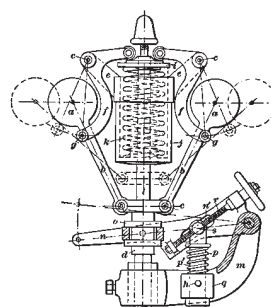
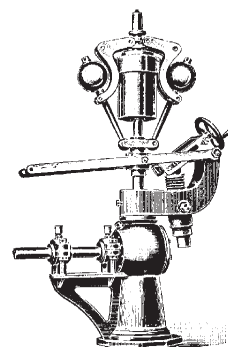
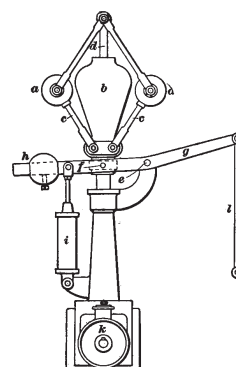
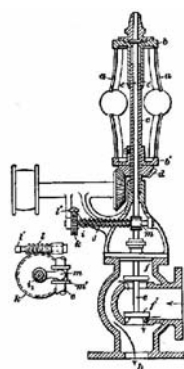
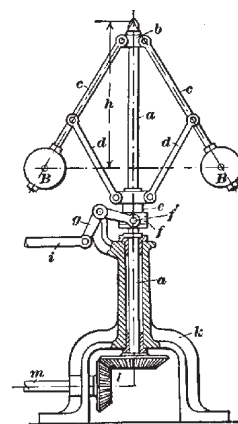
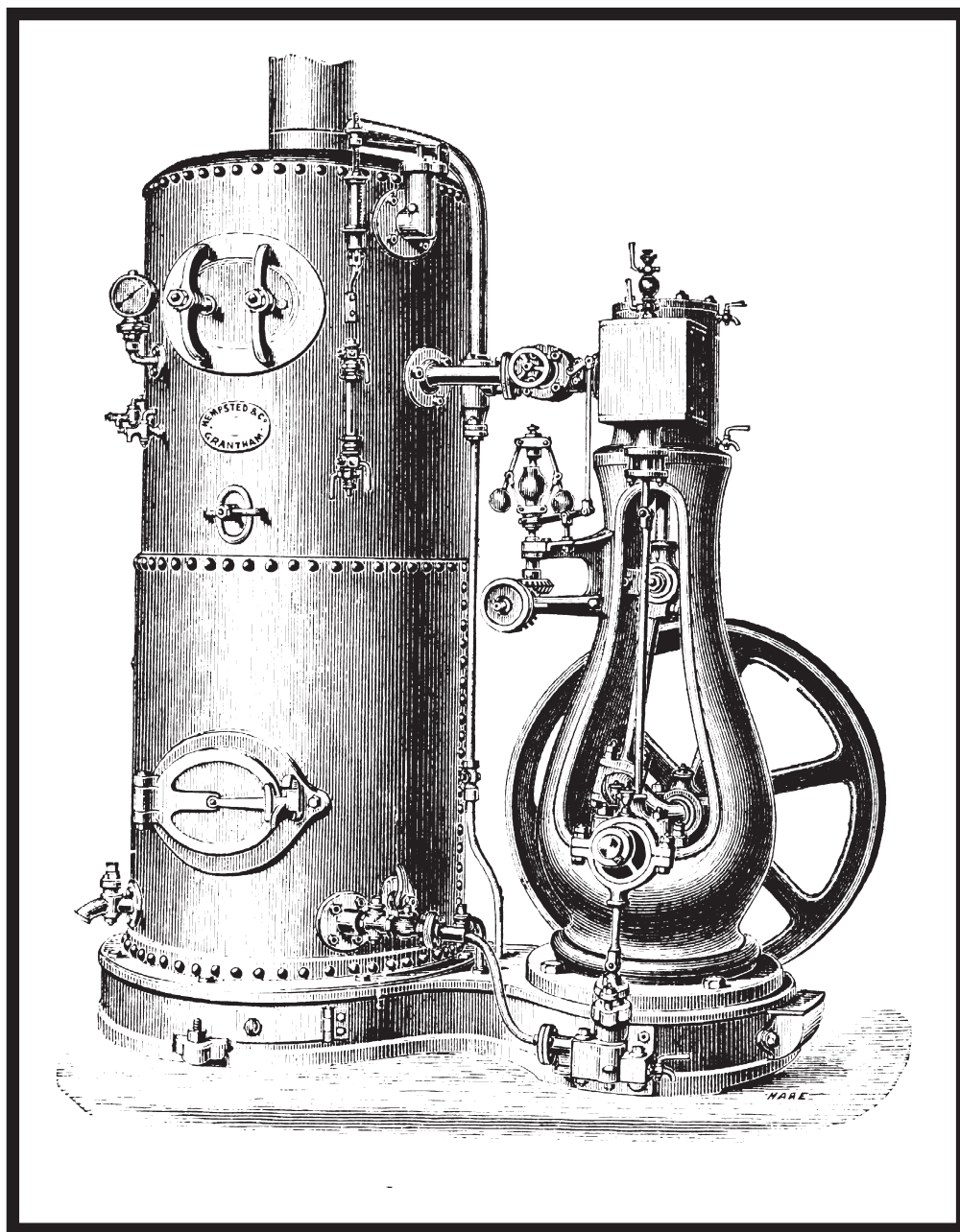


Escola Politécnica da Universidade de São Paulo

# LABORATÓRIO de CONTROLE



**Versão 2013a**

# Apresentação

---

## O curso Laboratório de Controle

O curso Laboratório de Controle tem por objetivo complementar o curso básico de Controle, que é ministrado no ano ou semestre anterior (conforme a ênfase). Enquanto que no curso teórico a ênfase é dada na apresentação dos conceitos básicos e das diferentes metodologias de análise e projeto de sistemas de controle, no curso de laboratório a ênfase recai principalmente nos aspectos práticos e na instrumentalização dos métodos aprendidos anteriormente.

O curso é dividido em três partes contendo um total de nove experiências. A estrutura geral é apresentada na Tabela 1. Cada experiência deve ser realizada em uma aula, após a qual resultados deverão ser apresentados. As primeiras experiências (PARTE I) visam basicamente a familiarização com os equipamentos e recursos disponíveis no laboratório, especialmente sua peça principal, o servomecanismo didático LJ MS15. As experiências seguintes (PARTE II) abordam as diferentes metodologias disponíveis para a obtenção de modelos matemáticos que descrevam o comportamento dinâmico desse servomecanismo. De posse desses modelos é possível explorar as diferentes possibilidades de Controle do sistema e suas variações, o que é feito na parte final do curso (PARTE III), que é composta de de uma sequência de projetos de controle com diferentes características.

Todo o curso de laboratório gira basicamente em torno desse servossistema didático, que é apresentado em detalhe na Experiência 1. Ele consiste de um motor DC, ligado a sensores de posição e velocidades angulares. Caso a tensão de alimentação do motor seja constante, digamos de 3 V, o motor girará a uma velocidade mais ou menos constante, digamos de 200 rpm, que vai depender da carga mecânica aplicada ao seu eixo (e que variará, caso a carga varie), e conseqüentemente a posição angular crescerá indefinidamente. Para um sistema como esse, pode-se desejar controlar sua velocidade, por exemplo fazendo com que o motor gire à velocidade desejada, mesmo que a carga mecânica varie (imagine uma esteira transportando caixas, que devem ser conduzidas a uma velocidade constante, quer sejam leves ou muito pesadas). Pode-se ainda controlar a posição angular do eixo, por exemplo fazendo com que o eixo gire até uma posição desejada e permaneça nela (imagine o braço de robô, que deve ser movido vazio até uma dada posição e que uma vez tendo atingido essa posição deva agarrar um objeto sem ceder ao seu peso, ou ainda um guindaste, que deva ser erguido até uma determinada altura). Apesar de se tratar de um equipamento eminentemente didático, em escala de bancada, uma infinidade de problemas práticos de controle é muito semelhante.

PARTE I FAMILIARIZAÇÃO	Consiste na apresentação dos equipamentos e programas de computador disponíveis e nas instruções para sua utilização. Como parte das atividades iniciais do curso, um modelo preliminar para o servomecanismo é obtido.	Exp. 1	Familiarização com o equipamento - I	Primeira apresentação do laboratório. Nesta experiência um primeiro modelo (fenomenológico) para o servomecanismo é obtido.
		Exp. 2	Familiarização com o equipamento - II	Apresentação dos programas de computador utilizados no laboratório.
PARTE II IDENTIFICAÇÃO	São utilizadas técnicas diferentes para a obtenção de modelos matemáticos de trabalho para o servomecanismo. Os modelos obtidos nessa parte do curso serão utilizados para projeto de controladores e análise de desempenho (via simulação computacional) na parte final do curso.	Exp. 3	Identificação I: Resposta em frequência	Obtenção de um modelo (entrada/saída) no domínio da frequência.
		Exp. 4	Identificação II: Respostas a degrau e modelagem não-linear	Obtenção de modelos lineares e não-linear (entrada/saída) no domínio do tempo
PARTE III CONTROLE	Nesta parte do curso diversas abordagens e possibilidades para o controle do servomecanismo são exploradas. São implementados controladores de velocidade e posição com diferentes objetivos e especificações. A aplicação de diferentes técnicas de projeto e comparações entre elas constituem-se no principal objetivo desta parte do curso.	Exp. 5	Controle proporcional	Primeiros sistemas básicos de controle. São analisados controladores proporcionais de velocidade e posição.
		Exp. 6	Controle de posição com realimentação auxiliar de velocidade	Um controlador de posição que visa obter um desempenho superior em comparação ao controle proporcional simples
		Exp. 7	Controle de posição com compensador por avanço de fase	Uma introdução ao uso de compensadores. O objetivo é analisar o emprego do compensador de avanço para controle de posição.
		Exp. 8	Controle de velocidade com compensador PI	Um outro tipo de compensador, utilizado para controle de velocidade.
		Exp. 9	Controladores PID: Abordagem prática	Uma breve introdução ao uso dos controladores mais populares dentro da Indústria, projetados e analisados segundo os métodos comumente empregados no meio industrial.
		Exp. 10	Atraso de transporte e controlador de modelo interno	Uma introdução à técnica de controladores de modelo interno, especialmente adequada a sistemas com atraso de transporte. Para esta experiência um atraso de transporte é artificialmente introduzido no servomecanismo

Tabela 1: Esquema geral do curso de Laboratório de Controle

## A documentação disponível

Este volume contém a documentação básica do curso. Ele é composto de uma Introdução seguida de uma sequência de nove experimentos. Cada uma dessas experiências trata de um assunto diferente e todas elas contêm tópicos comuns:

- i) *Introdução*. Um breve parágrafo descrevendo os objetivos do experimento.
- ii) *Descrição do problema*. Uma ou mais seções, que conforme o caso, apresentam alguns fundamentos teóricos necessários, metodologias de projeto a serem utilizadas, detalhamento de objetivos a serem alcançados, aspectos técnicos, etc.
- iii) *Atividades*. Uma sequência de atividades a serem realizadas pelos alunos. Algumas experiências dividem as atividades em *atividades prévias* (que devem ser feitas antes da realização da experiência, usualmente envolvendo preparação para o experimento em si), *atividades em aula* (implementações, medidas, etc. a serem realizadas em laboratório), *apresentação de resultados* (listando dados e resultados que deverão estar disponíveis para as experiências seguintes) e *relatório* (listando análises, etc. que devam ser incluídas nos relatórios). Mais do que uma sequência de meras tarefas, os itens relacionados servem de guia para a realização dos objetivos gerais das experiências. São passos a serem seguidos para a obtenção de resultados gerais, e não constituem um fim em si mesmo. Por exemplo, a primeira experiência contém cinco itens, enumerados de (a) a (e). Toda essa série de itens possui um único objetivo (além é claro do objetivo colateral de aprender o funcionamento dos equipamentos), que é a obtenção de um modelo matemático para o servomecanismo. Esse modelo será utilizado posteriormente para se fazer projetos de controladores, bem como para avaliar seu desempenho por meio de simulações. Note que cada item, considerado isoladamente, tem uma significância mínima.
- iv) *Bibliografia sugerida*. Contém uma lista referências de fácil acesso comentadas, que complementam os assuntos tratados em cada experiência. São boas fontes para aprofundar o aprendizado e reciclar o conhecimento obtido em outros cursos, bem como esclarecer eventuais dúvidas.

## Os relatórios

Três relatórios deverão ser redigidos ao longo do curso, a saber.

- *O Modelo do Servomecanismo*,  
que consolidará os resultados das Experiências de 1 a 4.  
Este relatório deve apresentar os diversos modelos matemáticos obtidos para o servomecanismo e compará-los. Uma análise crítica deve ser feita dos resultados. Questões tais como *qual modelo é melhor ou mais confiável e quando e porque um ou outro modelo deve ser utilizado* devem ser respondidas.
- *Controle de Posição*  
que consolidará resultados das Experiências 5, 6, 7 e 9.  
Este relatório deve apresentar as diferentes abordagens utilizadas para o controle de posição do servomecanismo e discutir suas características, vantagens, desvantagens, etc. Questões tais como *qual é o melhor algoritmo de controle num dado contexto e quando e porque um ou outro algoritmo deve ser utilizado* devem ser respondidas.
- *Controle de Velocidade*  
que consolidará resultados das Experiências 5, 8 e 9.  
Similar ao relatório acima, tratando de controle de velocidade.

Os relatórios devem ser breves, concisos e completos, e principalmente: devem ser escritos tendo-se em mente, do começo ao fim, que um objetivo, enunciado no começo, deva ser atingido ao final. No corpo do

relatório devem constar os passos necessários à obtenção desses resultados. Alguns pontos importantes na redação dos relatórios são apresentados abaixo:

- Um relatório não é a sequência de itens apresentada na apostila e nem depende diretamente dela. Um relatório deve ser um texto completo, com sustentação própria. Os itens da apostila são apenas guias para a realização das experiências, explicitando o que deve obrigatoriamente ser medido e o que deve ser comparado com o quê. Os relatórios devem visar objetivos, por exemplo obter um modelo matemático para o servomecanismo, ou projetar um controlador de posição e compará-lo com outros equivalentes.
- Repetir num relatório o texto das apostilas é desnecessário. Quando for necessário fazer menção a qualquer assunto tratado nas apostilas basta referenciá-las (e.g. “veja a equação (x.x) da Exp. Y”)
- Consolidar os dados obtidos sempre que possível na forma de gráficos e tabelas e faça figuras para explicar e exemplificar pontos mais complexos. É extremamente importante referenciar as figuras no texto, e não simplesmente deixá-las soltas. É também importante fazer títulos e legendas explicativos. Um título como “*gráfico de  $V \times T$* ” por exemplo, deixa muita margem a dúvidas. Algo como “*gráfico de  $V \times T$  para o sistema em malha fechada com controlador  $X$  e excitação degrau de amplitude  $3V$* ” é bem melhor.

## Entrega de documentos

A Tabela 2 a seguir apresenta a lista de documentos a serem entregues ao professor no decorrer do curso. Estes documentos podem ser *preparação de experimentos* (tarefas da seção *Atividades prévias* de cada experiência), *apresentação de resultados* (tarefas da seção *apresentação de resultados* de cada experiência) e *relatórios* (em número de três conforme a tabela).

Exp.	preparação	resultados	relatórios
1	-	-	
2	-	1 item: modelo Exp.1	
3	-	1 item: modelo revisado	
4	-	1 item: modelo Exp. 3	
5	-	-	Modelo do Servomecanismo
6	3 itens	2 itens: respostas ao degrau (posição/velocidade) Exp. 5	
7	2 itens	1 item: respostas ao degrau Exp. 6	
8	6 itens	-	
9	-	2 itens: resposta ao degrau Exp. 7 resposta ao degrau Exp. 8	
10	1 item	-	
Final do curso	-	-	Controle de Posição; Controle de Velocidade

Tabela 2: Entrega de documentos

# Experiência 1

## Familiarização com o equipamento - I

---

### 1.1 Introdução

O objetivo deste experimento é obter um modelo matemático para o servomecanismo existente no Laboratório de Controle do Departamento de Engenharia de Telecomunicações e Controle da EPUSP, identificando cada um dos parâmetros relevantes, bem como realizando uma primeira familiarização com alguns dos equipamentos do Laboratório.

### 1.2 O servomecanismo MS15 e a fonte de alimentação SP90

O conjunto formado por servomecanismo e fonte de alimentação é o principal equipamento do curso e será utilizado em todas as experiências.

#### 1.2.1 Análise física do servomecanismo

A Figura 1.1 mostra um diagrama do servomecanismo. Trata-se de um equipamento didático para a aplicação

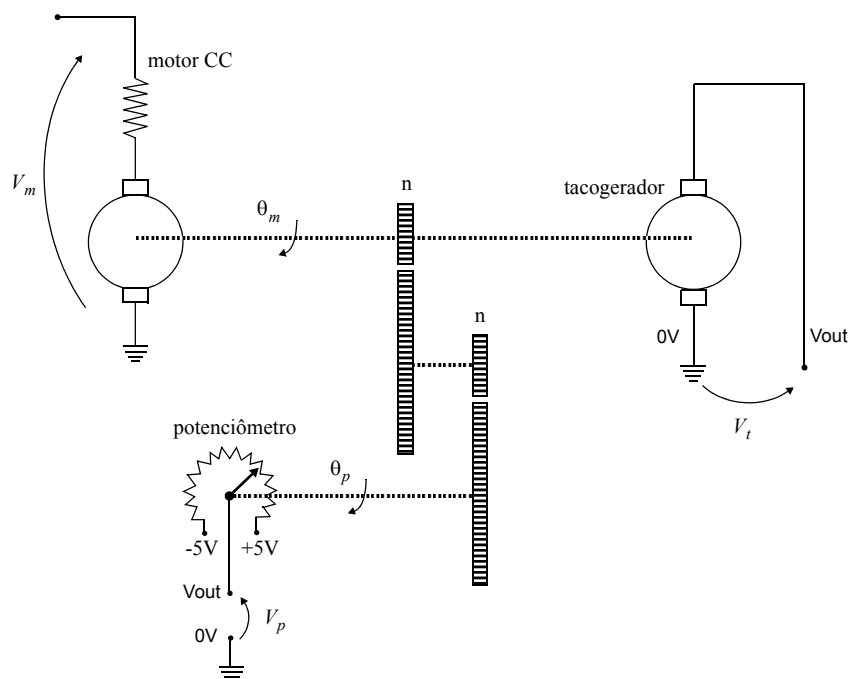
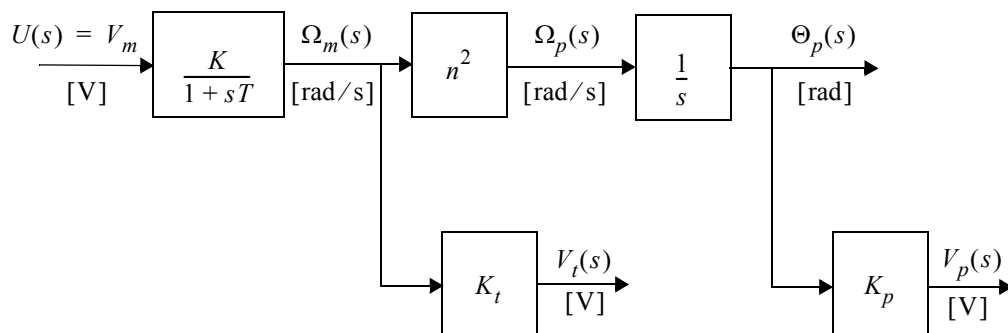


Figura 1.1: Esquema do servomecanismo

dos conceitos da Teoria de Controle. Compõe-se de um motor elétrico de corrente contínua controlado pela armadura e de sensores de posição e velocidade angulares. O eixo do motor e do tacogerador (o sensor de velocidade) está ligado ao eixo do potenciômetro (o sensor de posição) através de correias dentadas. Cada par de engrenagens, correspondente a uma correia possui uma constante de redução  $n = 1/3$ .

A Figura 1.2 apresenta um diagrama de blocos referente ao sistema da Figura 1.1. Note que o tacôme-



**Figura 1.2** Diagrama de blocos do servomecanismo

tro possui uma constante de proporcionalidade  $K_t$ , medida em  $V/(\text{rad/s})$  e que o potenciômetro também possui uma constante de proporcionalidade, denotada na figura por  $K_p$  e expressa em  $V/\text{rad}$ .

O equacionamento da Figura 1.2 é feito em termos de posições e velocidades angulares, mas como a monitoração do sistema, e consequentemente qualquer realimentação, é feita através dos sensores de velocidade e posição, convém incluir as constantes de proporcionalidade do tacômetro e do potenciômetro.

### 1.2.2 O painel de operação

A Figura 1.3 mostra o painel de operação do servomecanismo. Os bornes na parte superior do painel se destinam à alimentação do aparelho e conexão da fonte de tensão. O bloco denominado *motor drive input* se destina à alimentação da armadura do motor. O bloco abaixo, denominado *tachometer output* fornece a tensão gerada pelo tacômetro, e o último bloco *potentiometer output* dá acesso à tensão no potenciômetro.

Note que para que o servomecanismo opere aproximadamente segundo o modelo da Figura 1.2 é necessário que o freio eletromagnético seja desativado, o que é feito suspendendo-se a alavanca até a posição 0. No decorrer de todas as experiências, a menos quando mencionado explicitamente o contrário, o freio deve estar desativado. Além disso, verifique sempre se as chaves *motor drive* e *tachogenerator* estão nas posições corretas. A chave *motor drive* deve estar na posição  $V_{in}$  e a chave *tachogenerator* deve estar na posição  $V_{out}$ .

O mostrador digital localizado na parte inferior central do painel apresenta a velocidade angular do eixo do potenciômetro medida em rpm.

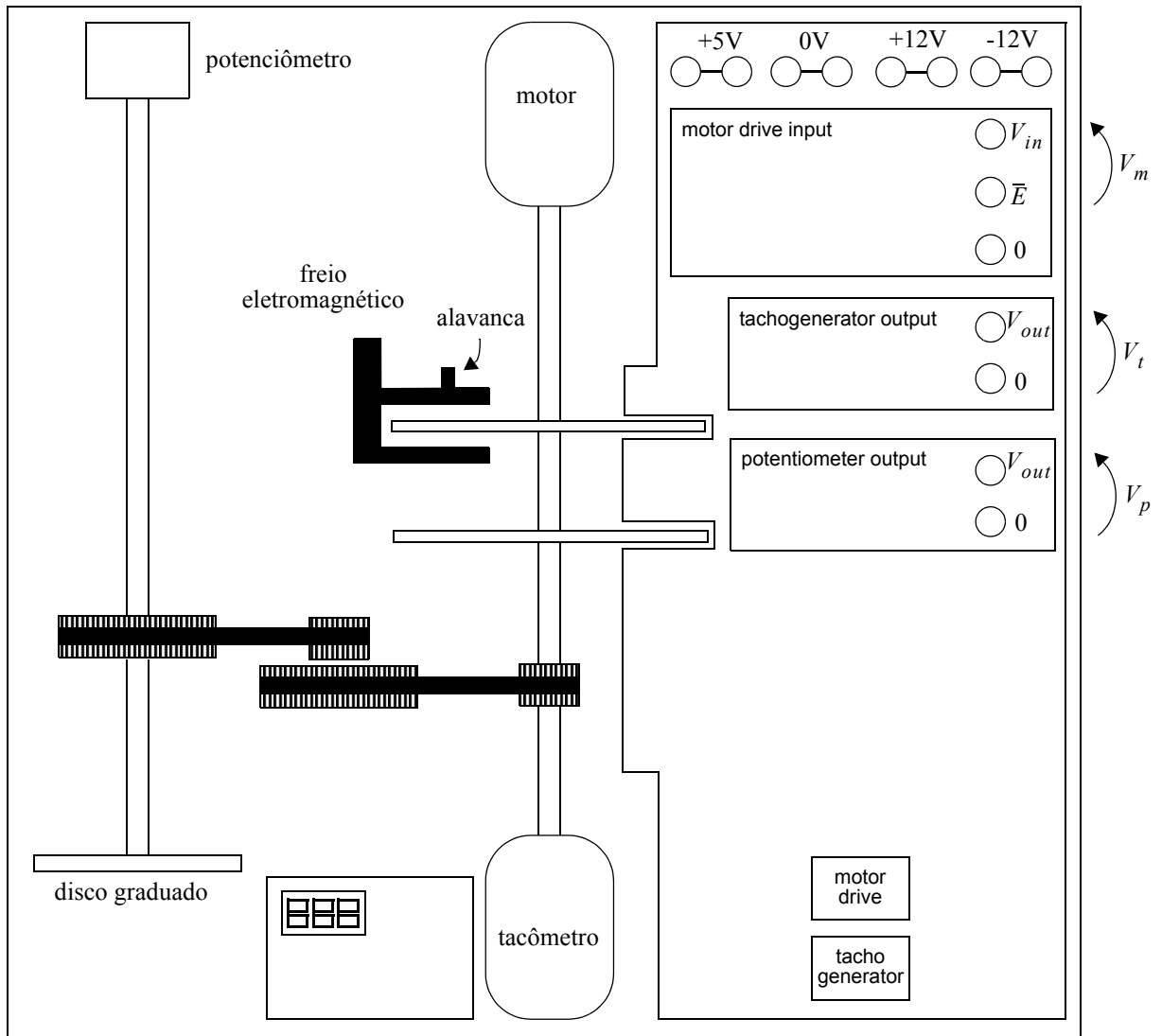
### 1.2.3 Conectando o servomecanismo e a fonte de alimentação

A Figura 1.4 mostra um diagrama para a ligação da fonte de alimentação ao servomecanismo. Essas ligações são apenas para se alimentar o aparelho, e ligações adicionais serão necessárias para se controlar o servomecanismo.

## 1.3 O sistema de aquisição de dados

### 1.3.1 O módulo Lynx AC1160-VA

O módulo Lynx AC1160-VA (veja a Figura 1.5) é produzido pela Lynx Tecnologia Eletrônica e possui 16 entradas analógicas que utilizam um conversor A/D de 16 bits operando em uma faixa pré-programada de  $\pm 10V$  com impedância de entrada de  $100k\Omega$ . Dessas, somente duas são utilizadas, denotadas por INP0 e INP1. As 2 saídas analógicas têm resolução de 12 bits, operam na faixa de  $\pm 10V$  e admitem uma carga mínima de  $2k\Omega$ , portanto muito cuidado deve ser tomado ao se conectar as saídas analógicas para não danifi-



**Figura 1.3** Painel do servomecanismo MS15

car o equipamento. As saídas analógicas são denotadas por OUT0 e OUT1.

O módulo se comunica com o computador através de uma conexão Ethernet 10BaseT de 10Mbit/s.

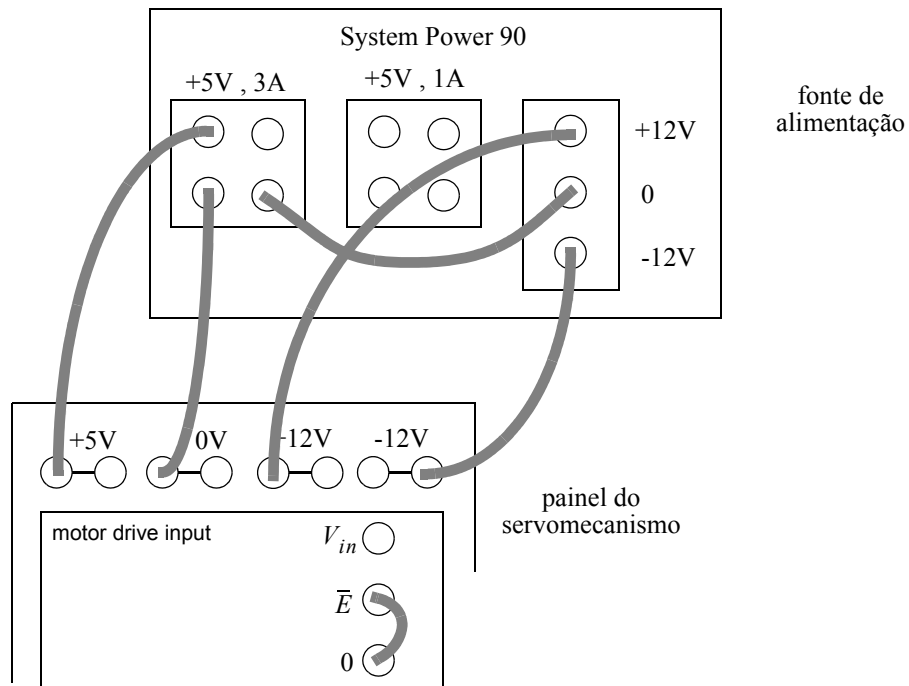
Adicionalmente o módulo possui entradas e saídas digitais e diferentes opções de sincronismo, porém esses recursos não serão utilizados no curso.

O módulo possui três indicadores luminosos (LEDs) no painel traseiro que apresentam informações diversas sobre seu funcionamento.

