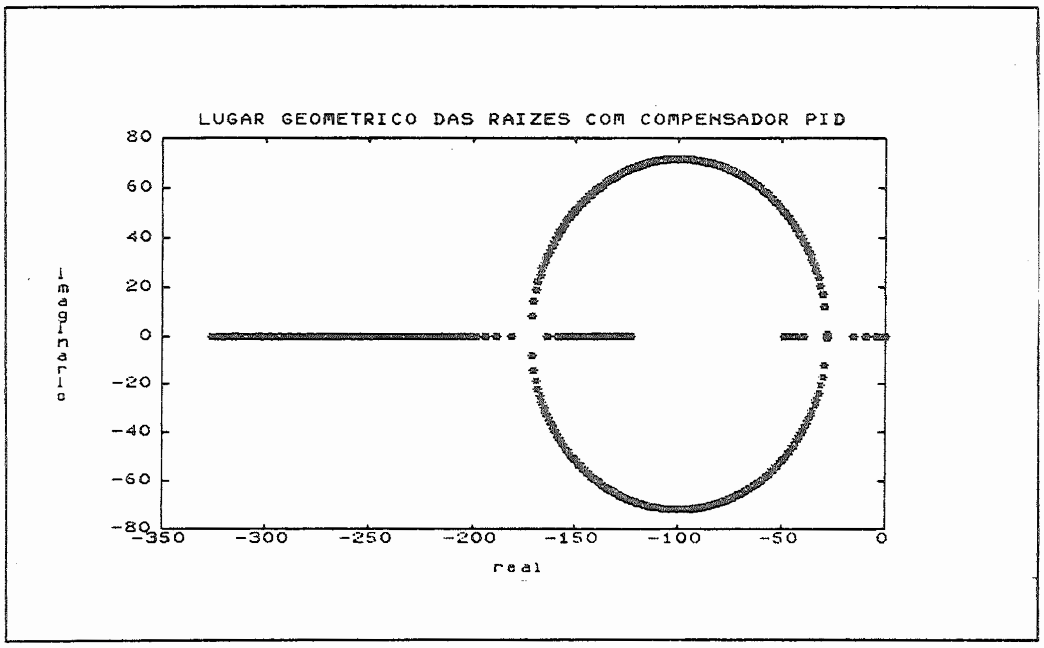


d3) LUGAR GEOMÉTRICO DAS RAÍZES

Para um ganho desejado (0,1) entrar com os dados no MATLAB:

```
k = 0:0.0005:1 ;
num  = [400 40000 2000] ;
den  = [1 48.5 0 0] ;
r  = rlocus(num,den,k) ;
plot(t,'*')
title('LUGAR GEOMÉTRICO DAS RAIZES - COMPENSADOR PID')
xlabel('real')
ylabel('imaginario')
<shift> <ptr sc>
```

O gráfico gerado pelo MATLAB é:

