2.17 Matlab em Sistemas de Controle

Nesta seção, os comandos básicos do CONTROL SYSTEM Toolbox do MATLAB são introduzidos. O comando *helpcontrol* fornece uma lista das diversas funções disponíveis. Para obter mais detalhes sobre cada uma das funções pode-se usar o comando help < nome da função >. Para a elaboração desta seção as seguintes referências foram utilizadas MathWorks (1997), Ogata (1997a) e Gaspar et al. (2002). No MATLAB os sistemas dinâmicos podem ser descritos por uma função de transferência ou por um modelo em espaço de estado. Na representação por função de transferência definem-se os coeficientes dos polinômios do numerador e denominador. Na representação em espaço de estado definem-se as quatro matrizes que caracterizam o modelo.

2.17.1 Função de transferência

As funções de transferência no domínio da variável complexa 's/z' são usadas para caracterizar as relações entre entrada e saída de sistemas contínuos/discretos que possam ser descritos por equações diferenciais/a diferença lineares invariantes no tempo. Para condições iniciais nulas, a partir da aplicação da transformada de Laplace às equações diferenciais lineares obtém-se a relação entre a entrada e a saída descrita por uma relação entre polinômios na variável complexa s. No MATLAB, os polinomios são descritos pelos seus coeficientes. Por exemplo, a função de transferência em 's'

$$\frac{y(s)}{r(s)} = \frac{3s^2 - 2}{s^2 + 2s + 4}$$

é introduzida pelos coeficientes dispostos em ordem decrescente das potências dos polinômios do numerador e denominador
≫ num=[3 -2];
≫ den=[1 2 4];

O comando $printsys(num, den)$ fornece a função de transferência na forma num/den=

$$\frac{3s - 2}{s\text{^}2 + 2s + 4}$$

o comando ≫ G=tf(num,den) fornece
Transfer function:

$$\frac{3s - 2}{s\text{^}2 + 2s + 4}$$

e o comando ≫ G = tf(num,den,TS) gera uma função de transferência para sistemas discretos com TS o tempo de amostragem. A saida G é um objeto do tipo TF. Evidentemente, a função pode ser chamada de outro nome qualquer.

2.17.2 Espaço de estado

Um sistema dinâmico formado por elementos concentrados pode ser representado por equações diferenciais ordinárias em que o tempo é a variável independente. Fazendo uso de notação matricial-vetorial, uma equação diferencial de ordem n pode ser representada por uma equação matricial-vetorial de primeira ordem. Se n elementos do vetor formam um conjunto de variáveis de estado, a descrição matricial vetorial denominada espaço de estado possui a forma

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Considere o seguinte sistemana forma espaço de estado

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 2 & -1 \\ 3 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 3 & -0.5 \end{bmatrix} x \end{aligned}$$

As matrizes são introduzidas no MATLABcomo segue
≫ A=[2 -1 ; 3 -1];
≫ B=[1 0]';
≫ C=[3 -0.5];
≫D=[0];
≫ printsys(A,B,C,D);

O comando *printsys* fornece o modelo espaço de estado na forma

A=

```
    x1         x2
x1 2.00000 -1.00000
x2 3.00000 -1.00000
```

B=

```
    u1
x1 1.00000
x2 0
```

C=

```
    x1         x2
y1 2.00000 -0.50000
```

D=

```
    u1
y1   0.
```

O comando *ss* fornece o modelo espaço de estado contínuo nomeado *Gss* com matrizes A, B, C, D

≫ Gss=ss(A,B,C,D)

A=

```
    x1 x2
x1 2   -1
x2 3   -1
```

B=

```
    u1
x1 1
x2 0
```

C=

```
    x1 x2
y1 2   -0.5
```

D=

```
    u1
y1   0
```

Continuous-time model.

O comando

≫ Gd = c2d(sys,Ts,method)

converte o sistema continuo *Gss* para um sistema discreto com 'Ts' o tempo de amostragem e 'method' o metodo de discretização a ser estudo no Capítulo 9.3

2.17.3 Conversão de representações

É possível realizar conversões de representação de sistemas de uma forma para outra

Função de transferência (*transfer function - tf*)
Espaço de estado (*state space - ss*)
Zeros, pólos e ganho (*zero pole gain - zp*)

Em todos os tipos de conversões lembrar que a função de transferência é dada por um polinômio no numerador e no denominador, a representação espaço de estado é dada pelas matrizes A, B, C, D e a representação zeros, pólos e ganho é dada pelo conjunto de pólos zeros e ganho.

Conversão entre espaço de estado e função de transferência

Para a conversão de espaço de estado a função de transferência e vice-versa utilizam-se as funções $ss2tf$ e $tf2ss$, respectivamente. Considere o seguinte sistema dado na forma espaço de estado

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -2 & -1 \\ 1 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned}$$

≫A = [-2 - 1; 1 - 2];
≫B = [1 2]';
≫C = [1 0];
≫D = [0];
≫[num,den] = ss2tf(A, B, C, D)
num=
0 1 0
den=
1 4 5

E, a função de transferência do sistema possui a forma

$$G(s) = \frac{s}{s^2 + 4s + 5}$$

Conversão entre pólo, zero e ganho e função de transferência

No caso do sistema estar descrito por pólos, zeros e ganho, a conversão para função de transferência é dada por

≫ z = 0; % zero do sistema
≫ p1 = -2 + i; % pólo do sistema
≫ p2 = -2 - i; % pólo do sistema
≫ k = 1; % ganho

≫[num,den]=zp2tf (z, [p1 p2], k);

obtendo-se portanto a mesma função de transferência do sistema que anteriormente.

No caso do sistema estar na forma pólo, zero e ganho, a conversão para espaço de estado é dada pelas funções $tf2ss$ e $zp2ss$

Considere a seguinte função de transferência

$$G(s) = \frac{s}{s^2 + 4s + 5}$$

As linhas de comando para obter a representação espaço de estado seguem
≫ num=[0 1 0]
≫den = [1 4 5]
≫[A, B, C, D]=tf2ss(num,den)
A sua representação em espaço de estado é como segue
A=

-4 -5
1 0

B=

1
0

C=

1 0

D=

0

Se fossem dados os *pólos, zeros e ganho* do sistema, a conversão para a representação espaço de estado seguiria os passos usados para obter a função de transferência, mas, obtendo-se as matrizes A, B, C, D e não os coeficientes do numerador e denominador da função de transferência.

Conversão para pólo, zero e ganho

A conversão para pólos, zeros e ganho pode ser realizada a partir aas seguintes funções

$tf2zp$ % obtenção pólos, zeros e ganho a partir da função de transferência
$ss2zp$ % obtenção pólos, zeros e ganho a partir da forma espaço de estado

Os pólos, zeros e ganho dos exemplos anteriores para sistemas na forma espaço de estado e na forma de função de transferência podem ser obtidos da seguinte forma

```
≫% Sistema representado no espaço de estado
≫A=[-2   - 1; 1   -2]
≫B=[1   2]'
≫C=[1   0]
≫D=[0]
≫[z, p, k]=ss2zp (A, B, C, D)

≫ % Sistema representado na forma de função de transferência
≫ num=[0  1  0]
≫ den=[1  4  5]
≫ [z, p, k]=tf2zp (num, den)

z=

0

p=

-2.0000 + 1.0000i
-2.0000 - 1.0000i

k=

1
```

2.17.4 Sistemas conectados

Existe um conjunto de funções que permite obter representações de sistemas formados por diversos subsistemas interligados. O diagrama de blocos de um sistema é uma representação das funções desempenhadas por cada um dos componentes e fluxos de sinais. O diagrama de blocos indica as conexões entre os vários componentes do sistema.