- |                       |   |
|-----------------------|---|
| a) ATV (LED verde)    | sinaliza o estado do módulo (veja a Tabela 1.6 a seguir); |
| b) LAN (LED amarelo)  | sinaliza a troca de mensagens na conexão Ethernet;        |
| c) LINK (LED amarelo) | sinaliza que a conexão Ethernet está ativada.             |

O principal indicador é o LED ATV. A Tabela 1.6 apresenta a codificação do estado do módulo em fun-





**Figura 1.4** Ligação da fonte de alimentação



(a) vista do painel frontal



(b) vista do painel traseiro

**Figura 1.5** O módulo Lynx AC1160-VA

ção das mensagens do indicador.

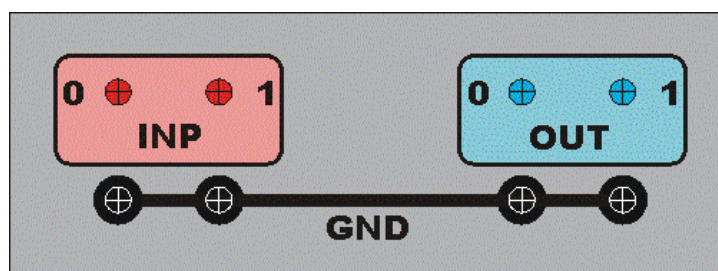
**Tabela 1.6** Indicador luminoso ATV

Mensagem	Significado	Observações
Aceso	Módulo ligado, mas não configurado.	Imediatamente após ser ligado.
OFF + 1P <sup>1</sup>	Módulo com configuração válida.	<i>Não esperado em situação normal.</i>
OFF + 2P	Módulo executando aquisição de dados.	Durante a operação.
OFF + 3P	Módulo em pausa na aquisição.	<i>Não esperado em situação normal.</i>
OFF + 4P	Módulo encerrou aquisição de dados.	Após o término da operação.
OFF + 5P (ou mais)	Erro de comunicação ou operação.	<i>Não esperado em situação normal. Pode indicar problemas na conexão de rede.</i>
Apagado	Módulo desligado ou em falha.	

1. OFF (pausa apagado) + 1P (1 piscada)

### 1.3.2 O módulo de conexão

Para se facilitar o manuseio do sistema, a conexão com o módulo Lynx AC1160-VA é feita através de um módulo de conexão, cujo painel frontal é mostrado na Figura 1.7.



**Figura 1.7** Painel de conexões

Os terminais de referência GND são conectados entre si e devem ser conectados aos terminais 0 do servomecanismo e da fonte de alimentação, conforme a Figura 1.8.

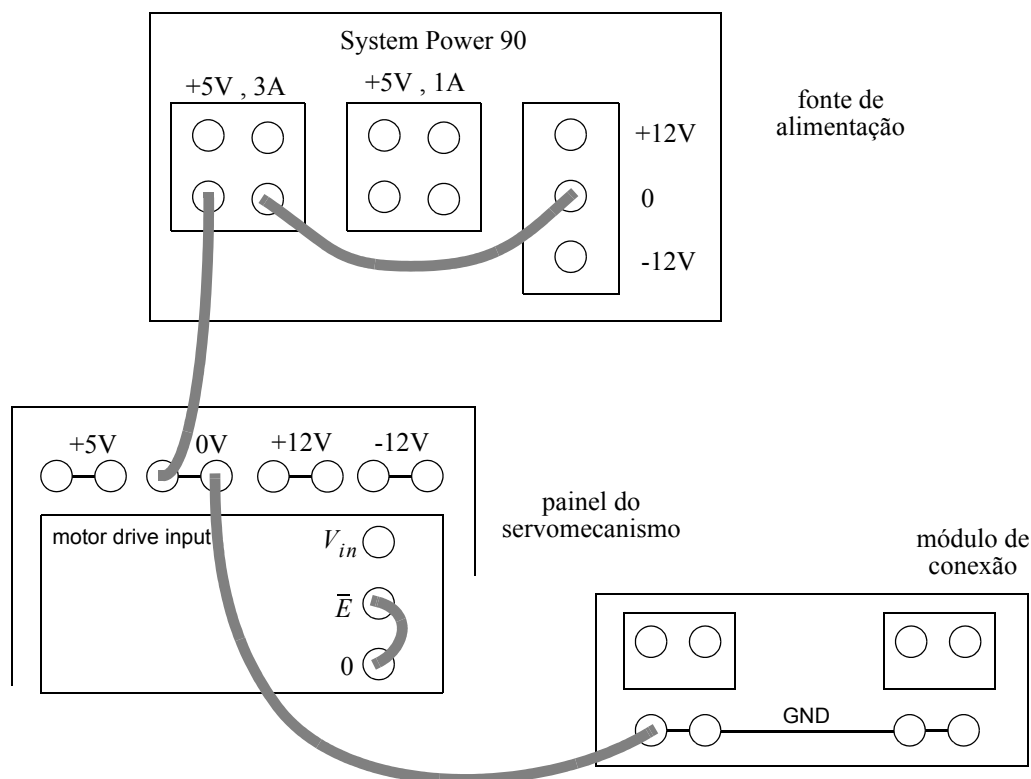
### IMPORTANTE



Jamais ligue os terminais OUT0 ou OUT1 diretamente a um terminal de terra ou aos terminais GND.

Jamais ligue os terminais OUT0 ou OUT1 a uma fonte de tensão (como um gerador de funções ou a rede elétrica).

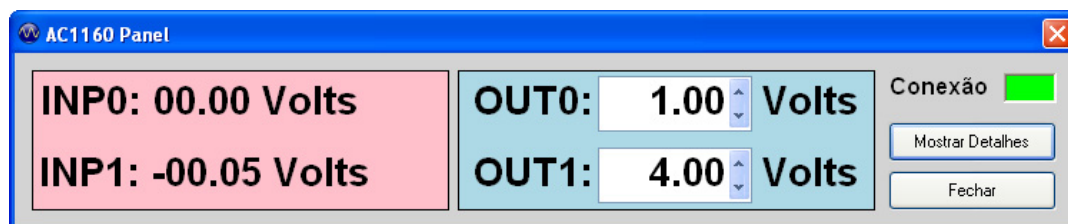
Jamais ligue os terminais INP0 ou INP1 a fontes de tensão acima da faixa de  $\pm 10V$  (como por exemplo a rede elétrica).



**Figura 1.8** Ligação do módulo de conexão (somente sinal de referência)

### 1.3.3 O aplicativo AC1160 Panel

Este aplicativo, disponível nos computadores do laboratório, permite o acesso às funções básicas do módulo Lynx AC1160-VA, isto é, permite ler na tela do computador os valores de tensão em cada um dos canais de entrada (de modo similar a um multímetro) e escrever valores nos canais de saída do módulo. Seu uso é bastante simples e dispensa maiores explicações. A tela principal do aplicativo é apresentada na Figura 1.9. Note a semelhança com o painel do módulo de conexão.



**Figura 1.9** O aplicativo AC1160 Panel

O indicador de conexão deve estar em verde para que o aplicativo opere corretamente.

## IMPORTANTE



O aplicativo AC1160 Panel não pode ser utilizado ao mesmo tempo que o Simulink (em modo de comunicação com o sistema de aquisição de dados), tal como será apresentado na Experiência 2. É necessário fechar o aplicativo para executar ensaios com o Simulink e vice-versa.

## 1.4 Atividades

### 1.4.1 Atividades em sala

O objetivo das atividades a seguir é a obtenção de um modelo matemático preliminar para o servomecanismo, isto é, identificar cada um dos parâmetros do diagrama da Figura 1.2.

**IMPORTANTE:** Sempre anote os números de identificação dos equipamentos que você utilizar. Espera-se que você utilize no decorrer de todo o curso sempre a mesma bancada e os mesmos equipamentos. Os modelos obtidos nesta e em futuras experiências valem apenas para o equipamento específico utilizado.

- Faça a conexão da fonte de alimentação ao servomecanismo como na Figura 1.4 e utilize o aplicativo AC1160\_PANEL para gerar tensões na faixa de -10 a 10 V. Obviamente é necessário conectar os terminais do módulo de conexão aos terminais correspondentes do servomecanismo.
- Levante a curva característica  $V_t$  (tensão no tacômetro) contra  $V_m$  (tensão de armadura). Identifique claramente as zonas de atrito e saturação.
- Utilizando o mostrador digital do painel do servomecanismo e a medida da tensão do tacômetro, obtenha o valor de  $K_t$  (em V/(rad/s)).

**ATENÇÃO:** O mostrador digital apresenta a velocidade angular no eixo do potenciômetro (isto é,  $\Omega_p$ ), e NÃO do eixo do tacômetro. Além disso, o valor é apresentado em rpm e não rad/s.

- Com o motor desacoplado (isto é, com o terminal  $V_{in}$  desconectado ou alimentado com 0 V), gire manualmente o eixo do motor e determine a característica V/rad, isto é, a constante  $K_p$ .

**IMPORTANTE:** NÃO gire o motor pelo disco graduado amarelo, pois isso pode comprometer a transmissão. Gire sempre pelos discos óticos do eixo do motor.

- Aplicando um degrau ao sistema, obtenha a função de transferência  $G_{t/u}(s) = \frac{V_t(s)}{U(s)}$ , isto é, determine os valores de  $K$  e  $T$ . Utilize o osciloscópio para coletar a curva.

**SUGESTÃO:** Se você não lembra como obter uma função de transferência de primeira ordem a partir da resposta ao degrau do sistema, procure em algum livro básico de controle (a bibliografia sugerida apresenta abaixo apresenta uma boa lista de livros básicos de controle) e dê uma olhada na Experiência 4.

### 1.4.2 Apresentação de resultados

O item desta seção deve ser apresentado ao professor para a realização da Exp. 2. em uma única folha de papel identificada com os nomes dos integrantes de seu grupo.

- Apresente o diagrama da Figura 1.2 com os valores de cada parâmetro identificados.

### 1.4.3 Relatório

Esta experiência deve ser incluída no relatório “O Modelo do Servomecanismo”.

## 1.5 Bibliografia sugerida

(DB98) Dorf, R. C.; Bishop, R. H. **Modern Control Systems**. Addison-Wesley, 8th. ed., 1998.

*A Seção 2.5 discute a modelagem de diversos sistemas físicos, inclusive motores elétricos.*

(FPE86) Franklin, G. F.; Powell, J. D.; Emami-Naeini, A. **Feedback Control of Dynamic Systems**. Addison-Wesley, 1st. ed., 1986.

*A Seção 2.4 apresenta uma boa discussão sobre modelagem de sistemas eletro-mecânicos (motores e geradores).*

(Kuo85) Kuo, B. C. **Sistemas de Controle Automático**. Prentice-Hall do Brasil, 4a. ed., 1985.

*A Seção 4.5 apresenta o funcionamento de alguns tipos de sensores, em particular detalhes sobre potenciômetros e tacômetros.*

(Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*Na Seção 2.6 é apresentado um modelo relativamente detalhado de um servomecanismo similar ao do laboratório. Respostas a degrau de sistemas lineares de primeira ordem são abordadas na Seção 4.3. A seção seguinte, que trata de sistemas de segunda ordem, utiliza um servomecanismo a título de exemplo.*

(CBS11) Castrucci, P. L.; Bittar, A.; Sales, R. M. **Controle Automático**. LTC, 1a. ed., 2011.

*A Seção 2.7 apresenta um sistema bastante similar ao do laboratório. Respostas a degrau de sistemas de primeira e segunda ordem são abordados nas Seções 3.4 e 3.5.*

O aplicativo AC1160 Panel foi escrito por Thiago Ragozo Contim.

RPM/2011a

# Experiência 2

## Familiarização com o equipamento - II

---

### 2.1 Introdução

Esta experiência conclui a etapa de familiarização com os recursos a serem usados nas próximas experiências, a saber:

Matlab for Windows versão R2008a (ou superior).

Simulink for Windows R2008a (ou superior).

O Matlab e o Simulink são programas para análise de dados, processamento numérico e simulação de sistemas dinâmicos. Eles serão usados em conjunto com o módulo Lynx (veja a experiência anterior) para coleta de dados e eventualmente implementação de controladores.

### 2.2 Recursos disponíveis

#### 2.2.1 O Matlab

O Matlab (de *MATrix LABoratory*) é um programa produzido pela The Mathworks, Inc. (maiores informações em <http://www.mathworks.com>), e a grosso modo serve para tratar matrizes e números complexos da mesma forma que uma calculadora trata números reais. Além disso ele possui recursos de programação, agindo como uma linguagem interpretada, semelhante a C ou Pascal, porém voltada para processamento numérico intensivo.

O sistema operacional utilizado no laboratório é o Windows XP, portanto você provavelmente encontrará o Matlab em algum lugar dentro da opção “Programas” do menu “Iniciar”. A interface do Matlab é composta basicamente de uma janela de comandos, com um *prompt* característico (>>). Além dessa janela de comandos, podem coexistir diversas janelas gráficas, denominadas *figures* e outras formas auxiliares de interface.

Mais informações podem ser encontradas no Apêndice A desta experiência.

#### 2.2.2 O Simulink

O Simulink, também produzido pela The Mathworks, Inc., é um programa destinado à simulação de sistemas dinâmicos, integrado ao Matlab. Sua interface com o usuário é totalmente gráfica, baseada na configuração e interconexão de blocos funcionais.

Mais informações podem ser encontradas no Apêndice B desta experiência.

### 2.3 Atividades

#### 2.3.1 Atividades em sala

As atividades a seguir têm por objetivo fixar a operação do uso do Matlab e do Simulink, programas que serão usados no decorrer de todo o curso, tanto como ferramenta para projeto de controladores, como para simulação de sistemas e análise de dados, e especialmente como ferramenta para implementação de programas a tempo real.



putador da bancada que você utilizar ao longo de todo o curso, porém NÃO há garantia nenhuma de que esses arquivos estarão no lugar em que você os deixou ou de que eles ainda existirão na próxima vez que você usar o laboratório, portanto **MANTENHA SEMPRE CÓPIAS DE TODOS OS SEUS ARQUIVOS.**

## 2.4 Bibliografia sugerida

Há diversos livros-texto de controle hoje que apresentam exemplos em Matlab e Simulink e são direcionados para o uso desses programas, como por exemplo (DB98). Há também diversos livros que tratam exclusivamente do Matlab e dos pacotes associados (não citados aqui), além é claro da sua documentação eletrônica.

(DB98) Dorf, R. C.; Bishop, R. H. **Modern Control Systems**. Addison-Wesley, 8th. ed., 1998.

(Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

RPM/2011a

Os *drivers* de comunicação com o módulo Lynx AC1160-VA foram escritos por Thiago Ragozo Contim.



# Apêndice A

## Tutorial do Matlab

---

### A.1 Introdução

O Matlab (de **Matrix Laboratory**) é um pacote matemático capaz de tratar matrizes, vetores e outras estruturas, com números reais ou complexos, da mesma forma que uma calculadora trata com valores escalares. Além disso, o Matlab possui recursos de programação estruturada que permitem a implementação de procedimentos mais complexos, além de visualização gráfica e simulação de sistemas dinâmicos (através do pacote integrado Simulink).

A janela básica para o uso do Matlab é a janela de comandos, identificada pelo *prompt* “>>”. A partir dessa janela é possível acessar todos os recursos do programa.

Este tutorial apresenta apenas os conceitos mais básicos. A consulta à documentação do sistema ou ao *site* do fabricante (<http://www.mathworks.com>), bem como aos diversos livros já publicados sobre o pacote é indispensável para um aprendizado mais aprofundado.

Este tutorial considera a versão 6.5 e posteriores. Todos os recursos apresentados são suportados em versões posteriores, mas alguns recursos não estão disponíveis para versões anteriores. .

### A.2 Operações básicas

#### A.2.1 Criando matrizes e vetores

Ao se digitar

```
>> a = 2
```

obtém-se a seguinte resposta:

```
a =
```

```
2
```

Este comando cria a variável “a” com o valor 2. Para se criar uma matriz, deve-se colocar os valores entre colchetes, separados por espaço (para separar colunas em uma mesma linha) ou por “;” (para separar linhas), assim

```
>> A = [1 2 3;3 2 1;5 0 0]
```

produz

```
A =
```

```
1     2     3
3     2     1
5     0     0
```

A variável “A” agora possui o valor matricial acima. Para evitar que o resultado do comando apareça na tela, basta terminá-lo com um “;”.

```
>> b = [10;5;3];
```

cria um vetor coluna, mas não imprime o resultado na janela de comandos. Isso é bastante útil caso o vetor ou matriz seja muito grande. Um exemplo de como criar vetores muito grandes é apresentado a seguir.

```
>> t = 0:0.01:10;
```

O comando acima cria um vetor em “t” da forma

```
[ 0 0.01 0.02 0.03 . . . 9.99 10 ].
```

Vetores e matrizes podem também ser facilmente criados a partir de outros vetores e matrizes. Por exemplo,

```
>> y = sin(t);
```

cria um vetor “y” contendo o seno dos elementos (supostos em radianos) do vetor “t”. Note que a função “sin” (seno trigonométrico) de um vetor resulta num novo vetor com o seno de cada elemento do vetor original.

#### VEJA TAMBÉM

Os comandos “who” e “whos” fornecem uma lista das variáveis criadas. Variáveis podem ser apagadas com o comando “clear”.

Os comandos “linspace” e “logspace” são também bastante úteis para a criação de vetores.

Para saber mais sobre um comando, basta digitar “help <comando>” ou “doc <comando>”.

O Matlab também trabalha com *strings*.

```
>> s = 'abc'
```

cria uma variável *string*. O sinal “ ’ ” delimita *strings*.

As variáveis criadas podem ser salvas em um arquivo e recuperadas posteriormente com o uso dos comandos “save” e “load”.

Para recuperar comandos já digitados, use as setas para cima e para baixo do teclado.

### A.2.2 Cálculos básicos

Os operadores básicos “+”, “-”, “\*”, “^”, etc. operam matricialmente, de modo que

- somente é possível adicionar e subtrair variáveis de mesmas dimensões;
- somente é possível multiplicar duas variáveis se o número de colunas da primeira for igual ao número de linhas da segunda;
- somente é possível exponenciar matrizes quadradas.

Uma exceção é uma soma com um número escalar, assim

```
>> r = [1 2 3] + 10
```

é uma operação permitida e resulta em

```
r =
```

Os operadores “/” e “\” são ligeiramente mais complexos e podem ser entendidos como sendo divisão à direita e divisão à esquerda.

$x = A \backslash b$  é a solução do sistema  $A * x = b$

$x = A / B$  é a solução do sistema  $x * A = B$

de modo que

```
>> x = A\b
```

fornece

```
x =
```

```
0.6000
```

```
0.0500
```

```
3.1000
```

De fato

```
>> A*x-b
```

fornece o resultado esperado:

```
ans =
```

```
0
```

```
0
```

```
0
```

Note que como não atribuímos o resultado desta operação a nenhuma variável, o Matlab automaticamente criou a variável “ans” (de *answer*).

Há também operadores que operam sobre cada elemento de um vetor ou matriz, por exemplo: “.\*”, “./”, “.^” e boa parte das funções, como “sin”, “cos”, “log”, “sqrt” (raiz quadrada), “exp”, etc.

```
>> ( [1 2 3] ).^2
```

resulta em

```
ans =
```

```
1
```

```
4
```

```
9
```

.

#### VEJA TAMBÉM

Algumas funções que operam matricialmente são “det” (determinante), “inv” (matriz inversa), “sqrtm”, “logm”, “expm” (versões matriciais das funções já vistas).

Note que “log” é o logaritmo neperiano. O logaritmo base 10 é denotado por “log10”.

### A.2.3 Números complexos e aritmética IEEE

O Matlab opera naturalmente com números complexos. O comando

```
>> f = sqrt(-2)
```

fornece

```
f =
```

```
0 + 1.4142i
```

Para facilitar a entrada de valores complexos, o Matlab fornece duas funções: “i” e “j”, ambas iguais a  $\sqrt{-1}$ . Assim

```
>> g = 2 + 3*i
```

resulta em (poderíamos também ter usado “j”)

```
g =
```

```
2.0000 + 3.0000i
```

**IMPORTANTE:** Caso sejam criadas variáveis “i” ou “j” pelo usuário, as funções originais deixam de ser acessíveis.

O Matlab utiliza o modelo aritmético dado pelo padrão ANSI/IEEE 754-1985. Uma decorrência importante disso é a introdução dos conceitos de “infinito” e de “não-número” (para valores indefinidos). O comando

```
>> w = 1/0
```

não produz uma mensagem de erro, apenas um aviso (ou não, conforme a versão do Matlab) e o resultado “infinito”.

```
Warning: Divide by zero.
```

```
w =
```

```
Inf
```

Ao passo que o comando abaixo produz o resultado “não-número”.

```
>> q = 0/0
```

```
Warning: Divide by zero.
```

```
q =
```

```
NaN
```

O uso desses conceitos é bastante conveniente, especialmente em seqüências complexas de cálculos, pois o Matlab não interrompe operações que podem eventualmente resultar válidas mesmo que um resultado intermediário divirja, como por exemplo:

```
>> 1/w + 10
```

```
ans =
```

```
10
```

(lembrando que  $1/\infty \rightarrow 0$ )..

#### VEJA TAMBÉM

As funções “inf” e “nan” permitem criar diretamente valores na situação “infinito” ou “não-número”.

## A.3 Manipulação de matrizes e vetores

### A.3.1 Operações com elementos de vetores ou matrizes

É possível acessar diretamente valores de matrizes ou vetores, bem como alterá-los ou eliminar submatrizes de uma matriz. A forma básica de se referir a um elemento de uma matriz é colocando os índices (linha e coluna) entre parênteses, separados por vírgula, ou seja:

```
>> A(2,3)
```

como a matriz “A” é dada por

1	2	3	}	coluna 3
3	2	1	}	linha 2
5	0	0	}	

temos

```
ans =
```

```
1
```

Podemos também extrair submatrizes da matriz “A”. Basta colocar vetores nos índices.

```
>> A([1 3],[2 3])
```

ou seja

1	2	3	}
3	2	1	}
5	0	0	}

, resultando em

```
ans =
```

```
2    3
```

```
0    0
```

```
>> A(:,2)
```

O símbolo “:” nesse caso pode ser entendido como “todos os elementos”, resultando em

```
ans =
```

```
2
```

```
2
```

```
0
```

Poderíamos substituir “:” por 1:3

```
>> A(1:3,2)
```

gerando o mesmo resultado.

Para eliminar uma linha, coluna ou submatriz, podemos proceder da seguinte forma:

```
>> A(3,:) = []
```

O símbolo “[]” pode ser lido como “matriz vazia”, tendo como efeito eliminar a terceira linha da matriz, resultando em

```
A =
```

```
1      2      3
```

```
3      2      1
```

### A.3.2 Concatenação

É possível também concatenar matrizes e vetores, de modo que para acrescentar uma nova linha à matriz “A”, basta o seguinte comando.

```
>> A = [ A ; [10 1 11] ]
```

```
A =
```

```
1      2      3
```

```
3      2      1
```

```
10     1     11
```

e

```
>> B = [A A]
```

resulta em

```
B =
```

```
1      2      3      1      2      3
```

```
3      2      1      3      2      1
```

```
10     1     11     10     1     11
```

**VEJA TAMBÉM**

Todos os exemplos deste tutorial usam matrizes bidimensionais, porém o Matlab também tem capacidade de tratar matrizes  $n$ -dimensionais com comandos específicos.

No caso mais simples em que a matriz se resume a um vetor, isto é, possui uma única linha ou coluna, podemos referenciá-lo com um único índice, ou seja

$$A(1, n) \rightarrow A(n)$$

$$A(m, 1) \rightarrow A(m)$$

Um comando extremamente útil para se localizar valores em um vetor ou matriz é o comando “find”. Vale a pena procurar saber mais sobre este comando.

**A.4 Polinômios**

Uma série de comandos e funções do Matlab, particularmente úteis, tratam com polinômios, conforme veremos a seguir.

**A.4.1 Codificando polinômios em vetores**

A codificação é extremamente simples, basta colocar os coeficientes de cada grau do polinômio em ordem decrescente (incluindo todos os graus, mesmo que tenham coeficientes nulos) em um vetor. Por exemplo:

```
>> poli1 = [1 4 2 0 20];
```

equivale a  $x^4 + 4x^3 + 2x^2 + 20$ .

```
>> poli2 = [5 7 0 0];
```

equivale a  $5x^3 + 7x^2$ .

**A.4.2 Operações com polinômios**

Para somar ou subtrair polinômios é necessário ajustar as dimensões e então somá-los ou subtraí-los como vetores comuns.

```
>> soma = poli1 + [ 0 poli2 ]
```

É necessário concatenar zero ao polinômio “poli2” para que ambos tenham a mesma dimensão, resultando em

```
soma =
```

```
1     9     9     0    20
```

que é equivalente a  $x^4 + 9x^3 + 9x^2 + 20$ .

Para realizar o produto de polinômios (ou convolução), basta usar a função “conv”.

```
>> prod = conv(poli1, poli2)
```

```
prod =
```

```
5     27     38     14    100    140     0     0
```

equivalente a  $5x^7 + 27x^6 + 38x^5 + 14x^4 + 100x^3 + 140x^2$ .

Para extrair as raízes do polinômio acima, usamos a função “roots”.

```
>> roots(prod)
```

```
ans =
```

```

0
0
0.7881 + 1.3254i
0.7881 - 1.3254i
-2.7881 + 0.7988i
-2.7881 - 0.7988i
-1.4000
```

O comando apresenta todas as raízes, incluindo as raízes múltiplas e as complexas..

#### VEJA TAMBÉM

Divisão polinomial pode ser feita com o uso da função “deconv”.

### A.4.3 Ajuste de curva com polinômios

Suponha que queiramos ajustar os pontos do gráfico ao lado (feito com o Matlab, como veremos no próximo item) a um polinômio de determinada ordem.

Inicialmente criamos os vetores com os dados.

```
>> x = [0 1 2 3];
>> y = [3 4 7 12];
```

Usamos então o comando “polyfit” para ajustar um polinômio de 3ª. ordem.

```
>> p3 = polyfit(x,y,3)
```

```
p3 =
```

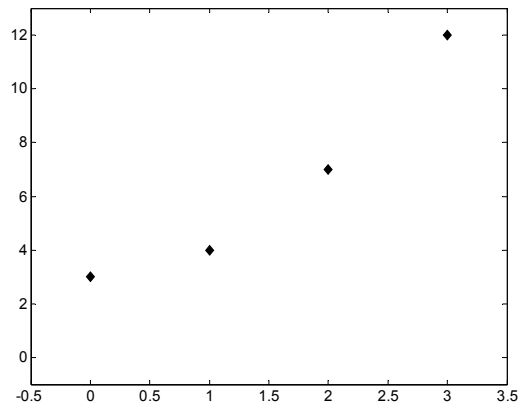
```
0.0000    1.0000    0.0000    3.0000
```

equivalente a  $x^2 + 3$ . O resultado nos sugere que um polinômio de grau 2 seria suficiente, assim

```
>> p2 = polyfit(x,y,2)
```

```
p2 =
```

```
1.0000    0.0000    3.0000
```





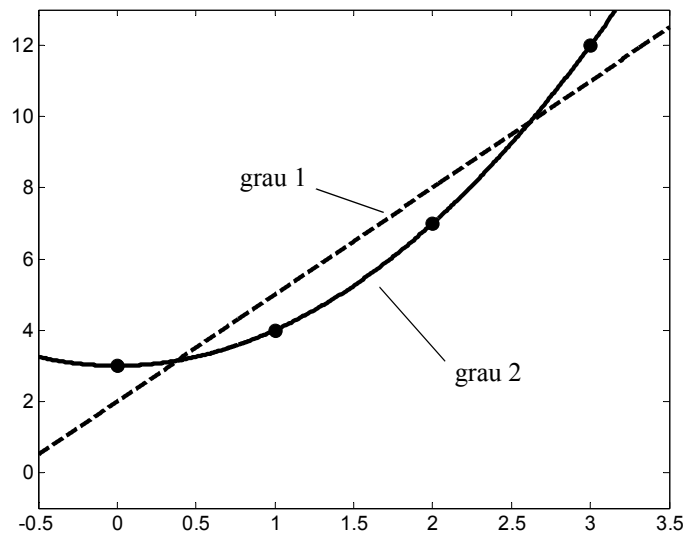
Caso quiséssemos ajustar uma reta aos pontos dados, bastaria usar um polinômio de grau 1.

```
>> p1 = polyfit(x,y,1)
```

```
p1 =
```

```
3.0000    2.0000
```

Os polinômios obtidos ajustam os dados conforme a figura abaixo.



#### VEJA TAMBÉM

O comando “polyval” permite calcular um dado polinômio em pontos escolhidos.

O Matlab também possui recursos de interpolação. Veja as funções “interp1” e “interp2”.

## A.5 Gráficos

O Matlab possui recursos gráficos extremamente sofisticados, porém neste tutorial somente os recursos mais simples serão explorados.

