

MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DESENVOLVIMENTO DE UMA ARQUITETURA PROGRAMÁVEL DE CONTROLE EM  
TEMPO REAL PARA UM SERVOPOSICIONADOR PNEUMÁTICO

por

Guilherme de Oliveira Kunz

Dissertação para obtenção do Título de  
Mestre em Engenharia

Porto Alegre, março de 2006

DESENVOLVIMENTO DE UMA ARQUITETURA PROGRAMÁVEL DE CONTROLE EM  
TEMPO REAL PARA UM SERVOPOSICIONADOR PNEUMÁTICO

por

Guilherme de Oliveira Kunz  
Engenheiro Mecânico

Dissertação submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Mestre em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. Eduardo André Perondi

Comissão de Avaliação:

Prof. Dr. Flávio José Lorini

Prof. Dr. José Antônio Esmério Mazzaferro

Prof. Dr. Luciano Antônio Mendes

Prof. Dr. Flávio José Lorini  
Coordenador do PROMEC

Porto Alegre, 23 de março de 2006

## RESUMO

Este trabalho aborda o desenvolvimento de uma arquitetura de controle em tempo real para servoposicionadores pneumáticos, baseada em computadores pessoais (PCs).

Os servoposicionadores pneumáticos são de baixo custo, leves, não poluentes e de fácil utilização. Como apresentam boa relação entre peso e força, são bastante atraentes em aplicações de robótica. Entretanto, devido a suas não linearidades, os servoposicionadores pneumáticos apresentam dificuldades em seu controle. Visando compensá-las, são desenvolvidos algoritmos de controle cada vez mais complexos, necessitando de ferramentas mais robustas quanto ao poder de processamento.

Ferramentas com características necessárias para o desenvolvimento de algoritmos e para o controle em tempo real de sistemas custam caro, o que dificulta o desenvolvimento de novas tecnologias de controle de servoposicionadores pneumáticos.

Este trabalho apresenta uma revisão das soluções utilizadas na construção de sistemas pneumáticos de posicionamento e daquelas adotadas no controle digital de sistemas automáticos. Descreve-se o processo de construção de uma bancada experimental, e o desenvolvimento das soluções em *hardware* e *software* para o controle digital é discutido. Visando uma solução economicamente atraente, são utilizados unicamente *softwares* de código aberto e de livre utilização, assim como *hardwares* de baixo custo.

Para verificar a eficiência da solução proposta, a arquitetura de controle é utilizada para realizar a identificação dos parâmetros do sistema pneumático. Dentre eles, destacam-se a vazão mássica e o atrito, informações importantes para simulação e controle do sistema. Também são utilizados controladores do tipo Proporcional-Integral-Derivativo, implementados para apoiar o estudo do desempenho da arquitetura no controle do servoposicionador pneumático.

## ABSTRACT

“Development of a programmable realtime control architecture for a pneumatic servopositioning system“

This work proposes the development of a PC-based, realtime control architecture for a pneumatic servopositioning system.

Pneumatic servopositioning systems are cheap, lightweight, non-pollutant and easy to use. As they have good weight-to-force ratio, they are attractive to robotics applications. However, due to their nonlinear characteristics, servopneumatic drivers are difficult to be controlled. In order to overcome these limitations, complex control algorithms are being developed, requiring more robust tools with respect to processing capabilities.

The computational tools with the necessary characteristics for the development of the algorithms and of the realtime control of these systems are rather expensive, restricting the development of new control technologies for pneumatic servopositioning systems.

This work presents a review of the solutions used on the construction of pneumatic positioning systems and of those adopted on the digital control of automatic systems. The process of building an experimental bench is described, and the development of solutions in both hardware and software for digital control is discussed. Aiming the development of an economically attractive solution, only open source softwares are used, as well as low cost hardware.

To verify the efficiency of the proposed solution, the control architecture is used to identify the pneumatic servopositioning system parameters. Such parameters include mass airflow rate and friction force, which are important for systems' simulation and control. Position-Integral-Derivative controllers were implemented to support the performance analysis of the software and hardware architecture in the control of the pneumatic servopositioning system.

## SUMÁRIO

<b>Lista de Símbolos</b>	<b>vi</b>
<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação	1
1.2 O Atuador Pneumático	2
1.3 O Servoposicionador Pneumático	3
1.4 O Controle Digital	4
1.5 Objetivos	5
1.6 Organização Geral	6
<b>2 Revisão Bibliográfica</b>	<b>7</b>
2.1 Soluções para Servoposicionadores	7
2.1.1 Servoválvula	7
2.1.2 Servoatuador	8
2.1.3 Sensores de Posição e Pressão	9
2.2 Soluções para o Controle Digital	11
2.2.1 Controlador Dedicado	11
2.2.2 Controladores Distribuídos	13
2.2.3 Controladores Orientados à Objetos	13
2.2.4 Microprocessadores e Microcontroladores	14
2.3 Técnicas de Controle	15
2.3.1 Controladores PID	16
2.3.2 Controladores por realimentação de estados	16
2.3.3 Linearização por Realimentação	17

2.3.4	Controle a Estrutura Variável . . . . .	18
2.3.5	Controles baseados na interpretação de divisão em dois subsistemas . . . . .	20
2.3.6	Controladores adaptativos . . . . .	21
2.3.7	Algoritmos Genéticos . . . . .	22
2.3.8	Controle por lógica difusa . . . . .	22
2.3.9	Controle por redes neurais . . . . .	23
<b>3</b>	<b>Implementação Experimental . . . . .</b>	<b>25</b>
3.1	Introdução . . . . .	25
3.2	Descrição Geral do Sistema . . . . .	25
3.3	Componentes . . . . .	27
3.3.1	Componentes do Subsistema de Tratamento do Fluido . . . . .	27
3.3.1.1	Filtro e Regulagem de Pressão . . . . .	27
3.3.1.2	Vaso de Pressão . . . . .	28
3.3.2	Componentes do Atuador Pneumático . . . . .	28
3.3.2.1	Servoválvula Direcional . . . . .	28
3.3.2.2	Cilindro Atuador . . . . .	29
3.3.3	Transdutores . . . . .	30
3.3.3.1	Fonte . . . . .	30
3.3.3.2	Encoder Incremental . . . . .	31
3.3.3.3	Sensores de Pressão . . . . .	32
3.3.4	Sistema de Aquisição de Dados . . . . .	32
3.3.4.1	Driver do <i>Encoder</i> . . . . .	33
3.3.4.2	Conversão AD/DA . . . . .	37
3.3.4.3	Condicionamento de Sinais . . . . .	38
3.3.4.4	Microcomputador . . . . .	39
<b>4</b>	<b>Sistema de Controle e Aquisição de Dados . . . . .</b>	<b>40</b>

4.1	Funcionalidades do Sistema de Controle . . . . .	40
4.2	Sistema de Aquisição e Acionamento . . . . .	41
4.2.1	Sistemas de Tempo Real . . . . .	41
4.2.2	Sistema Operacional . . . . .	44
4.2.3	Algoritmos de Controle dos Dispositivos . . . . .	46
4.3	Ambiente de desenvolvimento e monitoramento . . . . .	49
4.3.1	Scilab/Scicos . . . . .	49
<b>5</b>	<b>Determinação dos Parâmetros do Sistema . . . . .</b>	<b>52</b>
5.1	Caracterização do Sistema . . . . .	52
5.1.1	Vazão Mássica de Fluidos Compressíveis Através de Orifícios . . . . .	53
5.1.2	Identificação das Curvas de Vazão Mássica . . . . .	54
5.1.3	Identificação do Atrito . . . . .	56
5.1.4	Temporização . . . . .	62
<b>6</b>	<b>Considerações Finais e Conclusões . . . . .</b>	<b>64</b>
6.1	Trabalhos Futuros . . . . .	65
	<b>Referências . . . . .</b>	<b>67</b>
	<b>Apêndice A – Anexos . . . . .</b>	<b>73</b>
A.1	Código do <i>Decoder</i> . . . . .	73
A.2	Código de controle dos dispositivos do servoposicionador . . . . .	75
A.3	Bloco do Scilab . . . . .	80

## LISTA DE SÍMBOLOS

$A$	área útil do êmbolo do pistão [ $m^2$ ]
$A_o$	área transversal de passagem do gás em um orifício [ $m^2$ ]
$b$	razão de pressões crítica (ISO 6358)
$C$	coeficiente de vazão da válvula (condutância) (ISO 6358) [ $m^5/Ns$ ]
$C_f$	coeficiente de atrito viscoso equivalente [ $N/(m/s)$ ]
$c_p$	calor específico do ar a pressão constante [ $J/K_gK$ ]
$c_v$	calor específico do ar a volume constante [ $J/K_gK$ ]
$e$	erro
$e(t)$	função erro
$f(.)$	função dependente do estado
$F$	força [N]
$F_a$	força de atrito [N]
$F_c$	força de atrito seco (ou de Coulomb) [N]
$F_e$	força externa [N]
$F_s$	força de atrito estático (força de quebra) [N]
$f_u(.)$	função de tensão
$f_p(.)$	função de pressão
$f_p(.,.)$	função das pressões
$F$	vetor força
$g$	função vetorial contínua
$K$	ganho constante, coeficiente de calibração
$K_a$	ganho de aceleração [ $V_s^2/m$ ]
$K_d$	ganho constante
$K_p$	ganho proporcional
$K_{sup}$	coeficiente linear de calibração da pressão de suprimento [ $P_a/V$ ]
$K_v$	ganho de velocidade
$L$	curso útil do cilindro atuador [m]
$M$	massa [ $K_g$ ]



$p$	pressão absoluta [ $P_a$ ]
$p_{atm}$	pressão atmosférica [ $P_a$ ]
$p_{crit}$	relação de pressões crítica
$p_{\Delta}$	diferença de pressão entre dois pontos do circuito [ $P_a$ ]
$p_d$	pressão a jusante [ $P_a$ ]
$P$	constante positiva
$p_s$	pressão de suprimento [ $P_a$ ]
$p_{sup}$	pressão de suprimento [ $P_a$ ]
$p_u$	pressão a montante [ $P_a$ ]
$P_{\Delta}$	diferença de pressão [ $P_a$ ]
$q_m$	vazão mássica [ $K_g/s$ ]
$R$	constante do gás [ $K_gJ/K$ ]
$r$	razão de calores específicos do ar
$S(.)$	superfície de deslizamento
$T$	temperatura [K]
$t$	tempo [s]
$u$	sinal de controle [V]
$V$	volume [ $m^3$ ]
$V_{i0}$	( $i=1,2$ ) volume morto na câmara incluindo as tubulações [ $m^3$ ]
$\dot{V}$	vazão volumétrica [ $m^3/s$ ]
$v_s$	velocidade de limiar ou de Stribeck [m/s]
$x_v$	deslocamento do carretel controlador da servoválvula [m]
$y(t)$	deslocamento do êmbolo do pistão [m]
$y_d$	posição final desejada [m]
$y_d(t)$	trajetória desejada [m]
$\dot{y}_r$	velocidade de referência [m/s]
$\beta$	módulo de elasticidade volumétrica [ $N/m^2$ ]
$\rho$	massa específica [ $K_g/m_3$ ]

$w$	freqüência [rad/s]
$w_n$	freqüência natural [rad/s]
$\Delta$	variação
$(\dot{\quad})$	derivada primeira
$(\ddot{\quad})$	derivada segunda
$(\quad)_{atm}$	sub-índice de referência a pressão atmosférica
$(\quad)_{crit}$	sub-índice indicador de condição crítica ou de saturação
$(\quad)_d$	sub-índice indicador de condição desejada
$(\quad)_{ench}$	superíndice indicador de condições de enchimento da câmara
$(\quad)_{exau}$	superíndice indicador de condições de enchimento da câmara
$(\quad)_i$	sub-índice indicador de condições iniciais ou de elemento individual
$(\quad)_{max}$	sub-índice indicador de condição de valor máximo
$(\quad)_{min}$	sub-índice indicador de condição de valor mínimo
$(\quad)_{(n)}$	superíndice indicador de enésima derivada
$(\quad)_p$	sub-índice indicador de pressão
$(\quad)_r$	sub-índice indicador de referência
$(\quad)_s$	sub-índice indicador de condição de pressão de suprimento
$(\quad)_{sup}$	sub-índice indicador de condição de pressão de suprimento
$(\quad)_u$	sub-índice indicador de tensão
$(\quad)_{\Delta}$	sub-índice indicador de diferença

## LISTA DE FIGURAS

1.1	Equilíbrio de forças no êmbolo do cilindro atuador (Perondi, 2002) . . . . .	3
1.2	Servoposicionador pneumático . . . . .	4
1.3	Controle Digital de Processos . . . . .	5
1.4	Tendências de desenvolvimento de Software <i>versus</i> Hardware . . . . .	5
2.1	Cilindro Pneumático sem Haste (Festo Corporation, 2005) . . . . .	8
2.2	Aplicação de um cilindros pneumáticos sem haste (Festo Corporation, 2005) . . . . .	9
2.3	Controle dedicado . . . . .	12
2.4	Controle Distribuído . . . . .	13
2.5	Esquema simplificado para controle de posição PID . . . . .	16
2.6	Controle por realimentação de estados (PVA) . . . . .	17
2.7	Linearização por realimentação de estados . . . . .	18
2.8	Fenômeno de <i>Chattering</i> . . . . .	19
2.9	Controlador por modos deslizantes aplicado a servoposicionadores . . . . .	20
2.10	Interpretação do sistema de posicionamento . . . . .	20
2.11	Controlador auto-adaptativo . . . . .	21
2.12	Algoritmo genético para otimização dos parâmetros do controlador PID . . . . .	22
2.13	Controlador por lógica difusa . . . . .	23
2.14	Controlador MNN com realimentação proporcional de posição . . . . .	23
2.15	Controle por redes neurais . . . . .	24
3.1	Bancada de testes no Laboratório de Robótica da UFRGS. . . . .	25
3.2	Esquema da bancada experimental . . . . .	26
3.3	Unidade de tratamento e regulagem de pressão . . . . .	27
3.4	Servoválvula controladora de vazão . . . . .	29
3.5	Representação de forças no cilindro atuador (Festo Corporation, 2005) . . . . .	30
3.6	Encoder incremental . . . . .	31
3.7	Conexão do carro móvel do êmbolo do cilindro pneumático com a correia dentada .	32

3.8	Sinais do <i>encoder</i> incremental . . . . .	33
3.9	Condicionador de sinais para o encoder . . . . .	34
3.10	Esquema do circuito do microcontrolador PIC18F2550 . . . . .	35
3.11	Esquema do processo de contagem de pulsos (“Contador”) . . . . .	35
3.12	Comando “zerar” . . . . .	36
3.13	Diagrama esquemático do comando “ler” . . . . .	36
3.14	Esquema do processo de comunicação com a unidade de controle . . . . .	37
3.15	Condicionamento de sinais para os sensores de pressão . . . . .	38
3.16	Condicionamento de sinal do controle da servoválvula . . . . .	38
4.1	Diagrama de funcionalidades do sistema de controle . . . . .	41
4.2	Informações temporais sobre processos . . . . .	42
4.3	Sistemas determinísticos <i>versus</i> probabilísticos . . . . .	43
4.4	Visão geral do sistema . . . . .	46
4.5	Sistema de controle e acionamento . . . . .	48
4.6	Exemplo de implementação de programa de controle em ambiente Scilab/Scicos. . . . .	50
4.7	Sistema de desenvolvimento . . . . .	50
4.8	Sistema de Monitoramento . . . . .	51
5.1	Estrangulamento na passagem do fluido . . . . .	53
5.2	Vazão Mássica Teórica . . . . .	54
5.3	Curvas experimentais pressão x tempo x tensão . . . . .	57
5.4	Curvas experimentais vazão x tempo x tensão . . . . .	58
5.5	Curvas experimentais vazão x pressão x tensão . . . . .	59
5.6	Curvas experimentais estáticas de vazão <i>versus</i> pressão . . . . .	60
5.7	Curvas experimentais estáticas de vazão <i>versus</i> tensão . . . . .	60
5.8	Curva estática de força de atrito pela velocidade constante (Perondi, 2002) . . . . .	61
5.9	Resposta do controlador PID para velocidade de 0.1m/s . . . . .	62
5.10	Mapa estático de atrito . . . . .	63

5.11	Períodos dos ciclos de aquisição e acionamento . . . . .	63
A.1	Bloco de representação do servoposicionador no sistema Scicos/Scilab . . . . .	80

## LISTA DE TABELAS

3.1	Válvula reguladora e filtro . . . . .	28
3.2	Vaso de Pressão . . . . .	28
3.3	Válvula direcional . . . . .	29
3.4	Silenciadores . . . . .	29
3.5	Servoatuador . . . . .	30
3.6	Fonte chaveada . . . . .	31
3.7	Encoder incremental . . . . .	31
3.8	Precisão do sistema . . . . .	32
3.9	Sensores de Pressão . . . . .	32
3.10	Informações técnicas do PIC18F2550 . . . . .	34
3.11	Placa de aquisição de dados . . . . .	37
3.12	Informações técnicas do microcomputador . . . . .	39
5.1	Dados da bancada . . . . .	56
5.2	Atrito para $\dot{y} > 0$ . . . . .	62
5.3	Atrito para $\dot{y} < 0$ . . . . .	62
5.4	Tempo dos ciclos . . . . .	63

# 1 INTRODUÇÃO

Neste capítulo estão descritos os principais aspectos referentes à motivação para a realização do presente trabalho. Esta descrição engloba também uma breve introdução sobre a utilização de servoatuadores e servoposicionadores pneumáticos na indústria assim como as características necessárias aos sistemas para realizarem o seu controle. Além disso, neste capítulo são apresentados também os objetivos e a organização geral da dissertação.

## 1.1 Motivação

O uso da pneumática em sistemas de posicionamento é comum em diversos sistemas automáticos tais como: acionamentos de robôs industriais (Andrighetto et al., 2005a), manipulação e movimentação de material, máquinas industriais alimentícias, linhas de montagem e pequenas máquinas e sistemas automáticos utilizados em diferentes processos da manufatura (Virvalo, 1995).

Trabalhos realizados em indústrias manufatureiras indicam que aproximadamente 70% das aplicações de manipulação de peças são realizadas sobre massas que variam de 1 a 10kg, sendo requeridas precisões de posicionamento que variam de 0,2mm a 1mm. Isto se enquadra dentro do campo de aplicação da servopneumática (Latino e Sandoval, 1996).

Como vantagens da utilização de sistemas pneumáticos observa-se o baixo custo quando comparados com outros sistemas tecnológicos, boa relação peso/potência, facilidade de manutenção, rápida atuação em altas acelerações, flexibilidade de instalação e utilização de fonte de energia limpa e disponível em grande parte das plantas industriais: o ar (Cruz, 2003).

Em contraste com as vantagens descritas acima, encontra-se presente em sistemas pneumáticos um comportamento oscilatório proveniente do atrito nos atuadores e da compressibilidade do ar (Andrighetto et al., 2005a). Este comportamento é crítico principalmente em baixas velocidades de deslocamento (Guoliang e Xuanyin, 2003). O emprego da pneumática é portanto mais comum para situações de posicionamento ponto-a-ponto.

A servopneumática consiste de uma técnica de acionamento relativamente barata, onde o componente mais caro é a servoválvula. Devido a problemas oriundos da compressibilidade do ar e do atrito, a maior dificuldade de desenvolver aplicações precisas utilizando esta tecnologia está no seu controle. Sistemas pneumáticos geralmente necessitam de bons modelos matemáticos que representem as não linearidades inerentes ao sistema (Bobrow e McDonell, 1998). Pesquisas sobre a compensação do atrito em atuadores pneumáticos já vem sendo desenvolvidas, como em Perondi (2002) e Valdiero (2005).

A evolução tecnológica, principalmente na eletrônica e no ramo de *software*, tem proporcionado recursos computacionais para o desenvolvimento de algoritmos mais complexos, sendo uma tendência do mercado de investir mais em soluções em *software* frente ao *hardware* (Sun Microsystems, 2004).

A solução proposta por grande parte dos pesquisadores na área (Perondi, 2002; Rajendran e Bolton, 2003) para superar as dificuldades encontradas no emprego da servopneumática está no desenvolvimento de algoritmos de controle. Estes algoritmos, que estão a cada dia mais sofisticados e complexos, necessitam de sistemas de controle digital com maior flexibilidade e poder de processamento matemático. Trabalhos anteriores, como Vieira (1998), demonstram a necessidade da utilização de frequências de controle maiores que 1KHz devido à rápida resposta dos sistemas pneumáticos.

Soluções comerciais utilizadas para o sistema de controle de posicionamento servopneumático (Perondi, 2002; Vieira, 1998; Andrighetto et al., 2005b) chegam a custar mais de 30 mil reais (veja dSPACE GmbH (2005)). Deste modo, o alto custo compromete a expansão das aplicações de servopneumática na indústria em problemas de precisão.

Este trabalho tem como principal motivação o desafio de desenvolver um servoposicionador pneumático de baixo custo e preciso. Para tanto, propõe-se estruturar uma bancada para um servoposicionador pneumático e desenvolver uma arquitetura de controle econômica e flexível para a mesma. Esta arquitetura deve permitir a implementação de algoritmos de controle que forneçam um posicionamento preciso e confiável. Com isto, esta bancada deverá também colaborar com o desenvolvimento de tecnologias que permitam a disseminação de servoposicionadores na indústria brasileira através da diminuição de custos na implementação.

Para tanto, foi feito um levantamento bibliográfico sobre as soluções já utilizadas na implementação de servoposicionadores e nas possibilidades de controle. Foram analisados os requisitos necessários para manter a flexibilidade no desenvolvimento de algoritmos de controle, garantias sobre os tempos do processo e, devido a restrições orçamentárias, o baixo custo da solução.

## **1.2 O Atuador Pneumático**

Os sistemas que possibilitam posicionar uma carga mecânica a partir de um conjunto de coordenadas ( $x$ ,  $y$ ,  $z$ ) são chamados de sistemas de posicionamento. Esta localização pode ser fixa ou variável no tempo e, nestes casos, os sistemas de posicionamento são, algumas vezes, chamados de seguidores.

Um componente importante do sistema é o elemento que aplica a força sobre a carga mecânica para levá-la até a posição desejada. Este elemento é chamado de motor ou atuador podendo ser



elétrico, hidráulico ou pneumático (Cruz, 2003).

Atuadores pneumáticos têm como vantagem sobre acionamentos elétricos, a ausência de motores pesados, geralmente com sistemas complexos de transmissão por engrenagens (Liu e Bobrow, 1988). O acionamento elétrico também é relativamente caro devido ao alto custo dos servomotores. Em aplicações de robótica móvel, a ausência de engrenagens rígidas, como nas soluções nas quais utiliza-se *músculos pneumáticos*, são atrativos para o desenvolvimento de robôs flexíveis e com tamanho reduzidos (Granosik e Borenstein, 1998).

As equações de equilíbrio dinâmico de um cilindro pneumático, como mostrado no modelo da Fig. 1.1, descrevem o funcionamento básico do sistema.

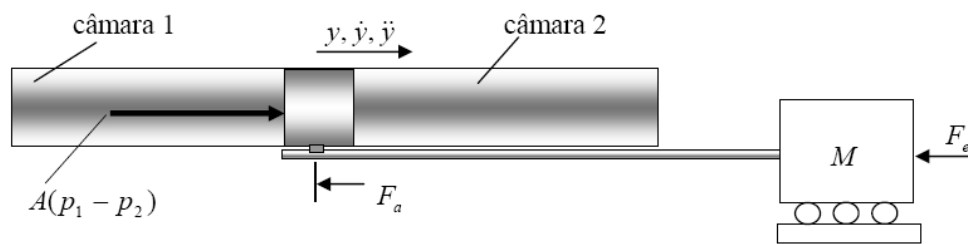


Figura 1.1 – Equilíbrio de forças no êmbolo do cilindro atuador (Perondi, 2002)

Este equilíbrio de forças é obtido pela aplicação da 2ª lei de Newton:

$$M\ddot{y} + F_a + F_e = A(p_1 - p_2) \quad (1.1)$$

onde  $M$  é a massa deslocada,  $F_a$  é a força de atrito,  $F_e$  é a força externa,  $p_1$  é a pressão na câmara 1,  $p_2$  é a pressão na câmara 2,  $A$  é a área do êmbolo e  $A(p_1 - p_2)$  é a força resultante da diferença entre as câmaras.

Para controlar a variação destas pressões assim como a posição, velocidade e aceleração do êmbolo, utiliza-se uma ou mais servoválvulas. Este sistema é chamado de servoatuador pneumático. Quando o sistema é realimentado com a posição (permitindo o controle da posição) ele é denominado servoposicionador pneumático.

### 1.3 O Servoposicionador Pneumático

Servoposicionadores pneumáticos são compostos basicamente por uma servoválvula que controla a vazão de ar na direção de um cilindro cujo êmbolo está ligado à massa que se deseja posicionar. Os principais elementos de um servoposicionador pneumático estão esquematizados na Fig. 1.2.

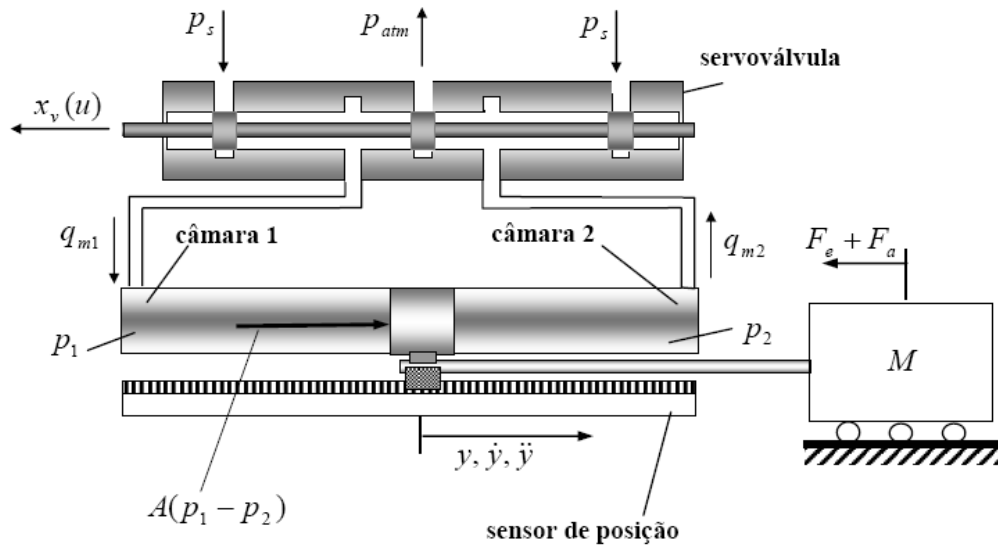


Figura 1.2 – Servoposicionador pneumático

A válvula direcional mostrada na Fig. 1.2 possui o sistema de controle da direção da vazão baseado em carretel deslizante. Segundo Pinches e Callear (1996) este é o tipo mais comum de válvula utilizada em servoposicionadores pneumáticos para a transmissão de potência para o atuador. O deslocamento do carretel da válvula  $x_v$  é muitas vezes provocado pela tensão ( $u$ ) aplicada ao solenóide. Assim, de acordo com a tensão aplicada, ocorre o deslocamento  $x_v(u)$  do carretel da servoválvula, estabelecendo a diferença de pressão no êmbolo do cilindro.

#### 1.4 O Controle Digital

Para o controle preciso de seguimento de trajetórias é necessário utilizar tecnologias que permitam a aquisição de dados (posição e, muitas vezes, também as pressões nas câmaras), seu tratamento por algoritmos de controle e a posterior atuação no sistema físico (sinal para servoválvula direcional). Este processo é chamado de controle digital de processos.

A utilização de sistemas de controle avançados possibilita, por exemplo, a compensação do atrito, aumentando a precisão no uso da pneumática para fins de posicionamento preciso na indústria.

O controle digital, é atualmente, na maioria dos casos, baseado em microprocessadores ou microcontroladores, que executam processos continuamente (Kilian, 2000). O controle de processo consiste na leitura da trajetória requerida e dos valores dos sensores e na utilização destes dados para calcular a saída enviada para o atuador, como mostra a Fig. 1.3.

Quanto menor o período de tempo necessário para a repetição deste processo, maior a possibilidade de precisão no posicionamento. Esta é a diferença fundamental quanto ao processo

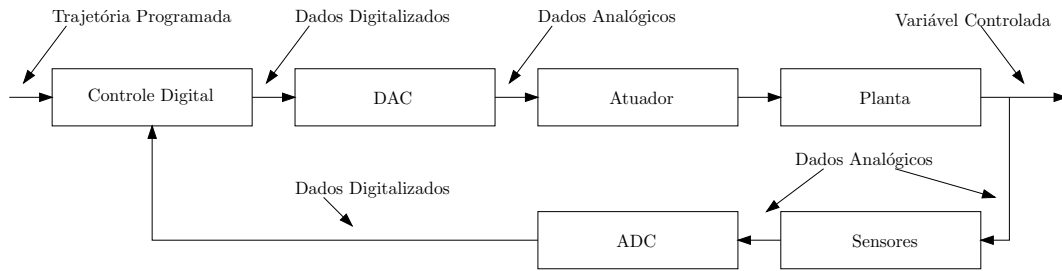


Figura 1.3 – Controle Digital de Processos

analógico que é contínuo e possui resposta muito mais rápida do que a grande maioria dos sistemas físicos controlados. Também, nos processos digitais, busca-se um tempo entre cada interação que seja muito pequeno quando comparado com o tempo de resposta do sistema físico controlado (Kilian, 2000).

Como já enfatizado, o avanço da tecnologia atual direciona-se à utilização de algoritmos de controle complexos do que ao desenvolvimento de *hardware* específico. Isto torna o sistema flexível e diminui os custos de atualização ou manutenção, conforme ilustrado na Fig. 1.4 (Sun Microsystems, 2004).

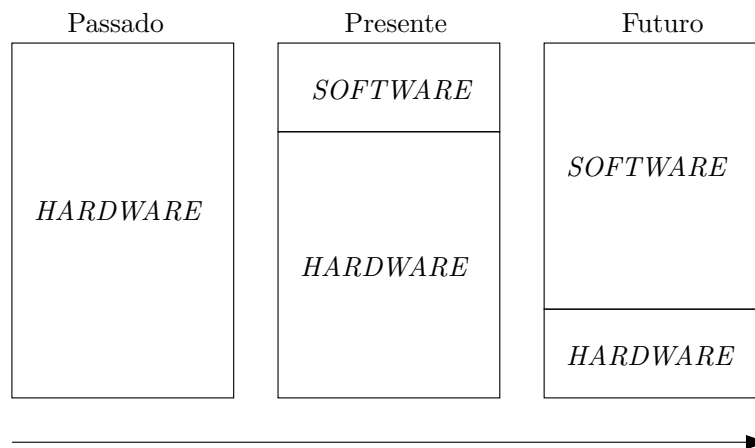


Figura 1.4 – Tendências de desenvolvimento de Software *versus* Hardware

## 1.5 Objetivos

O objetivo geral é o de desenvolver, a partir do uso de ferramentas computacionais gratuitas (*Software Livre*), um sistema de controle em tempo real a baixo custo, que permita controlar um servoposicionador pneumático.

Os principais objetivos específicos deste trabalho são os seguintes:

- estruturar uma bancada experimental com um servoposicionador pneumático.