Sistemas em cascata

O comando *series* permite a associação de blocos em cascata (Figura 2.12), tendo em consideração que os dois sistemas deverão ser do mesmo tipo (contínuos ou discretos).

Aqui e doravante, considera-se que um sistema *sys* pode estar representado por

Função de transferência: sys = (num, den)
Espaço de estado: sys=(A, B, C, D)

A seguinte função implementa a associação de sistemas em cascata
≫ sys=series(sys1,sys2)

Figura 2.12: Conecção em cascata.

Considere as funções de transferência do sistema nomeado *sys*

≫ num1 = [0 0 2];
≫ den1 = [1 2 3];
≫ num2 = [0 1 1];
≫ den2 = [4 5 6];
≫ sys1=tf(num1,den1);
≫ sys2=tf(num2,den2);
≫ sys=series(sys1,sys2)

Transfer function:

$$\frac{2s + 2}{4s\text{^}4 + 13s\text{^}3 + 28s\text{^}2 + 27s + 18}$$

Para os sistemas descritos por função de transferência pode-se também utilizar o comando *conv* para conectar sistemas em cascata
≫ num = conv (num1, num2)
≫ den = conv (den1, den2)
num =
0 0 0 2 2
den =
4 13 28 27 18
Assim, utilizando
≫ sys = tf(num, den)
obtém-se a função de transferência do sistema resultante.

Considere os modelos espaço de estado de um sistema conectado *sys*

```
≫A1=[2 -1 ; 3 -1];
≫B1=[1 0]';
≫C1=[2 -0.5];
≫D1=[0];
≫sys1=ss(A1,B1,C1,D1);
≫A2 = [ -4 - 5 ; 1 0 ];
≫B2 = [ 1 0 ]';
≫C2 = [ 1 0 ];
≫D2 = [ 0 ];
≫sys2=ss(A2,B2,C2,D2);
≫sys=series(sys1,sys2)

A =

     x1 x2 x3 x4
x1 -4   -5  2 -0.5
x2 1    0   0   0
x3 0    0   2   -1
x4 0    0   3   -1

B =

     u1
x1   0
x2   0
x3   1
x4   0

C =

     x1 x2 x3 x4
y1 1    0   0   0

D =

     u1
y1   0

Continuous-time model
```

2.17.5 Sistemas em paralelo

A ligação de sistemas em paralelo, (Figura 2.13) com os dois sistemas do mesmo tipo (contínuos ou discretos) pode ser feita usando os comandos

≫ sys = parallel (sys1, sys2)

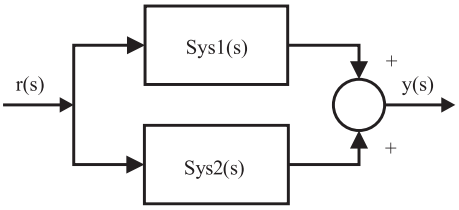

Figura 2.13: Conecção em paralelo.

O comando *parallel* é equivalente à soma direta de polinômios no caso dos sistemas estarem representados na forma de função de transferência.

2.17.6 Realimentação unitária

Um sistema de malha fechada com realimentação unitária (Figura 2.14) pode ser obtido com o seguinte comando

≫ sys = feedback(G) Considere o sistema de malha aberta $G(s)$

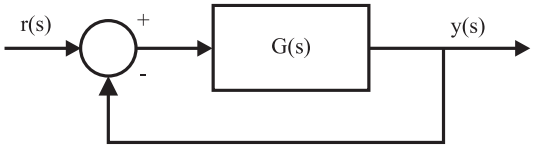

Figura 2.14: Realimentação unitária.

$$G(s) = \frac{s-6}{s^2-s+5}$$

A função de transferência de malha fechada com realimentação unitária $y(s)/r(s)$ pode ser obtida a partir das seguintes linhas de comando

≫ num = [0 1 - 6]
≫ den = [1 - 1 5]

≫ G=tf(num,den)
≫ sys=feedback(G,-1)

Transfer function:
s - 6
———
s^2 - 1
E, a função de transferência é da forma

$$sys : \frac{y(s)}{r(s)} = \frac{G(s)}{1 + G(s)} = \frac{s - 6}{s^2 - 1}$$

Observação 2.2 *Para realimentação não unitária negativa utiliza-se comando feedback (sys1,sys2,-1) e para realimentação não unitária positiva o comando feedback (sys1,sys2,1).*

2.17.7 Sistemas de 2ª ordem

Considere o sistema de 2ª ordem

$$\frac{y(s)}{r(s)} = \frac{w_n^2}{s^2 + 2\xi w_n + w_n^2}$$

A sua construção na forma de função de transferência ou espaço de estado pode ser realizada a partir da freqüência natural w_n e do coeficiente de amortecimento ξ. No caso de sistema na forma de função de transferência com uma freqüência natural de $w_n = 2.4rad/s$ e, com um coeficiente de amortecimento $\xi = 0.4$ pode-se utilizar

≫ [num, den] = ord2 (2.4, 0.4)
num =

1

den =

1.0000 1.9200 5.7600

correspondendo à função de transferência

$$\frac{y(s)}{r(s)} = \frac{1}{s^2 + 1.92s + 5.76}$$

2.17.8 Resposta no tempo

Em sistemas de controle é usual utilizar entradas de teste típicas para avaliação do desempenho do controlador. As entradas de teste usualmente utilizadas são *degrau*,

rampa e *impulso*. A escolha do tipo de entrada deve ser pautada nas características da entrada a que o sistema está submetido mais freqüentemente durante sua operação normal. A propriedade principal da dinâmica de um sistema de controle é a estabilidade. Diz-se que um sistema de controle é estável se a saída quando perturbada voltar ao seu estado de equilíbrio.

Resposta ao degrau unitário

A seguir considera-se a resposta de sistemas 2ª ordem a um degrau unitário. Seja

$$\frac{y(s)}{r(s)} = \frac{Kw_n^2}{s^2 + 2\xi w_n + w_n^2}$$

O comportamento dinâmico deste sistema de 2ª ordem pode ser descrito pela freqüência natural w_n e pelo coeficiente de amortecimento ξ. A função *step* fornece a resposta de um sistema no domínio do tempo a uma entrada em degrau unitário para condição inicial nula. Quando o comando é utilizado sem argumentos, esta função fornece graficamente a solução do sistema em resposta ao degrau unitário . A duração da simulação é determinada automaticamente baseada nos pólos e zeros do sistema

≫ step (sys)
onde
sys = tf(num,den) % função de transferência
sys = ss(A, B, C, D) % forma espaço de estado

Para obter o vetor da saída y e estado x no tempo, deve-se utilizar os argumentos no comando. O comando *step*, neste formato, não gera qualquer gráfico. O gráfico da resposta do sistema pode ser obtido com a função *plot*

≫ [y, t] = step(sys) %sistema representado por função de transferência
≫ [y, x, t] = step(sys) %sistema representado por espaço de estado
≫ plot(t, y)

O comando $[y, t] = step(sys)$ retorna a saída y e o tempo t gerado. Se sys tiver ny saídas, nu entradas e, comprimento $lt = length(t)$, y é um conjunto de tamanho $lt \times ny \times nu$ onde $y(:, :, j)$ fornece a resposta ao degrau da $j-$ésima entrada do sistema. O comando $[y, x, t] = step(sys)$ retorna o estado x de dimensão $lt \times nx \times nu$ se o sistema tiver nx variáveis de estado.