### A.5.1 Criando um gráfico simples

O comando mais simples para se gerar um gráfico é o comando “plot”. Inicialmente vamos gerar alguns vetores.

```
>> t = 0:0.01:10;
```

```
>> y = sin(t);
```

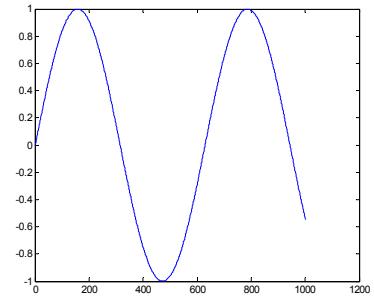
```
>> z = 0.5*cos(t);
```

```
>> w = y.*z;
```

O comando

```
>> plot(y)
```

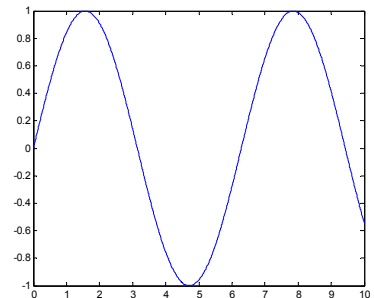
traça um gráfico com o vetor “y” na ordenada e os índices do vetor (indo de 1 até 1001), conforme a figura ao lado. Esta é a forma mais simples de se gerar um gráfico. Mais interessante seria traçar um gráfico do tipo  $t \times y$ .



Isso pode ser feito utilizando-se a forma

```
>> plot(t,y)
```

que gera um gráfico como este ao lado.



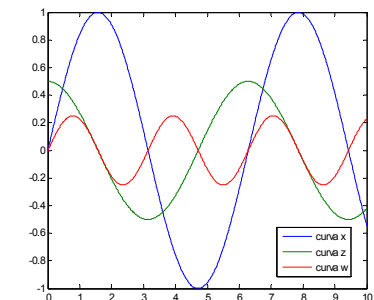
É possível traçar gráficos com diversas curvas. O comando

```
>> plot(t,y,t,z,t,w)
```

traça três curvas no mesmo gráfico. Para identificá-las, o comando “legend” é bastante útil.

```
>> legend('curva x','curva z','curva w')
```

identifica as curvas com uma legenda. Para mover a caixa de legendas para um lugar mais conveniente basta arrastá-la com o *mouse*.



O resultado final está no gráfico ao lado.

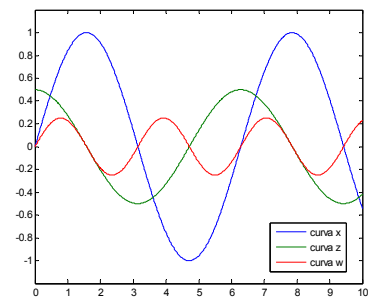
Para melhorar a aparência do gráfico podemos alterar a escala do gráfico, usando a função “axis”. O argumento da função é um vetor de quatro elementos, do tipo

$$\begin{bmatrix} x_{min} & x_{max} & y_{min} & y_{max} \end{bmatrix}$$

onde basta escolher os valores desejados ou “-inf” ou “inf” para deixar isso a cargo do Matlab. Assim

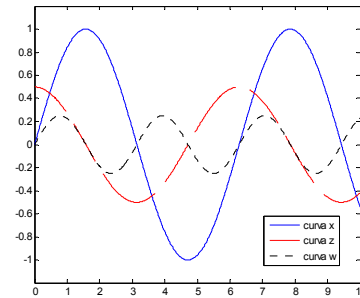
```
>> axis([-inf inf -1.2 1.2])
```

deixa o gráfico mais apresentável (ao lado).



Podemos alterar o tipo ou a cor das linhas diretamente no momento de traçar o gráfico. O Matlab possui diversas opções, para saber quais são consulte a documentação do programa. O comando a seguir produz uma linha azul (**blue**) contínua para a primeira curva, uma linha tracejada vermelha (**red**) para a segunda curva e uma linha pontilhada preta (**black**) para a terceira.

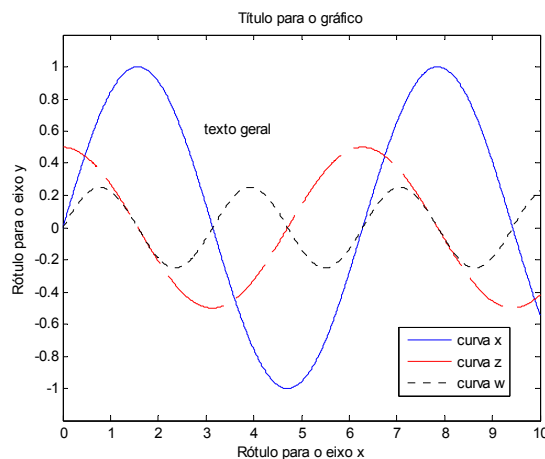
```
>> plot(t,y,'b-',t,z,'r--',t,w,'k:')
```



Podemos também incluir títulos, rótulos e textos diversos, utilizando os comandos

```
>> title('Título para o gráfico')
>> xlabel('Rótulo para o eixo x')
>> ylabel('Rótulo para o eixo y')
>> gtext('texto geral')
```

resultando em



#### VEJA TAMBÉM

Gráficos com escalas logarítmicas podem ser produzidos com os comandos “semilogx”, “semilogy” e “loglog”. Para sobrepor gráficos, basta utilizar o comando “hold on”. Os comandos a seguir farão com que os novos gráficos sejam traçados sobre os anteriores. “hold off” desativa a função. Da mesma forma, “grid on” e “grid off” servem para colocar grades nos gráficos. Também é possível traçar os pontos sem ligá-los, usando símbolos como “o” ou “+”. Novas janelas gráficas podem ser criadas com o comando “figure”.

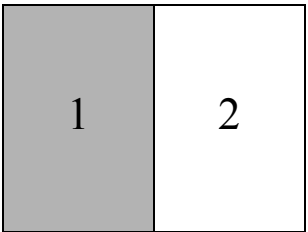
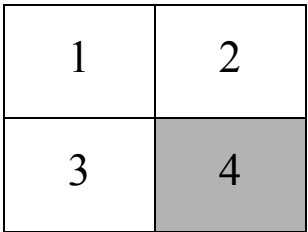
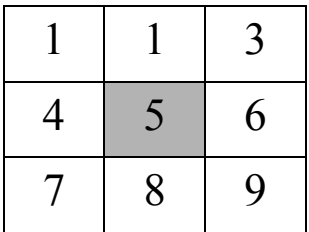
### A.5.2 Dividindo os gráficos

Os gráficos podem ser subdivididos com o uso do comando “subplot”. O comando possui a seguinte sin-

taxe.

```
subplot(1,c,n)
```

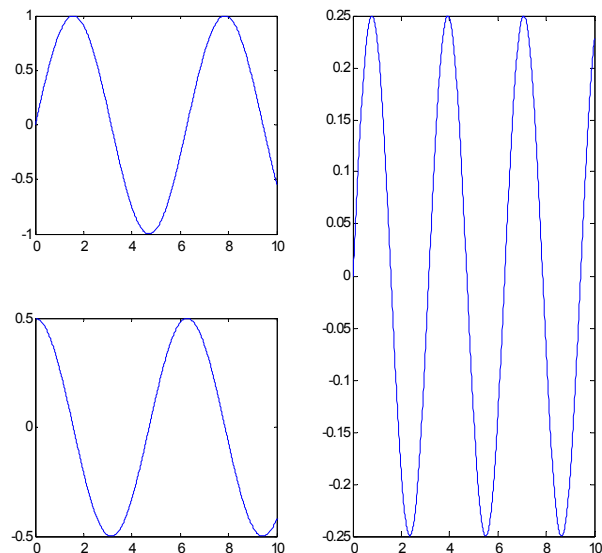
Este comando divide a janela em uma matriz de 1 linhas e c colunas, e ativa o n-ésimo elemento dessa matriz. Por exemplo:

subplot(1,2,1)	subplot(2,1,4)	subplot(3,3,5)
		

Divisões diferentes podem coexistir. Os comandos

```
>> subplot(2,2,1)
>> plot(t,y)
>> subplot(2,2,3)
>> plot(t,z)
>> subplot(1,2,2)
>> plot(t,w)
```

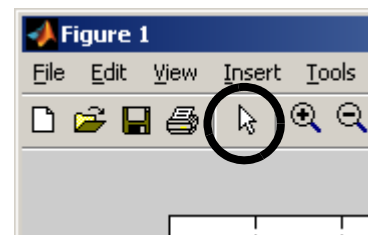
produzem o resultado do gráfico ao lado. Todos os comandos que se aplicam aos gráficos (como “title”, “xlabel”, etc. podem ser usados nos subgráficos.



### A.5.3 Propriedades avançadas dos gráficos

Todos os comandos vistos até o momento podem ser executados a partir da interface gráfica através de formulários e menus. Além disso, tarefas mais complexas, tais como alterar um tipo de linha, sua cor ou espessura podem ser realizados sem a necessidade de se refazer o gráfico. Outros recursos como alterar o tipo ou o tamanho dos caracteres, etc. também podem ser realizados através desta interface.

Para manipular os elementos de um gráfico, basta colocá-lo no modo de edição, o que é feito ativando o ícone de edição, localizado na janela gráfica.



**VEJA TAMBÉM**

O Matlab define a janela gráfica como um objeto, que possui propriedades que podem ser manipuladas pelo usuário. Não faz parte do escopo deste tutorial a manipulação de gráficos neste nível, porém o leitor interessado poderá consultar a documentação do Matlab a este respeito.

## A.6 O toolbox de controle

O Matlab possui uma biblioteca de funções específicas para projeto e análise de sistemas de controle lineares, o “Control System Toolbox”. Para mais informações basta digitar

```
>> help control
```

ou

```
>> doc control
```

Dentre as principais funções, pode-se citar “impulse”, “step”, “bode”, “rlocus”, entre outras.

Um aspecto importante do toolbox de controle é o objeto “Linear Invariante no Tempo” (LTI). Ele permite armazenar o modelo completo de um sistema linear invariante no tempo, assim como diversas de suas propriedades, em uma única variável. Além disso, diversas operações envolvendo sistemas podem ser realizadas diretamente com objetos LTI. Este é um tópico complexo demais para este tutorial, mas para sistemas descritos por funções de transferência (que são a regra neste curso), podemos criar um objeto LTI com o seguinte comando

```
>> sis = tf(num,den)
```

onde num e den são o numerador e denominador da função de transferência, descritos no padrão de polinômios, como visto acima. Definido sis, as diversas funções do toolbox podem ser utilizadas, tais como

```
>> step(sis); bode(sis)
```

dentre outras.

Com esses objetos pode-se fazer diversas operações. Por exemplo

```
>> sisp = sis1 + sis2
```

cria um sistema sisp com a soma das saídas dos sistemas sis1 e sis2 para uma mesma entrada (isso equivale a implementar sis1 e sis2 em paralelo num diagrama de blocos).

```
>> siss = sis1*sis2
```

cria um sistema siss com composição em série dos sistemas sis1 e sis2 (isso equivale a implementar sis1 e sis2 em sequência num diagrama de blocos)..

**VEJA TAMBÉM**

O toolbox de sistemas de controle é uma ferramenta extremamente poderosa para projeto e análise de sistemas de controle como os considerados neste curso. Recomenda-se um estudo detalhado deste toolbox e de sua documentação.

## A.7 Programação com o Matlab

Diversos comandos e funções não são codificados internamente no próprio Matlab, e estão definidos em arquivos de texto denominados *M-files*, escritos em uma linguagem de programação própria, denominada de “M”. A função “bode” por exemplo, é uma rotina cujo conteúdo pode ser visto com o comando

```
>> type bode
```

Os *M-files* podem ser de dois tipos: *functions* e *scripts*. *Scripts* são simplesmente seqüências de comandos, como acima, dispostos em um arquivo texto com extensão “.m”. *Functions* são estruturas mais complexas, que apresentam certas características básicas, que são as seguintes: i) Existe passagem de parâmetros entre o *workspace* do Matlab (*workspace* ou espaço de trabalho pode ser entendido a grosso modo como o conjunto de variáveis que pode ser observado com o comando `whos` na janela de comandos) e esta é a única troca de informações. Por exemplo, o comando `y=cos(x)` passa à função `cos` um argumento `x` e recebe como resposta `y`. Não há além disso nenhuma troca de informações. ii) O ambiente de cada função é independente, isto é, uma função não reconhece variáveis definidas fora dela, e outras funções ou o *workspace* não reconhecem variáveis definidas nela. Programadores de Pascal perceberão uma analogia clara entre *scripts* e *functions* do Matlab com *procedures* e *functions* de Pascal.

Um exemplo simples de função é o arquivo `log10.m`, que é uma função escrita para calcular logaritmos base 10 a partir de logaritmos naturais. O comando

```
>> type log10
```

mostra o conteúdo deste arquivo. Analise-o e entenda o seu funcionamento.

Espera-se que você seja capaz de escrever seus próprios *scripts* e *functions* para automatizar tarefas repetitivas e eventualmente realizar algumas mais complexas. A linguagem de programação do Matlab é muito semelhante a outras linguagens estruturadas, e se você estiver familiarizado com qualquer uma delas não terá problemas em fazer programas para o Matlab. Lembre de usar somente o seu diretório para armazenar programas e dados e de documentar bem os seus programas. É importante que você crie um cabeçalho inicial de comentários, pois o que o comando `help` faz nada mais é do que apresentar na tela o conteúdo deste cabeçalho. O comando

```
>> help lang
```

ou

```
>> doc lang
```

apresenta basicamente tudo o que é necessário saber para se iniciar em programação Matlab.

## A.8 Comentário final

Tenha em mente que o número de comandos e funções do Matlab, numa instalação normal, é usualmente da ordem de milhares, portanto é quase impossível memorizá-los todos, e é óbvio que este tutorial apresenta somente o mínimo necessário para se ter alguma desenvoltura com o Matlab. Use os comandos `doc`, `help` e `demo` sempre que estiver em dúvida sobre como fazer algo.

# Apêndice B

## Breve tutorial do Simulink


### B.1 Introdução

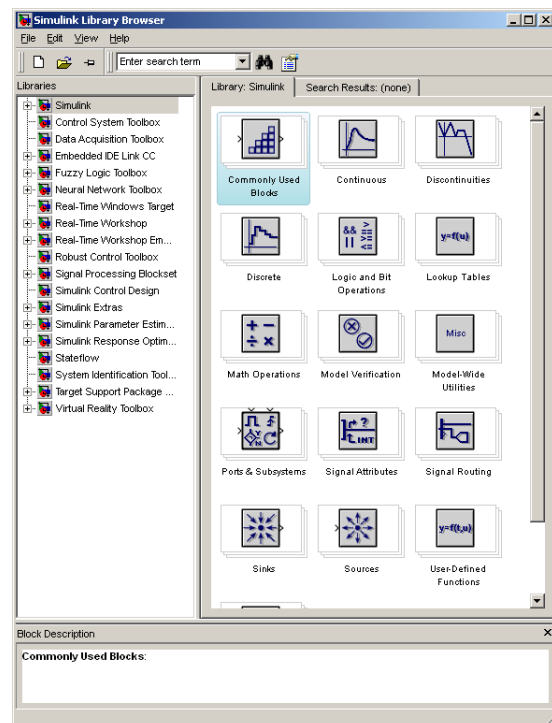
Este é um tutorial simplificado para o uso da versão 7.1 (R2008a) ou superiores do Simulink em conjunto com o módulo Lynx AC1160-VA. Ao contrário do Matlab, que é um programa com interface relativamente consolidada, o Simulink tem apresentado grandes variações em sua interface com a evolução das versões, de modo que a aparência do programa efetivamente utilizado no laboratório pode ser diferente da apresentada neste tutorial.

### B.2 Como iniciar o simulink

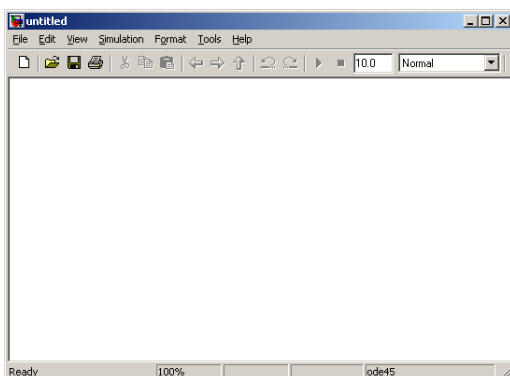
O simulink é acessado a partir do Matlab com o seguinte comando

```
>> simulink
```

ou a partir do botão com o ícone . A janela principal do Simulink tem a aparência da figura ao lado. Note que cada uma das bibliotecas (*Commonly Used Blocks*, *Continuous*, etc.) pode ser acessada a partir de um clique duplo do mouse, fazendo com que se tenha uma lista de blocos funcionais, que podem ser arrastados para os diagramas que você criar.



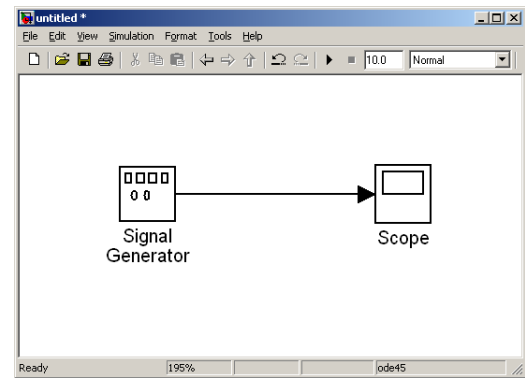
### B.3 Operação básica



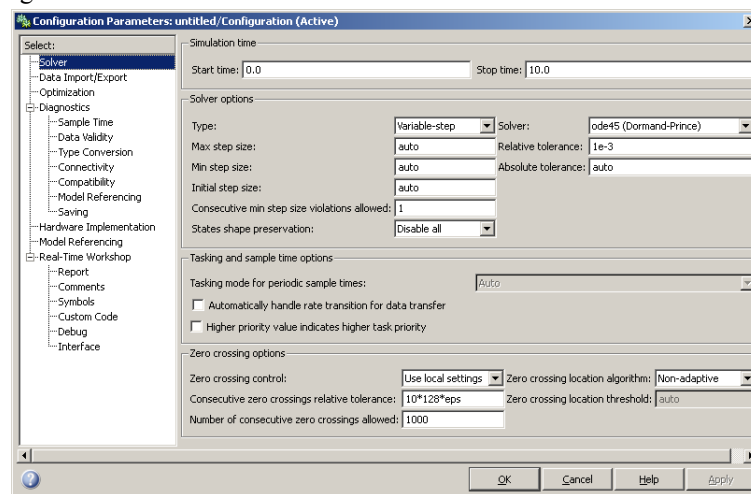
A idéia geral é que você utilize o Simulink para simular os sistemas que você mesmo criar, e principalmente para implementação de controladores em tempo real em conjunto com o módulo Lynx AC1160-VA. Para criar um novo sistema basta escolher FILE>NEW>MODEL no menu da parte superior da janela. Com isso uma nova janela em branco com uma barra de menus aparecerá na tela.

Como primeira tarefa você deverá criar um diagrama como este ao lado, copiando os blocos de suas respectivas bibliotecas. Faça isso arrastando-os com o mouse. A conexão entre os blocos é feita da mesma forma.

Uma vez que você tenha feito isso, use cliques duplos em cada bloco para acessar seus parâmetros. Configure o gerador de sinal para gerar uma onda dente de serra com uma frequência de 1 Hz e com uma amplitude de pico unitária. Configure o osciloscópio para exibir pelo menos uns 10 períodos da onda.



Antes de se iniciar a simulação é necessário entender um pouco o seu funcionamento. O simulink, como de resto todos os programas que fazem simulação de sistemas dinâmicos, se utiliza de algoritmos de integração para resolver a evolução temporal das equações do sistema numericamente. A janela abaixo, acessada a partir do menu SIMULATION>CONFIGURATION PARAMETERS ilustra como o usuário pode fazer uso dos diferentes algoritmos e de suas características.

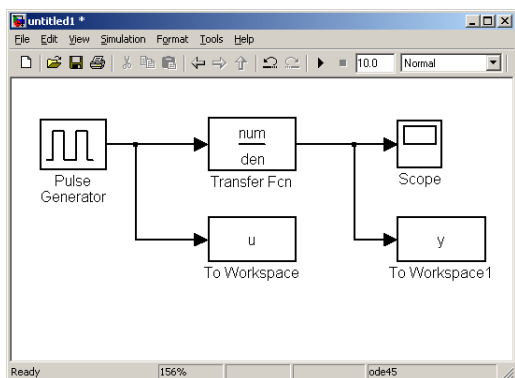


É importante ressaltar que o Simulink utiliza algoritmos de passo variável, isto é, o passo de integração varia entre os valores mínimo e máximo da janela de parâmetros conforme um algoritmo interno de escolha, relacionado com o valor do parâmetro de tolerância. Via de regra, os valores *default* desses parâmetros NÃO serão adequados para o seu sistema (qualquer que seja ele). Portanto verifique sempre se a simulação ocorre como esperado. Se você tiver problemas com o passo de integração, experimente fazer o passo mínimo igual ao passo máximo e diminuí-lo aos poucos.

Dos diversos algoritmos de integração, neste curso o preferido é o algoritmo de Euler com passo fixo. Mais detalhes sobre os algoritmos de integração, suas aplicações específicas e escolha de parâmetros você encontra na documentação eletrônica do Simulink.

Finalmente, para se iniciar a simulação basta usar a opção SIMULATION>START do menu principal.





Uma vez que você tenha cumprido a primeira tarefa, passe ao seguinte exercício.

Monte o diagrama ao lado. Note que você pode utilizar as variáveis do Matlab num e den como argumentos do bloco de função de transferência. Via de regra, qualquer bloco do Simulink aceita variáveis do *workspace* do Matlab como parâmetros de entrada.

Escolha a frequência do gerador de pulsos de modo que o sistema saia de uma situação de regime e atinja novamente o regime entre transições na entrada (isto é, use um

período suficientemente grande para que a resposta do sistema a cada pulso seja parecida com a resposta do sistema a um degrau). Faça a simulação para alguns períodos e verifique ao final, o conteúdo das variáveis u e y do Matlab (variáveis referidas nos blocos “To Workspace”).

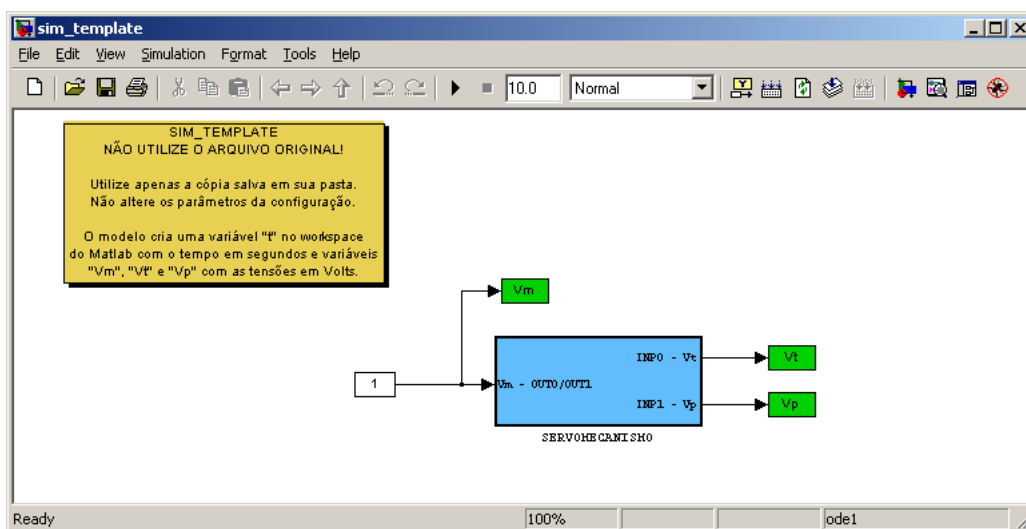
## B.4 O uso do Simulink em conjunto com o módulo Lynx AC1160-VA

### B.4.1 O bloco SERVOMECANISMO

O módulo Lynx pode ser usado em conjunto com o Simulink por meio de um bloco especial, que se encontra no modelo `sim_template` que pode ser acessado diretamente do Matlab (como de resto qualquer diagrama do Simulink) por meio do seguinte comando:

```
>> sim_template
```

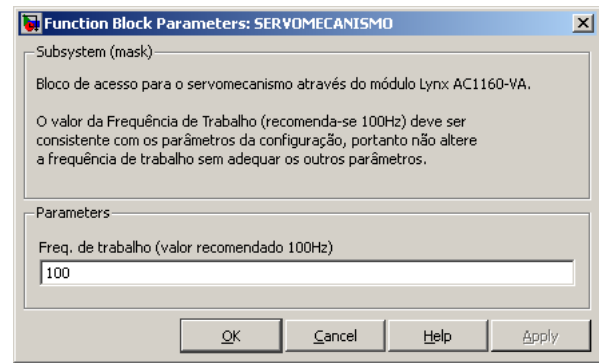
que deve carregar o diagrama abaixo.



O bloco `SERVOMECANISMO`, que no esquema de controle do laboratório cumpre o papel de planta, pode ser conectado a outros blocos e permite que se implemente sistemas de controle com o Simulink, bastando que se utilize os blocos adequados para montar a realimentação e eventuais compensadores. Note que para a configuração deste bloco funcionar corretamente é necessário conectar o terminal **INP0** do módulo de conexão ao tacômetro do servomecanismo, **INP1** ao potenciômetro, e qualquer dos terminais **OUT0** ou **OUT1** ao terminal da armadura do servomecanismo (assinalado **Vin** no painel do dispositivo).

| Um clique duplo no bloco fará com que seja exibida a janela de diálogo ao lado.

O único parâmetro deste bloco é a frequência de trabalho, ajustada para 100 Hz (não se recomenda alterar este valor). Esta frequência, combinada com um algoritmo de integração de passo fixo de 0,01 s (já configurado no diagrama *sim\_template*), faz com que este diagrama seja simulado em tempo real, isto é, com o cronômetro da simulação (o tempo simulado) em sincronia com o tempo físico (o tempo real).



## IMPORTANTE

O bloco SERVOMECANISMO não pode ser utilizado ao mesmo tempo que o aplicativo AC1160 Panel (vide Experiência 1).

É necessário fechar o aplicativo para executar ensaios com o Simulink e vice-versa.

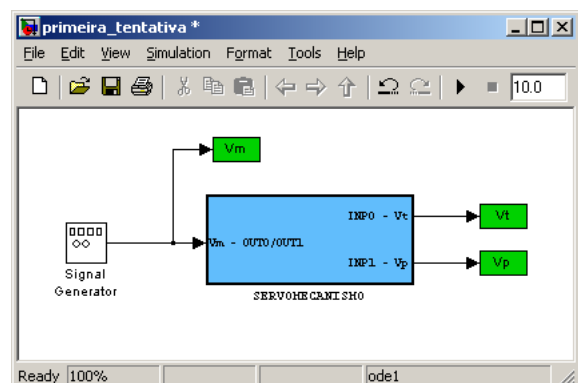
A simulação em tempo real requer que uma série de condições sejam satisfeitas, principalmente as seguintes:

- Que o sistema operacional, ou outros aplicativos não interrompam ou bloqueiem a execução da simulação. No Windows, não é possível garantir que tal não vá ocorrer, mas sob condições controladas (i.e. sem nenhum outro aplicativo ativo durante a simulação e sem intervenções do usuário), espera-se um desempenho adequado;
- Que o computador seja capaz de efetuar todas as operações necessárias para o cálculo de um passo da simulação no tempo alocado. De um modo geral, a capacidade de cálculo dos computadores do laboratório permite que isso seja satisfeito com bastante folga, porém tarefas mais complexas ou que requeiram acesso ao disco ou ao vídeo podem comprometer a sincronização e devem ser evitadas. Exemplos típicos são:
  - Traçado de gráficos durante a simulação, incluindo o uso de blocos do tipo Scope;
  - Acesso ao disco rígido, incluindo o uso de blocos To File e From File;
  - Exibição de valores na tela, incluindo o uso de blocos do tipo Display;

Caso o sistema perca a sincronização com o tempo real, uma mensagem de aviso (código R501) é enviada ao Matlab.

Monte o diagrama ao lado. Escolha uma onda senoidal de frequência baixa, por exemplo 1 Hz (equivalente a aproximadamente 6,2832 rad/s) e ligue a saída da placa ao osciloscópio.

Verifique se a simulação ocorre em tempo real e se não há avisos ou mensagens de erro na janela de comandos do Matlab.





## IMPORTANTE

Não arraste o bloco SERVOMECANISMO para outros diagramas. Para criar um diagrama que use este bloco, salve o diagrama `sim_template` com outro nome e use-o. A razão para isso é que a correta operação do diagrama depende de uma série de configurações incluídas em `sim_template`.

### B.4.2 Diagnóstico do módulo Lynx AC1160-VA

O diagnóstico do módulo Lynx AC1160-VA deve ser feito inicialmente a partir de seus indicadores luminosos e de seus estados típicos. Quando ligado, devemos ter a seguinte configuração: ATV aceso; LAN apagado (ou piscando esporadicamente); LINK aceso. Após o início da simulação devemos ter a seguinte configuração: ATV (OFF+2P); LAN aceso; LINK aceso. Após o término da simulação devemos ter: ATV (OFF+4P); LAN apagado (ou piscando esporadicamente); LINK aceso.

Qualquer configuração diferente das acima indica problemas no sistema, e deve ser reportada ao responsável.

Em qualquer situação, se o indicador LINK estiver apagado isso é uma indicação de ausência de conexão entre o computador e o módulo, o que impede o correto funcionamento do sistema.

