

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

ARQUITETURA MICROCONTROLADA PROGRAMÁVEL EM TEMPO REAL
APLICADA AO CONTROLE DE UM SERVOPOSICIONADOR PNEUMÁTICO

por

Celso Luís Teixeira

Dissertação para obtenção do Título de
Mestre em Engenharia

Porto Alegre, Junho 2020

ARQUITETURA MICROCONTROLADA PROGRAMÁVEL EM TEMPO REAL
APLICADA AO CONTROLE DE UM SERVOPOSICIONADOR PNEUMÁTICO

por

Celso Luís Teixeira
Tecnólogo em Automação Industrial
Licenciatura em Física

Dissertação submetida ao 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

Aprovada por:

Prof. Dr. Herbert Martins Gomes, UFRGS

Prof. Dr. Guilherme de Oliveira Kunz, Unioeste

Prof. Dr. Antônio Carlos Vardiero, UFSC

Prof. Dr. Fernando Marcelo Pereira

Coordenador do PROMEC

Porto Alegre, 18 de Junho de 2020

*“Lute com determinação, abrace a vida com paixão,
perca com classe e vença com ousadia, porque o
mundo pertencente a quem se atreve e a vida é
muito bela para ser insignificante”. Charles Chaplin*

AGRADECIMENTOS

Agradeço a Deus por me dar saúde, força, sabedoria e iluminação nos momentos difíceis.

A minha esposa Solange, pela dedicação, apoio incondicional, desprendimento, paciência, carinho, força e confiança ao longo dos anos. Sem seu apoio e incentivo este trabalho não seria realizado.

A minha filha Lívia Helena, pela incansável paciência nos meus momentos de ausência e por tornar a minha vida mais completa.

Ao meu orientador, professor Eduardo Perondi, pela lucidez, expertise, apoio e colaboração em diversas etapas deste trabalho.

A todos meus professores, que contribuíram com seus vastos conhecimentos em grande magnitude para o meu crescimento pessoal e profissional.

Aos funcionários do Laboratório de Mecatrônica e Controle da Universidade Federal do Rio Grande do Sul (LAMECC), Eduardo Basso e Eder Dorneles pelo auxílio técnico, colaboração e amizade.

Aos colegas de pós-graduação, em especial para Jusoan Lang Mór, Vinicius Bordignon, Leandro Tognon e Arivelto Fialho, por toda amizade, troca de conhecimentos, apoio, parceria, acolhimento e pelas palavras de incentivo.

A todos os meus colegas de trabalho e de profissão pelo apoio, força e incentivo, em especial a Edison Dachi e Anselmo Rafael Cukla.

Todos meus amigos, familiares e irmão, que de alguma forma participaram desta conquista, que estiveram presentes nas horas boas e ruins, que demonstraram apoio e que entenderam todos os momentos de afastamento.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema eletrônico para controle de servoposicionadores pneumáticos, baseado em sistemas de tempo real. Estes servoposicionadores apresentam diversas características atraentes para utilização em áreas como mecatrônica e robótica, como boa relação peso/potência, ausência de campos magnéticos, baixa manutenção e fácil instalação. Porém, em razão de características altamente não lineares relacionados à dinâmica dos gases e a efeitos de atrito, inerentes aos componentes do sistema, seu uso é dificultado para atividades que exigem precisão de posicionamento. O avanço da microeletrônica na construção de dispositivos com maior capacidade de processamento, periféricos integrados e facilidade de programação tem permitido a implementação de novos algoritmos e técnicas de controle sofisticadas, a fim de contornar tais não linearidades. Este trabalho apresenta uma alternativa econômica para o controle digital baseado em sistema de tempo real, a partir do uso de recursos de *software gratuitos* e componentes de baixo custo agregado. Para validação da solução proposta, a arquitetura desenvolvida foi utilizada para o controle de um servoposicionador pneumático, na qual foram programados dois algoritmos: controle por modos deslizantes (*Slide Mode Control*) e o controlador PID (proporcional, integral e derivativo). Os resultados experimentais de avaliação da variação do tempo de execução do laço de controle e do desempenho no seguimento de trajetórias polinomial e senoidal permitem aferir o desempenho da arquitetura desenvolvida.

Palavras-chave: Controle de servoposicionadores pneumáticos; Arquitetura de controle; Sistemas de tempo real; Controle microprocessado de sistemas mecânicos.

ABSTRACT

This work presents the development of an electronic system for the control of pneumatic servo-positioners, based on real-time system approach. Pneumatic servo-positioners present several attractive characteristics for their use in areas like mechatronics and robotics, such as good weight/power ratio, absence of magnetic fields, low maintenance and easy installation. However, due to highly nonlinear characteristics related to gas dynamics and friction effects, inherent to the system components, its use is difficult for activities that require positioning accuracy. The advancement of microelectronics in the construction of devices with greater processing capacity, integrated peripherals and ease of programming has allowed the implementation of new algorithms and sophisticated control techniques, in order to surpass such non-linearities. Current work presents an economical alternative to digital control based on a real-time system operation, by means of using of using free *software* resources and low-cost components. To validate the proposed solution, the developed architecture was used to control a pneumatic servopositioner, in which two algorithms were programmed: slide mode control and PID controller (proportional, integral and derivative). The experimental tests of evaluation of the of the control loop execution time variation and the performance in the tracking control of polynomial and sine trajectories allow to evaluate the performance of the developed architecture.