Do mesmo modo, pode-se estabelecer a duração da simulação através da imposição do tempo

≫ t = 0 : 0.1 : 10
≫ step (sys,t)

A função *step* permite a introdução de argumentos distintos para definir vários sistemas, de modo que sejam apresentados os traçados da resposta sobrepostos no mesmo gráfico, podendo os tipos, cores e marcadores das linhas de cada sistema serem definidos como nas funções básicas de criação de gráficos

≫ step (sys1,'y:',sys2,'g-')
Considere a função de transferência de malha fechada do seguinte sistema

$$\frac{y(s)}{r(s)} = \frac{25}{s^2 + 4s + 25}$$

A resposta do sistema no domínio do tempo a uma entrada em degrau unitário pode ser obtida da seguinte forma

$$y(s) = \frac{25}{s^2 + 4s + 25} r(s)$$

com $r(s) = \frac{1}{s}$. As linhas de comando para a visualização da resposta do sistema seguem (Figura 2.15)

≫ num = [0 0 25]
≫ den = [1 4 25]
≫ sys=tf(num,den)
≫ step (sys)

O sistema também pode estar na forma espaço de estado, sendo a resposta a um degrau unitário obtida com a função expressa do seguinte modo: $step(A, B, C, D)$

Resposta a entrada rampa

O MATLAB não dispõe de nenhuma função para obter a resposta de um sistema no domínio do tempo a uma entrada em rampa unitária. Para obter a resposta no tempo a este sinal de entrada particular, utiliza-se a função *step* aumentando um grau os coeficientes das sucessivas potências do polinômio que traduzem o *denominador* da função de transferência do sistema. O uso do comando segue o caso anterior. Considerando a mesma função de transferência anterior, a resposta do sistema no domínio do tempo a uma entrada em rampa unitária é obtida como segue

$$y(s) \quad = \quad \frac{25}{s^2 + 4s + 25} r(s)$$

com $r(s) = \frac{1}{s^2}$. As funções para obter os sistemas e a resposta à ramapa seguem (Figura 2.16)

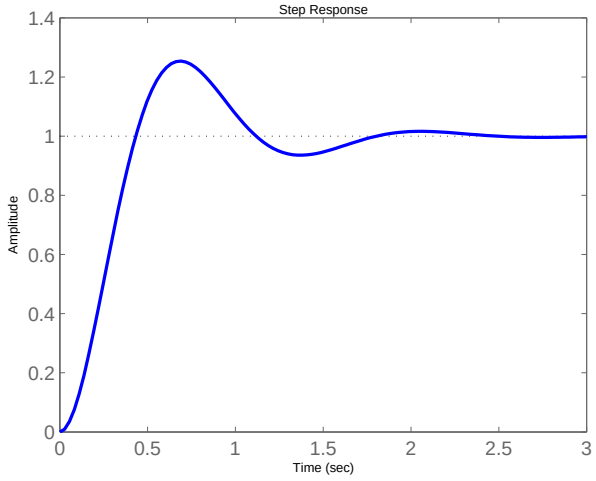

Figura 2.15: Resposta do sistema num = [0 0 25], den = [1 4 25] à uma entrada degrau unitário.

≫ num1=[0 0 25];
≫den1=[1 4 25];
≫num2=[0 0 1];
≫den2=[0 1 0];
≫sys1=tf(num1,den1); %função de transferência $25/(s^2+4s+25)$
≫sys2=tf(num2,den2); %função de transferência $1/s$
≫sys=series(sys1,sys2);
≫step(sys,'r',sys2,'k')
O erro estacionário para a resposta no domínio do tempo de um sistema de 2ª ordem sujeito a uma entrada em rampa unitária é dado por

$$e_{ss} = \frac{2\xi}{w_n}$$

Resposta ao impulso

A função *impulse* fornece a resposta de um sistema no domínio do tempo a uma entrada em impulso . Considere o sistema anterior sujeito a uma entrada impulso. A saída $y(s)$ é da forma

$$y(s) = \frac{25}{s^2+4s+25} r(s) \text{ com } r(s) = 1$$

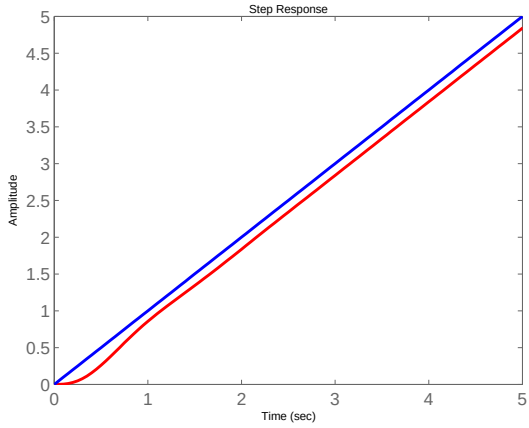

Figura 2.16: Resposta sistema $G(s) = 25/(s^2 + 4s + 25)$ a uma entrada rampa unitária.

As linhas de comando para a visualizar a resposta do sistema seguem (Figura 2.17)

$\gg$ num = [0 0 25];
$\gg$ den = [1 4 25];
$\gg$ sys=tf(num,den);
$\gg$ impulse(sys)

Como a resposta a uma entrada impulso corresponde à derivada da resposta a uma entrada em degrau unitário, o sobre-sinal máximo para a resposta a degrau unitário pode ser determinado a partir da correspondente resposta ao impulso, já que a área sob a curva de resposta ao impulso de Dirac desde $t = 0$ até t_p (tempo do primeiro cruzamento com zero) é dada por

$$1 + M_p$$

onde M_p corresponde ao sobre-sinal máximo para a resposta a degrau unitário.

Resposta a entradas e a condições iniciais

A função *initial* gera a resposta no domínio do tempo de um sistema no espaço de estado com condições iniciais não nulas. Considere um sistema representado no espaço de estado com condições iniciais não nulas

$$\begin{aligned} \dot{x} &= Ax + Bu, x(0) = x_0 \\ y &= Cx \end{aligned}$$

As linhas de comando que possibilitam a obtenção da resposta do sistema a este tipo

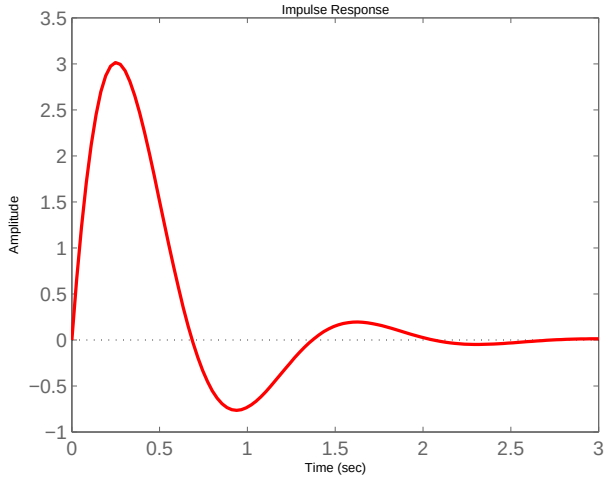

Figura 2.17: Resposta do sistema num = [0 0 25], den = [1 4 25] a uma entrada impulso.

particular de entrada segue

≫ initial(sys, x0)