- instrumentar a bancada com sensores de pressão nas câmaras e de posição do êmbolo do cilindro atuador.
- desenvolver um sistema de leitura de medidas de sensores e de controle para servoposicionadores pneumáticos.
- analisar e validar ferramentas para realização do desenvolvimento de algoritmos de controle.
- testar a solução de controle e a implementação experimental com a utilização de um algoritmo de controle linear.
- identificar dos parâmetros da bancada.

## **1.6 Organização Geral**

No Capítulo 2 apresenta-se uma revisão bibliográfica sobre as soluções já desenvolvidas para o sensoriamento e controle de servoposicionadores pneumáticos. No Capítulo 3 é feita uma descrição do sistema físico do servoposicionador e seu funcionamento. Neste capítulo também são tratados os procedimentos de montagem da bancada. No Capítulo 4 é feita uma análise sobre as soluções utilizadas para o controle da bancada. Neste mesmo capítulo são analisados e descritos os algoritmos utilizados e funcionalidades fornecidas pelo sistema. No Capítulo 5 são apresentados os resultados dos testes realizados para verificar o desempenho tanto do sistema de controle quanto do servoposicionador. No Capítulo 6 são apresentadas as conclusões do presente trabalho e sugestões para trabalhos futuros.

## **2 REVISÃO BIBLIOGRÁFICA**

Neste capítulo são descritas as soluções encontradas na bibliografia para a implantação de servoposicionadores pneumáticos. São abordados os meios encontrados para realizar o controle digital, incluindo o monitoramento da posição e pressão nas câmaras do cilindro e o acionamento da servoválvula. São também descritas algumas das técnicas de controle mais utilizadas para o controle de servoposicionadores.

### **2.1 Soluções para Servoposicionadores**

Em processos de fabricação industrial é comum encontrar aplicações onde são necessários deslocamentos de peças em linhas de montagem ou entre estações de trabalho. Em alguns casos, este deslocamento é realizado por robôs, para que trajetórias mais complexas possam ser executadas, como nas aplicações de soldagem robotizada.

A utilização da pneumática tem sido adotada para aplicações industriais que necessitam de controle preciso de posição e velocidade, principalmente quando custos reduzidos e ambiente limpo são requisitos básicos para a solução. Porém, é mais comum utilizar atuadores hidráulicos ou elétricos devido à dificuldade de trabalhar com os atuadores pneumáticos, pois estes possuem não-linearidades que dificultam seu controle, conforme discutido anteriormente.

Para aplicações de pequeno e médio porte (até 1kW), dispositivos pneumáticos possuem melhor relação custo-benefício quando comparados a dispositivos elétricos e hidráulicos (Harrison et al., 1987). Visando a diminuição de custos, trabalhos vêm sendo desenvolvidos para aumentar a eficiência dos sistemas pneumáticos, principalmente através da diminuição do desperdício de ar comprimido. Este ar, após passar pelo circuito pneumático, geralmente é jogado para atmosfera. A reutilização do ar comprimido é pauta de trabalhos como Bachmann e Surgenor (1997) e Arinaga et al. (2000) que visam o reaproveitamento do ar de exaustão para realimentar a planta.

Nas próximas seções são abordadas as diferentes possibilidades de implementação de cada componente de um sistema de servoposicionamento pneumático.

#### **2.1.1 Servoválvula**

Existem duas configurações geralmente utilizadas para as servoválvulas nos servoposicionadores pneumáticos. A primeira e menos utilizada consiste na disposição de uma servoválvula reguladora para cada câmara do atuador. Isto implica na possibilidade de existência de um sinal de controle para cada servoválvula, como encontrado em Cruz (2003) e Lin et al. (1996), adicio-

nando assim maior complexidade ao sistema de controle. A segunda configuração é mais comum e consiste no uso de uma servoválvula direcional que permite o controle da direção da vazão diretamente por um único sinal de controle como encontrado em Vieira (1998), Perondi (2002), Pu et al. (1997), Virvalo (1995), dentre outros. Segundo as especificações técnicas de servoválvulas direcionais, esta é a mais indicada para casos de controle de posição (Festo Corporation, 2005).

### 2.1.2 Servoatuador

O servoatuador pneumático é o responsável por converter a diferença de pressão do ar em movimento mecânico. Conforme Dorf (1989) existem dois tipos básicos de servoatuadores pneumáticos: atuadores lineares (pistão/cilindro) e rotacionais.

Atuadores rotacionais convertem a diferença de pressão do ar em movimento rotacional. Estes tipos de atuadores são utilizados, por exemplo, em parafusadeiras, podendo operar em diferentes velocidades (Renn e Liao, 2004).

Atuadores lineares pneumáticos possuem grande variedade de forma, tamanho e capacidade de carga. Quanto ao tipo de acionamento, existem dois tipos principais de configuração de atuadores lineares: cilindros de dupla ação e cilindros de ação unidirecional. Os cilindros de dupla ação são conectados a uma válvula direcional por meio de duas tubulações, uma de cada lado, que comandam a diferença de pressão no êmbolo, movimentando-o. Os cilindros de ação unidirecional, como os utilizando em Yin e Araki (1998), são conectados por uma única tubulação a válvula reguladora que controla a força unidirecional de atuação no êmbolo. A força contrária à atuação pneumática é realizada por um sistema de molas de retorno.

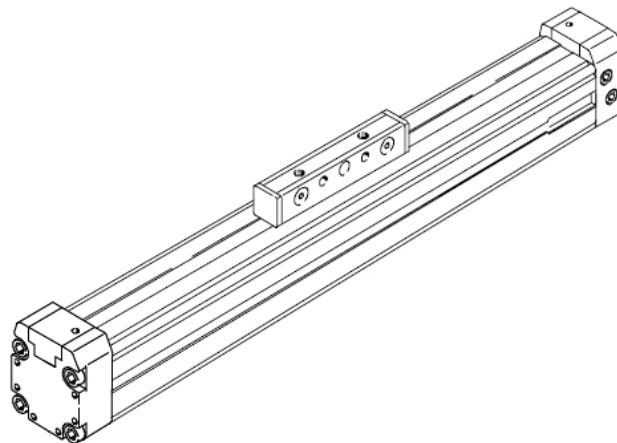


Figura 2.1 – Cilindro Pneumático sem Haste (Festo Corporation, 2005)

Além dos modos de ação do ar comprimido, os atuadores lineares podem ser fabricadas com haste ou sem haste. O cilindro pneumático sem haste possui como principal vantagem, compa-

rado a cilindros com haste, a redução de espaço necessário para instalação, permitindo aumento no curso de alcance do atuador. A partir das figuras Fig. 1.1 e Fig. 2.1 pode-se observar as características construtivas de ambas soluções e na Fig. 2.2 um exemplo de aplicação destes atuadores para construção de um robô cartesiano.

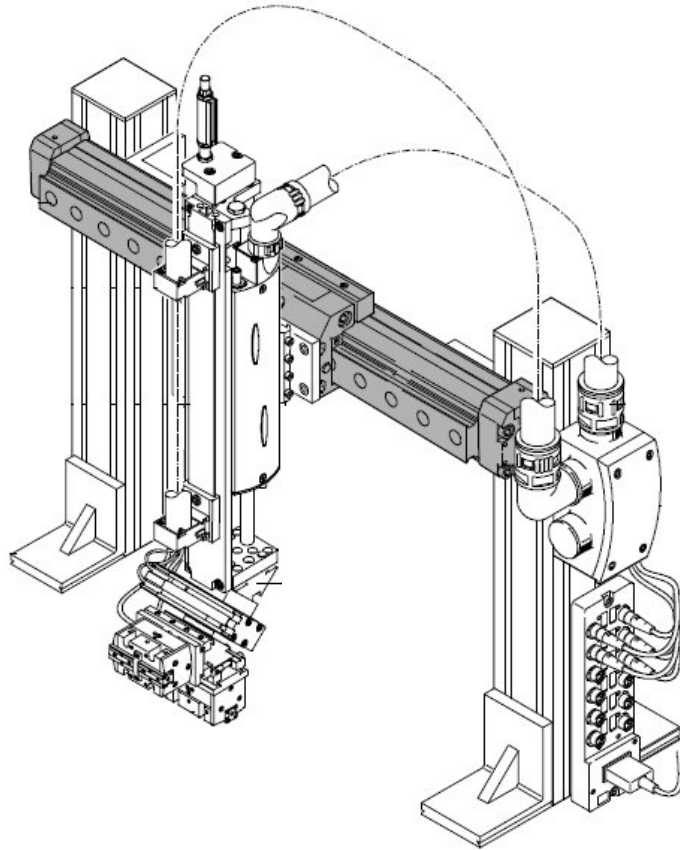


Figura 2.2 – Aplicação de um cilindros pneumáticos sem haste (Festo Corporation, 2005)

### 2.1.3 Sensores de Posição e Pressão

Para o controle da posição de um atuador pneumático é necessário monitorar em tempo real a posição do pistão para fechar o ciclo de controle, conforme mostrado na Fig. 1.3. Para isto empresas como a Festo já acoplam sensores de posição em algumas soluções de cilindros atuadores (Festo Corporation, 2005).

Os sensores de posicionamento geralmente utilizados são potenciômetros, *encoders* óticos e transdutores lineares absolutos (LVDT - *Linear Variable Differential Transformer*).

Os transdutores lineares absolutos (LVDT) utilizados, por exemplo, por Atmanand e Konnur (1999) e Richardson et al. (2001), possuem grande resolução e oferecem como vantagem a eliminação da necessidade de reposicionar o êmbolo na posição inicial durante cada inicialização do processo, assim como a eliminação de problemas característicos de sistemas mecânicos como folga em engrenagens e vibrações providas do contato entre os materiais.

Sensores potenciométricos normalmente possuem custo menor, podendo ser rotacionais ou lineares. Baseiam-se no princípio de conversão de diferença de resistência elétrica em tensão elétrica. Para aplicação de sensores potenciométricos há a necessidade de utilizar conversores AD como em Perondi (2002), Vieira (1998) e Andrighetto et al. (2005b).

A utilização de *encoders* óticos para aquisição da posição do atuador linear pode ser encontrado em trabalhos como Pu et al. (1992) e Lin et al. (1996). A maior das vantagens na utilização de *encoders* é que o sinal obtido já está no formato digital, evitando perdas de precisão providas do processo de conversão AD.

Quanto à forma de determinação da posição, existem dois tipos de *encoders* óticos: absolutos e relativos. As vantagens de utilizar *encoders* absolutos estão relacionados com a possibilidade de realizar as medições sem a necessidade de reposicionar o atuador em sua marca inicial a cada experimento. Os *encoders* absolutos fornecem o valor global de posição a cada instante, não ocorrendo erros acumulativos. Estes sensores, porém, não são aplicáveis para resoluções elevadas, além disso, suas características construtivas são mais complexas quando comparados a *encoders* incrementais. Geralmente, por isso, são mais caros.

*Encoders* incrementais possuem estrutura mais simples e necessitam da utilização de um *decoder*, ou seja, um sistema contador de pulsos que tem a função de definir a posição do atuador. O sistema de contagem de pulsos deve ter desempenho adequado à velocidade do acionamento, pois poderá “perder” pontos caso não consiga realizar a contagem na velocidade adequada. *Encoders* incrementais devem emitir, no mínimo, dois pulsos defasados de noventa graus. Estes sinais são normalmente chamados de pulso A e pulso B. De acordo com a sequência dos sinais, é possível obter a posição e a direção do elemento móvel (detalhes sobre o funcionamento de *encoders* incrementais estão no Cap. 3.3.4.1).

Máquinas ferramentas utilizam *encoders* lineares (translacionais) que são elementos geralmente caros e exigem montagem em plataformas com alto grau de rigidez e alinhamento muito preciso. *Encoders* rotacionais são mais baratos e mais frequentemente utilizados para a medição de posição em servoposicionadores pneumáticos.

Para a utilização de *encoders* rotacionais na medição da posição de atuadores lineares, utiliza-se geralmente uma correia dentada que é acoplada a uma polia ligada por sua vez ao *encoder*. Duas configurações básicas são encontradas na bibliografia como: correia fixa com *encoder* deslocando-se com o posicionador (Pu et al., 1992) e correia móvel com o *encoder* fixo (Lin et al., 1996). No primeiro caso, encontra-se dificuldade no alinhamento da correia com o *encoder*. A correia, neste caso, é geralmente acoplada ao êmbolo do atuador. No segundo caso a dificuldade fica por conta das vibrações geradas pela rápida movimentação do posicionador, oriundas do contato entre os dentes da polia com a correia. No presente trabalho optou-se por esta segunda configuração.



Os sensores de pressão são conectados a tubulação localizada entre a válvula e o atuador. É aconselhado manter este trecho de tubulação o mais curto possível e sem curvas, visando reduzir as quedas de pressão no sistema (Latino e Sandoval, 1996).

## 2.2 Soluções para o Controle Digital

Segundo Edge (1997), pode-se desenvolver sistemas de controle baseados em 3 diferentes tipos de implementações: Analógica, Híbrida e Digital. Observa-se também uma tendência na utilização de controladores com maior flexibilidade (ver Fig. 1.4), ou seja, controladores com maior percentual de soluções adequadas a cada tipo de aplicação. Muitos dos controladores digitais para movimento oferecem facilidades de programação, porém, os algoritmos de controle são relativamente fixos e não modificáveis após implementados (Edge, 1997).

O principal sinal adquirido em grande parte das técnicas de controle é a posição do êmbolo do cilindro pneumático. A partir da derivação numérica deste sinal pode-se obter informação sobre velocidade e aceleração. Estas informações são geralmente utilizadas pelos algoritmos de controle para determinar a intensidade do sinal de atuação na servoválvula. Grande parte dos trabalhos estudados determinam o tempo de atualização do sinal de controle em 1ms (Vieira, 1998; Andrighetto et al., 2005a, 2005b; Perondi, 2002; Giberti et al., 2001). Na bibliografia encontra-se também valores de 3ms (Latino e Sandoval, 1996), 4ms (Pu et al., 1992) e 5ms (Virvalo, 1989).

Devido a significativa queda nos preços e avanço da eletrônica, a utilização de centrais únicas de processamento (CPU - *Central Processing Unit*) para controle de processos tem sido mais difundida e, dependendo das necessidades da aplicação, pode-se utilizar diferentes tipos de implementação: Controlador Dedicado, Controle Distribuído e Controle Orientado a Objetos.

### 2.2.1 Controlador Dedicado

O controlador dedicado (ver Fig. 2.3) é um sistema isolado (*stand-alone*) com uma única central de processamento, responsável por processar todas as funções do sistema, incluindo o monitoramento, interfaceamento com o usuário e operações em tempo real (tais como a realimentação de controle, decisões quanto a segurança, tratamento de dados, etc.). Para isto são necessárias políticas de sincronização entre as tarefas, assim como, acesso aos dispositivos (sensores e transdutores).

Vários trabalhos utilizam controladores dedicados para a realização das tarefas de desenvolvimento de controladores e monitoramento dos sistemas. Como exemplo pode-se citar Perondi (2002), Vieira (1998), Andrighetto et al. (2005b) (estes autores utilizam placas de controle baseadas em DSP (*digital signal processor*)). Porém, esta é uma solução com custo relativamente

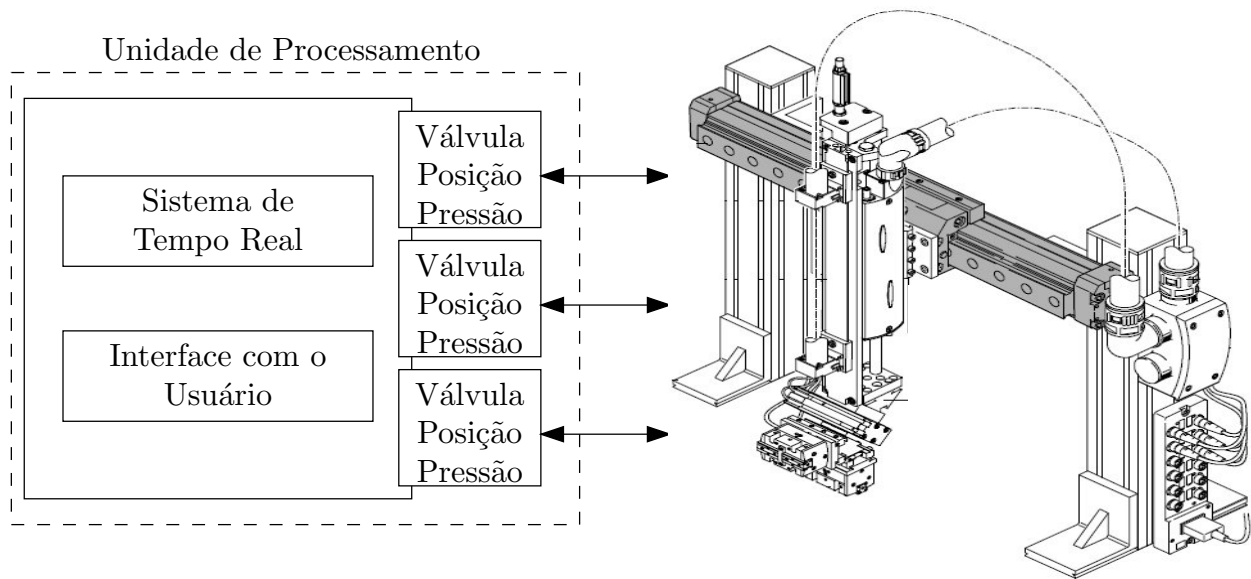


Figura 2.3 – Controle dedicado

alto. Placas como a DSpace (dSPACE GmbH, 2005), são hospedadas em *slots* de microcomputadores compatíveis com IBM-PC e possuem características tais com realizar a conversão analógico/digital, conversão digital/analógico e executar a lógica de controle, geralmente desenvolvida em ambientes amigáveis com programação por diagrama de blocos ou em linguagem de alto nível. Além de garantir a eficiência temporal de resposta, requisito necessário em controle de sistemas, placas como a DSpace também fornecem ambientes de monitoramento das variáveis da planta (dSPACE GmbH, 2005).

A possibilidade de desenvolver a solução de controle em FPGA (*field-programmable gate array*) foi estudada, porém o tempo e recursos necessários para o desenvolvimento de tal solução estavam além dos objetivos do projeto. A tecnologia baseada em FPGAs permite flexibilidade nas arquiteturas desenvolvidas. Por outro lado, porém, são necessários recursos avançados na área de Engenharia de Software e Hardware, fugindo do escopo deste trabalho.

Em Rajendran e Bolton (2003), uma placa de aquisição PCI 6024E DAQ é utilizada para realizar o controle da servoválvula e a aquisição da posição através de um potenciômetro linear. Em Lara-Rojo et al. (2001) é utilizado um microcontrolador PIC16C711 na implementação de um controlador baseado em técnica *fuzzy*. Os resultados de temporização do controlador desenvolvido foram satisfatórios. Em Anakwa et al. (2002) é utilizado um microcontrolador Motorola 68HC16 para realizar o controle de uma suspensão ativa pneumática.

Giberti et al. (2001) implementaram um sistema dedicado utilizando um microcomputador compatível com IBM-PC para a realização de controle de um robô acionado pneumáticamente. O controle foi realizado por uma placa AD/DA acoplada ao computador. O controle foi implementado utilizando como ferramenta para controle em tempo real, o sistema operacional *RT-Linux*

(ver maiores informações sobre esta arquitetura no Cap. 4), resultando em um tempo de ciclo de controle em 0.001 segundos. Para o controle de posição foi utilizado um *encoder* acoplado a uma placa decodificadora inserida no PC.

### 2.2.2 Controladores Distribuídos

Controladores distribuídos (Fig. 2.4) são unidades de processamento, em tempo real, dedicados individualmente a cada dispositivo. As unidades estão ligadas a um barramento de comunicação que estabelece a comunicação à outra unidade de processamento responsável pelo interfaceamento com o usuário, diminuindo, assim a solicitação de processamento individual do sistema e facilitando a utilização de interfaces mais *amigáveis*, por disponibilizar maiores recursos.

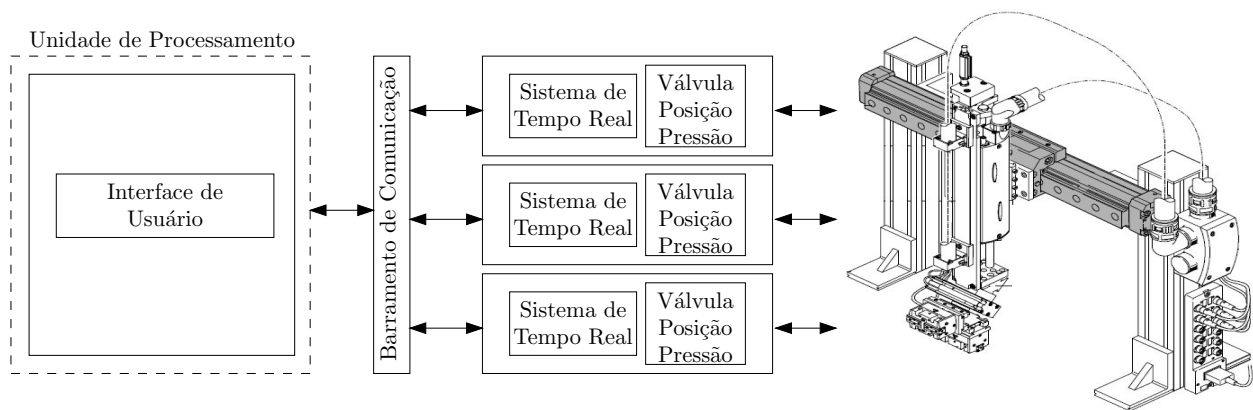


Figura 2.4 – Controle Distribuído

Em geral, a decisão de utilizar um sistema de controle distribuído é motivada pela redução nos custos e aumento da flexibilidade do sistema de controle. Em Lages e Alt (2003) é apresentada uma arquitetura de controle distribuída para um robô manipulador através de uma rede IP (*Internet Protocol*) convencional, onde as mensagens de controle estão sujeitas a atrasos durante o percurso. Neste trabalho são propostas técnicas para aumentar o desempenho do controlador, diminuindo estes atrasos.

No controle orientado a objetos encontram-se características intrínsecas da utilização desta filosofia de programação, tais como: alto grau de reutilização, portabilidade, modularidade e fácil aprendizagem. O orientação a objetos, a partir do desenvolvimento de novas ferramentas, vem sendo crescentemente utilizada em aplicações de automação (Pereira et al., 2005).

### 2.2.3 Controladores Orientados à Objetos

O esquema de controle orientado a objetos é similar ao de controladores distribuídos, porém o tratamento dispendido a cada dispositivo (válvula, sensor de posição, sensor de pressão e configuração do controle em tempo real) utiliza a filosofia de orientação a objetos, abstraindo informações

sobre implementações específicas de cada dispositivo, resultando em uma maior padronização de interfaces e na padronização de *softwares* para construção de algoritmos de controle (Edge, 1997).

A utilização de ferramentas orientadas a objetos utilizando política e requisitos de tempo real apresentam-se bastante promissoras. Um exemplo disto é a RTSJ (*Real Time Specification for Java*). Porém, trabalhos como Pereira et al. (2005) destacam alguns aspectos que merecem especial atenção devido a existência de melhorias necessárias para sua utilização em aplicações críticas.

#### 2.2.4 Microprocessadores e Microcontroladores

Os trabalhos estudados permitem concluir que, de maneira geral, as soluções baseadas no uso de microprocessadores ou microcontroladores são mais acessíveis financeiramente que as soluções flexíveis baseadas em DSP (como a *DSpace*).

Os microprocessadores são geralmente utilizados em aplicações de alto desempenho onde o custo e tamanho do *hardware* não são críticos. Isto deve-se à necessidade do uso de memória e dispositivos de entrada e saída externos. Microprocessadores são largamente aplicados em soluções onde a compatibilidade, desempenho e flexibilidade são requisitos importantes.

Os microcontroladores são geralmente utilizados para diminuir o valor total do sistema, pois incorporam memória e dispositivos de entrada e saída de sinais. Geralmente, fabricados para aplicações especializadas, assim, diminuindo a flexibilidade das soluções. Suas aplicações mais comuns são em sistemas de segurança, eletrodomésticos, dispositivos móveis, etc.

Tipicamente, os periféricos encontrados embutidos em microcontroladores são temporizadores, contadores, interfaces paralelas e seriais, conversores analógicos/digitais e digitais/analógicos. Estes periféricos geralmente possuem desempenho inferior aos dos circuitos dedicados para estas funções em conjunto com microprocessadores (Kilian, 2000).

Como vantagens de microcontroladores podem ser apresentadas:

- funções já presentes no dispositivo;
- baixo custo e tamanho minimizado;
- baixa potência de funcionamento.

Como desvantagens de microcontroladores tem-se:

- baixa flexibilidade;
- expansão limitada de memória ou dispositivos de comunicação;

- limitadas taxas de transferência de dados.

Resumidamente, pode-se destacar suas principais características como:

- microprocessadores são mais flexíveis;
- microcontroladores são mais compactos.

Conclui-se que uma solução híbrida, que utiliza as vantagens de ambas estratégias, conjugando microcontroladores e microprocessadores (*encoders* incrementais, por exemplo, necessitam soluções em *hardware* (Lages, 2004)), mantendo a flexibilidade necessária para o desenvolvimento de complexas técnicas de controle utilizando microprocessadores é adequada para o problema de controle de uma bancada servopneumática.

A tendência de utilizar recursos implementados em *software* em vez de implementados em *hardware*, não somente auxilia na flexibilidade, mas também permite utilizar dispositivos mais avançados de controle, como interface com câmeras, sistemas SCADA (*Supervisory Control and Data Acquisition*) e outros dispositivos de monitoramento (Kunz et al., 2003).

### 2.3 Técnicas de Controle

Existem dois tipos básicos de tarefas de posicionamento para servoposicionadores: problemas de controle de posição e problemas de controle de trajetória. Para o controle de posição trabalha-se somente com o erro que é definido pela diferença entre a posição atual e a posição desejada (final). O controle de trajetória faz com que o êmbolo do atuador pneumático siga uma dada trajetória em um certo tempo. O controle de trajetória tem como vantagem a possibilidade de impor velocidade e aceleração durante a transição entre posições (Scavarda e Thomasset, 1996).

Como a modelagem clássica, desprezando a dinâmica da servoválvula, resulta em um sistema não-linear de quarta ordem (Perondi, 2002), devido às propriedades do ar e da força de atrito, opera-se no sentido de modificar a dinâmica original do sistema, trabalhando em malha fechada para diminuir as influências das não-linearidades e incertezas de medição sobre a resposta do sistema, visando tornar o erro de posicionamento ou de seguimento de trajetória suficientemente pequeno.

Seguem as descrições de alguns tipos de controladores utilizados para servoposicionadores pneumáticos:

### 2.3.1 Controladores PID

Os controladores PID (Proporcional-Integral-Derivativo) são muito difundidos, principalmente para a tarefa de controle de sistemas lineares. Os controladores PID utilizam como realimentação o valor de erro, a integral do erro com relação ao último intervalo de tempo e a derivada do sinal de erro para determinar a correção a ser aplicada na malha de controle, como ilustrado na Fig. 2.5.

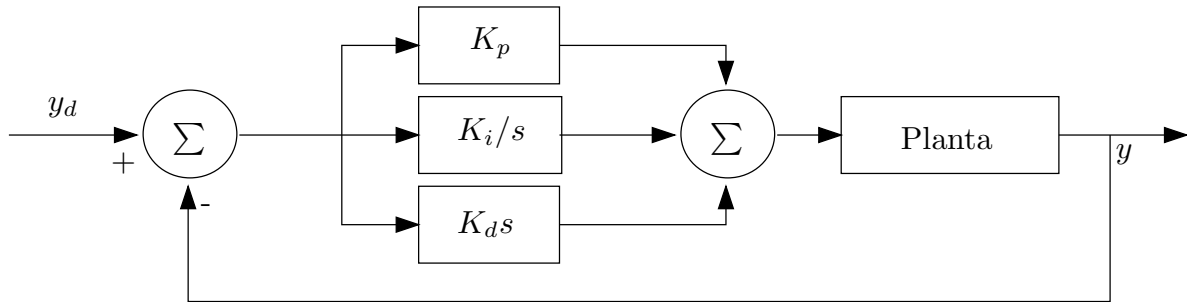


Figura 2.5 – Esquema simplificado para controle de posição PID

Perondi (2002) demonstra que as oscilações pouco amortecidas e o risco de instabilidade limitam o ganho de um controlador proporcional (P) a pequenos valores, resultando em um posicionamento lento em malha fechada. Segundo Virvalo (1995), pequenos ganhos proporcionais resultam em grandes erros provocados pelo atrito. Segundo Liu e Bobrow (1988) a introdução da parcela derivativa na realimentação não auxilia significativamente a resolução destas dificuldades.

Segundo Perondi (2002), uma parcela integral pode auxiliar na redução do erro de regime provocado pelo atrito, mas também não altera a posição dos pólos em malha fechada de forma a reduzir as oscilações pouco amortecidas e os riscos de instabilidade.

Controladores PID resultam insuficientes quando exigências da tarefa do posicionador são severas em termos de precisão e velocidade, conforme atesta uma grande quantidade de trabalhos como Kawamura et al. (1989), Tanaka et al. (1996), Song et al. (1997), Pandian et al. (1997) e Okiyama e Iciryu (2001).

Devido às dificuldades descritas acima, controladores PID muitas vezes são utilizados de forma híbrida com outras técnicas inclusive não-lineares (Kawamura et al., 1989; Song et al., 1997; Choi et al., 1998).

### 2.3.2 Controladores por realimentação de estados

*Variáveis de estado* são definidas como o menor conjunto de valores de variáveis que, quando conhecidas, podem determinar completamente a evolução de resposta do sistema (Ogata, 1998). Como variáveis de estado pode-se utilizar, por exemplo, posição, pressão, velocidade, aceleração, temperatura, etc.

O método de controle por realimentação de estados pode ser definido como uma técnica através da qual projeta-se uma compensação dinâmica para o sistema trabalhando-se diretamente com sua descrição em variáveis de estado (Franklin et al., 1994).

Controladores por realimentação de estados que utilizam posição, velocidade e aceleração são denominados PVA. A Fig. 2.6 apresenta um controlador PVA, sendo  $K_p$ ,  $K_v$  e  $K_a$ , respectivamente os ganhos do sinal de posição, velocidade e aceleração.

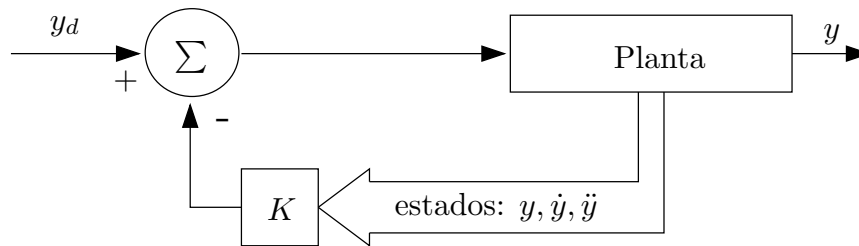


Figura 2.6 – Controle por realimentação de estados (PVA)

Uma das dificuldades de utilizar eficientemente controladores PVA está na medição da aceleração, pois sua medição é cara, difícil e seu cálculo numérico é contaminado pelos ruídos do sinal da velocidade e a utilização de filtros introduzem atrasos no sistema (Perondi, 2002).

