

Laboratório de Sistemas de Controle I

João Carlos Basilio



Universidade Federal do Rio de Janeiro
Escola de Engenharia – Depto. de Eletrotécnica
3ª Edição: 2004

Sumário

1	Introdução	1
2	Programando em MATLAB	5
2.1	Modos de programação	6
2.1.1	Modo imediato	7
2.1.2	Arquivo de comandos	9
2.1.3	Arquivos de funções	10
2.2	Funções pré-definidas pelo MATLAB	12
2.2.1	Funções para manipulação de matrizes	14
2.2.2	Funções matemáticas elementares	15
2.2.3	Manipulação de vetores	15
2.2.4	Funções polinomiais	19
2.3	Controle de fluxo	21
2.3.1	Declaração if	21
2.3.2	Laço while	24
2.3.3	Laço for	24
2.3.4	A função find	27
2.4	Representação gráfica	27
2.4.1	O comando plot	27
2.4.2	Representação de diversas curvas num mesmo gráfico .	29
2.4.3	Representação de diversos gráficos numa mesma figura	30
2.4.4	Outros comandos para representação gráfica	31
2.5	Comentários finais	33
3	Modelagem e Identificação	39
3.1	Motor CC controlado pela armadura	39
3.2	Identificação dos parâmetros	43
3.2.1	Identificação de K_t , K e \bar{K}	43

3.2.2	Identificação de τ	53
3.3	Ajuste da f.t. por mínimos quadrados	53
3.4	Experimento para identificação	60
3.4.1	Experimento para determinação de K , \bar{K} e K_t	61
3.4.2	Experimento para obtenção da resposta em frequência	62
3.5	Validação do modelo	62
4	Projeto do controlador de velocidade	67
4.1	Fundamentos teóricos	68
4.1.1	Rastreamento e rejeição assintótica de sinais de dinâmica conhecida	68
4.1.2	Sensibilidade	73
4.1.3	Complementos	75
4.2	Projeto do controlador de velocidade	76
4.2.1	Sistema de controle em malha aberta	78
4.2.2	Sistema de controle com realimentação	78
4.3	Comentários finais	81
5	Implementação do controlador	83
5.1	Amplificadores Operacionais	84
5.1.1	Conceitos básicos	84
5.1.2	Configurações básicas	85
5.2	Controlador automático de velocidade	93
5.2.1	Implementação do comparador	94
5.2.2	Implementação do controlador	95
5.2.3	Amplificador de potência	96
5.3	Experimento final	96
5.4	Comentários finais	97

Capítulo 1

Introdução

Um projeto de um sistema de controle compreende, de uma maneira geral, as seguintes etapas (Kheir et al., 1991; Stephan, 1991):

1. Modelagem/identificação do sistema a ser controlado;
2. Projeto de controladores que satisfaçam as especificações de desempenho e estabilidade relativa exigidas;
3. Simulação utilizando computadores digitais;
4. Implementação do controlador no sistema real.

Note que as fases acima podem vir a constituir um ciclo com diversas repetições. Isto se deve ao fato de que durante a modelagem de um sistema físico (planta), são feitas várias hipóteses simplificadoras (linearidade, por exemplo). Assim, ao se implementar um sistema de controle utilizando-se um controlador projetado a partir de um certo modelo matemático, o sistema real pode vir a ter um comportamento diferente daquele inicialmente previsto na simulação. Algumas vezes, as especificações de desempenho/estabilidade podem não ser satisfeitas quando da implementação do controlador no sistema real, embora tivessem sido plenamente verificadas nas simulações. Neste caso, torna-se necessário projetar um novo controlador ou até mesmo desenvolver um novo modelo matemático para a planta. Outro aspecto a ser observado é que, se durante a etapa de projeto/simulação as especificações de desempenho/estabilidade não forem atendidas para um determinado compensador com uma certa estrutura (controlador PI, por exemplo), uma nova estrutura para o controlador — ou até mesmo uma nova metodologia de projeto — deve ser adotada. Pode-se, até mesmo, em alguns casos, verificar se há alguma possibilidade de se relaxar as especificações.

No Laboratório de Sistemas de Controle I, procuraremos abranger todas

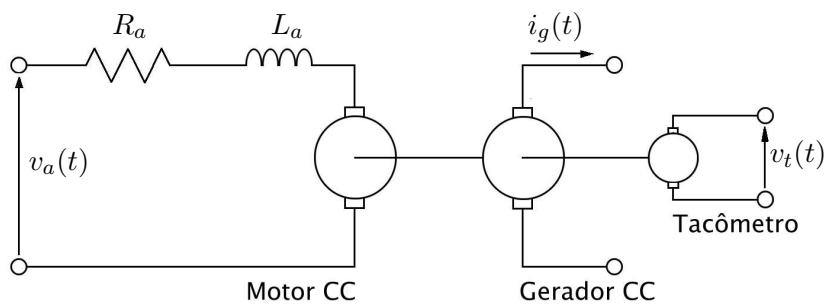


Figura 1.1: Circuito elétrico equivalente para o grupo motor-gerador com sensor de velocidade (tacômetro)

as fases de um projeto de um sistema de controle por realimentação. A planta a ser adotada é a de um grupo motor-gerador cujo circuito equivalente está representado na figura 1.1, onde $v_a(t)$ representa a tensão nos terminais da armadura do motor, $v_t(t)$ a tensão nos terminais do tacômetro (proporcional à velocidade angular do gerador) e $i_g(t)$ denota a corrente fornecida pelo gerador quando uma carga é conectada aos seus terminais.

Ao se formular um problema de controle, o primeiro passo é a definição da grandeza a ser controlada. Em nosso caso, a grandeza escolhida será a velocidade angular do gerador. A motivação para esta escolha está no fato de que na geração de tensões alternadas senoidais, a frequência angular deve ser mantida dentro de um intervalo bastante rígido. Como a frequência angular é proporcional à rotação do motor, o controle da velocidade angular do gerador surge como um objetivo claro de projeto. Nos sistemas de potência, a única diferença da planta aqui adotada é a substituição do gerador de corrente contínua (CC) por um de corrente alternada (CA).

De uma forma mais detalhada, o problema de controle a ser perseguido aqui pode ser enunciado da seguinte forma: projete um controlador de tal forma que o sistema realimentado: (i) seja estável; (ii) tenha erro de estado permanente nulo para uma determinada velocidade de referência; (iii) rejeite assintoticamente (quando o tempo tende para infinito) perturbações que, no sistema em estudo, são decorrentes da introdução de cargas nos terminais do gerador; tais cargas, ao demandarem corrente do gerador, aumentam o torque resistivo no eixo do motor, fazendo com que a velocidade de rotação

do grupo diminua.

No projeto de sistemas de controle, o engenheiro é geralmente assistido por programas computacionais tais como o MATLAB ou SIMULINK (MathWorks, 1996). Em nosso projeto utilizaremos o SIMULINK para a simulação do sistema projetado e o MATLAB como ferramenta auxiliar de projeto. Uma breve introdução à programação em ambiente MATLAB será apresentada no capítulo 2. No capítulo 3 será desenvolvido um modelo matemático para a planta e em seguida os parâmetros da função de transferência do sistema serão identificados. O método de identificação adotado é o da resposta em frequência, utilizando-se dois procedimentos distintos: *(i)* os diagramas de Bode, que permitem a determinação dos pólos e zeros da função de transferência do sistema a partir do traçado de assíntotas e *(ii)* o método dos mínimos quadrados, que ajustam as curvas de resposta em frequência de uma função de transferência de uma certa ordem aos pontos obtidos em laboratório de tal sorte que a soma dos quadrados das distâncias entre os pontos obtidos experimentalmente e aqueles a partir da função identificada (ambos tomados em frequências idênticas) seja mínima. No capítulo 4 será feito o projeto de um sistema de controle de velocidade para um grupo motor-gerador. Serão também realizadas simulações objetivando-se não só ilustrar os objetivos da realimentação como também mostrar a importância de programas computacionais tais como o MATLAB e o SIMULINK no projeto de sistemas de controle. Finalmente, no capítulo 5 será feita a implementação do controlador de velocidade desenvolvido no capítulo 4.

É importante ressaltar que, embora aplicado a um sistema específico, a abordagem aqui apresentada pode ser aplicada a diversos outros sistemas físicos. Assim sendo, caso o aluno venha a se deparar com um processo diferente daquele que estamos considerando neste laboratório, os mesmos passos que levaram à solução do problema aqui apresentado poderão ser seguidos na solução do problema em questão.

Capítulo 2

Programando em MATLAB

Uma das principais razões do sucesso da programação em ambiente MATLAB nos meios científicos é certamente a possibilidade que o usuário tem de manipular matrizes, vetores, polinômios e diversas funções matemáticas de forma idêntica a que está acostumado a fazer no dia-a-dia. A idéia central dessa linguagem é transportar para um ambiente computacional os mesmos símbolos e convenções da matemática – daí o nome MATLAB. Para tanto, o MATLAB não só dispõe de funções pré-definidas como também permite que o usuário desenvolva suas próprias funções. Isto fez com que essa linguagem crescesse, uma vez que mais e mais funções foram desenvolvidas, aumentando cada vez mais o potencial e a sua aplicabilidade a diversos campos da ciência.

No projeto de sistemas de controle, o MATLAB é também uma poderosa ferramenta. Além de realizar os cálculos, que em geral são longos, tediosos e de grande probabilidade de ocorrência de erro quando realizados pelo ser humano, o MATLAB tem uma excelente interface gráfica que auxilia o engenheiro de controle a verificar se o sistema compensado com o controlador projetado satisfaz as especificações de projeto. Outro aliado importante do engenheiro de controle é o programa SIMULINK, que faz simulações de sistemas lineares e não-lineares, discretos e contínuos no tempo. Uma vez projetado um controlador, pode-se inseri-lo no sistema e realizar a simulação, quando se poderá verificar se tal controlador satisfaz as especificações de desempenho.

Neste capítulo apresentaremos alguns tópicos sobre programação em ambiente MATLAB visando o seu uso em desenvolvimento de programas para identificação de sistemas e no projeto de sistemas de controle. Evidentemente há muito mais do que será apresentado aqui. Recomenda-se ao aluno mais

interessado consultar os manuais do MATLAB e do SIMULINK.

2.1 Modos de programação

Ao iniciar o MATLAB, estando o usuário em ambiente WINDOWS, será aberta uma janela. Esta janela define um espaço de trabalho. O espaço de trabalho nada mais é do que uma área de memória onde serão armazenadas as variáveis que vierem a ser definidas pelo usuário. Além dessas variáveis, o usuário, dentro de um espaço de trabalho tem acesso a: *(i)* funções definidas pelo MATLAB; *(ii)* funções desenvolvidas pelo usuário, desde que estejam armazenadas no próprio diretório em que o usuário esteja trabalhando ou em diretórios que constem do caminho de busca do MATLAB.

A programação em ambiente MATLAB pode ser feita nas seguintes formas:

1. Modo imediato;
2. Utilizando um arquivo de comandos;
3. Utilizando arquivos de funções.

No modo imediato, uma linha de comando é executada tão logo a tecla `<enter>` é pressionada. Com um arquivo de comandos pode-se executar uma série de instruções como se estivesse no modo imediato. A vantagem do arquivo de comandos sobre o modo imediato é que todos os comandos ficam guardados para uso futuro. A única desvantagem desse tipo de arquivo é que, rigorosamente falando, o usuário continua no modo imediato e, portanto, todas as variáveis que apareçam em um arquivo de comando virão a fazer parte do espaço de trabalho do usuário. Isto aumenta a área de memória em uso pelo MATLAB e pode ter o inconveniente de alterar o valor de alguma variável caso um novo valor seja atribuído a ela quando da execução do arquivo de comando. Este problema não ocorre quando se utiliza um arquivo de funções. Neste caso o MATLAB define uma nova área de memória para armazenamento das variáveis a serem definidas no arquivo de funções, permitindo assim que variáveis de mesmo nome que aquelas em uso no espaço de trabalho sejam definidas em um arquivo de funções sem, contudo, alterar o valor daquelas definidas pelo usuário.

2.1.1 Modo imediato

Uso de vírgula(,), ponto-e-vírgula (;) e reticências (...)

Comandos em uma mesma linha de comando podem ser separados por vírgula (,) ou ponto-e-vírgula (;) produzindo o mesmo efeito, exceto que no primeiro caso o resultado da operação é mostrado na tela. Caso uma linha de comandos seja demasiadamente longa, ela pode ser continuada na linha seguinte bastando para isso que a linha anterior termine com reticências seguida de `<enter>`. Neste caso o MATLAB somente irá executar a seqüência de comandos quando a tecla `enter` for pressionada sem que tenha sido precedida de reticências.

Exemplo 2.1

`>> a=2; b=5; <enter>`. Neste caso, o MATLAB armazena 2 e 5 nas posições de memória `a` e `b`, respectivamente, porém não exhibe os valores armazenados das variáveis.

`>> a=2, b=5 <enter>`. Esta linha de comandos produz o mesmo efeito que a linha de comando acima, exceto que a mensagem `a=2` e `b=5` é exibida na tela.

`>> c = a+b; <enter>`. Aqui o MATLAB armazena o valor 7 na posição de memória `c`, mas não exhibe na tela o seu conteúdo.

Caso se deseje, posteriormente, saber o valor de uma determinada variável, basta digitar o nome dessa variável seguida de `<enter>`. Por exemplo:

`>> c <enter>` faz com que o MATLAB exhiba na tela a mensagem `c = 7`.

□

Comandos importantes no modo imediato

Iremos agora apresentar alguns comandos que são geralmente usados no modo imediato. Esses comandos são usualmente empregados para mudança do diretório de trabalho (`chdir`), para listar todas as variáveis que estão sendo utilizadas pelo usuário (`who` e `whos`), para gravar as variáveis em um arquivo `.mat` (`save`), para carregar no espaço de trabalho as variáveis armazenadas em um arquivo `.mat` (`load`), para apagar determinadas variáveis do espaço de trabalho (`clear`) e para gravar em um arquivo de texto todos os comandos e mensagens de uma dada seção (`diary`). Na forma geral dos comandos a serem apresentados a seguir, `<caminho>` denota um determinado atalho, como por exemplo, `c:\usuário\matlab\labcont1`, `<variáveis>` denota uma lista contendo uma ou mais variáveis definidas pelo usuário, `<ar-`

quivo> representa um nome de arquivo cuja terminação (.txt, .mat ou .m) ficará clara no contexto. Uma descrição mais detalhada desses comandos é dada a seguir:

1. `chdir`, `cd`: mostram o diretório de trabalho atual;

`chdir <caminho>`: muda para o diretório especificado em `<caminho>`.

O comando `cd`, tem a mesma sintaxe de `chdir` e produz o mesmo efeito.

2. `who`: lista todas as variáveis em uso no espaço de trabalho. Note que variáveis definidas pelo usuário, mas já apagadas, não são listadas.

3. `whos`: lista na tela todas as variáveis em uso e as correspondentes dimensões.

4. `save`: armazena todas as variáveis em uso no arquivo `matlab.mat`;

`save <arquivo> <variáveis>`: guarda todas as variáveis da lista `<variáveis>` no arquivo `<arquivo>.mat`.

O exemplo seguinte ilustra o uso do comando `save`. Note que os valores atribuídos às variáveis são aqueles do exemplo 2.1, acima.

Exemplo 2.2

`>> save aula a b <enter>`. Inicialmente, MATLAB cria um arquivo denominado `aula.mat` e em seguida armazena as variáveis `a` e `b` nesse arquivo. Caso o arquivo `aula.mat` já existe, MATLAB apaga o seu conteúdo antes de guardar as variáveis.

`>> save aula <enter>`. Armazena as variáveis `a`, `b` e `c` no arquivo `aula.mat`.

`>> save <enter>`. Cria um arquivo `matlab.mat` e armazena nele as variáveis `a`, `b` e `c`. □

5. `load <arquivo>`: transfere todas as variáveis armazenadas em `<arquivo>.mat` para o espaço de trabalho. Por exemplo, o comando

`>> load <enter>`

localiza o arquivo `matlab.mat` e em seguida traz para o espaço de trabalho as variáveis nele armazenadas. Caso haja alguma variável no espaço de trabalho de mesmo nome que uma outra armazenada em `<arquivo>.mat`, ela será substituída pela variável que está sendo carregada, isto é, aquela armazenada em `<arquivo>.mat`.

6. `clear`: apaga todas as variáveis do espaço de trabalho.

`clear <variáveis>`: apaga somente as variáveis listadas em `<variáveis>`. Por exemplo, ao se digitar

```
>> clear a <enter>
```

o usuário irá apagar somente a variável `a`.

7. `diary <caminho\arquivo>`: grava no arquivo `<caminho><arquivo>` todos os comandos e mensagens de uma seção. Note que a terminação do arquivo, `.txt` por exemplo, deve fazer parte de `<arquivo>`. Este comando difere do comando `save` no seguinte aspecto: enquanto o comando `save` apaga o conteúdo do arquivo onde serão armazenadas as variáveis, `diary` acrescenta a mais recente mensagem ao conteúdo do arquivo. O comando `diary` é útil quando se deseja fazer um memorial do trabalho para consulta posterior.

`diary off` suspende a gravação dos comandos e mensagens no arquivo `<arquivo>`.

2.1.2 Arquivo de comandos

Os comandos editados em um arquivo de comandos podem ser executados diversas vezes. Para isso, basta digitar o nome do arquivo, sem a terminação `.m`, que o MATLAB executará todos as linhas de comandos do arquivo como se estivesse no modo imediato. A construção e a execução um arquivo de comandos segue os seguintes passos:

1. A partir do MATLAB, criar um bloco de notas `<arquivo>.m`;
2. Editar o arquivo `<arquivo>.m` com comandos a serem executados como se estivesse no modo imediato. Após alguma modificação no arquivo `<arquivo>.m`, deve-se gravar o arquivo, para que a alteração possa ser considerada pelo MATLAB na próxima vez que esse arquivo for executado.
3. Para executar os comandos do arquivo `<arquivo>.m` basta digitar `<arquivo> <enter>`.

Exemplo 2.3: Suponha que no arquivo `seq1.m`, sejam editados os seguintes comandos:

```
c=[1:2:10];d=[2:2:10];  
e=c';f=c'*d;g=c*d';  
h=c.*d;l=c./d;
```

Ao se digitar, na janela de comandos do MATLAB,

```
>> seq1 <enter>
```

o programa MATLAB, inicialmente, verifica se existe no espaço de trabalho alguma variável denominada `seq1`. Caso não encontre, MATLAB então procura no diretório de trabalho por um arquivo de comandos de nome

`seq1.m`. Encontrando o arquivo, executa os comandos editados nele. Caso contrário, procura em todos os caminhos de busca por um arquivo de comandos `seq1.m`. Exemplos de caminhos de busca são:

`c:\usuário\matlab\labcont1`

`c:\usuário\matlab\labcont2` □

No exemplo considerado, se o usuário estiver no mesmo diretório em que armazenou o arquivo `seq1.m`, então a seqüência de comandos acima será executada após `seq1 <enter>`. Caso contrário, o MATLAB iniciará uma procura em todos os caminhos de busca. Não achando, a mensagem

```
??? Undefined function or variable seq1.
```

será exibida na tela. É importante ressaltar que, se o arquivo `seq1.m` for gravado em dois diretórios, MATLAB executará aquele que estiver armazenado no diretório pertencente ao caminho de busca que aparece em primeiro lugar.

2.1.3 Arquivos de funções

Quando um arquivo de comandos é executado, todas as variáveis nele definidas passam a fazer parte do espaço de trabalho. Além de aumentar a área de memória utilizada pelo MATLAB (o que pode muitas vezes torna lenta a execução de comandos, visto que o MATLAB pode vir a ter que armazenar variáveis no disco rígido), isto faz com que o usuário tenha no seu espaço de trabalho um número, às vezes, excessivamente grande de variáveis. Outro aspecto a ser ressaltado é que, ao se usar arquivos de comandos, deve-se ter o cuidado de usar nomes de variáveis diferentes daquelas do espaço de trabalho, no caso de não se querer modificar o conteúdo delas.

Esses problemas são, em parte, solucionados com o emprego de arquivos de funções. Ao se executar uma função, o MATLAB define um novo espaço de trabalho diferente daquele definido inicialmente pelo usuário. Isto permite que variáveis de mesmo nome sejam definidas simultaneamente no espaço de trabalho e num arquivo de funções. O espaço de trabalho definido ao se executar uma função irá, como no caso da execução de um arquivo de comandos, aumentar a área de memória utilizada pelo MATLAB. Contudo, diferentemente do que ocorre quando se usa um arquivo de comandos, essa área será liberada após a execução da função.

Para se criar uma função, procede-se da seguinte forma:

1. Cria-se um arquivo `<função>.m`;
2. A primeira linha de execução desse arquivo deve ser:

`function [y1,y2,...,yn] = <função>(x1,x2,...,xm);`
 onde x_1, x_2, \dots, x_m são as variáveis de entrada e y_1, y_2, \dots, y_n são as variáveis de saída. Note que o nome da função, definido em `<função>`, deve ser o mesmo do arquivo onde ela foi editada.

3. Antes do primeiro comando a ser executado, pode-se escrever alguns comentários para orientação dos possíveis usuários daquela função. Toda linha de comentário se inicia com o sinal `%`.

4. Ao se digitar

```
>> help <função> <enter>
```

são exibidos, na janela aberta para o MATLAB, os comentários iniciais do arquivo `<função>.m`, isto é, aquelas iniciadas com `%` que precedem uma linha de comandos ou uma linha totalmente em branco.

Exemplo 2.4: Criar um arquivo de funções para calcular a norma euclidiana (norma-2) de um vetor $\underline{x} = [x_1 \ x_2 \ \dots \ x_n]^t$.

Inicialmente, lembre-se de que a norma euclidiana de um vetor é definida como:

$$\|\underline{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} = (\underline{x}^t \underline{x})^{1/2} \quad (2.1)$$

A função que calcula $\|\underline{x}\|_2$ deve, portanto, executar a equação (2.1), o que requer que a variável de entrada seja um vetor (\mathbf{x}) de dimensão n e a saída um escalar (y) representativo da norma-2 de \underline{x} . Se `norma2.m` é o nome do arquivo onde esta função será editada, então esse arquivo poderia ter o seguinte conteúdo:

```
% y=norma2(x) : calcula a norma euclidiana de um vetor
% onde: x : vetor de entrada (vetor coluna)
%       y : norma euclidiana de x
%
```

```
function y=norma2(x);
y=sqrt(x'*x);
```

□

Observe que digitando-se

```
>> help norma2 <enter>
```

as 4 primeiras linhas do arquivo serão exibidas na janela do MATLAB, porém o símbolo `%` não será exibido. Por exemplo

```
>> x=[1 2 2]';y=norma2(x);<enter>
```

faz com que MATLAB execute a função `norma2` para o vetor \mathbf{x} , armazenando na variável `y` o valor 3. Verifique!

Para que o resultado da função acima seja realmente a norma euclidiana do vetor \underline{x} , \mathbf{x} deve ser um vetor coluna. Caso contrário $\mathbf{x}'*\mathbf{x}$ será uma matriz e a saída \mathbf{y} será uma matriz $m \times m$ cujos elementos são as raízes quadradas de $x_i x_j$, onde x_i e x_j denotam as componentes i e j do vetor \underline{x} . Uma outra particularidade da função acima é o uso do comando `sqrt`, que é uma função pré-definida pelo MATLAB. Em um arquivo de função podem ser usadas outras funções quer sejam definidas anteriormente pelo usuário ou pré-definidas pelo MATLAB. Essas últimas permitem a manipulação de matrizes, vetores, polinômios e outras funções matemáticas de uma forma robusta, isto é por algoritmos que não são susceptíveis a problemas de mau condicionamentos.

Na seção seguinte serão apresentadas algumas funções pré-definidas pelo MATLAB. Essas funções foram escolhidas tendo em vista sua importância no desenvolvimento de programas para o projeto de sistemas de controle.

2.2 Funções pré-definidas pelo MATLAB

A característica principal do MATLAB é tratar indistintamente escalares, vetores e matrizes como sendo matrizes. Por exemplo, a variável `a=3` é para o MATLAB uma matriz de dimensão 1×1 ; um vetor de dimensão n é, na verdade, uma matriz de dimensões $n \times 1$ ou $1 \times n$, dependendo de se tratar de um vetor coluna ou de um vetor linha. Assim sendo, tanto o argumento (entrada) de uma função como o valor que ela retorna (saída) são matrizes, embora para o usuário tratam-se simplesmente de vetores ou até mesmo escalares.

Para se definir uma matriz, deve-se fornecer cada um dos seus elementos. Por exemplo, a matriz

$$A = \begin{bmatrix} 1 & 2,2 & 3 \\ 4,6 & 5 & 6 \end{bmatrix}$$

pode ser armazenada na variável `a` das seguintes formas:

(i) Fornecendo-se cada um dos elementos da matriz:

```
>> a(1,1)=1;a(1,2)=2.2;a(1,3)=3; <enter>
```

```
>> a(2,1)=4.6;a(2,2)=5;a(2,3)=6; <enter>
```

(ii) Entrando com os elementos da matriz entre colchetes (`[]`), sendo os elementos de uma mesma linha separados por vírgula (`,`) ou espaços em branco e as linhas da matriz por ponto-e-vírgula (`;`):

```
>> a=[1,2.2,3;4.6,5,6] <enter>
```

```
>> a=[1 2.2 3;4.6 5 6] <enter>
```

Deve-se tomar o cuidado para, no caso de números com parte decimal, separar a parte inteira da parte decimal usando ponto (.);

O acesso a qualquer elemento de uma matriz que tenha sido anteriormente definida, e que ainda esteja armazenada no espaço de trabalho, é feito de forma semelhante, isto é, fornecendo-se a posição do elemento desejado. Suponha, então, que os elementos de uma matriz $M : p \times q$ tenham sido armazenados na variável m .