Considere o seguinte sistema representado no espaço de estado

$$\dot{x} = \begin{bmatrix} -0.5572 & -0.7814 \\ 0.7814 & 0 \end{bmatrix} x$$
$$y = \begin{bmatrix} 1.969 & 6.4493 \end{bmatrix} x$$

e as condições iniciais

$$x(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

As linhas de comando para obter a resposta a x_0 seguem (Figura 2.18)

≫ A = [-0.5572 -0.7814; 0.7814 0];
≫ B = [0 0]';
≫ C = [1.9691 6.4493];
≫ D = [0];
≫ x0 = [1 0]';
≫ initial (A, B, C, D, x0)

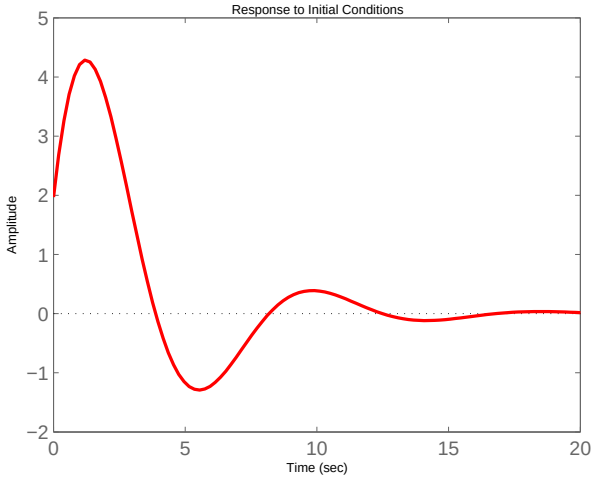

Figura 2.18: Resposta do sistema [A,B,C,D] à condição inicial x_0.

2.17.9 Resposta a uma entrada arbitrária

A função $lsim$ (linear simulation) simula a resposta de um sistema no domínio do tempo a uma entrada arbitrária e possui a mesma estrutura das funções anteriores
$\gg$ lsim (sys, u, t)

A matriz formada pela entrada u possui tantas colunas quanto a dimensão do vetor u no tempo $t(length(t))$. A função $lsim$ pode também gerar a resposta a entrada arbitrária de sistemas representados no espaço de estado com condições iniciais não nulas

$\gg lsim(sys, u, t, x0)$

O comando que gera o sinal arbitrário u é o comando *gensig* (generate signal). Este comando gera um sinal periódico escalar u da classe *type* e período τ, para simulações no domínio do tempo através do uso da função $lsim$. A função *gensig* suporta as seguintes classes de sinal
type = 'sin' : onda senoidal
type = 'square' : onda quadrada
type = 'pulse' : impulso periódico

$\gg [u, t] = gensig(type, \tau)$

Considere a seguinte função de transferência

$$\frac{y(s)}{r(s)} = \frac{s-1}{s^2+s+5}$$

Pode-se obter a resposta a uma entrada do tipo onda quadrada com um período de 4 segundos da seguinte forma. Inicialmente, gera-se a onda quadrada com a função *gensig* considerando um tempo de amostragem de 0.1s durante 10s e posteriormente, simula-se a resposta do sistema (Figura 2.19)

```
≫ [u,t] = gensig ('square',4,10,0.1 );
≫ num = [ 1 - 1 ];
≫ den = [ 1 1 2 ];
≫ sys=tf(num,den);
≫ lsim(sys,'r',u,t)
≫ axis([0 10 -1.5 1.5])
≫ hold
Current plot held
≫ plot(t,u,'k')
≫ axis([0 10 -1.5 1.5])
```

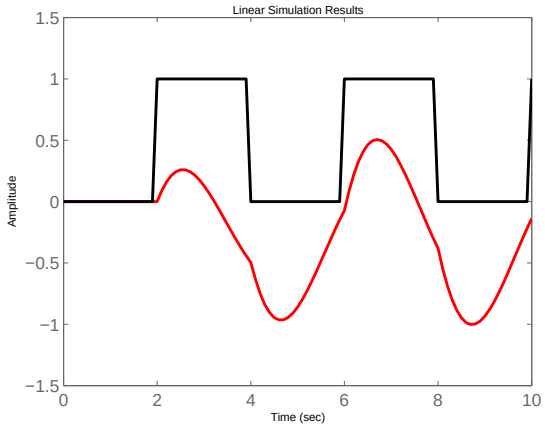

Figura 2.19: Resposta do sistema a uma entrada quadrada.

2.17.10 Lugar geométrico das raízes

A característica básica da resposta transitória de um sistema em malha fechada está intimamente relacionada com a localização dos pólos de malha fechada. Os pólos de

malha fechada são as raízes da equação característica $\delta(s) = 1 + G(s)H(s)$. O lugar geométrico das raízes é formado pelas raízes da equação característica em função de um parâmetro. O parâmetro usualmente utilizado é o ganho da função de transferência de malha aberta.

Pólos e zeros

A função *pzmap* sem argumento fornece um diagrama dos pólos e dos zeros de um sistema contínuo ou discreto. Os pólos são representados por x e os zeros representados por o. Com argumento, *pzmap* fornece duas colunas correspondentes aos pólos e aos zeros sem gerar qualquer diagrama

≫ pzmap(sys) %geração do diagrama de pólos e zeros do sistema
≫ [p, z] = pzmap(sys) %obtenção dos valores dos pólos e dos zeros

Considere a função de transferência de malha aberta do sistema

$$G(s) = \frac{2s^2 + 5s + 1}{s^2 + 2s + 3}$$

Pede-se os obter os pólos e os zeros com as linhas de comando como seguem (Figura 2.20)
≫num = [2 5 1];
≫den = [1 2 3];
≫sys=tf(num,den);
≫pzmap (sys,'r')

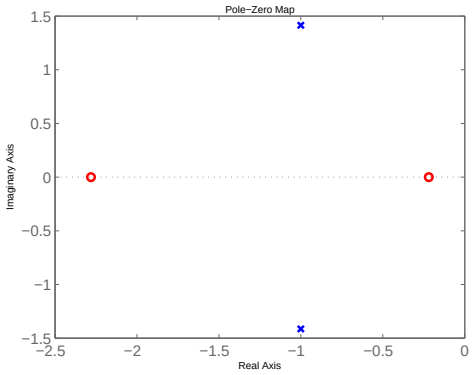

Figura 2.20: Mapa de pólos e zeros.

Usando rlocus

A função *rlocus* fornece o lugar geométrico das raízes do polinômio característico de um sistema que correspondem aos pólos do sistema realimentado em função da variação do ganho K desde zero até infinito. O lugar geométrico das raízes indica o lugar dos pólos de malha fechada do sistema em função do ganho de realimentação K para realimentação negativa. O lugar geométrico das raízes é usado no estudo dos efeitos da variação do ganho de realimentação na localização dos pólos de malha fechada. As diferentes funções de transferência de malha aberta de sistemas para implementação da função *rlocus*, são mostradas na Figura 2.21

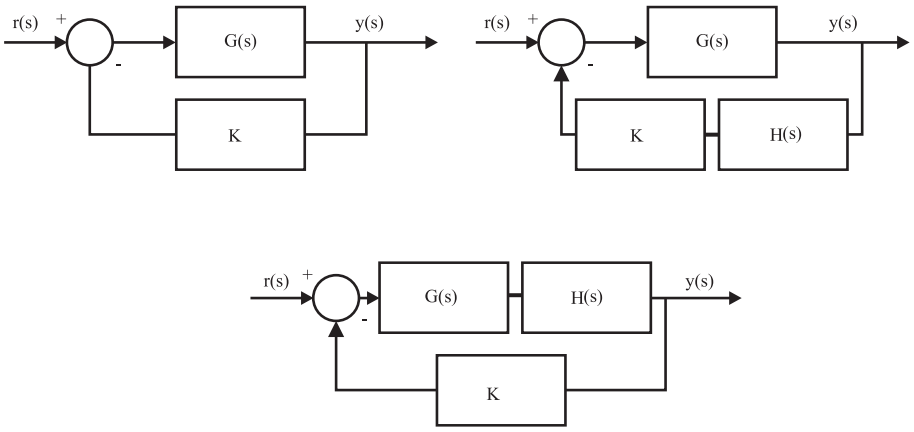

Figura 2.21: Diferentes configurações de realimentação.