Uma alternativa à utilização do sinal de aceleração é o uso da diferença de pressão entre as câmaras do servomotor no lugar da aceleração, visto que o sinal de pressão é menos sensível a ruídos externos. Esta técnica é chamada de  $PV\Delta p$  (Boudart et al., 1991; Surgenor et al., 1995; Perondi e Guenther, 1999a). Trabalhos como Perondi e Guenther (1999a) mostram que, para finalidade de seguimento de trajetória, os erros obtidos são cerca de 50% maiores ao utilizar  $PV\Delta p$  em comparação a PVA, pois sua capacidade de responder às perturbações ocasionadas pelo atrito é menor.

Em aplicações de controle de posição de servoposicionadores pneumáticos, em que apenas a posição atual do êmbolo deve ser atingida dentro de uma precisão específica, verifica-se que o uso de controladores de estados com realimentação de pressão permite obter um bom desempenho (Perondi e Guenther, 1999b). Porém, quando a tarefa do posicionador exige o seguimento de trajetória, o atrito exerce uma influência muito acentuada no comportamento do posicionador, principalmente quando as velocidades são pequenas. Neste caso os controladores de estado tradicionais podem ser insuficientes, necessitando-se adicionalmente de uma compensação para o atrito.

### 2.3.3 Linearização por Realimentação

A idéia básica da linearização por realimentação é a de transformar parcial ou completamente um sistema não-linear em um sistema com dois laços de realimentação: um interno, utilizado para

compensar as não-linearidades através de uma realimentação adequada dos seus estados e um laço externo, onde são aplicadas as técnicas usuais de controle de sistemas lineares.

A Fig. 2.7 apresenta um esquema da técnica de linearização por realimentação aplicado em conjunto com a técnica linear de realimentação de estados (PVA).

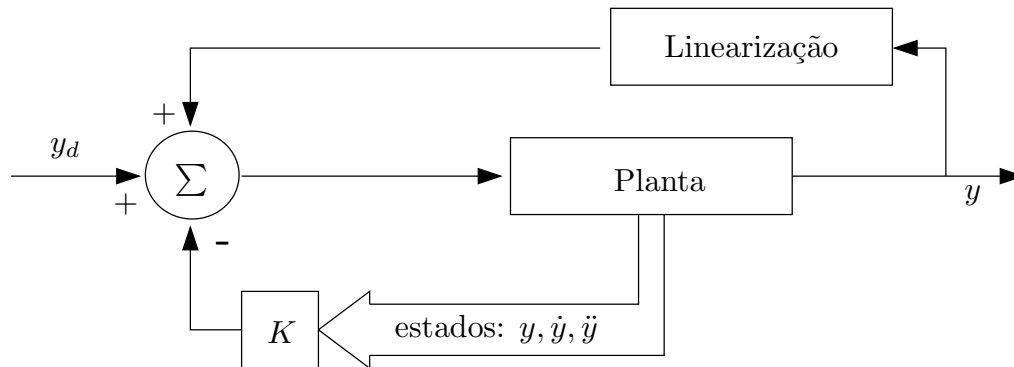


Figura 2.7 – Linearização por realimentação de estados

Na bibliografia sobre controle de servoposicionadores pneumáticos há vários trabalhos que contemplam a aplicação do método de linearização por realimentação de estados tais como Hanchin et al. (1992), Belgharbi et al. (1998), Brun et al. (2000), Thomasset et al. (1993), Richard e Scavarda (1996), Scavarda e Thomasset (1996) e Gottert e Neumann (1999).

Segundo comparações realizadas por Gottert e Neumann (1999), para seguimento de trajetória, o método de linearização por realimentação de estados fornece resultados com menor erro de seguimento e menos oscilatórios do que os resultados obtidos com o uso dos controladores por realimentação de estados. Esses resultados são confirmados por Brun et al. (2000), que acrescenta, ainda, a verificação da diminuição do fenômeno adere-desliza (*stick-slip*).

O método de linearização por realimentação normalmente requer o conhecimento do vetor de estados completo e é aplicável aos sistemas de fase mínima (com dinâmica interna estável), não garantindo, portanto, a robustez quanto às incertezas de parâmetros ou perturbações (Slotine, 1991).

### 2.3.4 Controle a Estrutura Variável

A robustez é uma característica dos sistemas de posicionamento geralmente necessária para sua aplicação em robótica. Um sistema é dito robusto quando ele apresenta mudanças aceitáveis no desempenho na presença de alteração na planta e imprecisões no modelo (Dorf, 1989). Assim, o objetivo do projeto de sistemas robustos é manter as características dinâmicas e de estabilidade do sistema apesar das mudanças na planta e imprecisões do modelo.

Os controladores com estrutura variável são conhecidos pela sua propriedade de alta robu-



tez (Edge, 1997). Vários autores têm proposto e estudado a aplicação de algoritmos de controle com estrutura variável em servoposicionadores pneumáticos, tais como Hanchin et al. (1992), Tang e Walker (1995), Thomasset et al. (1993), Paul et al. (1993, 1994), Bouri et al. (1994), Surgenor et al. (1995), Scavarda e Thomasset (1996), Surgenor e Vaughan (1997), Pandian et al. (1997, 1998, 1999), Hatipoglu et al. (1999) e Su e Kuo (2000).

Basicamente, esta técnica consiste no chaveamento entre diferentes leis de controle com o propósito de manter o sistema em equilíbrio. O equilíbrio do sistema é definido pela chamada *superfície de deslizamento* (especificada em projeto) pois, em situação ideal, uma vez atingida esta superfície, o sistema mantém-se no chamado *regime de deslizamento* ou *modo deslizante*.

Porém, na prática de controle de servoposicionadores, para manter a trajetória de estados no regime de deslizamento, é necessário a aplicação de repetidas trocas de leis de controle para manter o sistema dentro de um erro especificado em projeto. Este chaveamento, em sistemas reais, ocorre em um tempo finito que pode dar origem a oscilações (fenômeno conhecido por *chattering*) que podem provocar desgastes nas peças móveis do atuador e introduzem ruído no sistema.

A Fig. 2.8 exemplifica o ruído de *chattering* gerado pelas tentativas de manter o sistema em modo deslizante, onde  $y$  e  $\dot{y}$  são duas variáveis de estados (posição e velocidade) e  $S(x) = 0$  denotando a curva da trajetória de estados relativa à superfície de deslizamento.

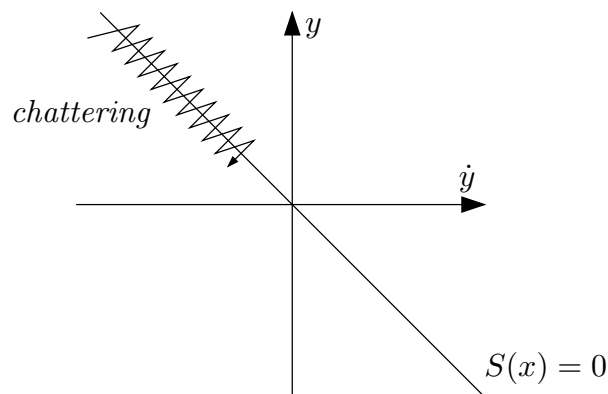


Figura 2.8 – Fenômeno de *Chattering*.

Este ruído, inerente a esta técnica de controle, somado ao ruído proveniente da derivação do sinal de velocidade para a obtenção da aceleração é uma das maiores dificuldades para a aplicação desta técnica para servoposicionadores pneumáticos.

A Fig. 2.9 exemplifica um esquema de um controlador por modos deslizantes aplicado a um servoposicionador, onde  $\eta$  representa os modos de chaveamento e  $K_p$ ,  $K_v$  e  $K_a$  os ganhos, respectivamente, para a posição, velocidade e aceleração.

Resultados apresentados por Thomasset et al. (1993) concluem que os desempenhos obtidos tanto com a utilização de controladores baseados em linearização por realimentação quanto por

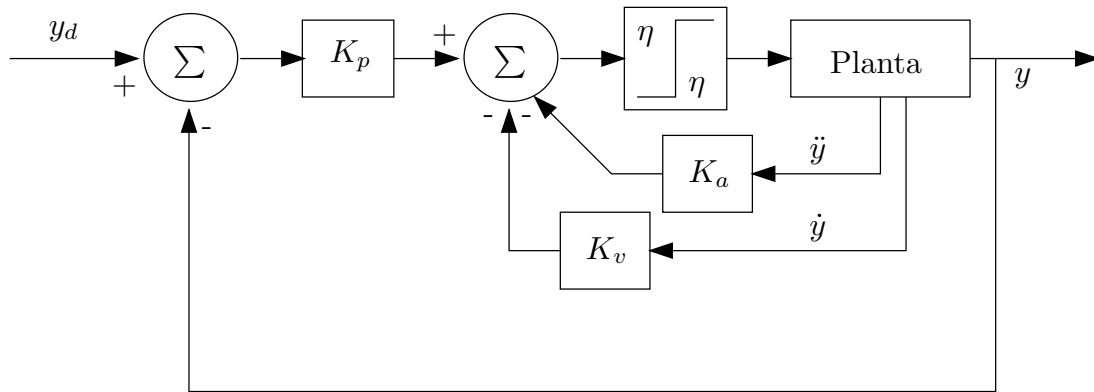


Figura 2.9 – Controlador por modos deslizantes aplicado a servoposicionadores

controladores baseados em estrutura variável, aplicados a servoposicionadores pneumáticos, são satisfatórios.

Resultados de estudos de comparação entre controladores por modos deslizantes e controladores por realimentação de estados PVA e  $PV\Delta_p$  apresentados por Surgenor et al. (1995) e Surgenor e Vaughan (1997) comprovam a maior robustez do controlador a estrutura variável quando há variação da carga a ser deslocada.

### 2.3.5 Controles baseados na interpretação de divisão em dois subsistemas

A idéia básica desta técnica de controle para servoposicionadores pneumáticos está na interpretação do atuador como composto por dois subsistemas: um mecânico e outro pneumático (Kawamura et al. (1989), Lai et al. (1990), Noritsugu e Takaiwa (1995), McDonell (1996), McDonnell e Bobrow (1997), Choi et al. (1998), Bobrow e McDonell (1998)).

Conforme ilustrado na Fig. 2.10, o subsistema pneumático necessita fornecer uma força  $AP_\Delta$  para que o subsistema mecânico consiga executar uma determinada trajetória.

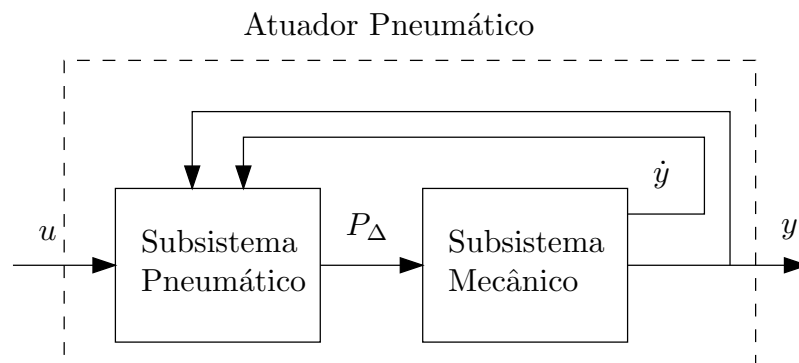


Figura 2.10 – Interpretação do sistema de posicionamento

A estratégia de controle baseada na divisão em dois subsistemas facilita o projeto de controle, pois permite a escolha de leis de controle mais adequadas para cada subsistema. Isto viabiliza que

as leis de controle clássicas projetadas para os sistemas com acionamento elétrico, por exemplo, sejam utilizadas para o controle de sistemas com outras tecnologias de acionamento (McDonnell e Bobrow, 1997).

### 2.3.6 Controladores adaptativos

Os controladores adaptativos executam o ajuste dos parâmetros do controlador automaticamente, de acordo com as condições predominantes de trabalho. Isto pode ser feito usando o conhecimento prévio dos fatores que influem no desempenho (principalmente os ganhos do sistema) ou pelo emprego de um esquema auto-adaptativo (Edge, 1997).

Para realizar a programação de ganhos é necessário o conhecimento prévio de como os parâmetros afetam o comportamento do sistema. A utilização de *tabelas* de ganhos (*gain schedule*) em problemas de servoposicionadores pneumáticos foi estudada por Pu et al. (1993).

No método auto-adaptativo, o controlador adapta-se ao modo que a resposta dinâmica do sistema pneumático e a mudança de parâmetros do controlador são reinseridos na base de conhecimento. Estas oscilações ocorrem usualmente através de variações no carregamento ou na fonte de pressão e como resultado das não-linearidades do sistema. Este método geralmente utiliza de um modelo da planta para identificar seus parâmetros usando os dados de entrada e saída do sistema (Edge, 1997).

A Fig. 2.11 demonstra o funcionamento de um controlador auto-adaptativo utilizando o método de alocação de pólos.

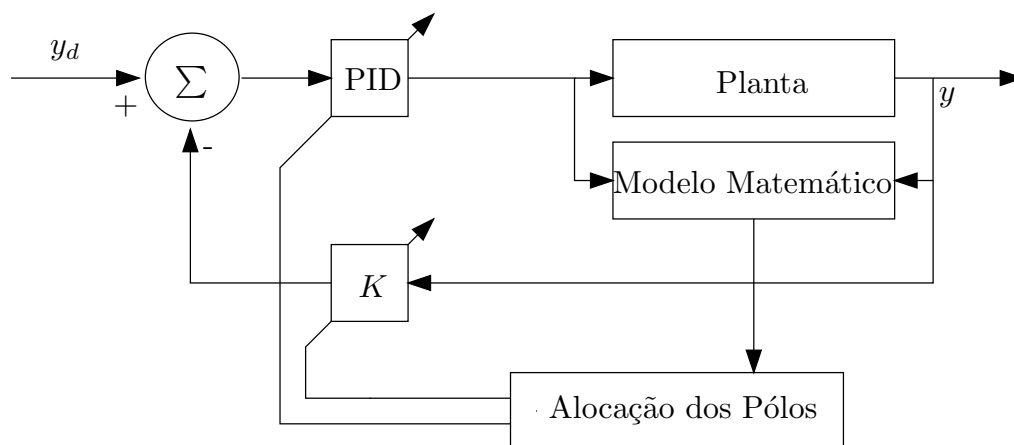


Figura 2.11 – Controlador auto-adaptativo

Os resultados obtidos por Tanaka et al. (1996) mostram que no problema de deslocamento ponto a ponto o uso de controlador adaptativo mantém o desempenho de um servoposicionador pneumático praticamente inalterado, mesmo variando em 6 vezes a massa deslocada e com a aplicação de forças externas que variam de 0 a 10N.

Outro esquema similar ao auto-adaptativo é o MRAC (*Model Reference Adaptive Control*). Este método tem como vantagem a não necessidade de utilização de um modelo matemático da planta (Edge, 1997). Porém, testes comparativos executados por Bobrow e Jabbari (1991), mostram melhores resultados na utilização de alocação de pólos em relação ao MRAC.

### 2.3.7 Algoritmos Genéticos

Os algoritmos genéticos são técnicas não-determinísticas de busca, otimização e aprendizagem de máquina. Estes algoritmos manipulam um espaço de soluções potenciais utilizando mecanismos inspirados nas teorias de seleção natural de Charles Darwin e na genética de Gregor Mendel. Os algoritmos genéticos são robustos e eficientes em espaços de procura irregulares, multidimensionais e complexos (Coelho e Rodrigues, 1999).

Um dos principais benefícios da utilização de algoritmos genéticos na otimização dos parâmetros de controle é que eles não são específicos para cada sistema, sendo de fácil aplicação (Edge, 1997). Algoritmos genéticos necessitam da determinação dos indicadores de qualidade e dos parâmetros de entrada e de saída necessários do sistema, conforme ilustrado na Fig. 2.12, onde está apresentado um exemplo de aplicação para otimização dos ganhos de um controlador PID.

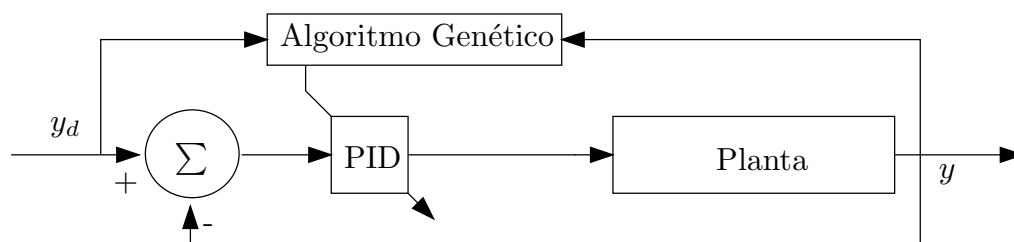


Figura 2.12 – Algoritmo genético para otimização dos parâmetros do controlador PID

### 2.3.8 Controle por lógica difusa

O controle por lógica difusa (*Fuzzy Logic Control*) apresenta a vantagem de não necessitar de um modelo matemático e de o ajuste do controlador poder ser realizado heurísticamente ou baseado em outras informações ou conhecimentos prévios (Edge, 1997). Shih e Huang (1997) utiliza um esquema de controle por largura de pulsos (PWM - *pulse with modulation*) com lógica difusa para controlar as posições linear e rotacional de um cilindro pneumático especial (com dois graus de liberdade).

A Fig. 2.13 apresenta o diagrama de blocos de um controlador por lógica difusa. A utilização de lógica difusa (*fuzzification*) é responsável por converter o sinal em termos lingüísticos, tais como: 'perto', 'muito perto', 'longe' e 'muito longe'. Estes termos, enviados ao bloco controlador são comparados a uma base de dados com regras previamente definidas e assim determinando a

ação correspondente que passa pelo bloco de recuperação do sinal (*defuzzification*) que converte os termos linguísticos em sinal de controle.

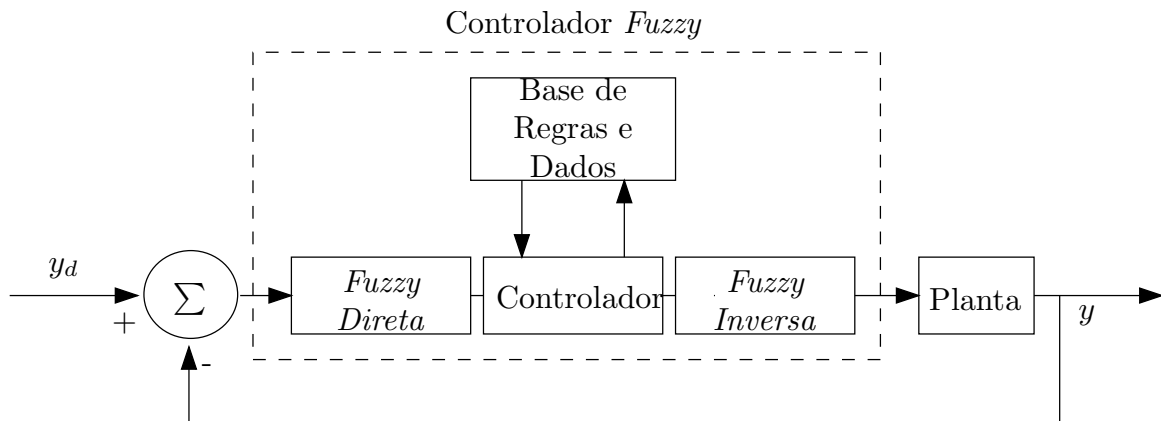


Figura 2.13 – Controlador por lógica difusa

### 2.3.9 Controle por redes neurais

O controle por redes neurais, para ser implementado, não necessita de um modelo matemático do servoposicionador. Em sua forma mais simples, uma rede neural atua como uma função de mapeamento não-linear entre um vetor de entrada e um vetor de saída, e é geralmente utilizada diretamente como controlador ou como um meio de modelagem dinâmica da planta (Edge, 1997).

Gross e Rattan (1997) utilizam, juntamente com um controlador proporcional, o esquema multicamadas (MNN - *multilayer neural network*) para compensar as não-linearidades. Este controlador está representado pelo esquema apresentado na Fig. 2.14.

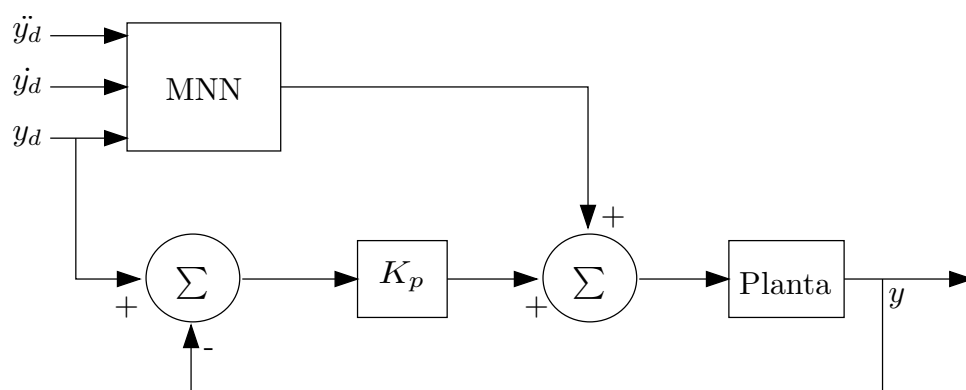


Figura 2.14 – Controlador MNN com realimentação proporcional de posição

O esquema MNN está representado na Fig. 2.15(a). Cada círculo do conjunto de camadas corresponde a um neurônio lógico (ver Fig. 2.15(b)), onde cada um realiza um somatório das entradas (cada entrada com um *peso* representado por  $w_1$ ,  $w_2$  e  $w_3$ ) e passa este resultados para uma função não linear que determina a saída para o próximo neurônio. Esta função não-linear é

resultado de treinamento prévio do controlador realizado através de testes com o servoposicionador pneumático

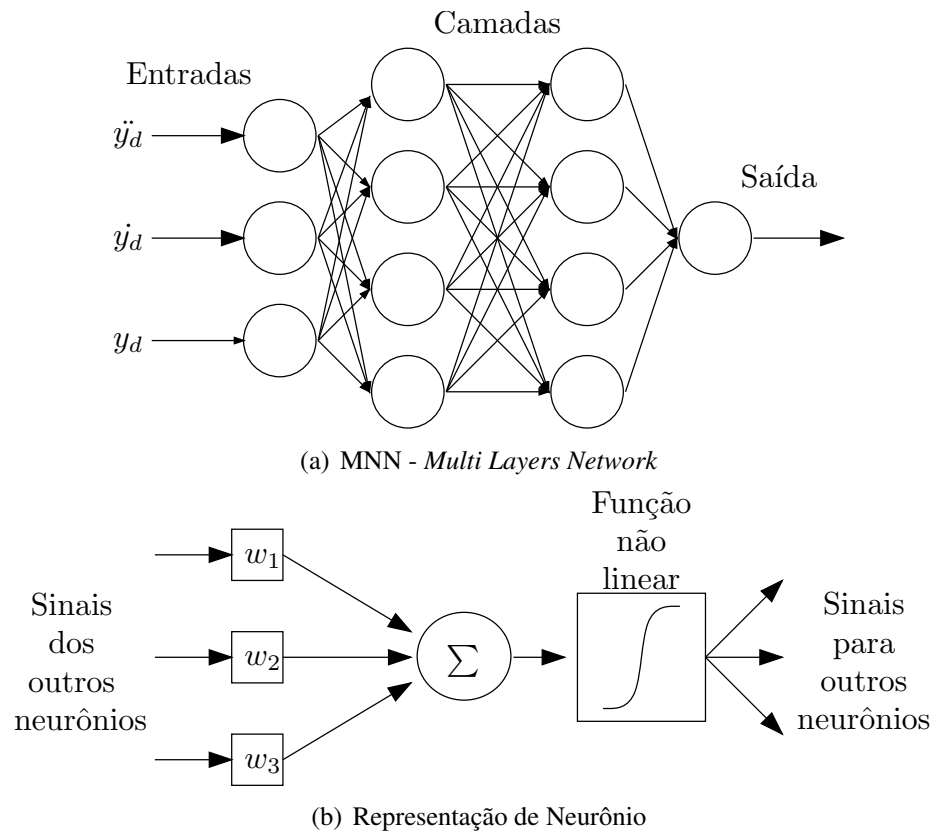


Figura 2.15 – Controle por redes neurais

### 3 IMPLEMENTAÇÃO EXPERIMENTAL

Neste capítulo são apresentados os principais tópicos relacionados à montagem física do servoposicionador pneumático. O equipamento utilizado e as soluções desenvolvidas para condicionamento, controle e aquisição de sinais são descritos aqui.

#### 3.1 Introdução

A implementação experimental do servoposicionador pneumático no Laboratório de Robótica do Departamento de Engenharia Mecânica da Universidade Federal do Rio Grande do Sul é mostrada na foto apresentada na Fig. 3.1.

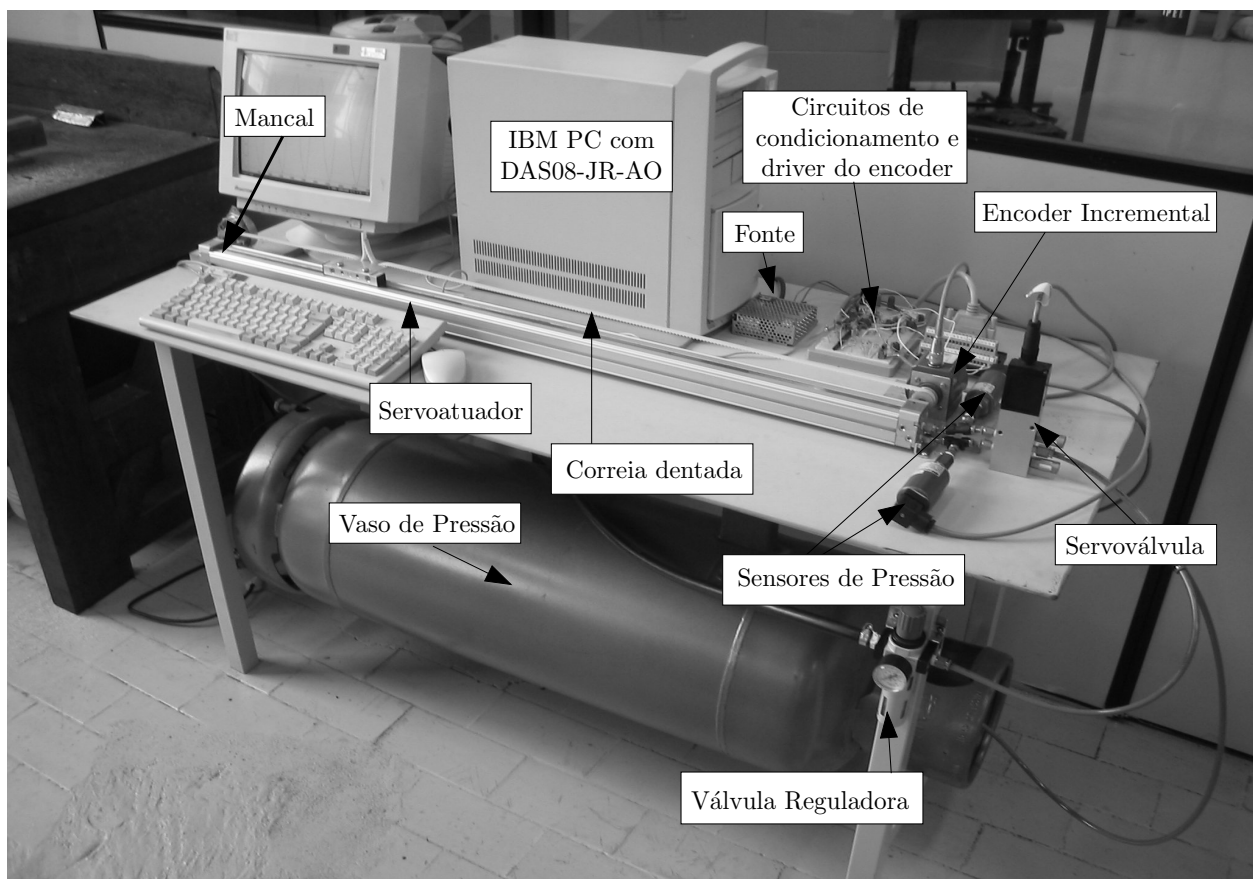


Figura 3.1 – Bancada de testes no Laboratório de Robótica da UFRGS.

#### 3.2 Descrição Geral do Sistema

Conforme apresentado nas Figuras 3.1 e 3.2, o sistema é constituído pelos seguintes componentes: uma válvula reguladora de pressão com filtro acoplado, um vaso de pressão, uma servoválvula direcional, dois sensores de pressão, um cilindro atuador, um sensor óptico rotacional

(*encoder*), uma correia dentada, circuitos condicionadores e a placa de conversão de sinais. O microcomputador (IBM PC) é responsável pelo controle e supervisão da bancada.

Pode-se observar o esquema das conexões através da Fig. 3.2

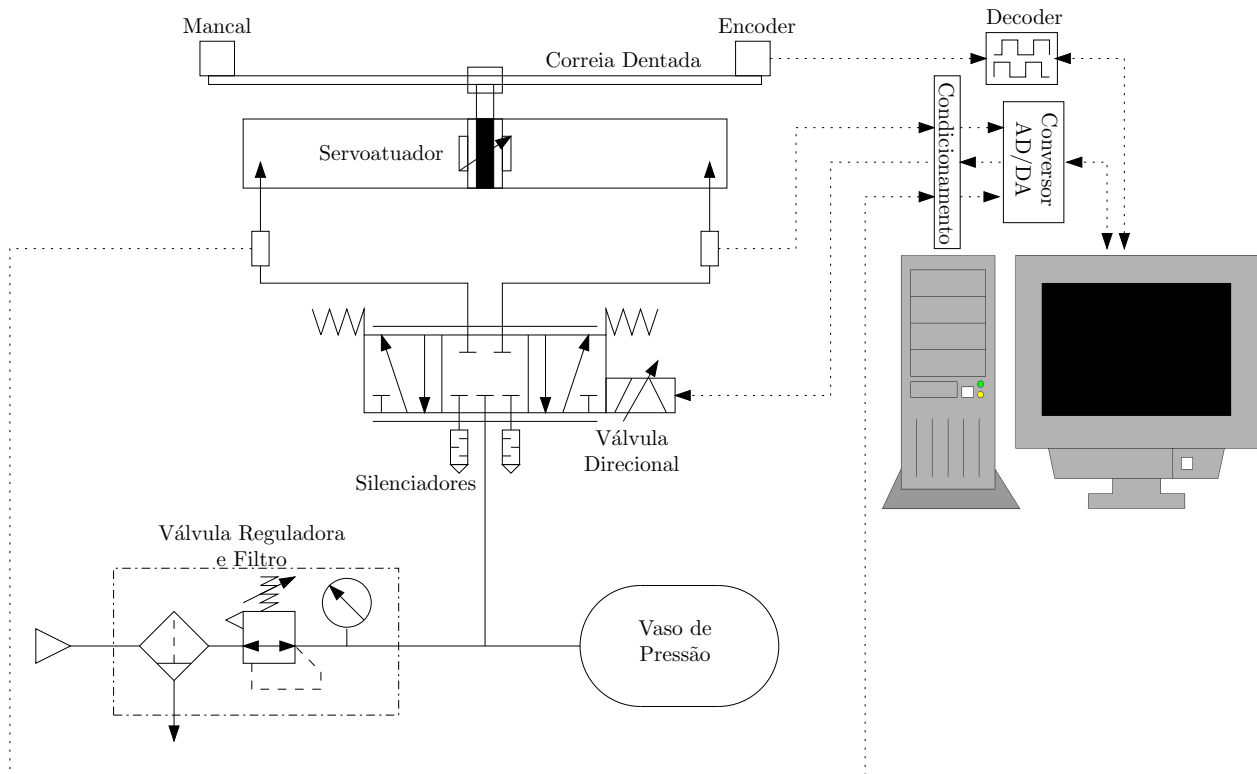


Figura 3.2 – Esquema da bancada experimental

Visando facilitar a sua descrição, o sistema está subdividido em 4 partes: tratamento do fluido, atuador pneumático, transdutores e sistema de aquisição e controle. Seguem suas descrições.