### B.4.3 Diagnóstico de software

O sistema de acesso ao módulo emite mensagens de erro de dois tipos básicos: erros de parâmetros, denotados por PXXX e erros de runtime, denotados por RXXX. Erros de parâmetros interrompem obrigatoriamente a execução do programa, ao passo que erros de runtime apenas produzem mensagens de advertência na tela de comandos do Matlab. Apesar de não interromperem a execução do código, erros de runtime indicam situações que devem obrigatoriamente ser corrigidas.

Nem todas as possibilidades de uso incorreto das rotinas são contempladas pelos códigos acima, de modo que configurações não previstas podem gerar erros não listados neste documento.

Há dois módulos funcionais distintos no software, o módulo SRT (Soft Real Time) que controla a sincronização com o tempo real e o módulo DAS (Digital Acquisition System) que controla a comunicação com o módulo Lynx AC1160-VA.

A Tabela 1 abaixo apresenta a lista de erros de parâmetros.

**Tabela 1:** Erros de parâmetro

SRT		
erro	etapa	comentário
P401	configuração	Ausência do timer HRPC. Indica uma incompatibilidade do hardware do computador com o sistema de acesso ao módulo.
DAS		
erro	etapa	comentário
P402	configuração	Este erro indica uma falha interna do software ou problema de integração com o MS-Windows.

A Tabela 2 apresenta os erros de runtime. Estes erros não interrompem a execução do programa, porém

indicam problemas que devem ser obrigatoriamente abordados para o seu correto funcionamento.

**Tabela 2:** Erros de runtime

<b>DAS</b>		
erro	etapa	comentário
R002	configuração	Estes erros indicam falhas internas do software ou problemas de integração com o MS-Windows. A ocorrência de qualquer destes erros impede a correta operação do sistema.
R003	configuração	
R004	configuração	
R005	configuração	
R006	configuração	
R007	configuração	Este erro pode indicar uma falha na conexão de rede com o módulo Lynx AC1160-VA ou falha de hardware no módulo.
R008	configuração	Estes erros indicam falhas de hardware no módulo Lynx AC1160-VA.
R009	configuração	
R010	configuração	
R011	configuração	Este erro pode indicar uma falha na conexão de rede com o módulo Lynx AC1160-VA ou falha de hardware no módulo.
R012	configuração	Este erro indica uma falha interna do software ou problema de integração com o MS-Windows.
R013	configuração	Este erro pode indicar uma falha na conexão de rede com o módulo Lynx AC1160-VA ou falha de hardware no módulo.
R014	configuração	Este erro indica uma falha interna do software ou problema de integração com o MS-Windows
R015	configuração	Erro inesperado de software (sem diagnóstico).
R201	execução	Este erro indica uma falha interna do software ou problemas de integração com o MS-Windows.
R301	finalização	Estes erros indicam falhas internas do software ou problemas de integração com o MS-Windows.
R302	finalização	Este erro pode indicar uma falha na conexão de rede com o módulo Lynx AC1160-VA ou falha de hardware no módulo.
R303	finalização	Estes erros indicam falhas de hardware no módulo Lynx AC1160-VA.
R304	finalização	
R309	finalização	
R310	finalização	Erro inesperado de software (sem diagnóstico).
<b>SRT</b>		
R501	execução	Este erro indica perda de tempo real. As causas mais frequentes para esse erro são as seguintes: (i) Frequência de amostragem alta demais para a quantidade de cálculos determinada (neste caso o erro R501 ocorre com frequência elevada, possivelmente em todos os períodos de amostragem); (ii) Outros processos interferem com a execução do programa atrasando sua execução (neste caso o erro R501 ocorre esporadicamente).

#### **B.4.4 Problemas e dúvidas frequentes**

Esta seção se encontra em construção. O histórico de utilização do sistema ainda não permite avaliar com clareza quais são os problemas mais frequentemente encontrados.

a) **Algum canal de entrada (INP0, INP1) não coleta nenhum sinal ou coleta sinais estranhos.**

Diagnóstico 1: Verifique se o canal de entrada não se encontra desconectado. Canais de entrada desconectados podem copiar sinais de outros canais por indução (esta parece ser uma característica do hardware do módulo Lynx AC1160-VA).

Diagnóstico 2: Verifique o aterramento. O módulo e o sistema conectado a ele (a planta) devem ter referências comuns.

# Experiência 3

## Identificação I: Resposta em frequência

---

### 3.1 Introdução

Esta experiência trata da obtenção de um modelo matemático para o servomecanismo MS15 por meio de sua resposta em frequência. Tanto o Matlab como o Simulink serão utilizados para coleta e análise de dados, e a resposta em frequência será obtida com o uso de algoritmos de otimização.

### 3.2 Conceito de resposta em frequência

Seja  $U(j\omega)$  a representação complexa (módulo e fase) de um sinal senoidal de frequência angular  $\omega$ . Admita que este sinal excite um sistema linear estável e invariante no tempo. Uma vez que o sistema é estável, há o estabelecimento de um regime permanente com um sinal senoidal  $Y(j\omega)$  na saída. A resposta em frequência  $G(j\omega)$  do sistema é definida como o quociente entre as representações complexas da saída e da entrada, isto é:

$$G(j\omega) = \frac{Y(j\omega)}{U(j\omega)}. \quad (3.1)$$

A resposta em frequência se relaciona com a função de transferência  $G(s)$  do sistema por meio da expressão

$$G(j\omega) = G(s)|_{s=j\omega}. \quad (3.2)$$

Comumente a resposta em frequência é expressa por meio de *diagramas de Bode*<sup>1</sup>, compostos por dois gráficos, o primeiro exibindo o módulo de  $G(j\omega)$  (em escala logarítmica ou em dB) contra a frequência angular  $\omega$ , em escala logarítmica, e o segundo exibindo a fase de  $G(j\omega)$  (em escala linear) contra  $\omega$  em escala logarítmica.

Uma vez que se conheça a resposta em frequência do sistema (e um dos objetivos desta experiência é exatamente obter um gráfico dessa resposta) pode-se obter a sua função de transferência, como se verá a seguir.

### 3.3 Obtenção da resposta em frequência

Dois modos são apresentados para se obter a resposta em frequência de um sistema. Um modo baseado em inspeção visual e outro baseado em métodos numéricos. Em ambos os casos é necessário alimentar o sistema com uma onda senoidal e verificar o sinal na saída do sistema após o regime estacionário ter sido atingido. Os modos são os seguintes.

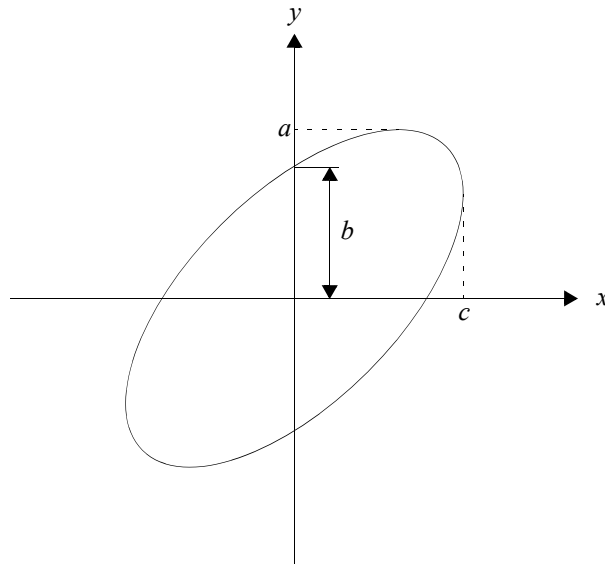
#### 3.3.1 Figuras de Lissajous

Utilizando um osciloscópio em modo XY, e injetando o sinal de entrada  $u(t)$  no canal X e o sinal  $y(t)$  no canal Y, obtém-se na tela do osciloscópio uma figura característica (tais figuras são denominadas *figuras de Lissajous*<sup>2</sup>). Para sistemas lineares estáveis e invariantes no tempo, se  $u(t)$  for um sinal senoidal, então  $y(t)$ , após ser atingido o regime permanente, também será um sinal senoidal de mesma frequência, e a figura obtida

---

1. Introduzidos por Hendrik W. Bode (1905-1982), dos laboratórios Bell, na década de 30.

será uma elipse, como na Figura 3.1.



**Figura 3.1** Figura de Lissajous

Nesse caso o módulo da resposta em frequência ou ganho  $M(\omega)$  é dado por

$$M(\omega) = \frac{|Y(j\omega)|}{|U(j\omega)|} = \frac{a}{c} \quad (3.3)$$

e a fase  $\phi(\omega)$  é dada por

$$\phi(\omega) = \angle\left(\frac{Y(j\omega)}{U(j\omega)}\right) = \sin^{-1}\left(\frac{b}{a}\right), \quad (3.4)$$

com  $a$ ,  $b$  e  $c$  indicados na Figura 3.1.

### 3.3.2 Otimização numérica

Incluídos entre os arquivos que você deve ter copiado para o seu diretório, estão três arquivos contendo funções do Matlab, `senofit.m` (a função principal), `senofit_erquad.m` e `senofit_ic.m` (funções auxiliares). A função `senofit` serve para ajustar, por meio de um algoritmo de otimização numérica, uma senóide a uma curva qualquer. O algoritmo age no sentido de obter uma senóide que aproxime uma dada curva de modo a minimizar a integral quadrática da diferença entre elas. Uma vez que se obtenham aproximações para os sinais de entrada e saída, basta calcular o quociente entre as amplitudes das senóides para se obter o módulo da resposta em frequência e calcular a diferença entre as fases das senóides para se obter a fase da resposta em frequência.

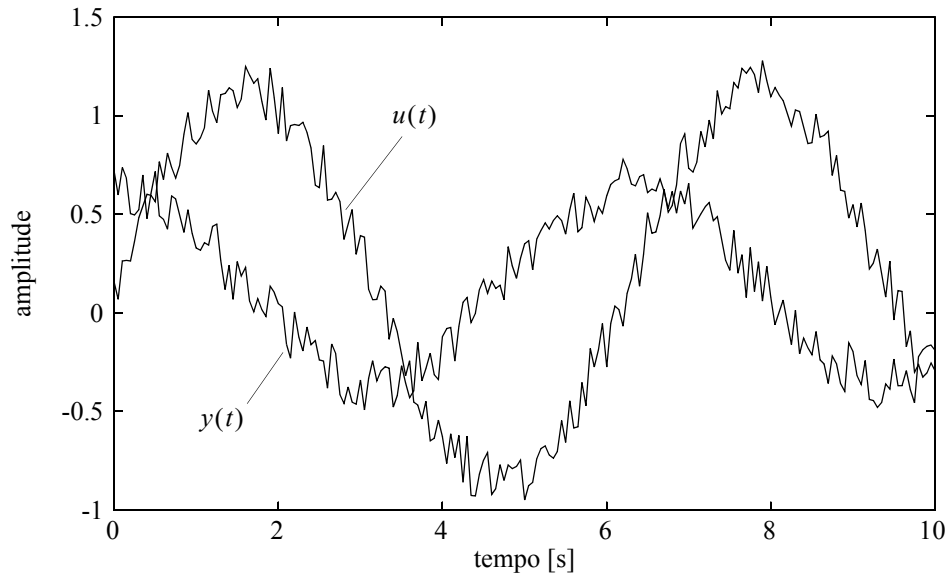
Como exemplo, imagine que tenham sido coletados no computador os seguintes dados, referentes à entrada e à saída de um sistema qualquer (Figura 3.2). A função `senofit`, aplicada a cada uma das curvas fornece os seguintes resultados:

$$u(t) \cong 1,0492 \sin(1,015t + 0,0258) \quad (3.5)$$

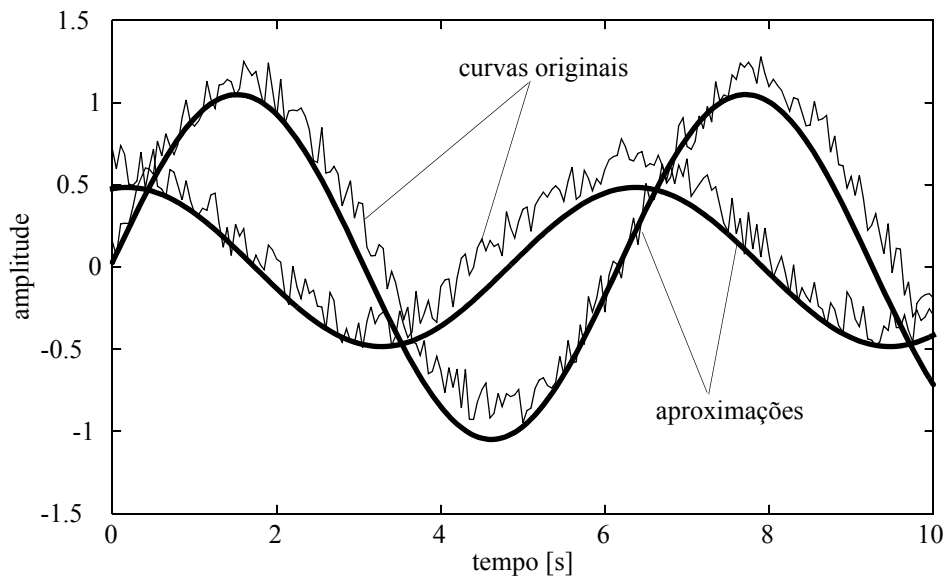
$$y(t) \cong 0,4839 \sin(1,015t + 1,3854) \quad (3.6)$$

que correspondem às linhas mais espessas da Figura 3.3.

2. Em homenagem ao físico francês Jules A. Lissajous (1822-80) que conduziu estudos extensivos sobre o assunto.



**Figura 3.2** Exemplo de sinais coletados para obtenção de resposta em frequência



**Figura 3.3** Sinais originais e aproximações senoidais

De (3.5) e (3.6) tem-se que  $M(\omega) = 0,4839/1,0492 = 0,4612$  (obtido do quociente entre as amplitudes da entrada e da saída) e  $\phi(\omega) = 1,3854 - 0,0258 = 1,3597$  rad (obtido da defasagem entre os sinais de entrada e saída) são o módulo e a fase da resposta em frequência para  $\omega = 1,015$  rad/s.

Realizando os mesmos cálculos para diferentes valores de  $\omega$  obtém-se um conjunto de pontos que permite traçar os gráficos da resposta em frequência.

No Matlab, o comando

```
>> help senofit
```

exibe as instruções de uso da função. É importante que você verifique o conteúdo dessa função e compreenda o seu funcionamento.



Algumas observações sobre o funcionamento da função `senofit` que podem ser úteis: Quanto maior o número de períodos considerados, maior é o número de pontos de mínimo locais que o problema apresenta, o que prejudica a obtenção da solução. Procure não coletar um número excessivo de períodos, um mínimo de um e um máximo de três é o ideal. A convergência é extremamente sensível às condições iniciais do algoritmo (em particular à fase inicial escolhida e em menor escala à frequência), portanto escolha-as com muito cuidado.

### 3.4 Obtenção da função de transferência

Os procedimentos descritos acima fazem com que sejam obtidos gráficos da resposta em frequência, porém do ponto de vista de Controle, é interessante que seja disponível uma equação que descreva a função de transferência. Tal equação pode ser obtida a partir dos gráficos da resposta em frequência de diversas maneiras, das quais destacaremos duas.

#### 3.4.1 Ajuste manual por assíntotas

Como consta de qualquer bom livro básico de Controle (veja a bibliografia sugerida), diagramas de Bode podem ser razoavelmente aproximados por meio de assíntotas. Não cabe discutir aqui como isso é feito (para tanto consulte algum desses livros), mas você deve ser capaz de, a partir de curvas de módulo e fase, obter aproximações assintóticas graficamente e associá-las a uma função de transferência.

A identificação das assíntotas é tarefa simples quando as frequências de canto apresentam-se razoavelmente espaçadas (isto é, distantes entre si por uma década ou mais).

Por exemplo, a função de transferência  $G(s) = \frac{1}{s+1}$  pode ser aproximada por meio de assíntotas, a partir da curva de resposta em frequência (módulo) da figura abaixo. :

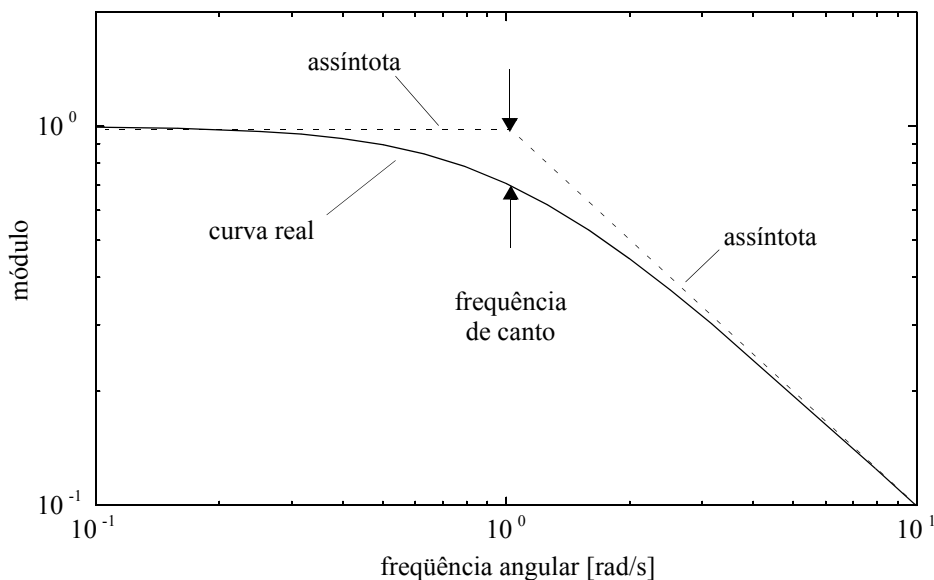


Figura 3.4 Aproximação de uma função de transferência por meio de assíntotas

#### 3.4.2 Ajuste por otimização numérica

Da mesma forma que a função `senofit` foi utilizada para ajustar uma senóide a uma dada curva, uma função semelhante pode ser escrita para ajustar o diagrama de Bode de uma função de transferência. Tal função não está incluída na documentação desta experiência. Parte do procedimento experimental exige que você escreva esta função utilizando o Matlab e a implemente em laboratório (veja o item (b)) das atividades).

## 3.5 Atividades

### 3.5.1 Atividades em sala

O objetivo básico destas atividades é obter um modelo matemático para o servomecanismo do laboratório a partir de sua resposta em frequência. É extremamente importante que você estude o funcionamento das funções `senofit` e `senofit_erquad` antes de realizar a experiência.

Para minimizar o efeito de não-linearidades e maximizar a relação sinal/ruído, é conveniente que você use sinais de amplitude tão grande quanto possível (obviamente sem saturar os amplificadores).

As atividades propostas são as seguintes:

- a) Obtenha a resposta em frequência do servomecanismo utilizando o gerador de funções em sua bancada (não use o gerador de funções do Simulink) para gerar entradas senoidais e considerando a tensão no tacômetro como saída.

OBS. 1: Use o diagrama `exp3_template` como base. Digite `exp3_template` da linha de comando do Matlab para acessar o diagrama.

OBS. 2: Colete os sinais tanto da entrada como da saída utilizando o Simulink e armazene os dados em variáveis do Matlab. Utilize a função `senofit` para obter os pontos de seu gráfico de resposta em frequência e em caráter opcional confirme se os resultados obtidos são coerentes analisando as correspondentes figuras de Lissajous. Não se esqueça de que é necessário traçar tanto a curva de módulo do ganho como também de fase.

OBS. 3: É necessário esperar que o regime permanente senoidal seja atingido antes de se iniciar a coleta de dados.

DICA 1: Trace o diagrama de Bode da função de transferência obtida na experiência anterior para ter uma melhor idéia dos pontos a serem levantados. Use o Matlab para traçar a resposta em frequência à medida que for levantando os pontos. Levantar os pontos e deixar para verificar o resultado em casa é uma forma quase garantida de insucesso nesta experiência.

DICA 2: A obtenção da resposta em frequência é um processo bastante demorado. Procure manter no *workspace* do Matlab todos os dados obtidos e salve-os com bastante frequência. Evite que os dados de uma coleta sobrescrevam os dados obtidos anteriormente (renomear as variáveis ao longo do trabalho é um jeito simples de conseguir isso).

- b) Escreva uma rotina em Matlab análoga à função `senofit` para se obter uma função de transferência a partir da resposta em frequência (o gráfico obtido no item anterior). Por conveniência, o nome `bodefit` é sugerido.

DICA: Uma boa condição inicial é a função de transferência obtida e validada nas experiências anteriores.

SUGESTÃO: A seguinte função pode ser encarada como sendo equivalente à função `senofit_erquad` para o problema de se identificar uma função de transferência a partir da resposta em frequência. Use-a como modelo (ela se encontra disponível no disco rígido de seu computador e você possivelmente já deve tê-la copiado para seu diretório).

```
function f=bodefit_erquad(KKtT,w,G,F)
% BODEFIT_ERQUAD Calcula o erro entre uma curva obtida experimentalmente
% e a resposta em frequência de uma função de transferência da forma
%
%           G(s) = KKt / (Ts+1).
%
%      Uso:   f = bodefit_erquad(KKtT,w,G,F)
%      onde:  x = [KKt; T]
%            w é o vetor de frequências angulares (em rad/s) para as
```

```

%           quais foi obtido o módulo da resposta em frequência
%           G é o vetor de módulos correspondente a w (em ganho linear,
%           isto é, não em dB)
%           F é o vetor de fases (em rad) correspondente a w
%
%           Esta função deve ser utilizada em conjunto com uma função
%           equivalente a senofit.

% RP Marques / 2010a

w=w(:); G=G(:); F=F(:);           % converte em vetores coluna

KKt=KKtT(1); T=KKtT(2);           % desempacota o vetor
sis=tf([KKt],[T 1]);              % cria um objeto sistema

S=G.*cos(F) + j.*G.*sin(F); % converte G e F em fasores

[GG,FF]=bode(sis,w);              % GG,FF são equivalentes a G,F
GG=squeeze(GG); FF=squeeze(FF)*pi/180;

SS=GG.*cos(FF) + j.*GG.*sin(FF); % SS é equivalente a S

f=norm(SS-S);                     % erro quadrático

% gráfico de acompanhamento
subplot(211); semilogx(w,20*log10(GG),'go',w,20*log10(G),'r');
tit = sprintf('KKt = %g ; T = %g',KKtT);
title(tit); ylabel('ganho [dB]');
subplot(212); semilogx(w,180/pi*FF,'go',w,180/pi*F,'r');
ylabel('fase [graus]'); xlabel('w [rad/s]')
drawnow; figure(gcf);             % atualiza a tela imediatamente

```

OBSERVAÇÃO: A função `bode` do Matlab calcula a fase do sistema em graus, e não radianos. Note também que o ganho não é calculado em dB.

- c) Não é possível levantar a resposta em frequência (pelo menos como definida aqui) tomando como saída a tensão no potenciômetro. Explique o porquê.

### 3.5.2 Apresentação de resultados

O item desta seção deve ser apresentado ao professor para a realização da Exp. 4. em uma única folha de papel identificada com os nomes dos integrantes de seu grupo.

- d) Apresente a resposta em frequência (em módulo e fase) levantada e a função de transferência ajustada.

### 3.5.3 Relatório

Esta experiência deve ser incluída no relatório “O Modelo do Servomecanismo”.

## 3.6 Bibliografia sugerida

(Oga93) Oga, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*A Seção 6.2 trata exclusivamente de Diagramas de Bode.*

(Kuo85) Kuo, B. C. **Sistemas de Controle Automático**. Prentice-Hall do Brasil, 4a. ed., 1985.

*Diagramas de Bode, bem como Diagramas de Nyquist são apresentados no Apêndice A.*

(CBS11) Castrucci, P. L.; Bittar, A.; Sales, R. M. **Controle Automático**. LTC, 1a. ed., 2011.

*As Seções 5.2 a 5.6 apresentam um panorama bastante completo de resposta em frequência e Diagramas de Bode e Nyquist.*

# Apêndice A

## Rotinas utilizadas

---

A função `senofit` cumpre basicamente duas tarefas: (i) a obtenção de estimativas iniciais para os parâmetros da curva senoidal, que é realizada pela função `senofit_ic` (não incluída aqui) e; (ii) a execução do procedimento de otimização, que é feito basicamente pela função `fminunc` (disponível no toolbox de otimização do Matlab). Ambas as tarefas são complexas, e em situações desfavoráveis os resultados obtidos podem ser absolutamente inadequados.

```
function [A,w,phi]=senofit(t,y,Ao,wo,phio)
% SENOFIT Ajusta uma funcao senoidal a um sinal dado.
% Os parametros a serem ajustados sao a amplitude, a
% frequencia angular e a fase.
% Caso não sejam fornecidos valores iniciais para es-
% ses parâmetros a rotina tenta obtê-los automaticamente.
%
% Uso:      [A,w,phi] = senofit( t,y );
%           [A,w,phi] = senofit( t,y,Ao,wo,phio );
%
% onde:     t,y   ordenadas e abscissas do sinal coletado.
%
%           Ao,wo,phio   valores aproximados para amplitude, fase
%                       e frequência angular. Esses valores são utilizados
%                       pelo algoritmo de otimização, portanto quanto
%                       melhores essas aproximações, maiores as chances
%                       de sucesso da funcao senofit.
%
%           A,w,phi   valores que minimizam o erro quadratico. Sao
%                       obtidos com o uso de um algoritmo de otimização,
%                       implementado na função senofit.
%
%           Veja também os arquivos senofit.m, senofit_erquad.m e senofit_ic.m.

% RP Marques / 2011a

if ( nargin == 2 )
    % Tenta obter automaticamente Ao, phio, wo
    % Importante: Nao se garante sucesso para todos os casos
    % O sinal tem que ser no minimo parecido com uma senoide,
    % conter pelo menos um periodo (mas não muitos) e
    % apresentar pouca distorsão.
    [Ao,wo,phio] = senofit_ic(t,y);      % função para obter as c.i.
elseif ( nargin == 5 ),
    % Ao, wo, phio fornecidos pelo usuário
else
    error('Número de argumentos incorreto.')
end;

% Exibe o ajuste inicial
plot(t,y,'g',t,Ao*sin(wo*t + phio),'r');
tit = sprintf('Ao = %g ; wo = %g ; phio = %g',Ao,wo,phio);
title(tit);
legend('curva original','aproximação inicial');
xlabel('PRESSIONE QUALQUER TECLA PARA INICIAR O AJUSTE');
figure(gcf);
pause;
```

```

% Otimização
Awphio = [ Ao ; wo ; phio ];           % empacota as c.i. em um vetor

% Opções para o algoritmo (opcional)
options = optimset('Display','off','TolFun',1e-4,'TolX',1e-4);

% Invoca o algoritmo
% Use help fminunc para mais informações
% Este comando minimiza a função senofit_erquad no argumento Awphi, isto é,
% obtém os valores {A,w,phi} que melhor ajustam a curva y(t).
% CAVEAT: Não há garantia de sucesso.
Awphi = fminunc(@(Awphi) senofit_erquad(Awphi,t,y),Awphio,options);

A = Awphi(1); w = Awphi(2); phi = Awphi(3);   % Retorna os valores

```

A função `senofit_erquad` não é utilizada diretamente pelo usuário, mas é executada a cada passo do algoritmo de otimização, e seu entendimento também é essencial.

```

function f=senofit_erquad(Awphi,t,y)
%SENOFIT_ERQUAD   Calcula a norma euclidiana do erro entre o sinal
%                 (possivelmente senoidal)e os valores de uma senóide ajustada.
%
%                 uso: f = senofit_erquad(Awphi,t,y)
%
%                 onde:Awphi = [A; w; phi] é um vetor com parâmetros
%                        para a senóide A*sen(w*t+phi);
%
%                 t,y representam o sinal coletado (t é o tempo
%                 e y(t) o sinal);
%                 f é a norma do erro entre as curvas;
%
%                 A função também traça um gráfico para avaliação do ajuste.

% RP Marques / 2010a
t=t(:); y=y(:);           % convertidose em vetores coluna

yy = Awphi(1)*sin(Awphi(2)*t+Awphi(3)); % yy é uma aproximação para y
f = norm( yy-y );          % erro quadratico entre yy e y

plot(t,y,'g',t,yy,'r');   % traça o gráfico com o ajuste
tit = sprintf('A = %g ; w = %g ; phi = %g',Awphi);
title(tit);
drawnow; figure(gcf);     % atualiza a tela imediatamente

```

# Experiência 4

## Identificação II: Respostas a degrau e modelagem não-linear

---

### 4.1 Introdução

Na experiência anterior tratou-se da obtenção de um modelo linear para o servomecanismo MS15 por meio de sua resposta em frequência. Nesta experiência se fará a obtenção de modelos equivalentes por meio das respostas a degrau do sistema.

A descrição do comportamento do servomecanismo por meio de modelos puramente lineares apresenta deficiências consideráveis. Para se obter modelos mais coerentes com o comportamento realmente obtido se faz necessário considerar não-linearidades, tais como o atrito seco (atrito de Coulomb<sup>1</sup>) e saturações nos amplificadores e outros componentes do sistema eletromecânico. Mais especificamente, apenas o atrito de Coulomb será considerado aqui, e um modelo que inclua a sua influência deverá também ser obtido.