(i) Para se ter acesso ao elemento (i, j) de m faz-se $m(i, j)$. Por exemplo, para modificar o elemento $(1, 3)$ da variável a , acima, fazendo-o igual a 7, basta digitar:

```
>> a(1,3)=7 <enter>
```

Para se armazenar o elemento $(1, 2)$ na variável x , faz-se:

```
>> x = a(1,2) <enter>
```

(ii) Fazendo-se $m(\text{vetorlinha}, \text{vetorcoluna})$ forma-se uma submatriz de uma dada matriz m , formada pelos elementos comuns às linhas cujos índices são dados no vetor vetorlinha e às colunas indicadas pelas componentes do vetor vetorcoluna . Por exemplo, para se tomar os elementos da interseção das linhas 1 e 2 e das colunas 2 e 3 de a e armazená-las em x , deve-se fazer:

```
>> x = a([1 2],[2 3]) <enter>
```

(iii) Para ter acesso à coluna j da matriz m faz-se $m(:, j)$. Por exemplo, para atribuir a terceira coluna da matriz a , acima, à variável x , faz-se:

```
>> x = a(:,3) <enter>
```

(iv) Analogamente, a coluna i de uma matriz m pode ser acessada fazendo-se $m(i, :)$. Por exemplo, suponha que se queira armazenar a linha 2 da matriz a , acima, em x . Para tanto, deve-se fazer

```
>> x = a(2,:) <enter>
```

Algumas matrizes não requerem que o usuário entre com todos os seus elementos. São elas: as matrizes identidade e diagonal, a matriz cujos elementos são todos iguais a 1 ou zero. A atribuição dessas matrizes a variáveis é feita da seguinte forma:

(i) $i = \text{eye}(m)$: atribui-se à variável i uma matriz identidade de ordem m ;

(ii) $d = \text{diag}([d1 d2 \dots dm])$: forma uma matriz diagonal d , $m \times m$, cujos elementos da diagonal principal são $d1, d2, \dots, dm$;

(iii) $um = \text{ones}(m,n)$: cria uma variável um , $m \times n$, em que todos os elementos são iguais à unidade;

(iv) $\mathbf{z} = \text{zeros}(m,n)$: forma uma matriz \mathbf{z} , $m \times n$, cujos elementos são todos iguais a zero.

O MATLAB possui um conjunto de funções pré-definidas que permitem manipular matrizes e vetores de forma análoga a que o engenheiro está acostumado a fazer no seu dia-a-dia. Estas funções são apresentadas a seguir.

2.2.1 Funções para manipulação de matrizes

Sejam

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1q} \\ b_{21} & b_{22} & \dots & b_{2q} \\ \vdots & \vdots & & \vdots \\ b_{p1} & b_{p2} & \dots & b_{pq} \end{bmatrix}$$

matrizes de dimensões $m \times n$ e $p \times q$, respectivamente, com elementos pertencentes ao conjunto dos números complexos. Suponha que as matrizes A e B tenham sido armazenadas nas variáveis \mathbf{a} e \mathbf{b} , respectivamente. Então:

(i) $[\mathbf{m},\mathbf{n}]=\text{size}(\mathbf{a})$: armazena, respectivamente, nas variáveis \mathbf{m} e \mathbf{n} , o número de linhas e de colunas da matriz \mathbf{a} ;

$\mathbf{m}=\text{size}(\mathbf{a},1)$: armazena na variável \mathbf{m} o número de linhas de \mathbf{a} ;

$\mathbf{n}=\text{size}(\mathbf{a},2)$: armazena em \mathbf{n} o número de colunas de \mathbf{a} .

(ii) $\mathbf{c}=\mathbf{a}*\mathbf{b}$: calcula o produto das matrizes \mathbf{a} e \mathbf{b} (se $n = p$) e armazena na variável \mathbf{c} . Quando $n \neq p$, MATLAB retorna uma mensagem de erro.

(iii) $\mathbf{c}=\mathbf{a}.*\mathbf{b}$ e $\mathbf{c}=\mathbf{a}./\mathbf{b}$: forma uma matriz \mathbf{C} tal que $c_{ij} = a_{ij}b_{ij}$ e $c_{ij} = a_{ij}/b_{ij}$, respectivamente, e armazena o resultado na variável \mathbf{c} . Este comando requer $m = p$ e $n = q$.

(iv) $\mathbf{c}=\text{inv}(\mathbf{a})$: calcula a inversa da matriz armazenada na variável \mathbf{a} (se $m = n$) e armazena na variável \mathbf{c} .

(v) $\mathbf{c}=\text{det}(\mathbf{a})$: calcula o determinante da matriz armazenada em \mathbf{a} , armazenando o resultado na variável \mathbf{c} .

(vi) $\mathbf{c}=\mathbf{a}+\mathbf{b}$ e $\mathbf{c}=\mathbf{a}-\mathbf{b}$: computa, respectivamente, a soma e a subtração das matrizes \mathbf{a} e \mathbf{b} e as armazena em \mathbf{c} .

(vii) $[\mathbf{w},\mathbf{l}]=\text{eig}(\mathbf{a})$: retorna a matriz \mathbf{w} , cuja i -ésima coluna é formada pelo o i -ésimo autovetor de \mathbf{a} associado ao i -ésimo autovalor e uma matriz diagonal \mathbf{l} , cujo elemento (i, i) corresponde ao i -ésimo autovalor de \mathbf{a} .

$\mathbf{l}=\text{eig}(\mathbf{a})$: retorna um vetor de dimensão n cujas componentes são os autovalores de \mathbf{a} .

2.2.2 Funções matemáticas elementares

Seja $Z : m \times n$, uma matriz cujo elemento i, j é $z_{ij} = |z_{ij}| e^{j\theta_{ij}} = x_{ij} + jy_{ij}$ e assumamos que a matriz Z tenha sido armazenada na variável \mathbf{z} . Tem-se que:

(i) $\mathbf{w} = \text{abs}(\mathbf{z})$ retorna uma matriz \mathbf{w} cujo elemento (i, j) é $w_{ij} = |z_{ij}|$;
(ii) $\mathbf{w} = \text{angle}(\mathbf{z})$ retorna uma matriz \mathbf{w} cujo elemento (i, j) é $w_{ij} = \theta_{ij}$, em radianos;

(iii) $\mathbf{w} = \text{real}(\mathbf{z})$ é uma matriz onde $w_{ij} = x_{ij}$;
(iv) $\mathbf{w} = \text{imag}(\mathbf{z})$ é tal que $w_{ij} = y_{ij}$;
(v) $\mathbf{w} = \text{conj}(\mathbf{z})$ retorna uma matriz \mathbf{w} tal que $w_{ij} = x_{ij} - jy_{ij}$;
(vi) $\mathbf{w} = \mathbf{z}'$ calcula uma matriz $W = Z^* = \bar{Z}^t$ (associada da matriz Z), isto é $w_{ij} = x_{ji} - jy_{ji}$ e armazena na variável \mathbf{w} ;

(vii) $\mathbf{w} = \text{exp}(\mathbf{z})$ retorna uma matriz cujo elemento (i, j) é $w_{ij} = \exp(z_{ij})$;
(viii) $\mathbf{w} = \text{log}(\mathbf{z})$ retorna uma matriz onde $w_{ij} = \ln(x_{ij} + jy_{ij})$;
(ix) $\mathbf{w} = \text{log10}(\mathbf{z})$ produz uma matriz em que $w_{ij} = \log(x_{ij} + jy_{ij})$;
(x) $\mathbf{w} = \text{sin}(\mathbf{z})$, $\mathbf{w} = \text{cos}(\mathbf{z})$, $\mathbf{w} = \text{tan}(\mathbf{z})$, $\mathbf{w} = \text{acos}(\mathbf{z})$, $\mathbf{w} = \text{asin}(\mathbf{z})$ e $\mathbf{w} = \text{atan}(\mathbf{z})$

calculam matrizes W tais que $w_{ij} = \text{Sen}(z_{ij})$, $w_{ij} = \text{Cos}(z_{ij})$, $w_{ij} = \text{tg}(z_{ij})$, $w_{ij} = \text{arc Cos}(z_{ij})$, $w_{ij} = \text{arc Sen}(z_{ij})$ e $w_{ij} = \text{arc tg}(z_{ij})$, respectivamente, armazenando o resultado na variável \mathbf{w} . Note que: (a) as funções **sin**, **cos** e **tan** assumem que os argumentos são expressos em radianos; (b) os ângulos retornados pelas funções **acos**, **asin** e **atan** são expressos em radianos.

É importante salientar os seguintes pontos relativos ao uso dos comandos acima:

(1) quando se desejar obter a transposta de uma matriz e não o conjugado complexo da sua transposta (associada), deve-se proceder de uma das seguintes formas: $\mathbf{w} = \mathbf{z}'$ ou então $\mathbf{w} = \text{conj}(\mathbf{z}')$;

(2) deve-se tomar cuidado ao usar os comandos **log** e **log10** — o primeiro calcula o logaritmo neperiano, enquanto o último retorna o logaritmo decimal. Os exercícios propostos 2.1 e 2.2, ao final deste capítulo, servirão para ilustrar o uso dos comandos apresentados acima.

2.2.3 Manipulação de vetores

Para o MATLAB, vetores são, na verdade, matrizes nas quais uma das dimensões (número de linhas ou colunas) é igual a 1. Portanto, podem ser manipulados de forma idêntica a que é feito para as matrizes. Por exemplo, o vetor $\underline{v} = [1 \ 2.5 \ 7]$ pode ser definido das seguintes formas:

(1.i) `>> v(1,1)=1;v(1,2)=2.5;v(1,3)=7; <enter>`

(1.ii) `>> v=[1 2.5 7]; <enter>`

(1.iii) >> v=[1,2.5,7]; <enter>

Caso se tratasse de um vetor coluna, ($\underline{w} = \underline{v}^t$), poder-se-ia definir uma variável \underline{w} como se segue:

(2.i) >> w(1,1)=1;w(2,1)=2.5;w(3,1)=7; <enter>

(2.ii) >> w=[1 2.5 7]'; <enter>

(2.iii) >> w=[1;2.5;7]; <enter>

Apesar de considerar todas as variáveis como sendo matrizes, o MATLAB é capaz de reconhecer quando se trata de um vetor. Para isso, basta que uma das dimensões seja igual a 1 ou que, ao se definir o vetor, apenas uma das dimensões seja dada. Neste último caso, o MATLAB armazenará o vetor como sendo um vetor linha. Desta forma, para se definir os vetores \underline{v} e \underline{w} , acima, poder-se-ia substituir as instruções (1.i) e (2.i) por:

>> v(1)=1;v(2)=2.5;v(3)=7;w=v'; <enter>.

Outro aspecto importante na definição de um vetor é quando as componentes formam uma seqüência. Por exemplo, para armazenar a seqüência

$${}^s f = [a_1 \quad a_1 + r \quad a_1 + 2r \quad \dots \quad a_n],$$

onde $a_n = a_1 + nr$, na variável \underline{sf} faz-se

>> sf=[a1:r:an] <enter>;

Deve-se salientar que, caso a_n não fosse um elemento da seqüência, o último elemento do vetor \underline{sf} seria o maior valor tal que $a_1 + kr < a_n$, para um dado k . É importante lembrar que pode também ser feito o armazenamento de uma seqüência decrescente. Para tanto, a_1 deve ser maior que a_n e r deve ser negativo.

MATLAB também possui funções especiais para a manipulação de vetores. Sejam

$$\underline{x} = [x_1 \quad x_2 \quad \dots \quad x_n]$$

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

vetores de dimensões n e m , respectivamente, e assumamos que \underline{x} e \underline{y} estejam armazenados nas variáveis \underline{x} e \underline{y} . Os seguintes comandos são especialmente usados na manipulação de vetores:

(i) `length(x)` e `length(y)` retornam as dimensões dos vetores \underline{x} e \underline{y} (n e m , respectivamente);

(ii) `norm(x,p)` retorna a norma-p do vetor \underline{x} ($\|\underline{x}\|_p$). Deve ser ressaltado que a função `norm` permite que o vetor, cuja norma-p se deseja calcular, seja linha ou coluna. Lembre-se de que, a norma-p de um vetor \underline{x} é definida como

$$\|\underline{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

e, portanto:

$$\begin{aligned} \|\underline{x}\|_1 &= \sum_{i=1}^n |x_i| \\ \|\underline{x}\|_2 &= \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2} \end{aligned}$$

(iii) `[mx,i]=max(x)`. Se \mathbf{x} é um vetor cujas componentes são números reais, então `mx` é o maior valor de \mathbf{x} e `i` corresponde à posição onde o máximo ocorre. Caso \mathbf{x} tenha duas componentes onde os valores máximos ocorrem, `i` corresponderá à componente onde o valor máximo ocorre primeiro. Quando \mathbf{x} é um vetor complexo, `mx` corresponderá à componente de maior valor absoluto de \mathbf{x} e `i` corresponderá ao índice dessa componente.

(iv) `hflip=fliplr(x)` e `vflip=flipud(y)` retornam os seguintes vetores:

$$\begin{aligned} \underline{hflip} &= [x_n \ x_{n-1} \ \dots \ x_2 \ x_1] \\ \underline{vflip} &= \begin{bmatrix} y_m \\ y_{m-1} \\ \vdots \\ y_2 \\ y_1 \end{bmatrix} \end{aligned}$$

Exemplo 2.5: Escreva um arquivo do tipo funções para calcular a derivada de um polinômio de qualquer grau.

Seja

$$p(x) = p_0x^n + p_1x^{n-1} + p_2x^{n-2} + \dots + p_{n-1}x + p_n$$

um polinômio de grau n . Sabe-se que a derivada de $p(x)$ é dada por:

$$\frac{d}{dx} = np_0x^{n-1} + (n-1)p_1x^{n-2} + \dots + 2p_{n-2}x + p_{n-1}$$

sendo, portanto, um polinômio de grau $n - 1$. A função a seguir calcula a derivada de $p(x)$ em relação a x :

```
% DERIVADA : calcula a derivada de um polinomio p(x)
% pderiv = derivada(p)
%
% onde p = [po p1 p2 ... pn] sao os coeficientes do
% polinomio
%
function pderiv = derivada(p);
n=length(p)-1;pderiv=[n:-1:1].*p(1:n);
```

Agora, verifique se o programa está correto, calculando a derivada em relação a x do polinômio $p(x) = 5x^3 + 2x^2 - x + 1$. \square

Exemplo 2.6: Escreva um arquivo de comandos para calcular o instante de pico, o percentual de ultrapassagem, o valor máximo e o valor de estado permanente da resposta ao degrau do sistema cuja função de transferência é

$$G(s) = \frac{0,6618}{s^2 + 1,0069s + 0,6618}.$$

Considere que o intervalo de simulação seja de 0 a 20s

```
% INDICES : calcula indices de desempenho de um sistema
% [tp,po,ymax,yss] = indices(t,y)
%
% onde : y      : vetor contendo as repostas correspondentes
%           aos instantes de simulacao armazenados no
%           vetor t
%         t      : vetor contendo os instantes de simulacao
%         tp     : tempo de pico
%         po     : percentual de ultrapassagem
%         ymax   : valor maximo da resposta
%         yinf   : valor de estado permanente da
%
function [tp,po,ymax,yinf] = indices(t,y);
% calculando o ponto de maximo e o instante onde
%o maximo ocorre:
[ymax,i]=max(y);tp=t(i);
% calculando o valor de estado permanente:
yinf=y(length(y));
```

```
% calculando o percentual de ultrapassagem:
po=(ymax-yinf)/yinf*100;
```

□

Para se executar este programa, torna-se necessário, inicialmente, definir os vetores \mathbf{t} e \mathbf{y} . Supondo que os instantes de simulação estejam espaçados de 0.1s, tem-se que \mathbf{t} será definido da forma seguinte:

```
>> t=[0:0.1:20]; <enter>
```

Para obter \mathbf{y} , deve-se usar a função `step`, pertencente à caixa de ferramentas de controle (*control toolbox*), que calcula a resposta ao degrau de um sistema, sendo conhecidos os polinômios do numerador e do denominador da função de transferência do sistema. O terceiro argumento da função `step` é o vetor contendo os instantes de simulação. As linhas de comando seguintes permitem obter \mathbf{y} e também o tempo de pico (\mathbf{tp}), o percentual de ultrapassagem (\mathbf{po}), o valor máximo de \mathbf{y} (\mathbf{ymax}) e o valor de estado permanente de $\mathbf{y}(t)$ (\mathbf{yinf}):

```
>> n=0.6618;d=[1 1.0069 0.6618];y=step(n,d,t); <enter>
>> [tp,po,ymax,yinf]=indices(t,y); <enter>
```

É importante observar que quando o intervalo de simulação não for suficientemente grande, o valor de estado permanente \mathbf{yinf} poderá não ser o correto, isto é, o valor do $\lim_{t \rightarrow \infty} y(t)$. Portanto, o uso da função acima só faz sentido quando se dispõe da representação gráfica de $t \times y$ ou então verificando-se o vetor \mathbf{y} está, de fato, convergindo para o valor final esperado. □

2.2.4 Funções polinomiais

Sejam $p(x) = p_0x^n + p_1x^{n-1} + \dots + p_{n-1}x + p_n$ e $q(x) = q_0x^m + q_1x^{m-1} + \dots + q_{m-1}x + q_m$ polinômios de graus n e m , ($n \geq m$) respectivamente. Para se trabalhar com polinômios no MATLAB, devem ser definidos vetores cujas componentes são os coeficientes dos polinômios. Assim, os polinômios $p(x)$ e $q(x)$ serão, para o MATLAB, referidos como $\underline{p} = [p_0 \ p_1 \ \dots \ p_{n-1} \ p_n]$ e $\underline{q} = [q_0 \ q_1 \ \dots \ q_{m-1} \ q_m]$. MATLAB permite que polinômios sejam multiplicados ou divididos, que suas raízes sejam calculadas, ou o oposto, que se obtenha os coeficientes de um polinômio a partir de suas raízes, ou ainda, calcular o valor de $p(x)$, para diversos valores de x , conforme será descrito a seguir:

(i) Se o usuário desejar obter as raízes de $p(x)$ e armazená-las no vetor \underline{y} deve executar o seguinte comando: `y=roots(p)`.

(ii) O produto $r(x) = p(x)q(x)$ é efetuado pelo comando `r=conv(p,q)`.

(iii) Para se dividir $p(x)$ por $q(x)$, obtendo-se como quociente $a(x)$ e resto $r(x)$ faz-se `[a,r]=deconv(p,q)`. Se o usuário não desejar obter o resto, basta fazer `a=deconv(p,q)`.

(iv) Suponha que se deseje saber o valor de $p(x)$ nos pontos x_1, x_2, \dots, x_q . Para isso deve-se, inicialmente, definir um vetor $x = [x_1 \ x_2 \ \dots \ x_q]$ e, em seguida, fazer `y=polyval(p,x)`, obtendo $y = [p(x_1) \ p(x_2) \ \dots \ p(x_q)]$.

(v) Finalmente o comando `p=poly(raizes)` fornece um vetor p formado pelos coeficientes do polinômio cujas raízes são as componentes do vetor `raizes`. Por exemplo, `p=poly([-1 -2])` retorna o vetor $p=[1 \ 3 \ 2]$, formado pelos coeficientes do polinômio $p(x) = (x+1)(x+2) = x^2 + 3x + 2$.

Exemplo 2.7: Programa para calcular a resposta em frequência de uma dada função de transferência $G(s) = n(s)/d(s)$, sendo dado um vetor $\underline{\omega} = [\omega_0 \ \omega_1 \ \dots \ \omega_q]$, onde $\omega_i, i = 0, 1, \dots, q$ são as frequências onde se deseja calcular $G(j\omega)$.

```
% RESPFREQ : calcula a resposta em frequencia
% de G(s)=n(s)/d(s)
% gjw = respfreq(n,d,w)
% onde: n,d : vetores cujas componentes sao os
%          coeficientes de n(s) e d(s)
%          w : vetor cujas componentes sao as frequencias
%          onde se deseja calcular g(jw)=n(jw)/d(jw)
function gjw = respfreq(n,d,w);
jw = j*w;gjw=polyval(n,jw)./polyval(d,jw);
```

Suponha agora que se queira calcular a resposta em frequência do sistema cuja função de transferência seja

$$T(s) = \frac{20}{s^2 + 3s + 5}.$$

Para tanto, o primeiro passo é a definição de um vetor w contendo as frequências nas quais se deseja calcular $T(j\omega)$. Supondo que a menor frequência seja $\omega_0 = 0$ e a maior seja igual a $\omega_f = 100$. Neste caso fazendo-se:

```
>> n=20;d=[1 3 5];w=[0:1:100];
>> tjw=respfreq(n,d,w);plot(tjw) <enter>
```

obtem a resposta em frequência de $T(s)$ para os pontos considerados, cuja representação gráfica é dada na figura 2.1(a). Note que, nas baixas frequências, a representação gráfica é grosseira. Isto se deve ao fato de que, como

$T(j\omega)$ varia muito rapidamente nessas frequências, os pontos imagens estarão bastante separados uns dos outros, fazendo com que as retas que os ligam sejam mais ressaltadas. Este problema não ocorrerá se o vetor w for definido usando o comando `logspace`, cuja forma geral é `logspace(exp1,exp2,n)`. Este comando, cria um vetor de n elementos espaçados logaritmicamente, cujo primeiro elemento é 10^{exp1} e o último 10^{exp2} . Quando n for omitido, o comando acima criará um vetor composto de 50 elementos. Assim sendo, se substituirmos a linha de comandos acima por

```
>> n=20;d=[1 3 5];w=logspace(-2,2,100);
>> tjwt=respfreq(n,d,w);plot(tjwt) <enter>
```

teremos uma representação gráfica da resposta em frequência de $T(s)$ mais uniforme, conforme mostrado na figura 2.1(b).

2.3 Controle de fluxo

Às vezes, em um determinado programa, torna-se necessário mudar a ordem de execução dos comandos ou ainda repetir uma seqüência de comandos até que uma determinada condição seja verificada. Isto é feito, em MATLAB, usando-se as declarações `if` e os laços `for ... end` e `while ... end`, conforme será visto a seguir.

2.3.1 Declaração if

A forma geral para o uso da declaração `if` é a seguinte:

```
if <expressão booleana>
    <seqüência de comandos nº 1>
else
    <seqüência de comandos nº 2>
end
```

A declaração `if` funciona da seguinte forma: quando a `<expressão booleana>` for verdadeira, as linhas de comandos definidas em `<seqüência de comandos nº 1>` serão executadas e, caso contrário, serão executadas os comandos definidos na `<seqüência de comandos nº 2>`. Uma forma mais simples da declaração `if` é:

```
if <expressão booleana>
    <seqüência de comandos>
end
```

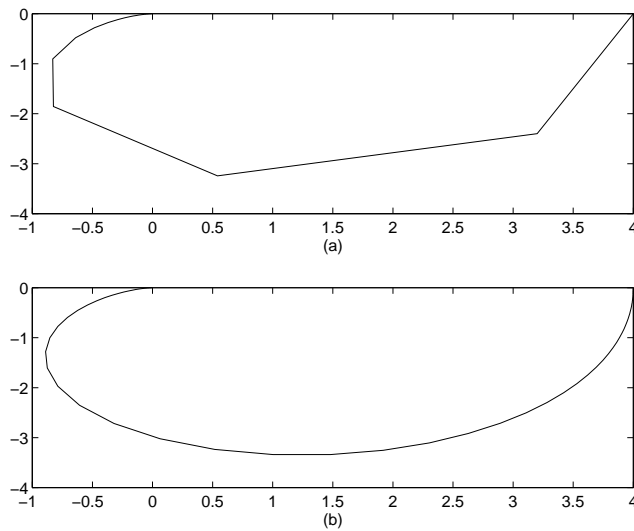



Figura 2.1: Representação gráfica de resposta em frequência de $T(s) = 20/(s^2 + 3s + 5)$ para (a) $w=[0:1:100]$ e (b) $w=\text{logspace}(-2,2,100)$

Neste caso, a <seqüência de comandos> será executada somente quando a <expressão booleana> for verdadeira.

O resultado da <expressão booleana> (verdadeiro ou falso) é definido a partir de operadores lógicos $\&$ (e), $|$ (ou) e \sim (não), cujas tabelas verdades são mostradas na tabela 2.1 e dos operadores relacionais ($<$ (menor que), $>$ (maior que), $=<$ (menor ou igual), $>=$ (maior ou igual), $==$ (igual) e $\sim=$ (diferente)).

Exemplo 2.8: O resultado encontrado pela função `norma2` do exemplo 2.4 estará correto quando a entrada for um vetor coluna. Para que o usuário

$\&$	V	F
V	V	F
F	F	F

(a)

$ $	V	F
V	V	V
F	V	F

(b)

\sim	V
V	F
F	V

(b)

Tabela 2.1: Tabelas verdades para os operadores lógicos $\&$ (e), $|$ (ou) e \sim (não)

possa utilizá-lo também para calcular a norma euclidiana de um vetor linha, o arquivo deve ter linhas que verifiquem se o vetor é do tipo linha ou coluna. Uma outra deficiência da função `norma2` é que ela é incapaz de distinguir se o usuário forneceu um vetor ou uma matriz. Todas essas deficiências são suprimidas utilizando-se declarações `if`, conforme mostrado abaixo:

```
% y=norma2(x) : calcula a norma euclidiana de um vetor
% onde:  x :  vetor de entrada
%        y :  norma euclidiana de x
%
function y=norma2(x);
[m,n]=size(x);
if m>1 & n>1;
    disp(' ');disp('????ERRO: a entrada deve ser um vetor');
    disp(' ');
    y=[];
else
    if m==1;y=sqrt(x*x');else;y=sqrt(x'*x);end
end
```

□

Agora verifique o programa para os vetores

$$\underline{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ e } \underline{w} = [1 \ 1]$$

e para a matriz

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}.$$

Note que, ao se fazer

```
>> a=[1 2;3 4];na=norma2(a);<enter>
```

o MATLAB retornou a mensagem

```
???? ERRO: a entrada deve ser um vetor.
```

Isto se deve ao comando `disp`, cuja forma geral é `disp('<texto>')`. Este comando faz com que a mensagem contida em `<texto>` seja impressa na janela do MATLAB. No arquivo acima usou-se, além do comando `disp` que contém a mensagem de erro a ser impressa na tela, outros dois contendo espaços em branco como texto. Eles foram usados para que o MATLAB deixe uma linha em branco entre a última linha de comando digitada pelo usuário e a mensagem impressa na tela e uma outra linha em branco entre essa mensagem e o próximo sinal de solicitação `>>`.

2.3.2 Laço while

A estrutura de um laço `while` é a seguinte:

```
while <expressão booleana>
    <seqüência de comandos>
end
```

Na estrutura acima, a <seqüência de comandos> será executada enquanto a <expressão booleana> retornar um valor verdadeiro. Por exemplo, considere um arquivo de comandos (`comandos.m`), cujo conteúdo é o seguinte:

```
resp='sim';i=0;
while resp=='sim'
    i=i+1
    resp=input('Deseja continuar (sim/nao)? ','s');
end
```

Crie um arquivo denominado `comandos.m` e digite os comandos acima.

Em seguida execute o arquivo digitando

```
>> comandos <enter>.
```

Observe que foram impressas na tela as seguintes mensagens:

```
i = 1
Deseja continuar (sim/nao)?
```

e o programa permanecerá parado até que o usuário digite `sim`, para continuar e `nao`, para parar definitivamente a execução do programa. Isto se deve ao comando `input`, que permite que o usuário atribua um valor a uma variável.