**Sistema de tratamento de fluidos.** Este sistema é composto pela unidade de filtragem, pela válvula proporcional reguladora de pressão e pelo vaso de pressão. O filtro é responsável por purificar o ar, eliminando suas impurezas, removendo também vapor d'água provindo da rede de suprimento. A válvula reguladora de pressão permite ajustar a pressão de trabalho de forma a mantê-la aproximadamente constante, independentemente das variações sofridas pela rede de distribuição de ar. O vaso de pressão, instalado em paralelo com a servoválvula, serve para reduzir as flutuações da pressão de trabalho decorrentes das variações de demanda de consumo pelo sistema em operação.

**Atuador pneumático.** Este subsistema é constituído pela servoválvula direcional e pelo cilindro atuador. A servoválvula direcional serve para controlar a força e direção do atuador pneumático de acordo com um sinal de controle, atuando na diferença de pressão entre as câmaras do atuador que fornece a força necessária para seu deslocamento.

**Transdutores.** São os componentes que fornecem para o sistema de controle as informações



sobre as pressões nas câmaras e a posição do atuador pneumático. A posição é adquirida através de um encoder acoplado a uma correia dentada que, por sua vez, está conectada ao êmbolo móvel do atuador pneumático. As pressões nas câmaras são adquiridas por sensores de pressão conectados entre a entrada do atuador pneumático e a saída da válvula proporcional. Ambos os sensores (pressão e posição) necessitam de condicionamento de sinais antes destes serem adquiridos pelo sistema de controle.

**Sistema de aquisição de dados.** É constituído pelo circuito de condicionamento e operação (*drivers*) do encoder e a placa de conversão A/D e D/A. A função deste sistema é realizar a aquisição dos sinais utilizados em tempo real pelo sistema de controle e enviar os sinais analógicos de controle para a servoválvula.

### 3.3 Componentes

Segundo dados do fabricante, os dutos utilizados em todas as conexões do sistema podem operar com pressões de até 1 MPa. Para os encaixes entre os dispositivos (válvulas, sensores e atuadores) e as tubulações foram utilizadas engates rápidos (Festo Corporation, 2005).

#### 3.3.1 Componentes do Subsistema de Tratamento do Fluido

Seguem as descrições dos componentes do subsistema de tratamento do fluido.

##### 3.3.1.1 Filtro e Regulagem de Pressão

As funções de filtragem e regulagem da pressão de trabalho são realizadas por uma única unidade de tratamento de ar (Festo, LFR-1/4-D-MINI) proporcionando menor custo e economia de espaço. A Fig. 3.3 mostra a representação esquemática da unidade com suas funções, onde 1 é a entrada de ar comprimido, 2 é a saída de ar processado e 3 é a saída de água, resultado da desumidificação e controlada por um dreno manual. A regulagem de pressão é feita por uma válvula manual e pode ser visualizada por meio de um manômetro acoplado à unidade de tratamento.

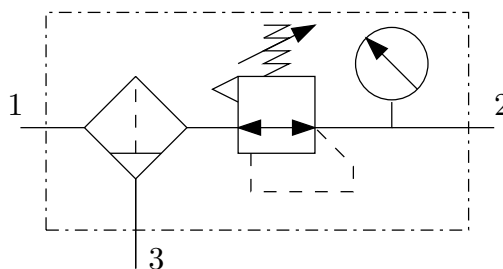


Figura 3.3 – Unidade de tratamento e regulagem de pressão

As informações técnicas da unidade de tratamento de ar estão apresentadas na Tabela 3.1.

Tabela 3.1 – Válvula reguladora e filtro

Grau de filtragem	40 $\mu m$
Volume máximo do condensado	22 $cm^3$
Indicador de pressão	manômetro
Faixa de regulação de pressão	0,05 - 1,2 MPa
Pressão de entrada 1	0,1 - 1,6 MPa
Histerese de pressão máxima	0,02 MPa
Vazão nominal padrão	1.400 l/min
Temperatura do meio	-10 - 60 °C
Temperatura ambiente	-10 - 60 °C
Peso	460 g

### 3.3.1.2 Vaso de Pressão

Conforme a Fig. 3.2 o vaso de pressão fica conectado entre a unidade de tratamento de ar e a servoválvula, funcionando como um filtro para oscilações na pressão de alimentação do sistema. Suas características estão descritas na Tabela 3.2.

Tabela 3.2 – Vaso de Pressão

Norma	NBR 8460
Pressão de Serviço	1.7 MPa
Massa	45 Kg
Volume	108 litros

### 3.3.2 Componentes do Atuador Pneumático

As descrições dos componentes do subsistema de atuação estão apresentados a seguir.

#### 3.3.2.1 Servoválvula Direcional

Foi utilizada uma válvula direcional proporcional Festo - MPYE-5-1/8-HF-010-B. Conforme esquematizado na Fig. 3.4, a conexão 1 é a entrada de ar comprimido e as conexões 2 e 4 são as saídas de ar balanceadas conforme o sinal de controle aplicado à entrada 6.

As principais características técnicas da servoválvula Festo - MPYE-5-1/8-HF-010-B estão relacionados na Tabela 3.3.

Nas conexões 3 e 5, onde o ar é exaurido para a atmosfera, foram utilizados silenciadores Festo - U-1/8-B (ver Tabela 3.4), diminuindo o ruído ambiental.

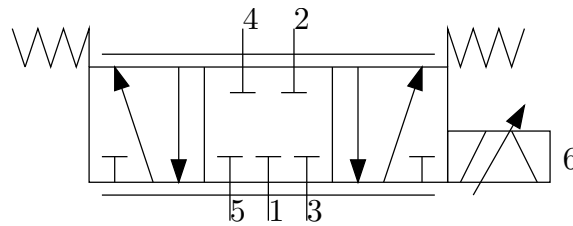


Figura 3.4 – Servoválvula controladora de vazão

Tabela 3.3 – Válvula direcional

Diâmetro nominal	6 mm
Tipo de acionamento	elétrico
Princípio construtivo	Válvula corrediça do êmbolo
Tipo de reposição	Mola magnética
Tipo de pilotagem	direto
Sentido do fluxo	não reversível
Pressão operacional	0 - 1 MPa
Vazão nominal padrão	700 l/min
Frequência limite	100 Hz
Histerese máxima	0,4 %
Tensão de serviço DC	17 - 30 V
Ondulação residual	5 %
Valores Teórico/Real	Tipo de tensão 0 - 10 V
Meio operacional	Grau de filtragem 5 $\mu$ m
Temperatura do meio	5 - 40 °C
Temperatura ambiente	0 - 50 °C
Peso	330 g

Tabela 3.4 – Silenciadores

Pressão de operação	0 - 1 MPa
Vasão para atmosfera	1,204 l/min
Nível de pressão sonora	74 dB(A)
Peso	8 g
Conexão	G1/8

### 3.3.2.2 Cilindro Atuador

Utiliza-se um cilindro Festo - DGP-25-1000-PPV-A-B. Este cilindro é contruído sem haste, com transmissão direta do movimento do êmbolo para o carro móvel através de um guia conectora. A vedação é realizada através de cintas magnéticas.

As características técnicas estão listadas na Tabela 3.5 e a representação de forças na Fig. 3.5.

Tabela 3.5 – Servoatuador

Curso	1 m
Diâmetro do êmbolo	25 mm
Amortecimento	Regulável em ambos os lados
Princípio de arraste	de união positiva (fenda)
Pressão operacional	0,2 - 0,8 MPa
Modo de operação	de dupla ação
Meio operacional	Ar comprimido filtrado
Temperatura ambiente	-10 - 60°C
Curso de amortecimento	18 mm
Massa móvel	180 g
Carga de flexão no eixo X (ver Fig. 3.5)	330 N
Carga de flexão no eixo Y (ver Fig. 3.5)	330 N
Momento longitudinal $M_y$ (ver Fig. 3.5)	20 Nm
Torque lateral $M_x$ (ver Fig. 3.5)	1 Nm
Torque de torção $M_z$ (ver Fig. 3.5)	3 Nm
Força teórica a 0.6 MPa, retorno	295 N
Força teórica a 0.6 MPa, avanço	295 N
Massa móvel com curso de 0 mm	180 g
Peso básico para curso de 0 mm	840 g
Fator de massa adicional por 10 mm de curso	36 g

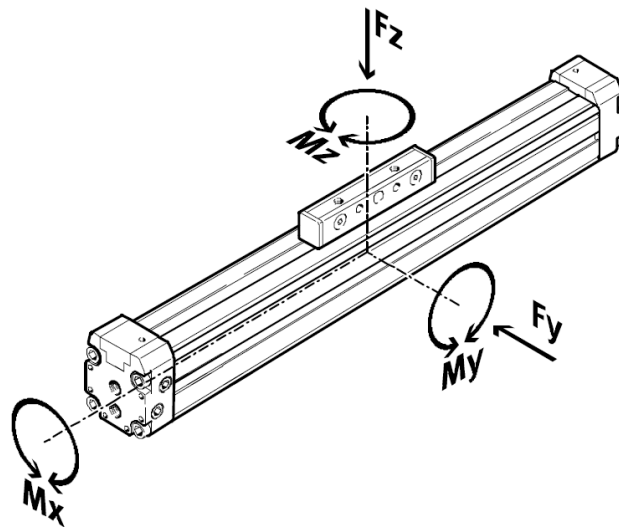


Figura 3.5 – Representação de forças no cilindro atuador (Festo Corporation, 2005)

### 3.3.3 Transdutores

#### 3.3.3.1 Fonte

A fonte de alimentação utilizada para os sensores de pressão, servoválvula e encoder incremental é uma fonte industrial chaveada de 24 V e 1.1 A (MeanWell, 2004). Maiores informações sobre a fonte chaveada estão na Tabela 3.6.

Tabela 3.6 – Fonte chaveada

Voltagem	24 V
Corrente	1.1 A
Potência	26.4 W
Ruído Máximo	100 mVp-p

### 3.3.3.2 Encoder Incremental

Para medições do deslocamento do êmbolo móvel do cilindro atuador do servoposicionador foi utilizado um *encoder* incremental Veeder-Root - BA 3022-1000. Este *encoder* foi aparafusado à bancada e, através de uma polia fixada a seu eixo, conectado a uma correia dentada, conforme a Fig. 3.6.

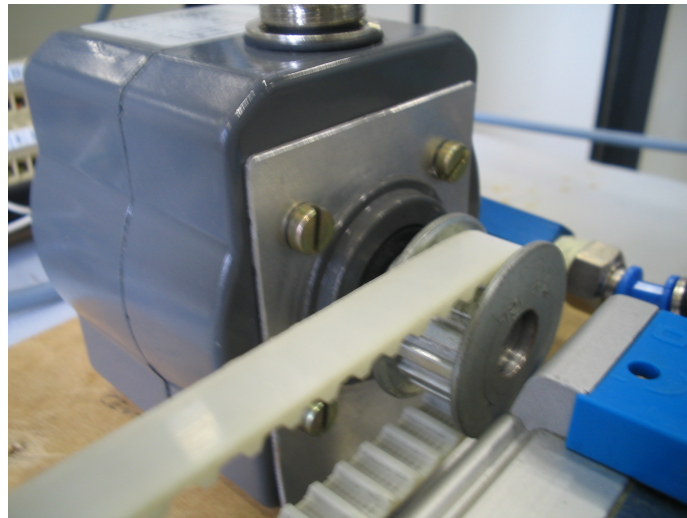


Figura 3.6 – Encoder incremental

A correia dentada foi acoplada ao carro móvel (ver Fig. 3.7) através de uma placa vazada que utiliza os dentes da correia para fixá-la.

Na Tabela 3.7 estão apresentadas informações sobre o encoder. O termo “*push-pull*” significa que a saída de sinal de tensão é equivalente ao nível de tensão do terra do circuito quando o sinal é nulo e igual à tensão de alimentação (24V) quando o sinal é positivo.

Tabela 3.7 – Encoder incremental

Tipo	Incremental
Diâmetro	10 mm
Modo de saída	“Push-Pull”
Pulsos por volta	1000
Canais	A e B

No outro extremo do curso do carro móvel foi utilizado um mancal para suspender e tensionar a correia dentada. As polias utilizadas no encoder e no mancal são polias dentadas de 32mm

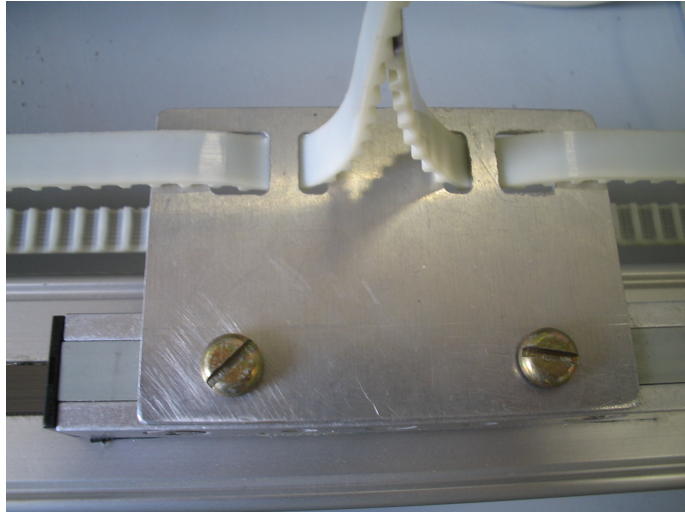


Figura 3.7 – Conexão do carro móvel do êmbolo do cilindro pneumático com a correia dentada

de diâmetro. Isto resulta em 12830 pulsos (de um único canal) emitidos pelo encoder durante o deslocamento ao longo do curso servoatuador (1m). Informação sobre a precisão do *encoder* no sistema de servoposicionamento estão apresentadas na Tabela 3.8.

Tabela 3.8 – Precisão do sistema

Diâmetro das Polias	32 mm
Total de Pulsos	12830 pulsos
Precisão do <i>Encoder</i>	0.078 mm

### 3.3.3.3 Sensores de Pressão

Para aquisição das pressões em ambas câmaras do cilindro atuador foram utilizados dois sensores de pressão relativa Huba Control - 510. Cada sensor está conectado a uma das saídas da servoválvula (ver conexões 2 e 4 na Fig. 3.4). As principais características técnicas sobre os sensores estão apresentadas na Tabela 3.9.

Tabela 3.9 – Sensores de Pressão

Escala	0 a 1 MPa
Calibração	Calibrado de fábrica
Sinal de saída	4 - 20 mA
Alimentação	11 - 33 VDC
Tipo de conexão	1/4 - 18 NPT

### 3.3.4 Sistema de Aquisição de Dados

Os componentes do subsistema de aquisição de dados estão descritos nas seções que seguem.

### 3.3.4.1 Driver do *Encoder*

Um sinal típico obtido através de uma medição realizada pelo *encoder* é ilustrado na Fig. 3.8, onde A e B são sinais quadráticos defasados  $90^\circ$ . Cada pulso emitido pelo sinal A denota um deslocamento da polia e, quando comparado ao sinal B, permite obter a direção do movimento. De acordo com a Fig. 3.8 é possível ter-se 4 diferentes configurações para os sinais A e B. A partir destas configurações e da seqüência em que elas ocorrem, é possível determinar a velocidade e a direção do movimento.

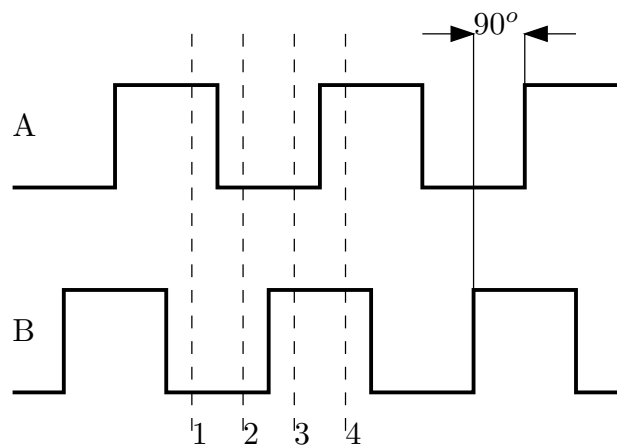


Figura 3.8 – Sinais do *encoder* incremental

Considerando as limitações do cilindro atuador, o servoposicionador deve funcionar a velocidades de até 1m/s. Por metro de deslocamento são adquiridos 12830 pulsos de *encoder*, logo é necessário adquirir os dados a uma frequência maior que 12kHz. Para realizar a decodificação dos sinais, é necessária uma frequência de aquisição de, no mínimo, 48kHz, para viabilizar a comparação entre os canais A e B e decodificar a posição e a direção do movimento do servoatuador.

Inicialmente foram realizados testes utilizando a porta paralela do microcomputador (ver Tabela 3.12) para decodificar a posição do servoatuador. Os resultados dos testes indicaram que a necessidade de um esforço computacional considerável para processar os sinais à frequência de 48kHz comprometia a eficiência do sistema, pois reduzia significativamente o número de instruções por ciclo de processamento disponível para a complementação dos algoritmos em tempo real. A solução foi implementar o *decoder* em um microcontrolador dedicado que transmite para o PC somente os resultados decodificados (posição calculada).

O *decoder* foi implementado em um microcontrolador PIC da Microchip, devido ao fácil acesso a literatura, programas computacionais para desenvolvimento e esquemas para circuitos de gravação.

Para utilizar este microcontrolador, foi necessário desenvolver dois circuitos eletrônicos: o cricuito gravador, responsável por copiar para a memória do microcontrolador o programa desen-

volvido no PC, e o circuito decodificador, responsável por receber os sinais do *encoder*, passá-los ao microcontrolador para decodificação e enviá-los para o PC.

A gravação do PIC foi feita por intermédio do *software* de distribuição gratuita WinPic (Buescher, 2005). O circuito de gravação utilizado foi o *Tait* (Tait's, 1999). O desenvolvimento dos algoritmos responsáveis por decodificação e comunicação foram escritos em linguagem C, utilizando o ambiente MPLAB IDE em conjunto com a biblioteca MPLAB C18, ambos *softwares* com versões gratuitas (Microchip, 2005).

Para tratar o sinal a ser decodificado foi necessário utilizar um circuito “ceifador”, responsável por reduzir o sinal de 24V (saída do *encoder*) para 5V (entrada do PIC). Este circuito é mostrado na Fig. 3.9. Após ser processado o “*corte*”, o integrado 74ACT14 reduz o ruído para o controlador através de um comparador *Schmitt Trigger*.

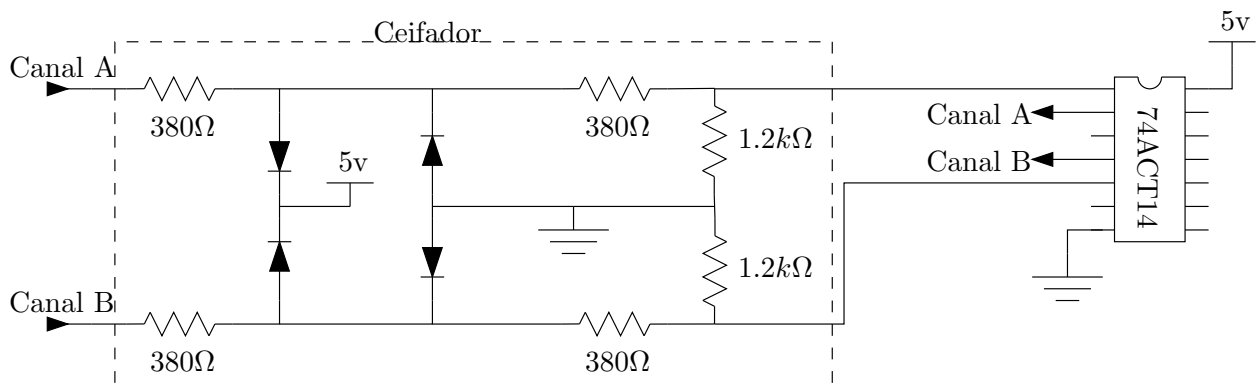


Figura 3.9 – Condicionador de sinais para o encoder

O circuito decodificador utilizado na bancada está esquematizado na Fig. 3.10. O microcontrolador utilizado foi um Microchip - PIC18F2550 funcionando a 24MHz. Maiores informações sobre o microcontrolador estão na Tabela 3.10.

Tabela 3.10 – Informações técnicas do PIC18F2550

Frequência Máxima	48MHz
Memória para Programa (Bytes)	32768
Memória para Programa (Instruções)	16384
Memória de Dados (Bytes)	2048
Memória de Dados EEPROM (Bytes)	256
Fontes de Interrupção	19
Temporizadores	4
Captura/Comparadores/Módulos PWM	2
<i>Universal Serial Bus</i> (USB)	1
Conversão Análogo/Digital (10 bits)	10

Para o microcontrolador realizar as operações de contagem dos pulsos enviados pelo encoder e comunicar-se com a central de processamento (PC), foram desenvolvidos dois algoritmos que são



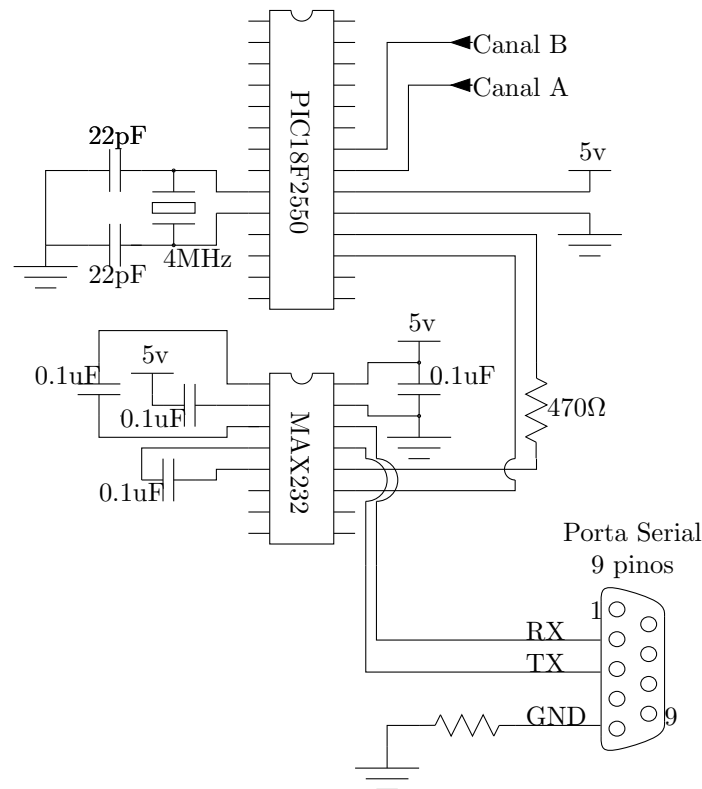


Figura 3.10 – Esquema do circuito do microcontrolador PIC18F2550

executados em paralelo pelo microcontrolador. O primeiro processo, esquematizado na Fig. 3.11, é responsável por contar os pulsos verificando a direção de movimento do servoatuador.

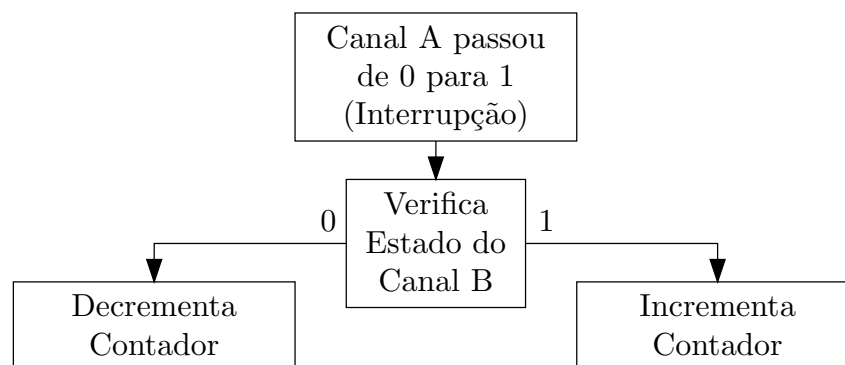


Figura 3.11 – Esquema do processo de contagem de pulsos (“Contador”)

Este processo é executado a partir de uma interrupção (na porta RB0) do microcontrolador, que está conectada ao canal A do encoder. Quando o sinal desta porta passa de 0 para 1 gera uma interrupção, “congelando” a execução de outros processos enquanto executa a contagem.

O outro processo é o de comunicação com o PC. Esta comunicação tem como propósito manter a central de controle (PC) informada da posição atual do servoatuador. Esse processo fica aguardando comandos vindos da porta serial (ver integrado MAX232, apresentado na Fig. 3.10) cada comando lido determina a execução de uma função no microcontrolador. Estes comando são enviados pela unidade de controle (PC).

Foram programados dois comandos: “zerar” e “ler”. O comando “zerar” (ver Fig. 3.12), reinicializa o contador de pulsos do microcontrolador PIC.

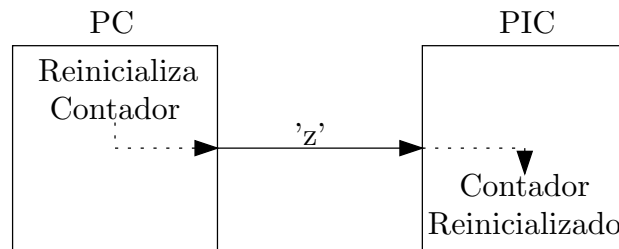


Figura 3.12 – Comando “zerar”

O comando “ler” (ver Fig. 3.13) faz com que o microcontrolador retorne o valor da posição armazenado no contador na forma de dois *bytes*.

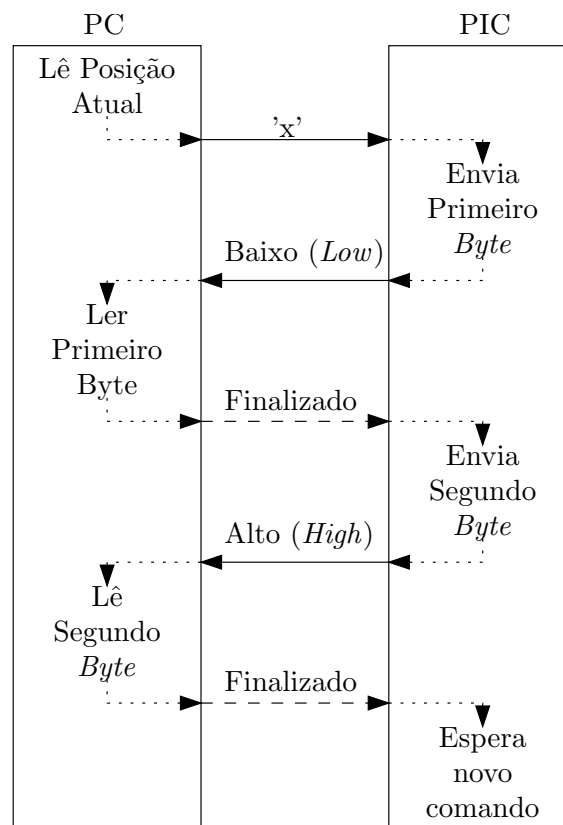


Figura 3.13 – Diagrama esquemático do comando “ler”

O algoritmo que executa os comandos “zerar” e “ler” pode ser visualizado na Fig. 3.14.

No bloco “Configuração do Sistema” são definidas a frequência de funcionamento do microcontrolador, a frequência de comunicação serial (no caso 115.2kBps) e a porta que irá receber a interrupção apresentada na Fig. 3.11. O código fonte utilizado para programar o PIC18F2550 está apresentado no Anexo A.1.

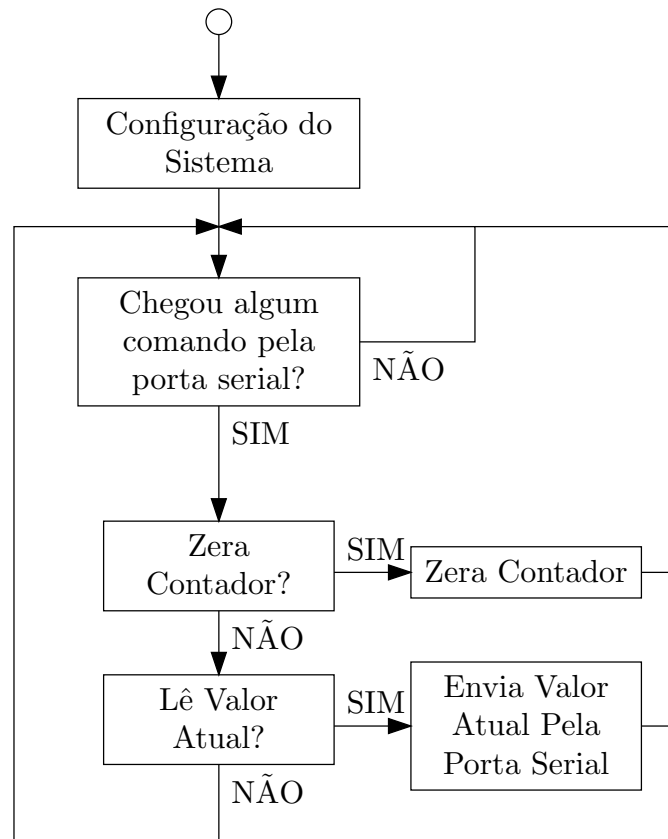


Figura 3.14 – Esquema do processo de comunicação com a unidade de controle

### 3.3.4.2 Conversão AD/DA

Para conversão do sinal de analógico para digital (para os sinais dos sensores de pressão) e digital para analógico (para o comando da servoválvula) foi utilizada uma placa CIO-DAS08-JR-AO. Seus detalhes técnicos estão especificados na Tabela 3.11.

Tabela 3.11 – Placa de aquisição de dados

Resolução A/D	12 bits
Número de Canais A/D	8 single-ended
Sinal de entrada	$\pm 5V$
Tempo de conversão A/D	$25 \mu s$
Resolução D/A	12 bits
Número de Canais D/A	2
Sinal de Saída	$\pm 5V$
Tempo de estabilização D/A	$10 \mu s$
Entradas Digitais	8
Saídas Digitais	8

Para conexão da placa de aquisição com os atuadores e sensores, foi utilizada uma borneira LR Informática Industrial - LRI-BORN37.

### 3.3.4.3 Condicionamento de Sinais

Os sensores de pressão utilizados (Huba Control - 510, ver Tabela 3.9) trabalham na faixa de 0 a 1 MPa com saída de sinal entre 4 a 20mA. Como a placa de aquisição funciona com entradas de  $\pm 5V$  foi necessário utilizar um circuito para realizar a conversão do sinal. A Fig. 3.15 mostra o esquema de um circuito comparador utilizado para calibração do sinal, visando aproximá-lo da faixa de aquisição da placa AD.

Observa-se que não foram inseridos filtros de sinais. Este assunto está apresentado no Capítulo 5.