Seja $G(s)$ dada por

$$G(s) = \frac{n(s)}{d(s)}.$$

Os pólos do sistema realimentado são dados pelas raízes de

$$d(s) + Kn(s) = 0$$

Quando o comando *rlocus* é utilizado sem argumentos obtém-se o lugar dos pólos em função do ganho K. Em caso contrário, apresenta duas colunas correspondentes à localização das raízes complexas r e, o respectivo ganho K, sem gerar qualquer gráfico

$\gg$ rlocus (sys) %geração do lugar das raízes
$\gg$rlocus (sys,K) %mapa das raízes para um determinado ganho K
$\gg$[r,K] = rlocus(sys) %geração dos vetores das raízes e respectivo ganho
$\gg$r = rlocus (sys,K) %raízes para um ganho fixo K

Para visualizar o lugar das raízes (Figura 2.22) utilizam-se os seguintes comandos

≫ num=conv([1 10],[1 10]);
≫ den=conv([1 -20],[1 4 68]);
≫ sys=tf(num,den)
≫ rlocus(sys)

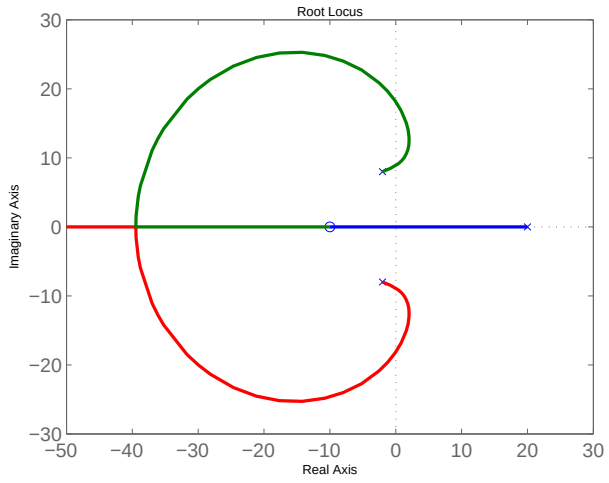

Figura 2.22: Lugar das raízes de do polinômio característico $\delta(s) = d(s) + Kn(s)$ em função de K para num=conv([1 10],[1 10])e den=conv([1 -20],[1 4 68].

Ganho estático

A função *rlocfind* utiliza a regra da magnitude do lugar geométrico das raízes para determinar o *ganho* para uma localização particular das raízes. Trata-se de uma função interativa, já que permite ao usuário selecionar a localização das raízes no traçado do lugar de raízes para as quais pretende determinar o ganho de realimentação. No entanto, pode ser utilizada com argumentos de modo a calcular o ganho de realimentação para uma localização especifica das raízes.

≫ [K,poles]= rlocfind (sys) %obtenção do ganho K interativamente
≫[K,poles] = rlocfind (sys,p) %obtenção do ganho K para uma localização especifica dos pólos.

Curvas de w_n e ξ constantes

A função *sgrid* permite gerar um gráfico no plano s de curvas de freqüência natural constante e de coeficiente de amortecimento constante. O gráfico é gerado sobre o traçado do lugar geométrico das raízes ou sobre o mapa dos pólos e zeros obtido anteriormente, com um espaçamento de 0.1 desde 0 até 1 para as curvas de coeficiente de amortecimento ξ constante, e com um espaçamento de 1 rad/s desde 0 até 90 rad/s para as curvas de freqüência natural w_n constante. Pode-se especificar como argumento as linhas de coeficiente de amortecimento constante ξ e as linhas freqüência natural constante w_n $\gg$ sgrid
$\gg$ sgrid(xi,wn)
Os gráficos mostrados nas Figuras 2.23 e 2.24 ilustram o uso do comando *sgrid*. Considere a função de transferência de malha aberta do sistema. Para gerar o mapa de pólos e zeros sobre o lugar dos pólos, as linhas no plano s de coeficiente de amortecimento constante e de freqüência natural constante utiliza-se a função *sgrid*. Seja

$$G(s) = \frac{2s^2 + 5s + 1}{s^2 + 2s + 3}$$

As linhas de comando para gerar o mapa de pólos e zeros e o lugar das raízes seguem
$\gg$ num = [2 5 1];
$\gg$ den = [1 2 3];
$\gg$ sys=tf(num,den);
$\gg$ pzmap (sys)
$\gg$ sgrid
$\gg$ num=conv([1 10],[1 10]);
$\gg$ den=conv([1 -20],[1 4 68]);
$\gg$ sys=tf(num,den)
$\gg$ rlocus(sys)
$\gg$ sgrid

2.17.11 Resposta em frequência

Pelo termo resposta em freqüência entende-se a resposta em regime estacionário de um sistema sujeito a uma entrada exponencial complexa. Existe um conjunto de funções no Matlab que permite analisar a resposta no domínio da freqüência de um dado sistema. Os diagramas de Bode, Nyquist e Nichols são usualmente utilizados seja para o sistema descrito por uma função de transferência seja pelo modelo espaço de estado.

Diagrama de Bode

A função de transferência para $s = jw$ pode ser caracterizada pelo seu módulo e ângulo de fase em função da freqüência em escala logarítmica. A função *bode* fornece o diagrama de Bode da resposta em freqüência de um dado sistema pela avaliação da função de

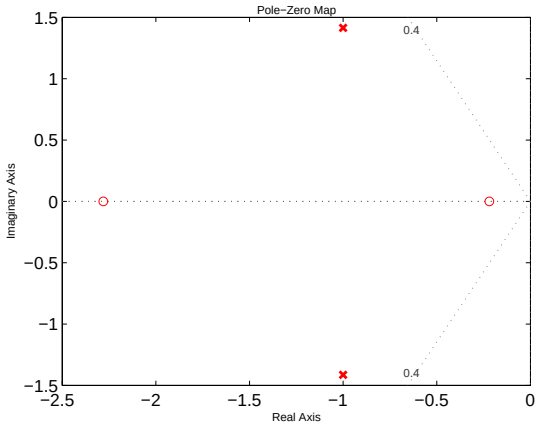

Figura 2.23: Pólos e zeros no plano s com $\times$ indicando pólo, o zero e curva de $w_n = 5$ e $\xi = 0.4$.