### 4.2 Modelos

#### 4.2.1 Modelagem linear

Seja o modelo linear  $G_{\theta/u}(s)$  do servomecanismo em malha aberta, representado pela função de transferência entre a tensão de entrada e a posição angular do potenciômetro, com

$$G_{\theta/u}(s) = \frac{\Theta_p(s)}{U(s)} = \frac{Kn^2}{s(1+sT)}, \quad (4.1)$$

onde  $U(s) = V_m(s)$ , sendo que  $V_m(s)$  é a tensão aplicada à armadura do motor.

Da mesma forma pode-se definir a função de transferência entre a tensão de entrada e a velocidade angular do potenciômetro,

$$G_{\omega/u}(s) = \frac{\Omega_p(s)}{U(s)} = \frac{Kn^2}{1+sT}. \quad (4.2)$$

O diagrama abaixo (Figura 4.1) mostra uma realização para estas funções de transferência.

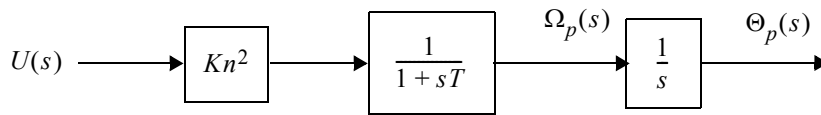
Funções de transferência como as das equações (4.1) e (4.2) podem ser estimadas por meio de sua resposta a degrau. As figuras abaixo apresentam curvas típicas dessas respostas a degrau e como suas características se relacionam aos parâmetros que se deseja estimar.

Para a função de transferência da equação (4.2), os parâmetros  $K$  e  $T$  podem ser obtidos a partir da curva da Figura 4.2 por meio das relações

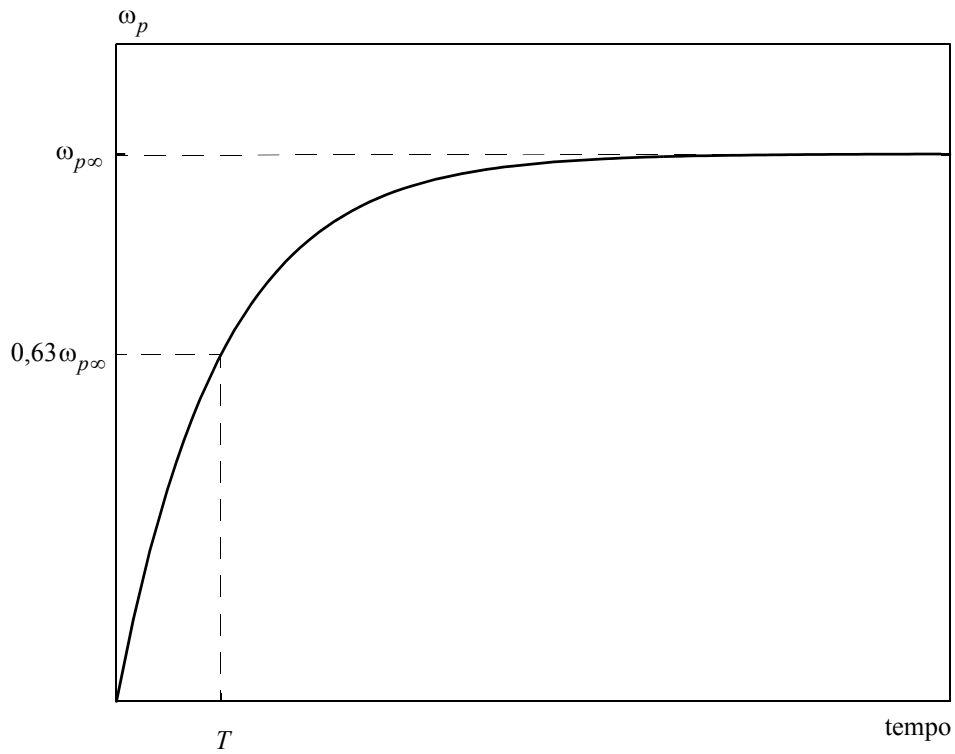
$$K = \frac{\omega_{p\infty}}{Un^2} \quad (4.3)$$

---

1. Em homenagem ao físico francês Charles-Augustin de Coulomb (1736-1806). Coulomb estudou o atrito no maquinário de moinhos de vento e propôs os modelos apresentados aqui.



**Figura 4.1** Diagrama de blocos do servomecanismo



**Figura 4.2** Resposta a degrau (primeira ordem)

e

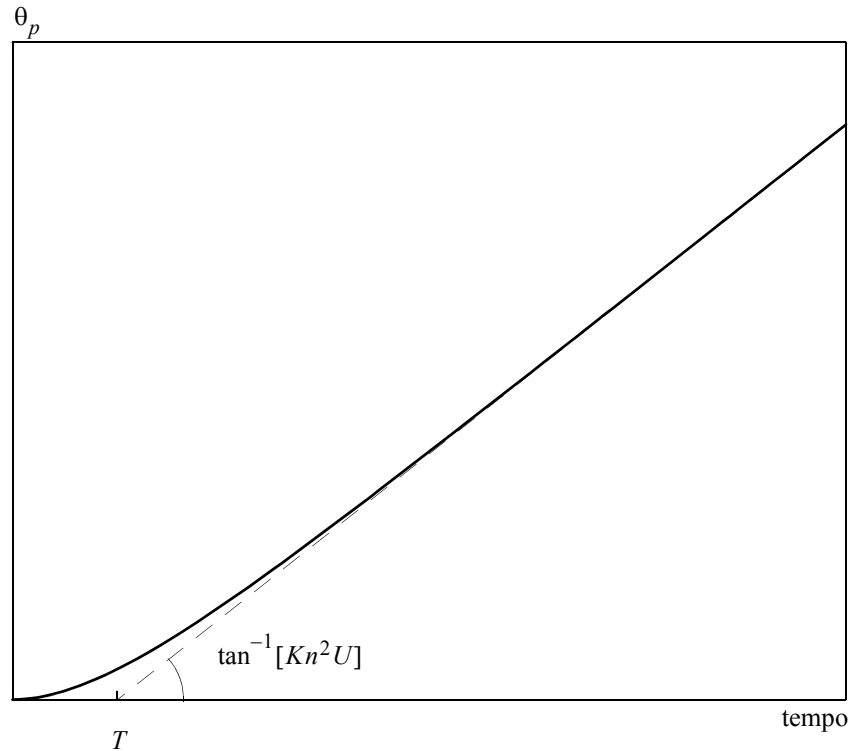
$$\omega_p(T) \cong 0,63 \omega_{p\infty}, \quad (4.4)$$

onde  $U$  é a amplitude do degrau

A resposta a degrau do sistema da equação (4.1) pode ser vista na Figura 4.3. Note que esta resposta é a integral da resposta apresentada na Figura 4.2. Quando  $t \rightarrow \infty$ , a resposta tende para uma reta dada pela equação

$$\theta_{p\infty}(t) = (Kn^2 U)t - Kn^2 UT, \quad (4.5)$$

( $U$  é a amplitude do degrau).



**Figura 4.3** Resposta a degrau (segunda ordem com um pólo na origem)

#### 4.2.2 Inclusão do atrito de Coulomb no modelo

Sempre que superfícies mecânicas escorregam umas sobre as outras há o aparecimento de forças de atrito, podendo ser classificadas dentre os três tipos abaixo

- i) atrito viscoso linear: proporcional à velocidade relativa entre as superfícies;
- ii) atrito de Coulomb: composto por uma força constante, sempre se opondo ao movimento relativo entre as superfícies e;
- iii) atrito de destaque (ou atrito de Coulomb de destaque), que é uma força de oposição que só aparece ao se iniciar o movimento.

A modelagem do servomecanismo segundo (4.1) parte do pressuposto de que no motor de corrente contínua a relação entre torque e velocidade angular no regime estacionário é linear, isto é, considera-se apenas a influência do atrito viscoso. Nesse caso o torque total do sistema, em regime estacionário, que se resume ao torque de atrito (medido no eixo do potenciômetro) é dado por:

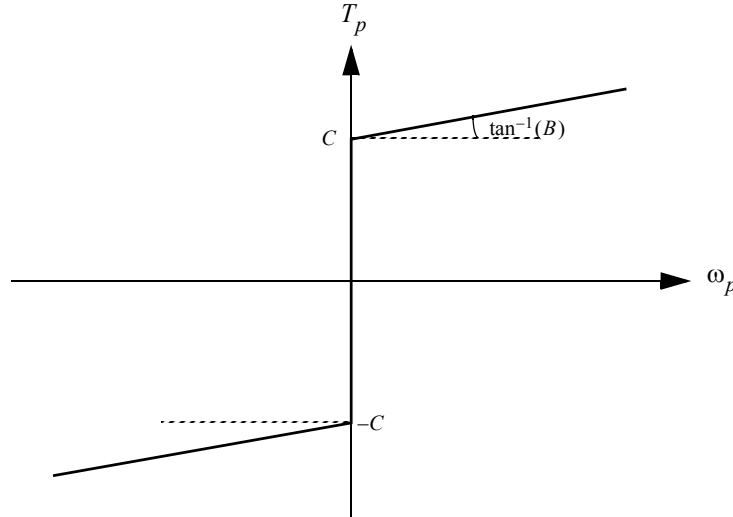
$$T_{at_p} = B\omega_p, \quad (4.6)$$

onde  $B$  é o coeficiente de atrito viscoso

Assumindo-se que tanto a força de atrito viscoso como a força de atrito de Coulomb sejam significativas, obtém-se a curva característica de torque contra velocidade de regime da Figura 4.4 (referidos, tanto o torque como a velocidade de regime ao eixo do potenciômetro). Neste caso, o torque total do sistema, que se resume ao torque de atrito seria dado por:

$$T_{at_p} = B\omega_p + C \operatorname{sgn}(\omega_p), \quad (4.7)$$





**Figura 4.4** Característica Torque X Velocidade

onde  $B$  e  $C$  (torque de atrito de Coulomb) estão indicados na figura e  $\text{sgn}(x)$  é a função “sinal algébrico” de  $x$ .

Sem considerar o atrito de Coulomb, o torque no eixo do potenciômetro seria dado por

$$T_p(t) = J\ddot{\theta}_p(t) + B\dot{\theta}_p(t). \quad (4.8)$$

A equação diferencial associada à função de transferência  $G_{\theta/u}(s)$  de (4.1) é

$$\ddot{\theta}_p(t) + \frac{\dot{\theta}_p(t)}{T} = \frac{Kn^2}{T}V_m(t), \quad (4.9)$$

que pode ser comparada diretamente com (4.8).

Ao se considerar o atrito de Coulomb, tem-se o seguinte:

$$T_p(t) = J\ddot{\theta}_p(t) + B\dot{\theta}_p(t) + C\text{sgn}(\dot{\theta}_p(t)). \quad (4.10)$$

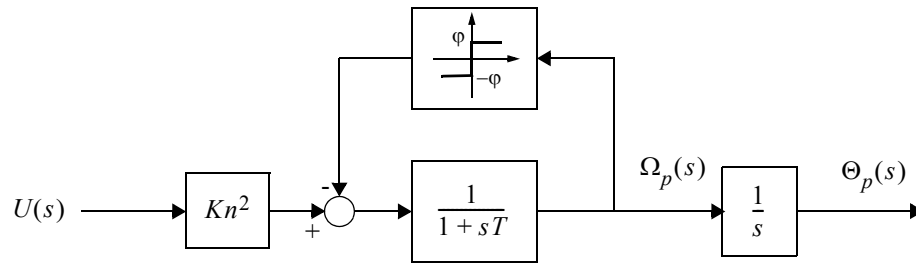
Note que  $J$  representa uma associação de momentos de inércia referidos ao eixo do potenciômetro. Tem-se também que

$$\ddot{\theta}_p(t) + \frac{\dot{\theta}_p(t)}{T} + \frac{\varphi}{T}\text{sgn}(\dot{\theta}_p(t)) = \frac{Kn^2}{T}V_m(t), \quad (4.11)$$

onde  $\varphi$  é uma constante proporcional a  $C$  de (4.7). Esse termo adicional leva ao diagrama de blocos da Figura 4.5 (compare com o seu equivalente linear na Figura 4.1).

O modelo acima é capaz de descrever o comportamento do sistema com bastante acurácia dentro de limites de validade bem determinados. Note que o termo que expressa o torque resistente do atrito de Coulomb, sob certas situações, pode ser maior em módulo e de sinal algébrico oposto ao termo que representa o torque motor aplicado ao sistema, o que levaria à criação de movimento pelo torque de atrito, o que fisicamente não faz sentido. Assim, este modelo deve ser utilizado apenas para descrever a velocidade do sistema, desde que esta não mude de sentido ou seja nula ou muito pequena.

É desejável utilizar este modelo também para descrever a posição do sistema, o que pode ser feito de maneira limitada, mas com razoável fidelidade ao comportamento real do sistema. Para tanto é necessário eliminar as inconsistências físicas do modelo expresso no diagrama da Figura 4.5. Note no diagrama que caso



**Figura 4.5** Diagrama de blocos do servomecanismo com atrito de Coulomb (a variável de Laplace<sup>a</sup> “s” aparece por abuso de notação, já que o sistema resultante é não-linear).

a. Pierre-Simon, Marquês de Laplace (1749-1827). Matemático, astrônomo e físico francês.

$Kn^2 u(t)$  seja menor em módulo que  $\phi$ , a velocidade angular  $\omega_p(t)$  pode mudar de sentido. Para eliminar essa situação, basta impor que o termo de atrito não seja maior que o outro termo nas situações em que tal ocasionaria reversão do movimento. Fisicamente isso pode ser associado ao fato de que o movimento do servomecanismo é interrompido sempre que a velocidade angular se aproxima de zero e o torque aplicado não é capaz de vencer o atrito. Nessa situação o movimento somente será retomado quando o torque aplicado for maior que o atrito de destaque.

O arquivo `exp4_template.mdl` (digite `>> exp4_template` no prompt do Matlab) contém uma versão adaptada do diagrama da Figura 4.5 sem as inconsistências físicas do diagrama da figura. Note que não há diferenças entre os comportamentos produzidos pelos modelos nas condições de validade de (4.11).

#### 4.2.3 Identificando modelos com atrito de Coulomb

Considere um sistema descrito pela equação (4.11). Suponha que para uma entrada  $u_1(t)$  sua resposta seja  $\dot{\theta}_{p1}(t) = \omega_{p1}(t)$  e que para uma entrada  $u_2(t)$  a resposta seja  $\dot{\theta}_{p2}(t) = \omega_{p2}(t)$ . Admita também que para  $t > t_0$ ,  $\text{sgn}(\omega_{p1}(t)) = \text{sgn}(\omega_{p2}(t))$ . Assim, para  $t > t_0$ , a diferença

$$z(t) = \omega_{p1}(t) - \omega_{p2}(t) \quad (4.12)$$

entre as duas respostas é regida pelo modelo

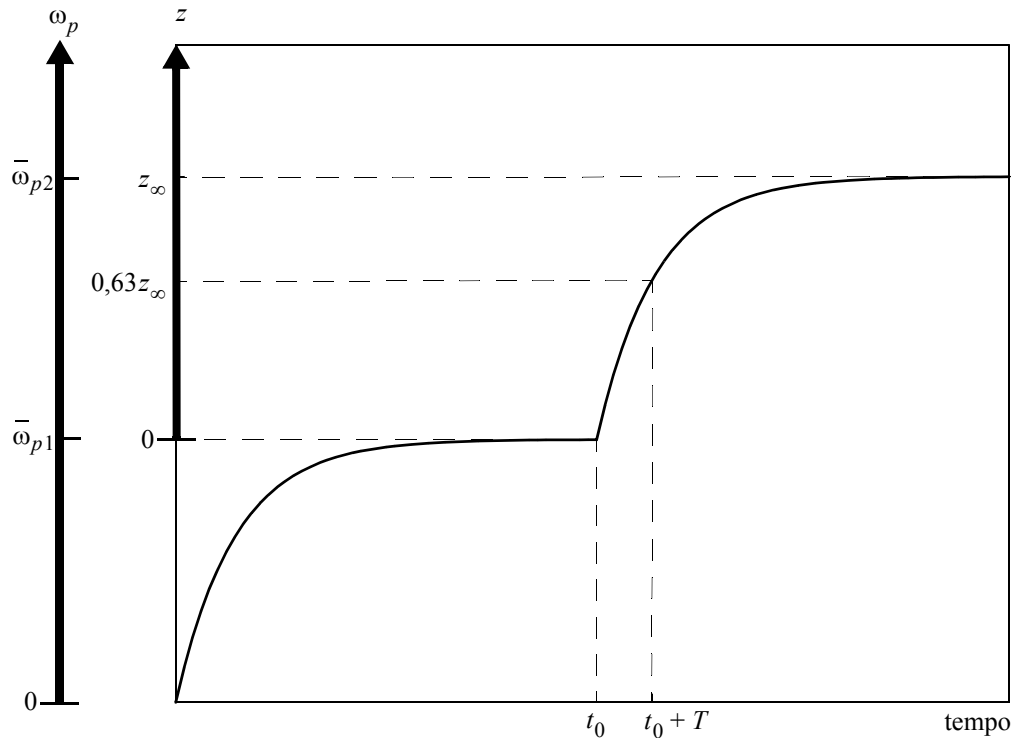
$$\dot{z}(t) + \frac{z(t)}{T} = \frac{Kn^2}{T} (u_1(t) - u_2(t)). \quad (4.13)$$

Note que o modelo (4.13) é linear, apesar do atrito de Coulomb. Uma situação em que esse modelo é aplicável pode ser obtida em laboratório. Basta excitar o servomotor inicialmente com um degrau de amplitude  $U_1$ , e quando atingido o regime permanente (i.e. quando a velocidade angular no eixo do potenciômetro for  $\bar{\omega}_{p1}$  constante, ou seja quando  $t = t_0$ ), altera-se a entrada adicionando mais um degrau  $\Delta U$  de modo que a entrada adquira o valor  $U_2 = U_1 + \Delta U$ , obtendo-se uma curva como a da Figura 4.6, onde  $\bar{\omega}_{p2}$  é a velocidade angular resultante em regime estacionário após a aplicação do segundo degrau. Note que  $z = \omega_p - \bar{\omega}_{p1}$ , e que tanto  $z$  como  $\omega_p$  representam velocidades angulares no eixo do potenciômetro. Apenas as referências dos eixos (veja a figura) são diferentes.

A partir dessa curva obtém-se diretamente  $T$  e calcula-se

$$K = \frac{z_\infty}{n^2 \Delta U} \quad (4.14)$$

e



**Figura 4.6** Curva para identificação do atrito seco

$$\varphi = Kn^2 U_1 - \bar{\omega}_{p1}. \quad (4.15)$$

Note que o valor de  $K$  obtido para o modelo não-linear pode diferir daquele obtido anteriormente para o modelo linear.

## 4.3 Atividades

### 4.3.1 Atividades em sala

- Obtenha modelos lineares para o sistema (sem considerar o atrito de Coulomb) a partir de suas respostas a degrau, tomando como saídas tanto a velocidade como a posição angulares no eixo do potenciômetro. Obtenha esses modelos de duas maneiras: i) a partir de inspeção visual dos gráficos, como sugerido pelas figuras 4.2 e 4.3 e; ii) usando um algoritmo de otimização (escreva funções no Matlab análogas à função `senofit` e `senofit_erquad` da Experiência 3 e use as respostas a degrau no lugar das senóides). Avalie a qualidade dos resultados obtidos quando se considera velocidade ou posição como saída e discuta eventuais discrepâncias.

**OBSERVAÇÃO:** Não se esqueça de que as funções de transferência descritas nesta experiência consideram como saída posição e velocidade angulares, enquanto que as curvas são coletadas em termos de tensões no tacômetro e no potenciômetro.

- Obtenha um modelo não-linear para o sistema, como sugerido pela Figura 4.6, utilizando o modelo do diagrama `exp4_template`. Obtenha esse modelo por inspeção visual do gráfico.
- Compare os modelos obtidos (nesta experiência e na Experiência 3) entre si e comente as diferenças.

**IMPORTANTE:** Esses modelos serão usados futuramente, tanto para projeto de controladores como em simulações para verificação de desempenho.

### 4.3.2 Relatório

Esta experiência conclui o relatório “O Modelo do Servomecanismo”.

## 4.4 Bibliografia sugerida

(Kuo85) Kuo, B. C. **Sistemas de Controle Automático**. Prentice-Hall do Brasil, 4a. ed., 1985.

*A Seção 4.3 faz uma breve apresentação de diversos tipos de atrito.*

(Oga92) Ogata, K. **System Dynamics**. Prentice-Hall. 2nd. ed., 1992.

*A Seção 3-4 trata exclusivamente de modelos para atrito.*

(Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*O Capítulo 4 trata de respostas transitórias de sistemas lineares e é uma boa fonte de consulta para a parte linear desta experiência.*

JJC,RPM/96 rev. RPM/2014a



# Experiência 5

## Controle proporcional

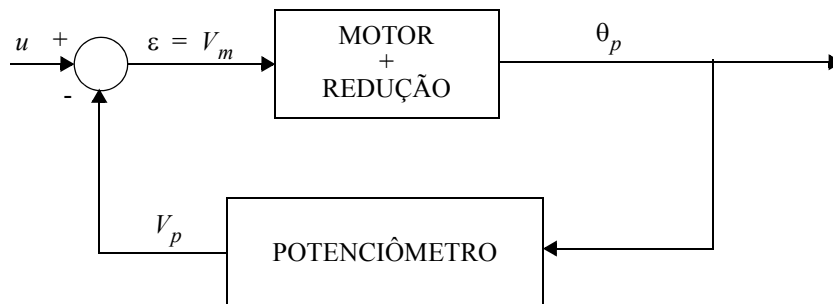
### 5.1 Introdução

A experiência visa a montagem de controladores de posição e velocidade simples do tipo proporcional para o servomecanismo do laboratório, e posteriormente a comparação dos desempenhos obtidos com os previstos por um modelo matemático.

### 5.2 Controle de posição

#### 5.2.1 Realimentação unitária

Para se controlar a posição angular do servomecanismo, é necessário fechar a malha através do potenciômetro (o sensor de posição), comparando essa medida com um sinal de referência. O erro obtido é então aplicado à entrada do sistema, resultando na malha de realimentação esquematizada na figura 5.1.



**Figura 5.1** Esquema do controlador de posição.

A função de transferência do servomecanismo em malha aberta (correspondente ao bloco MOTOR/REDUÇÃO da Figura 5.1) possui a seguinte forma:

$$G_{\theta/u}(s) = \frac{\Theta_p(s)}{U_s} = \frac{Kn^2}{s(1 + sT)}, \quad (5.1)$$

ao passo que o bloco POTENCIÔMETRO obedece à seguinte relação:

$$V_p(s) = K_p \Theta_p(s). \quad (5.2)$$

Pode-se supor que a tensão de referência  $u(t)$  seja proporcional a uma posição angular de referência  $\theta_r(t)$ , e que guardam entre si a mesma relação que entre  $V_p$  e  $\theta_p$ . Daí

$$U(s) = K_p \Theta_r(s), \quad (5.3)$$

de onde se obtém a seguinte função de transferência em malha aberta:

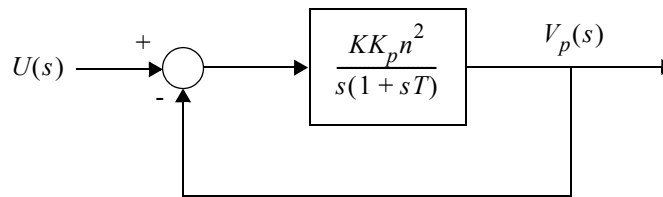
$$G_{\theta/\theta}(s) = \frac{\Theta_p(s)}{\Theta_r(s)} = \frac{KK_p n^2}{s(1+sT)}. \quad (5.4)$$

Como fisicamente é muito mais simples tratar com tensões do que com posições angulares, a função de transferência que será utilizada é a seguinte:

$$G_{p/u}(s) = \frac{V_p(s)}{U(s)} = \frac{KK_p n^2}{s(1+sT)}. \quad (5.5)$$

Note que esta função de transferência é rigorosamente igual a (5.4).

Finalmente o diagrama de blocos, em termos das funções de transferência, é o seguinte:

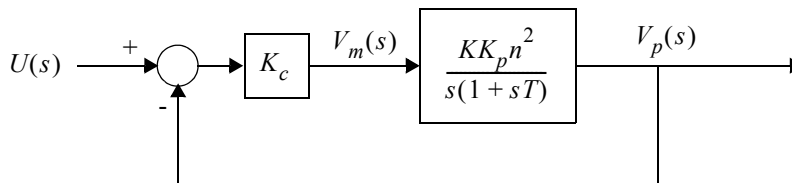


**Figura 5.2** O controlador de posição.

### 5.2.2 O controlador proporcional

No item anterior discutiu-se apenas um sistema de controle por realimentação unitária. Em um sistema de controle deste tipo, a entrada da planta é o sinal de erro, formado pela diferença entre a referência e a saída da planta. Nem sempre esse sinal de erro é por si só capaz de satisfazer aos requisitos exigidos, sendo necessário o uso de compensadores.

A forma mais simples de se alterar o sinal de erro que alimenta a planta é a utilização de um ganho proporcional, que dá ao sistema de controle a estrutura abaixo:

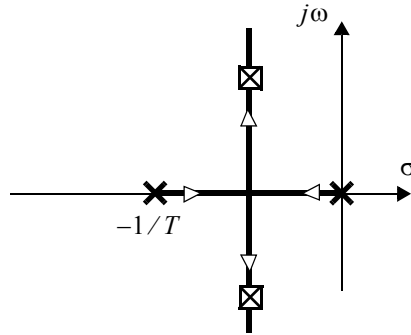


**Figura 5.3** O controlador proporcional.

A função de transferência em malha fechada  $G_{v/u}(s)$  é dada por:

$$G_{v/u}(s) = \frac{KK_c K_p n^2}{s(1+sT) + KK_c K_p n^2}, \quad (5.6)$$

ao passo que o lugar das raízes do sistema é o apresentado na Figura 5.4.

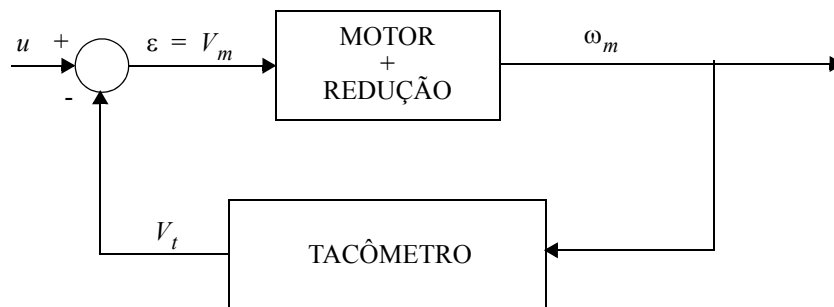


**Figura 5.4** Lugar geométrico das raízes para o sistema da equação (5.6).

## 5.3 Controle de velocidade

### 5.3.1 Realimentação unitária

A construção de um regulador de velocidade requer que se realimente o sinal de velocidade. Para tanto utiliza-se o tacômetro como sensor de velocidade e compara-se a tensão de saída deste com o sinal de referência que se deseja acompanhar. A diferença entre esses sinais é o erro de acompanhamento (Figura 5.5)



**Figura 5.5** Esquema do controlador de velocidade.

A função de transferência entre a tensão de entrada  $V_m(s)$  e a velocidade angular  $\Omega_m(s)$  do eixo do motor é:

$$G_{\Omega/V}(s) = \frac{K}{1 + sT}. \quad (5.7)$$

O tacômetro fornece uma tensão  $V_t(s)$  proporcional a  $\Omega_m(s)$ , constante de proporcionalidade  $K_t$ :

$$V_t(s) = K_t \Omega_m(s). \quad (5.8)$$

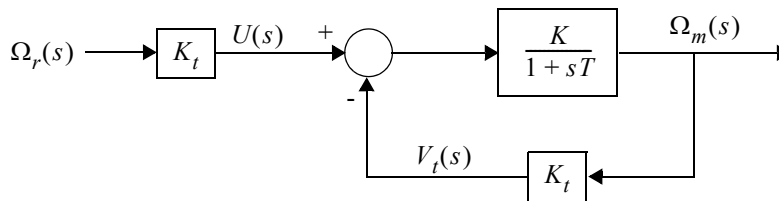
Imaginando que a tensão de referência represente uma velocidade angular tal que:

$$U(s) = K_t \Omega_r(s), \quad (5.9)$$

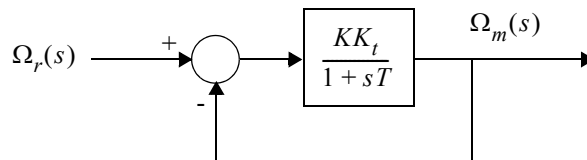
então pode-se desenhar o diagrama de blocos que representa o controlador de velocidade em malha fechada da



Figura 5.6.

**Figura 5.6** Diagrama de blocos do regulador de velocidade.

Redesenhando o diagrama acima obtém-se a forma de realimentação unitária representada na Figura 5.7.