É necessário também usar um circuito de condicionamento de sinais para o acionamento da servoválvula. A servoválvula recebe sinais de 0 a 10V e o circuito D/A da placa emite sinais de  $\pm 5V$ . Para conversão deste sinal utilizou-se um circuito comparador esquematizado na Fig. 3.16.

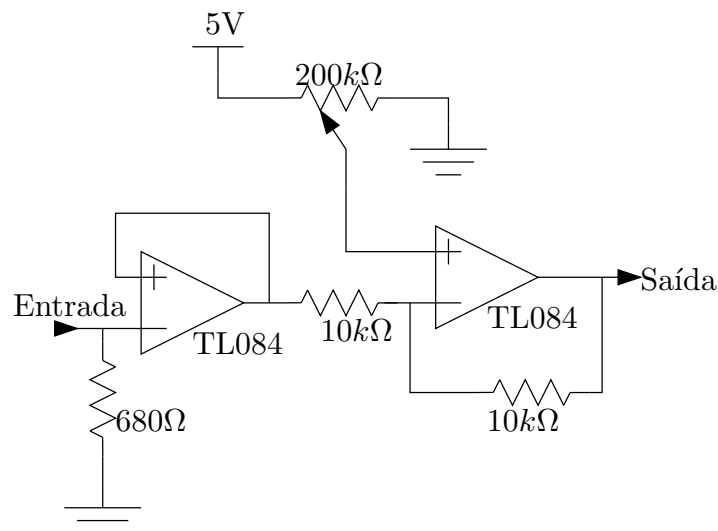


Figura 3.15 – Condicionamento de sinais para os sensores de pressão

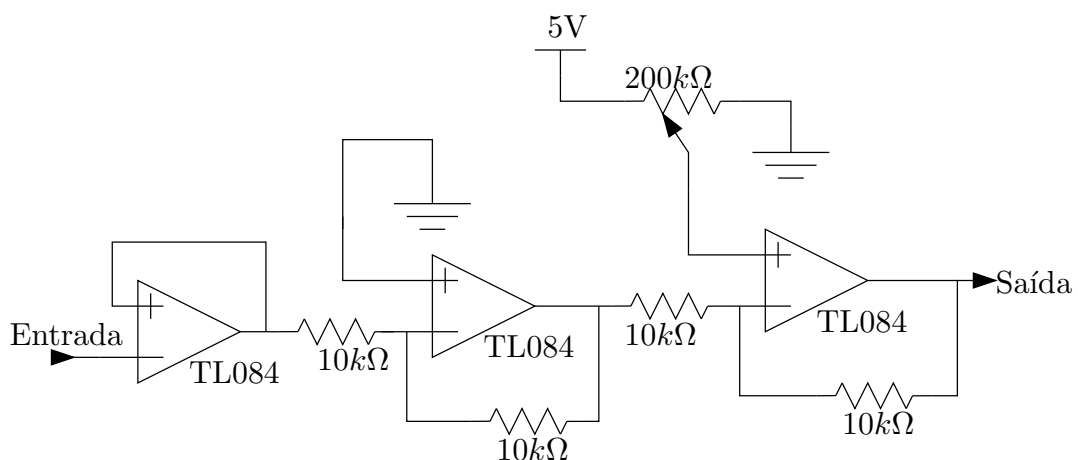


Figura 3.16 – Condicionamento de sinal do controle da servoválvula

Para ambos os circuitos mostrados nas Figuras 3.15 e 3.16, as fontes de 5 V e de  $\pm 12 V$

(alimentação dos amplificadores operacionais) foram tomadas na própria placa de aquisição de sinais.

#### 3.3.4.4 Microcomputador

O microcomputador é o componente responsável pelo processamento das informações adquiridas pelos sensores, o processamento do algoritmo de controle e o registro dos resultados desses processos. Maiores informações sobre o microcomputador estão na Tabela 3.12.

Tabela 3.12 – Informações técnicas do microcomputador

Frequência	133MHz
Memória RAM	64Mb
Disco Rígido	6.3Gb
Sistema Operacional	Gentoo/RTAI 3.2/Linux 2.6.9
Portas de Comunicação	Paralela e Serial

## 4 SISTEMA DE CONTROLE E AQUISIÇÃO DE DADOS

Neste capítulo são analisadas as funcionalidades necessárias para o controle de um servoposicionador, assim como as ferramentas utilizadas para modelagem e testes das diferentes técnicas de controle. São também apresentados os algoritmos desenvolvidos para o acionamento de servoválvula e atualização das variáveis medidas da planta pneumática.

### 4.1 Funcionalidades do Sistema de Controle

Sistemas para modelagem de controladores (RPC - *Rapid Control Prototyping*) necessitam basicamente de dois componentes: um *software* para projeto de sistemas de controle auxiliado por computador (CACSD - *Computer Aided Control System Design*) e um sistema de *hardware* dedicado processando um sistema operacional com suporte a tempo real (Bucher et al., 2004).

Sistemas de controle auxiliado por computador (CACSD) são sistemas constituídos por diferentes ferramentas que auxiliam desde o desenvolvimento dos algoritmos de controle até simulações em tempo real, com e sem interface com os dispositivos da planta (Bucher e Mannori, 2003).

A Fig. 4.1, ilustra as principais funcionalidades de um sistema para teste das técnicas de controle: o ambiente de desenvolvimento da técnica de controle, o ambiente de monitoramento e o sistema de aquisição de sinais e acionamento dos dispositivos servopneumáticos. Observa-se a existência de dois objetos externos ao sistema de controle. O usuário, que utiliza o sistema de controle e o sistema pneumático que é operado pelo sistema de controle.

O ambiente de desenvolvimento é constituído basicamente pela interface de programação utilizada pelo usuário para implementação das técnicas de controle (ver Capítulo 2).

O sistema de monitoramento consiste das ferramentas computacionais através das quais o usuário pode iniciar, acompanhar ou parar o funcionamento do sistema de servoposicionamento e, se necessário, alterar parâmetros da técnica de controle em execução.

O sistema de aquisição de dados e acionamento dos dispositivos pneumáticos não possui interface direta com o usuário, como pode ser visualizado na Fig. 4.1. Este sistema é responsável por manter a consistência das informações da planta pneumática, atualizando os processos atuantes no sistema de controle com as variáveis de controle.

As seções que seguem apresentam separadamente as descrições das ferramentas utilizadas e dos algoritmos desenvolvidos. Assim, na Seção 4.2 é realizada uma análise sobre as garantias temporais (sistemas de tempo real) e a solução desenvolvida para a aquisição e acionamento dos dispositivos da planta pneumática. Na Seção 4.3 são analisadas as soluções utilizadas para flexibi-

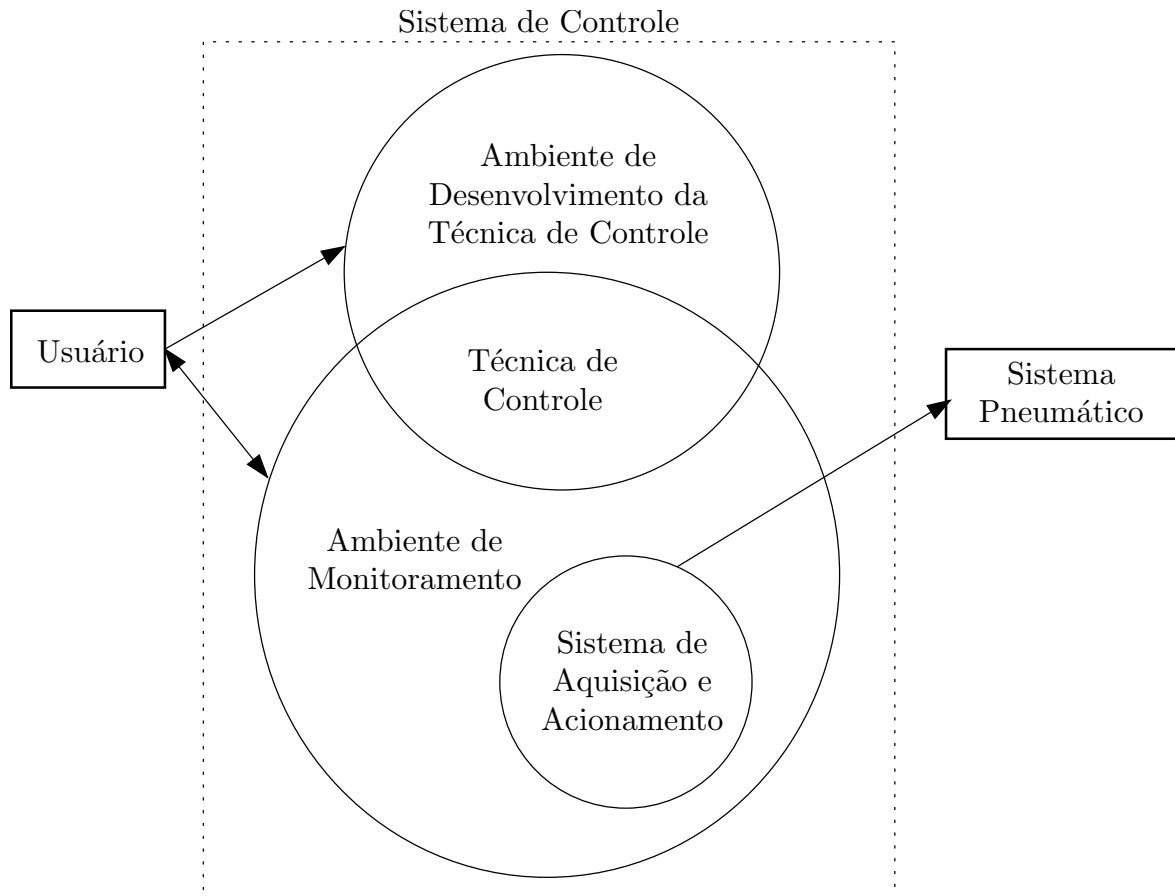


Figura 4.1 – Diagrama de funcionalidades do sistema de controle

lização da programação das técnicas de controle e monitoramento do sistema de controle.

## 4.2 Sistema de Aquisição e Acionamento

### 4.2.1 Sistemas de Tempo Real

Tanto o sistema de aquisição e acionamento quanto a técnica de controle necessitam de garantias temporais para a eficiência de seu funcionamento. Para possibilitar um controle temporal seguro é necessário que a implementação destes sistemas esteja baseada em um sistema de tempo real.

O termo *tempo real* é aplicado de diferentes formas, de acordo com a área de conhecimento onde é utilizado. Aqui utiliza-se no sentido consagrado na Ciência da Computação, onde é dividido em duas grandes áreas: tempo real “probabilístico” (*soft real-time*) e tempo real “determinístico” (*hard real-time*) (Dozio e Mantegazza, 2003).

Sistemas *soft real-time* são caracterizados pela habilidade do sistema de executar um processo garantindo a média do tempo determinado durante o desenvolvimento. Processos *soft real-time* geralmente são empregados em soluções onde variações, ou mesmo atrasos no tempo previsto,

ocorrem sem comprometimento do funcionamento do sistema. Dispositivos de vídeo geralmente utilizam *soft real-time*. O dispositivo de vídeo troca informações de modo que pequenas variações no desempenho ou mesmo a supressão de *frames* (imagens estáticas), são aceitáveis, pois são “quase” imperceptíveis ao olho humano. Em geral, as interfaces homem-máquina não necessitam de determinismo temporal acurado.

Sistemas *hard real-time* são caracterizados pela existência de garantias temporais mais rígidas, ou seja, maior grau de confiabilidade no sentido de que não serão ultrapassados os limites preestabelecidos durante o desenvolvimento do sistema. Assim, sistemas *hard real-time* não utilizam a média para compensar possíveis atrasos. Um exemplo de aplicação típica está no controle de plantas industriais.

Outra característica de sistemas *hard real-time* é a baixa latência, ou seja, o baixo tempo de atraso na resposta entre o comando de execução e a real execução do processo. O conceito de latência está ilustrado na Fig. 4.2.

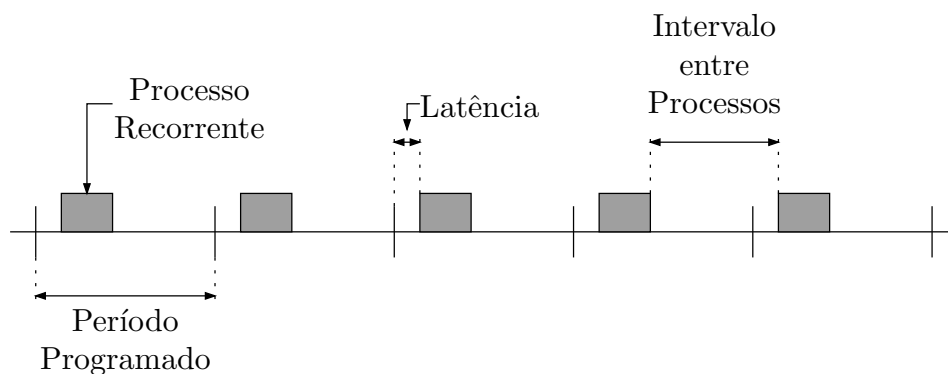


Figura 4.2 – Informações temporais sobre processos

Na Fig. 4.2, além da latência, pode-se observar outros conceitos fundamentais na análise de sistemas de tempo real. O “processo recorrente” é o conjunto de comandos que devem ser repetidos de acordo com o “período programado”. Observa-se que o processo recorrente não utiliza todo o tempo disponível para sua execução, o restante é utilizado por outros possíveis processos em execução no sistema, tais como interrupções, acesso a disco, etc.

Ao determinar o “período programado”, deve ser considerado, além da latência do sistema, assim como o tempo necessário para completar os comandos do processo recorrente, os possíveis atrasos e a possibilidade de falhas no sistema. O tratamento das possíveis falhas aumenta a integridade do sistema e pode ser realizado a partir de mensagens de alarme ou parada de dispositivos atuantes, evitando danos ao sistema controlado. As mensagens de falha são programadas para o caso em que o tempo de execução de um processo recorrente ultrapasse o tempo previsto.

Pode-se comparar sistemas *hard real-time* (determinísticos) e *soft real-time* (probabilísticos) através do gráfico da Fig. 4.3 (Sun Microsystems, 2004).



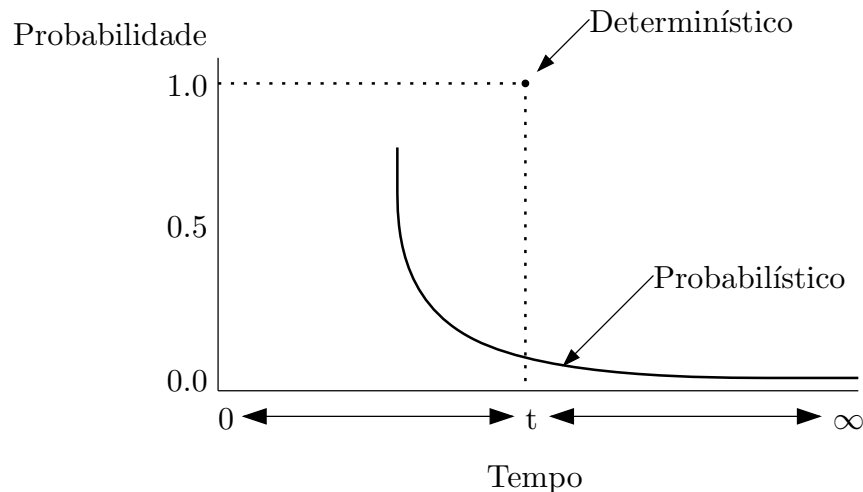


Figura 4.3 – Sistemas determinísticos *versus* probabilísticos

Geralmente, centrais de processamento, como estações de trabalho-*workstations*, não são projetados para aplicações *hard real-time*. Isto ocorre devido à falta de disponibilidade de informações sobre as latências do *hardware* utilizado. Assim, grande parte dos controladores com requisitos de tempo real são implementados em sistemas dedicados, como processadores digitais de sinais (*DSPs - Digital Signal Processors*) e microcontroladores, os quais possuem maiores garantias quanto a latência devido à baixa quantidade de instruções utilizadas para processar as interrupções (Dozio e Mantegazza, 2003).

Neste trabalho estuda-se o emprego de microcomputadores compatíveis com IBM-PC em controle de tempo real.

Os maiores benefícios da utilização de microcomputadores para o controle digital está na flexibilidade e na grande capacidade de processamento, chegando a milhões de operações de ponto flutuante por segundo (*FLOPS - Floating-Point Operations per Second*), tipo de instrução que aumenta significativamente a eficiência no processamento matemático. Esta capacidade possibilita a rápida resposta do ciclo de controle, mesmo com o uso de técnicas de controle complexas. Acrescenta-se também a possibilidade de utilizar sistemas de alto desempenho, como *Symmetric Multi Processors (SMP)*, com arquitetura de *hardware* com mais de um processador na mesma unidade de processamento, resultando em uma melhor relação entre custo e capacidade de processamento.

Atualmente, devido ao rápido desenvolvimento tecnológico (tanto em *software* quanto em *hardware*), o acesso a microcomputadores está muito facilitado. Devido à grande disponibilidade de microcomputadores no mercado alguns projetos vêm sendo desenvolvidos para a construção de sistemas operacionais com suporte a *hard real-time* em computadores pessoais. Muitos destes sistemas, como será visto nas seções que seguem, funcionam em equipamentos já considerados obsoletos, diminuindo os custos para o desenvolvimento de soluções para controle em tempo real.

Além do suporte a tempo real, foi dada prioridade na utilização de tecnologias abertas ou *Open Source*. Estas soluções possuem como características:

- livre utilização e distribuição (gratuito);
- permite acesso também livre às tecnologias utilizadas (código fonte);
- permite a alteração do código fonte.

A prioridade na utilização de tecnologias *Open Source* na arquitetura de controle do servoposicionador tem como propósito evitar a inserção de tecnologias que possam restringir a utilização ou criar dependência financeira ou mesmo tecnológica. Assim, a solução para controle em tempo real foi desenvolvida com base em sistemas abertos, desde o sistema operacional até as interfaces com o usuário.

#### 4.2.2 Sistema Operacional

O GNU/Linux é um sistema operacional tipo UNIX, largamente utilizado em soluções onde a flexibilidade é fundamental. Por ser um sistema de código aberto e de livre utilização, permite alterações em seu código fonte, possibilitando assim o redimensionamento do sistema de acordo com as necessidades da aplicação. GNU/Linux é facilmente customizado e largamente utilizado em dispositivos embarcados, como, por exemplo, relógios, computadores de mão e placas PC-104 (Kunz et al., 2003). Placas PC-104 são dispositivos compatíveis com IBM-PCs, porém com tamanho reduzido.

Encontra-se no mercado várias soluções de sistemas operacionais com suporte a tempo real (Tornado/VxWorks, Windows CE, QNX, RTAI, pSOS, VRTX dentre outros). Porém, avaliando as necessidades descritas anteriormente optou-se pelo sistema GNU/Linux em conjunto com RTAI (*Real Time Application Interface*) (DIAPM, 2005).

O sistema operacional GNU/Linux não possui suporte (nativo) a tarefas de tempo real. Porém, utilizando o sistema RTAI, este passa a ser um sistema operacional de tempo real ou RTOS (*Real Time Operation System*). O RTAI é um projeto iniciado pelo "Departimento di Ingegneria Aerospaziale del Politecnico di Milano"(DIAPM) em 1996 (DIAPM, 2005). Este usa o projeto ADEOS (*Adaptive Domain Environment for Operating Systems*), para realizar modificações no núcleo (*kernel*) do sistema operacional GNU/Linux, acrescentando o suporte de tempo real aos processos na utilização dos módulos (bibliotecas) disponibilizados pelo projeto RTAI.

O sistema operacional GNU/Linux, modificado pelo projeto RTAI, permite o desenvolvimento de quatro tipos diferentes de tarefas (Lages e Alt, 2003):

- Processos Linux: são os processos comuns do Linux que executam apenas quando não há tarefa de tempo real em operação.
- Tarefas de tempo real no *kernel*: são tarefas de tempo real que executam no espaço do *kernel*, possuindo a menor latência possível. Este tipo de tarefa não pode acessar a maioria dos serviços do *kernel* do Linux ou de bibliotecas, como a biblioteca padrão da linguagem C.
- Tarefas LXRT (*Linux RealTime*) *soft real-time*: são tarefas que executam no modo do usuário sem sofrer preempção dos processos do Linux, mas podem sofrer preempção de interrupções do Linux e das demais tarefas de tempo real. A vantagem deste tipo de tarefa está no acesso aos serviços do *kernel* do Linux e de bibliotecas.
- Tarefas LXRT *hard real-time*: são tarefas que executam no modo do usuário sem sofrer preempção de processos ou de interrupções do Linux, podendo sofrer preempção apenas de tarefas e interrupções de tempo real com prioridade mais alta. O acesso a serviços do *kernel* do Linux são permitidos através de um mecanismo de chaveamento implícito para o modo *soft real-time*, com recuperação do modo *hard real-time* após o retorno da chamada ao *kernel*.

Tarefas LXRT *hard real-time*, são mais aconselhadas pois não necessitam de conhecimentos específicos sobre o *kernel* do Linux, resultando em uma maior facilidade de desenvolvimento e mantendo as características de tempo real necessárias para o controle.

Para o acesso à placa AD/DA foi utilizado o projeto Comedi (*Control and Measurement Interface*) (Schleef e Hess, 2005). Este projeto oferece a biblioteca *Kcomedilib*, a qual permite o acesso ao *hardware* de conversão (placa AD/DA) a partir dos processos de tempo real. Esta ferramenta possui *drivers* para dezenas de placas de conversão AD/DA, inclusive para placa utilizada neste projeto (CIO-DAS08-JR-AO). Assim como os projetos RTAI e GNU/Linux, o projeto Comedi também é *Open Source*.

A comunicação com o *decoder* (responsável por disponibilizar a posição do êmbolo do servo-atuador) foi realizado utilizando o suporte nativo do sistema operacional GNU/Linux para acesso à porta serial do microcomputador.

Com as ferramentas descritas acima é possível projetar algoritmos de controle e testá-los no sistema de servoposicionamento pneumático. Porém, para isto, é necessário que o projetista possua conhecimentos significativos sobre programação C e o sistema operacional GNU/Linux, visto que as mesmas não possuem interface gráfica para o desenvolvimento dos algoritmos.

### 4.2.3 Algoritmos de Controle dos Dispositivos

O “servidor de dados”, processo responsável pela comunicação entre os dispositivos da bancada e o ciclo de controle, está representado na Fig. 4.4. Este processo foi implementado em linguagem C e utiliza tarefas LXRT com *hard real-time*.

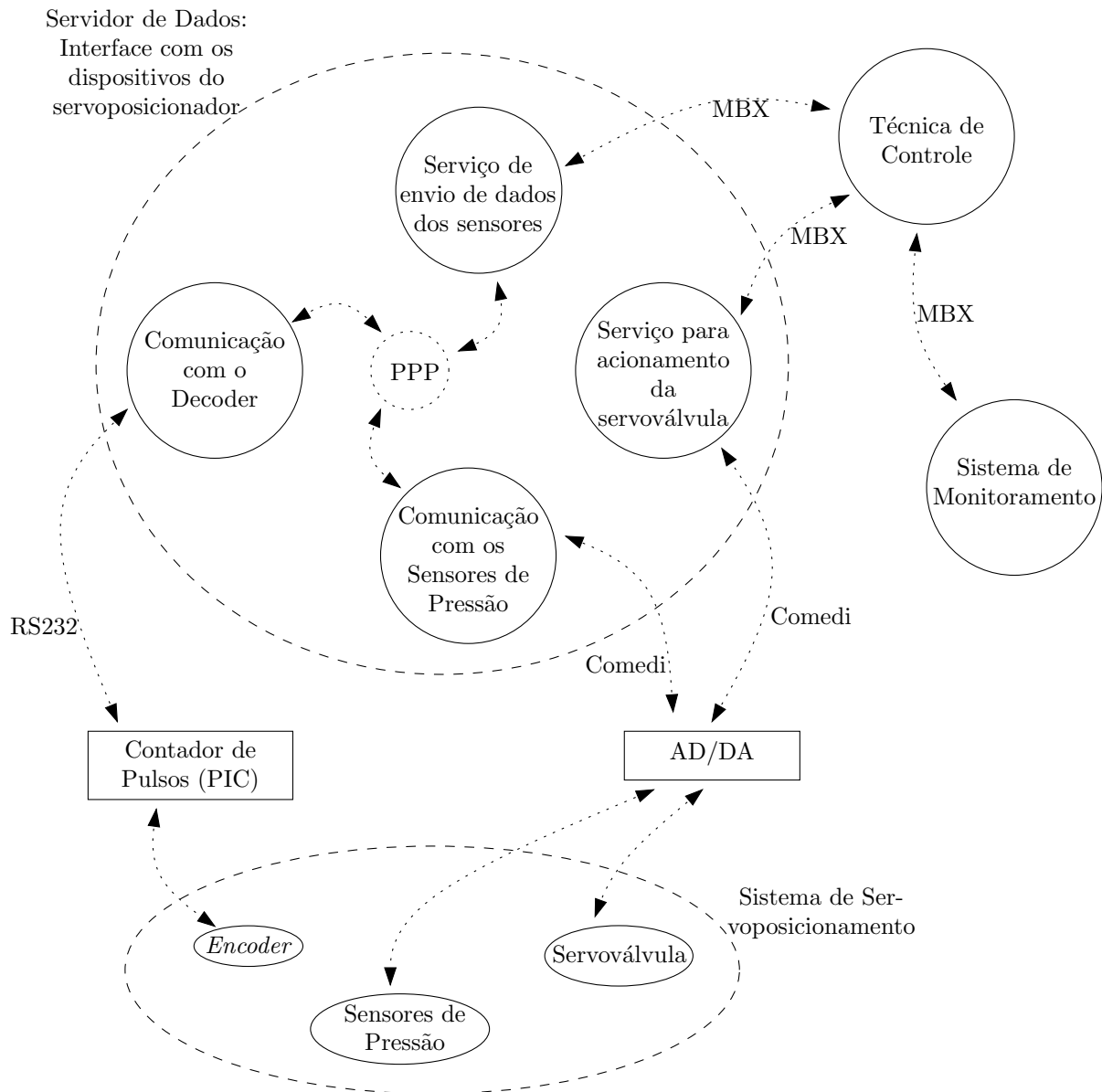


Figura 4.4 – Visão geral do sistema

O processo servidor, que consiste na interface entre o ciclo de controle e a planta pneumática, é responsável por manter o ciclo atualizado com as variações dos sensores e receber novos comandos para o acionamento da servoválvula.

A troca de informações entre o processo responsável pelo ciclo de controle e o processo servidor é realizado por meio de mensagens. As mensagens são enviadas utilizando a biblioteca *Mailboxes* (MBX) do projeto RTAI. Este método possibilita a troca de mensagem entre múlti-

plos receptores e emissores, inclusive através de uma rede (LAN). Isto permite que o processo responsável pelo ciclo de controle e o processo servidor dos dados da planta sejam executados em microcomputadores diferentes, conectados através de uma rede de comunicação, como por exemplo uma rede *ethernet*.

A comunicação entre o sistema de monitoramento e o processo responsável pelo ciclo de controle também pode ser realizado por rede. A comunicação por rede pode aumentar o tempo necessário para comunicação entre processos, porém, dilui a carga de processamento através da sua distribuição entre diferentes microprocessadores. Isto pode reduzir os custos em *hardware* pois assim estes microprocessadores necessitam menor poder de processamento isoladamente.

Observa-se na Fig. 4.4 que o servidor de dados possui quatro sub-processos (*threads*) sendo executados em paralelo: comunicação com o *decoder*, comunicação com os sensores de pressão, serviço de envio de dados dos sensores e serviço para acionamento da servoválvula. Cada um destes sub-processos foi programado com um período de *1ms*.

O círculo nomeado de “PPP” (valores da Pressão A, da Pressão B e da Posição), representa um recurso compartilhado, ou seja, um bloco de informações (memória) que é lido ou escrito por diferentes *threads*. Para sincronizar estas leituras e escritas, evitando conflito entre os *threads* em execução, foi utilizada a técnica de semáforos, a qual restringe o processo de forma que somente um *thread* acesse o bloco compartilhado por vez, tanto para leitura quanto para escrita. Se um dos *threads* está lendo ou escrevendo no recurso “PPP”, os outros ficam esperando a liberação em uma fila de acesso ao recurso. Devido à necessidade de um tempo para espera de liberação do recurso “PPP”, foi reservado 10% (0.1ms) do tempo total de ciclo nos *threads* que utilizam esta estrutura. Este tempo de espera é ilustrado na Fig. 4.5 pelo bloco “PPP”.

O *thread* de comunicação com o *decoder* realiza o envio e o recebimento de dados através da porta serial do microcomputador comunicando-se com o microcontrolador PIC que executa a ação de *decoder* (ver. Capítulo 3), assim mantendo atualizada a estrutura “PPP” com o valor atual da posição do servoposicionador.

O *thread* de comunicação com os sensores de pressão realiza a leitura de dois canais de conversão AD da placa CIO-DAS08 (ver Tabela 3.11). Cada canal corresponde a um dos sensores de pressão conectados às câmaras do servoatuador. Após a leitura são atualizados os valores de pressão na estrutura “PPP”.

O *thread* de serviço de envio dos dados dos sensores realiza a leitura periódica da estrutura “PPP”, permanecendo em modo de espera até a solicitação de atualização pelo processo responsável pelo ciclo de controle. Após uma solicitação de leitura ou quando o tempo de espera ultrapassa 0.9ms, este *thread* cancela a espera e reinicia o ciclo de atualização.

O *thread* de serviço para acionamento da servoválvula inicia em modo de espera até receber uma solicitação de escrita na servoválvula. A cada solicitação, imediatamente é realizado o acionamento na servoválvula por intermédio de uma conversão DA realizada por um dos canais da placa CIO-DAS08 (ver Tabela 3.11). O tempo programado de 1ms para este *thread* evita que o sistema permaneça em modo de espera após solicitação de finalização pelo usuário.

A comunicação entre os *threads* de serviço e o ciclo de controle pode ser visualizado pela Fig. 4.5.