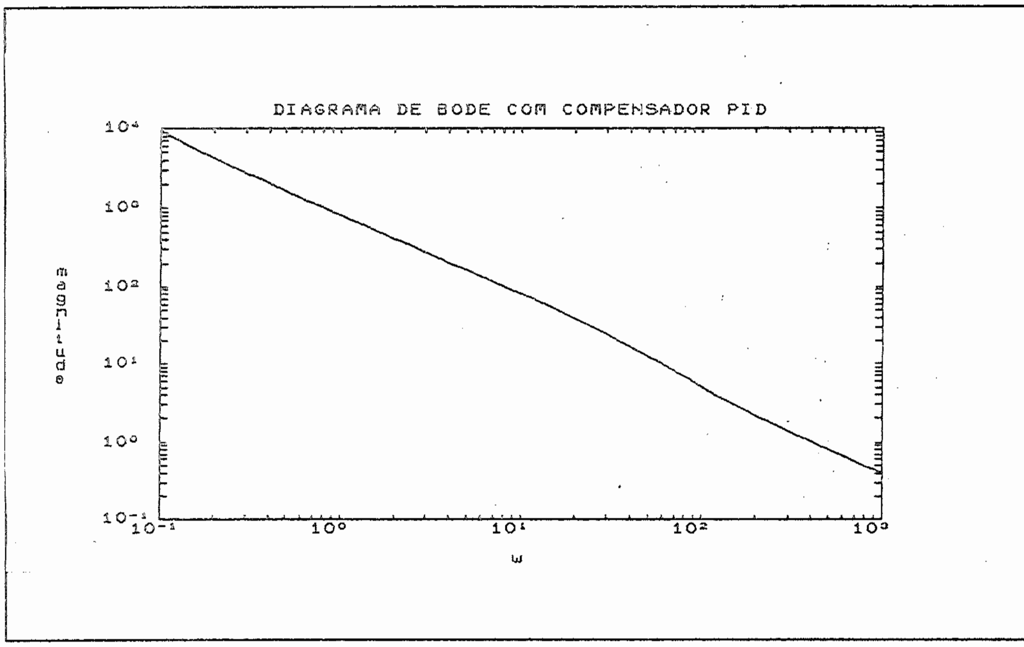


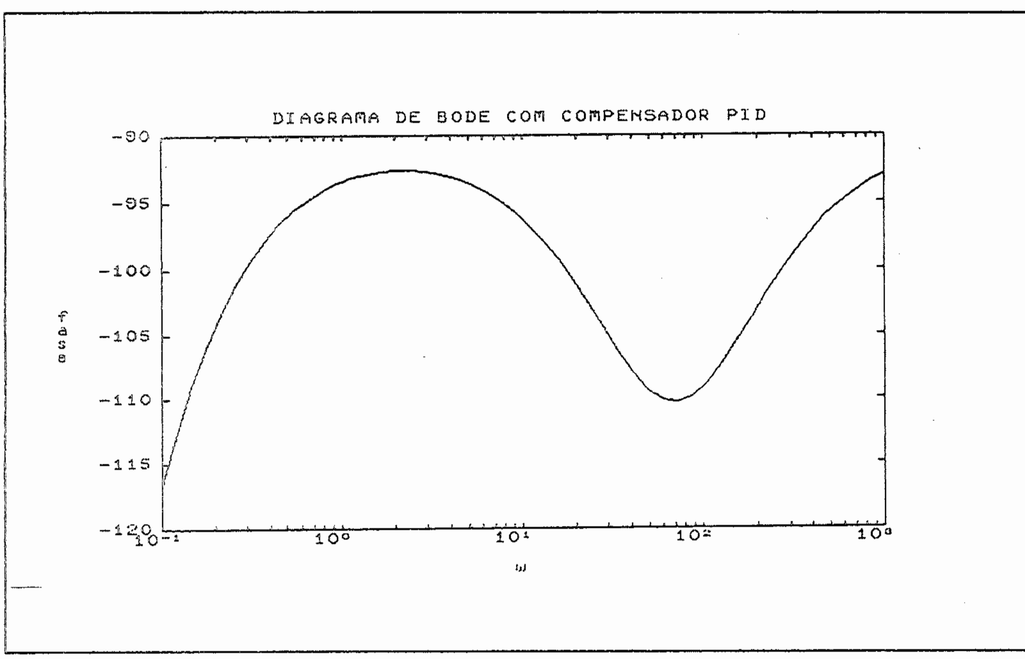
d4) **TRAÇAR O DIAGRAMA DE BODE**

Para um intervalo (10^{-1} ,10^{3}) entrar com os dados no MATLAB:

```
w = logspace(-1,3) ;
num  = [400 40000 2000] ;
den  = [1 48.5 0 0] ;
[mag,fase] = bode(num,den,w);
loglog(w,mag)
title('DIAGRAMA DE BODE COM COMPENSADOR PID')
xlabel('w')
ylabel('magnitude')
meta g09a
semilogx(w,fase)
title('DIAGRAMA DE BODE COM COMPENSADOR PID')
xlabel('w')
ylabel('fase')
meta g09b
```

O gráfico gerado pelo MATLAB é:



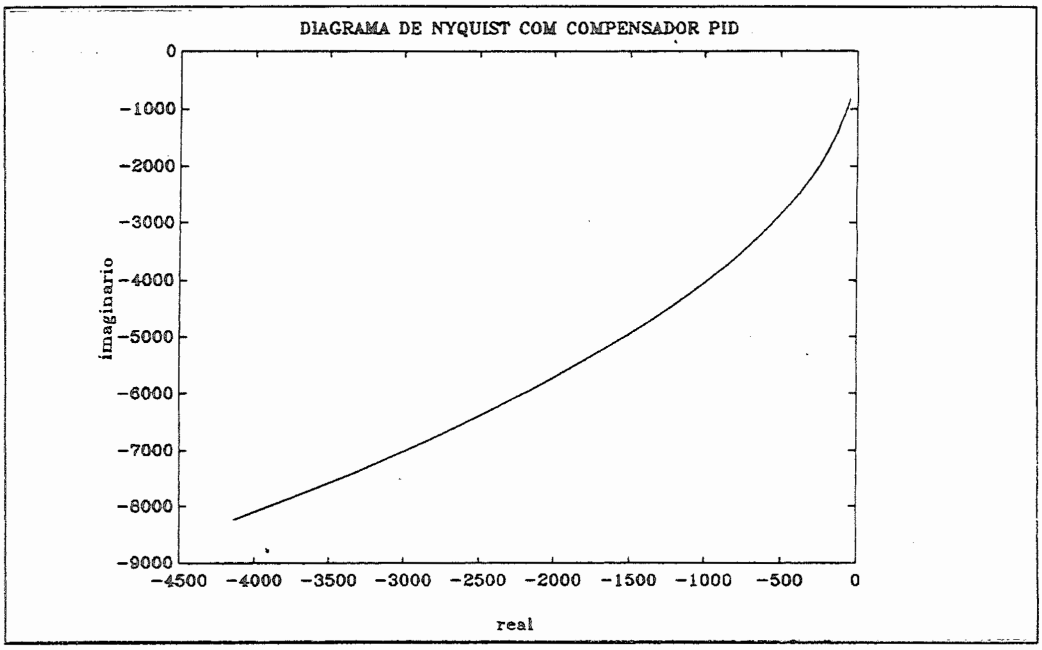


d5) TRAÇAR O DIAGRAMA DE NYQUIST

Para um intervalo $(10^{-1},10^{1})$ entrar com os dados no MATLAB:

```
w = logspace(-1,1) ;
num  = [400 40000 2000] ;
den  = [1 48.5 0 0] ;
[re,im] = nyquist(num,den,w);
plot(re,im)
title('DIAGRAMA DE NUQUIST COM COMPENSADOR PID')
xlabel('real')
ylabel('imaginario')
meta g10
```

O gráfico gerado pelo MATLAB é:



32. BIBLIOGRAFIA

MANUAL MATLAB for MS-DOS : Personal Computer. Version 3.1-PC, 1987.

CONTROL SYSTEM TOOLBOX for use with MATLAB. User's Guide. Version 2.2, 1986.

D'AZZO, J.J. & HOUPIS, C.H. Análise e projeto de sistemas de controle lineares. Traduzido por Bernardo S. Silva Filho. 2a. ed. Rio de Janeiro: Guanabara Dois, 1984.