**Figura 5.7** Diagrama de blocos do regulador de velocidade com realimentação unitária.

A função de transferência de malha fechada do sistema acima é dada por:

$$G_{\Omega/\Omega}(s) = \frac{KK_t}{1 + KK_t + sT}. \quad (5.10)$$

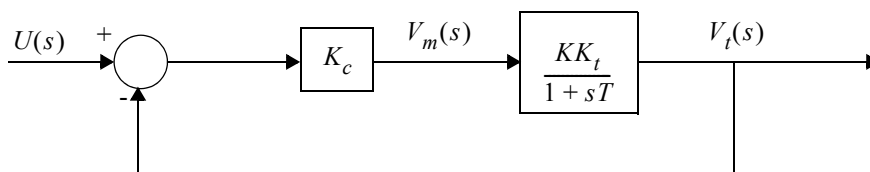
Ou, considerando as proporcionalidades expressas pelas equações (5.8) e (5.10),

$$G_{V/u}(s) = \frac{KK_t}{1 + KK_t + sT}. \quad (5.11)$$

Da mesma forma que no caso do controlador de posição, esta forma será utilizada aqui pela maior simplicidade na manipulação das tensões em vez das velocidades angulares.

### 5.3.2 Controlador proporcional

De modo similar ao início da seção 5.2.2, considera-se aqui um controlador proporcional conforme a Figura 5.8.

**Figura 5.8** Diagrama de blocos do controlador proporcional de velocidade.

Neste controlador, a tensão  $V_m$  aplicada na entrada do motor é proporcional ao erro  $\varepsilon$  entre a tensão de referência  $u$  e a tensão do tacômetro  $V_t$ . A função de transferência em malha fechada  $G_{V/u}(s)$  resulta então:

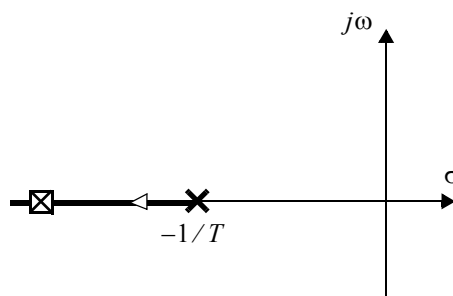
$$G_{V/u}(s) = \frac{KK_tK_c}{1 + KK_tK_c + sT}. \quad (5.12)$$

Observa-se que a constante de tempo do sistema em malha fechada é:

$$T_{mf} = \frac{T}{1 + KK_tK_c} \quad (5.13)$$

e que, aumentando-se o ganho do controlador proporcional  $K_c$ , obtém-se maior rapidez de resposta do sistema (isto é, um  $T_{mf}$  menor).

O lugar das raízes do sistema da Figura 5.8 é o seguinte:



**Figura 5.9** Lugar das raízes do sistema de controle de velocidade.

## 5.4 Atividades

### 5.4.1 Atividades em sala

- Utilizando o Simulink para realizar o controlador proporcional, monte o sistema de controle de posição representado na Figura 5.3. Note que para  $K_c = 1$  tem-se o caso de realimentação unitária..
- Registre as respostas do sistema a um degrau de referência com tensão correspondente a uma rotação de  $90^\circ$  no eixo da escala graduada do servo (calcule esse valor a partir do ganho do potenciômetro  $K_p$ ). Utilize para isso três valores de ganho do controlador proporcional:

b.1)  $K_c = 1$  ;

b.2) escolha  $K_c$  tal que a resposta do sistema resulte superamortecida (anote o valor de  $K_c$ );

b.3) escolha  $K_c$  tal que a resposta do sistema resulte sub-amortecida (anote o valor de  $K_c$ );

OBSERVAÇÃO: Utilize a mesma tensão de referência para os três ensaios.

- Compare as respostas obtidas no item (b) com as esperadas teoricamente com base no modelo obtido em experiências anteriores. Comente os resultados obtidos. Considere o modelo completo, com atrito de Coulomb incluído.
- Monte o sistema de controle de velocidade representado na Figura 5.8.

- e) Registre as respostas do sistema a um degrau de referência correspondente a uma rotação de 100 rpm no eixo do potenciômetro (para o caso em que  $K_c \rightarrow \infty$ . Note que para ganhos finitos o valor de regime do sistema não será igual ao do degrau aplicado. Utilize o ganho do tacômetro  $K_t$  para calcular esse valor). Utilize para isso três valores de ganho:

e.1)  $K_c = 1$  ;

e.2)  $K_c = 1,5$  ;

e.3)  $K_c = 2,5$  .

OBSERVAÇÃO: Utilize a mesma tensão de referência para os três ensaios.

- f) Compare as respostas obtidas no item (e) com aquelas obtidas por simulação dos modelos linear e não-linear. Comente os resultados obtidos.
- g) Registre a resposta a degrau para um dos valores de  $K_c$  que você utilizou no item (e), mas abaixe a alavanca do freio eletromagnético após o sistema ter atingido o regime permanente. Comente o resultado obtido, comparando-o com o correspondente do item (e) e procurando justificá-lo.

OBSERVAÇÃO: A resposta do sistema controlado a perturbações como esta será um dos índices de qualidade utilizado na comparação entre este e futuros controladores.

#### 5.4.2 Apresentação de resultados

O item desta seção deve ser apresentado ao professor para a realização da Exp. 6 em uma única folha de papel identificada com os nomes dos integrantes de seu grupo.

- h) Apresente os gráficos com as respostas ao degrau dos itens (b) e (e) - dois gráficos com três curvas cada um.

#### 5.4.3 Relatório

Esta experiência deve ser incluída nos relatórios “Controle de Velocidade” e “Controle de Posição”.

### 5.5 Bibliografia sugerida

Todos os livros citados abaixo apresentam os fundamentos de Lugar Geométrico das Raízes e os princípios básicos de Controle por Realimentação.

- (DB98) Dorf, R. C.; Bishop, R. H. **Modern Control Systems**. Addison-Wesley, 8th. ed., 1998.
- (FPE86) Franklin, G. F.; Powell, J. D.; Emami-Naeini, A. **Feedback Control of Dynamic Systems**. Addison-Wesley, 1st. ed., 1986.
- (Kuo85) Kuo, B. C. **Sistemas de Controle Automático**. Prentice-Hall do Brasil, 4a. ed., 1985.
- (Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.
- (CBS11) Castrucci, P. L.; Bittar, A.; Sales, R. M. **Controle Automático**. LTC, 1a. ed., 2011.

# Experiência 6

## Controle de posição com realimentação auxiliar de velocidade

### 6.1 Introdução

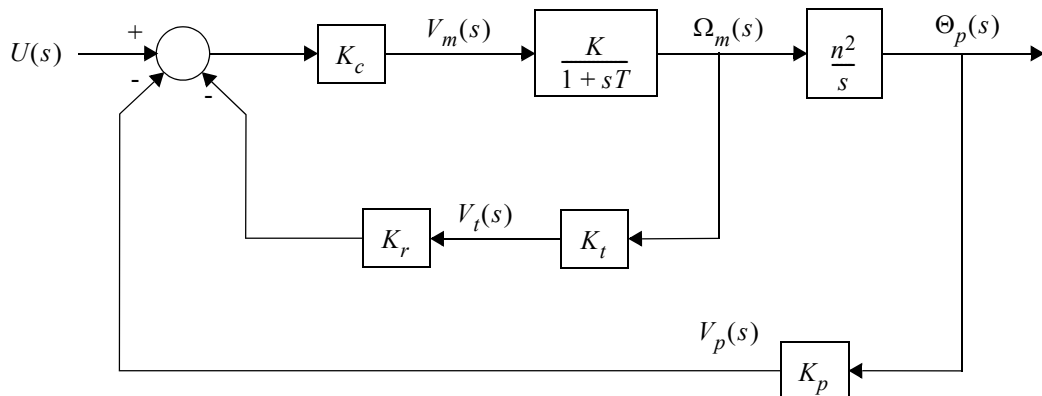
Nesta experiência far-se-á a montagem de um controlador de posição para o servomecanismo utilizando realimentação auxiliar de velocidade. O método do Lugar Geométrico das Raízes será utilizado para se analisar os efeitos desse tipo de realimentação e para se comparar com o controlador proporcional estudado anteriormente.

### 6.2 Realimentação auxiliar de velocidade

#### 6.2.1 Estrutura do controlador

O controlador proporcional apresenta o seguinte conflito: é desejável fazer o ganho do controlador  $K_c$  tão grande quanto possível a fim de se reduzir o tempo de subida para a entrada degrau e/ou o erro de regime para entradas rampa, e adicionalmente minimizar os efeitos de não-linearidades como o atrito seco. Porém valores grandes de  $K_c$  tendem a aumentar o sobressinal da resposta a degrau do sistema em malha fechada, o que é indesejável.

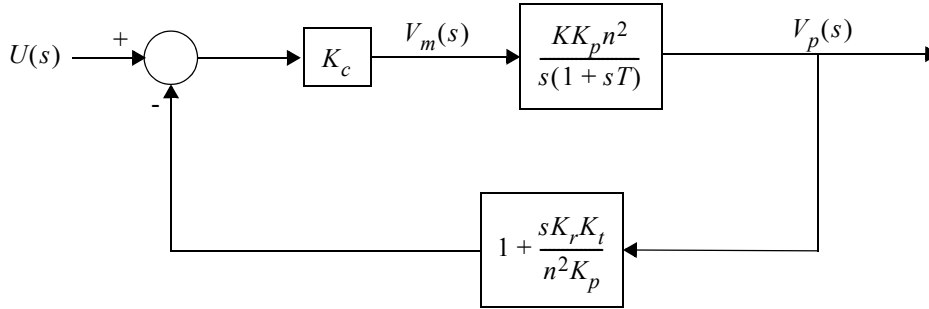
Nem sempre é possível atingir um compromisso razoável apenas com uma realimentação de posição, tornando necessário o uso de outros tipos de ação de controle. Um tipo específico de ação de controle que atua nesse sentido é a *realimentação auxiliar de velocidade*, que consiste na realimentação de um sinal proporcional à velocidade angular do servomecanismo somado a um sinal proporcional à posição, como no diagrama da Figura 6.1.



**Figura 6.1** O controlador de posição com realimentação auxiliar de velocidade.

Note que foi introduzido um ganho  $K_r$  multiplicando o sinal  $V_t$  fornecido pelo tacogerador. Esse ganho determina o efeito da malha auxiliar de velocidade sobre o controlador. Em particular, observe que se  $K_r = 0$ , então a malha de controle se resume a um controlador proporcional.

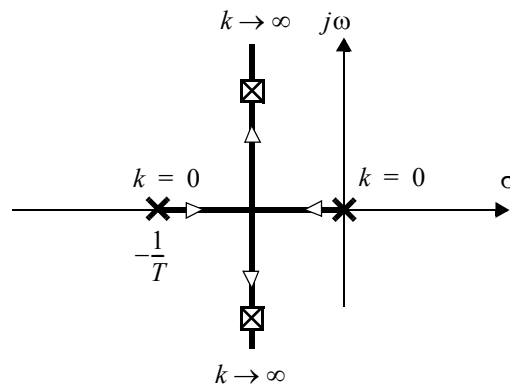
O diagrama da Figura 6.1 pode ser simplificado, resultando no diagrama da Figura 6.2.



**Figura 6.2** Simplificação do diagrama da Figura 6.1.

### 6.2.2 Análise do sistema em malha fechada

Conforme mencionado, quando se considera  $K_r = 0$ , o sistema de controle se reduz a um controlador proporcional. O lugar das raízes para um sistema como esse é o seguinte (Figura 6.3):

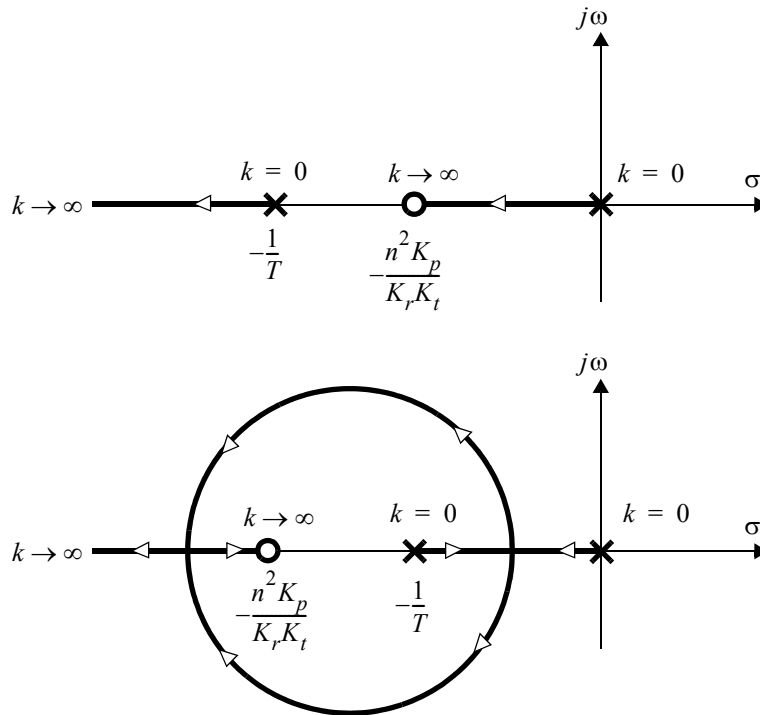


**Figura 6.3** Lugar das raízes para  $K_r = 0$ .

Se entretanto  $K_r \neq 0$ , pode-se mostrar que há a introdução de um zero real em  $s = -\frac{n^2 K_p}{K_r K_t}$  na função

de transferência de malha aberta (lembrando que a função de transferência de malha aberta para um sistema como o da Figura 6.2 é dada pelo produto da função de transferência direta pela função de transferência da malha de realimentação). Deste modo, dependendo de onde o zero se situar no eixo real, se entre os pólos ou à esquerda deles, resultam os lugares de raízes da Figura 6.4.

Ao contrário do sistema com realimentação de posição (veja a Figura 6.3), onde, aumentando-se o ganho de malha, obtém-se pólos de malha fechada cada vez mais afastados do eixo real, o que implica em um comportamento oscilatório cada vez mais pronunciado para o sistema, o sistema com realimentação auxiliar de velocidade apresenta em qualquer um dos casos (Figura 6.4) um valor de ganho acima do qual ambos os pólos de malha fechada serão reais e negativos. E mesmo no segundo caso (na parte inferior da Figura 6.4), onde há uma faixa de valores de ganho para a qual o sistema apresenta comportamento oscilatório, o coeficiente de amortecimento do sistema se apresenta limitado inferiormente, não havendo o risco, como no caso em que  $K_r = 0$ , de se produzir um coeficiente de amortecimento cada vez menor à medida que se aumenta o ganho de malha, o que implica em um sobressinal cada vez maior.



**Figura 6.4** Lugares das raízes para  $K_r \neq 0$ .

## 6.3 Atividades

### 6.3.1 Atividades prévias

Atividades a serem realizadas antes da realização da experiência. Os resultados obtidos nestes itens deverão estar disponíveis para uso no laboratório.

- Deduz a fórmula da função de transferência  $T_{aux_{p/u}}(s) = \frac{V_p(s)}{U(s)}$  para o sistema em malha fechada conforme a Figura 6.2.
- Calcule, a partir dos valores conhecidos de  $K$ ,  $T$ ,  $K_t$  e  $K_p$ , os intervalos de valores de  $K_r$  em que ocorrem as situações descritas na Figura 6.4. Considere que o ganho  $K_r$  possa variar entre 0 e 10.
- Descreva qualitativamente, no que se refere a oscilações, como evolui a forma de resposta a degrau do sistema em malha fechada à medida que o ganho  $K_c$  varia de 0 a  $\infty$ . Considere as duas situações da Figura 6.4.

**IMPORTANTE:** A realização da experiência em tempo hábil requer a realização prévia dos itens acima.

### 6.3.2 Atividades em sala

Este item trata basicamente dos experimentos e das medidas a serem efetuadas. Tenha em mente que esses dados serão utilizados posteriormente nas atividades do item 6.3.4, e que, portanto, os dados devem ser tomados de forma adequada e com o rigor necessário para que sejam analisados corretamente (por exemplo: um esboço muito simplificado de resposta a degrau pode não ser adequado para se determinar posteriormente o coeficiente de amortecimento e a frequência natural do sistema).

- Monte o controlador de posição com realimentação auxiliar de velocidade (Figura 6.1) para o servomecanismo, utilizando-se do computador para fechar a malha.

- e) Obtenha as respostas a degrau do sistema em malha fechada para diversas combinações de valores de  $K_r$  e  $K_c$  variando entre 0 e 10. Utilize pelo menos 3 valores de  $K_r$  ( $K_r = 0$  e outros dois valores, um para cada situação da Figura 6.4) e, para cada um desses valores, varie o ganho  $K_c$  de modo a percorrer o lugar das raízes. Use preferencialmente os mesmos valores de  $K_c$  da experiência anterior.

IMPORTANTE: Certifique-se de que não ocorra saturação em nenhuma das situações, especialmente para valores altos de  $K_c$ .

### 6.3.3 Apresentação de resultados

O item desta seção deve ser apresentado ao professor para a realização da Exp. 7 em uma única folha de papel identificada com os nomes dos integrantes de seu grupo.

- f) Apresente os gráficos com as respostas ao degrau do item (e) - um gráfico para cada valor de  $K_r$  ( $K_r = 0$  não é necessário) contendo cada um deles uma curva para cada valor utilizado de  $K_c$ .

### 6.3.4 Relatório

Esta experiência faz parte do relatório "Controle de Posição".

- g) Compare as respostas a degrau obtidas com as simulações dos modelos linear e não-linear do sistema. Comente os resultados e explique eventuais discrepâncias.

## 6.4 Bibliografia sugerida

(Kuo85) Kuo, B. C. **Sistemas de Controle Automático**. Prentice-Hall do Brasil, 4a. ed., 1985.

*Realimentação de velocidade, referida como realimentação tacométrica ou de relação, é discutida na Seção 8.5, e também na Seção 4.5.*

(Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*A Seção 4.4, que discute um servossistema similar ao do laboratório, apresenta um esquema de realimentação de velocidade. Também na Seção 7.7, a realimentação de velocidade é vista num contexto mais amplo.*

(CBS11) Castrucci, P. L.; Bittar, A.; Sales, R. M. **Controle Automático**. LTC, 1a. ed., 2011.

*Realimentação auxiliar pode ser vista no item 4.6.4, num contexto bastante similar ao presente.*

atual. JJC,JCA/97 rev. RPM/2011a

# Experiência 7

## Controle de posição com compensador por avanço de fase

---

### 7.1 Introdução

Esta experiência tem por objetivos o projeto de um compensador por avanço de fase para controlar a posição angular do servomecanismo do laboratório e posterior análise do desempenho do sistema em malha fechada.

### 7.2 Formulação básica

Via de regra, a resposta transitória pode ser associada à resposta do sistema a altas frequências, ao passo que o regime permanente pode ser relacionado às baixas frequências.

Um compensador por avanço de fase serve basicamente para aumentar o ganho em altas frequências da planta em malha aberta, podendo ao mesmo tempo melhorar a sua margem de fase. Uma vez que se feche a malha, isso tem por efeito uma melhor resposta transitória do sistema, com pouca influência na situação de regime permanente.

A estrutura de um compensador por avanço de fase é a seguinte:

$$G_{av}(s) = \frac{K_{oc}(1 + s\alpha T_c)}{1 + sT_c}, \quad (7.1)$$

com  $\alpha > 1$ . Caso se tivesse  $\alpha < 1$  o compensador acima seria denominado compensador por atraso de fase. Esse segundo tipo de compensação não será discutido aqui.

O projeto de um compensador por avanço de fase consiste na escolha adequada de valores de  $K_{oc}$ ,  $\alpha$  e  $T_c$  de forma que o sistema em malha fechada atenda a certas especificações de desempenho. A Figura 7.1 apresenta a resposta em frequência do compensador.

### 7.3 Projeto do controlador para o servomecanismo

#### 7.3.1 O compensador e as especificações de projeto

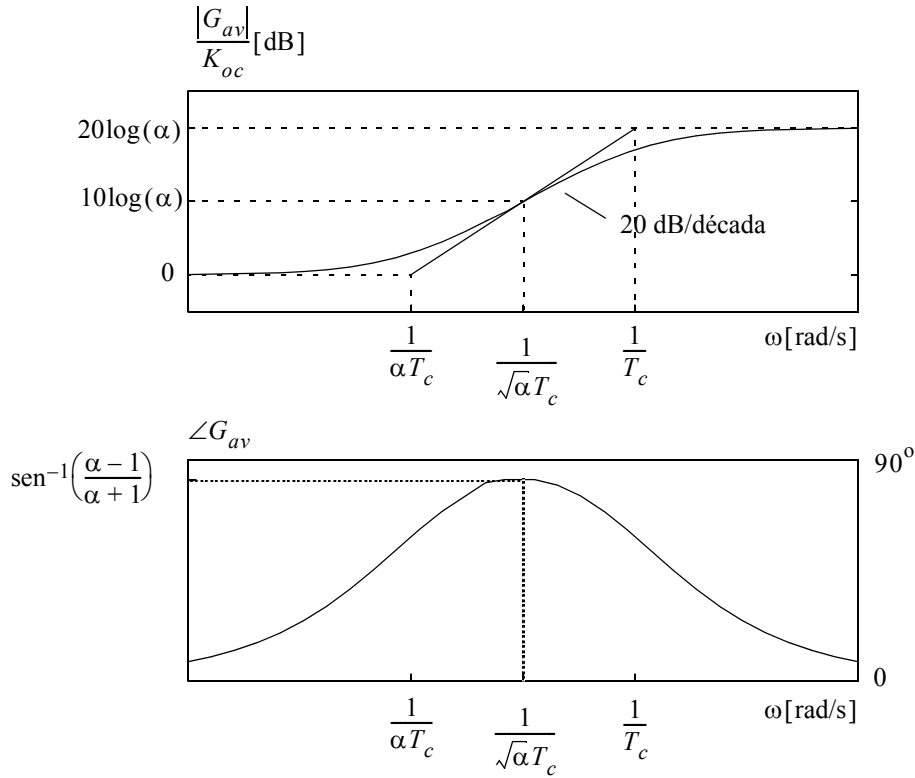
Associando-se o compensador por avanço de fase em série com a planta (veja a Figura 7.2) obtém-se a seguinte função de transferência de malha aberta, para realimentação unitária:

$$G_{av}(s)G_{p/u}(s) = \frac{K_{oc}(1 + s\alpha T_c)}{1 + sT_c} \cdot \frac{KK_p n^2}{s(1 + sT)} \quad (7.2)$$

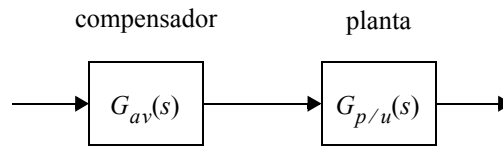
Escolhendo adequadamente os valores de  $\alpha$  e  $T_c$ , pode-se fazer o zero do compensador cancelar o polo do motor, de modo que se tenha

$$G_{av}(s)G_{p/u}(s) = \frac{K_{oc}KK_p n^2}{s(1 + sT_c)} \quad (7.3)$$





**Figura 7.1** Resposta em frequência do compensador por avanço de fase.



**Figura 7.2** Compensador em série com a planta.

Note que, com esse compensador, a dinâmica de malha aberta passa a ser dada por  $T_c$  e não mais pelo polo do motor.

Pode-se escrever (7.3) na forma abaixo, em termos de  $\zeta$  e  $\omega_n$ , obtendo-se:

$$G_{av}(s)G_{p/u}(s) = \frac{K_{oc}KK_p n^2}{s(1+sT_c)} = \frac{\omega_n^2}{s(s+2\zeta\omega_n)}, \quad (7.4)$$

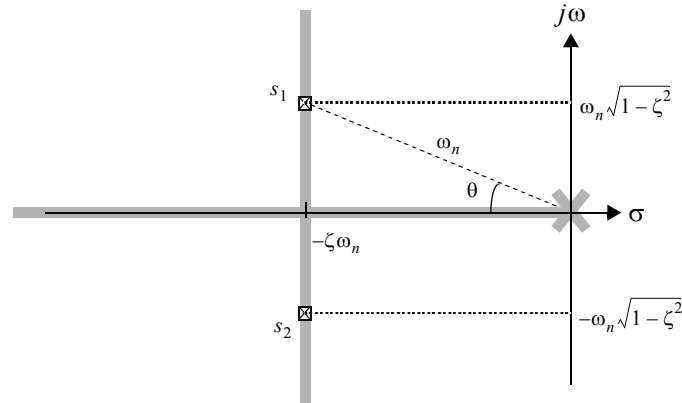
sendo  $\zeta$  o coeficiente de amortecimento, e  $\omega_n$  a frequência natural do sistema. Note que essa é a mesma estrutura da função de transferência de malha aberta quando se usa apenas um controlador proporcional, porém com o polo realocado.

Fechando a malha com realimentação unitária, os valores de  $\zeta$  e  $\omega_n$  determinam a localização no plano  $s$  dos polos do sistema de segunda ordem da equação (7.4), como se pode ver na Figura 7.3.

Valem então as seguintes relações:

$s_{1,2}$  = polos de malha fechada

$$\theta = \cos^{-1}(\zeta)$$



**Figura 7.3** Os parâmetros  $\zeta$  e  $\omega_n$  e sua relação com os polos de malha fechada do sistema.

$$t_s = \frac{4}{\zeta\omega_n} \quad (\text{tempo de acomodação para 2\%}) \quad (7.5)$$

$$t_r = \frac{\pi - \cos^{-1}(\zeta)}{\omega_n\sqrt{1-\zeta^2}} \quad (\text{tempo de subida - 0 a 100\%}) \quad (7.6)$$

$$M_p(\%) = 100e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \quad (\text{sobressinal}) \quad (7.7)$$

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \quad (\text{instante de pico do sobressinal}) \quad (7.8)$$

$$\omega_r = \omega_n\sqrt{1-2\zeta^2} \quad , \zeta < 0,707 \quad (\text{frequência de ressonância}) \quad (7.9)$$

$$M_r = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad , \zeta < 0,707 \quad (\text{pico de ressonância}) \quad (7.10)$$

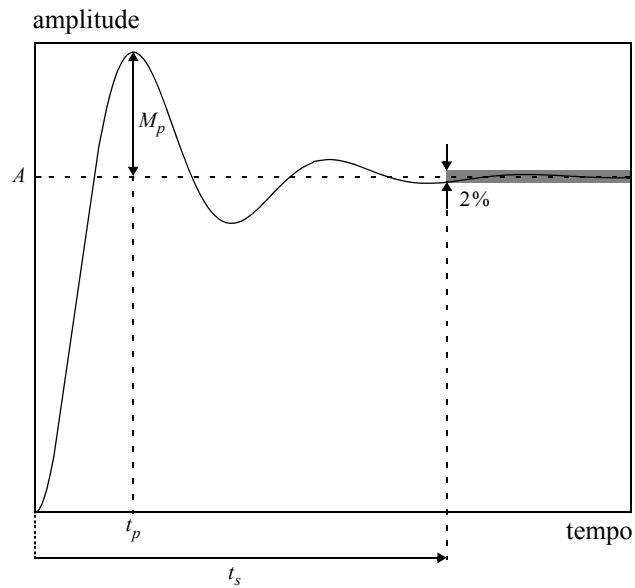
$$\gamma = \tan^{-1}\left(2\zeta\sqrt{\frac{1}{4\zeta^4 + 1 - 2\zeta^2}}\right) \cong 100\zeta \quad (\text{margem de fase em graus}) \quad (7.11)$$

A Figura 7.4 apresenta a resposta do sistema em malha fechada a um degrau de amplitude  $A$ . Na figura estão indicados os valores de  $M_p$ ,  $t_p$  e  $t_s$ . A resposta em frequência do sistema em malha fechada pode ser vista na Figura 7.5. Nessa figura estão indicados  $\omega_n$ ,  $\omega_r$ ,  $M_r$  e  $\gamma$ .

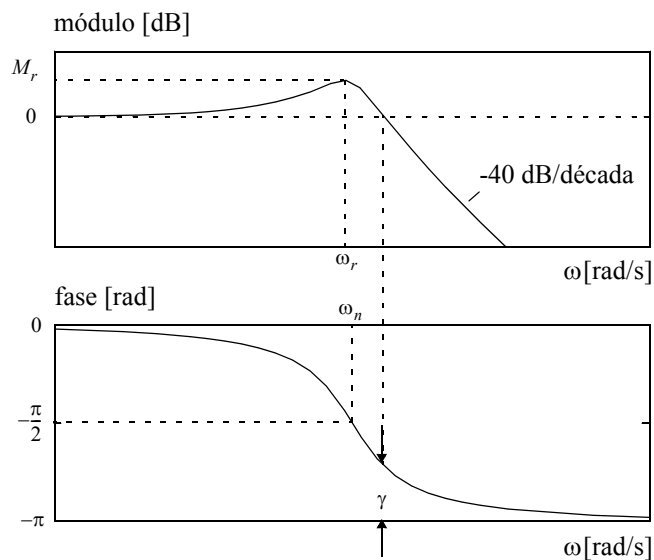
As equações (7.5) a (7.11) podem ser usadas para se determinar um par  $(\zeta, \omega_n)$  que satisfaça especificações de projeto pré-determinadas. Usa-se então (7.4) para se determinar valores de  $K_{oc}$  e  $T_c$  necessários para se obter esse par.