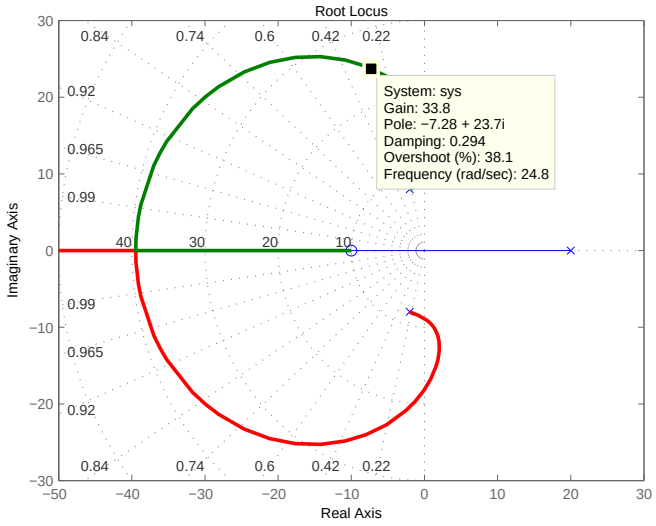

Figura 2.24: Lugar das ráizes com o comandos rlocus e sgrid.

transferência $G(s)$ no eixo imaginário $s = jw$. Apenas são consideradas freqüências positivas. Para sistemas na forma espaço de estado, a resposta em freqüência é dada por

$$G(jw) = D + C(jwI - A)^{-1}B,\ w \geq 0$$

Quando o comando é utilizado sem argumentos, esta função gera o módulo e do ângulo de fase em função da freqüência. O diagrama de Bode são utilizados para analisar propriedades do sistema, tais como margem de ganho, margem de fase, largura de banda, estabilidade, etc. A faixa de freqüências, em rad/s, é determinada automaticamente baseada nos pólos e zeros do sistema. As diversas sintaxes desta função seguem

≫ bode (sys) %geração diagrama de Bode de módulo e fase
≫ bode (sys,w) %geração diagrama de Bode com faixa de freqüência especificada
≫ bode (sys1, sys2,..., sysN) %geração do diagrama de Bode diversos sistemas sobrepostos

Quando o comando *bode* é usado com os seus argumentos, obtém-se os vetores do módulo, ângulo de fase e a freqüência w e, o gráfico não é exibido.
≫ [mod,fase,w] = bode(sys) %obtenção dos valores do módulo, fase e freqüência
≫ [mod,fase] = bode(sys,w) %obtenção dos valores do módulo e fase para a faixa de freqüência especificada

Considere a seguinte função de transferência em malha aberta

$$G(s) = \frac{s^2 + 0.1s + 7.5}{s^4 + 0.12s^3 + s^2}.$$

O diagrama de Bode pode ser obtido com as seguintes linhas de comando (Figura 2.25)
≫ num = [1 0.1 7.5];
≫ den = [1 0.1 2 9 0 0];
≫ sys=tf(num,den)
≫ bode (sys, 'r')

Para obter o gráfico da resposta do sistema no domínio da freqüência para uma faixa maior de freqüência, por exemplo de 0.1 até 100 rad/s, basta introduzir o comando (Figura 2.26)

≫ w = logspace (-1, 2);
≫num = [1 0.1 7.5];
≫den = [1 0.1 2 9 0 0];
≫ sys=tf(num,den)
≫ bode (sys,'r', w)

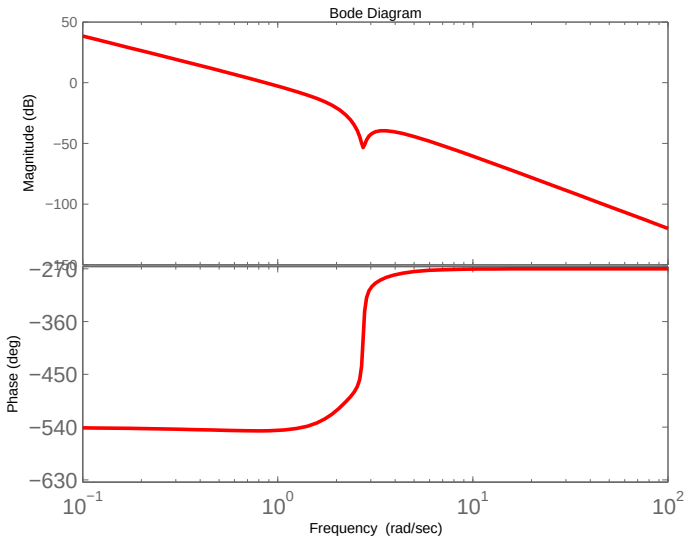

Figura 2.25: Resposta em freqüencia.

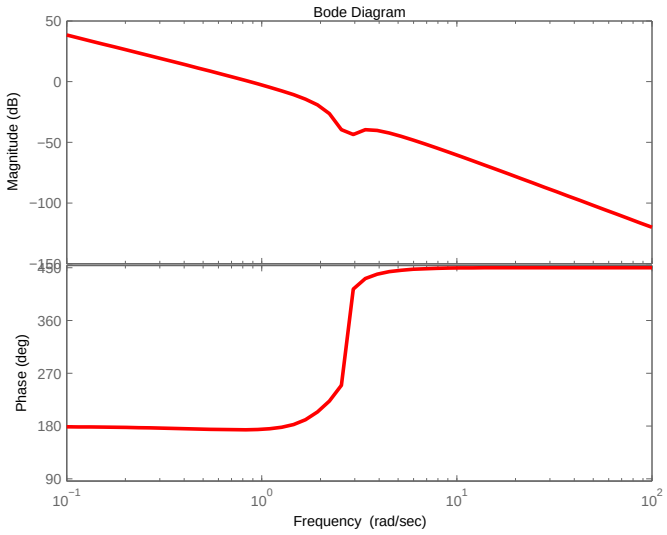

Figura 2.26: Resposta em freqüência selecionando a faixa.

Margem de ganho e margem de fase

Na análise da resposta de sistemas no domínio da freqüência, as margens de ganho e de fase permitem avaliar a estabilidade relativa do sistema.

A margem de fase é definida como o atraso de fase adicional na freqüência de cruzamento do ganho (w_{cg}) necessário para levar o sistema ao limiar da instabilidade. A freqüência de cruzamento do ganho é a freqüência na qual $|G(jw)|$, o módulo da função de transferência de malha aberta, é unitário ou 0 dB no caso do módulo da função de transferência de malha aberta ser representado em dB. A margem de fase (MF) pode ser calculada da seguinte forma

$$MF = 180^o + \angle G(w_{cg})$$
$$MF = 180^o + \phi$$

Portanto, a MF é 180º mais o ângulo ϕ da função transferência de malha aberta na freqüência de cruzamento de ganho denotado (w_{cg}). Para um sistema de fase mínima (todos os pólos e zeros no semiplano s da esquerda) ser estável, a margem de fase deve ser positiva.

A margem de ganho é definida como o inverso do módulo $|G(jw)|$ na freqüência de cruzamento da fase (w_ϕ). A freqüência de cruzamento da fase é a freqüência na qual $\angle G(jw_{cg})$, o ângulo de fase da função de transferência de malha aberta, é igual a 180o. A margem de ganho (MG) pode ser obtida da seguinte forma

$$MG = -20log\,|G(w_\phi)| \quad dB$$

Uma margem de ganho positiva significa que o sistema é estável, e o contrário que o sistema é instável. Para um sistema de fase mínima estável, a margem de ganho indica quanto o ganho pode ser aumentado antes que o sistema se torne instável. Para um sistema instável, a margem de ganho é indicativa de quanto o ganho deve ser diminuído para tornar o sistema estável.