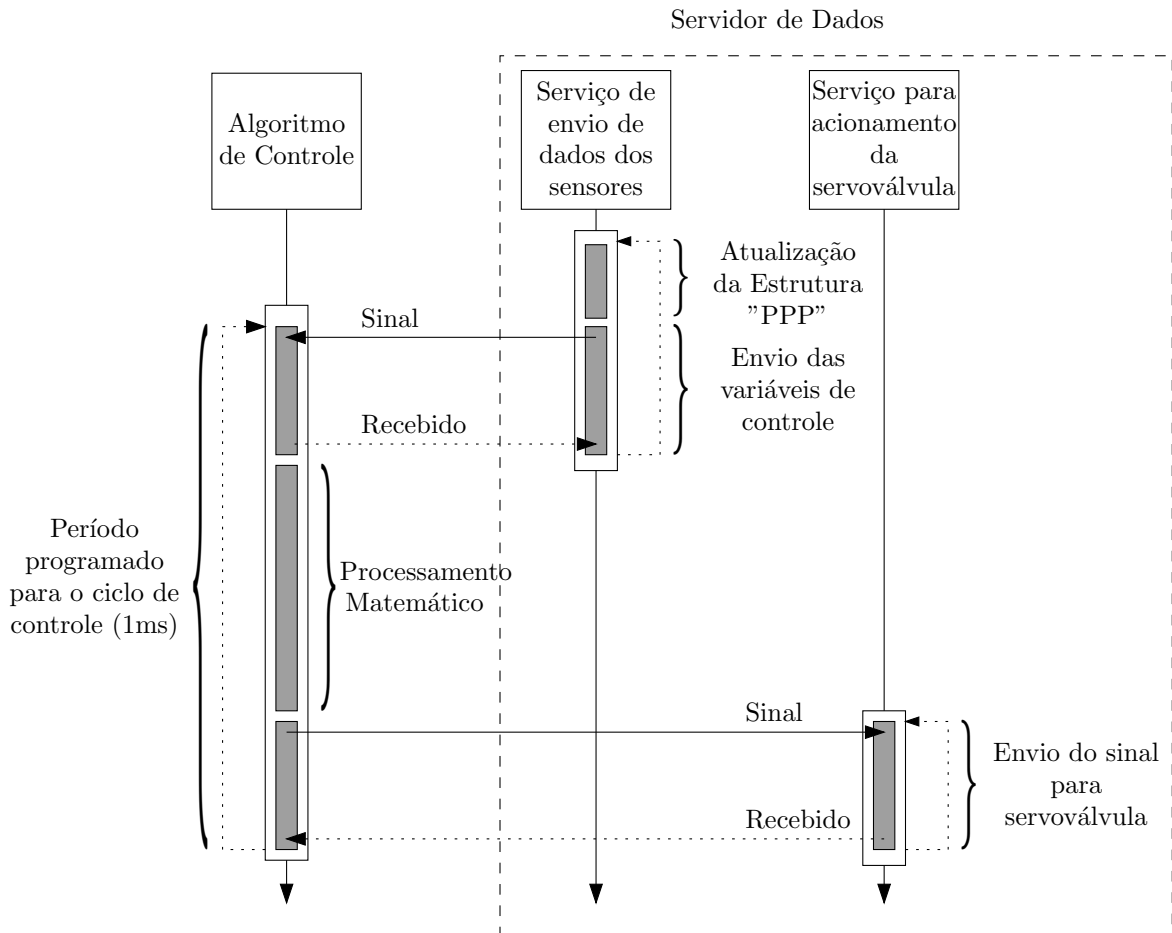


Figura 4.5 – Sistema de controle e acionamento

Os dois *threads* de comunicação entre o servidor e o serviço de gerenciamento do ciclo de controle estão apresentados em paralelo. O bloco "PPP" (localizado no *thread* de serviço de envio de dados dos sensores) representa a espera de liberação do semáforo para a leitura da estrutura "PPP". Observa-se que o processo responsável pelo ciclo de controle realiza, sincronizadamente, dois acessos ao sistema servidor: o primeiro de leitura dos sensores da bancada e o segundo de envio do sinal para acionamento da servoválvula.

O servidor atualiza os dados dos sensores a cada 1ms, porém não há necessidade do controlador possuir o mesmo período (1ms). Isto é realizado pelo usuário durante a modelagem do controlador.

Na Fig. 4.5 foi exemplificado um ciclo programado de 1ms, ou seja, os acessos (envio e recebimento de mensagens) somados ao processamento matemático da técnica de controle possuem período máximo de execução de 1ms.

### 4.3 Ambiente de desenvolvimento e monitoramento

Suites utilizadas para modelagem de controladores como Matlab/Simulink (MathWorks, 2005) e Scilab/Scicos (INRIA, 2005b), além de permitirem a programação em linguagem de blocos, auxiliam a modelagem de controladores através da disponibilização de bibliotecas para tratamento de sinais que podem ser facilmente inseridas no controlador.

RTAI-Lab (DIAPM, 2005) é um projeto de código aberto que promove a integração entre o suporte de tempo real do sistema operacional com sistemas CACSD. O RTAI-Lab suporta tanto a suite comercial MATLAB/Simulink/RealTime-Workshop quanto a suite de código aberto SCILAB/Scicos. Visando a utilização de *software* livre, emprega-se neste trabalho a suite SCILAB/Scicos.

#### 4.3.1 Scilab/Scicos

*Scilab* é um ambiente voltado para o desenvolvimento de *software* para resolução de problemas numéricos, criado em 1990 por um grupo de pesquisadores do INRIA (Institut de Recherche en Informatique et en Automatique) e do ENPC (Ecole Nationale des Ponts et Chaussées).

Esta ferramenta pode ser utilizada para criar controladores com suporte a tempo real, gerando o código fonte e compilado diretamente do ambiente gráfico de programação em blocos (Scicos) (INRIA, 2005a). Um exemplo de programação em blocos no ambiente Scicos pode ser visualizado na Fig. 4.6. Através desta interface o usuário pode modelar o sistema de controle. O resultado da compilação do sistema modelado é um código executável nativo para GNU/Linux, incluindo o suporte a tempo real. Este programa é implementado utilizando LXRT *hard real-time*, o qual opera em espaço de usuário.

No âmbito do presente trabalho, foi desenvolvido um bloco de programação para o sistema Scicos, facilitando a programação da comunicação com o serviço de acesso às variáveis do sistema pneumático. Logo, o usuário não necessita conhecer detalhes da implementação em *hardware* da arquitetura controladora. Este bloco possui uma entrada (sinal para a servoválvula) e três saídas (pressões nas câmaras e posição do servoposicionador) (ver Fig. A.1 do Anexo A.3). Após compilado diretamente através do diagrama de blocos no ambiente Scicos, o executável pode ser enviado para o sistema hospedeiro. O esquema do processo de compilação e envio ao sistema hospedeiro pode ser visualizado na Fig. 4.7.

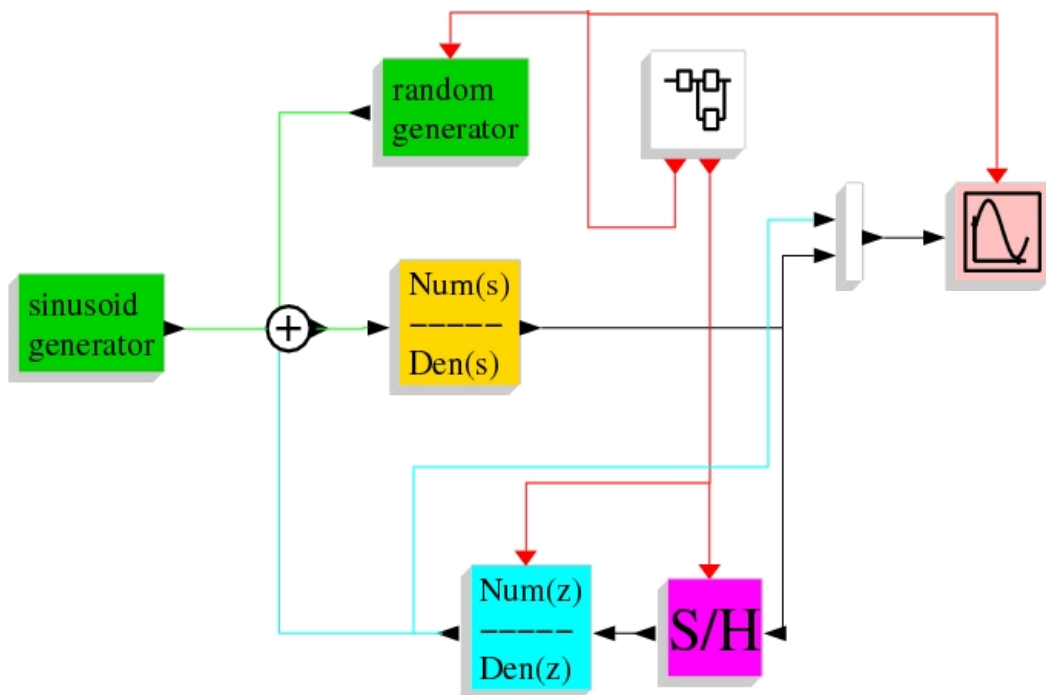


Figura 4.6 – Exemplo de implementação de programa de controle em ambiente Scilab/Scicos.

A partir da compilação do programa e execução pelo sistema Scicos, o seu monitoramento pode ser realizado através do programa *XRtaiLab*, o qual faz parte do projeto RTAI. Este programa, permite o desenvolvimento de telas de interface com o usuário para facilitar o acompanhamento do processo, assim como sua execução e representação gráfica das variáveis. Veja exemplo de tela do programa *XRtaiLab* na Fig. 4.8.

Como em toda interface de acesso direto pelo usuário, esta deve ser de fácil utilização e manutenção, promovendo acesso a diferentes tipos de visualização dos dados. Mesmo sendo o sistema de monitoramento uma interface com o usuário (não tempo real), são necessárias garantias de que

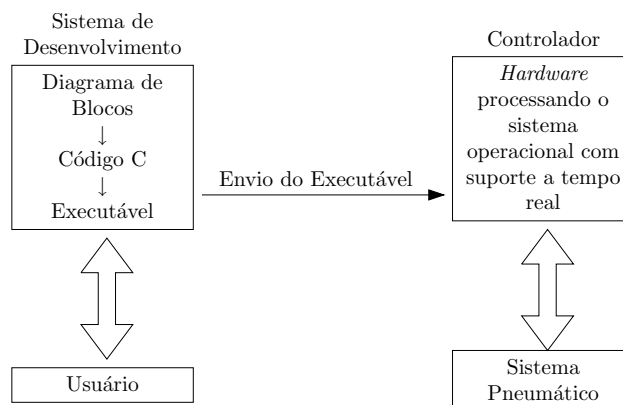


Figura 4.7 – Sistema de desenvolvimento



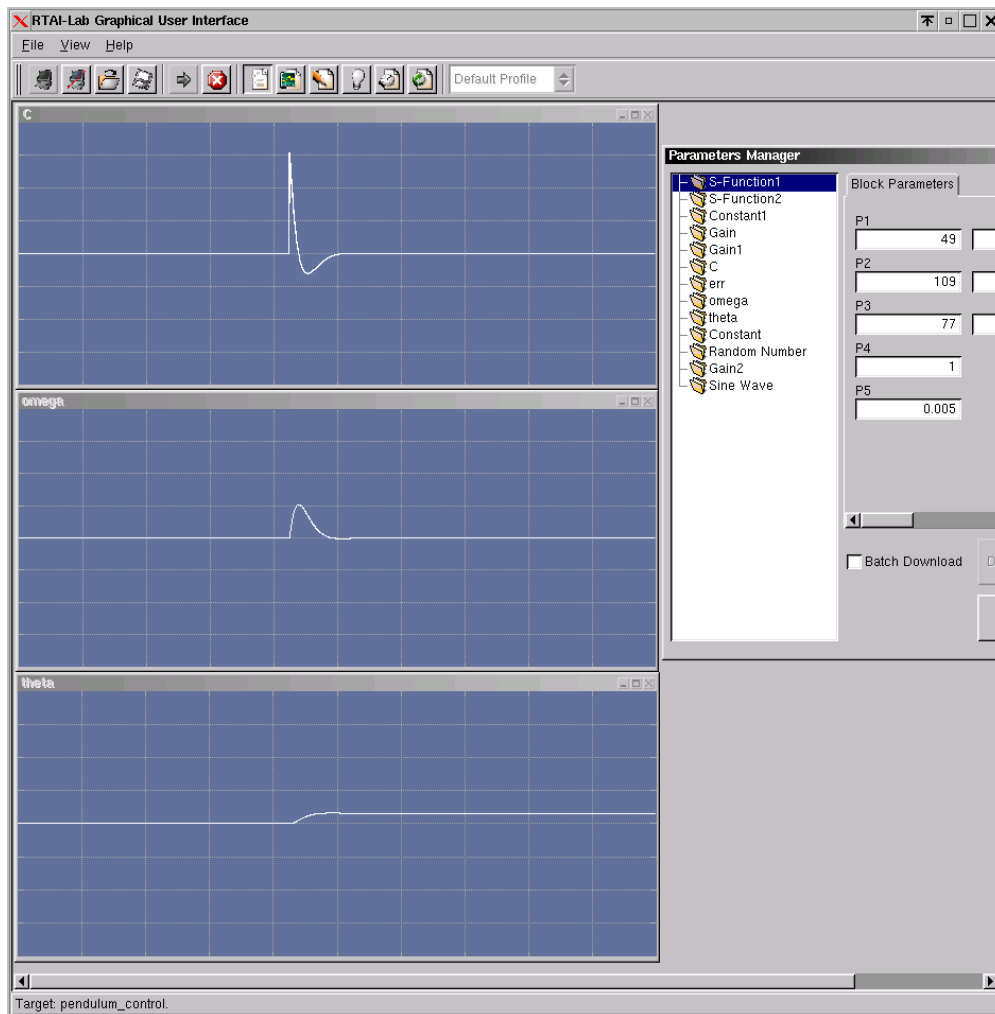


Figura 4.8 – Sistema de Monitoramento

os dados apresentados sejam confiáveis e precisos com relação ao tempo. A comunicação com o controlador é implementada em *soft real-time*. Como ocorre na comunicação entre o ciclo de controle e o sistema de aquisição e acionamento da planta pneumática, o sistema de monitoramento pode comunicar-se via rede com o programa gerenciador do ciclo de controle.

## 5 DETERMINAÇÃO DOS PARÂMETROS DO SISTEMA

Neste capítulo são apresentados os resultados e os procedimentos para obtenção das curvas características de vazão mássica e atrito do sistema de servoposicionamento servopneumático. Também são apresentados os resultados da eficiência sob o aspecto de temporização do sistema.

### 5.1 Caracterização do Sistema

Seja o servoposicionador pneumático esquematizado na Fig. 1.2, as equações 5.1 e 5.2 apresentadas por Perondi (2002) constituem o modelo dinâmico do sistema pneumático de posicionamento e definem uma dinâmica não-linear de 4ª ordem.

$$M\ddot{y} + F_a + F_e = Ap_\Delta \quad (5.1)$$

onde  $M$  é a massa deslocada,  $F_a$  é a força de atrito,  $F_e$  é a força externa,  $p_\Delta$  é a diferença entre as pressões nas câmaras,  $A$  é a área do êmbolo e  $\ddot{y}$  é a aceleração do êmbolo.

$$\dot{p}_\Delta = RrT \left[ \frac{q_{m1}(p_1, u)}{Ay + V_1} - \frac{q_{m2}(p_2, u)}{A(L - y) + V_2} \right] \quad (5.2)$$

onde  $\dot{p}_\Delta$  é a derivada da diferença entre as pressões nas câmaras,  $R$  é a constante universal dos gases,  $r$  é a razão dos calores específicos do ar,  $T$  é a temperatura,  $q_{m1}(p_1, u)$  é a função de vazão mássica na câmara 1 dependente da pressão na câmara 1  $p_1$  e do sinal de tensão  $u$ ,  $q_{m2}(p_2, u)$  é a função de vazão mássica na câmara 2 dependente da pressão na câmara 2  $p_2$  e do sinal de tensão  $u$ ,  $A$  é a área do êmbolo,  $y$  é o deslocamento do êmbolo,  $L$  é o curso útil do cilindro atuador,  $V_1$  é o volume *morto* na linha de conexão 1 e  $V_2$  o volume *morto* na linha de conexão 2.

Observa-se que se os parâmetros do sistema forem conhecidos, esta dinâmica pode ser alterada através da variação controlada das vazões mássicas  $q_{m1}$  e  $q_{m2}$ . Estas vazões dependem das pressões nas câmaras e das seções de passagem do ar que podem, por sua vez, ser controladas pelo valor da tensão  $u$  aplicada à servoválvula.

Logo, o controle da dinâmica do sistema depende do monitoramento de seus estados e do conhecimento dos seus parâmetros, incluindo as relações entre as vazões mássicas na servoválvula com as pressões internas nas câmaras concomitantemente com as relações entre as vazões mássicas na servoválvula com o sinal de controle  $u$ .

### 5.1.1 Vazão Mássica de Fluidos Compressíveis Através de Orifícios

O estudo das vazões mássicas na servoválvula pode ser abordado utilizando a teoria da mecânica dos fluidos que trata sobre o escoamento de fluidos compressíveis através de orifícios, como já realizado por Bobrow e McDonell (1998), Perondi e Guenther (1999b) e Mare et al. (2000).

Assumem-se as hipóteses de que o processo de compressão do volume do ar devido ao estrangulamento na entrada do orifício de passagem é adiabático, reversível e realiza-se a uma velocidade bastante alta, caracterizando um comportamento isentrópico. A Fig. 5.1 exemplifica o processo de estrangulamento de um fluido compressível.

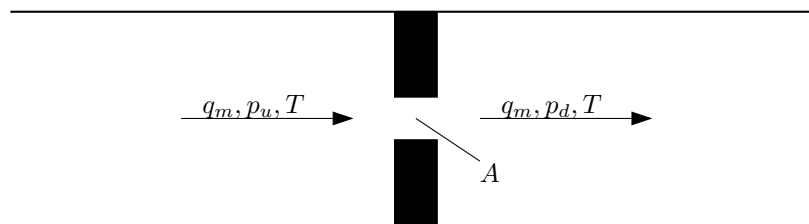


Figura 5.1 – Estrangulamento na passagem do fluido

As hipóteses de velocidade constante, escoamento unidirecional e condição estática a montante para o gás são assumidas. Nestas hipóteses ocorrem dois modos de vazão: sônica ou subsônica. O regime de vazão depende da relação  $p_d/p_u$  entre as pressões jusante ( $p_d$ ) e montante ( $p_u$ ). A vazão mássica assume comportamento subsônico para  $p_d/p_u < p_{crit}$ , onde  $p_{crit}$  é definido como a relação de pressões crítica que caracteriza a transição de regime de escoamento saturado para subsônico.

A vazão em regime subsônico é definida por (Perondi, 2002):

$$q_m = A_0 p_u \sqrt{\frac{2r}{(r-1)RT} \left[ \left( \frac{p_d}{p_u} \right)^{2/r} - \left( \frac{p_d}{p_u} \right)^{r+1/r} \right]} \quad (5.3)$$

De acordo com Mare et al. (2000), nos casos em que  $p_d/p_u \leq p_{crit}$ , a vazão mássica entra em regime saturado (ou sônico), e apresenta valor máximo e fixo dado por:

$$q_m = A_0 p_u \sqrt{\frac{r}{RT} \left( \frac{2}{r+1} \right)^{\frac{r+1}{r}}} \quad (5.4)$$

A Fig. 5.2 apresenta o comportamento teórico da vazão mássica em função da razão  $p_d/p_u$ , com  $p_{crit}$  constante. Segundo Martin (1995), a relação crítica de pressões em que a transição ocorre é dada por  $p_{crit} = (2/(r+1))^{r/(r+1)}$ . No caso do ar, o referido valor crítico é dado por 0,528 (Nouri et al., 2000).

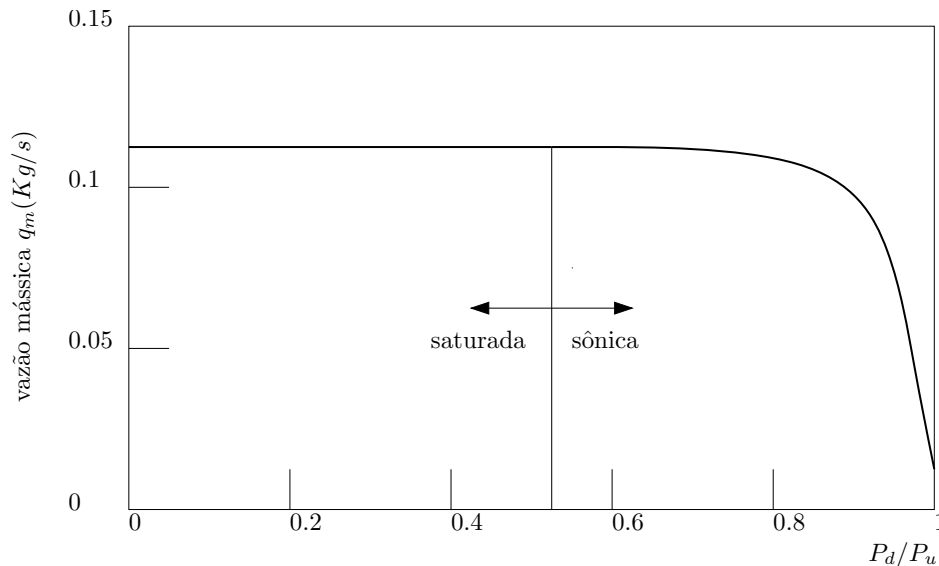


Figura 5.2 – Vazão Mássica Teórica

### 5.1.2 Identificação das Curvas de Vazão Mássica

De acordo com o projeto construtivo da servoválvula, variando a tensão de entrada  $u$  ocorre a alteração na área do orifício de passagem de ar ( $A_0$ ). Logo, a vazão mássica teórica pode ser dada por:

$$q_m = A_0(u) p_u \Psi(p_d/p_u) \quad (5.5)$$

onde a função  $\Psi(p_d/p_u)$  tem um valor constante para  $p_d/p_u \leq 0,528$  quando o escoamento é saturado, e um valor dependente da relação de pressões quando é subsônico ( $p_d/p_u > 0,528$ ).

Devido às diferenças entre a abertura efetiva das servoválvulas e às perdas e irreversibilidades que ocorrem no processo de passagem de ar no orifício, as pressões críticas encontradas são, na prática, menores que as teóricas (Mare et al., 2000). Bobrow e McDonell (1998) mostram que as expressões teóricas (5.3) e (5.4) podem ser insuficientes para representar o comportamento da vazão em um sistema complexo como, por exemplo, um servoposicionador pneumático.

Isto leva à necessidade de obtenção experimental das curvas de vazão mássica em função das pressões nas câmaras e da tensão  $u$  de acionamento da servoválvula. Para tanto, descreve-se a vazão mássica como o produto de uma função das pressões por uma função de tensão, de acordo com a Eq. (5.6) (note que esta equação representa a mesma estrutura da Eq. (5.5)):

$$q_m(p_u, p_d, u) = [q_m]_{max} f_p(p_u, p_d) f_u(u) \quad (5.6)$$

onde  $[q_m]_{max}$  é a vazão mássica máxima através do orifício,  $0 \leq f_p(p_u, p_d) \leq 1$  é uma função das pressões, e  $0 \leq f_u(u) \leq 1$  é uma função que considera a abertura efetiva e depende da tensão na servoválvula  $u$ .

Servosistemas pneumáticos são projetados de forma que se possa geralmente assumir as hipóteses de que:

1. A pressão de suprimento é constante
2. A pressão de exaustão é igual à pressão atmosférica

Para o caso de enchimento de uma câmara do cilindro, a pressão montante  $p_u = p_{sup}$  enquanto que a pressão jusante é a pressão  $p_i$  da câmara  $i$  do cilindro. Assim, a vazão mássica da câmara  $i$  do cilindro é descrita por:

$$q_{mi}^{ench}(p_i, u) = [q_{mi}^{ench}]_{max} f_{pi}^{ench}(p_i) f_{ui}^{ench}(u) \quad (5.7)$$

onde  $f_{pi}^{ench}(p_i)$  é a função da pressão quando no enchimento da câmara  $i$  e  $f_{ui}^{ench}(u)$  é a função de tensão correspondente ao enchimento da câmara  $i$ .

Para o caso de exaustão de uma câmara do cilindro, a pressão montante  $p_u = p_i$  e a pressão jusante são a pressão atmosférica  $p_i = p_{atm}$  da câmara  $i$  do cilindro. A vazão mássica da câmara  $i$  do cilindro é descrita por:

$$q_{mi}^{exau}(p_i, u) = [q_{mi}^{exau}]_{max} f_{pi}^{exau}(p_i) f_{ui}^{exau}(u) \quad (5.8)$$

onde  $f_{pi}^{exau}(p_i)$  é a função da pressão de exaustão da câmara  $i$  e  $f_{ui}^{exau}(u)$  é a função de tensão correspondente à exaustão da câmara  $i$ .

O levantamento experimental destas funções é realizado com o êmbolo do cilindro em posição fixa, logo com o volume das câmaras constantes. Assim pode-se reescrever a Eq. (5.2) como:

$$q_{mi}(p_i(t), u) = \frac{V_i}{RrT} \dot{p}_i(t) \quad (5.9)$$

Baseando-se na expressão (5.9), os seguintes procedimentos foram realizados para a aquisição dos valores experimentais de ambas as câmaras do cilindro, tanto para as situações de enchimento quanto de exaustão:

1. Para uma seqüência de valores de tensão na servoválvula, medição do comportamento das pressões ao longo do tempo. Para isto foi desenvolvido um programa que automatiza este processo.
2. Cálculo das derivadas das pressões em relação ao tempo, obtendo as vazões mássicas em função do tempo para cada valor de tensão na servoválvula.
3. Obtenção das curvas estáticas vazão-pressão, cruzando termo a termo os valores das curvas obtidas nos passos 1 e 2.

4. Determinação a partir das curvas obtidas no passo 2, dos valores máximos da vazão em cada câmara.

O procedimento descrito foi realizado para ambas as câmaras do cilindro na situação de volume máximo.

Para as curvas apresentadas a seguir foi utilizado o filtro Savitzky-Golay (MathWorks, 2005).

As curvas experimentais para pressão em função da tensão e do tempo estão apresentadas na Fig. 5.3 enquanto que as curvas experimentais para a vazão em função da tensão e do tempo estão apresentadas na Fig. 5.4. Finalmente, as curvas experimentais para a vazão em função da tensão e da pressão estão apresentadas na Fig. 5.5.

Os gráficos das vazões mássicas normalizadas em relação a pressão estão apresentados na Fig. 5.6, estes são adquiridos a partir dos pontos de máxima vazão para as diferentes tensões aplicadas conforme Fig. 5.4. Os gráficos das vazões mássicas normalizadas em relação a tensão estão apresentados na Fig. 5.7, estes são adquiridos a partir da curva que segue do ponto de máxima vazão, seguindo a mesma tensão até a estabilidade da pressão da Fig. 5.5.

As vazões mássicas máximas estão expostas na Tabela 5.1, adquiridas a partir dos valores máximos da Fig. 5.4.

Tabela 5.1 – Dados da bancada

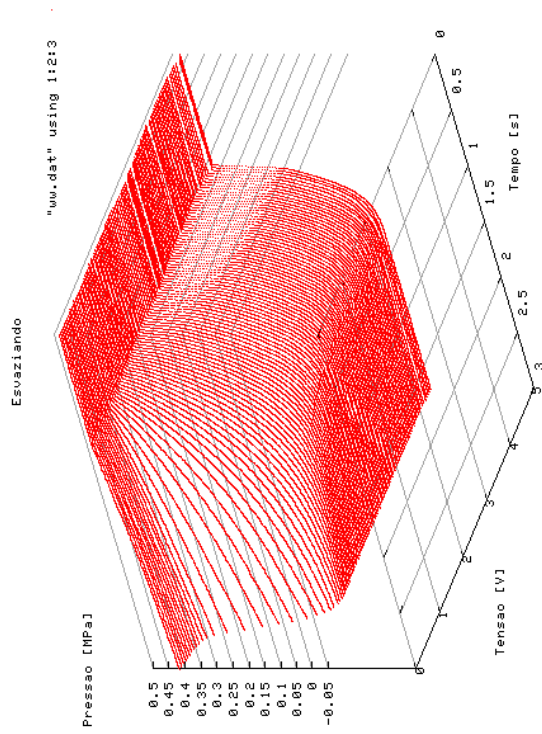
Vazão mássica máxima enchendo (câmara 1)	0.1765 Kg/s
Vazão mássica máxima exaurindo (câmara 1)	0.1486 Kg/s
Vazão mássica máxima enchendo (câmara 2)	0.2094 Kg/s
Vazão mássica máxima exaurindo (câmara 2)	0.1928 Kg/s

### 5.1.3 Identificação do Atrito

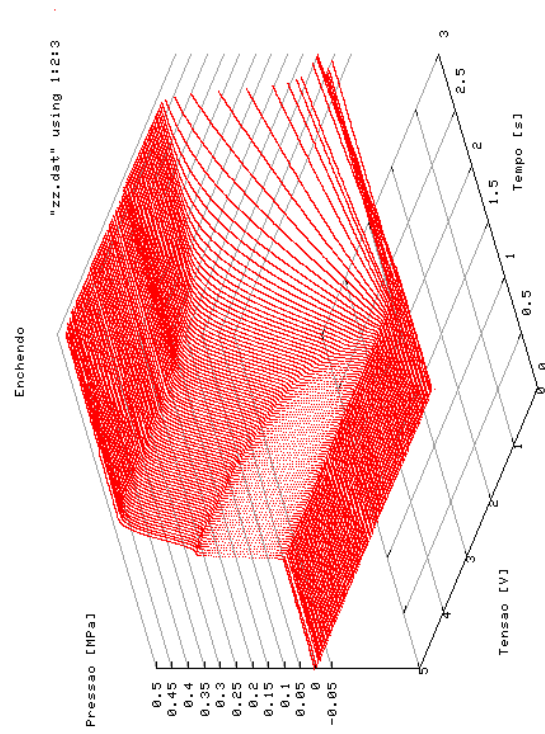
O atrito é uma das principais não-linearidades existentes nos sistemas servopneumáticos, dificultando o controle de servoposicionadores (Perondi, 2002). Em sistemas servopneumáticos a principal fonte de atrito está no contato entre o anel de vedação e o cilindro. Em cilindros com haste também há atrito no contato da haste com a extremidade do cilindro.

O atrito depende de condições ambientais tais como temperatura, lubrificação, direção do movimento, velocidade do êmbolo, imperfeições do cilindro, entre outros.

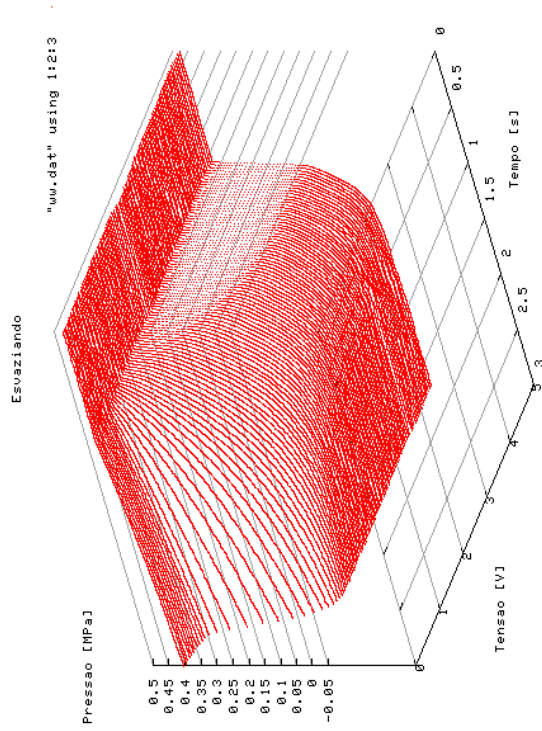
Para realizar simulações que representem o sistema em análise, é necessário utilizar um modelo de atrito. Um exemplo é o modelo de Lugre (Canudas et al. (1995)). Este modelo tem como característica a consideração do movimento em regime de pré-deslizamento.