### 7.3.2 Limitações no projeto do compensador

A resposta a um degrau de amplitude  $A$  de um compensador como (7.1) apresenta o aspecto da Figura 7.6. Caso o valor de  $\alpha$  seja muito grande, pode haver problemas de saturação. Se o procedimento de projeto



**Figura 7.4** Resposta a degrau de um sistema de segunda ordem subamortecido.



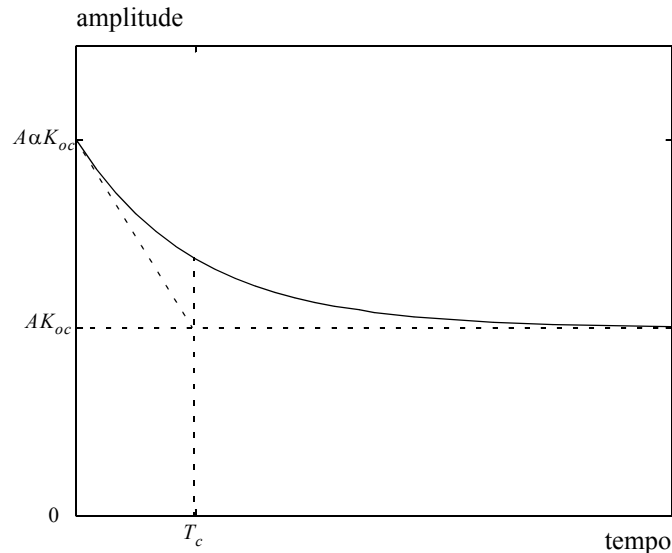
**Figura 7.5** Resposta em frequência de um sistema de segunda ordem subamortecido.

acima gerar um compensador que apresente esse tipo de problema, deve-se estudar o interesse de uma solução diferente, com outro tipo de compensador (e.g. avanço-atraso, PID).

## 7.4 Atividades

### 7.4.1 Atividades prévias

Atividades a serem realizadas antes da realização da experiência. Os resultados obtidos nestes itens deverão estar disponíveis para uso no laboratório.



**Figura 7.6** Resposta a degrau do compensador por avanço de fase.

- a) Projete um compensador por avanço de fase para o controlador de posição (use a função de transferência  $G_{p/u}(s)$  identificada anteriormente). As especificações de desempenho para o sistema em malha fechada são as seguintes:

$$t_r \leq 0,5 \text{ s} \quad M_p \leq 20 \%$$

- b) Determine os valores de  $t_p$ ,  $\gamma$ ,  $\omega_r$  e  $M_r$  para o sistema compensado, assumindo que o modelo utilizado no item 7.4.1-(a) represente fielmente o seu comportamento.

IMPORTANTE: A realização da experiência em tempo hábil requer a realização prévia dos itens acima.

#### 7.4.2 Atividades em sala

Os itens a seguir devem ser feitos no laboratório. Não se esqueça de que os dados tomados no laboratório serão utilizados posteriormente, e por isso a tomada de dados deve ser feita com o rigor necessário.

- c) Implemente o compensador projetado no Simulink e obtenha a resposta a degrau do sistema, determinando  $t_p$  e  $M_p$ . Tome cuidado para que não ocorram saturações durante os ensaios.
- d) Caso o sistema não satisfaça às especificações, ajuste o compensador até obter o desempenho desejado e verifique as fontes de discrepâncias.

#### 7.4.3 Apresentação de resultados

O item desta seção deve ser apresentado ao professor para a realização da Exp. 9 em uma única folha de papel identificada com os nomes dos integrantes de seu grupo.

- e) Apresente o gráfico com as respostas ao degrau dos itens (c) e (d) - um gráfico com duas curvas.

#### 7.4.4 Relatório

Esta experiência deve ser incluída no relatório “Controle de Posição”.

- f) Compare as respostas a degrau obtidas nesta experiência com as respostas a degrau do sistema com o controlador proporcional e com o controlador com realimentação auxiliar de velocidade que foram obtidas nas experiências anteriores.

- g) Compare as respostas a degrau obtidas nesta experiência com as especificações e o desempenho previsto quando do projeto. Analise qualitativamente quaisquer discrepâncias encontradas.

## 7.5 Bibliografia sugerida

(DB98) Dorf, R. C.; Bishop, R. H. **Modern Control Systems**. Addison-Wesley, 8th. ed., 1998.

*A Seção 10.5 apresenta a compensação por avanço no plano “s”. Compensação por avanço, no domínio da frequência, é vista na Seção 10.4.*

(FPE86) Franklin, G. F.; Powell, J. D.; Emami-Naeini, A. **Feedback Control of Dynamic Systems**. Addison-Wesley, 1st. ed., 1986.

*Compensadores por avanço são apresentados na Seção 4.5, juntamente com redes avanço-atraso.*

(Kuo85) Kuo, B. C. **Sistemas de Controle Automático**. Prentice-Hall do Brasil, 4a. ed., 1985.

*Compensação por avanço de fase é vista na Seção 8.3. Na Seção 10.2 ela é apresentada sob um enfoque diferente.*

(Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*A Seção 7.3 trata exclusivamente de compensadores por avanço de fase com diferentes abordagens.*

(CBS11) Castrucci, P. L.; Bittar, A.; Sales, R. M. **Controle Automático**. LTC, 1a. ed., 2011.

*A Seção 5.11 apresenta compensação por avanço, atraso e avanço-atraso.*

atual. RPM/2011a

# Experiência 8

## Controle de velocidade com compensador PI

---

### 8.1 Introdução

Esta experiência tem por objetivos projetar um compensador PI para um sistema de controle de velocidade, verificar o desempenho do sistema compensado em malha fechada e comparar o desempenho de um compensador PI com o de um controlador proporcional (Experiência 5) com relação ao erro estacionário.

### 8.2 Princípios teóricos

Os sistemas de controle usualmente designados como “reguladores” são caracterizados pela existência de um erro de regime finito na resposta temporal de malha fechada a um degrau de entrada. Sua resposta em frequência de malha aberta é caracterizada por um ganho constante e finito nas baixas frequências.

O funcionamento em regime permanente dos reguladores pode ser consideravelmente melhorado pela introdução de um integrador na FTMA (Função de Transferência de Malha Aberta), o que leva o ganho de baixas frequências a infinito e o erro de regime estacionário a zero.

O compensador dinâmico com função de transferência

$$G_c(s) = K_{PI} \left( 1 + \frac{1}{T_I s} \right) \quad (8.1)$$

é denominado um compensador Proporcional+Integral (PI). O procedimento de projeto para se obter um compensador deste tipo consiste em se escolher o ganho  $K_{PI}$  e a constante de integração  $T_I$  de modo que o sistema compensado em malha fechada atenda a especificações de desempenho.

A resposta em frequência de um compensador PI é mostrada na Figura 8.1 e a resposta a um degrau de amplitude  $A$ , na Figura 8.2.

A função de transferência de malha aberta para o controlador de velocidade  $G_{\Omega/V}(s)$ , em série com o compensador PI, é:

$$G_{ma}(s) = G_c(s)G_{\Omega/V}(s) = K_{PI} \left( 1 + \frac{1}{T_I s} \right) \cdot \frac{K}{1 + sT}, \quad (8.2)$$

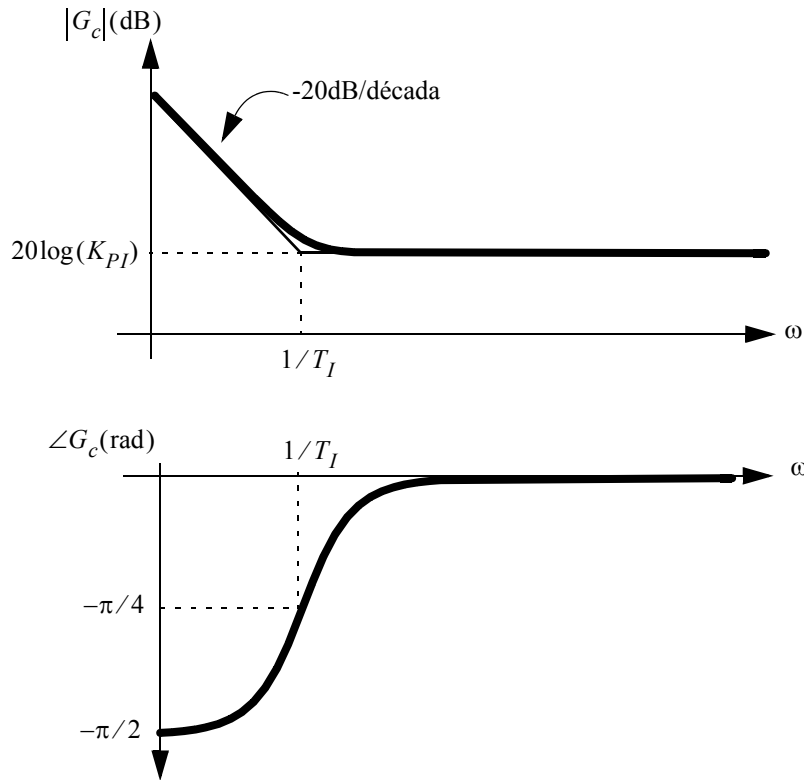
onde  $K_{PI}$  e  $T_I$  são os parâmetros do compensador PI, e  $K$  e  $T$  são os parâmetros da equação (5.7) (Exp 5).

Escolhendo-se o parâmetro  $T_I$  de modo a cancelar o pólo de  $G_{\Omega/V}(s)$ , a função de transferência de malha aberta torna-se:

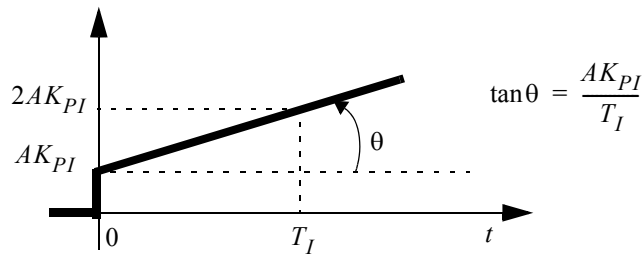
$$G_{ma}(s) = G_c(s)G_{\Omega/V}(s) = \frac{KK_{PI}}{T_I s} = \frac{K_{ma}}{s}, \quad (8.3)$$

onde  $K_{ma}$  é o ganho total de malha aberta.

O ganho  $K_{ma}$  pode ser obtido por uma especificação sobre o transitório como, por exemplo, o tempo



**Figura 8.1** Resposta em frequência de compensador PI.



**Figura 8.2** Resposta temporal de um compensador PI a um degrau de amplitude  $A$ .

de subida.

## 8.3 Atividades

### 8.3.1 Atividades prévias

Atividades a serem realizadas antes da realização da experiência. Os resultados obtidos nestes itens deverão estar disponíveis para uso no laboratório.

- Usar  $T_I$  igual à constante de tempo da planta identificada na Experiência 4. Nessas condições, determine a função de transferência de malha fechada do controlador de velocidade e o correspondente erro estacionário para entradas do tipo degrau.
- Nas condições do item anterior, deduza uma expressão para o valor do ganho do compensador PI ( $K_{PI}$ ) em função do tempo de subida da resposta temporal do sistema em malha fechada para entradas do tipo degrau.

- c) Usando a expressão desenvolvida no item anterior, obtenha valores de ganho do compensador PI para os seguintes tempos de subida:  $t_{r1} = 0,3$  s,  $t_{r2} = 0,5$  s e  $t_{r3} = 1,0$  s.

OBS: Use a definição de tempo de subida de sua preferência e justifique a escolha.

- d) Analise, através do Lugar Geométrico das Raízes, o que acontece com o desempenho do sistema em malha fechada se o cancelamento do polo do motor for imperfeito.

IMPORTANTE: A realização da experiência em tempo hábil requer a realização prévia dos itens acima.

### 8.3.2 Atividades em sala

- e) Monte um diagrama do Simulink com o compensador PI para controlar o servomecanismo.
- f) Verifique o cancelamento do pólo da planta, através do ajuste da resposta a degrau do compensador PI, ou seja, ajuste o compensador de maneira a obter e verificar o cancelamento do polo da planta. Comente o método e os resultados obtidos.
- g) Levante as respostas a degrau correspondentes aos ganhos do item (c) e verifique se os valores projetados estão coerentes. Justifique eventuais diferenças.
- h) Compare as respostas obtidas no item anterior com as respostas a degrau obtidas na Experiência 5, com relação a tempo de subida e erro estacionário.
- i) Compare também com as simulações dos modelos linear e não-linear.
- j) Aplique um degrau como sinal de referência para o controlador de velocidade e aguarde até o sistema atingir o regime permanente. Aplique em seguida o freio eletromagnético e registre a resposta do sistema. Compare o resultado obtido com o correspondente da Experiência 5 e justifique as diferenças observadas.

### 8.3.3 Apresentação de resultados

O item desta seção deve ser apresentado ao professor para a realização da Exp. 9 em uma única folha de papel identificada com os nomes dos integrantes de seu grupo.

- k) Apresente o gráfico com as respostas ao degrau do item (g) - um gráfico com três curvas.

### 8.3.4 Relatório

Esta experiência deve ser incluída no relatório “Controle de Velocidade”.

## 8.4 Bibliografia sugerida

- (DB98) Dorf, R. C.; Bishop, R. H. **Modern Control Systems**. Addison-Wesley, 8th. ed., 1998.

*A Seção 7.7 trata de controladores PID (O controlador PI sendo um caso particular) do ponto de vista do Lugar Geométrico das Raízes.*

- (Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*Veja as Seções 7.4 e 3.2.*

- (CBS11) Castrucci, P. L.; Bittar, A.; Sales, R. M. **Controle Automático**. LTC, 1a. ed., 2011.

*O Capítulo 6 trata de controladores PID sob diversos aspectos.*





# Experiência 9

## Controladores PID: Abordagem prática

---

### 9.1 Objetivos

Esta experiência tem por objetivo o estudo, implementação e sintonia de controladores PID (de Proporcional+Integral+Derivativo) aplicados ao servomecanismo do Laboratório

### 9.2 Introdução aos PIDs

#### 9.2.1 Controladores industriais

A maioria absoluta dos controladores empregados na Indústria é baseada em algoritmos PID. Eles são empregados desde a década de 30 em diferentes implementações. Os primeiros eram implementados por meio de sistemas hidráulicos ou pneumáticos, tendo evoluído depois para eletrônica analógica, sendo hoje implementados digitalmente por meio de microprocessadores.

Os PIDs aparecem em inúmeros produtos industriais para controle de processos, desde os mais simples, chamados de *single-loops* ou *multi-loops*, equipamentos dedicados em tamanho padrão, que implementam apenas algoritmos PIDs, capazes de controlar uma ou diversas malhas (como os da Figura 9.1) até os mais



(a) SMAR CD 600 (Brasil)



(b) Foxboro 718 (EUA)



(c) Yokogawa YS100 (Japão)

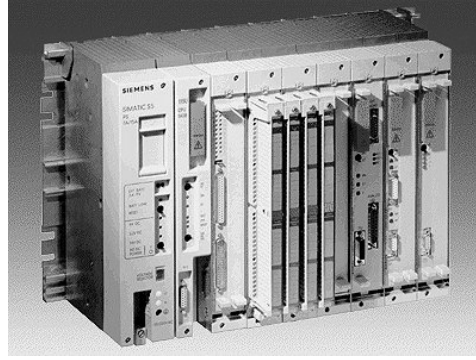
**Figura 9.1** Alguns controladores industriais de pequeno porte

sofisticados equipamentos, que são programados graficamente (de maneira semelhante ao Simulink, utilizado no Laboratório) onde os PIDs aparecem como blocos básicos utilizados para compor sistemas de controle que podem conter centenas de malhas (um equipamento desse tipo é apresentado na Figura 9.2).

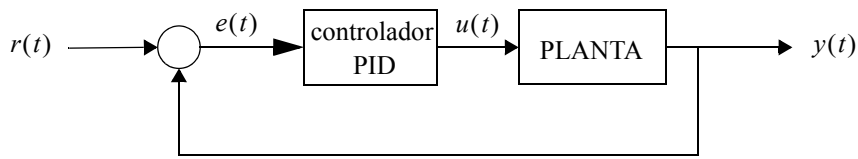
Conceitualmente falando, os modernos PIDs não são diferentes de seus ancestrais das décadas de 30 e 40. A evolução se deu principalmente no que se refere a métodos de sintonia (isto é, a escolha dos ganhos do controlador), e hoje diversos produtos contêm PIDs que se auto-sintonizam, utilizando as mais variadas técnicas.

#### 9.2.2 Formulação básica do controlador

Em um controlador PID, como o do diagrama abaixo (Figura 9.3), a ação de controle é gerada por uma soma de três termos, isto é



**Figura 9.2** Um sistema de controle de maior porte (Siemens S-5).



**Figura 9.3** Controle PID

$$u(t) = u_P(t) + u_I(t) + u_D(t), \quad (9.1)$$

onde os termos se referem às parcelas proporcional, integral e diferencial respectivamente.

Essas parcelas são dadas por

$$u_P(t) = K_P e(t), \quad (9.2)$$

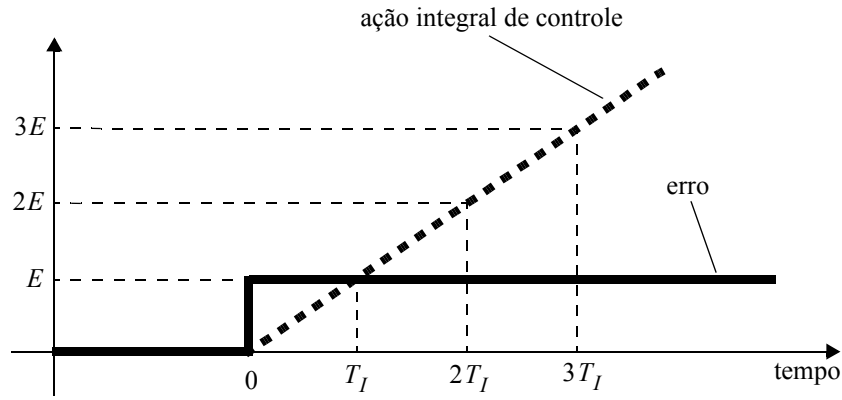
$$u_I(t) = \frac{K_P}{T_i} \int_0^t e(\tau) d\tau, \text{ ou no domínio da frequência } U_I(s) = \frac{K_P}{sT_i} E(s) \text{ e} \quad (9.3)$$

$$u_D(t) = K_P T_D \frac{de(t)}{dt} \text{ ou } U_D(s) = K_P T_D s E(s). \quad (9.4)$$

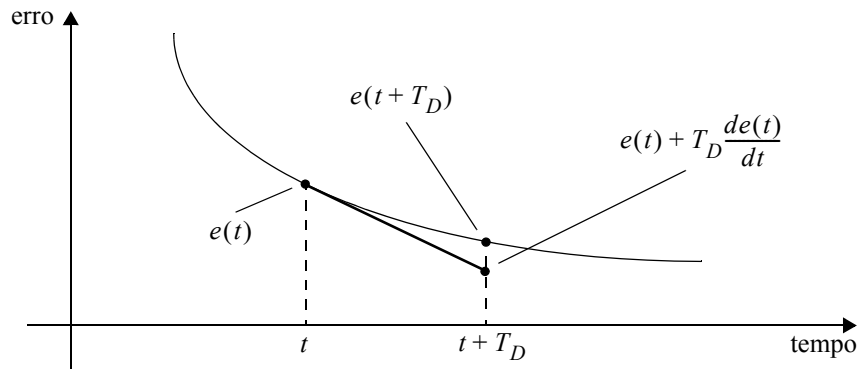
De forma que a equação total do controlador no domínio da frequência é

$$U(s) = K_P \left( 1 + \frac{1}{sT_i} + sT_D \right) E(s). \quad (9.5)$$

Esta parametrização do controlador em termos de  $K_P$ ,  $T_i$  e  $T_D$  é chamada de *forma padrão* ou *forma ISA* (de Instrument Society of America - atualmente International Society of Automation). Ela, é claro, não é a única possível, mas apresenta algumas vantagens. A principal é que os ganhos assumem significados físicos.  $K_P$  representa a amplificação geral da ação de controle, enquanto que  $T_i$  representa, para um sinal de erro constante, o tempo que a ação integral de controle varia de 0 até um valor igual ao erro, conforme ilustrado no gráfico da Figura 9.4. Por esse motivo,  $T_i$  é usualmente denominado *tempo de repetição*. A ação diferencial pode ser entendida como dando um caráter antecipativo ao controle. Considere o gráfico da Figura 9.5, onde é apresentado um sinal de erro variando no decorrer do tempo. Pode-se ver no gráfico que a ação de controle diferencial no instante  $t$  é baseada em uma extrapolação linear do sinal de erro  $e(t)$   $T_D$  instantes de tempo à frente, ou seja  $e(t) + T_D \frac{de(t)}{dt}$ . Por esse fato,  $T_D$  é usualmente denominado *tempo diferencial* (ou *tempo derivativo*) ou *horizonte de predição* e possui um significado físico palpável.



**Figura 9.4** Ilustração de uma ação integral de controle



**Figura 9.5** Ilustração de uma ação diferencial de controle

Outra parametrização popular é a chamada *forma paralela*, dada por

$$U(s) = \left( k + \frac{k_i}{s} + s k_d \right) E(s). \quad (9.6)$$

Na parametrização paralela os parâmetros perdem seu significado físico, porém esta parametrização é mais interessante quando se deseja tratar separadamente as diferentes ações de controle ou para se fazer manipulações algébricas.

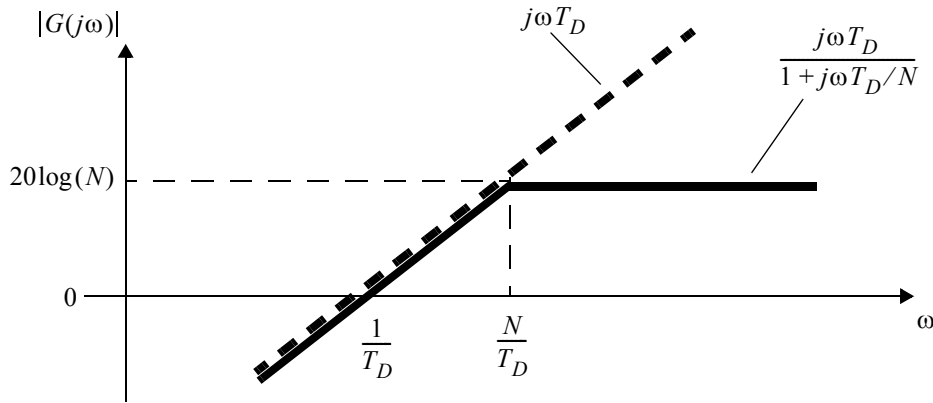
### 9.2.3 Implementação da ação diferencial

Uma ação diferencial como a da equação (9.4) apresenta alguns problemas de implementação.

O primeiro deles se refere à variação do ganho com a frequência do sinal de entrada. Uma função de transferência do tipo  $sT_D$  apresenta ganho tendendo a infinito com o aumento da frequência, o que é altamente indesejável do ponto de vista de controle, pois sinais espúrios de alta frequência, como ruídos de medida, seriam amplificados em escala muito maior que o sinal de erro pertinente ao controle. Além disso é sabidamente impossível se realizar fisicamente uma implementação desse tipo. Geralmente é feita a seguinte aproximação

$$sT_D \cong \frac{sT_D}{1 + sT_D/N}, \quad (9.7)$$

onde  $N$  é tipicamente escolhido como um valor entre 5 e 20 (veja a Figura 9.6).



**Figura 9.6** Diagrama de Bode da ação diferencial (apenas as assíntotas foram traçadas)

O outro problema é que usualmente o sinal de referência (ou *setpoint*) é um sinal constante por trechos (como os degraus aplicados aos diferentes sistemas implementados no Laboratório), de maneira que o sinal de erro pode vir a ser descontínuo, o que do ponto de vista da ação diferencial é bastante inconveniente. Uma maneira simples de se evitar esse problema é aplicar a saída da planta com sinal trocado à parcela diferencial do controlador, e não o sinal de erro, tendo-se então no lugar de (9.4)

$$U_D(s) = \frac{-sT_D}{1 + sT_D/N} Y(s), \quad (9.8)$$

de forma que o controlador PID fica (no domínio da frequência)

$$U(s) = K_P \left( 1 + \frac{1}{sT_i} \right) E(s) + K_P \left( \frac{-sT_D}{1 + sT_D/N} \right) Y(s). \quad (9.9)$$

### 9.2.4 Wind up e anti-windup

Frequentemente, durante a operação de um sistema de controle, ocorre da saída do controlador saturar sem que a saída da planta atinja o setpoint, especialmente no caso de saturação do atuador (e. g. uma válvula que abre ou fecha totalmente, um amplificador que satura, um motor que atinge sua velocidade máxima, etc.) ou caso o setpoint escolhido esteja fora da capacidade da planta (e. g. especificar uma vazão ou nível maior que a capacidade do sistema ou querer que o fusquinha do Prof. Walter Kaiser acelere até atingir 180 km/h). Nessa situação, caso o controlador possua características de integrador, a saída calculada crescerá cada cada vez mais (dado que um erro não nulo será constantemente integrado), mesmo que a saída física já tenha saturado e não seja capaz de fazer com que o erro se anule.

Essa situação em que a componente integral do controle cresce cada vez mais com a saída física do controlador saturada é denominada *wind up*. O real problema não é o *wind up* em si, mas a volta do sistema a uma situação normal após algum tempo em *wind up*. Caso se altere o setpoint ou se corrija algum problema que estivesse causando a saturação do atuador, o controlador demorará muito tempo a sair da saturação, pois o integrador terá nesse momento um valor muito alto, que cairá lentamente de acordo com a constante de tempo implementada.

Para evitar que esse problema ocorra, basta que o integrador seja congelado quando a saída do controlador atingir o máximo ou mínimo de sua excursão admissível, de modo que não haja acumulação desnecessária no integrador, impedindo a ocorrência de *wind up*. A essa estratégia se dá o nome de *anti-windup*.

## 9.3 Sintonia de PIDs

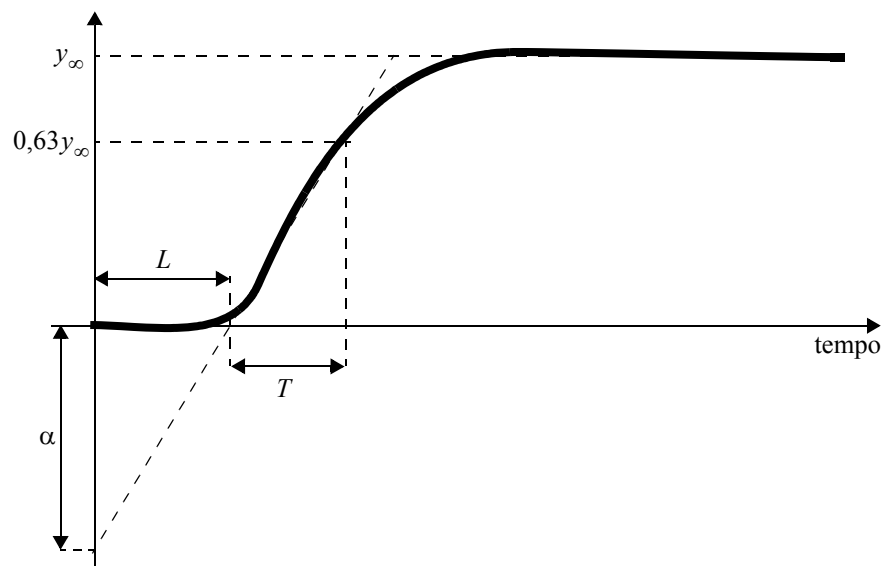
A sintonia de um controlador PID, ou seja, a escolha adequada de seus parâmetros, pode ser realizada

de diversas maneiras, tendo em vista a otimização de algum índice de desempenho. Exemplos de índices comuns de desempenho são o erro de regime estacionário (ou acompanhamento de setpoint), rejeição de perturbações, robustez a variações na planta, máximo sobressinal, etc. A obtenção de parâmetros adequados pode ser feita analiticamente a partir dos modelos do controlador e da planta, por tentativa e erro, por processos iterativos e por utilização direta de regras práticas de sintonia.

Regras práticas de sintonia têm sido desenvolvidas ao longo do tempo com o fim de se caracterizar de maneira rápida uma dada planta e a partir dessa caracterização se obter uma sintonia para o controlador. Três regras bastante conhecidas são apresentadas nesta apostila. As duas primeiras usam a resposta a degrau da planta em malha aberta para caracterizá-la, enquanto que a terceira utiliza propriedades de sua resposta em frequência.