A forma geral do comando `input` é

```
<variável> = input('<texto>');
```

quando a <variável> é numérica ou

```
<variável> = input('<texto>','s');
```

quando a <variável> é do tipo cadeia de caracteres. Observe, no exemplo acima que, como a expressão booleana é `resp=='sim'`, o programa, na verdade, interromperá sua execução sempre o usuário digitar qualquer expressão que não seja `sim`.

2.3.3 Laço for

A forma geral do laço `for` é a seguinte:

```
for <contador>=<valor inicial>:<incremento>:<valor final>
    <seqüência de comandos>
end
```

onde

<contador> : variável de controle;
 <valor inicial> : valor inicial da variável;
 <incremento> : incremento dado à variável <contador>, podendo ser positivo ou negativo; quando o incremento for igual a 0, o laço for não será executado;
 <valor final> : maior ou menor valor que a variável <contador> pode assumir.

O funcionamento do laço for pode ser explicado com a ajuda das estruturas if e while, conforme mostrado abaixo:

```

<contador> = <valor inicial>;
if <incremento> > 0
  while <contador> <= <valor final>
    <seqüência de comandos>
    <contador> = <contador> + <incremento>
  end
else
  while <contador> >= <valor final>
    <seqüência de comandos>
    <contador> = <contador> + <incremento>
  end
end
end
  
```

Exemplo 2.9: Uma das formas de se representar matematicamente um sistema físico é através das chamadas equações de estados. Nesta representação, um sistema de ordem n será representado por um sistema de n equações diferenciais de primeira ordem, dado por:

$$\begin{aligned}\dot{\underline{x}}(t) &= A\underline{x}(t) + \underline{b}u(t) \\ y(t) &= \underline{c}\underline{x}(t) + du(t)\end{aligned}$$

onde $\underline{x}(t) \in \mathbb{R}^{n \times 1}$ é o vetor de estados, $u(t) \in \mathbb{R}$ e $y(t) \in \mathbb{R}$ denotam, respectivamente a entrada e a saída do sistema, $A \in \mathbb{R}^{n \times n}$, $\underline{b} \in \mathbb{R}^{n \times 1}$ e $\underline{c} \in \mathbb{R}^{1 \times n}$. As equações de estados permitem, por exemplo, verificar se o sistema é controlável, isto é, se existe uma entrada $u(t)$ capaz de levar o sistema a qualquer estado \underline{x}_f . Para tanto é necessário verificar se a matriz de controlabilidade

$$\mathcal{C} = [\underline{b} \quad A\underline{b} \quad A^2\underline{b} \quad \dots \quad A^{n-1}\underline{b}]$$

tem posto igual a n . O seguinte arquivo de comandos (`controle.m` pode ser utilizado para verificar se o sistema é controlável ou não:

```
% CONTROLE : verifica se um SLIT é controlável ou não
% onde : a:n X n (matriz de transição de estados) e
%        b:n X 1 (matriz de entrada)
[ma,na]=size(a);[mb,nb]=size(b);
% verificando se as dimensoes das matrizes
% a e b sao compatíveis
if ma==na & mb==na
    n=ma;ci=b;c=ci;
% construindo a matriz controlabilidade
    for i=1:n-1
        ci=a*ci;c=[c ci];
    end
% verificando se o sistema é controlavel
    disp(' ');
    if rank(c)==n
        disp('O sistema e controlavel');disp(' ');
    else
        disp('O sistema nao e controlavel');disp(' ');
    end
else
    disp(' ');
    disp('As dimensoes das matrizes nao sao compatíveis');
    disp(' ');
end
```

Agora verifique se os seguintes sistemas são controláveis:

(a)

$$\dot{\underline{x}}(t) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$$

(b)

$$\dot{\underline{x}}(t) = \begin{bmatrix} 1 & 2 \\ 0 & 4 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u(t)$$

2.3.4 A função find

Embora não tenha o poder de alterar a seqüência de execução de um arquivo, a função `find`, pré-definida pelo MATLAB, tem como argumento uma expressão booleana. Sua forma geral é a seguinte:

```
<variável> = find(<expressao booleana contendo um vetor>)
```

A função `find` retorna todos os índices das componentes do vetor para os quais a expressão booleana é verdadeira. Por exemplo, suponha que os vetores `t` e `y` contêm os instantes de simulação e os correspondentes valores da saída de um sistema de segunda ordem sub-amortecido para uma entrada igual ao degrau unitário. A função `find` pode ser usada para determinar os instantes de tempo imediatamente anterior e posterior àquele em que a saída atinge y_{ss} pela primeira vez e, conseqüentemente permite que se determine o tempo de subida da resposta, conforme mostra a seguinte seqüência de comandos:

```
yss=y(length(y));ind=find(y>yss);
% instante de tempo imediatamente posterior a tr
trp=t(ind(1));yrp=y(ind(1));
% instante de tempo imediatamente anterior ou igual a ts
tra=t(ind(1)-1);yra=y(ind(1)-1);
% calculo do tempo de subida (por interpolacao)
tr=tra+(trp-tra)*(yss-yra)/(yrp-yra)
```

Use os vetores `y` e `t` do exemplo 2.6 para verificar se os comandos acima estão, de fato, levando a uma correta determinação do tempo de subida da resposta.

2.4 Representação gráfica

2.4.1 O comando plot

Suponha que se deseje representar graficamente um ponto de coordenadas (x, y) no plano cartesiano. Isto pode ser feito em MATLAB utilizando-se o comando `plot`, cuja forma geral é a seguinte:

```
plot(x,y,'<cor><forma>')
```

onde `<cor>` define a cor usada na representação gráfica (conforme tabela 2.2) e `<forma>` define como o ponto será representado, isto é se será usado um ponto, círculo, um 'x', uma cruz ou um asteriscos, de acordo com as primeiras cinco linhas da tabela 2.3. Por exemplo, caso se deseje representar grafica-

Cor	Símbolo
Amarelo	y
Magenta	m
Azul claro	c
Vermelho	r
Verde	g
Azul escuro	b
Branco	w
Preto	k

Tabela 2.2: Cores usadas para representação gráfica em MATLAB

Forma	Símbolo
Ponto	.
Círculo	o
x	x
Cruz	+
Asteriscos	*
Linha contínua	-
Linha pontilhada	:
Linha tracejada	--
Linha traço-pontilhada	-.

Tabela 2.3: Formas de se representar pontos e curvas em MATLAB

mente o ponto de coordenadas $(-2, 1)$, na cor azul claro e com uma cruz, faz-se

```
>> plot(-2,1,'c+') <enter>
```

É importante ressaltar que, caso a forma de representação não esteja presente no comando `plot`, o MATLAB assumirá, como padrão, o ponto (`.`). Por exemplo, caso a linha de comando acima seja substituída por

```
>> plot(-2,1,'c') <enter>
```

então, o MATLAB utilizará o ponto (`.`) em vez de `+` para representar o ponto $(-2, 1)$ no plano cartesiano.

Suponha, agora, que se deseje representar graficamente os pontos de coordenadas (x, y_1) , (x_2, y_2) , \dots , (x_n, y_n) num mesmo gráfico. Para tanto, o

primeiro passo é definir dois vetores, \underline{x} e \underline{y} por exemplo, cujas componentes são, respectivamente, as abscissas e as ordenadas dos pontos que se deseja representar. Em seguida, usar o comando `plot`, de acordo com a forma geral dada acima, onde \mathbf{x} e \mathbf{y} são agora vetores e não as coordenadas de um ponto, conforme consta da forma geral do comando `plot`. Por exemplo, suponha que se queira representar graficamente os pontos $(-1, 1)$, $(0, 2)$, $(1, -1)$ e $(2, 2)$. A linha de comando seguinte permite representá-los na cor amarela, unindo-os com retas:

```
>> x=[-1,0,1,2];y=[1 2 -1 2];plot(x,y) <enter>
```

Note que, no comando `plot` acima, não foram definidas a `<cor>` e a `<forma>` dos pontos. Quando isto ocorre, o MATLAB utiliza a cor amarela (definida como padrão) e liga os pontos por meio de retas. Para se representar os mesmos pontos acima sem uni-los, isto é, isoladamente, com asteriscos e na cor azul escuro, substitui-se a linha de comando acima por:

```
>> x=[-1,0,1,2];y=[1 2 -1 2];plot(x,y,'b*') <enter>
```

Finalmente, para representar aqueles pontos, unindo-os com linhas verdes tracejadas, deve-se digitar a seguinte linha de comando:

```
>> x=[-1,0,1,2];y=[1 2 -1 2];plot(x,y,'g--') <enter>
```

2.4.2 Representação de diversas curvas num mesmo gráfico

Você deve ter observado que, toda vez que foi digitado um novo comando `plot`, o gráfico anterior foi apagado, sendo substituído pelo gráfico cujas coordenadas dos pontos foram fornecidas no comando `plot` mais recente. A representação gráfica de diversas curvas em um mesmo gráfico se faz necessária em muitos problemas, principalmente quando se deseja analisar simultaneamente o comportamento de diversas grandezas ou, ainda, quando se deseja analisar a resposta de um sistema a diferentes entradas. Suponha, portanto, que se deseje utilizar um único gráfico para representar as curvas C_1, C_2, \dots, C_n , cujas abscissas e ordenadas estão armazenadas nos vetores $\underline{x}_1, \underline{y}_1, \underline{x}_2, \underline{y}_2, \dots$ e $\underline{x}_n, \underline{y}_n$. Isto pode ser feito de duas formas: (i) através de um comando `plot` com vários pares de vetores, isto é

```
>> plot(x1,y1,'<cor1><forma1>',...,xn,yn,'<corn><forman>')
```

ou (ii) utilizando-se o comando `hold`. A forma geral do comando `hold` é a seguinte:

```
hold on
```

para fazer com que todos os comandos `plots` após o uso do comando `hold on` representem graficamente os pontos sobre o último gráfico criado antes do

comando `hold on`. Para se desativar o comando `hold` basta digitar

```
>> hold off <enter>
```

Exemplo 2.10: Para se representar em um mesmo gráfico as funções $y(t) = \text{Sen}(t)$ e $z(t) = \text{Cos}(t)$, $t \in [0, 2\pi]$ deve-se, inicialmente, definir um vetor `t` cujas componentes são os instantes em que devem ser calculados $y(t)$ e $z(t)$ e, em seguida, calcular $y(t) = \text{Sen}(t)$ e $z(t) = \text{Cos}(t)$, conforme as linhas de comando a seguir:

```
>> t=[0:2*pi/100:2*pi]; <enter>
```

```
>> y=sin(t);z=cos(t); <enter>
```

Caso queira utilizar o comando `plot` com múltiplas entradas para representar $y(t)$ e $z(t)$ nos pontos definidos nas linhas de comando acima, o usuário deve digitar:

```
>> plot(t,y,t,z) <enter>
```

A partir do gráfico obtido com o comando acima, vê-se que, embora o usuário não tenha definido nenhuma cor para as abscissas `y` e `z`, o MATLAB utilizou o amarelo para representar $y(t)$ e a magenta para representar $z(t)$. De fato, quando do uso do comando `plot` com múltiplas entradas, caso não sejam definidos `<cor>` e `<forma>` para cada curva, o MATLAB atribuirá cores diferentes a cada uma das curvas (na seqüência das linhas da tabela 2.2) e utilizará linhas contínuas (isto é, ligará os pontos por retas).

Uma maneira alternativa de representar as funções $y(t)$ e $z(t)$ num mesmo gráfico é usando o comando `hold`, conforme a linha de comando a seguir:

```
>> plot(t,y);hold on;plot(t,z);hold off
```

A diferença principal entre as duas formas de se representar diversas curvas em um mesmo gráfico está no fato de que, ao se usar o comando `hold on`, e não havendo definição de `<cor>` e `<forma>`, então o MATLAB usará a cor amarela para representar a curva e utilizará linha contínuas. Isto faz com que visualização de curvas representadas em um mesmo gráfico com o auxílio do comando `hold` não é tão boa quanto aquela fornecida pelo comando `plot` com múltiplas entradas, a menos que o usuário forneça detalhes referentes à `<cor>` e `<forma>` cada vez que usar o comando `plot`. \square

2.4.3 Representação de diversos gráficos numa mesma figura

Vários gráficos podem ser representados numa mesma figura com a ajuda do comando `subplot`. Este comando tem a forma geral

```
subplot(m,n,k)
```

e gera uma matriz de gráficos de dimensão $m \times n$, e faz com que todos os comandos associados à representação gráfica tenham efeito somente sobre o gráfico da posição k . Esta posição é definida como 1 para o gráfico correspondente à posição (1,1) da matriz, 2 para o gráfico da posição (1,2), n para o gráfico da posição (1, n), $n + 1$ para o gráfico da posição (2,1) da matriz e assim por diante. Por exemplo, para se representar as duas função do exemplo 2.10 em dois gráficos diferentes, um situado no topo da figura ($y(t) = \text{Sen}(t)$) e o outro na parte inferior da figura ($z(t) = \text{Cos}(t)$) procede-se da seguinte forma:

```
>> subplot(2,1,1);plot(t,y) <enter>
>> subplot(2,1,2);plot(t,z) <enter>
```

2.4.4 Outros comandos para representação gráfica

Muitas vezes, a representação gráfica, por si só, não é auto-explicativa. Ela requer legendas para que se visualize, com maior clareza, o que significa cada gráfico (no caso de múltiplos gráficos) ou cada linha do gráfico (quando se trara de um gráfico com diversas curvas). Para tanto, utiliza-se os seguintes comandos:

(i) `title('texto')` : coloca um título (<texto>) na parte superior do gráfico.

(ii) `xlabel('texto')` : escreve o <texto> imediatamente abaixo do eixo das abscissas.

(iii) `ylabel('texto')` : escreve o <texto> à esquerda do do eixo das ordenadas.

(iv) `gtext('texto')` : escreve na tela o <texto> em posição a ser definida pelo usuário. Após teclar <enter>, o MATLAB apresenta a figura e um sinal +, para ser posicionado, com o auxílio do *mouse*, sobre o ponto do gráfico onde se deseja escrever o <texto>. Deve-se tomar o cuidado de maximizar a janela que contém a figura para que o ponto escolhido para se colocar o <texto> corresponda àquele em que ele, de fato, será posicionado.

(v) `grid` : quadricula a tela com linhas tracejadas.

Finalmente, deve ser ressaltado que o MATLAB permite representar gráficos em escala logarítmica (muito importante na análise da resposta em frequência de sistemas lineares estáveis). Isto é feito utilizando-se os seguintes comandos:

(vi) `semilogx(x,y,'cor<forma>')`, `semilogy(x,y,'cor<forma>')` e, ainda, `loglog(x,y,'cor<forma>')` : permitem representar gráficos nos

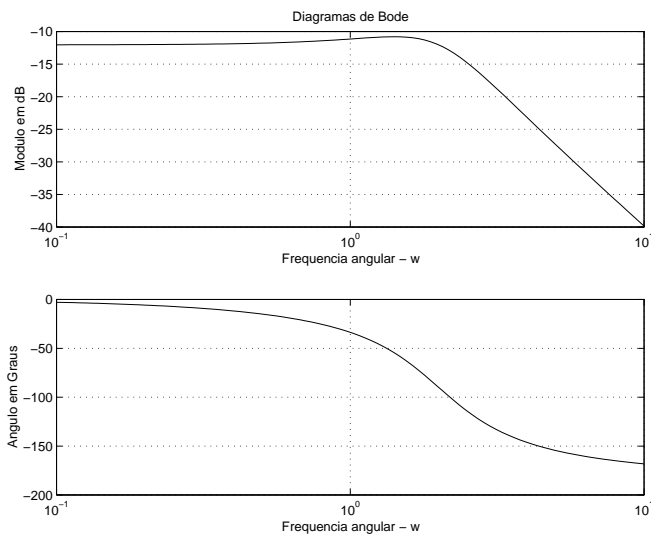


Figura 2.2: Diagramas de Bode para a função de transferência $G(s) = 1/(s^2 + 2s + 4)$

quais as abscissas, ordenadas e ambas, respectivamente, estão em escalas logarítmicas.

Exemplo 2.11: Escreva um arquivo de funções para calcular e traçar os diagramas de módulo e fase de Bode de uma dada função de transferência.

Solução: Denominemos `diagbode` a função que calcula os diagramas de Bode de módulo e de fase. Como entradas da função, temos os vetores formados pelos coeficientes dos polinômios do numerador e denominador da função de transferência, `n` e `d`, respectivamente, e o vetor `w` contendo as frequências onde se deseja calcular os diagramas de Bode. As saídas serão `gjwdb` e `fasegjw`, que são vetores cujas componentes são, respectivamente, o módulo, em decibéis, e a fase, em graus, da função de transferência nas frequências dadas em `w`.

```
% DIAGBODE : Calcula e traça os diagramas de Bode de uma
              funcao de transferencia G(s)=b(s)/a(s), onde
              b(s) = b0.s^m + b1.s^(m-1) + ... + bm
              a(s) = a0.s^n + a1.s^(n-1) + ... + an
% [gjwdb,fasegjw] = diagbode(b,a,w);
```

```

% onde b = [b0 b1 ... bm]
%       a = [a0 a1 ... an]
%       w = [w1 w2 ... wp]
function [gjwdb,fasegjw] = diagbode(b,a,w);
jw=j*w;gjw=polyval(b,jw)./polyval(a,jw);
gjwdb=20*log10(abs(gjw));fasegjw=angle(gjw)*180/pi;
subplot(2,1,1);semilogx(w,gjwdb);grid;
title('Diagramas de Bode');xlabel('Frequencia angular - w');
ylabel('Modulo em dB');
subplot(2,1,2);semilogx(w,fasegjw);grid
xlabel('Frequencia angular - w');ylabel('Angulo em Graus');

```

□

Para se obter os diagramas de Bode da função de transferência $G(s) = 1/(s^2 + 2s + 4)$, procede-se da seguinte forma:

```

>> n=1;d=[1 2 4];w=logspace(-1,1,100);
>> [gjwdb,fasegjw]=diagbode(n,d,w);

```

A função `diagbode` retornará o módulo de $G(j\omega)$ em decibéis e a fase de $G(j\omega)$, em graus, nas frequências definidas no vetor `w`, armazenando-os, respectivamente, nos vetores `gjwdb` e `fasegjw` e, também, produzirá os diagramas de módulo e fase, representados na figura 2.2.

2.5 Comentários finais

Conforme foi mencionado na introdução deste capítulo, o MATLAB possui muito mais recursos do que foi apresentado aqui. O objetivo principal deste capítulo foi introduzir os comandos básicos para que o usuário se sinta confortável ao utilizar o MATLAB e tenha a ferramenta mínima necessária para que possa vir a usar com sucesso linguagem. Para se ter acesso a comandos/recursos mais poderosos, recomenda-se a consulta aos manuais do MATLAB e do SIMULINK.

Exercícios propostos

2.1 Entre com a matriz

$$Z = \begin{bmatrix} 1-j & 1+2j & 2-j \\ & 2 & 1+2j \\ & & 3j \end{bmatrix}$$

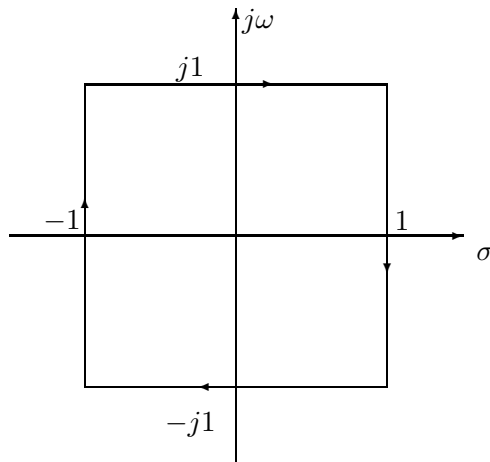


Figura 2.3: Contorno Γ para o exercício 2.4

e execute os comandos (i) a (vi) da seção 2.2.2.

2.2 Para o vetor

$$\underline{x} = [1 \quad 10 \quad 7 \quad 0.1]$$

execute os comandos (vii) a (ix) da seção 2.2.2.

2.3 Faça um programa MATLAB (arquivo de função) para traçar o diagrama de Nyquist de uma determinada função de transferência

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

sendo dados os coeficientes de $n(s)$ e $d(s)$ e um vetor $\underline{\omega}$ contendo as frequências de interesse. Por exemplo, a função poder-se-ia chamar de `meunyg` sendo descrita por `gjwt = meunyg(n,d,w)`.

2.4 Seja $s = \sigma + j\omega$ uma variável complexa e seja

$$F(s) = \frac{s}{s+2}.$$

Escreva um arquivo de comandos em MATLAB para calcular e representar graficamente o mapeamento de $F(s)$ sobre Γ , onde Γ é o contorno representado na figura 2.3.

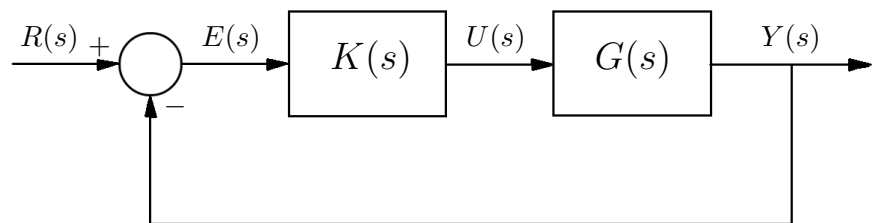


Figura 2.4: Diagrama de blocos para o exercício 2.6

2.5 Escrever um arquivo de função que permita somar dois polinômios $p(s) = p_0s^m + p_1s^{m-1} + \dots + p_{m-1}s + p_m$ e $q(s) = q_0s^n + q_1s^{n-1} + \dots + q_{n-1}s + q_n$. Teste a função para os polinômios $p(s) = s + 1$ e $q(s) = s^2 + 2s - 1$

2.6 Suponha que sejam dadas as funções de transferências da planta e do controlador

$$G(s) = \frac{n_G(s)}{d_G(s)} \text{ e } K(s) = \frac{n_K(s)}{d_K(s)}.$$

Escreva arquivos de funções que permitem calcular as seguintes funções de transferências para o sistema cujo diagrama de blocos está representado na figura 2.4: (a) $T_{RY}(s) = Y(s)/R(s)$, (b) $T_{RE}(s) = E(s)/R(s)$ e (c) $T_{RU}(s) = U(s)/R(s)$

Nota: As entradas para as funções serão os vetores formados pelos coeficientes dos polinômios $n_G(s)$, $d_G(s)$, $n_K(s)$ e $d_K(s)$ e as saídas serão vetores formados pelos coeficientes dos polinômios do numerador e do denominador da função de transferência considerada. Por exemplo, a função MATLAB que calcula o numerador e o denominador da função de transferência $T_{RY}(s)$, poderia ser definida como `[ntry,dtry] = try(ng,dg,nk,dk)`.

2.7 Suponha que sejam conhecidas as coordenadas de dois pontos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ e a abscissa de um terceiro ponto $P_3 = (x_3, y_3)$, onde $x_1 < x_3 < x_2$. Escreva um arquivo do tipo função que permita encontrar y_3 interpolando-se os pontos P_1 e P_2 por uma reta, isto é, a ordenada de P_3 será $y_3 = ax_3 + b$, onde a e b são, respectivamente, os coeficientes angular e linear da reta que passa pelos pontos P_1 e P_2 .

2.8 Suponha que sejam conhecidas as coordenadas de dois pontos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ e a ordenada de um terceiro ponto $P_3 = (x_3, y_3)$, onde $y_1 < y_3 < y_2$. Escreva um arquivo do tipo função que permita encontrar x_3 interpolando-se os pontos P_1 e P_2 por uma reta.

2.9 Escreva um arquivo do tipo função que permita calcular: (i) os tempos de subida (t_r) e de acomodação (t_s) para um sistema superamortecido ou criticamente amortecido e (ii) os tempos de subida (t_r), de pico (t_p) e de acomodação (t_s) e o percentual de ultrapassagem (PO) para um sistema subamortecido.

Nota: Assuma que sejam conhecidos t e $y(t)$, estando armazenados nos vetores $\underline{t} = [t_1 \ t_2 \ \dots \ t_n]$ e $\underline{y} = [y(t_1) \ y(t_2) \ \dots \ y(t_n)]$, respectivamente.

Atenção: Use o comando `find` e interpolação.

2.10 Suponha que

$$Q(s) = G(s)K(s) = K \frac{n_Q(s)}{d_Q(s)},$$

onde $n_Q(s)$ e $d_Q(s)$ são conhecidos e K deve ser arbitrado. Escreva um arquivo de funções que:

(a) Verifique se o sistema realimentado (realimentação negativa) é estável para um dado valor de K .

(b) Caso o sistema seja estável, encontre as margens de fase e de ganho do sistema.

Nota: Use o comando `find` e faça interpolação para achar os valores corretos.

2.11 Dado um conjunto de pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, escreva um arquivo de funções em MATLAB que utilize o método dos mínimos quadrados para calcular o coeficiente angular da reta $y = \alpha x$, conforme a figura 2.5.

Nota: Veja capítulo 3, equação 3.41

2.12 Suponha que a partir de um experimento, obtém-se os vetores

$$\begin{aligned} \underline{\omega} &= [\omega_0 \ \omega_1 \ \dots \ \omega_q] \\ \underline{\phi} &= [\phi_0 \ \phi_1 \ \dots \ \phi_q] \\ \underline{h} &= [|H(j\omega_0)| \ |H(j\omega_1)| \ \dots \ |H(j\omega_q)|] \end{aligned}$$

onde ω_i , $i = 0, 1, \dots, q$ são as frequências angulares e $H(j\omega_i) = |H(j\omega_i)| \exp(j\phi_i)$. Escreva um arquivo de funções que utilize o método dos mínimos

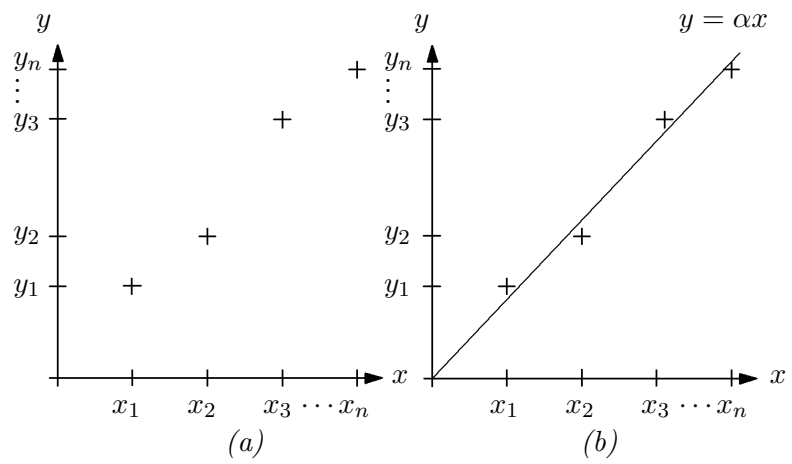


Figura 2.5: Representação cartesiana dos pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ e da reta ajustada $y = \alpha x$ referentes ao exercício 2.11

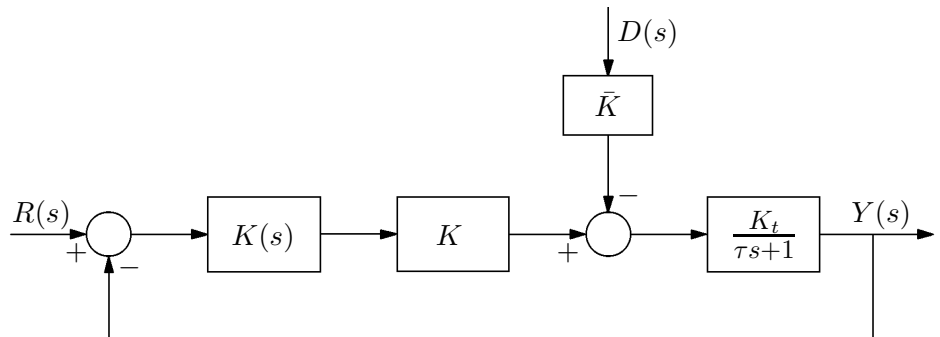


Figura 2.6: Diagrama de blocos para o sistema do exercício 2.13

quadrados (veja capítulo 3, algoritmo 5) para calcular os coeficientes da função de transferência

$$H(s) = \frac{b(s)}{a(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n},$$

onde os graus dos polinômios do numerador e do denominador, m e n , respectivamente, são arbitrados.