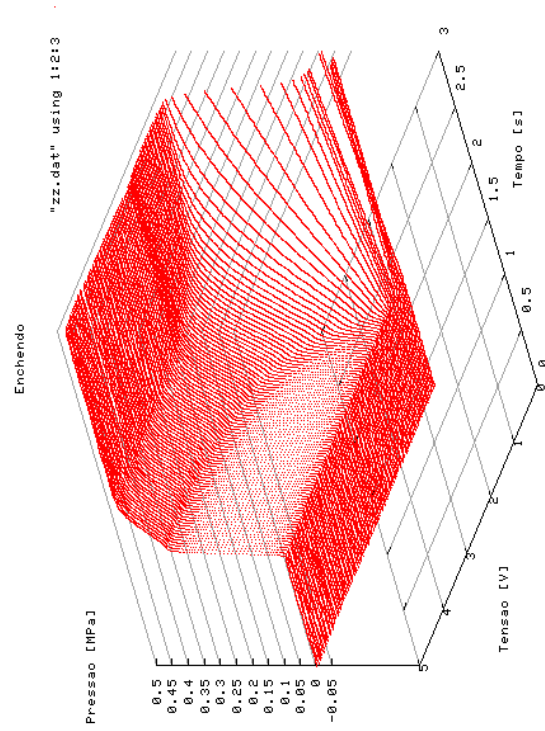
(a) câmara 1 es vazizando



(b) câmara 1 enchendo

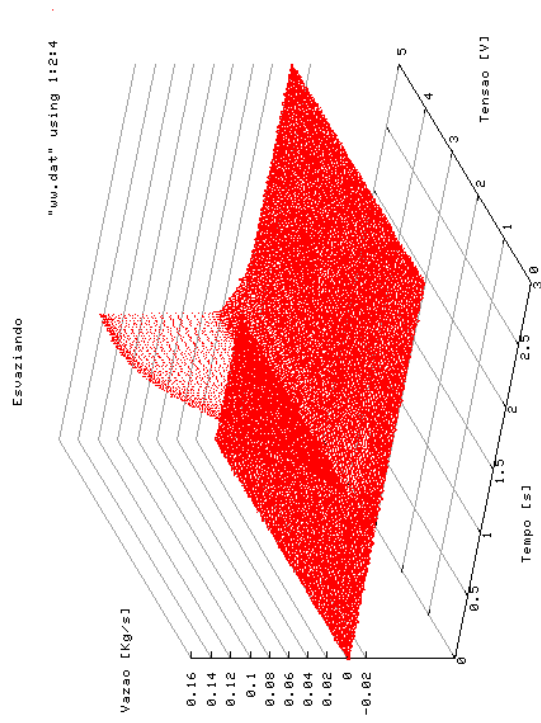


(c) câmara 2 es vazizando

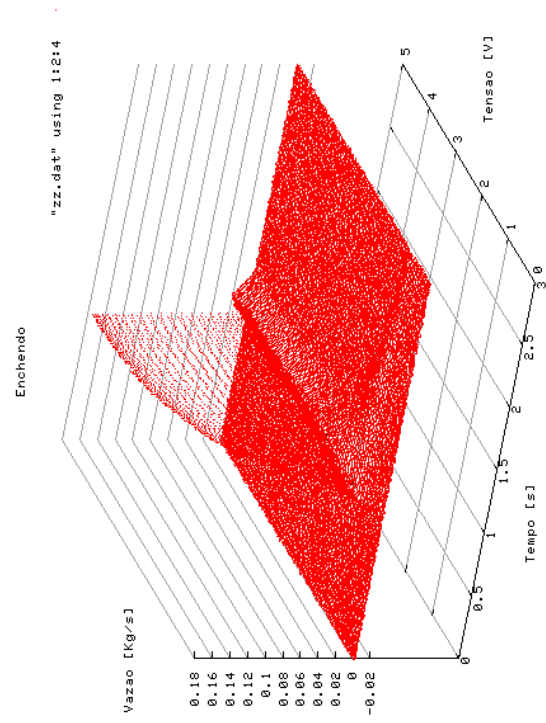


(d) câmara 2 enchendo

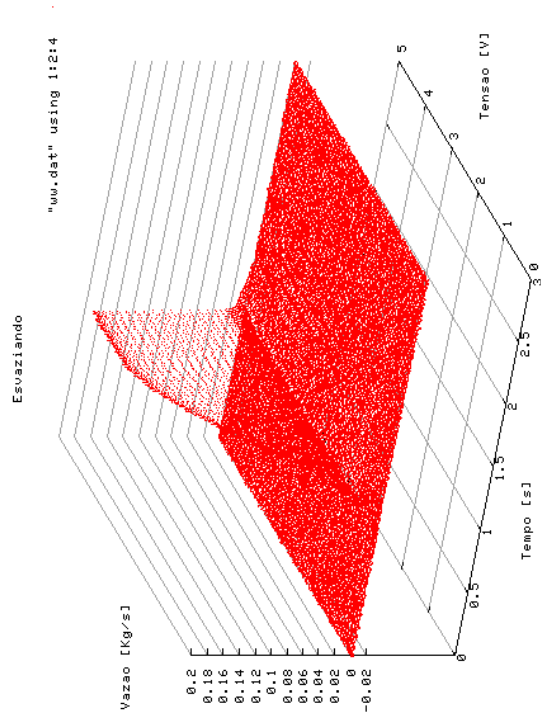
Figura 5.3 – Curvas experimentais pressão x tempo x tensão



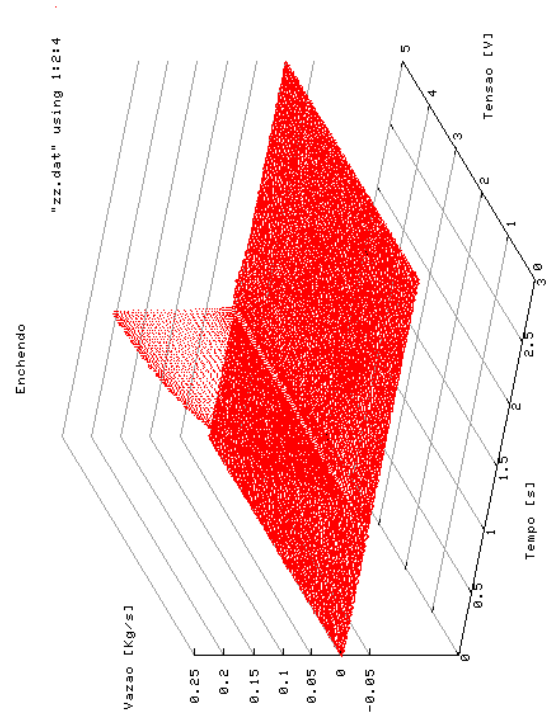
(a) câmara 1 esvaziando



(b) câmara 1 enchendo



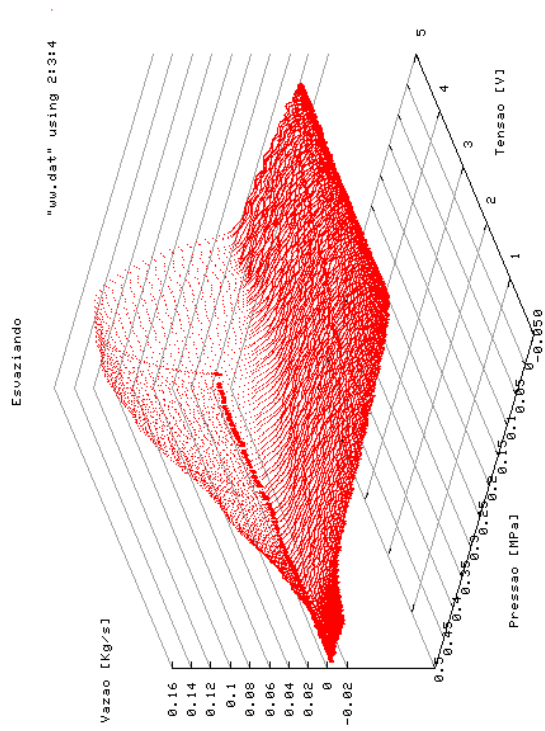
(c) câmara 2 esvaziando



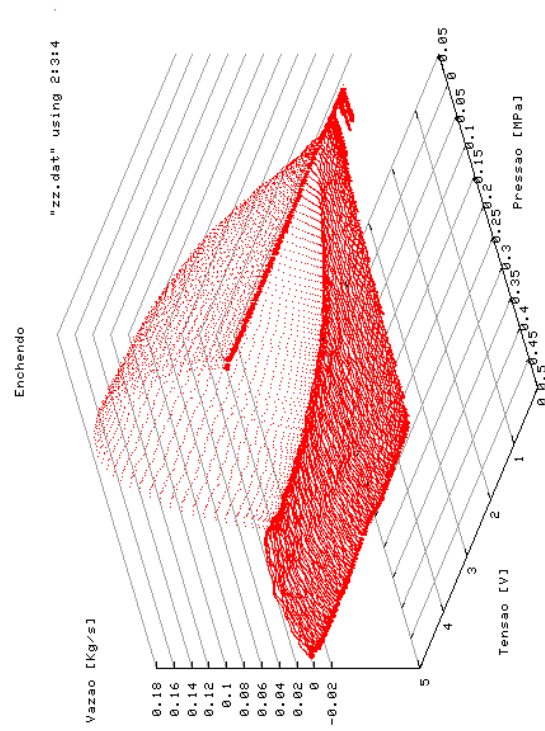
(d) câmara 2 enchendo

Figura 5.4 – Curvas experimentais vazão x tempo x tensão

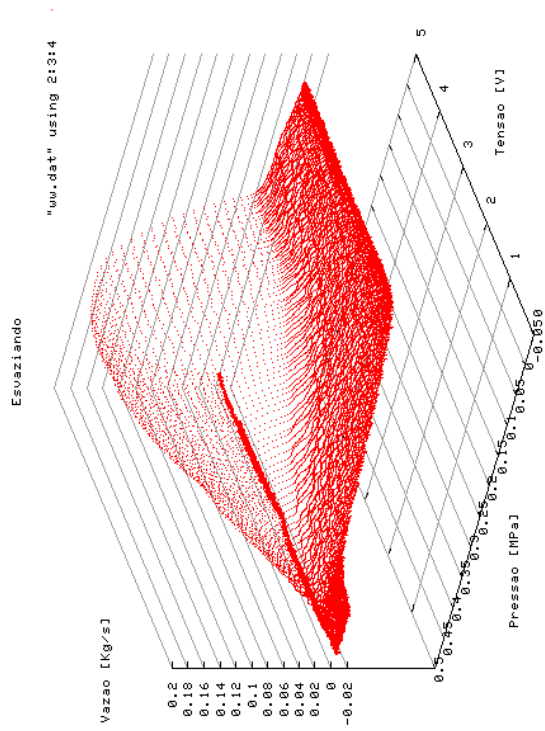




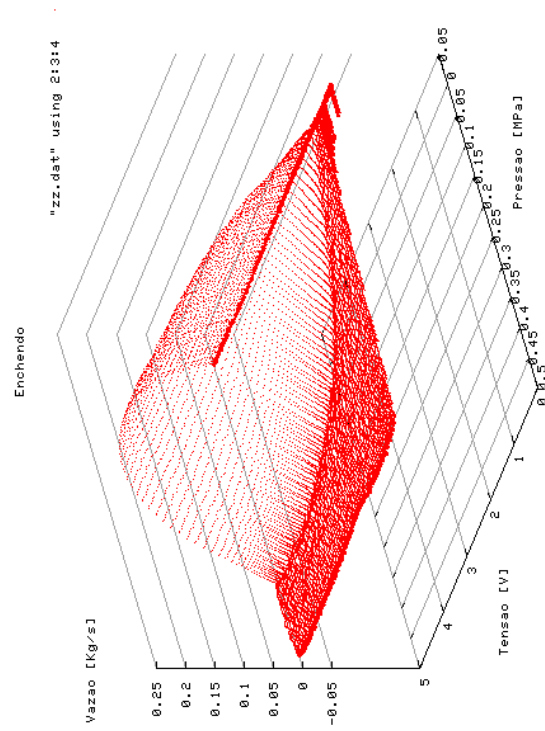
(a) câmara 1 esvaziando



(b) câmara 1 enchendo

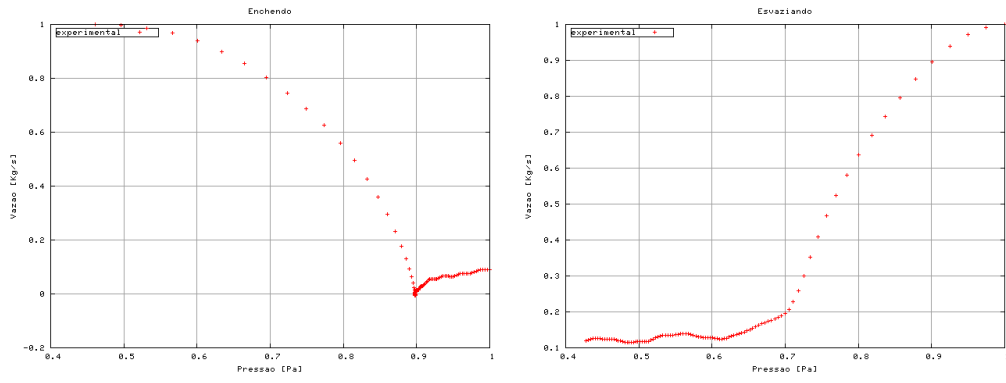


(c) câmara 2 esvaziando



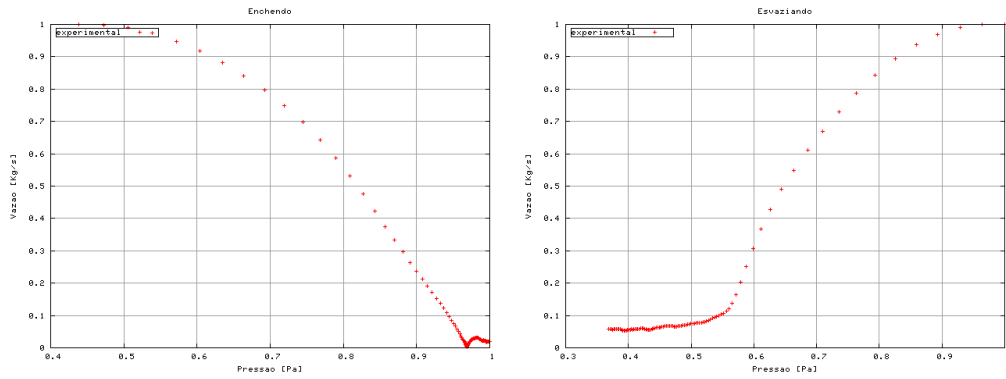
(d) câmara 2 enchendo

Figura 5.5 – Curvas experimentais vazão x pressão x tensão



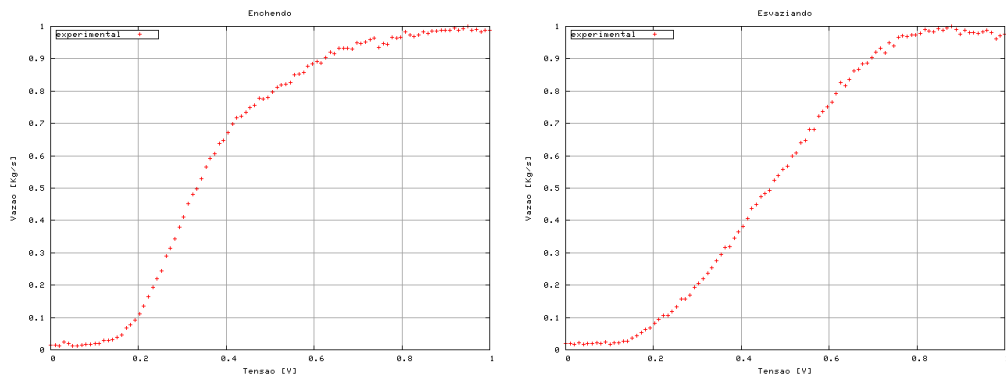
(a) câmara 1 enchendo

(b) câmara 1 esvaziando



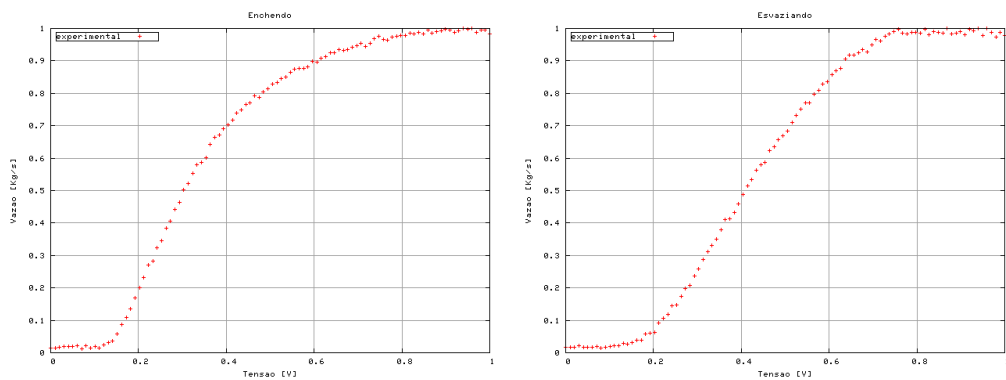
(c) câmara 2 enchendo

(d) câmara 2 esvaziando

Figura 5.6 – Curvas experimentais estáticas de vazão *versus* pressão

(a) câmara 1 enchendo

(b) câmara 1 esvaziando



(c) câmara 2 enchendo

(d) câmara 2 esvaziando

Figura 5.7 – Curvas experimentais estáticas de vazão *versus* tensão

Canudas et al. (1995) propõem o uso da parametrização apresentada na Fig. 5.8 que leva em consideração os atritos seco e viscoso, bem como o efeito de Stribeck (Armstrong-Helouvry et al., 1994) e é expressa por:

$$g(\dot{y}) = F_c + (F_s + F_c)e^{-(\dot{y}/v_s)^2} \quad (5.10)$$

onde  $\dot{y}$  é a velocidade relativa entre duas superfícies e  $g(\dot{y})$  é uma função que depende de fatores como propriedades dos materiais, lubrificação e temperatura.  $F_c$  é a força de atrito seco (ou Coulomb),  $F_s$  é a força de atrito estático e  $v_s$  é a velocidade de Stribeck.

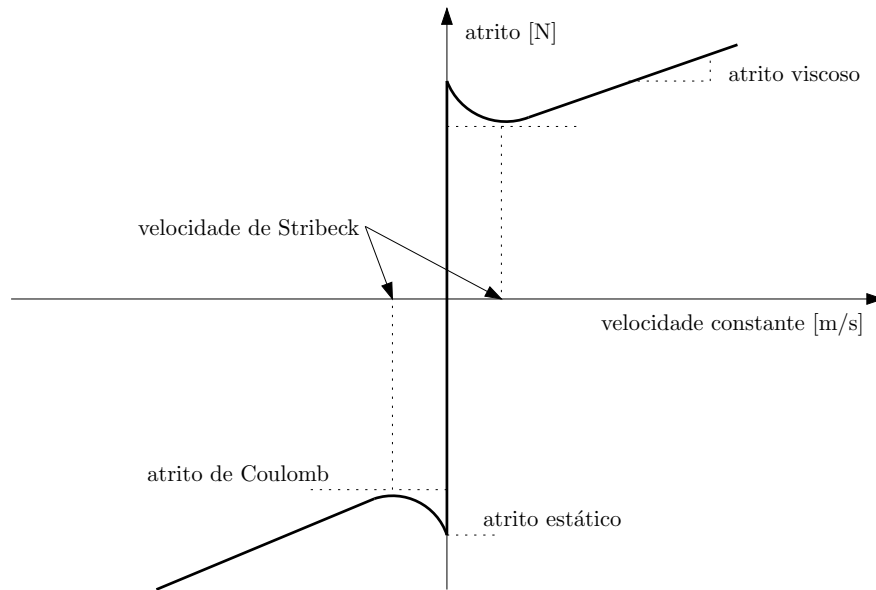


Figura 5.8 – Curva estática de força de atrito pela velocidade constante (Perondi, 2002)

O mapa estático de atrito adquirido na bancada está apresentado na Fig. 5.10. Este foi obtido através da imposição de velocidades constantes (positivas e negativas) realizadas através de um controlador PID calibrado utilizando o método Ziegler-Nichols. A Fig. 5.9 apresenta o resultado do controlador PID para velocidade constante de 0.1m/s.

A partir do momento em que o controlador estabiliza, mantendo a velocidade do êmbolo constante, a diferença entre as pressões nas câmaras 1 e 2 são adquiridas e multiplicadas pela área do êmbolo, resultando na força de atrito ( $F_a$ ). Este procedimento é repetido para uma amostra de velocidades.

Outros atritos existentes no sistema do servoposicionador, como por exemplo, o atrito entre a correia e a polia são desconsiderados.

Através da Fig. 5.10 verifica-se que o efeito de Stribeck (ver, por exemplo, em Armstrong-Helouvry et al. (1994)) não se mostra significativo, conforme característica de atuadores sem haste já referenciada, por exemplo, em Santos (1996). O efeito de Stribeck, responsável pelo efeito de *stick-slip* (adere-desliza), é resultado das não linearidades provindas do atrito no servoatuador, característico em baixas velocidades.

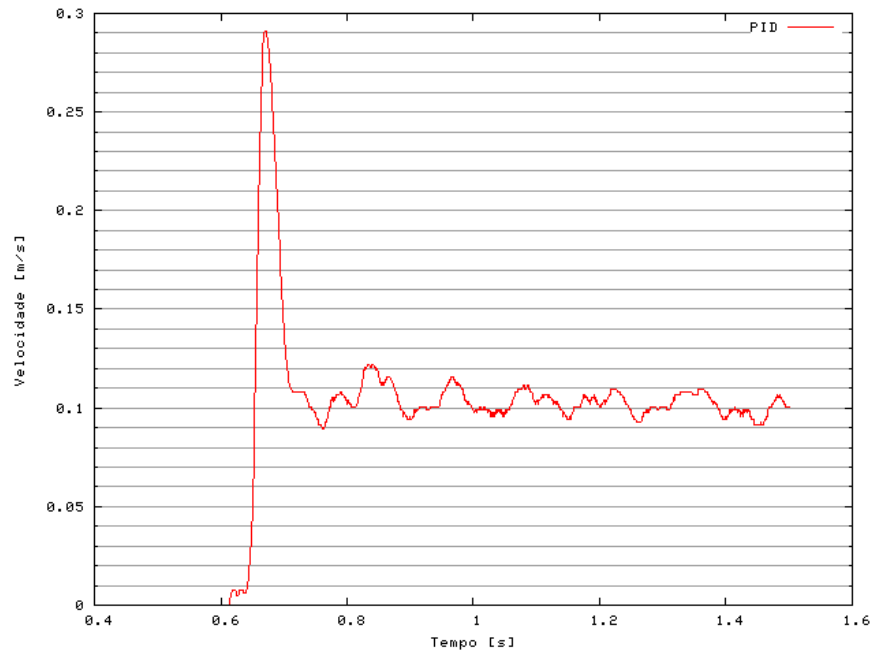


Figura 5.9 – Resposta do controlador PID para velocidade de 0.1m/s

Os valores de atrito adquiridos representados na Fig. 5.10 para  $\dot{y} > 0$  estão dispostos na Tabela 5.2 e para  $\dot{y} < 0$  estão dispostos na Tabela 5.3.

Tabela 5.2 – Atrito para  $\dot{y} > 0$

Atrito estático	75.0N
Atrito de Coulomb	64.3N
Atrito viscoso	0.82N
Velocidade de Stribeck	0.02N

Tabela 5.3 – Atrito para  $\dot{y} < 0$

Atrito estático	-62.6N
Atrito de Coulomb	-51.5N
Atrito viscoso	1.18N
Velocidade de Stribeck	-0.03N

#### 5.1.4 Temporização

No projeto de desenvolvimento de uma arquitetura de controle para servoposicionadores pneumáticos uma das variáveis importantes é a determinação do tempo dos ciclos de controle. A Fig. 5.11 mostra o tempo do ciclo de controle adquirido durante a execução de um PID.

Estes dados estão sintetizados na Tab. 5.4. Observa-se uma baixa latência existente na arquitetura para o controle do servoposicionador, levando em consideração a utilização de um microcomputador de baixo desempenho (ver Tabela 3.12). Conclui-se que o sistema executa o algoritmo

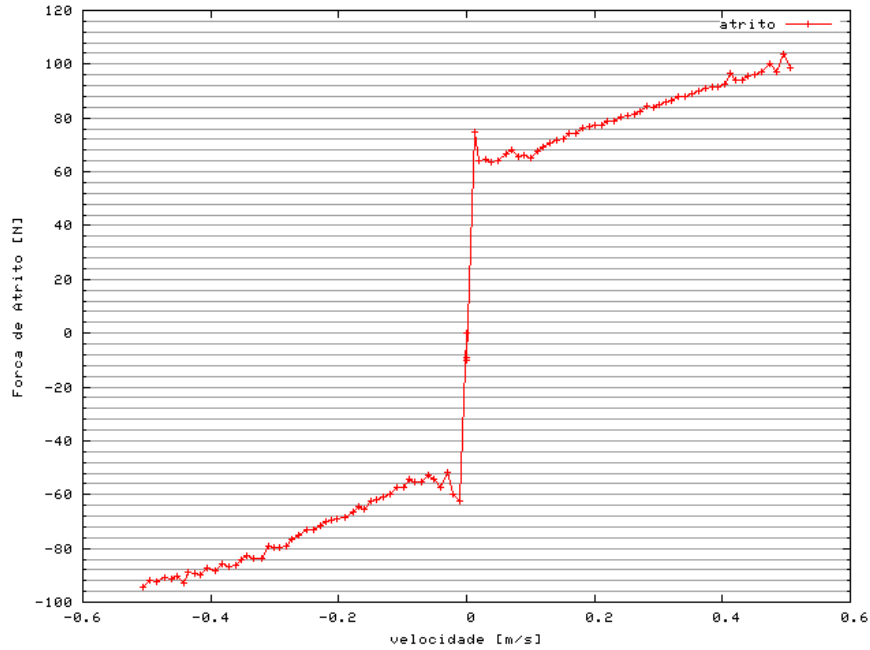


Figura 5.10 – Mapa estático de atrito

de controle com eficiência, de acordo com os requisitos temporais determinados no projeto do servoposicionador.

Tabela 5.4 – Tempo dos ciclos

Média [ms]	0.9999
Máximo [ms]	1.0700
Mínimo [ms]	0.9294
Desvio Padrão [ms]	0.0078

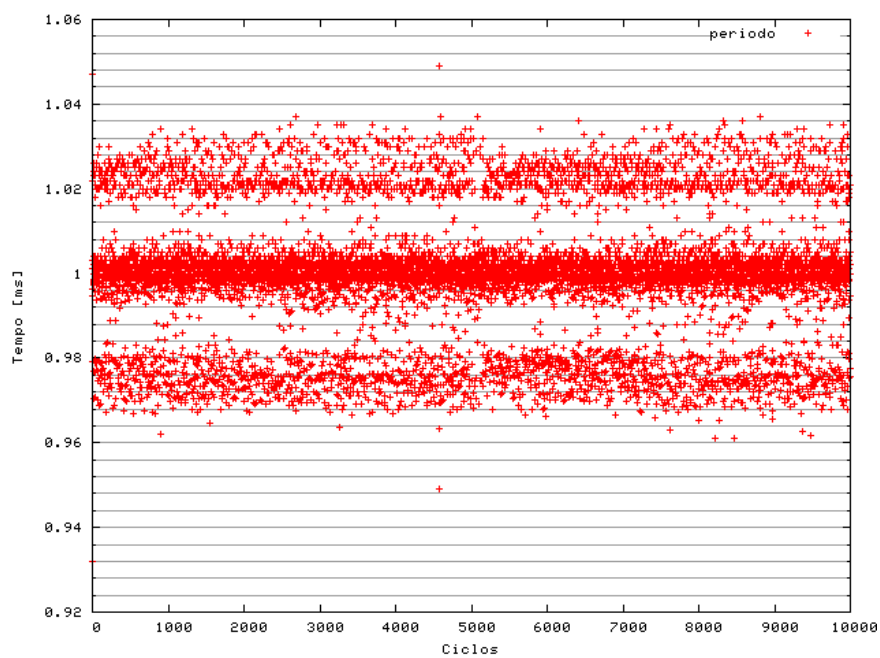


Figura 5.11 – Períodos dos ciclos de aquisição e acionamento

## 6 CONSIDERAÇÕES FINAIS E CONCLUSÕES

No presente trabalho foi realizada uma revisão bibliográfica que abrange a análise de soluções para o controle digital em tempo real, os componentes pneumáticos utilizados em servoposicionadores pneumáticos e as principais técnicas de controle de servoposicionadores pneumáticos.

Foi realizado o projeto de uma bancada experimental e dimensionados os componentes pneumáticos e os equipamentos utilizados para processamento e sensoriamento da bancada. Posteriormente foram desenvolvidos os sistemas de interfaceamento dos sensores com a central de processamento, condicionamento de sinais e os processos de controle e monitoramento do servoposicionador pneumático. Por fim foram realizados os testes para determinação dos parâmetros do sistema (tais como a vazão mássica, força de atrito e a eficiência temporal na execução dos algoritmos de controle).

Na revisão bibliográfica foram pesquisadas as principais soluções para a construção de servoposicionadores pneumáticos, avaliando as condições favoráveis e desfavoráveis de cada uma das soluções segundo critérios de custo, desempenho, etc. Foram pesquisadas as técnicas de controle mais utilizadas para o controle de servoposicionadores. Observou-se assim a necessidade do desenvolvimento de um sistema de controle flexível visto a tendência de aumento da complexidade dos algoritmos para controle utilizados em servoposicionadores pneumáticos. Após estudos de viabilidade e custo, optou-se pelo desenvolvimento de um sistema híbrido com microprocessadores e microcontroladores, visando principalmente a flexibilidade do sistema final.

Como resultado da revisão bibliográfica foi verificada a necessidade de precisão temporal no controle da bancada experimental. Este requisito foi resolvido através da implementação de processos de controle desenvolvidos sobre o sistema operacional GNU/Linux com suporte a tempo real (RTAI). Em conjunto a estas ferramentas, foram utilizados os *softwares* para o desenvolvimento de algoritmos de controle (Scilab/Scicos), e de monitoramento (XRtaiLab). Todos os programas descritos anteriormente são de código aberto e de livre distribuição, condições favoráveis, portanto, aos critérios de custo e de independência tecnológica.

Por fim, foi realizada a caracterização do sistema através da identificação das curvas de vazão mássica através dos orifícios da servoválvula para ambas as câmaras do cilindro pneumático, a identificação do atrito do cilindro pneumático e a verificação da eficiência da temporização determinada pelo sistema de tempo real.

A arquitetura de controle demonstrou-se eficiente para o controle de servoposicionadores pneumáticos principalmente com relação aos seguintes itens:

- Tempo de Ciclo. O sistema demonstrou ter garantias quanto a determinação do tempo de

execução de seus processos, mesmo utilizando *hardware* de baixo custo.

- **Tecnológico.** Todos os aplicativos utilizados para controlar a bancada possuem código aberto e de livre utilização, evitando assim a dependência de tecnologias fechadas. Alterações necessárias podem ser realizadas independentemente.
- **Econômico.** O sistema resultante demonstrou ser economicamente atrativo visto que o sistema implementado não tem dependência de materiais (*hardware*) nem programas (*softwares*) caros (cerca de 90%).
- **Flexibilidade.** Para o desenvolvimento de algoritmos de controle para servoposicionadores pneumáticos tem-se muita flexibilidade através da utilização dos blocos de programação fornecidos pelo sistema Scilab/Scicos.
- **Modularidade.** O sistema desenvolvido é modular, podendo ser dividido, inclusive, entre diferentes centrais de processamento e monitoramento.