A margem de ganho e a margem de fase e respectivas freqüências de cruzamento, são calculadas a partir da função de transferência em malha aberta pela função *margin*. Indicam a estabilidade relativa do sistema em malha fechada. Quando o comando é usado sem argumentos, exibe os diagramas de Bode de módulo e ângulo de fase com as referidas margens. As sintaxes possíveis para esta função seguem

≫ margin(sys) %diagrama de Bode com as margens
≫[Gm, Pm, Wcg, Wcp] = margin(sys) % margens e freqüências de cruzamento
≫ margin(mag,fase,w) % diagrama de Bode com as margens
≫[Gm, Pm, Wcg, Wcp]=margin (mag,fase,w) % margens e freqüências de cruzamento
≫ allmargin(sys) % todas as margens de ganho e fase

Considere a seguinte função de transferência em malha aberta

$$G(s)H(s) = \frac{K}{s(s+1)(s+5)}$$

Para $K = 10$ e $K = 100$ as margens de ganho e fase podem ser obtidas como segue (Figura 2.27)
≫ % Para K=10
≫ num=[10];
≫ den1=conv([1 1],[1 5]);
≫den=conv([1 0],den1);
≫sys=tf(num,den)
≫ margin(sys)

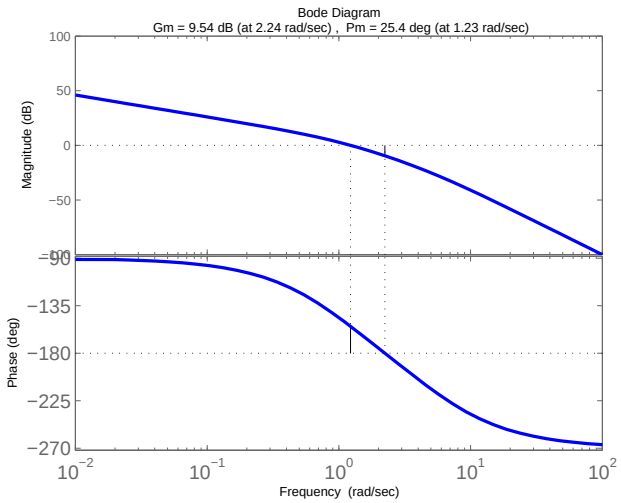

Figura 2.27: Margens de estabilidade para K=10.

≫% Para K=100
≫ num=[100];
≫den1=conv([1 1],[1 5]);
≫den=conv([1 0],den1);
≫sys=tf(num,den)
≫margin(sys)

Critério de Nyquist

O critério de estabilidade de Nyquist permite investigar a estabilidade absoluta e a estabilidade relativa de sistemas lineares de malha fechada a partir da resposta em freqüência do respectivo sistema em malha aberta. Considere o seguinte sistema representado pela

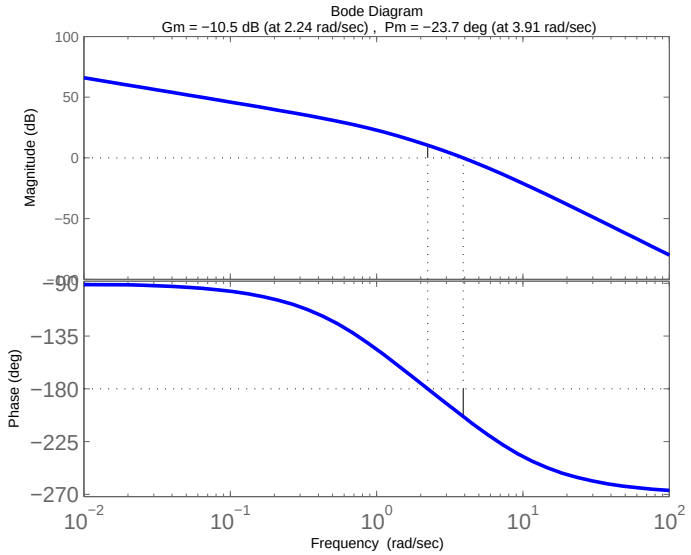

Figura 2.28: Margens de estabilidade para K=100.

sua função de transferência de malha fechada

$$\frac{y(s)}{r(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Para que este sistema seja estável, todas as raízes da equação característica $\delta(s) = 1 + G(s)H(s) = 0$ devem permanecer no semiplano lateral esquerdo s. O critério de Nyquist estabelece que Z=N+P com Z o número de pólos instáveis do sistema em malha fechada N o número de envolvimentos do gráfico de Nyquist (plano da função $G(s)H(s)$) no sentido horário em torno do ponto $-1 + j0$ e P o número de pólos de $G(s)H(s)$ no semiplano lateral direito. Se o diagrama de Nyquist envolver o ponto $1 + j0$ uma vez no sentido horário, N=1, se no sentido anti-horário N=-1. Do critério pode-se concluir o seguinte

1. Se $P \neq 0$, para estabilidade deve-se ter $Z = 0$ o que implica $N = -P$, ou seja, o número de envolvimentos do gráfico de Nyquist, no sentido anti-horário em torno do ponto $-1 + j0$ deve ser igual ao número de pólos de $G(s)H(s)$ no semiplano lateral direito do plano s

2. Se $P = 0$, para estabilidade deve-se ter $Z = N = 0$, ou seja, não pode haver envolvimento do ponto $-1 + j0$

A função *nyquist* fornece o gráfico de $G(jw)H(jw)$ em função da frequência w. O diagrama de Nyquist permite estudar também a estabilidade relativa do sistema em malha fechada, ou seja, a margem de ganho e margem de fase. Quando o comando *nyquist* é usado com argumentos obtém-se as partes real e imaginárias de $G(jw)H(jw)$ em função de w. As sintaxes possíveis para este comado seguem
≫ nyquist (num, den) %diagrama de Nyquist
≫nyquist (num, den, w) %diagrama de Nyquist para faixa de freqüências especificada
≫[re, im, w] = nyquist (num, den) %obtenção partes real e imaginária de $G(jw)H(jw)$
Considere a seguinte função de transferência em malha aberta

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

Para obter o diagrama de Nyquist utilizar os seguintes comandos (Figura 2.29)
≫ num = [2 5 1];
≫den = [1 2 3];
≫sys=tf(num,den)
≫ nyquist (sys,'r')

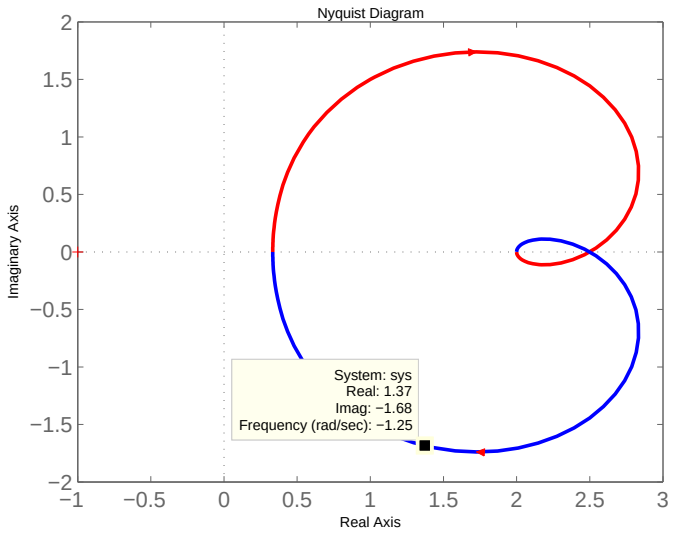

Figura 2.29: Diagrama de Nyquist para w de $-\infty$ a ∞.