2.13 Construa um arquivo SIMULINK para fazer a simulação do sistema da figura 2.6, onde $K = 127.9$, $K_t = 0.0056$, $\tau = 0.026$ e $\bar{K} = 451.8$, para as seguintes situações:

(a) $r(t) = 12u_0(t)$ e $d(t) = 0$ e $K(s) = 5.4/s$, onde $u_0(t)$ representa um degrau unitário aplicado no instante $t = 0$.

(b) $r(t) = 12u_0(t)$ e $d(t) = 0$ e $K(s) = (s + 30)/s$.

(c) Reptita os itens (a) e (b) para $d(t) = 0.2u_0(t - 1)$

Nota: Em todos os casos acima, o intervalo de simulação deve ser de -0.5 a 2s.

Capítulo 3

Modelagem e identificação dos parâmetros da função de transferência do grupo motor-gerador

3.1 Modelo matemático de um motor CC controlado pela armadura

Para se modelar o grupo motor-gerador, cujo circuito equivalente é aquele da figura 3.1, basta realizar a modelagem de um motor CC controlado pela armadura (Dorf, 1986). A influência do gerador CC no modelo do motor será levada em conta pelos seguintes fatores: *(i)* aumento do momento de inércia da carga mecânica e *(ii)* aparecimento de um torque de perturbação, resultado da introdução de cargas de natureza elétrica nos terminais do gerador. Isto mostra que, conforme ressaltado no capítulo 1, o fato de termos um gerador CC no lugar de um CA, como seria o caso de um sistema real, não traz qualquer alteração no que se refere à modelagem do sistema. Considere, portanto, o circuito equivalente de um motor-CC controlado pela armadura da figura 3.2, onde $i_a(t)$ e $v_a(t)$ denotam, respectivamente, a corrente e a tensão de armadura, V_f e I_f representam, respectivamente, a tensão e a corrente de campo (constantes, por hipótese), $\omega(t)$ é a velocidade angular do motor e J e f são o momento de inércia da carga e o coeficiente de atrito nos mancais, respectivamente.

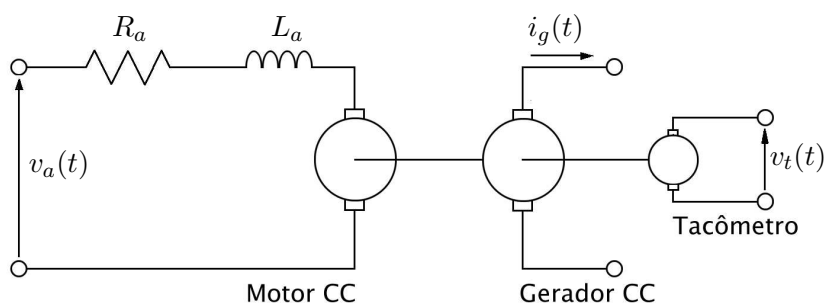


Figura 3.1: Circuito elétrico equivalente para o grupo motor-gerador com sensor de velocidade (tacômetro)

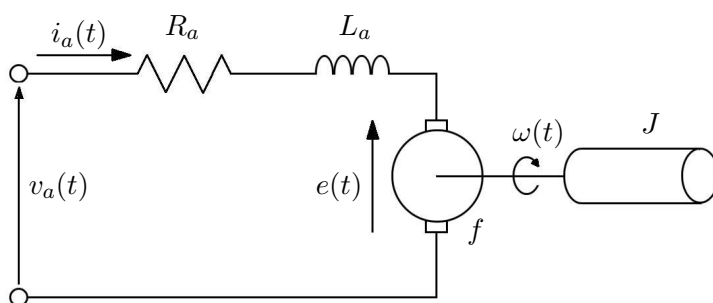


Figura 3.2: Circuito equivalente de um motor CC controlado pela armadura

Sabe-se que o torque produzido pelo motor, $t_m(t)$, é proporcional ao fluxo magnético no entreferro ($\phi(t)$) e à corrente de armadura ($i_a(t)$), isto é,

$$t_m(t) = K_f \phi(t) i_a(t) = K_m i_a(t) \quad (3.1)$$

onde $K_m = K_f \phi = K_f K_\phi I_f$, com K_f , K_ϕ e I_f constantes. Aplicando-se a lei das tensões de Kirchhoff ao circuito da armadura, obtém-se:

$$v_a(t) = R_a i_a(t) + L_a \frac{d}{dt} i_a(t) + e(t) \quad (3.2)$$

onde $e(t)$ representa a força contra-eletromotriz, que é proporcional à velocidade angular do motor, sendo dada por:

$$e(t) = K_e \omega(t) \quad (3.3)$$

Finalmente, usando-se a lei de Newton para o movimento rotacional, pode-se escrever:

$$t_m(t) - t_d(t) - f\omega(t) = J \frac{d}{dt} \omega(t) \quad (3.4)$$

onde $t_d(t)$ representa um torque externo (perturbação).

As equações (3.1) a (3.4) nos permitem obter a função de transferência que relaciona as transformadas de Laplace da entrada ($V_a(s)$) da saída ($W(s)$). Para tanto, aplicando-se a transformada de Laplace a ambos os membros das equações (3.1) a (3.4), resulta:

$$\left. \begin{aligned} T_m(s) &= K_m I_a(s) \\ V_a(s) &= R_a I_a(s) + L_a s I_a(s) + E(s) \\ E(s) &= K_e W(s) \\ T_m(s) - T_d(s) - fW(s) &= JsW(s) \end{aligned} \right\} \quad (3.5)$$

e após alguma manipulação algébrica, obtém-se:

$$W(s) = \frac{K_m/(R_a f)}{(\tau_e s + 1)(\tau_m s + 1) + K_e K_m/(R_a f)} V_a(s) - \frac{(\tau_e s + 1)/f}{(\tau_e s + 1)(\tau_m s + 1) + K_e K_m/(R_a f)} T_d(s) \quad (3.6)$$

onde $\tau_e = L_a/R_a$ e $\tau_m = J/f$. Note que a função de transferência da equação (3.6) modela o motor CC como um sistema de 2ª ordem. Porém,

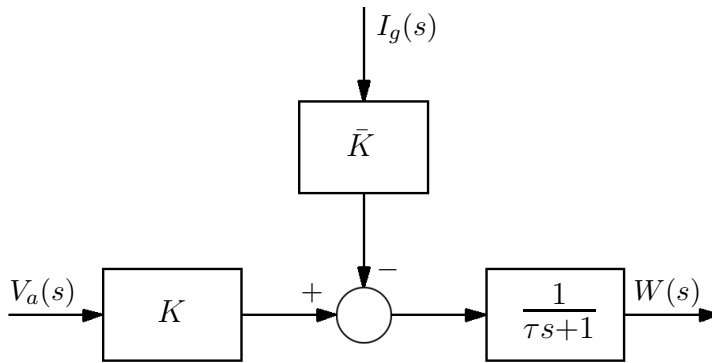


Figura 3.3: Diagrama de blocos do sistema composto pelo grupo motor-gerador

este sistema pode ser bem aproximado por um modelo de 1ª ordem, se levarmos em conta que $L_a/R_a \ll 1$ e portanto, $\tau_e s + 1 \approx 1$ para as frequências de interesse. Desta forma, o modelo matemático do grupo motor-gerador que iremos adotar será o seguinte:

$$W(s) = \frac{K}{\tau s + 1} V_a(s) - \frac{K_d}{\tau s + 1} T_d(s) \quad (3.7)$$

onde $K = K_m/(R_a f + K_e K_m)$ e $K_d = R_a/(R_a f + K_e K_m)$ e $\tau = J R_a/(R_a f + K_e K_m)$.

Note ainda que, como $t_d(t) = K_g i_g(t)$, a equação acima pode ser escrita como:

$$W(s) = \frac{K}{\tau s + 1} V_a(s) - \frac{\bar{K}}{\tau s + 1} I_g(s) \quad (3.8)$$

onde $\bar{K} = K_d K_g$. A representação por diagramas de blocos para este sistema está mostrada na figura 3.3.

É importante ressaltar que a medição da velocidade angular é feita por meio de tacômetros. Um tacômetro nada mais é do que um gerador CC de pequena potência, cuja tensão gerada é constante e proporcional à velocidade do eixo ao qual ele está acoplado. Desta forma, a tensão nos terminais do tacômetro, $v_t(t)$, será:

$$v_t(t) = K_t \omega(t), \quad (3.9)$$

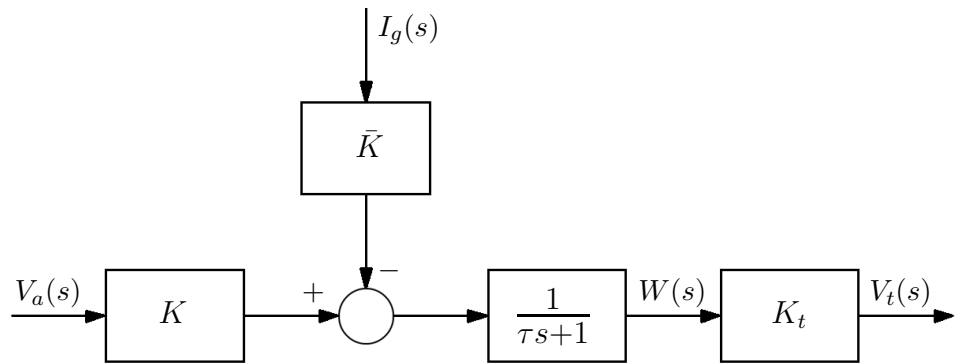


Figura 3.4: Diagrama de blocos representativo do modelo matemático do grupo motor-gerador-tacômetro

onde K_t é uma constante. Portanto, a relação entre $V_t(s)$, $V_a(s)$ e $I_g(s)$ pode ser expressa pela seguinte equação:

$$V_t(s) = \frac{K K_t}{\tau s + 1} V_a(s) - \frac{\bar{K} K_t}{\tau s + 1} I_g(s), \quad (3.10)$$

cujos diagrama de blocos está representado na figura 3.4.

3.2 Identificação dos parâmetros K_t , K , \bar{K} e τ

3.2.1 Identificação de K_t , K e \bar{K}

Uma vez que se dispõe de um modelo matemático da planta, o próximo passo é a identificação dos ganhos K_t , K e \bar{K} e da constante de tempo τ . Os ganhos serão identificados a partir da resposta em estado permanente a uma entrada igual ao degrau. A identificação de τ será feita utilizando-se métodos de resposta em frequência, em particular o diagrama de módulo de Bode e o diagrama polar. Nesta seção procederemos à identificação dos ganhos K_t , K e \bar{K} , ficando a identificação de τ adiada para a seção seguinte.

Suponha que um degrau de tensão de amplitude V_a seja aplicado aos terminais do motor, isto é,

$$v_a(t) = \begin{cases} 0, & t < 0 \\ V_a, & t \geq 0 \end{cases} \quad (3.11)$$

e assuma, inicialmente, que não há cargas conectadas nos terminais do gerador. Neste caso $V_a(s) = V_a/s$ e $I_g(s) = 0$, e portanto a equação (3.10) pode ser re-escrita da seguinte forma:

$$V_t(s) = \frac{KK_tV_a}{s(\tau s + 1)}. \quad (3.12)$$

A resposta do sistema à entrada descrita em (3.11) pode ser obtida calculando-se a transformada inversa de Laplace da equação (3.12). Procedendo desta forma obtemos:

$$v_t(t) = KK_tV_a(1 - e^{-\frac{1}{\tau}t}), \quad t \geq 0 \quad (3.13)$$

Quando $t \rightarrow \infty$, a equação acima se reduz a

$$v_t(t) = V_t = KK_tV_a, \quad (3.14)$$

o que mostra que quando uma tensão de valor constante é aplicada a um motor CC, a tensão em regime permanente nos terminais do tacômetro acoplado ao eixo deste motor também será constante e proporcional ao valor da tensão aplicada.

Definindo

$$K_{ta} = KK_t \quad (3.15)$$

pode-se escrever:

$$V_t = K_{ta}V_a. \quad (3.16)$$

O desenvolvimento acima sugere o seguinte procedimento para a identificação de K_{ta} :

Algoritmo 3.1

1. Excita-se o motor CC com tensões constantes e iguais a $V_{a_1}, V_{a_2}, \dots, V_{a_n}$, medindo-se os valores de tensão correspondentes, $V_{t_1}, V_{t_2}, \dots, V_{t_n}$, nos terminais do tacômetro. Note que, ao final deste passo, terão sido obtidos n pares cartesianos: $(V_{a_1}, V_{t_1}), (V_{a_2}, V_{t_2}), \dots, (V_{a_n}, V_{t_n})$, conforme ilustrado na figura 3.5(a).

2. Utilize o método dos mínimos quadrados para ajustar K_{ta} de tal forma que a soma dos quadrados das diferenças entre os valores observados experimentalmente $(V_{t_1}, V_{t_2}, \dots, V_{t_n})$ e $(K_{ta}V_{a_1}, K_{ta}V_{a_2}, \dots, K_{ta}V_{a_n})$, computados a partir da reta ajustada, seja mínima. A figura 3.5(b) mostra a representação cartesiana dos pontos $(V_{a_1}, V_{t_1}), (V_{a_2}, V_{t_2}), \dots, (V_{a_n}, V_{t_n})$ bem como a função linear $V_t = K_{ta}V_a$, cujo valor de K_{ta} se deseja ajustar. \square

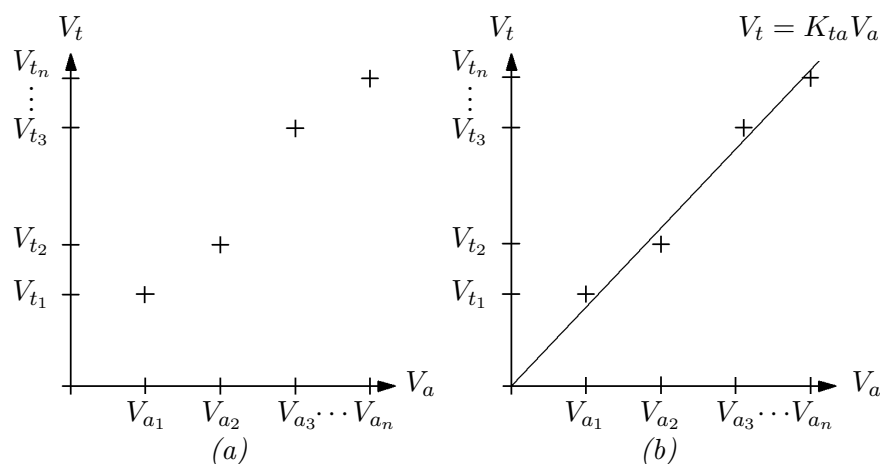


Figura 3.5: Representação (a) cartesiana dos pontos $(V_{a_1}, V_{t_1}), (V_{a_2}, V_{t_2}), \dots, (V_{a_n}, V_{t_n})$ e (b) da reta $V_t = K_{ta} V_a$

O método dos mínimos quadrados é um procedimento bastante utilizado devido a sua simplicidade, o que faz com que o ajuste proposto no algoritmo 3.1 não imponha qualquer complicação adicional ao cálculo de K_{ta} . A esta altura não iremos nos deter nos aspectos teóricos relacionados com a obtenção de uma expressão matemática que nos permita obter K_{ta} em função dos pontos $(V_{a_1}, V_{t_1}), (V_{a_2}, V_{t_2}), \dots, (V_{a_n}, V_{t_n})$ obtidos experimentalmente. Ao invés disso, apresentaremos ao final desta seção um algoritmo genérico que permite ajustar o coeficiente angular α de uma função linear $y = \alpha x$ a partir de um conjunto de pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ tal que a soma dos quadrados das diferenças entre as ordenadas (y_1, y_2, \dots, y_n) dos pontos observados e aquelas computadas a partir da reta ajustada $(\alpha x_1, \alpha x_2, \dots, \alpha x_n)$ seja mínima. A razão para se adiar a solução até o final desta seção se deve ao fato de que, conforme será visto mais adiante, as identificações de K , K_t e \bar{K} serão feitas a partir da solução de problemas idênticos ao descrito no passo 2 do algoritmo 3.1.

Uma vez que o valor de K_{ta} tenha sido determinado e se, por exemplo, o valor de K vier a ser encontrado, então de acordo com a equação (3.15), o valor de K_t pode ser calculado dividindo-se K_{ta} por K . Analogamente, se K_t vier a ser determinado então K pode ser determinado fazendo-se $K = K_{ta}/K_t$. Para o problema que estamos considerando, tanto a identificação

de K como a de K_t pode ser feita de maneira imediata, conforme será visto a seguir. Assim sendo faremos a identificação de ambos os ganhos (K e K_t) e usaremos o valor obtido para K_{ta} para verificar se os resultados estão corretos, isto é, identificando-se K , K_t e K_{ta} independentemente, deve-se ter:

$$K_{ta} \approx KK_t. \quad (3.17)$$

Vamos inicialmente proceder à identificação de K . A partir da equação (3.8), com $I_g(s) = 0$ e $V_a(s) = V_a/s$, tem-se:

$$W(s) = \frac{KV_a}{s(\tau s + 1)} \quad (3.18)$$

e portanto, a velocidade angular do motor, em estado permanente, será:

$$\omega(t) = W = KV_a. \quad (3.19)$$

Um procedimento análogo ao algoritmo 3.1 pode ser adotado também para a determinação de K , conforme ilustrado na figura 3.6, qual seja:

Algoritmo 3.2

1. Excita-se o motor com tensões constantes de valores iguais a $V_{a_1}, V_{a_2}, \dots, V_{a_n}$ (V) e mede-se as correspondentes velocidades angulares do motor W_1, W_2, \dots, W_n (*rpm*), utilizando-se para isso tacômetros digitais ópticos.
2. Utilizando o método dos mínimos quadrados, calcule o valor de K . \square

Vamos agora a determinar do ganho do tacômetro (K_t). Para tanto, considere a equação (3.9), cujo valor de tensão, em estado permanente, para uma entrada igual ao degrau de amplitude W é dado por:

$$V_t = K_t W. \quad (3.20)$$

A equação (3.20) acima mostra que os passos para a identificação de K_t são idênticos aos seguidos na determinação dos ganhos K_{ta} e K , sendo descritos no seguinte algoritmo.

Algoritmo 3.3

1. Excita-se o motor com tensões constantes e iguais a $V_{a_1}, V_{a_2}, \dots, V_{a_n}$ e para cada um desses valores meça os correspondentes valores de W e V_t , obtendo-se os pares cartesianos $(W_1, V_{t_1}), (W_2, V_{t_2}), \dots, (W_3, V_{t_3})$.
2. Use o método dos mínimos quadrados para ajusta a reta $V_t = K_t W$ aos pontos obtidos no passo anterior (veja figura 3.7). \square

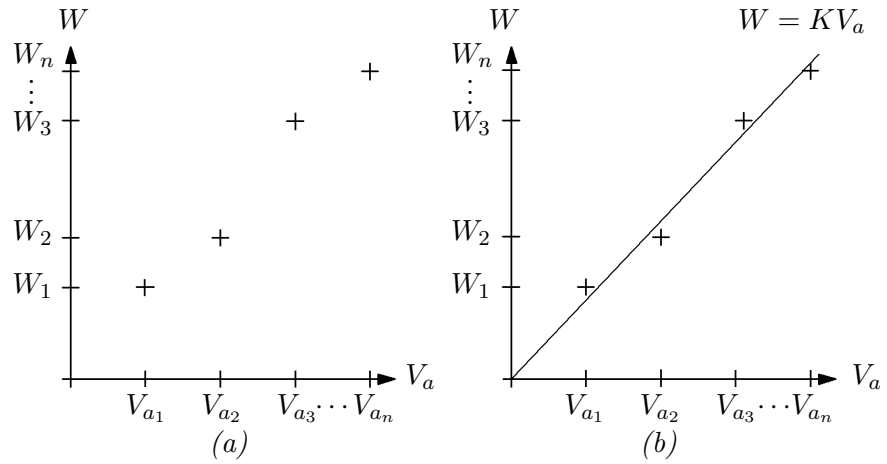


Figura 3.6: Representação (a) cartesiana dos pontos (V_{a_1}, W_1) , (V_{a_2}, W_2) , \dots , (V_{a_n}, W_n) e (b) da reta $W = KV_a$

É importante ressaltar mais uma vez que, embora K_t pudesse ter sido determinado a partir dos ganhos K_{ta} e K pela simples divisão de K_{ta} por K , o procedimento descrito no algoritmo 3.3 nos permite verificar se os valores obtidos para K e K_t são compatíveis.

Finalmente, vamos considerar a identificação de \bar{K} . Para tanto, suponha que seja conectada uma carga resistiva¹ nos terminais do gerador. Neste caso, haverá a circulação de uma corrente contínua de valor I_g que é função da tensão aplicada nos terminais do motor conforme mostra a equação abaixo:

$$V_t(s) = \frac{KK_t V_a}{s(\tau s + 1)} - \frac{\bar{K} K_t I_g}{s(\tau s + 1)} \quad (3.21)$$

que foi obtida a partir da equação (3.10) substituindo-se $V_a(s)$ por V_a/s e $I_g(s)$ por I_g/s

Aplicando-se o teorema do valor final à expressão acima, tem-se que quando $t \rightarrow \infty$:

$$v_t(t) = V_t' = V_t - \bar{K} K_t I_g \quad (3.22)$$

onde $V_t = KK_t V_a$, é o valor da tensão nos terminais do tacômetro para

¹Verifique o que aconteceria se uma carga indutiva ou capacitiva fosse conectada aos terminais do gerador.

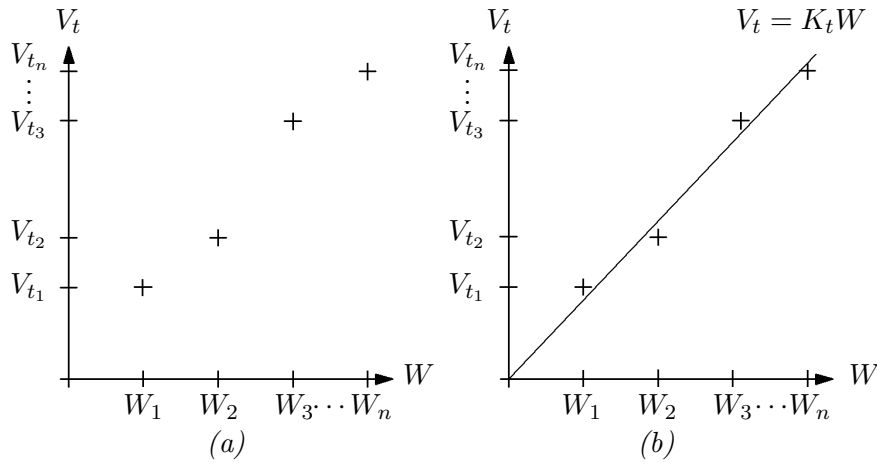


Figura 3.7: Representação (a) cartesiana dos pontos $(W_1, V_{t_1}), (W_2, V_{t_2}), \dots, (W_n, V_{t_n})$ e (b) da reta $V_t = K_t W$

uma entrada igual ao degrau, assumindo-se que não há cargas conectadas ao gerador (conforme equação 3.14).

Definindo

$$\bar{V}_t = V_t - V'_t \quad (3.23)$$

pode-se escrever a equação acima como

$$\bar{V}_t = \bar{K} K_t I_g \quad (3.24)$$

que nos permite desenvolver o seguinte algoritmo para a identificação de \bar{K} :

Algoritmo 3.4

1. Inicialmente, sem carga alguma conectada aos terminais do gerador, aplica-se ao motor CC uma tensão de valor igual a $V_{a_1}(V)$ e mede-se a tensão resultante nos terminais do tacômetro $V'_{t_1}(v)$.

2. Mantendo a mesma tensão aplicada ao motor, conecte uma carga resistiva ao gerador, medindo-se: (i) a corrente que passa a circular por essa resistência ($I_{g_1}(A)$) e (ii) a tensão nos terminais do tacômetro após a inserção da carga ($V'_{t_1}(V)$).

3. Defina $\bar{V}_{t_1} = V_{t_1} - V'_{t_1}$ e forme o par ordenado (I_{g_1}, \bar{V}_{t_1}) .

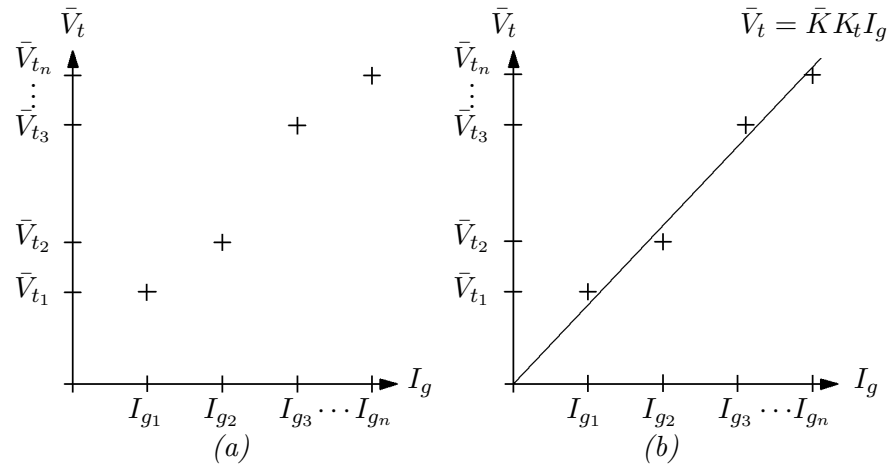


Figura 3.8: Representação (a) cartesiana dos pontos $(I_{g_1}, \bar{V}_{t_1}), (I_{g_2}, \bar{V}_{t_2}), \dots, (I_{g_n}, \bar{V}_{t_n})$ e (b) da reta $\bar{V}_t = \bar{K} \bar{K}_t I_g$

4. Repita os passos 1 a 3 acima para outros valores de V_a , obtendo, ao final, os pares cartesianos $(I_{g_1}, \bar{V}_{t_1}), (I_{g_2}, \bar{V}_{t_2}), \dots, (I_{g_n}, \bar{V}_{t_n})$.