### 9.3.1 Método de Ziegler-Nichols para o domínio do tempo

Considere um sistema estável em malha aberta cuja resposta a degrau tenha o aspecto ilustrado na Figura 9.7. A partir dos parâmetros identificados  $\alpha$  e  $L$  pode-se obter uma sintonia utilizando-se o método de



**Figura 9.7** Parâmetros de uma resposta a degrau típica

Ziegler-Nichols para resposta a degrau (Tabela 9.1) ou uma variação que permite tentar ajustar o sobressinal

Controlador	$K_P$	$T_I$	$T_D$
P	$1/\alpha$		
P+I	$0,9/\alpha$	$3L$	
P+I+D	$1,2/\alpha$	$2L$	$0,5L$

**Tabela 9.1:** Método de Ziegler-Nichols para resposta a degrau

da resposta em malha fechada, o método de Chien-Hrones-Reswick (Tabela 9.2).

Sobressinal	0%			20%		
Controlador	$K_P$	$T_I$	$T_D$	$K_P$	$T_I$	$T_D$
P	$0,3/\alpha$			$0,7/\alpha$		
P+I	$0,6/\alpha$	$4L$		$0,7/\alpha$	$2,3L$	
P+I+D	$0,95/\alpha$	$2,4L$	$0,42L$	$1,2/\alpha$	$2L$	$0,42L$

Tabela 9.2: Método de Chien-Hrones-Reswick para resposta a degrau

### 9.3.2 Método de Ziegler-Nichols para o domínio da frequência

Outro método, também devido a Ziegler e Nichols, se aplica quando a resposta em malha fechada com controlador proporcional tem caráter oscilatório. Ela pode ser esquematizada da seguinte maneira: Inicialmente feche a malha apenas com controle proporcional, isto é,  $T_D = 0$  e  $T_I \rightarrow \infty$ . Aumente o ganho proporcional  $K_P$  até que se atinja o limite de estabilidade, situação em que o sistema passa a oscilar espontaneamente e praticamente sem amortecimento. O ganho limite  $K_u$  e o período de oscilação  $T_u$  levam à regra de sintonia do método de Ziegler-Nichols para resposta em frequência, apresentado na Tabela 9.3

Controlador	$K_P$	$T_I$	$T_D$
P	$0,5K_u$		
P+I	$0,45K_u$	$T_u/1,2$	
P+I+D	$0,6K_u$	$T_u/2$	$T_u/8$

Tabela 9.3: Método de Ziegler-Nichols para oscilação espontânea em malha fechada

### 9.3.3 Método do Relé

Seja um relé com histerese, conforme a curva característica esquematizada na 9.8. A saída  $u$  do relé

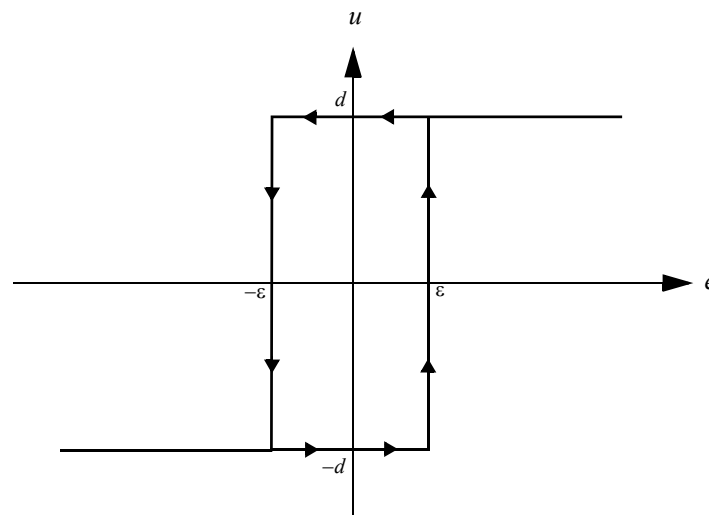


Figura 9.8 Relé com histerese

pode assumir os valores  $\pm d$  conforme o valor da entrada  $e$  e da trajetória (ascendente ou descendente) em que o sistema se encontra. A histerese do relé é representada pelo parâmetro  $\varepsilon$ . Se  $\varepsilon = 0$  o relé não apresenta histerese.

Para se aplicar o método do relé, fecha-se a malha utilizando o relé acima como compensador, con-

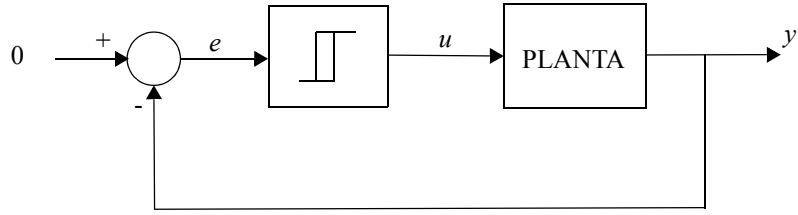


Figura 9.9 Compensador a relé

forme a Figura 9.9. Caso o arranjo gere uma oscilação sustentada na saída  $y(t)$  do sistema, como a da Figura 9.10, é possível identificar os parâmetros  $a$  (amplitude de oscilação) e  $T_u$  (período de oscilação). Nessas con-

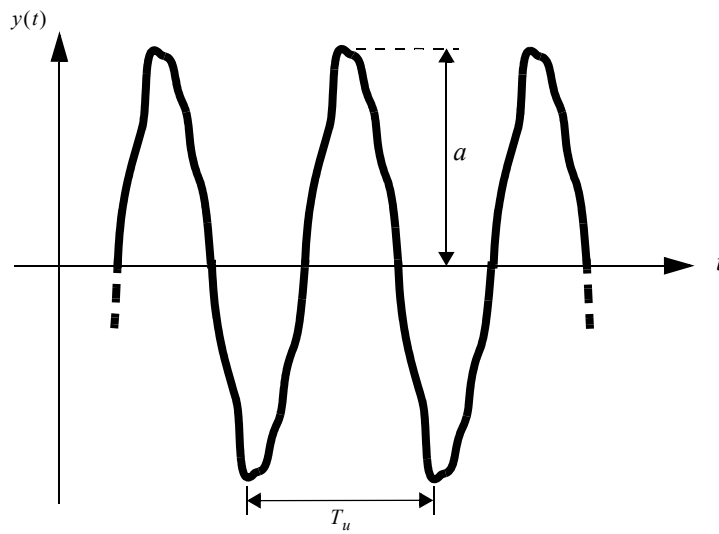


Figura 9.10 Oscilação sustentada

dições, o sinal  $u(t)$  deve ser uma onda quadrada de amplitude  $d$ , mesmo período e fase oposta ao sinal  $y(t)$ .

Para o caso mais simples em que não há histerese (ou em que ela é muito pequena), pode-se mostrar que para a oscilação sustentada acima, a função descritiva  $\Gamma$  do compensador a relé é dada por

$$\Gamma(a) = \frac{4d}{\pi a}, \quad (9.10)$$

derivada grosseiramente da componente fundamental da expansão em série de Fourier de uma onda quadrada. O que nos sugere o seguinte valor para o ganho limite.

$$K_u = \Gamma(a) = \frac{4d}{\pi a}. \quad (9.11)$$

Com isso, pode-se utilizar os valores obtidos de  $K_u$  e  $T_u$  na Tabela 9.3 para ajustar um compensador PID. (9.12)

O método do relé apresenta uma vantagem importante em relação ao método de Ziegler-Nichols no domínio da frequência: Não é necessário levar o sistema à instabilidade para a obtenção dos parâmetros do



controlador. Além disso, pode-se ajustar o parâmetro  $d$  do relé para se obter uma amplitude de entrada adequada a cada planta.

Para o caso em que não se quer ou não se pode adotar histerese nula, pode-se mostrar que a função descritiva do compensador a relé é dada por

$$\Gamma(a) = \frac{4d}{\pi a} \left( \sqrt{1 - \left(\frac{\varepsilon}{a}\right)^2} - j\frac{\varepsilon}{a} \right), \quad (9.13)$$

o que nos sugere o ganho limite

$$K_u = |\Gamma(a)|. \quad (9.14)$$

A histerese pode ser útil para evitar que ruídos ocasionem o chaveamento indevido do relé, para limitar a frequência de chaveamento ou para adequar o modelo do relé a dispositivos reais com histerese conhecida.

### 9.3.4 Advertência sobre os métodos

É importante ter em mente que esses métodos são baseados em aproximações e hipóteses que podem não se aplicar a todos os tipos de sistema, sendo que as regras apresentadas acima podem eventualmente levar a resultados bastante pobres ou nem serem passíveis de aplicação. Há casos em que a resposta a degrau de um dado sistema não apresenta os parâmetros  $\alpha$  e  $L$  bem definidos ou em que se aumentando o ganho do controlador proporcional não ocorra instabilidade ou ainda em que seja possível se atingir o limite de instabilidade, mas que essa situação seja proibida pela operação normal da planta ou mesmo perigosa. Os métodos de Ziegler-Nichols assumem um decaimento do sinal de  $1/4$ , isto é, o segundo pico da resposta a degrau do sistema deve apresentar um erro 4 vezes menor que o primeiro, o que para algumas aplicações pode ser uma característica indesejável.

Pode ser impossível obter uma sintonia pelo método do relé caso o sistema não apresente comportamento oscilatório, ou mesmo que seja possível, não há garantias de que a sintonia seja adequada ou que o comportamento esperado seja atingido.

### 9.3.5 Outras técnicas

Nas experiências anteriores vimos diversas formas de se sintonizar analiticamente controladores semelhantes a um PID (por exemplo o controle proporcional com realimentação auxiliar de velocidade, semelhante a um controlador PD, ou o controle por avanço de fase ou o próprio controlador PI da Experiência 8).

Uma das abordagens mais utilizadas na Indústria, tem sido a sintonia por tentativa e erro, e talvez seja esse o motivo de ser tão comum se encontrar controladores mal sintonizados ou com a ação diferencial desativada. De fato, ajustar três parâmetros por tentativa e erro com conhecimento limitado sobre o comportamento da planta pode-se tornar uma tarefa bastante difícil e eventualmente demorada.

## 9.4 Atividades

### 9.4.1 Atividades em sala

- Implemente no Simulink um controlador PID como o da equação (9.9) com o fim de controlar o servomecanismo.
- Das regras práticas apresentadas, quais podem ser utilizadas no sistema do laboratório (considere tanto posição como velocidade como saída do sistema)? Justifique sua escolha.

- c) Utilize o método do relé, com histerese nula ou muito pequena, para ajustar os parâmetros do PID para controlar a posição do servomecanismo nas configurações da Tabela 9.4

	Amplitude de $d$	Compensador	Observações
i)	baixa	PI	
ii)	baixa	PID	Explore a influência do parâmetro $N$
iii)	elevada	PI	
iv)	elevada	PID	Explore a influência do parâmetro $N$

Tabela 9.4: Roteiro para ajuste do controlador de posição

Apresente as respostas a degrau correspondentes (escolha a amplitude do degrau para evitar saturação do controlador) e comente as diferenças. Seria de se esperar que o ajuste se alterasse com a mudança da amplitude  $d$ ?

- d) Tente refinar o ajuste do item anterior por tentativa e erro. Comente os resultados obtidos.

#### 9.4.2 Relatório

Esta experiência deve ser incluída nos relatórios “Controle de Velocidade” e “Controle de Posição”.

### 9.5 Bibliografia sugerida

(Lev96) Levine, W. S., ed. **The Control Handbook**. CRC-IEEE, 1996.

*O texto apresentado aqui é baseado principalmente na Seção 10.5 deste livro.*

(Oga93) Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, 2a. ed., 1993.

*Sintonia de PID's é apresentada na Seção 7.6. As ações básicas de controle são apresentadas na Seção 3.2.*

(Shi94) Shinskey, F. G. **Feedback Controllers for the Process Industries**. McGraw-Hill, 1994.

*Contém informações extensivas sobre controladores PID, sintonia e aplicações.*

(Ast95) Åström, K.; Hägglund, T. **PID Controllers: Theory, Design and Tuning**. Instrumentation Society of America, 2nd ed., 1995.

*O método do relé é discutido na Seção 2.6. O livro é todo dedicado a controladores PID.*



# Experiência 10

## Atraso de transporte e controlador de modelo interno

---

### 10.1 Introdução

Esta experiência tem por objetivos servir como uma breve e simplificada apresentação a sistemas com atraso de transporte, introduzir controladores de modelo interno, conhecidos por IMC (do inglês *Internal Model Controllers*) e usá-los para realizar o controle da velocidade angular do servomecanismo. O desempenho será comparado com um controlador do tipo PI, como o da Experiência 8, fechando a malha do servomecanismo com um atraso de transporte.

### 10.2 Sistemas com atraso

Sistemas com atraso de transporte são bastante comuns em aplicações de controle, e representam um desafio para o projetista, resultando em dificuldades tanto para estabilização como para se atingir um desempenho adequado.

O atraso de transporte pode ser entendido como a existência de um intervalo de tempo entre uma atuação sobre um dado sistema e a manifestação de seus efeitos sobre o comportamento deste. Veja a Figura 10.1 para uma ilustração simples de sistemas com atraso.

Se a duração do atraso for constante ou se ela depender apenas do tempo, o fenômeno é certamente linear, e para o caso de duração constante, o atraso tem uma representação simples e elegante no domínio de Laplace,

$$\Theta(s) = e^{-\theta s}, \quad (10.1)$$

onde  $\theta$  é a duração do atraso. A função de transferência acima possui ganho unitário em toda a faixa de frequências e defasagem aumentando linearmente com a frequência. Tendo essas características em mente, é fácil entender como a presença de atraso causa complicações ao projeto de controle.

### 10.3 Uma proposta de atraso para o servomecanismo

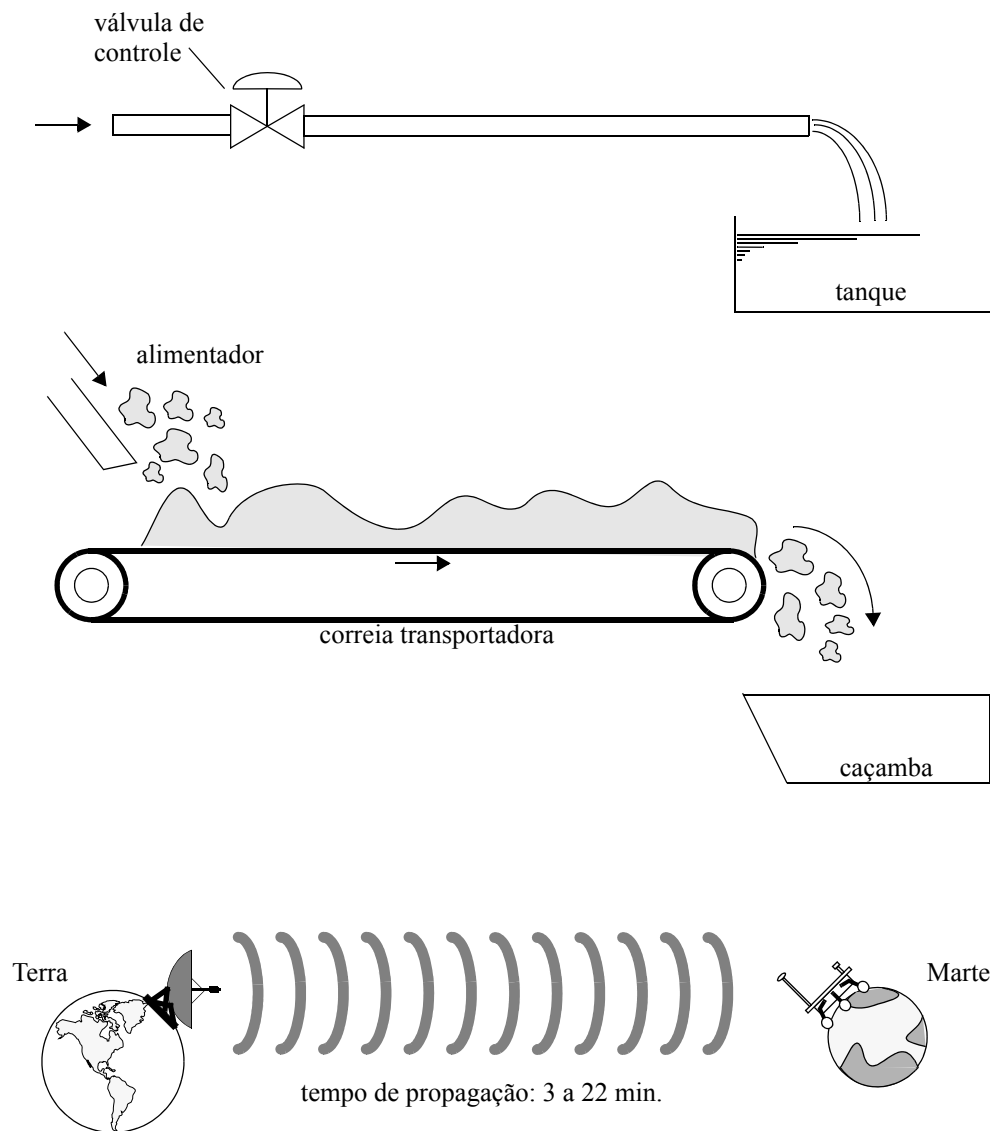
Como o servomecanismo do laboratório não apresenta atraso de transporte, um atraso na entrada do sistema de duração constante foi artificialmente introduzido através do computador. A introdução deste atraso afeta consideravelmente a dinâmica do sistema, que passa a requerer estratégias de controle capazes de lidar com esse efeito.

Com a introdução do atraso, a função de transferência do sistema pode ser descrita por

$$H_{V/u}(s) = e^{-\theta s} \frac{KK_t}{Ts + 1}, \quad (10.2)$$

onde  $\theta$  é a duração do atraso de transporte. Por simplicidade abreviaremos a notação, fazendo  $H(s) = H_{V/u}(s)$ .

O servomecanismo com atraso pode ser acessado através do diagrama de simulação



**Figura 10.1** Exemplos de sistemas com atraso de transporte

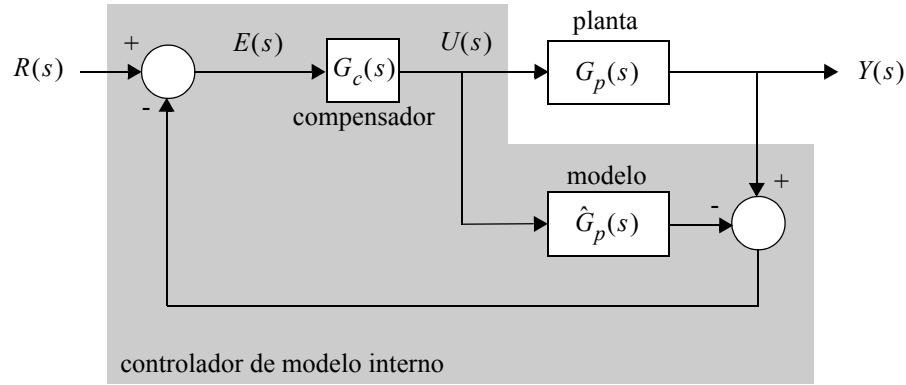
`delay_template`. Note que a menos do atraso, o comportamento do sistema é totalmente similar ao observado anteriormente.

## 10.4 Controladores de modelo interno

Os controladores de modelo interno constituem-se numa classe de sistemas de controle que se distinguem por conterem em sua implementação explicitamente um modelo do processo a ser controlado. Na abordagem usual de controle por realimentação, um modelo da planta pode ser utilizado para o projeto do sistema de controle, mas a implementação contém apenas filtros e compensadores. Com isso, a estrutura básica para os IMCs difere um pouco da usual, como pode ser visto na Figura 10.2.

Esta estrutura permite que se trate sem grandes dificuldades efeitos como o atraso de transporte, o que facilita o projeto de sistemas de controle com bom desempenho, bastando que um modelo fiel do processo seja disponível, e que alguns requisitos técnicos sejam satisfeitos.

Considere o esquema de controle da Figura 10.2. Para essa estrutura de controle, pode-se mostrar com relativa facilidade que



**Figura 10.2** Esquema típico do IMC (para sistemas lineares invariantes no tempo)

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)(G_p(s) - \hat{G}_p(s))} \quad \text{e} \quad (10.3)$$

$$\frac{E(s)}{R(s)} = \frac{1 - G_c(s)\hat{G}_p(s)}{1 + G_c(s)(G_p(s) - \hat{G}_p(s))}. \quad (10.4)$$

Supondo que o modelo  $\hat{G}_p(s)$  seja uma descrição quase perfeita da planta  $G_p(s)$  ou seja, que  $\hat{G}_p(s) \cong G_p(s)$ , podemos simplificar as equações acima obtendo

$$\frac{Y(s)}{R(s)} \cong G_c(s)G_p(s) \quad \text{e} \quad (10.5)$$

$$\frac{E(s)}{R(s)} \cong 1 - G_c(s)\hat{G}_p(s). \quad (10.6)$$

Note que a função de transferência aproximada de malha fechada (10.5) depende da planta  $G_p(s)$  e que a função de transferência do erro em malha fechada (10.6) depende do modelo  $\hat{G}_p(s)$ .

Supondo que o objetivo do sistema de controle seja rastrear o sinal de referência  $R(s)$ , seria desejável que tivéssemos  $G_c(s)\hat{G}_p(s) = 1$  ou seja, que  $G_c(s) = \hat{G}_p(s)^{-1}$ . Nessa situação a saída da planta  $Y(s)$  seria uma cópia fiel do sinal de referência e o sistema de controle teria um desempenho perfeito.

A condição de estabilidade do sistema em malha fechada nesse caso é simplesmente que tanto  $G_c(s)$  como  $G_p(s)$  sejam estáveis (pense no porquê).

Apesar de desejável, a estratégia acima cria uma série de problemas e conflitos que a tornam praticamente inviável. Para citar alguns: a inversão do modelo somente pode ser feita se  $\hat{G}_p(s)$  for uma função de transferência biprópria (em qualquer outra situação a função ou a sua inversa seriam impróprias);  $\hat{G}_p(s)$  não pode ter polos ou zeros no semi-plano direito, pois eles seriam cancelados por  $G_c(s)$ , o que torna o sistema em malha fechada instável.

Uma solução de compromisso é escolher

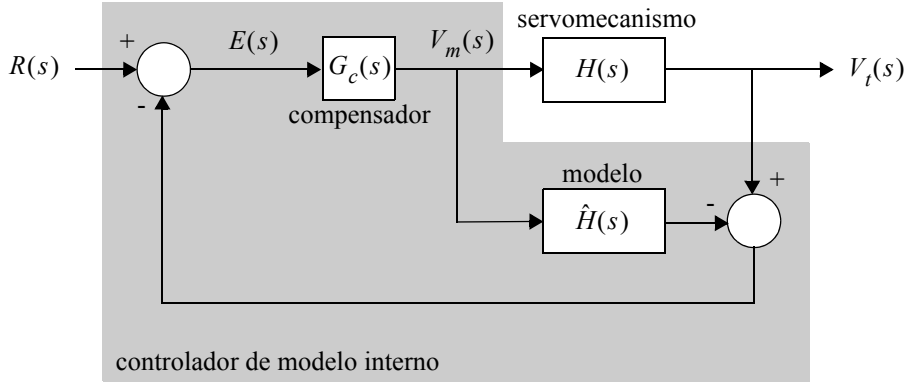
$$G_c(s)\hat{G}_p(s) = G_f(s), \quad (10.7)$$

onde  $G_f(s)$  é uma função de transferência pré-escolhida para evitar os problemas acima e garantir algum tipo de desempenho ao sistema. Nesse contexto, o projeto de controle se resume a uma escolha adequada de  $G_f(s)$ , e há diversas técnicas e abordagens para esse fim. Aqui exploraremos apenas uma abordagem muito

simples, adequada ao problema de controle que temos em mão.

## 10.5 Uma proposta de controle para o servomecanismo com atraso

Seja o diagrama a seguir, na versão para o controle do servomecanismo com atraso.



**Figura 10.3** Esquema IMC para o servomecanismo com atraso

De (10.5) e supondo que o modelo seja perfeito, temos que a função de transferência de malha fechada é dada por  $G_f(s)$ . Definimos então os seguintes requisitos para o controlador.

- $G_f(s)$  deve ser estável, por razões óbvias.
- $G_f(s)$  deve ser um filtro passa-baixa. A ideia é imunizar o sistema a ruídos (evitando a amplificação excessiva de sinais de alta frequência) e minimizar a agressividade do controle, limitando a banda do sistema em malha fechada, pois sabe-se que bandas excessivamente largas são associadas a respostas temporais muito rápidas, que para serem atingidas exigem sinais de controle muito rápidos e de amplitude elevada.
- Erro de regime nulo para entrada degrau, que é um requisito quase sempre desejável.
- $G_c(s)$  deve ser uma função de transferência própria, por questões de causalidade.
- $G_c(s)$  deve ser uma função de transferência estável.

### 10.5.1 Um controlador simples

Como vimos,

$$G_c(s) = G_f(s)\hat{H}(s)^{-1}, \quad (10.8)$$

ou seja

$$G_c(s) = G_f(s) \cdot \frac{e^{\theta s}(Ts + 1)}{KK_t}. \quad (10.9)$$

Uma proposta imediata, e que atende os requisitos acima é adotar a dinâmica mais simples possível para a função de transferência de malha fechada, que é aproximada por  $G_f(s)$ ,

$$G_f(s) = \frac{e^{-\theta s}}{Ts + 1}. \quad (10.10)$$

Como  $\hat{H}(s)^{-1}$  possui um avanço de transporte e é imprópria, é necessário cancelar a sua dinâmica, como é feito ao se adotar  $G_f(s)$  como em (10.10). Note que com isso a função de transferência de malha fechada é um filtro passa-baixa, estável. Para termos erro de regime nulo à entrada degrau, conforme (10.6) e o teorema do valor final, devemos ter

$$\lim_{s \rightarrow 0} G_c(s)\hat{H}(s) = 1, \quad (10.11)$$

de modo que  $G_c(s)$  é dado por

$$G_c(s) = \frac{1}{KK_t}. \quad (10.12)$$

Note que  $G_c(s)$  assim especificada é própria e estável.

### 10.5.2 Um controlador de melhor desempenho

Uma limitação do compensador dado por (10.12) é que a função de transferência de malha fechada tem a dinâmica dada aproximadamente por (10.10). Temos um atraso de transporte igual ao da planta em malha aberta (o que é inevitável), erro de regime ao degrau nulo (o que é bastante desejável), mas a mesma constante de tempo  $T$  da malha aberta (o que pode estar aquém do desejado).

Uma alternativa relativamente simples para melhorar o desempenho, é fazer com que  $G_c(s)$  cancele o polo da planta, substituindo-o por outro mais rápido, ou seja

$$G_c(s) = \frac{1}{KK_t} \cdot \frac{T_s + 1}{T_c s + 1}. \quad (10.13)$$

Com isso,

$$G_f(s) = \frac{e^{-\theta s}}{T_c s + 1}. \quad (10.14)$$

Escolhendo-se adequadamente  $T_c$  é possível impor a dinâmica da função de transferência de malha fechada. Tipicamente desejariamos que o sistema em malha fechada respondesse mais rapidamente do que em malha aberta, e escolheríamos  $T_c < T$ .

Por fim, cabem algumas advertências sobre esta estratégia de controle. A técnica tende a produzir compensadores que invertem a planta e visam o cancelamento indiscriminado de polos e zeros, o que é uma estratégia não recomendável caso estes se situem no semi-plano direito ou possuam caráter muito oscilatório. Outra característica é que requisitos não realistas para o desempenho do sistema em malha fechada levam a ganhos excessivos do controle ou bandas muito largas para o sistema em malha fechada, características extremamente indesejáveis do ponto de vista prático. Cabe ao projetista identificar e evitar tais situações.

## 10.6 Atividades

### 10.6.1 Atividades prévias

- Esboce o diagrama de Bode em malha aberta do modelo linear do servomecanismo com atraso (por exemplo de 0,3s) em série com um compensador do tipo PI (por exemplo, como na Experiência 8). Verifique que devido ao atraso as margens de ganho e de fase do sistema são reduzidas.  
DICA: Veja no Matlab os comandos `margin`, `bode` e a propriedade `ioDelay` dos objetos de sistema.

### 10.6.2 Atividades em sala

- No Matlab, abra o diagrama `delay_template`.



- c) Identifique a duração do atraso de transporte do servomecanismo com atraso.
- d) Monte um compensador PI para controlar o servomecanismo com atraso conforme a Experiência 8. Dentre as sintonias utilizadas naquela experiência, use aquela que apresentou melhor desempenho.
- e) Reduza o ganho do compensador, caso necessário, até que o nível de oscilação seja aceitável.
- f) Implemente um controlador de modelo interno como sugerido na Seção 10.5. Tenha como objetivo obter um sistema em malha fechada com uma constante de tempo equivalente a 65% da constante de tempo da malha aberta.
- g) Compare as respostas obtidas e avalie o desempenho do IMC.
- h) Compare também com as simulações dos modelos linear e não-linear.

### 10.6.3 Relatório

Esta experiência deve ser incluída no relatório “Controle de Velocidade”.

## 10.7 Bibliografia sugerida

(Mor89) Morari, M.; Zafiriou, E. **Robust Process Control**. Prentice-Hall, 1998.

*O capítulo 3 trata de controle IMC num caso similar ao do laboratório.  
Este livro está disponível on-line em  
<http://books.google.com.br/books?isbn=0137821530>*

(Lev11) Levine, W. (org.). **The Control Handbook**. CRC Press, 2nd. ed., 2011.

*Veja a Seção 32.24.3.*