2.17.12 Diagrama de Nichols

A função *Nichols* fornece a resposta de um sistema no domínio da freqüência e exibe o diagrama nas coordenadas de Nichols . Esta função é útil para analisar as propriedades de sistema em malha aberta e em malha fechada. Quando o comando é usado sem argumentos produz a exibição do diagrama de Nichols. A função *ngrid* gera as linhas de grade para o traçado de Nichols que correspondem às linhas de módulo e ângulo de fase constantes da função de transferência de malha aberta. A grade e a faixa de freqüências é determinada automaticamente a partir dos pólos e zeros do sistema
≫ nichols (sys) %diagrama de Nichols
≫ nichols (sys,w) %diagrama de Nichols com faixa de freqüência especificada
≫ nichols (sys1, sys2,..., sysN) %diagramas de Nichols sobrepostos de diversos sistemas
Quando o comando é usado com os argumentos do módulo, os vetores da resposta são exibidos em lugar do diagrama de Nichols
≫[mod,fase,w]= nichols (sys) %valores do módulo, fase e freqüência
≫[mod,fase,w]= nichols (sys,w) % valores do módulo e fase para faixa de freqüência especificada
Considere o seguinte sistema dado pela sua função de transferência em malha aberta

$$G(s)H(s) = \frac{-4s^2 + 48s^3 - 18s^2 + 250s + 600}{s^2 + 30s^3 - 18s^2 + 282s + 60}$$

O diagrama de Nichols pode ser obtido a partir dos comandos (Figura 2.30.)
≫num = [-4 48 -18 250 600];
≫den = [1 30 282 525 60];
≫sys=tf(num,den);
≫nichols (num, den)
≫ngrid

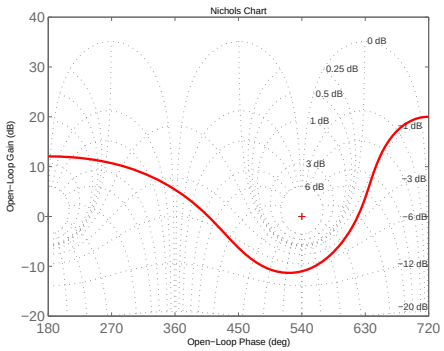

Figura 2.30: Diagrama de Nichols.

2.17.13 Identificação de modelos simples

Os parâmetros de modelos com polos e zeros e um atraso para descrever processos utilizando respostas a entradas típicas podem ser obtidos com o uso de uma interface gráfica denotada 'ident'. A seguir três processos frequentemente utilizados em laboratório são identificados: motor CC, mini-forno e circuito RLC. Para identificar o modelo para o primeiro e segundo processos considera-se a resposta ao degrau e para o terceiro a resposta em frequencia.

Identificação dos parâmetros do modelo de um motor CC

Identificar as constantes de tempo e ganho constante do modelo de um motor CC a partir da resposta à um degrau de tensão aplicado na armadura. Considere a função de transferência entre a velocidade $wrad/s$ e a tensão aplicada na armadura v_a

$$\begin{aligned} G(s) &:= w(s)/v_a(s) \\ &= \frac{K_m}{(s+\gamma)(s+\delta)} \end{aligned}$$

As linhas de comando para chamar e definir dados de entrada e saída do motor CC são mostradas na Tabela 2.1

Tabela 2.1: Linhas de comando para uso do 'ìdent' para obter parâmetros do motor CC

```
clear all; close all;
clc;
load dados.dat;
                                    %carregar dados do ensaio de tensão
t = dados(:,1);                     % tempo (s) %
w = dados(:,2);                     % velocidade angular (rad/s)%
Ts = 1/3000                         %taxa de amostragem
motor = iddata;motor.Tstart = 0;    % define objeto
motor.Ts = Ts
motor.InputData = 12*ones(size(t)); % define entrada degrau
motor.OutputData = w;               % define resposta ao degrau obtida
ident                               % chamar a interface gráfica
                                    % importar dados via objeto 'motor'
                                    % escolher modelo
                                    % executar o comando 'Estimate'
motorid.sid                         % salvar a seção
ident('motorid.sid'                 % chamar a seção salva previamente
```

Identificação do modelo de um miniforno

Identificar a constante de tempo, atraso e ganho constante do modelo de um miniforno a partir da resposta à um degrau de tensão aplicado na resistência. Considere a função de transferência entre a temperatura temperatura em ^{o}C e a tensão aplicada na resistencia térmica

$$G(s) \quad := \quad \frac{Ke^{-sL}}{(1+sT)}$$

em que K é o ganho estático, T é a constante de tempo, e L é o atraso. As linhas de comando para chamar e definir dados de entrada e saída do miniforno são apresentadas na Tabela 2.2.

Tabela 2.2: Linhas de comando para uso do 'ìdent' para obter parâmetros do miniforno

```
clear all; close all; clc;
load dados;
y = (dados.Y.Data*100/.38); tempo = dados.X.Data;
u = 50*ones(size(y));                                  % tensão de 50V aplicada
D = dados.Capture.Downsampling;
Ts = dados.Capture.SamplingPeriod;
Ta = double(D)*Ts;                                     % tempo de amostragem
f1 = 0; f2 = 1; filtro = [0 (2*pi)*f2];                % Filtro passa banda [0,1Hz]
dadoss = iddata(y',u',Ta); dadoss.Tstart = 0;          % definindo objeto
tempos = dadoss.SamplingInstants;
disp('Filtrando...')
dadosfilt = idfilt(dadoss,filtro);
disp('Filtragem...OK')
disp('Reamostrando...')
P = 1; Q = 1000;
dadosr = resample(dadosfilt,P,Q,25,700);
tempor = dadosr.SamplingInstants;
figure(1)
disp('Reamostragem...OK')
saida = dadosr.Outputdata; entrada = dadosr.Inputdata;
ident                                                  %chamar a interface gráfica
```

Identificação do modelo de um circuito RLC

Identificação das constantes de tempo e ganho constante a partir da resposta em frequência. Seja um circuito RLC série como na Figura 5.3, tendo como entrada uma tensão $u(t)$ e como saída a tensão no capacitor $y(t)$, cuja função de transferência se

relaciona com a função na forma canônica de um sistema de segunda ordem da seguinte maneira:

$$G(s) = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}.$$

Considera-se o seguinte modelo autoregressivo com entrada externa (ARX, das iniciais em inglês) para representar os dados

$$A(s)y(s) = B(s)u(s) + e(s)$$

em que $A(s)$ e $B(s)$ denotam os polinômios $A(s) = s^{na} + a_1 s^{na-1} + \cdots + a_{na}$ e $B(s) = b_1 s^{nb-1} + \cdots + b_{nb}$, respectivamente. As linhas de comando para chamar e definir dados de entrada e saída do processo 2.3.

Tabela 2.3: Linhas de comando para uso do 'ìdent' para obter parâmetros do circuito RLC

```
load dados.dat;              %carregar dados da resposta em frequencia
An = dados(:,1);             % amplitude das oscilações
Fn = dados(:,2);             % fase das oscilações em graus
w=dados(:,3);                % frequencia das oscilações em rad/s
Ts=0;                        %sistema contínuo

RLC=idfrd(resposta,w,Ts);    % define objeto
                             %resposta=An*exp(i*Fn*pi/180)

ident                        % chamar a interface gráfica
                             % importar dados via objeto 'RLC'
                             % escolher modelo ARX de ordem na=2 e nb=1
                             % executar o comando 'Estimate'
RLC.sid                      % salvar a seção
ident('RLC.sid')             % chamar a seção previamente salva
```