5. Utilize o método dos mínimos quadrados para obter $\bar{K} \bar{K}_t$, conforme mostrado na figura 3.8. \square

Observação 3.1 É interessante notar que, para uma carga fornecida por um resistor R , existe uma relação entre a corrente I_g e a tensão nos terminais do gerador (V_g), dada por:

$$\frac{V_g}{I_g} = R.$$

Como V_g também é proporcional à tensão V_t , pode-se escrever:

$$V_g = K_G V_t,$$

e, finalmente, usando a equação (3.20),

$$I_g = \frac{K_G}{R} V_t = \frac{K_G K_t}{R} W, \quad (3.25)$$

ou seja, a carga resistiva tem o mesmo efeito que o atrito viscoso, representado por f , na equação (3.5). \square

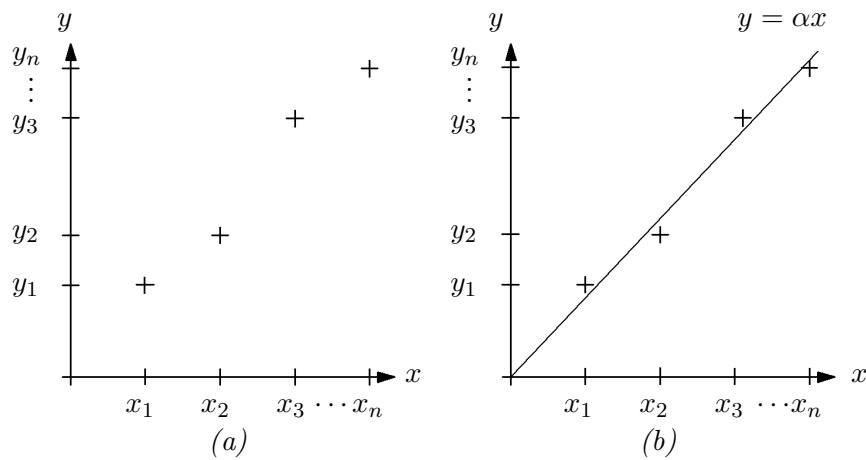


Figura 3.9: Representação (a) cartesiana dos pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ e (b) da reta ajustada $y = \alpha x$

Os passos 2, dos algoritmos 3.1 a 3.3, e 5, do algoritmo 3.4, se resumem a utilizar o método dos mínimos quadrados para o ajuste do coeficiente angular de uma função linear $y = \alpha x$ dado n -pares cartesianos. Este problema pode ser formulado da seguinte forma: suponha que sejam dados n -pares cartesianos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, conforme mostra a figura 3.9(a) e considere o problema de se ajustar o coeficiente angular de uma reta $y = \alpha x$ de tal sorte que a soma dos quadrados das diferenças entre as ordenadas y_1, y_2, \dots, y_n e $\alpha x_1, \alpha x_2, \dots, \alpha x_n$ seja mínima, onde αx_i representa a ordenada do ponto imagem de x_i para a função $y = \alpha x$ (veja figura 3.9(b)).

Inicialmente, note que se os pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ pertencessem à reta $y = \alpha x$ teríamos:

$$\begin{aligned}
 y_1 &= \alpha x_1 \\
 y_2 &= \alpha x_2. \\
 &\vdots \\
 y_n &= \alpha x_n
 \end{aligned}
 \tag{3.26}$$

É fácil verificar que, definindo-se

$$\begin{aligned}\underline{x} &= [x_1 \ x_2 \ \cdots \ x_n]^t \\ \underline{y} &= [y_1 \ y_2 \ \cdots \ y_n]^t\end{aligned}\quad (3.27)$$

pode-se escrever a equação (3.26) na forma vetorial:

$$\underline{y} = \alpha \underline{x} \quad (3.28)$$

onde $\alpha \in \mathbb{R}$ é o coeficiente angular a ser determinado.

Uma análise mais minuciosa do sistema de equações (3.28) revela que quando os pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ pertencem a uma reta, então os vetores \underline{x} e \underline{y} estão em um mesmo espaço vetorial, ou equivalentemente, são colineares. Neste caso, definindo

$$\alpha_i = \frac{y_i}{x_i}, \quad i = 1, 2, \dots, n \quad (3.29)$$

pode-se facilmente concluir que

$$\alpha_i = \alpha_j, \quad i, j = 1, 2, \dots, n, \quad i \neq j \quad (3.30)$$

e, portanto,

$$\alpha = \alpha_i, \quad i = 1, 2, \dots, n \quad (3.31)$$

Contudo, em geral, o sistema de equações (3.28) não tem solução, uma vez que $\underline{y} - \alpha \underline{x} \neq \underline{0}$. Seja, portanto,

$$\underline{e} = \underline{y} - \hat{\underline{y}} \quad (3.32)$$

um vetor representativo do erro entre o vetor \underline{y} e o vetor

$$\hat{\underline{y}} = \alpha \underline{x} = [\alpha x_1 \ \alpha x_2 \ \cdots \ \alpha x_n]^t \quad (3.33)$$

onde α é o coeficiente angular a ser determinado.

O problema dos mínimos quadrados pode ser enunciado como se segue: calcule α tal que

$$\|\underline{e}\|_2^2 = \|\underline{y} - \hat{\underline{y}}\|_2^2 = \|\underline{y} - \alpha \underline{x}\|_2^2 \quad (3.34)$$

seja minimizada, onde $\|\underline{e}\|_2$ representa a norma euclidiana do vetor \underline{e} , qual seja:

$$\|\underline{e}\|_2 = \langle \underline{e}, \underline{e} \rangle^{1/2} = (\underline{e}^t \underline{e})^{1/2} = \left(\sum_{i=1}^n e_i^2 \right)^{1/2} \quad (3.35)$$

onde $\underline{e} = [e_1 \ e_2 \ \dots \ e_n]^t$ e $\langle \cdot, \cdot \rangle$ denota produto escalar. A partir das expressões (3.34) e (3.35) tem-se que

$$\|\underline{e}\|_2^2 = (y_1 - \alpha x_1)^2 + (y_2 - \alpha x_2)^2 + \dots + (y_n - \alpha x_n)^2 \quad (3.36)$$

que mostra que ao se minimizar $\|\underline{e}\|_2^2$ estamos, de fato, minimizando a soma dos quadrados das diferenças entre y_i e αx_i , $i = 1, 2, \dots, n$.

Usando a definição (3.35) na equação (3.34), obtém-se:

$$\begin{aligned} \|\underline{e}\|_2^2 &= \langle \underline{e}, \underline{e} \rangle = \langle \underline{y} - \alpha \underline{x}, \underline{y} - \alpha \underline{x} \rangle \\ &= \underline{y}^t \underline{y} - \alpha \underline{y}^t \underline{x} - \alpha \underline{x}^t \underline{y} + \alpha^2 \underline{x}^t \underline{x} \\ &= \underline{x}^t \underline{x} \alpha^2 - 2 \underline{y}^t \underline{x} \alpha + \underline{y}^t \underline{y}. \end{aligned} \quad (3.37)$$

No desenvolvimento acima foi feito uso do fato que $\underline{y}^t \underline{y}$, $\underline{x}^t \underline{x}$, $\underline{y}^t \underline{x}$, $\underline{x}^t \underline{y} \in \mathbb{R}$ e portanto $\underline{y}^t \underline{x} = (\underline{y}^t \underline{x})^t = \underline{x}^t \underline{y}$. O valor de α pode então ser calculado utilizando-se ferramentas básicas de cálculo diferencial, isto é, calculando-se o valor de α para o qual a função definida pela equação (3.37) atinge o seu mínimo. Como a função (3.37) é quadrática e o coeficiente de α^2 é positivo, a abscissa do seu ponto de mínimo corresponde ao valor de α para o qual a derivada de $\|\underline{e}\|_2^2$ em relação a α se anula, isto é:

$$\frac{d}{d\alpha} \|\underline{e}\|_2^2 = 2 \underline{x}^t \underline{x} \alpha - 2 \underline{y}^t \underline{x} = 0 \quad (3.38)$$

e portanto

$$\alpha = \frac{\underline{y}^t \underline{x}}{\underline{x}^t \underline{x}} = \frac{\underline{y}^t \underline{x}}{\|\underline{x}\|_2^2} \quad (3.39)$$

Em resumo, podemos dizer que dado um conjunto de pontos (x_1, y_1) , (x_2, y_2) , \dots , (x_n, y_n) não alinhados, então o valor de α que minimiza a norma euclidiana do erro entre os vetores

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad e \quad \hat{\underline{y}} = \alpha \underline{x} = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ \vdots \\ \alpha x_n \end{bmatrix} \quad (3.40)$$

é dado por:

$$\alpha = \frac{\underline{y}^t \underline{x}}{\|\underline{x}\|_2^2}. \quad (3.41)$$

3.2.2 Identificação de τ

Fazendo-se $I_g(s) = 0$ na equação (3.10) obtém-se:

$$G(s) = \frac{V_t(s)}{V_a(s)} = \frac{KK_t}{\tau s + 1} \quad (3.42)$$

que representa a função de transferência de um sistema de 1ª ordem com pólo em $1/\tau$. Uma maneira imediata de se obter τ é a partir do traçado das assíntotas da curva de módulo dos diagramas de Bode² determinando-se, portanto, a frequência de canto $\omega = 1/\tau$. Deve ser ressaltado ainda que a assíntota de baixa frequência ($20 \log(KK_t)$) deve ser traçada utilizando-se os valores de KK_t determinados a partir do algoritmo 3.1 da seção 3.2.1.

3.3 O método dos mínimos quadrados no ajuste dos parâmetros de uma função de transferência

Seja

$$Y(s) = G(s)U(s) \quad (3.43)$$

onde

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (3.44)$$

representa a função de transferência de um sistema linear invariante no tempo cujos parâmetros a_i, b_j , $i = 1, 2, \dots, n$, $j = 0, 1, \dots, m$ devem ser determinados. Suponha ainda que os graus dos polinômios do numerador e do denominador (m e n respectivamente) ou sejam conhecidos ou serão arbitrados pelo engenheiro de controle. Por exemplo, a função de transferência do grupo motor-gerador que estamos considerando, de acordo com a notação da equação (3.44), é dada por:

$$G(s) = \frac{KK_t/\tau}{s + 1/\tau} \quad (3.45)$$

e portanto $m = 0$, $n = 1$, $b_0 = KK_t/\tau$ e $a_1 = 1/\tau$. Caso a modelagem do sistema não tivesse sido realizada, os valores de m e n poderiam ser arbitrados

²O aluno é aconselhado a realizar uma revisão sobre diagramas de Bode (Basilio, 1995) com vistas a ter a base necessária para justificar formalmente as conclusões aqui apresentadas.

a partir de um compromisso entre a exatidão do ajuste e a complexidade do modelo obtido³.

Sabe-se que a resposta em estado permanente ($y_{ss}(t)$) a uma entrada senoidal $u(t) = U_{m_0} \text{Sen } \omega_0 t$ aplicada ao sistema cuja função de transferência é aquela da equação (3.43) é:

$$y_{ss}(t) = Y_{m_0} \text{Sen } (\omega_0 t + \phi_0) \quad (3.46)$$

onde $Y_{m_0} = |G(j\omega_0)|U_{m_0}$ e ϕ_0 é a defasagem entre as senoides de entrada e de saída. Portanto, para uma dada frequência ω_0

$$G(j\omega_0) = |G(j\omega_0)|e^{j\phi_0} \quad (3.47)$$

onde $|G(j\omega_0)| = Y_{m_0}/U_{m_0}$ e ϕ_0 é obtido medindo-se a defasagem entre os sinais de entrada e de saída.

Substituindo $s = j\omega_0$ na equação (3.44), obtém-se:

$$G(j\omega_0) = \frac{(j\omega_0)^m b_0 + (j\omega_0)^{m-1} b_1 + \dots + (j\omega_0) b_{m-1} + b_m}{(j\omega_0)^n + (j\omega_0)^{n-1} a_1 + \dots + (j\omega_0) a_{n-1} + a_n} \quad (3.48)$$

onde, de acordo com a equação (3.47), $G(j\omega_0) = |G(j\omega_0)|e^{j\phi_0}$. Multiplicando-se ambos os membros da equação (3.48) por $(j\omega_0)^n + (j\omega_0)^{n-1} a_1 + \dots + (j\omega_0) a_{n-1} + a_n$ resulta:

$$G(j\omega_0)(j\omega_0)^n + \sum_{i=1}^n G(j\omega_0)(j\omega_0)^{n-i} a_i = \sum_{i=0}^m (j\omega_0)^{m-i} b_i \quad (3.49)$$

que é equivalente a:

$$\sum_{i=1}^n G(j\omega_0)(j\omega_0)^{n-i} a_i - \sum_{i=0}^m (j\omega_0)^{m-i} b_i = -G(j\omega_0)(j\omega_0)^n \quad (3.50)$$

Definindo

$$\underline{x}^t = [a_1 \quad a_2 \quad \dots \quad a_n \quad b_0 \quad b_1 \quad \dots \quad b_n] \quad (3.51)$$

pode-se escrever:

³Pode-se mostrar que quanto maiores os valores de m e n , melhor o ajuste da curva de resposta em frequência do modelo aos pontos obtidos experimentalmente. Porém, valores grandes de n dão origem a modelos de ordem elevada, o que torna difícil o projeto do controlador.

$$[G(j\omega_0)(j\omega_0)^{n-1} \dots G(j\omega_0)(j\omega_0) G(j\omega_0) - (j\omega_0)^m \dots - (j\omega_0) - 1] \underline{x} = G(j\omega_0)(j\omega_0)^n \quad (3.52)$$

Repetindo-se também para as frequências ω_i , $i = 1, 2, \dots, q$, obtém-se a seguinte equação matricial:

$$M \underline{x} = \underline{y} \quad (3.53)$$

onde $M = [M_1 \quad M_2]$,

$$M_1 = \begin{bmatrix} G(j\omega_0)(j\omega_0)^{n-1} & \dots & G(j\omega_0)(j\omega_0) & G(j\omega_0) \\ G(j\omega_1)(j\omega_1)^{n-1} & \dots & G(j\omega_1)(j\omega_1) & G(j\omega_1) \\ \vdots & & \vdots & \vdots \\ G(j\omega_q)(j\omega_q)^{n-1} & \dots & G(j\omega_q)(j\omega_q) & G(j\omega_q) \end{bmatrix}, \quad (3.54)$$

$$M_2 = \begin{bmatrix} -(j\omega_0)^m & \dots & -(j\omega_0) & -1 \\ -(j\omega_1)^m & \dots & -(j\omega_1) & -1 \\ \vdots & & \vdots & \vdots \\ -(j\omega_q)^m & \dots & -(j\omega_q) & -1 \end{bmatrix}, \quad (3.55)$$

$$\underline{y} = \begin{bmatrix} -G(j\omega_0)(j\omega_0)^n \\ -G(j\omega_1)(j\omega_1)^n \\ \dots \\ -G(j\omega_q)(j\omega_q)^n \end{bmatrix}, \quad (3.56)$$

$M \in \mathbb{C}^{(q+1) \times (m+n+1)}$ e $\underline{y} \in \mathbb{C}^{q+1}$. Uma vez que a matriz M e o vetor \underline{y} são complexos, podem ser escritos como:

$$M = Re(M) + jIm(M) \quad \text{e} \quad \underline{y} = Re(\underline{y}) + jIm(\underline{y}) \quad (3.57)$$

e, portanto, a equação (3.56) é idêntica a:

$$[Re(M) + jIm(M)] \underline{x} = Re(\underline{y}) + jIm(\underline{y}). \quad (3.58)$$

Comparando as partes real e imaginária, resulta:

$$\begin{aligned} Re(M) \underline{x} &= Re(\underline{y}) \\ Im(M) \underline{x} &= Im(\underline{y}) \end{aligned} \quad (3.59)$$

que é equivalente a:

$$A\underline{x} = \underline{b} \quad (3.60)$$

onde

$$A = \begin{bmatrix} \operatorname{Re}(M) \\ \operatorname{Im}(M) \end{bmatrix} \quad \text{e} \quad \underline{b} = \begin{bmatrix} \operatorname{Re}(\underline{y}) \\ \operatorname{Im}(\underline{y}) \end{bmatrix}, \quad (3.61)$$

$A \in \mathbb{R}^{2(q+1) \times (m+n+1)}$, $\underline{b} \in \mathbb{R}^{2(q+1)}$ e $\underline{x} \in \mathbb{R}^{m+n+1}$.

Antes de se propor uma solução para a equação (3.60) é oportuno fazer uma análise das dimensões das matrizes envolvidas. Para tanto, lembre-se de que $q + 1$ representa o número de freqüências usadas no experimento para a obtenção da resposta em freqüência do sistema e m e n denotam, respectivamente, os graus dos polinômios do numerador e denominador de $G(s)$. Para um melhor ajuste da curva de resposta em freqüência do modelo aos valores obtidos experimentalmente, q deve ser grande o suficiente para varrer todas as freqüências de interesse. Em contrapartida, m e n não devem ser demasiadamente grandes pois aumentam significativamente a complexidade do modelo. Portanto, na prática, $q \gg m + n$, o que implica que a matriz A possui, em geral, um número de linhas muito maior que o número de colunas. Observe ainda que, para o sistema de equações (3.60) ter solução é necessário que o vetor \underline{b} possa ser escrito como uma combinação linear das colunas de A . O fato da matriz A ter um número de linhas muito maior que o número de colunas torna essa possibilidade bastante remota, o que implica que, geralmente $A\underline{x} \neq \underline{b}$.

Seja, portanto,

$$\hat{\underline{b}} = A\underline{x} \quad (3.62)$$

e defina

$$\underline{e} = \hat{\underline{b}} - \underline{b}. \quad (3.63)$$

Considere o seguinte problema: encontre \underline{x} tal que $\|\underline{e}\|_2^2$ seja mínima. Note que, se existir \underline{x} tal que $A\underline{x} = \underline{b}$ então $\|\underline{e}\|_2^2 = 0$, o que implica que mesmo no caso em que o sistema de equações (3.60) tem solução, o método dos mínimos quadrados também levará a essa solução.

A partir das equações (3.62) e (3.63) tem-se:

$$\begin{aligned} \|\underline{e}\|_2^2 &= \langle A\underline{x} - \underline{b}, A\underline{x} - \underline{b} \rangle \\ &= (A\underline{x} - \underline{b})^t (A\underline{x} - \underline{b}) \\ &= \underline{x}^t A^t A \underline{x} - \underline{x}^t A^t \underline{b} - \underline{b}^t A \underline{x} + \underline{b}^t \underline{b} \end{aligned} \quad (3.64)$$

e como $\underline{x}^t A^t \underline{b} \in \mathbb{R}$ então $\underline{x}^t A^t \underline{b} = (\underline{x}^t A^t \underline{b})^t = \underline{b}^t A \underline{x}$, o que nos permite escrever:

$$\|\underline{e}\|_2^2 = \underline{x}^t A^t A \underline{x} - 2 \underline{b}^t A \underline{x} + \underline{b}^t \underline{b} \quad (3.65)$$

Antes de abordar o problema de minimização proposto em (3.65), alguns conceitos devem ser lembrados:

1. Seja $y = f(x_1, x_2, \dots, x_n)$ uma função de \mathbb{R}^n em \mathbb{R} . A derivada de y em relação ao vetor \underline{x} é definida como:

$$\frac{\partial y}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} \quad (3.66)$$

2. De acordo com a notação introduzida na equação (3.66), tem-se que:

(a) Se $y = \underline{b}^t \underline{x}$, onde $\underline{b} \in \mathbb{R}^n$ então

$$\frac{\partial y}{\partial \underline{x}} = \underline{b} \quad (3.67)$$

Prova: Seja $\underline{b}^t = [b_1 \ b_2 \ \dots \ b_n]$. Então $y = \underline{b}^t \underline{x} = b_1 x_1 + b_2 x_2 + \dots + b_n x_n$, e portanto:

$$\frac{\partial y}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \underline{b}. \quad (3.68)$$

(b) Se $y = \underline{x}^t H \underline{x}$, onde H é uma matriz simétrica, isto é $H^t = H$, então

$$\frac{\partial y}{\partial \underline{x}} = 2H \underline{x} \quad (3.69)$$

Prova: Por indução finita sobre n .

(i) $n = 2$. Neste caso

$$y = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = h_{11}x_1^2 + 2h_{12}x_1x_2 + h_{22}x_2^2 \quad (3.70)$$

Portanto:

$$\begin{aligned} \frac{\partial y}{\partial \underline{x}} &= \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2h_{11}x_1 + 2h_{12}x_2 \\ 2h_{12}x_1 + 2h_{22}x_2 \end{bmatrix} \\ &= 2 \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2H\underline{x} \end{aligned} \quad (3.71)$$

(ii) Assuma que a expressão (3.69) seja válida quando $H \in \mathbb{R}^{n \times n}$, isto é, se $y = \underline{x}^t H \underline{x}$ então $\partial y / \partial \underline{x} = 2H\underline{x}$.

(iii) Seja agora $H \in \mathbb{R}^{(n+1) \times (n+1)}$, isto é:

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} & h_{1,n+1} \\ h_{12} & h_{22} & \cdots & h_{2n} & h_{2,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ h_{1n} & h_{2,n} & \cdots & h_{nn} & h_{n,n+1} \\ h_{1,n+1} & h_{2,n+1} & \cdots & h_{n,n+1} & h_{n+1,n+1} \end{bmatrix}. \quad (3.72)$$

Definindo

$$\begin{aligned} H_n &= \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{12} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & & \vdots \\ h_{1n} & h_{2,n} & \cdots & h_{nn} \end{bmatrix} \\ \underline{h}_{n+1}^t &= [h_{1,n+1} \quad h_{2,n+1} \quad \cdots \quad h_{n,n+1}] \\ \underline{x}_n &= [x_1 \quad x_2 \quad \cdots \quad x_n] \end{aligned} \quad (3.73)$$

então $y = \underline{x}^t H \underline{x}$ é equivalente a:

$$\begin{aligned} y &= [\underline{x}_n^t \quad x_{n+1}] \begin{bmatrix} H_n & \underline{h}_{n+1} \\ \underline{h}_{n+1}^t & h_{n+1,n+1} \end{bmatrix} \begin{bmatrix} \underline{x}_n \\ x_{n+1} \end{bmatrix} \\ &= [\underline{x}_n^t H_n + x_{n+1} \underline{h}_{n+1}^t \quad \underline{x}_n^t \underline{h}_{n+1} + x_{n+1} h_{n+1,n+1}] \begin{bmatrix} \underline{x}_n \\ x_{n+1} \end{bmatrix} \\ &= \underline{x}_n^t H_n \underline{x}_n + x_{n+1} \underline{h}_{n+1}^t \underline{x}_n + \underline{x}_n^t \underline{h}_{n+1} x_{n+1} + x_{n+1}^2 h_{n+1,n+1} \\ &= \underline{x}_n^t H_n \underline{x}_n + 2x_{n+1} \underline{h}_{n+1}^t \underline{x}_n + x_{n+1}^2 h_{n+1,n+1} \end{aligned} \quad (3.74)$$

onde para a obtenção da última expressão fez-se uso do fato de que como $\underline{x}_n^t \underline{h}_{n+1} \in \mathbb{R}$ então $\underline{x}_n^t \underline{h}_{n+1} = (\underline{x}_n^t \underline{h}_{n+1})^t = \underline{h}_{n+1}^t \underline{x}_n$.