Conclui-se que o trabalho atingiu as metas propostas de desenvolver um servoposicionador pneumático barato e preciso. Por ser um produto flexível, robusto e de baixo custo, o sistema desenvolvido pode ser utilizado em outras áreas de controle de posicionamento. Com isto, esta bancada colabora com o desenvolvimento de tecnologias que permitam a disseminação de servoposicionadores na indústria através da diminuição de custos, mantendo a robustez das soluções.

## 6.1 Trabalhos Futuros

Pesquisas futuras sobre o desenvolvimento de arquiteturas de controle poderão abordar os seguintes itens:

- **Aplicação da arquitetura de controle em sistemas embarcados.** A implementação desta arquitetura em *hardwares* mais compactos (embarcados) que permitam que esta solução seja aplicada em uma maior quantidade de aplicações, como, por exemplo, no controle aeroespacial.
- **Desenvolvimento de sistemas de controle orientado a objetos.** A implementação desta arquitetura de controle utilizando a filosofia de orientação a objetos, aumentando a modularidade do sistema.
- **Desenvolvimento de sistemas de controle para robótica paralela.** A utilização desta arquitetura para controle de dois ou mais servoatuadores em paralelo, auxiliando no desenvolvimento de soluções mais robustas para manipulação de objetos.

- **O modelo de Stribeck.** Uma adequada identificação experimental do mapa estático de atrito do servoposicionador, depende da utilização de um controlador eficiente para a velocidade. Assim, na medida em que novos controladores forem desenvolvidos (além do PID), mapas mais acurados poderão ser obtidos.
- **Avaliar as vibrações.** Verificar os modos de frequência existentes no sistema e, se for necessário, diminuir a interferência da vibração da correia dentada na leitura da posição.
- **Segurança contra falhas.** Implementar algoritmos de verificação de segurança do sistema, como por exemplo a chegada do fim de curso no cilindro pneumático ou falhas no sistema de controle e aquisição.



## REFERÊNCIAS

- ANAKWA, W. K. N.; THOMAS, D. R.; JONES, S. C.; BUSH, J.; GREEN, D.; ANGLIN, G. W.; RIO, R.; SHENG, J.; GARRETT, S.; CHEN, L. Development and control of a prototype pneumatic active suspension system. *IEEE TRANSACTIONS ON EDUCATION*, v. 45, p. 43–49, 2002.
- ANDRIGHETTO, P. L.; VALDIERO, A. C.; CARLOTTO, L.; HARTER, I. I. Desenvolvimento de um manipulador robótico acionado pneumáticamente. *IX Seminário de Automação e Processos*, out. 2005.
- ANDRIGHETTO, P. L.; VALDIERO, A. C.; VINCENSI, C. N. Experimental comparisons of the control solutions for pneumatic servo actuators. 2005.
- ARINAGA, T.; KAWAKAMI, Y.; TERASHIMA, Y.; KAWAI, S. Approach for energy-saving of pneumatic systems. In: *Proceedings of the 1st FPNI - PhD Symposium, Fluid Power Net International*, 2000.
- ARMSTRONG-HELOUVRY, B.; DUPONT, P.; WIT, C. C. de. A survey of models, analysis tools, and compensation methods for control of machines with friction. *Automatica*, v. 30, p. 183–1138, 1994.
- ATMANAND, M. A.; KONNUR, M. S. A novel method of using a control valve for measurement and control of flow. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, v. 48, p. 1224–1226, 1999.
- BACHMANN, R. J.; SURGENOR, B. W. On the Design and Performance of a Closed Circuit Pneumatic Positioning System. In: *The Fifth Scandinavian International Conference on Fluid Power*, 1997.
- BELGHARBI, M.; SESMAT, S.; THOMASSET, D.; SCAVARDA, S. Force Tracking Control of an Electro-pneumatic Actuator Using a Linearized Model Around the Reference Trajectory. In: *Power Transmission and Motion Control*, 1998.
- BOBROW, J. E.; JABBARI, F. Adaptive pneumatic force actuation and position control. *ASME Journal of Dynamic Systems, Measurement and Control*, v. 113, p. 267–272, 1991.
- BOBROW, J. E.; MCDONELL, B. W. Modeling, identification, and control of a pneumatically actuated force controllable robot. 1998.
- BOUDART, S.; RICHARD, E.; SCAVARDA, S. A modified linear reduced observer for a pneumatic servodrive. In: *Proceedings of Fourth Bath International Fluid Power Workshop*, 1991.
- BOURI, M.; THOMASSET, D.; RICHARD, E.; SCAVARDA, E. Nonlinear sliding mode control of an electropneumatic servodrive. In: *Proceedings 7th Bath International Fluid Power Workshop*, 1994. p. 21–23.
- BRUN, X.; THOMASSET, D.; BIDEAUX, E.; SCAVARDA, S. An accurate tracking control of an electropneumatic actuator. In: , 2000. p. 215–226.
- BUCHER, R.; DOZIO, L.; MANTEGAZZA, P. Rapid control prototyping with scilab/scicos and linux rtaí. *Scilab-2004*, 2004.
- BUCHER, R.; MANNORI, S. Using scilab/scicos with rtaí-lab. In: *Fifth Real-Time Linux Workshop*, 2003.

- BUESCHER, W. *WinPic*. 2005. Disponível em: <<http://www.qsl.net/dl4yhf/winpicpr.html>>. Acesso em: 05/12/2005.
- CANUDAS, C.; OLSSON, H.; ASTROM, K. J.; LISCHINSKY, P. A new model for control systems with friction. *IEEE Trans. on Automatic Control*, v. 40, p. 419–425, 1995.
- CHOI, G. S.; LEE, H. K.; CHOI, G. H. A study on tracking position control of pneumatic actuators using neural network. In: *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, 1998. v. 3, p. 1749–1753.
- COELHO, L. dos S.; RODRIGUES, A. A. Algoritmos evolutivos em identificação e controle de processos: Uma visão integrada e perspectivas. v. 10, 1999.
- CRUZ, F. B. C. *Modelagem e Controle Não-Lineares de um Posicionador Servopneumático Industrial*. Dissertação (Mestrado) — UFSC, Florianópolis, 2003.
- DIAPM, D. of Aerospace Engineering of Politecnico di M. *Real-Time Application Interface*. 2005. Disponível em: <<http://www.rtai.org/>>. Acesso em: 05/12/2005.
- DORF, R. H. *Modern Control Systems*: Addison-Wesley Publishing Company, 1989.
- DOZIO, L.; MANTEGAZZA, P. Linux real time application interface in low cost high performance motion control. In: ASSOCIAZIONE NAZIONALE ITALIANA PER L'AUTOMAZIONE. *Motion Control 2003*, 2003.
- dSPACE GmbH. set. 2005. Disponível em: <[www.dspace.com](http://www.dspace.com)>. Acesso em: 05/12/2005.
- EDGE, K. A. The control of fluid power systems - responding to the challenges. *Proc. Inst. Mechanical Engineering*, v. 211, p. 91–110, 1997.
- Festo Corporation. 2005. Disponível em: <[www.festo.com](http://www.festo.com)>. Acesso em: 05/12/2005.
- FRANKLIN, G. F.; POWELL, J. D.; EMANI-NAEINI. *Feedback Control of Dynamic Systems*: Addison Wesley Publishing Company, 1994.
- GIBERTI, H.; RIGHETTINI, P.; TASORA, A. Design and experimental test of a pneumatic translational 3dof parallel manipulator. In: *10th International Workshop on robotics in Alpe-Adria-Danube region*, 2001.
- GOTTERT, M.; NEUMANN, R. Nichlineare regelungskonzepte fur servopneumatische roboter. *3th German-Polish seminar in Innovation and Progress in Fluid Engineering*, 1999.
- GRANOSIK, G.; BORENSTEIN, J. Minimizing air consumption of pneumatic actuators in mobile robots. 1998.
- GROSS, D. C.; RATTAN, K. S. A feedforward mnn controller for pneumatic cylinder trajectory tracking control. In: *International Conference on Neural Networks*, 1997. v. 2, p. 794 –799.
- GUOLIANG, T.; XUANYIN, W. Research on pneumatic-servo calligraphy robot. 2003.
- HANCHIN, G. D.; OZGUNER, U.; DRAKUNOV, S. Nonlinear control of a rodless pneumatic servoactuator. *First IEEE Conference on Control Applications*, v. 1, p. 516–521, 1992.
- HARRISON, R.; WESTON, R. H.; MOORE, P. R.; THATCHER, T. W. A study of application areas for modular robots. *Robotica*, 1987.

- HATIPOGLU, C.; BENNETT, M. A.; AMATO, W. P.; MORAN, S. P. Overcoming actuator performance degradation caused by non-smooth nonlinearities. In: *Proceedings of the 1999 American Control Conference*, 1999. v. 1, p. 167–168.
- INRIA. *Scicos*. 2005. Disponível em: <<http://www.scicos.org>>. Acesso em: 05/12/2005.
- INRIA. *Scilab*. 2005. Disponível em: <<http://www.scilab.org/>>. Acesso em: 05/12/2005.
- KAWAMURA, S.; MIYATA, K.; HANAFUSA, H.; ISIDA, K. Pi type hierarchical feedback control scheme for pneumatic robots. In: , 1989. v. 3, p. 1853–1858.
- KILIAN, C. T. *Modern Control Technology - Components and Systems*: Delmar, 2000.
- KUNZ, G.; DAMASIO, F.; KICHALOWSKI, M. Aplicações scada utilizando gnu/linux. *InTech*, v. 51, p. 46, 2003.
- LAGES, W. F. Arquiteturas para acionamento e controle em tempo real de robôs. *II Simpósio de Computação, Informática e Tecnologia do Vale do Taquari*, 2004.
- LAGES, W. F.; ALT, G. H. Controle em tempo real de sistemas dinâmicos através de redes ip. In: *3o Congresso Internacional de Automação, Sistemas e Instrumentação*, 2003.
- LAI, Y. J.; MENQ, C. H.; SINGH, R. Accurate position control of a pneumatic actuator. *ASME Journal of Dynamic Systems, Measurement and Control*, v. 112, p. 734–739, 1990.
- LARA-ROJO, F.; N.SANCHEZ, E.; ZALDIVAR-NAVARRO, D. Real-time fuzzy microcontroller for a didatic level system. In: *International electronic enginner conference*, 2001.
- LATINO, F.; SANDOVAL, D. Quit overspending for servomotion systems. *Machine Design*, p. 93–96, 1996.
- LIN, X.; SPETTEL, F.; SCAVARDA, S. Modeling and test of an eletropneumatic servovalve controlled long rodless actuator. *ASME Journal of Dynamic Systems, Measurement and Control*, v. 118, p. 457–462, set. 1996.
- LIU, S.; BOBROW, J. E. An analysis of a pneumatic servo system and its application to a computer-controlled robot. 1988.
- MARE, J. C.; GEIDER, O.; COLIN, S. An improved dynamic model of pneumatic actuators. *Internation Journal of Fluid Power*, v. 2, p. 39–47, 2000.
- MARTIN, H. *The Design of Hydraulic Components and Systems*: Ellis Horwood Limited, 1995.
- MATHWORKS. *MatLab*. 2005. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 05/12/2005.
- MCDONELL, B. W. *Modeling, Identification, and Control of a Pneumatically Actuated Robotic Manipulator*. Tese (Doutorado) — University of California, 1996.
- MCDONELL, B. W.; BOBROW, J. E. Modeling, identification, and control of a pneumatically actuated robot. In: *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, 1997. v. 1, p. 124–129.
- MEANWELL. *25W Single Output Switching Power Supply*, 2004.
- MICROCHIP. 2005. Disponível em: <[www.microchip.com](http://www.microchip.com)>. Acesso em: 05/12/2005.

- NORITSUGU, T.; TAKAIWA, M. Robust positioning control of pneumatic servo system with pressure control loop. In: *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, 1995. v. 3, p. 2613–2618.
- NOURI, B.; BENDER, F. A. I.; SWEVERS, J.; VANHERCK, P.; BRUSSEL, H. V. Modeling a pneumatic servo positioning system with friction. In: *Proceedings of the ACC*, 2000. p. 1067–1071.
- OGATA, K. *Engenharia de Controle Moderno*: Prentice Hall do Brasil LTDA, 1998.
- OKIYAMA, K.; ICIRYU, K. Study of pneumatic motion base control characteristics. In: , 2001. p. 228–232.
- PANDIAN, S. R.; HAYAKAWA, Y.; KANAZAWA, Y.; KAMOYAMA, Y.; KAWAMURA, S. Practical design of a sliding mode controller for pneumatic actuators. *ASME Journal of Dynamic Systems, Measurement and Control*, v. 119, p. 666–674, 1997.
- PANDIAN, S. R.; LEDA, K.; KAMOYAMA, Y.; KAWAMURA, S. Modeling and control of a pneumatic rotary actuator. In: *Power Transmission and Motion Control*, 1998.
- PANDIAN, S. R.; TAKEMURA, F.; HAYAKAWA, Y.; KAWAMURA, S. Control performance of an air motor - can air motors replace electric motors? In: *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, 1999. v. 1, p. 518–524.
- PAUL, A. K.; MISHRA, J. E.; RADKE, M. G. Sliding mode control of pneumatic actuator for robotic application. In: *Proceedings of the 1993 IRIRR/RSJ International Conference on Intelligent Robot and Systems*, 1993. v. 2, p. 1268–1275.
- PAUL, A. K.; MISHRA, J. E.; RADKE, M. G. Reduced order sliding mode control for pneumatic actuator. *IEEE Transactions on Control Systems Technology*, v. 2, p. 271–276, 1994.
- PEREIRA, C.; KUNZ, G.; ATAIDE, F.; FREITAS, E.; TEODORO, E.; CARVALHO, F. Performance evaluation of a java architecture used in embedded real time systems. In: *ETFA*, 2005.
- PERONDI, E. *Controle não-linear em cascata de um servoposicionador pneumático com compensação de atrito*. Tese (Doutorado) — UFSC, Florianópolis, 2002.
- PERONDI, E. A.; GUENTHER, R. Controle com compensação de atrito de um servoposicionador pneumático. In: *Anais do IV Congresso Iberoamericano de Engenharia Mecânica*, 1999.
- PERONDI, E. A.; GUENTHER, R. Controle em malha fechada de um servoposicionador pneumático. *Anais do XV Congresso Brasileiro de Engenharia Mecânica*, 1999.
- PINCHES, M. J.; CALLEAR, B. J. *Power pneumatics*. 1996.
- PU, J.; MOORE, P. R.; HARRISON, R.; WESTON, R. H. A study of gain-scheduling method for controlling the motion of pneumatic servos. In: *Proceedings of Sixty Bath International Fluid Power Workshop*, 1993. p. 193–210.
- PU, J.; WANG, J. H.; MOORE, P. R.; WONG, C. B. A new strategy for closed-loop control of servo-pneumatic systems with improved energy efficiency and system response. 1997.

- PU, J.; WESTON, R. H.; MOORE, P. R. Digital motion control and profile planning for pneumatic servos. *ASME Journal of Dynamic Systems, Measurement and Control*, v. 114, p. 634–640, dez. 1992.
- RAJENDRAN, S.; BOLTON, R. W. Position control of a servopneumatic actuator using fuzzy compensation. In: *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*, 2003.
- RENN, J. C.; LIAO, C. M. *A study on the speed control performance of a motor and the application to pneumatic tools*, 2004.
- RICHARD, E.; SCAVARDA, S. Comparison between linear and nonlinear control of an electropneumatic servovalve. *Journal of Dynamic Systems, Measurement and Control*, v. 118, p. 245–252, 1996.
- RICHARDSON, R.; PLUMMER, A. R.; BROWN, M. D. Self-tuning control of a low-friction pneumatic actuator under the influence of gravity. *IEEE Transactions on Control Systems Technology*, v. 9, p. 330–334, 2001.
- SANTOS, E. A. P. *Análise Teórico Experimental de um Posicionador Pneumático*. Tese (Doutorado) — UFSC, Florianópolis, 1996.
- SCAVARDA, S.; THOMASSET, D. Modeling and control of electropneumatic systems: An overview of recent french contributions. *Internacional Conference on Control*, v. 2, p. 1462–1467, 1996.
- SCHLEEF, D.; HESS, F. M. *Comedi Project*. 2005. Disponível em: <<http://www.comedi.org/>>. Acesso em: 05/12/2005.
- SHIH, M. C.; HUANG, C. G. Fuzzy pwm control oh the positions of a pneumatic robot cylinder using high speed solenoid valve. *JSME - Japan Society of Mechanical Engineers International Journal*, v. 40, p. 469–476, 1997.
- SLOTINE, J.-J. E. *Applied Nonlinear Control*: Pearson Education, 1991.
- SONG, J.; BAO, X.; ISHIDA, Y. An application of mnn trained by meka for the position control of pneumatic cylinder. *International Conference on Neural Networks*, v. 2, p. 829–833, 1997.
- SU, W. C.; KUO, C. Y. Variable structure control of a rodless pneumatic servoactuator with discontinuous sliding surfaces. In: *Proceedings of the 2000 American Control Conference*, 2000. v. 3, p. 1617–1621.
- Sun Microsystems. *The Real-Time Java Platform*. jun. 2004.
- SURGENOR, B. W.; VAUGHAN, N. D. Continuous sliding mode control of a pneumatic actuator. *ASME Journal of Dynamic Systems, Measurement and Control*, v. 119, p. 578–581, 1997.
- SURGENOR, B. W.; VAUGHAN, N. D.; UEBLING, E. Continuous sliding mode control of a pneumatic positioning system. In: *Proceedings of Eighth Bath International Fluid Power Workshop*, 1995. p. 270–285.
- TAIT'S, D. 1999. Disponível em: <<http://members.optusnet.com.au/dontronics/dtait/piclincs.html>>. Acesso em: 05/12/2005.

TANAKA, K.; YAMADA, Y.; SHIMIZU, A.; SHIBATA, S. Multi-rate adaptative pole-placement control for pneumatic servo system with additive external forces. In: , 1996. v. 1, p. 213–218.

TANG, J.; WALKER, G. Variable structure control of a pneumatic actuator. *Journal of Dynamic Systems, Measurement and Control*, v. 117, p. 88–92, 1995.

THOMASSET, D.; RICHARD, E.; SCAVARDA, S.; LIN, X. F.; SESMAT, S.; BOUHAL, A. Control of an electro-pneumatic servodrive: a state-affine or a sliding approach. *IFAC World Congress*, 1993.

VALDIERO, A. C. *Controle de Robôs Hidráulicos com Compensação de Atrito*. Tese (Doutorado), 2005.

VIEIRA, A. D. *Análise Teórico-experimental de Servoposicionadores Lineares Pneumáticos*. Tese (Doutorado), 1998.

VIRVALO, T. Designing a pneumatic position servo system. *Power International*, p. 141–147, 1989.

VIRVALO, T. Modeling and design of a pneumatic position servo system realized with commercial components. 1995.

YIN, Y. B.; ARAKI, K. Modelling and analysis of an asymmetric valve-controlled single-acting cylinder of a pneumatic force control system. In: *Proceedings of the 37th SICE Annual Conference*, 1998. p. 1099 –1104.

## APÊNDICE A – ANEXOS

### A.1 Código do *Decoder*

Abaixo é apresentado o código desenvolvido para implementação do sistema de contagem do *encoder* incremental utilizando o microcontrolador PIC18F2550. Este programa também realiza a comunicação com o sistema de controle principal através da porta serial do microcomputador.

```
#pragma config WDTPS    = 32768
#pragma config MCLRE    = OFF
#pragma config LPT1OSC  = OFF
#pragma config PBAEN    = OFF
#pragma config CCP2MX   = OFF
#pragma config STVREN   = OFF
#pragma config LVP      = OFF
#pragma config ICPRT    = OFF
#pragma config XINST    = OFF
#pragma config DEBUG    = OFF
#define CHANA  PORTBbits.RB0
#define CHANB  PORTBbits.RB1
#define SIZE   0x02
int cnt=0, i;
unsigned char la, lb, last;
unsigned char buff[SIZE];
void ab_handler (void);
#pragma code ab_interrupt = 0x8
void ab_int (void)
{
    _asm goto ab_handler _endasm
}
#pragma code
#pragma interrupt ab_handler
void ab_handler (void)
{
    la = CHANA;
    lb = CHANB;
    if(la != last)
    {
        if (la==lb) cnt--;
        else      cnt++;
        if (cnt>=0x3200) cnt=0;
        last = la;
    }
    INTCONbits.INTOIF = 0;
}
void main (void)
{
    unsigned char c;
    int tmp;
    /* Configure all PORTB pins for output */
    TRISB = 0;
```

```

/* Configure PORTB pins for input */
TRISBbits.TRISB0 = 1;
TRISBbits.TRISB1 = 1;
baudUSART (BAUD_16_BIT_RATE &
           BAUD_WAKEUP_OFF &
           BAUD_AUTO_OFF);
OpenUSART (USART_TX_INT_OFF &
           USART_RX_INT_OFF &
           USART_ASYNC_MODE &
           USART_EIGHT_BIT &
           USART_CONT_RX &
           USART_BRGH_HIGH, 68);
OpenRBOINT(PORTB_CHANGE_INT_ON &
           RISING_EDGE_INT &
           PORTB_PULLUPS_ON);
/* Enable interrupt priority */
RCONbits.IPEN = 1;
/* Make receive interrupt high priority */
IPR1bits.RCIP = 1;
/* Enable all high priority interrupts */
INTCONbits.GIEH = 1;

while (1)
{
    while (!DataRdyUSART());
    c = getcUSART();
    switch(c)
    {
        case 'z':
            cnt = tmp = 0;
            break;
        case 'c':
            tmp = cnt;
            break;
        default:
            buff[0] = cnt;
            buff[1] = cnt>>8;

            for (i=0; i<SIZE; i++)
            {
                while(BusyUSART());
                putcUSART(buff[i]);
            }
    }
}

CloseRBOINT();
CloseUSART();
}

```



## A.2 Código de controle dos dispositivos do servoposicionador

A seguir é apresentado o código desenvolvido para um microcomputador IBM-PC com sistema operacional GNU/Linux. Este programa realiza as funções de controle, em tempo real, dos dispositivos do servoposicionador, realizando a comunicação com o algoritmo de controle projetado no ambiente Scilab/Scicos.

```

#define IG          5.00
#define PG          1.00
#define DG          1.65

#define MAX         4.0

#define TICK_TIME   1e6
#define TIME_OUT    (TICK_TIME-1e5)

#define DISTBGN     0
#define DISTEND     12000

#define BASESR      0x2F8

#define PTY_MAIN    0
#define PTY_ENC     1
#define PTY_VAL     2
#define PTY_PRS     3
#define PTY_SND     4

static RTIME expected;
static RTIME period;
static volatile int stop_all;

static RT_TASK *Encoder_Task;    //1
static RT_TASK *Valvula_Task;    //2
static RT_TASK *Pressure_Task;   //3
static RT_TASK *SendBox_Task;    //4

static pthread_t encoder_thread;
static pthread_t valvula_thread;
static pthread_t pressure_thread;
static pthread_t sendbox_thread;

static void *encoder_fun (void *arg);
static void *valvula_fun (void *arg);
static void *pressure_fun(void *arg);
static void *sendbox_fun (void *arg);

static char *filename = "/dev/comedi0";
static int range = 0;
static int subai=0, subao=1;
static int aref = AREF_GROUND;

```

```

static void *device;

static SEM *barrier;
static SEM *semval;

/*
  val 0: posicao
  val 1: pressao 1
  val 2: pressao 2
  val 3: timer posicao (ms)
  val 4: timer pressao (ms)
  val 5: timer val (ms)
*/
struct values {
  double val[6];
};

static struct values last_value;
static void endme (int dummy) { stop_all = 1; }

MBX *mbx_val = NULL;
MBX *mbx_ppp = NULL;

int main (int argc, char **argv)
{
  if (!(device = comedi_open (filename)))
  {
    printf("CANNOT OPEN COMEDI DEVICE\n");
    exit (0);
  }

  RT_TASK *Main_Task;
  if (!(Main_Task = rt_task_init_schmod (nam2num ("MNTSK"), PTY_MAIN, 0, 0,
SCHED_FIFO, 0xFF)))
  {
    printf ("CANNOT INIT MAIN TASK\n");
    exit (1);
  }

  semval = rt_sem_init(nam2num("SEMAPH"), 1);
  barrier = rt_sem_init(nam2num("PREMSM"), 5);
  pthread_create (&encoder_thread, NULL, encoder_fun, NULL);
  pthread_create (&valvula_thread, NULL, valvula_fun, NULL);
  pthread_create (&pressure_thread, NULL, pressure_fun, NULL);
  pthread_create (&sendbox_thread, NULL, sendbox_fun, NULL);

  rt_set_one_shot_mode ();
  period = start_rt_timer (nano2count (TICK_TIME));
  rt_sem_wait_barrier(barrier);
  expected = rt_get_time () + 100 * period;

  rt_task_make_periodic (Encoder_Task, expected, period);

```

```

rt_task_make_periodic (Valvula_Task, expected, period);
rt_task_make_periodic (Pressure_Task, expected, period);
rt_task_make_periodic (SendBox_Task, expected, period);

while(stop_all!=1)
    rt_sleep(nano2count (TICK_TIME));
rt_sem_wait_barrier(barrier);

pthread_join (encoder_thread, NULL);
pthread_join (valvula_thread, NULL);
pthread_join (pressure_thread, NULL);
pthread_join (sendbox_thread, NULL);
pthread_join (looping_thread, NULL);

stop_rt_timer ();
comedi_close(device);
rt_sem_delete(barrier);
rt_sem_delete(semval);
rt_task_delete (Main_Task);
return 0;
}

static void *encoder_fun (void *arg)
{
    short int pos, i;
    unsigned char buff[2] = {0x00, 0x00};

    if (iop1(3)) {
        printf("IOPL ERROR ENCODER TASK\n");
        exit(1);
    }

    outb_p(0x00, BASESR+1); //Turn off interrupts
    outb_p(0x80, BASESR+3); //Set DLAB ON
    outb_p(0x01, BASESR+0); //SetBaudRate
    outb_p(0x00, BASESR+1);
    outb_p(0x03, BASESR+3); //8N1
    outb_p(0xC7, BASESR+2); //FIFO Control Register
    outb_p(0x08, BASESR+4); //Turn ON DTR, RTS and OUT2
    outb_p(0x7A, BASESR+0); //Zera Contador

    if (!(Encoder_Task = rt_task_init_schmod (nam2num ("ENCTSK"), PTY_ENC, 0, 0,
    SCHED_FIFO, 0xFF)))
    {
        printf ("CANNOT INIT ENCODER TASK\n");
        exit (1);
    }

    mlockall (MCL_CURRENT | MCL_FUTURE);
    rt_make_hard_real_time ();
    rt_sem_wait_barrier(barrier);
    while (!stop_all)

```

```

{
    outb_p(0x00, BASESR); //Envia Comando
    for (i=0; i<2; i++) {
        while((inb_p(BASESR+5) & 0x01) != 1);
        buff[i] = inb_p(BASESR+0);
    }
    pos = buff[1];
    pos = pos<<8;
    pos ^= buff[0];
    if(rt_sem_wait_timed(semval, nano2count(100000))) {
        last_value.position = pos;
        rt_sem_signal(semval);
    }
    rt_task_wait_period ();
}
rt_sem_wait_barrier (barrier);
rt_make_soft_real_time ();
rt_task_delete (Encoder_Task);
return 0;
}

static void *valvula_fun (void *arg)
{
    lsampl_t data;
    double dbl;

    if (!(mbx_val = (MBX*)rt_mbx_init(nam2num("VALMBX"), sizeof(dbl)))) {
        printf("CANNOT CREATE MAILBOX RCVMBX\n");
        exit(1);
    }

    if (!(Valvula_Task = rt_task_init_schmod (nam2num ("VALTSK"), PTY_VAL, 0, 0,
    SCHED_FIFO, 0xFF)))
    {
        printf ("CANNOT INIT VALVULA TASK\n");
        exit (1);
    }

    mlockall (MCL_CURRENT | MCL_FUTURE);
    rt_make_hard_real_time ();
    rt_sem_wait_barrier(barrier);

    while (!stop_all)
    {
        //espera chegar algum valor atraves de mbx - expira pouco antes
        //do timer do thread para dar tempo para enviar dados para placa
        if(!rt_mbx_receive_timed(mbx_val, &dbl, sizeof(dbl), nano2count(TIME_OUT)))
        {
            data = (dbl+5)*4096/10;
            comedi_data_write(device, subao, 0, range, aref, data);
        }
        else

```

```

    {
        data = 2048; //zero
        comedi_data_write(device, subao, 0, range, aref, data);
    }
    rt_task_wait_period ();
}

rt_sem_wait_barrier(barrier);
rt_mbx_delete(mbx_val);
rt_make_soft_real_time ();
rt_task_delete (Valvula_Task);
return 0;
}

static void *pressure_fun (void *arg)
{
    lsampl_t p1, p2;

    if (!(Pressure_Task = rt_task_init_schmod (nam2num ("PRSTSK"), PTY_PRS, 0, 0,
    SCHED_FIFO, 0xFF)))
    {
        printf ("CANNOT INIT PRESSURE TASK\n");
        exit (1);
    }

    mlockall (MCL_CURRENT | MCL_FUTURE);
    rt_make_hard_real_time ();
    rt_sem_wait_barrier(barrier);

    while (!stop_all)
    {
        comedi_data_read (device, subai, 0, range, aref, &p1);
        comedi_data_read (device, subai, 1, range, aref, &p2);

        if(rt_sem_wait_timed(semval, nano2count(TIME_OUT)))
        {
            //converta a press p/ val real!
            last_value.pressure1 = (double)(p1);
            last_value.pressure2 = (double)(p2);
            rt_sem_signal(semval);
        }
        rt_task_wait_period ();
    }

    rt_sem_wait_barrier(barrier);
    rt_make_soft_real_time ();
    rt_task_delete (Pressure_Task);
    return 0;
}

static void *sendbox_fun (void *arg)
{

```

```

struct values updated;

if (!(mbx_ppp = (MBX*)rt_mbx_init(nam2num("PPPMBX"), sizeof(last_value)))) {
    printf("CANNOT CREATE MAILBOX SNDMBX\n");
    exit(1);
}

if (!(SendBox_Task = rt_task_init_schmod (nam2num("PPPTSK"), PTY_SND, 0, 0,
SCHED_FIFO, 0xFF))) {
    printf ("CANNOT INIT SENDBOX TASK\n");
    exit (1);
}

mlockall (MCL_CURRENT | MCL_FUTURE);
rt_make_hard_real_time ();
rt_sem_wait_barrier (barrier);

while (!stop_all)
{
    rt_sem_wait(semval);
    updated = last_value;
    rt_sem_signal(semval);
    rt_mbx_send_timed (mbx_ppp, &updated, sizeof(updated), nano2count(TIME_OUT));
    rt_task_wait_period ();
}

rt_sem_wait_barrier(barrier);
rt_mbx_delete(mbx_ppp);
rt_make_soft_real_time ();
rt_task_delete (SendBox_Task);
return 0;
}

```

### A.3 Bloco do Scilab

A figura a seguir mostra o bloco desenvolvido para o ambiente Scilab/Scicos. Este bloco possui duas entradas (tempo e sinal para o servomotor) e três saídas (valores das pressões e da posição).

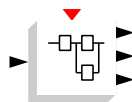


Figura A.1 – Bloco de representação do servoposicionador no sistema Scicos/Scilab