Desta forma, $\partial y/\partial \underline{x}$ será:

$$\begin{aligned} \frac{\partial y}{\partial \underline{x}} &= \begin{bmatrix} \frac{\partial y}{\partial x_n} \\ \frac{\partial y}{\partial x_{n+1}} \end{bmatrix} = \begin{bmatrix} 2H_n x_n + 2x_{n+1} h_{n+1} \\ 2h_{n+1}^t x_n + 2h_{n+1,n+1} x_{n+1} \end{bmatrix} \\ &= 2 \begin{bmatrix} H_n & h_{n+1} \\ h_{n+1}^t & h_{n+1,n+1} \end{bmatrix} \begin{bmatrix} x_n \\ x_{n+1} \end{bmatrix} = 2H \underline{x} \end{aligned} \quad (3.75)$$

□

Estamos, agora, em condições de considerar o problema de encontrar $\underline{x} \in \mathbb{R}^{m+n+1}$ que minimiza $\|\underline{e}\|_2^2$ expressa pela equação (3.65). A partir dos resultados 2(a) e 2(b), podemos escrever:

$$\frac{\partial}{\partial \underline{x}} \|\underline{e}\|_2^2 = 2(A^t A) \underline{x} - 2A^t \underline{b} \quad (3.76)$$

e, portanto, o valor de \underline{x} que minimiza $\|\underline{e}\|_2^2$ será tal que

$$2(A^t A) \underline{x} - 2A^t \underline{b} = \underline{0}. \quad (3.77)$$

Observe que, como, em geral, $m + n + 1 \ll 2(q + 1)$, o posto de A é, geralmente, igual a $m + n + 1$ e, portanto, $A^t A$ é não singular, sendo desta forma inversível. Conseqüentemente, o vetor \underline{x} que minimiza $\|\underline{e}\|_2^2$ será:

$$\underline{x} = (A^t A)^{-1} A^t \underline{b}. \quad (3.78)$$

O desenvolvimento acima nos permite escrever o seguinte algoritmo para a identificação dos parâmetros da função de transferência $G(s) = b(s)/a(s)$, onde $b(s) = b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m$ e $a(s) = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n$:

Algoritmo 3.5

1. Arbitre valores para m e n e forme o vetor

$$\underline{x} = [a_1 \quad a_2 \quad \dots \quad a_n \quad b_0 \quad b_1 \quad \dots \quad b_{m-1} \quad b_m]^t$$

2. Excite o sistema com funções senoidais $u(t) = U_{m_i} \text{Sen}(\omega_i t)$, $i = 0, 1, \dots, q$, medindo-se as amplitudes das respostas correspondentes (Y_{m_i}) e as defasagens em relação às respectivas senoides de entrada (ϕ_i) e forme

$$G(j\omega_i) = \frac{Y_{m_i}}{U_{m_i}} e^{j\phi_i}, \quad i = 1, 2, \dots, n$$

3. Construa as matrizes $M = [M_1 \quad M_2]$ e A ,

$$M_1 = \begin{bmatrix} G(j\omega_0)(j\omega_0)^{n-1} & \dots & G(j\omega_0)(j\omega_0) & G(j\omega_0) \\ G(j\omega_1)(j\omega_1)^{n-1} & \dots & G(j\omega_1)(j\omega_1) & G(j\omega_1) \\ \vdots & & \vdots & \vdots \\ G(j\omega_q)(j\omega_q)^{n-1} & \dots & G(j\omega_q)(j\omega_q) & G(j\omega_q) \end{bmatrix},$$

$$M_2 = \begin{bmatrix} -(j\omega_0)^m & \dots & -(j\omega_0) & -1 \\ -(j\omega_1)^m & \dots & -(j\omega_1) & -1 \\ \vdots & & \vdots & \vdots \\ -(j\omega_q)^m & \dots & -(j\omega_q) & -1 \end{bmatrix}$$

e

$$A = \begin{bmatrix} Re(M) \\ Im(M) \end{bmatrix},$$

e os vetores \underline{y} e \underline{b} ,

$$\underline{y} = \begin{bmatrix} -G(j\omega_0)(j\omega_0)^n \\ -G(j\omega_1)(j\omega_1)^n \\ \dots \\ -G(j\omega_q)(j\omega_q)^n \end{bmatrix}$$

e

$$\underline{b} = \begin{bmatrix} Re(\underline{y}) \\ Im(\underline{y}) \end{bmatrix}.$$

4. Calcule $\underline{x} = (A^t A)^{-1} A^t \underline{b}$. Observe que os coeficientes de $a(s)$ correspondem às n primeiras componentes de \underline{x} e os coeficientes de $b(s)$ são obtidos tomando-se as $m + 1$ últimas componentes de \underline{x} . \square

3.4 Experimento para identificação da função de transferência do grupo motor-gerador

Uma vez que já dispomos do modelo matemático do sistema, podemos passar agora ao laboratório para realizarmos os ensaios necessários à identificação da função de transferência do grupo motor-gerador. Este experimento é composto, basicamente de duas partes: (i) excitação do motor CC com tensões constantes com vistas à identificação dos ganhos K , K_t e \bar{K} ; (ii) excitação do sistema com sinais senoidais de diferentes frequências objetivando-se obter a

resposta em frequência do sistema para que, a partir da curva de módulo dos diagramas de Bode e, também utilizando-se o diagrama polar, identificar τ .

No experimento para a identificação da função de transferência do grupo motor-gerador, serão utilizados os seguintes equipamentos:

1. Osciloscópio digital com, pelo menos, dois canais;
2. Gerador de funções;
3. Fonte de tensão CC (regulável);
4. Tacômetro óptico;
5. Multímetros digitais (3);
6. Amplificador de potência;
7. Resistência de $0,5\Omega$

3.4.1 Experimento para determinação de K , \bar{K} e K_t

São os seguintes os passos necessários para a obtenção dos dados que levam à determinação dos parâmetros K , \bar{K} e K_t :

1. Conecte as saídas da fonte de tensão CC aos terminais do motor.
2. Conecte um multímetro (funcionando como voltímetro) à saída da fonte de tensão e um outro multímetro (funcionando também como voltímetro) aos terminais do tacômetro.
3. Conecte um dos terminais do multímetro (funcionando como amperímetro) a um dos terminais da carga (resistência de $0,5\Omega$) o outro terminal da resistência a um dos terminais do gerador. Note que para que o circuito seja fechado, o terminal de entrada do multímetro deverá ser conectado ao outro terminal do gerador. Não o faça agora. Isto será feito mais adiante.
4. Excite o motor com o primeiro valor de tensão (V_a) sugerido na primeira coluna da tabela 3.4.1, anotando na segunda coluna o valor da tensão efetivamente aplicada. Em seguida, meça a tensão nos terminais do tacômetro (V_t) e a velocidade angular do motor (ω), utilizando um tacômetro óptico, preenchendo, respectivamente, as colunas 3 e 4 da tabela 3.4.1.
5. Mantendo a mesma tensão V_a , feche o circuito formado pelo gerador, multímetro e pela resistência de 10Ω , medindo a corrente que circula na carga (I_g) e o novo valor da tensão nos terminais do tacômetro (V_t'), anotando-os, respectivamente nas colunas 5 e 6 da tabela 3.4.1. Deve-se tomar o cuidado de certificar de que não houve alteração no valor de V_a após a introdução da carga no circuito. Tendo a leitura sido feita, abra o circuito gerador-multímetro-resistência.

V_a (V) sugerida	V_a (V) medida	V_t (V)	ω (rpm)	I_g (mA)	V_t' (V)
6,0					
7,5					
9,0					
10,5					
12,0					

Tabela 3.1: Experimento para determinação dos parâmetros K , K_t e \bar{K}

6. Repita os passos 4 e 5 para os demais valores de tensão sugeridos na tabela 3.4.1.

3.4.2 Experimento para obtenção da resposta em frequência

Para a obtenção da curva de resposta em frequência do sistema é necessário excitar o motor com uma tensão senoidal. Os geradores de funções de que o laboratório dispõe não têm potência suficiente para fazer o motor funcionar. Para superar este problema, faz-se passar o sinal fornecido pelo gerador de funções por um amplificador de potência, que fornecerá, então, a potência necessária para o motor girar. Com isso em mente, siga os passos seguintes para obter os dados necessários para o levantamento da curva de resposta em frequência do sistema:

1. Conecte os terminais do gerador de funções aos terminais de entrada do amplificador de potência. Em seguida conecte os terminais de saída do amplificador de potência aos terminais do motor.

2. Conecte os terminais de saída do amplificador de potência ao canal 1 do digitalizador e os terminais do tacômetro ao canal 2 do digitalizador.

3. Excite o motor com uma tensão senoidal de aproximadamente 10V de amplitude, variando a frequência da tensão senoidal fornecida pelo gerador de funções de forma a excitar o motor com valores de frequência próximos daqueles sugeridos na primeira coluna da tabela 3.2. Em seguida, complete as demais colunas da tabela.

3.5 Validação do modelo

Tendo sido determinados os parâmetros da função de transferência do sistema, o passo seguinte é a validação do modelo obtido, *i.e.*, verificar se os

f (Hz) sugerida	f (Hz) medida	V_a (V)	V_t (V)	V_t/V_a	ϕ (rad)
0,10					
0,20					
0,30					
0,40					
0,50					
0,65					
0,85					
1,10					
1,50					
1,85					
2,40					
3,10					
4,10					
5,30					
6,90					
9,50					
11,8					
13,5					
15,3					
18,0					
20,0					

Tabela 3.2: Tabela para levantamento da resposta em frequência do sistema motor-gerador-tacômetro

valores calculados a partir dos experimentos de identificação são tais que o modelo matemático adotado represente, com precisão aceitável, o sistema real. No caso do grupo motor-gerador (sistema adotado para este laboratório) foram realizados ensaios para se determinar os parâmetros K , K_t , \bar{K} e τ . Note que:

1. Os ganhos K e K_t foram identificados a partir do ajuste dos coeficientes angulares de retas que passam pela origem, de acordo com os algoritmos 3.2 e 3.3. A exatidão dos valores encontrados para K e K_t pôde ser verificada a partir da comparação com o produto KK_t , identificado de acordo com o algoritmo 3.1.

2. O valor de \bar{K} , por sua vez, foi identificado de uma maneira apenas (algoritmo 3.4). Isto requer, então, que sejam realizadas algumas simulações em computador e comparar o resultado destas com o desempenho do sistema real.

3. A constante de tempo τ foi calculada de duas formas: (i) a partir da determinação da frequência de canto do diagrama de módulo de Bode e (ii) a partir do ajuste do diagrama polar de $KK_t/(\tau s + 1)$ a um conjunto de pontos obtidos experimentalmente. O valor de τ será escolhido a partir de simulações em computador, comparando-se os resultados dessas com o desempenho do sistema real.

Assim sendo, para a validação do modelo matemático do grupo motor-gerador, dado na figura 3.4 e representado novamente na figura 3.9, proceda da seguinte forma:

1 Encontre o erro percentual ($E_{KK_t}(\%)$) entre os produtos de K e K_t , obtidos a partir dos algoritmos 3.2 e 3.3 e de K_{ta} , obtido de acordo com o algoritmo 3.1, isto é:

$$E_{KK_t}(\%) = \frac{KK_t - K_{ta}}{KK_t} 100(\%) \quad (3.79)$$

Se o erro for menor que 1%, os valores de K e K_t obtidos experimentalmente podem ser adotados como representativos do modelo, com razoável grau de confiança. Caso o erro seja maior que 1%, deve-se, inicialmente, verificar se os resultados obtidos nos ensaios de laboratório foram de fato utilizados no cálculo de K , K_t e KK_t . Isto permitirá que se encontre possíveis discrepâncias entre os valores obtidos pelos três métodos, evitando que todos os experimentos realizados para a determinação de K , K_t e K_{ta} tenham que ser realizados novamente. Caso não tenha sido encontrado qualquer erro de manipulação dos dados, o aluno deve retornar ao laboratório para realizar novamente aqueles experimentos cujos valores obtidos inicialmente apresentam maiores discrepâncias.

2 Excite o grupo motor-gerador com um pulso de tensão de amplitude igual a 10V com frequência de aproximadamente 0,3Hz e faça a aquisição dos sinais⁴ de entrada ($v_a(t)$) e de saída ($v_t(t)$) e dos correspondentes instantes de tempo (t).

3 Excite novamente o grupo motor-gerador com um pulso de tensão de amplitude igual a 10V com frequência de aproximadamente 0,3Hz. No meio

⁴Esta aquisição de dados pode ser feita utilizando-se um computador com placas de aquisição de dados ou um osciloscópio digital com capacidade de armazenamento em disco.

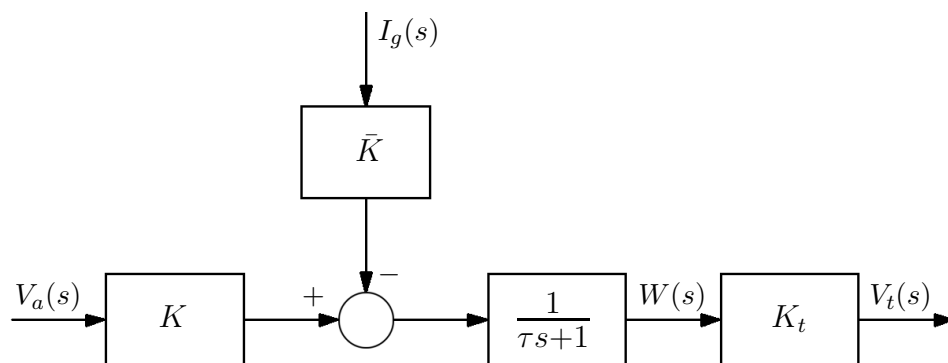


Figura 3.10: Diagrama de blocos representativo do modelo matemático do grupo motor-gerador-tacômetro

do pulso, conecte a resistência de $0,5\Omega$ aos terminais do motor e meça a corrente de carga. Em seguida, faça, como no item anterior, a aquisição dos sinais de entrada ($v_a(t)$ e $i_g(t)$) e de saída ($v_t(t)$).

4(a) Com os valores de K e K_t , obtidos no passo 1, e com o valor de τ , obtido a partir do diagrama de módulo de Bode, construa um modelo em SIMULINK equivalente ao diagrama de blocos representativo do grupo motor-gerador dado pela figura 3.10 no qual o sinal de entrada seá $t \times v_a(t)$ (obtidos no passo 2). Para tanto, crie dois vetores colunas \mathbf{t} e \mathbf{va} , cujas componentes são as correspondentes abscissas e ordenadas do sinal de entrada e utilize o bloco `from workspace` como entrada, sendo \mathbf{t} e \mathbf{va} os parâmetros. Realize uma simulação tendo como instantes inicial e final, $\mathbf{t}(1)$ e $\mathbf{t}(\text{length}(\mathbf{t}))$, respectivamente. Note que, após a simulação serão gerados dois vetores: \mathbf{ts} , que corresponde aos instantes de tempo utilizados na simulação e \mathbf{vts} , que corresponde à tensão nos terminais do tacômetro para o modelo obtido. Represente, em um mesmo gráfico, as curvas $t \times v_a(t)$, $t \times v_t(t)$ e $t_s \times v_t(t_s)$, onde t_s é um intervalo cujos extremos são $\mathbf{ts}(1)$ e $\mathbf{ts}(\text{length}(\mathbf{ts}))$.

4(b) Proceda de forma idêntica à anterior, porém com o valor de τ obtido a partir do ajuste do diagrama polar.

4(c) Entre as curvas $t_s \times v_t(t_s)$ (representadas graficamente nos item 4(a) e 4(b)), escolha aquela que mais se aproxima da resposta do sistema real $t \times v_t(t)$. O valor de τ que corresponde ao modelo que produziu a curva mais

próxima será o escolhido.

5 A partir dos dados obtidos ao final do passo 3, crie agora quatro vetores colunas, \mathbf{t} , \mathbf{va} , \mathbf{ig} e \mathbf{vt} correspondente aos dados armazenados para t , $v_a(t)$, $i_g(t)$ e $v_t(t)$, respectivamente. Em seguida, utilizando o mesmo modelo construído no item 4(a), excite o sistema com blocos `from workspace` cujos parâmetros serão: (i) \mathbf{t} e \mathbf{va} (entrada de referência); (ii) \mathbf{t} e \mathbf{ig} (perturbação). Represente, em um mesmo gráfico, as curvas $t \times v_a(t)$, $t \times i_g(t)$, $t \times v_t(t)$ e $t_s \times v_t(t_s)$, onde t_s é um intervalo cujos extremos são $\mathbf{ts}(1)$ e $\mathbf{ts}(\text{length}(\mathbf{ts}))$. Ao final deste passo ficará claro se há algum erro em \bar{K} .

É de se esperar que as curvas obtidas no trabalho de simulação não sejam exatamente coincidentes com as curvas obtidas para o sistema real. Porém, para que o modelo adotado possa ser considerado satisfatório, essas curvas devem ser bastante próximas. Qualquer discrepância mais acentuada pode desqualificar o experimento de identificação ou até mesmo o modelo matemático; por exemplo, levando à necessidade de escolha de um modelo não-linear.

Capítulo 4

Projeto do controlador de velocidade

Obtido um modelo matemático para o sistema, o passo seguinte é projetar um controlador que satisfaça as seguintes exigências: *(i)* estabilidade; *(ii)* rastreamento assintótico de um sinal de referência (*i.e.* para uma dada velocidade, o motor deve, em estado permanente, girar nessa velocidade); *(iii)* rejeição assintótica de perturbações (*i.e.* quando uma carga for inserida nos terminais do gerador, o motor, após um ‘pequeno’ transitório, deve permanecer com a mesma rotação de antes da introdução da carga); *(iv)* o sistema compensado não deve ter o seu desempenho comprometido por eventuais erros no modelo, cometidos durante a fase de identificação, o que significa dizer que o sistema tem elevada robustez ou equivalentemente pouca sensibilidade a variações nos parâmetros da função de transferência da planta) e, finalmente, *(v)* bom desempenho transitório.

O objetivo deste capítulo é desenvolver o projeto de um controlador de velocidade para o grupo motor-gerador que satisfaça as condições *(i)* a *(v)* acima. Como finalidade didática, será, inicialmente, considerada a possibilidade de se fazer a compensação em malha aberta para, em seguida, projetar um sistema de controle realimentado. Esta abordagem terá a finalidade de ilustrar os benefícios da realimentação num sistema de controle.

Este capítulo está estruturado da seguinte forma: na seção 4.1 serão apresentados os fundamentos teóricos necessários para o projeto de um controlador que rastreie assintoticamente um sinal de referência, $R(s)$, e rejeite (também assintoticamente) uma perturbação, $D(s)$, conforme a figura 4.1.

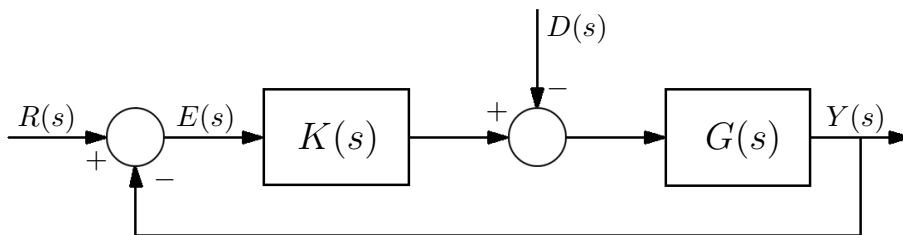


Figura 4.1: Diagrama de blocos para o projeto de controladores com objetivos de rastreamento assintótico de um sinal referência e rejeição assintótica de perturbação

Outro tópico a ser abordado na seção 4.1 é a chamada sensibilidade de um sistema em relação à variação de parâmetros da função de transferência da planta. Na seção 4.2 será considerado o projeto de um controlador que satisfaça as exigências (i) a (v). Vale lembrar que um bom domínio do método do lugar das raízes é fundamental para o projeto de compensadores, principalmente no que se refere à melhoria da resposta transitória. Outros conceitos também importantes são aqueles referentes à resposta transitória de um sistema de segunda ordem. Caso esses assuntos não estejam bem sedimentados, os alunos são aconselhados a revisá-los.

4.1 Fundamentos teóricos

4.1.1 Rastreamento e rejeição assintótica de sinais de dinâmica conhecida

Considere o sistema realimentado da figura 4.1 onde, $G(s)$ e $K(s)$ representam as funções de transferências da planta e do controlador, respectivamente, sendo

$$G(s) = \frac{n_G(s)}{d_G(s)} \text{ e } K(s) = \frac{n_K(s)}{d_K(s)}, \quad (4.1)$$

$n_G(s)$ e $d_G(s)$ são polinômios conhecidos e $n_K(s)$ e $d_K(s)$ são polinômios a serem determinados. Sejam $R(s)$, $D(s)$ e $Y(s)$ as transformadas de Laplace

dos sinais de referência, de perturbação externa e de saída, onde

$$R(s) = \frac{\alpha(s)}{\beta(s)} \text{ e } D(s) = \frac{\gamma(s)}{\delta(s)}, \quad (4.2)$$

$\beta(s)$ e $\delta(s)$ são polinômios conhecidos (dinâmicas de $R(s)$ e $D(s)$, respectivamente), $\alpha(s)$ e $\gamma(s)$ são também polinômios, porém, como será visto mais adiante, não são necessariamente conhecidos.

A partir do diagrama de blocos da figura 4.1, pode-se escrever:

$$Y(s) = -G(s)D(s) + G(s)K(s)R(s) - G(s)K(s)Y(s) \quad (4.3)$$

e, conseqüentemente:

$$Y(s) = \frac{G(s)K(s)}{1 + G(s)K(s)}R(s) - \frac{G(s)}{1 + G(s)K(s)}D(s). \quad (4.4)$$

A partir da equação (4.4), pode-se observar que o sinal de saída $Y(s)$ possui duas componentes: (i) $Y_R(s)$, que é devida ao sinal de referência $R(s)$ e (ii) $Y_D(s)$, que se deve à perturbação externa $D(s)$. Portanto, $Y(s)$ pode ser escrito como:

$$Y(s) = Y_R(s) - Y_D(s) \quad (4.5)$$

onde

$$Y_R(s) = \frac{G(s)K(s)}{1 + G(s)K(s)}R(s) \text{ e } Y_D(s) = \frac{G(s)}{1 + G(s)K(s)}D(s) \quad (4.6)$$

O rastreamento e a rejeição de perturbações externas (assintoticamente) requerem que $y(t) \rightarrow r(t)$ quando $t \rightarrow \infty$, o que equivale a exigir que

$$\lim_{t \rightarrow \infty} y_D(t) = 0 \quad (4.7)$$

e

$$\lim_{t \rightarrow \infty} e_R(t) = 0 \quad (4.8)$$

onde

$$e_R(t) = y_R(t) - r(t). \quad (4.9)$$

Os problemas do rastreamento assintótico de um sinal de referência e da rejeição assintótica de um sinal externo de perturbação (ambos de dinâmicas conhecidas) serão abordados considerando-se as condições impostas pelas equações (4.7)–(4.9), acima. Antes, porém, considere a seguinte fatoração:

seja $p(s)$ um polinômio e fatore $p(s)$ como $p(s) = p^-(s)p^+(s)$, onde $p^-(s)$ é um polinômio de Hurwitz cujos zeros são os zeros de $p(s)$ com parte real negativa e $p^+(s)$ é um polinômio cujos zeros são os zeros de $p(s)$ com parte real positiva ou nula. Por exemplo, o polinômio $p(s) = s^3 - 2s^2 - 3s$ pode ser fatorado como $p^-(s)p^+(s)$, onde $p^-(s) = s + 1$ e $p^+(s) = s(s - 3)$.

Condições necessárias e suficientes para que o sistema da figura 4.1 rejeite assintoticamente um sinal $D(s) = \gamma(s)/\beta(s)$, $\beta(s)$ conhecido, são apresentadas no seguinte teorema.

Teorema 4.1 Seja $K(s)$ um controlador que estabiliza $G(s)$, isto é, $K(s)$ é tal que o sistema realimentado da figura 4.1 é estável e seja $D(s) = \gamma(s)/\delta(s)$, $\delta(s)$ conhecido. Então

$$\lim_{t \rightarrow \infty} y_D(t) = 0, \quad (4.10)$$

isto é, o sistema rejeitará assintoticamente o sinal de perturbação $d(t)$, se e somente se

$$n_G(s)d_K(s) = \chi(s)\delta^+(s) \quad (4.11)$$

onde $\delta^+(s)$ é um polinômio formado com os zeros de $\delta(s)$ com parte real positiva ou nula e $\chi(s)$ é um polinômio qualquer.

Prova: Usando as notações das equações (4.1) e (4.2), tem-se que, a expressão (4.6) pode ser escrita como:

$$Y_D(s) = \frac{n_G(s)d_K(s)}{[n_G(s)d_K(s) + d_G(s)d_K(s)]} \frac{\gamma(s)}{\delta^+(s)\delta^-(s)} \quad (4.12)$$

Note que

$$n_G(s)d_K(s) + d_G(s)d_K(s) = p_C(s), \quad (4.13)$$

onde $p_C(s)$ denota o polinômio característico de malha fechada. Como, por hipótese, $K(s)$ estabiliza $G(s)$ então $p_C(s)$ é um polinômio de Hurwitz e, portanto,

$$\lim_{t \rightarrow \infty} y_D(t) = \lim_{s \rightarrow 0} sY_D(s) = 0 \quad (4.14)$$

se e somente se $\delta^+(s)$ for um divisor de $n_G(s)d_K(s)$, ou equivalentemente, se existir um polinômio $\chi(s)$ tal que $n_G(s)d_K(s) = \chi(s)\delta^+(s)$ \square

O teorema 4.1 acima mostra que para que haja rejeição assintótica de um sinal externo de perturbação aplicado na entrada da planta, as dinâmicas desse sinal que possuem parte real positiva ou nula devem ser zeros da planta

ou pólos do controlador. Vale ressaltar que, somente em casos especiais, essas dinâmicas serão também zeros da função de transferência da planta e portanto, para se conseguir a rejeição assintótica de um sinal de perturbação, é mais comum fazer com que essas dinâmicas sejam também pólos do controlador, isto é,

$$K(s) = \frac{1}{\delta^+(s)} \bar{K}(s) \quad (4.15)$$

onde $\bar{K}(s) = \bar{n}_K(s)/\bar{d}_K(s)$, $\bar{n}_K(s)$ e $\bar{d}_K(s)$ serão escolhidos de tal forma que $n_G(s)\bar{n}_K(s) + d_G(s)\bar{d}_K(s)\delta^+(s)$ seja um polinômio de Hurwitz.

Uma vez obtida uma condição necessária e suficiente para a rejeição assintótica de um sinal externo de perturbação, o passo seguinte é considerar o problema do rastreamento assintótico de um sinal de referência. A abordagem deste problema é feita de forma análoga à anterior, levando à condição necessária e suficiente do teorema seguinte.

Teorema 4.2 Seja $K(s)$ um controlador para o qual o sistema realimentado da figura 4.1 seja estável e assuma que a transformada de Laplace do sinal de referência $r(t)$ é $R(s) = \alpha(s)/\beta(s)$, $\beta(s)$ conhecido. Então

$$\lim_{t \rightarrow \infty} e_R(t) = 0, \quad (4.16)$$

isto é, $y(t)$ rastreia assintoticamente $r(t)$, se e somente se

$$d_G(s)d_K(s) = \eta(s)\beta^+(s) \quad (4.17)$$

onde $\beta^+(s)$ é um polinômio formado com os zeros de $\beta(s)$ com parte real positiva ou nula e $\eta(s)$ é um polinômio qualquer.

Prova: A partir das equações (4.6) e (4.9), pode-se escrever:

$$\begin{aligned} E_R(s) &= R(s) - Y_R(s) \\ &= \frac{1}{1 + G(s)K(s)} R(s) \end{aligned} \quad (4.18)$$

e usando a notação da equação (4.1) tem-se:

$$E_R(s) = \frac{d_G(s)d_K(s)}{[n_G(s)n_K(s) + d_G(s)d_K(s)]} \frac{\alpha(s)}{\beta^+(s)\beta^-(s)}. \quad (4.19)$$

Como, por hipótese, $G(s)$ estabiliza $K(s)$, então $p_c(s) = n_G(s)n_K(s) + d_G(s)d_K(s)$ é um polinômio de Hurwitz e, portanto,

$$\lim_{t \rightarrow \infty} e_R(t) = \lim_{s \rightarrow \infty} sY_R(s) = 0 \quad (4.20)$$

se e somente se $d_G(s)d_K(s)$ for múltiplo de $\beta^+(s)$, ou equivalentemente, se existir um polinômio $\eta(s)$ tal que $d_G(s)d_K(s) = \eta(s)\beta^+(s)$. \square

De acordo com o teorema 4.2, para que haja rastreamento assintótico de um sinal de referência, as dinâmicas desse sinal com parte real positiva ou nula devem ser pólos do controlador ou da planta. Observe que, somente em casos muito especiais haverá coincidência dos pólos da planta com essas dinâmicas. Portanto, de uma forma geral, o rastreamento assintótico será obtido fazendo-se:

$$K(s) = \frac{1}{\beta^+(s)} \bar{K}(s) \quad (4.21)$$

onde $\bar{K}(s) = \bar{n}_K(s)/\bar{d}_K(s)$, $\bar{n}_K(s)$ e $\bar{d}_K(s)$ serão tais que $n_G(s)\bar{n}_K(s) + d_G(s)\bar{d}_K(s)\beta^+(s)$ seja estável.

Finalmente, note que os objetivos de rastreamento assintótico de um sinal de referência $R(s) = \alpha(s)/[\beta^-(s)\beta^+(s)]$ e a rejeição assintótica de um sinal externo de perturbação $D(s) = \gamma(s)/[\delta^+(s)\delta^-(s)]$ serão, em geral, atingidos simultaneamente, com um mesmo controlador, fazendo-se

$$K(s) = \frac{1}{\text{mmc}[\beta^+(s), \delta^+(s)]} \bar{K}(s) \quad (4.22)$$

onde $\text{mmc}[\beta^+(s), \delta^+(s)]$ denota o mínimo múltiplo comum de $\beta^+(s)$ e $\delta^+(s)$ e $\bar{K}(s) = \bar{n}_K(s)/\bar{d}_K(s)$, com $\bar{n}_K(s)$ e $\bar{d}_K(s)$ sendo calculados de forma que

$$p_C(s) = n_G(s)\bar{n}_K(s) + d_G(s)\bar{d}_K(s)\text{mmc}[\beta^+(s), \delta^+(s)] \quad (4.23)$$

seja um polinômio de Hurwitz.

Exemplo 4.1: Suponha que para o sistema realimentado da figura 4.1, $R(s) = A/s$ e $D(s) = B/s$, onde $A, B \in \mathbb{R}$ são as amplitudes dos degraus de referência e de perturbação, respectivamente. Portanto, para que sejam atingidos, simultaneamente, os objetivos de rastreamento assintótico de $r(t)$ e rejeição, também assintótica, de $d(t)$, o controlador deve possuir ação integral, isto é,

$$K(s) = \frac{1}{s} \bar{K}(s) \quad (4.24)$$

$\bar{K}(s)$ sendo calculado de tal sorte que o sistema realimentado seja estável com bom desempenho transitório. Guarde bem este resultado! Ele será importante quando do projeto de um controlador de velocidade para o grupo motor-gerador.

4.1.2 Sensibilidade

A sensibilidade de um sistema está associada à não manutenção do desempenho e da estabilidade em presença de variações nos parâmetros da planta. Essas variações são, em geral, devidas a: (i) erros de identificação da função de transferência do sistema, uma vez que os valores dos parâmetros, obtidos a partir da identificação, jamais representarão exatamente o sistema real e (ii) envelhecimento dos componentes do sistema. Desta forma, é importante analisar como o sistema irá se comportar em presença de tais variações (comumente denominadas de incertezas no modelo), seja no que se refere ao desempenho do sistema compensado, ou com relação à manutenção da estabilidade.

A sensibilidade de um sistema em relação à variação em um determinado elemento é quantificada pela razão entre a variação percentual da função de transferência do sistema e a variação percentual da função de transferência do elemento considerado, *i.e.*,

$$S_G^T(s) = \frac{\frac{\Delta T(s)}{T(s)} 100\%}{\frac{\Delta G(s)}{G(s)} 100\%} \quad (4.25)$$

onde $T(s)$ representa a função de transferência global do sistema e $G(s)$ é a função de transferência do elemento considerado (em geral, a planta). A implicação imediata da equação (4.25) acima é que, quão mais perto de 1 for S_G^T , mais sensível a variações nos parâmetros de $G(s)$ será o sistema como um todo. Observe que a equação (4.25) pode ser escrita como:

$$S_G^T(s) = \frac{\Delta T(s)}{\Delta G(s)} \frac{G(s)}{T(s)} \quad (4.26)$$

e, portanto, quando $\Delta G(s) \rightarrow 0$ tem-se que $\Delta T(s) \rightarrow 0$ e conseqüentemente,

$$S_G^T(s) = \frac{G(s)}{T(s)} \frac{\partial T(s)}{\partial G(s)}. \quad (4.27)$$

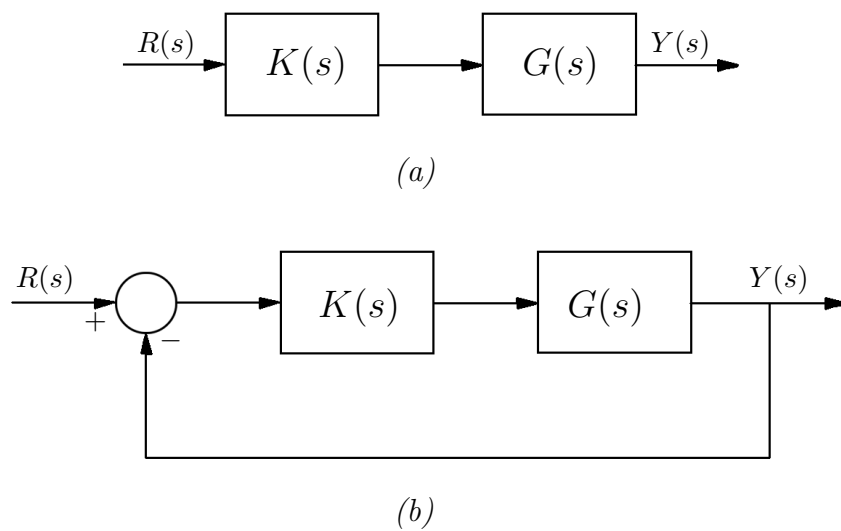


Figura 4.2: Diagrama de blocos: (a) sistema em malha aberta; (b) sistema realimentado

A redução da sensibilidade do sistema à variação dos parâmetros da função de transferência da planta é uma das razões para se usar a realimentação. Este fato será evidenciado no exemplo seguinte.

Exemplo 4.2: Sejam os sistemas em malha aberta e fechada representados na figura 4.2. Calcule S_G^T para ambos os casos.

Consideremos inicialmente o sistema em malha aberta. Neste caso $T(s) = G(s)K(s)$ e, portanto:

$$S_G^T = \frac{G}{T} \frac{\partial T}{\partial G} = \frac{G}{GK} K = 1 \quad (4.28)$$

que revela que qualquer erro de identificação ou variação nos parâmetros de $G(s)$ irá implicar num erro, de mesma magnitude na função de transferência (e por conseguinte, no desempenho) do sistema como um todo.

Para o sistema em malha fechada, tem-se que $T(s) = G(s)K(s)/[1 +$

$G(s)K(s)$], e então:

$$\begin{aligned} S_G^T &= \frac{G}{T} \frac{\partial T}{\partial G} \\ &= G \frac{1 + GK}{GK} \frac{(1 + GK)K - GK^2}{(1 + GK)^2} \\ &= \frac{1}{1 + GK} \end{aligned} \quad (4.29)$$

Suponha que $K(s)$ estabiliza o sistema da figura 4.2(b) então, para uma dada frequência ω_0 , S_G^T será reduzida de 1 (sistema em malha aberta) para

$$|S_G^T(j\omega_0)| = \frac{1}{|1 + G(j\omega_0)K(j\omega_0)|}, \quad (4.30)$$

que é menor que 1, evidenciando o benefício da realimentação no que se refere à diminuição da sensibilidade do sistema à variação nos parâmetros de $G(s)$.
□

4.1.3 Complementos

(a) Compensação em malha aberta

Suponha que se deseje fazer uma compensação em malha aberta conforme o diagrama de blocos da figura 4.3 e seja $K(s) = K_p$. Nessas condições:

- (1) Para quais valores de K_p , o sistema será estável?
- (2) Entre os valores de K_p que tornam o sistema estável, encontre aquele que faz com que o sistema tenha um erro de regime permanente nulo para uma entrada igual ao degrau de amplitude V_r .

(b) Compensação em malha fechada (controlador integral)

Suponha, agora, que o objetivo seja projetar um sistema de controle realimentado para o sistema, conforme o diagrama de blocos da figura 4.4. Conforme visto na seção 4.1.1, exemplo 4.1, a estrutura mais simples para $K(s)$ de tal sorte que o sistema realimentado rastreie e rejeite assintoticamente um degrau é $K(s) = K_I/s$, onde K_I é calculado para que o sistema realimentado seja estável e com bom desempenho transitório. Construa o gráfico do lugar das raízes do sistema e responda:

- (1) Para quais valores de K_I , o sistema realimentado será estável?

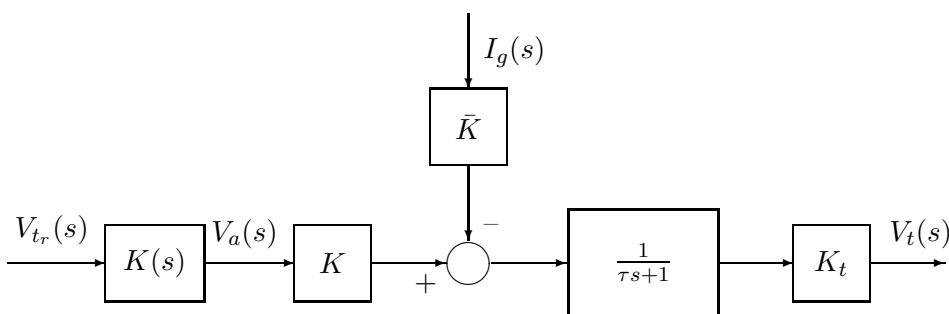


Figura 4.3: Diagrama de blocos para o sistema em malha aberta

(2) Para quais valores de K_I , o sistema realimentado será: (i) super-amortecido; (ii) criticamente amortecido; (iii) sub-amortecido e (iv) sub-amortecido com percentual menor ou igual a 5%.

(c) Compensação em malha fechada (controlador proporcional + integral)

Seja, agora

$$K(s) = K_p \frac{s+z}{s} = K_p \left(1 + \frac{z}{s}\right) = K_p \left(1 + \frac{1}{T_i s}\right) \quad (4.31)$$

onde $T_i = 1/z$, K_p e $z > 1/\tau$ a serem determinados. Esboce o gráfico do lugar das raízes do sistema realimentado da figura 4.4 com o controlador (4.31) e responda:

(1) Para quais valores de K_p , o sistema realimentado será estável?

(2) Para qual valor de K_p , os pólos do sistema realimentado serão complexos e com parte real igual a $-1/\tau$?

4.2 Projeto do controlador de velocidade

Como o objetivo do sistema de controle a ser aqui desenvolvido é controlar a velocidade de rotação do motor, isto é, mantê-la em um determinado valor, é natural considerar como sinal de referência um degrau de amplitude V_r Volts, isto é:

$$v_{tr}(t) = \begin{cases} V_r (V), & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (4.32)$$

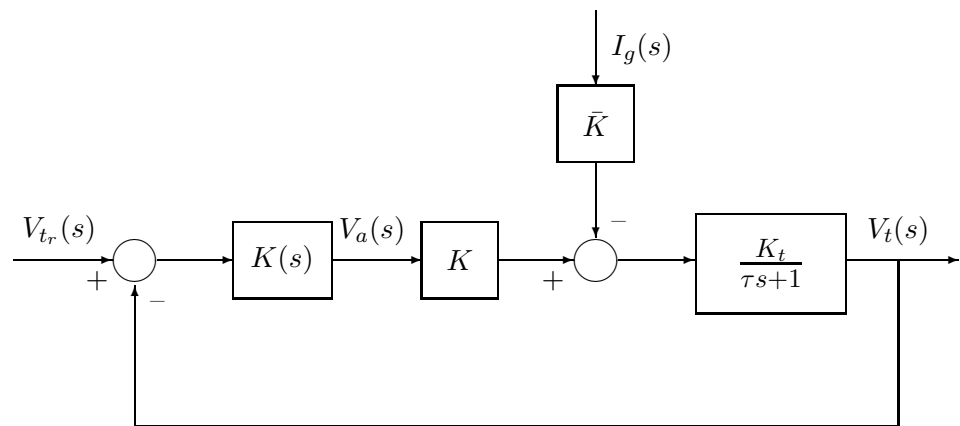


Figura 4.4: diagrama de blocos para o sistema realimentado

Note que está implícito no objetivo de rastreamento do sinal $v_r(t)$, a necessidade do sistema ser estável. Além desses objetivos (estabilidade e rastreamento assintótico do sinal de referência) o sistema deve:

1. Rejeitar sinais de perturbação, que podem ser modelados como degraus de amplitude I_g Ampères, isto é:

$$i_g(t) = \begin{cases} I_g (A), & t \geq t_0 \\ 0, & t < t_0 \end{cases} \quad (4.33)$$

onde, $t_0 \geq 0$.

2. Baixa sensibilidade a erros de identificação dos parâmetros do modelo do grupo motor gerador.
3. Bom desempenho transitório. Como se trata de uma planta didática, o desempenho do sistema será definido unicamente em termos do tempo de acomodação da resposta ao degrau do sistema em malha aberta.

É sabido que os objetivos de um sistema de controle somente serão alcançados com sistemas realimentados. Porém, como forma de ilustrar os benefícios da realimentação, iremos, inicialmente, considerar a possibilidade de se usar um controle em malha aberta e, em seguida, projetar um sistema de controle realimentado.

4.2.1 Sistema de controle em malha aberta

A partir do diagrama de blocos da figura 4.3 pode-se escrever:

$$V_t(s) = \frac{K}{\tau s + 1} K(s) V_{tr}(s) - \frac{\bar{K}}{\tau s + 1} I_g(s). \quad (4.34)$$

É fácil verificar que o sistema de controle em malha aberta será estável se e somente se $K(s)$ for estável. Como, por simplicidade foi adotado como satisfatório o tempo de acomodação do sistema sem compensação, pode-se utilizar um controlador estático, isto é:

$$K(s) = K_P \quad (4.35)$$

onde K_P deve ser determinado de acordo com a seção 4.1.3(a) de tal sorte que, em regime permanente, $v_t(t) = V_r$.

Uma vez obtido K_P , o passo seguinte é fazer a análise do desempenho do sistema compensado, utilizando o SIMULINK. Para tanto, construa um modelo, em SIMULINK, equivalente ao diagrama de blocos da figura 4.3. Em seguida, tendo $0s$ e $2s$ como instantes inicial e final de simulação, proceda à seguinte simulação:

1. Para uma corrente $i_g(t) = 0$ (A), aplique um degrau de amplitude igual a $10V$ com início em $t = 0s$. Em seguida, encontre o erro de estado permanente e o valor do tempo de acomodação (t_s) da resposta.
2. Suponha, agora, que tenha havido um erro de 10% na identificação de K_a , isto é $K_{a_{\text{real}}} = 0,9K_a$. Ainda com $i_g(t) = 0$ (A), aplique um degrau de amplitude igual a $10V$ com início em $t = 0s$ e encontre o erro de estado permanente.
3. Retorne K_a ao valor obtido na identificação e, em seguida, aplique simultaneamente as entradas $v_{tr}(t)$ e $i_g(t)$, ambas degraus, com amplitudes respectivamente iguais a $10V$ e $0,450A$ e inícios em $t = 0$ e $t = 1s$ (s). Realizada a simulação, encontre o erro de regime permanente.

4.2.2 Sistema de controle com realimentação

Você deve ter observado que nas simulações 2 e 3 realizadas na seção anterior, o erro de regime permanente foi diferente de zero, isto é, o sistema

compensado em malha aberta não foi capaz de rejeitar um sinal externo de perturbação e foi, também, sensível a erros de identificação. A realimentação surge, então, como a única alternativa para superar essas limitações do controlador em malha aberta. Conforme visto no exemplo 4.1, para que o sistema realimentado seja capaz de rastrear assintoticamente um degrau e rejeitar, também assintoticamente, sinais externos de perturbação do tipo degrau, o controlador deve ser dinâmico e, mais crucial ainda, ter um pólo em $s = 0$. Portanto, a forma mais simples para esse controlador é o tipo integral (I), sendo dado por:

$$K(s) = \frac{K_I}{s} \quad (4.36)$$

onde K_I deve ser calculado de tal forma que o sistema realimentado seja estável com desempenho transitório satisfatório, isto é, que satisfaça as especificações de desempenho. Lembre-se de que, no presente caso, requer-se que o tempo de acomodação da resposta ao degrau para o sistema realimentado seja aproximadamente igual ao do sistema em malha aberta.

Para verificar os benefícios da introdução da realimentação acrescida de uma compensação dinâmica, vamos inicialmente considerar os seguintes casos: (i) sistema realimentado criticamente amortecido e (ii) sistema realimentado subamortecido com percentual de ultrapassagem menor ou igual a 5%. Para tanto, proceda da seguinte forma:

1. Calcule o valor de K_I de tal forma que o sistema realimentado seja criticamente amortecido.
2. Com o valor de K_I , calculado no item anterior, e com os valores de K , K_t , \bar{K} e τ , identificados para o grupo motor-gerador, construa um modelo em SIMULINK equivalente ao diagrama de blocos da figura 4.4 e ajuste os tempos inicial e final de simulação em 0 e $2s$, respectivamente.
3. Para uma corrente $i_g(t) = 0$ (A), aplique um degrau de amplitude igual a $10V$ com início em $t = 0s$. Em seguida, encontre o erro de estado permanente e o valor do tempo de acomodação (t_s) da resposta.
4. Suponha, agora, que tenha havido um erro de 10% na identificação de K_a , isto é $K_{a_{\text{real}}} = 0,9K_a$. Ainda com $i_g(t) = 0$ (A), aplique um degrau de amplitude igual a $10V$ com início em $t = 0s$ e encontre o erro de estado permanente.

5. Retorne K_a ao valor obtido na identificação e, em seguida, aplique simultaneamente as entradas $v_{tr}(t)$ e $i_g(t)$, ambas degraus, com amplitudes respectivamente iguais a 10 V e $0,450\text{ A}$ e inícios em $t = 0$ e $t = 1\text{ s}$ (s). Realizada a simulação, encontre o erro de regime permanente.
6. Calcule, agora, o valor de K_I de tal forma que o sistema realimentado seja subamortecido com percentual de ultrapassagem menor ou igual a 5%.
7. Repita os itens 2 a 5, acima.

O aluno deve ter observado nas simulações realizadas que o sistema realimentado com o controlador integral, em ambos os casos (criticamente amortecido e subamortecido com percentual de ultrapassagem menor ou igual que 5%), eliminou a sinal externo de perturbação e teve erro de regime permanente nulo, mesmo quando se supôs um erro de 10% no ganho da função de transferência da planta. Contudo, o tempo de acomodação da resposta ao degrau para o sistema realimentado é, para ambos os casos, o dobro do tempo de acomodação da resposta ao degrau do sistema em malha aberta. Para se justificar esse fato, construa o diagrama do lugar das raízes do sistema realimentado para $K(s) = K_I/s$ e, a partir do valor da parte real dos pólos, mostre que, de fato, o tempo de acomodação da resposta ao degrau deveria ser o dobro daquele obtido para o sistema em malha aberta.

Assim sendo, para que o desempenho do sistema realimentado seja aproximadamente igual ao do sistema em malha aberta, os pólos de malha fechada devem ter parte real aproximadamente igual a $-1/\tau$ (pólo da planta). Isto implica que o diagrama dos lugar das raízes deve se deslocar para a esquerda e, para tanto, é necessário que exista um zero igual a $-z$ ($z > 0$) à esquerda de $-1/\tau$. Portanto, $K(s)$ deve ter a seguinte forma:

$$K(s) = \frac{K_p(s+z)}{s} = K_p \left(1 + \frac{z}{s}\right) = K_p \left(1 + \frac{1}{T_i s}\right) \quad (4.37)$$

onde $T_i = 1/z$. Note, pela equação 4.37 acima, que a introdução do zero no controlador equivale a dotá-lo de ação proporcional, resultando num controlador PI.

Finalmente, para verificar se tal controlador, de fato, tornou o sistema realimentado mais rápido, proceda da seguinte forma:

1. Calcule o valor de K_p e T_i de tal sorte que os pólos do sistema realimentado tenham parte real igual a $-1/\tau$.
2. Substitua a função de transferência do bloco referente ao controlador pela equação 4.37, com os valores de K_p e T_i calculados no item anterior.
3. Para uma corrente $i_g(t) = 0$ (A), aplique um degrau de amplitude igual a $10V$ com início em $t = 0$ s. Em seguida, encontre o valor do tempo de acomodação (t_s) da resposta.

4.3 Comentários finais

Uma vez obtida uma função de transferência para o controlador, o passo final é a sua implementação no sistema real. A implementação da função de transferência 4.37 pode ser feita utilizando dispositivos eletrônicos analógicos ou digitais. Neste laboratório utilizar-se-á um controlador analógico, deixando o controlador digital para um laboratório futuro. A implementação do controlador no sistema real será o assunto do próximo capítulo.

Capítulo 5

Implementação do controlador de velocidade

Neste capítulo trataremos da implementação do controlador projetado no capítulo anterior. Como se trata de um laboratório para um primeiro curso de Sistemas de Controle, será desenvolvido um controlador analógico, em cuja construção serão utilizados amplificadores operacionais¹. Esses amplificadores, pelas suas características de resposta e preço, são freqüentemente utilizados na implementação de controladores analógicos.

Antes de entrarmos na implementação do controlador faremos um breve estudo introdutório sobre amplificadores operacionais. Neste estudo não consideraremos detalhes construtivos (que geralmente é feito nos cursos de Eletrônica), nos preocupando somente com os aspectos externos do dispositivo e com as suas características que, quando exploradas, nos permitirão construir circuitos que tenham um determinado comportamento. Isso será realizado na seção 5.1. Caso o aluno já possua conhecimentos suficientes desse assunto, pode omitir esse estudo e caso o aluno deseje obter maiores detalhes sobre teoria e prática de amplificadores operacionais deve consultar Graeme et al. (1981). Dotado dos fundamentos necessários à utilização dos amplificadores operacionais, o passo seguinte será a construção de um circuito eletrônico para o controlador, o que será feito na seção 5.2.

¹Este mesmo controlador poderia ser implementado digitalmente, isto é, utilizando-se computadores digitais.

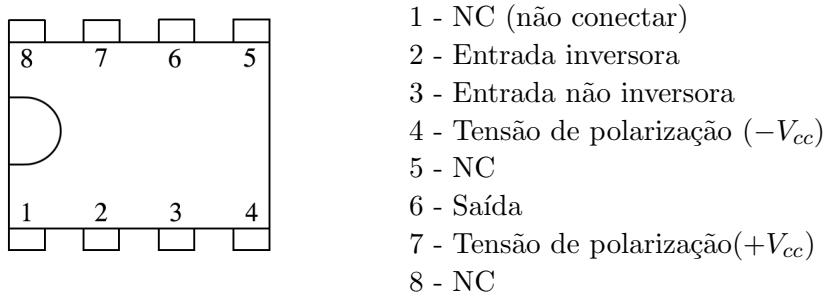


Figura 5.1: Representação esquemática de um amplificador operacional

5.1 Amplificadores Operacionais

5.1.1 Conceitos básicos

Amplificadores operacionais são amplificadores que têm elevado ganho e são geralmente utilizados para amplificar sinais que se estendem sobre uma ampla faixa de frequências. Na figura 5.1 está representado um amplificador operacional de oito pinos, com as respectivas indicações das conexões que devem ser efetuadas em cada um dos seus terminais e, na figura 5.2, o símbolo mais comumente adotado para representá-lo em um circuito. Observe na figura 5.1 o chanfro na parte esquerda do retângulo. Esse chanfro é utilizado nos circuitos integrados encapsulados para servir como referência para a numeração dos pinos. Note, ainda na figura 5.1, a necessidade das tensões externas de polarização ($\pm V_{CC}$), que são devidas ao fato dos amplificadores operacionais serem, na verdade, circuitos integrados construídos com diversos estágios de transistores, que devem, portanto, ser polarizados.

A maioria dos amplificadores operacionais possui dois terminais de entrada e um terminal de saída, conforme representado na figura 5.2. A entrada (-) é denominada inversora enquanto a entrada (+) é chamada não-inversora. Isto faz com que esses amplificadores sejam conhecidos como diferenciais, uma vez a tensão de saída é proporcional à diferença entre as tensões aplicadas nos seus terminais de entrada. Utilizando-se a notação da figura 5.2, essa característica é matematicamente descrita pela seguinte equação:

$$v_0 = A(v_2 - v_1) \quad (5.1)$$

onde A é o ganho de malha aberta ($A > 45.000$), sendo definido pelos tran-

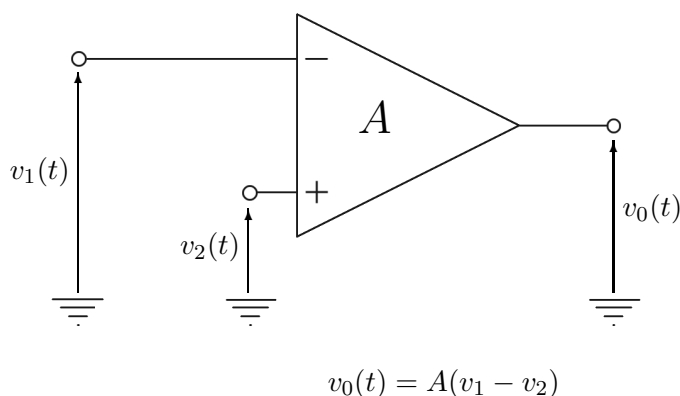


Figura 5.2: Simbologia adotada para amplificadores operacionais

sistores internos do amplificador. Usualmente, na prática, este ganho é considerado infinito ($A \rightarrow \infty$). Outras características, geralmente adotadas na prática são: (i) $v_0 = 0$ quando $v_1 = v_2$ (tensão ‘offset’ de entrada nula); (ii) impedância de entrada infinita, que permite baixa ou nenhuma interferência com os elementos colocados em cascata no circuito ($Z_{in} = \infty$); (iii) impedância de saída nula ($Z_{out} = 0$); (iv) largura de faixa infinita; e (v) atraso de resposta nulo. Na tabela 5.1 são mostrados os valores reais de algumas dessas grandezas para os amplificadores operacionais 741 e LF356. É importante ressaltar que o valor da impedância de entrada para o LF356 é aproximadamente $10^{12}\Omega$, isto é, praticamente infinita². Em face disso, o LF356 será utilizado na implementação do controlador de velocidade do grupo motor-gerador.

5.1.2 Configurações básicas

A configuração em malha aberta da figura 5.2 é raramente utilizada. Ao se construir um circuito utilizando um amplificador operacional são geralmente utilizados outros elementos de circuitos (resistores, capacitores, indutores, etc) para formar redes de realimentação. Essas configurações farão com que o novo circuito tenha características diferentes daquelas do amplificador em malha aberta, como por exemplo: modificações do ganho, da resposta em

²A razão dessa elevada impedância de entrada reside no fato desse amplificador ser construído com transistores de efeito de campo.

	741	LF356
Ganho de malha aberta (V/mV)	200	200
Tensão 'offset' de entrada (mV)	2	3
Impedância de entrada ($M\Omega$)	2	10^6
Tensão de polarização (V)	± 15	± 15
Banda passante (MHz)	1	1
Corrente de saída máxima (mA)	5	5

Tabela 5.1: Características dos amplificadores operacionais 741 e LF356

freqüência e a possibilidade de se efetuar operações matemáticas como soma, integração, derivação e etc. Algumas dessas configurações são apresentadas a seguir:

Circuito inversor

Considere o circuito da figura 5.3. Aplicando-se a lei das correntes de Kirchhoff ao nó 1 e lembrando que, como o amplificador operacional tem impedância de entrada infinita, a corrente que flui para a entrada inversora é nula, pode-se escrever:

$$\begin{cases} \frac{e_i(t) - e_s(t)}{R_i} + \frac{e_o(t) - e_s(t)}{R_f} = 0 \\ e_o(t) = -Ae_s(t) \quad (A \rightarrow \infty) \end{cases} \quad (5.2)$$

Substituindo $e_s(t) = -e_o(t)/A$ na primeira equação, resulta:

$$\frac{e_i(t)}{R_i} + e_o(t) \left(\frac{1}{R_f} + \frac{R_i}{A} + \frac{R_f}{A} \right) = 0. \quad (5.3)$$

Finalmente, fazendo $A \rightarrow \infty$, obtém-se:

$$e_o(t) = -\frac{R_f}{R_i} e_i(t) \quad (5.4)$$

As seguintes observações podem ser feitas a partir da equação 5.4 acima: (i) o ganho não é mais determinado pelo amplificador e sim pelos elementos externos R_i e R_f ; (ii) quando o sinal de entrada for limitado, isto é, $|e_1(t)| < E_i$, tem-se que $|e_o(t)| < (R_f/R_i)E_i = E_0$ e, portanto limitado. Como $A \rightarrow \infty$, resulta:

$$|e_s(t)| = \frac{|e_o(t)|}{A} \rightarrow 0. \quad (5.5)$$

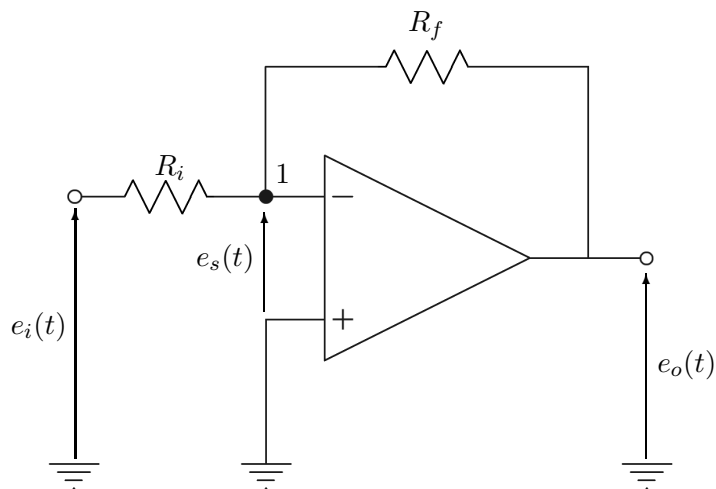


Figura 5.3: Circuito inversor

Por essa razão, os terminais referentes às entradas inversora e não-inversora são dito estarem em terra virtual, isto é, para efeitos de análises de redes contendo amplificadores, $e_s(t)$ pode ser feito identicamente nulo.

Circuito não-inversor

Considere agora o circuito da figura 5.4(a), que difere do circuito inversor da figura 5.3 principalmente pelo fato do sinal externo $e_i(t)$ estar agora conectado à entrada não-inversora. Aplicando a lei das correntes de Kirchhoff ao nó 1 e levando em conta os fatos de que a corrente que entra no amplificador operacional é nula e que os terminais + e - estão em terra virtual pode-se escrever:

$$\frac{e_i(t)}{R_i} + \frac{e_i(t) - e_o(t)}{R_f} = 0. \quad (5.6)$$

Após algumas manipulações algébricas simples, obtém-se:

$$e_o(t) = \frac{R_f + R_i}{R_i} e_i(t) \quad (5.7)$$

Note, na equação 5.7 acima que o ganho de malha fechada será sempre maior ou igual a 1 e é, mais uma vez, determinado por R_i e R_f . Observe ainda

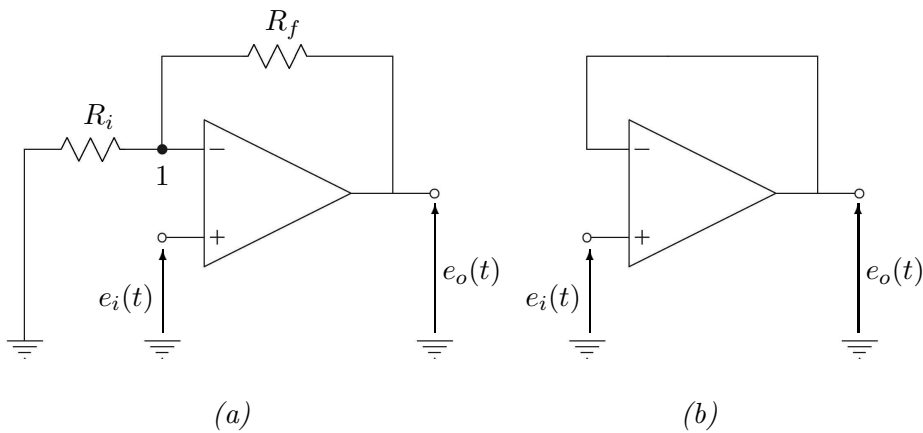


Figura 5.4: (a) Circuito não-inversor; (b) Seguidor de tensão

que, se R_f é feito igual a zero (curto-circuito) e $R_i = \infty$ (circuito aberto), conforme mostrado na figura 5.4(b), tem-se que o ganho será exatamente igual a 1. Nesse caso, o amplificador atua como um ‘seguidor de tensão’, isto é, a tensão de saída segue exatamente a tensão de entrada. A configuração de seguidor de tensão é largamente utilizada para prover isolamento entre os sinais da fonte e da carga, evitando assim interações indesejáveis.

Circuito comparador

Uma das maneiras de se fazer a comparação entre dois sinais é utilizando-se o circuito da figura 5.5. Aplicando-se a lei das correntes de Kirchhoff ao nó A e lembrando que os pontos A e B têm o mesmo potencial (terra virtual), pode-se escrever:

$$\begin{cases} e_B(t) = e_A(t) = \frac{R_3}{R_2 + R_3} e_2(t) \\ \frac{e_1(t) - e_B(t)}{R_1} + \frac{e_o(t) - e_B(t)}{R_f} = 0. \end{cases} \quad (5.8)$$

Após algumas manipulações algébricas elementares, obtém-se:

$$e_o(t) = -\frac{R_f(R_2 + R_3)}{R_1(R_2 + R_3)} e_1(t) + \frac{R_3(R_1 + R_f)}{R_1(R_2 + R_3)} e_2(t). \quad (5.9)$$

Finalmente, fazendo $R_1 = R_2 = R_3 = R_f = R$ na equação 5.9 acima, resulta:

$$e_o(t) = e_2(t) - e_1(t). \quad (5.10)$$

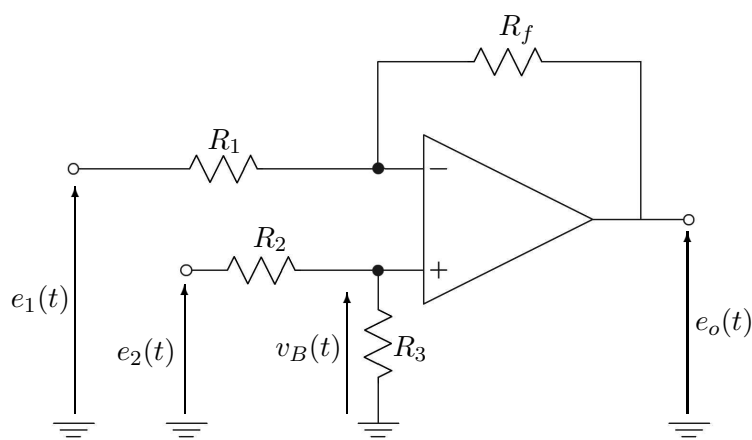


Figura 5.5: Circuito comparador genérico

Deve ser ressaltado que, na prática, é quase impossível ter quatro resistores de valores exatamente iguais e, portanto, a igualdade (5.10) não pode, a princípio, ser implementada fisicamente. Este problema pode ser superado inserindo-se potenciômetros de valores R'_1 , R'_2 , R'_3 e R'_f em série com os resistores R_1 , R_2 , R_3 e R_f . Esses potenciômetros devem ser ajustados de tal sorte que $R_1 + R'_1 = R_2 + R'_2 = R_3 + R'_3 = R_f + R'_f$. Para se verificar de que a igualdade (5.10) está sendo verificada (a menos de um 'offset'), aplica-se sinais iguais em $e_1(t)$ e $e_2(t)$ e mede-se o sinal $e_o(t)$, que nesse caso deve ser aproximadamente igual a zero (a menos de um 'offset').

Circuito integrador

O circuito da figura 5.6(a) representa uma maneira de se calcular analógicamente a integral de um sinal. Pode-se mostrar que a função de transferência do circuito é dada por:

$$\frac{E_o(s)}{E_i(s)} = -\frac{1}{R_i C_f s} \quad (5.11)$$

e, portanto,

$$e_o(t) = -\frac{1}{R_i C_f} \int_0^t e_i(\lambda) d\lambda. \quad (5.12)$$

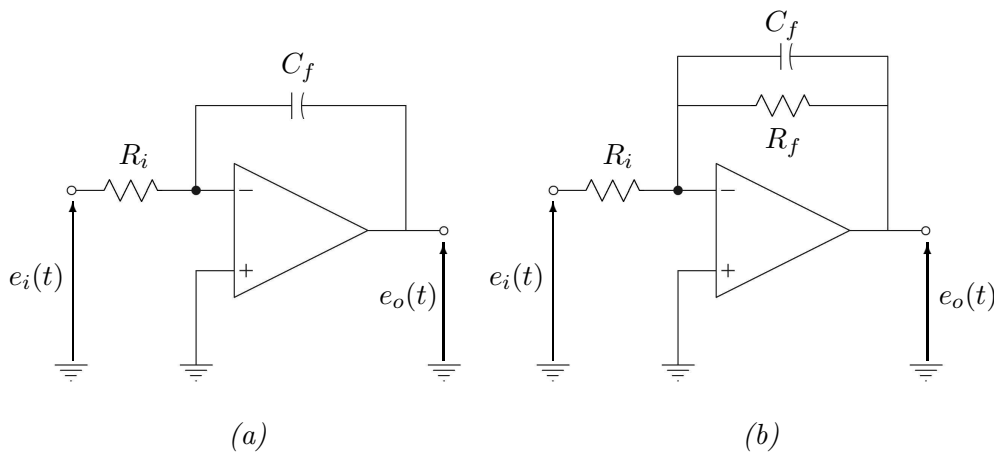


Figura 5.6: Circuitos integradores: (a) ideal; (b) prático.

A realimentação composta unicamente por um capacitor torna a configuração da figura 5.6(a) bastante sensível para alguns amplificadores operacionais (inclusive o 741), o que torna a sua implementação problemática. Para solucionar este problema, introduz-se um resistor em paralelo com o capacitor, conforme representado na figura 5.6(b). Neste caso, a função de transferência do sistema passa a ser a seguinte:

$$\frac{E_o(s)}{E_i(s)} = -\frac{R_f}{R_i} \frac{1}{R_f C_f s + 1} \quad (5.13)$$

cujas assíntotas do diagrama de módulo de Bode estão representadas na figura 5.7. Note que aumentando-se o valor do resistor R_f , o pólo $-1/R_f C_f$ se aproxima de zero ao mesmo tempo em que o ganho DC torna-se cada vez maior. Nestas condições, a resposta em frequência do sistema assemelha-se cada vez mais à de um integrador ideal e, portanto, o circuito da figura 5.6(b) pode ser usado com razoável precisão para se efetuar a integração de um sinal.

Circuito derivador

A derivação de um sinal analógico pode ser feita idealmente utilizando-se o circuito da figura 5.8(a). Para esse circuito, tem-se que:

$$e_o(t) = -R_f C_i \frac{d}{dt} e_i(t). \quad (5.14)$$

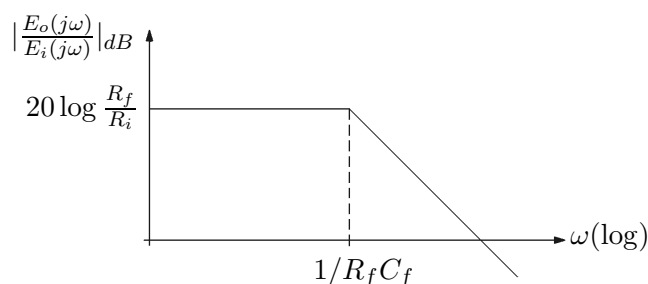


Figura 5.7: Assíntotas do diagrama de módulo de Bode para o circuito integrador prático da figura 5.6(b)

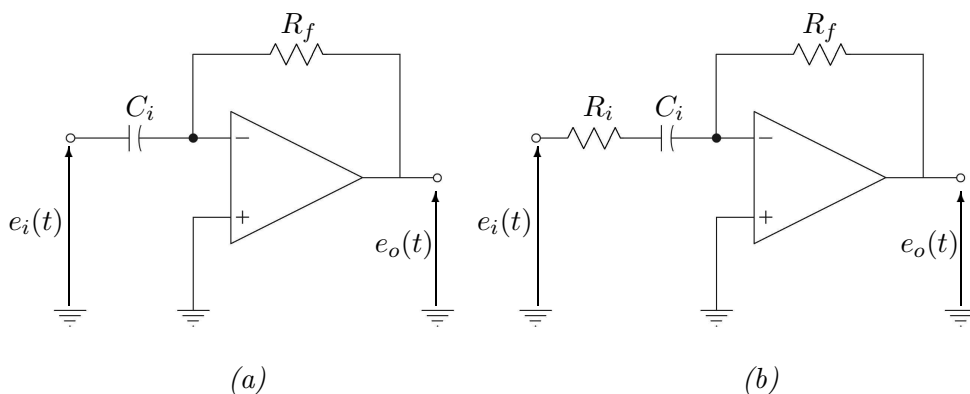


Figura 5.8: Circuitos derivadores: (a) ideal; (b) prático

A utilização desse circuito como um derivador pode trazer sérios problemas no que se refere a sinais com componentes de frequências muito altas, uma vez que tais sinais seriam amplificados por valores infinitos (pelo menos teoricamente). Para solucionar este problema, faz-se uso do circuito da figura 5.8(b), cuja função de transferência é dada por:

$$\frac{E_o(s)}{E_i(s)} = -\frac{sR_f C_i}{R_i C_i s + 1}, \quad (5.15)$$

cujas assíntotas do diagrama de módulo de Bode estão representadas na figura 5.9. Note que, diminuindo R_i , o pólo $-1/R_i C_i$ se afasta cada vez mais de

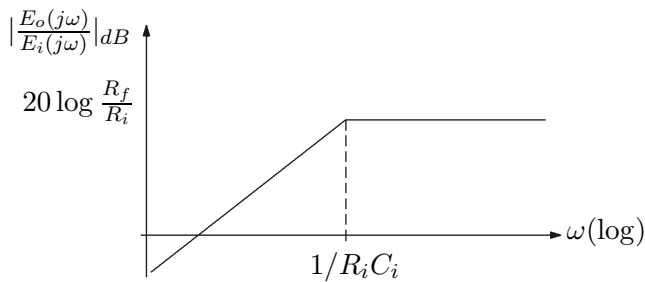


Figura 5.9: Assíntotas do diagrama de módulo de Bode para o circuito derivador prático da figura 5.8(b)

zero, fazendo com que o ganho de alta frequência aumente, o que torna a resposta em frequência desse sistema mais próxima daquela de um derivador ideal. Assim sendo, o circuito da figura 5.8(b) pode ser utilizado para se efetuar a derivação de um sinal.

Um circuito para controladores PID

Um circuito cuja saída $e_o(t)$ é proporcional à entrada $e_i(t)$, à sua integral e à sua derivada, isto é,

$$e_o(t) = K_p \left(e_i(t) + \frac{1}{T_i} \int_0^t e_i(\lambda) d\lambda + T_d \frac{d}{dt} e_i(t) \right) \quad (5.16)$$

está representado na figura 5.10. Pode-se mostrar que

$$\begin{aligned} K_p &= \frac{R_f}{R_i} \\ T_d &= R_i C_i \\ T_i &= R_f C_f. \end{aligned} \quad (5.17)$$

para $T_i \gg T_d$.

Em geral, os valores de K_p , T_i e T_d são determinados durante a fase de projeto do controlador. O problema passa, então, a ser o de encontrar R_i , R_f , C_i e C_f de tal forma que o circuito do controlador PID tenha valores de K_p , T_i e T_d o mais próximo possível dos valores calculados teoricamente. Para tanto, deve-se resolver o sistema de três equações e quatro incógnitas (5.17). O número maior de incógnitas, faz com que a escolha dos valores

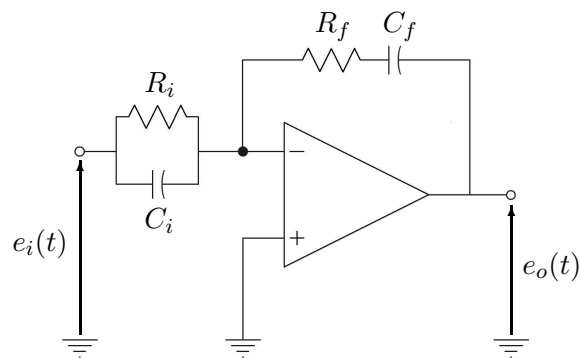


Figura 5.10: Circuito para controladores PID

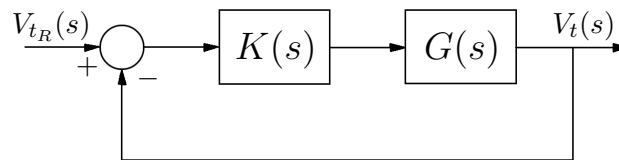


Figura 5.11: Diagrama de blocos correspondente ao sistema de controle do grupo motor-gerador

das capacitâncias e resistências não seja única, o que é bom do ponto de vista prático, uma vez que pode-se ‘jogar’ com os capacitores e resistores disponíveis no laboratório.

5.2 Controlador automático de velocidade

Após a breve introdução sobre amplificadores operacionais apresentada na seção anterior, estamos em condições de construir um circuito eletrônico analógico para o controlador de velocidade do grupo motor-gerador. O diagrama de blocos para o sistema está representado na figura 5.11, onde $G(s)$ representa a função de transferência da planta e

$$K(s) = K_p \frac{s+z}{s} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (5.18)$$

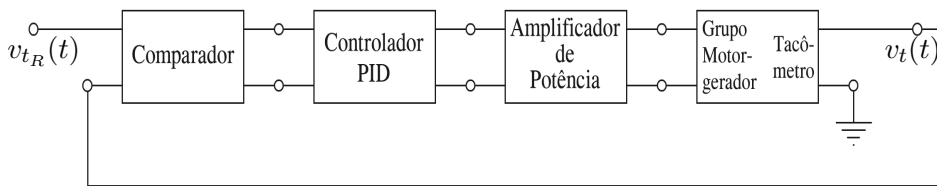


Figura 5.12: Representação esquemática do sistema de controle de velocidade do grupo motor-gerador

onde os valores de K_p e T_i foram determinados ao final do capítulo 4.

A partir do diagrama da figura 5.11, vê-se que a implementação do sistema de controle requer a construção de um circuito comparador e de um circuito para o controlador PID. Em princípio, nenhum outro elemento seria necessário e bastaria o engenheiro de controle conectar a saída do controlador ao motor. Porém, amplificadores operacionais têm corrente de saída muito baixa (aproximadamente $5mA$, conforme mostrado na tabela 5.1). Isto faz com que seja necessário acoplar na saída do amplificador operacional um amplificador de potência para poder ‘drenar’ a corrente necessária para o motor, o que faz com que a implementação do sistema de controle de velocidade para o grupo motor-gerador seja feita conforme mostrado no diagrama de blocos da figura 5.12

5.2.1 Implementação do comparador

O circuito utilizado para se efetuar a comparação entre o sinal de referência ($v_{t_R}(t)$) e o sinal do tacômetro ($v_t(t)$) será aquele representado na figura 5.5. Para sua implementação, proceda da seguinte forma:

1. Selecione quatro resistores de mesmo valor (de acordo com seu código de barras coloridas) com valores entre $1k\Omega$ e $10k\Omega$. Valores de resistores mais baixo que os sugeridos podem levar o amplificador a não executar exatamente a operação que dele esperamos.
2. Meça a resistência de cada um dos resistores R_1 , R_2 , R_3 e R_4 e anote-as.
3. Ajuste os quatro potenciômetros R'_1 , R'_2 , R'_3 e R'_4 de tal modo que $R_1 + R'_1 = R_2 + R'_2 = R_3 + R'_3 = R_4 + R'_4$.

4. Monte o circuito da figura 5.5 utilizando um amplificador LF356, efetuando-se as conexões de seus terminais de acordo com a figura 5.1. Lembre-se de que o amplificador deve ser polarizado com as tensões de $\pm 15V$ (ver tabela 5.1).
5. Para verificar se o circuito somador está, de fato, efetuando a subtração dos sinais, aplique sinais iguais a ambos os terminais. Isto pode ser feito, por exemplo, conectando-se ambas as entradas ao ponto de potencial zero (terra). Neste caso, de acordo com a tabela 5.1 a tensão ‘offset’ de saída deverá ser de, aproximadamente, $2mV$. Caso a tensão seja muito maior que $2mV$, varie suavemente cada um potenciômetro, observando se tal movimento implica numa diminuição da tensão de saída. Termine este procedimento quando a tensão de saída se aproximar de $2mV$ ou quando qualquer variação no potenciômetro resultar num aumento da tensão de ‘offset’.
6. Finalmente, como $e_o(t) = e_2(t) - e_1(t)$ (equação 5.10), faça $e_1(t) = v_t(t)$ (a tensão de saída do tacômetro) e $e_2(t) = v_{t_R}(t)$ (a tensão de referência).

5.2.2 Implementação do controlador

O circuito da figura 5.10 permite a implementação de um controlador PID. Porém, o controlador de velocidade desenvolvido ao final do capítulo 4 é do tipo PI. Assim sendo, caso o circuito da figura 5.10 seja usado para a implementação do controlador da equação 5.18, o ganho derivativo deve ser feito igual a zero. Com isso em mente, proceda da seguinte forma para a implementação do controlador:

1. Faça, inicialmente, $C_i = 0F$. Em seguida, utilizando um resistor R_f de aproximadamente $10k\Omega$, encontre um capacitor C_f e um resistor R_i , de tal forma que, a partir das equações 5.17, sejam obtidos valores de K_p e T_i aproximadamente iguais aos encontrados no final do capítulo 4.
2. Utilizando um amplificador LF356 monte o circuito da figura 5.10.
3. Para verificar se o controlador PI foi, de fato, implementado corretamente, aplique ao controlador um sinal de onda quadrada de frequência igual a $0,3Hz$, amplitude igual a $1V$ e valor médio igual a zero. Como o circuito tem ações proporcional e integral, o sinal de saída deve ser uma

rampa com inclinação definida pela constante de tempo de integração, T_i , e deslocada verticalmente, sendo este deslocamento determinado pelo ganho proporcional.

5.2.3 Amplificador de potência

O amplificador de potência utilizado no laboratório é também inversor e tem um ganho igual a cinco. Portanto, o aluno deve tomar o cuidado de certificar-se de que o ganho em malha aberta do sistema real (somador + PI + amplificador de potência) seja igual ao projetado e as inversões levem a uma realimentação negativa.

5.3 Experimento final: funcionamento do sistema de controle

Implementados o comparador e o controlador PI e tendo sido feito o ajuste para que o amplificador de potência tenha ganho igual a -1 , estamos em condições de verificar na prática se o sistema de controle projetado irá funcionar satisfatoriamente. Para tanto proceda da seguinte forma:

1. Conecte o terminal de saída do comparador ao ponto correspondente à entrada do controlador PI e a saída do controlador PI à entrada do amplificador de potência. Conecte ainda a saída do tacômetro à entrada correspondente do circuito comparador.
2. Aplique um sinal de onda quadrada de frequência igual a $0.3Hz$, amplitude igual a $9V$ e tensão mínima igual a zero ao terminal do comparador correspondente à tensão de referência.
3. Conecte a saída do amplificador de potência aos terminais do motor e observe no osciloscópio os sinais de referência ($v_{tR}(t)$) e da saída do tacômetro ($v_t(t)$). Responda às seguintes questões:
 - (i) Houve rastreamento do sinal de referência?
 - (ii) Em caso afirmativo, qual o tempo de acomodação aproximado da resposta?
 - (iii) O tempo de acomodação para o sistema real está próximo daquele obtido durante a simulação realizada no capítulo 4?

4. Conecte agora uma carga nos terminais do gerador e feche este circuito num instante próximo do meio do pulso. Verifique se o sistema rejeita esta perturbação assim como na simulação realizada no capítulo 4.

5.4 Comentários finais

Este laboratório procurou abordar do ponto de vista prático, os principais conceitos introduzidos na disciplina Sistemas de Controle I. Espera-se que os alunos tenham verificado que o ferramental teórico pode ser aplicado, efetivamente, na prática, para levar o sistema real a ter um comportamento bastante próximo daquele previsto durante a simulação. Outro aspecto a ser ressaltado, ao fim desse curso, é que as ferramentas utilizadas na identificação da função de transferência do grupo motor-gerador e no projeto do controlador de velocidade não se aplicam somente ao sistema utilizado nesse laboratório, podendo ser utilizadas em uma grande variedade de outros sistemas físicos e devem se usadas no futuro quando ao aluno for solicitado o projeto de um sistema de controle.

Referências Bibliográficas

- Basilio, J. C. (1995). *Notas de aula de Sistemas de Controle I*, UFRJ - Escola da Engenharia.
- Dorf, R. C. (1986). *Modern Control Systems*, Addison-Wesley.
- Graeme, J. G., Tobei, G. E. e Huelsman, L. P. (1981). *Operational Amplifiers: Design and Applications*, McGraw-Hill Book Company.
- Kheir, N. A., Aström, K. J., Auslander, D., Cheok, K. C., Franklin, G. F., Masten, M. e Rabins, M. (1991). Control systems engineering education, *Automatica* **32**: 147–166.
- MathWorks (1996). *Using Matlab*, The Math Works Inc.
- Stephan, R. M. (1991). *Laboratório de Controle I*, UFRJ - Escola de Engenharia.