Keywords: Control of pneumatic servo-positioners; Control architecture; Real-time systems; Microprocessed control of mechanical systems.

ÍNDICE

1	INTRODUÇÃO	1
1.1	O Servoposicionador Pneumático	2
1.2	O Controle Digital	4
1.3	Objetivo Geral	5
1.4	Objetivos Específicos	5
1.5	Justificativa	6
1.6	Organização do Trabalho	6
2	FUNDAMENTAÇÃO TEÓRICA	7
2.1	O Estado da Arte	7
2.2	Principais Componentes dos Sistemas Servopneumáticos	11
2.2.1	Servoatuadores Pneumáticos	11
2.2.2	Servoválvulas Pneumáticas	12
2.2.3	Sensores / Transdutores	13
2.3	Controladores Digitais	16
2.3.1	Computadores Pessoais (PC)	16
2.3.2	Microcontroladores	18
2.3.3	Processador Digital de Sinais (DSP)	20
2.4	Comunicação de Dados	22
2.5	Técnicas de Controle Implementadas	24
2.5.1	Controle Linear Proporcional-Integral-Derivativo (PID)	25
2.5.2	Controle Robusto por Modos Deslizantes (SMC)	27
3	SISTEMAS DE CONTROLE EM TEMPO REAL	30
3.1	Classificação de Sistemas de Tempo Real	31
3.2	Tarefas e Políticas de Escalonamento	32
3.3	Relações de Precedência e Exclusão e Mecanismos de Sincronização e Comunicação	35
3.4	Sistemas Operacionais de Tempo Real (RTOS)	37
3.4.1	RTAI / Xenomai	40
3.4.2	QNX Neutrino	41
3.4.3	FreeRTOS	42
4	IMPLANTAÇÕES EXPERIMENTAIS	44

4.1	Introdução	44
4.2	Descrição Geral do Sistema	45
4.3	Componentes.....	46
4.3.1	Componentes do Sistema de Tratamento de Fluido.....	46
4.3.2	Componentes do Atuador Pneumático.....	48
4.3.3	Sensores e Transdutores	51
4.3.4	Fonte de Tensão Elétrica.....	53
4.3.5	Microcontrolador.....	55
4.3.6	Sistema de Aquisição de Dados	57
4.3.7	Comunicação USB	62
4.3.8	Microcomputador.....	63
4.3.9	Custos.....	63
5	ROTINAS COMPUTACIONAIS DO SISTEMA DE CONTROLE	64
5.1	Rotinas Computacionais para o Controlador Digital	64
5.1.1	Rotina de Controle	65
5.1.2	Rotina de Envio de Dados via USB	67
5.2	Configuração do Sistema Operacional e Organização das Rotinas Computacionais no Controlador Digital	68
5.3	Ambiente de Programação	69
5.4	Descrição do Ensaio Padrão.....	69
5.4.1	Controle PID	69
5.4.2	Controle SMC	71
5.4.3	Referências de Trajetória Utilizadas	73
5.4.4	Compensação da Zona Morta da Servoválvula Proporcional.....	75
6	RESULTADOS.....	77
6.1	Tempo de Execução do Laço de Controle	77
6.1.1	Controlador PID	78
6.1.2	Controlador por Modos Deslizantes (SMC)	79
6.2	Resultados Obtidos no Seguimento de Trajetórias	80
6.2.1	Controle PID	80
6.2.2	Controle por Modos Deslizantes (SMC).....	84
6.3	Discussões sobre os Resultados Obtidos.....	89
7	CONCLUSÕES	91

7.1	Trabalhos Futuros.....	92
REFERÊNCIAS BIBLIOGRÁFICAS		94
APÊNDICE A	Circuito de Condicionamento de Sinais para o MCP3208 / ESP32 ...	103
APÊNDICE B	Circuito Driver para Acionamento da Servoválvula	109
APÊNDICE C	Custos de Aquisição de Componentes.....	113

LISTA DE FIGURAS

Figura 1.1	Esquema de um servoposicionador pneumático acoplado à carga externa	3
Figura 1.2	Configuração de um sistema de controle digital	4
Figura 2.1	Exemplo de cilindro pneumático de dupla ação sem haste.....	12
Figura 2.2	Esquema de funcionamento de uma válvula.....	13
Figura 2.3	Sensor potenciométrico linear.....	15
Figura 2.4	Sensores de pressão.....	15
Figura 2.5	ESP32.....	19
Figura 2.6	Pacote de dados protocolo USB.....	23
Figura 2.7	Diagrama de blocos de um controlador PID	26
Figura 2.8	Chattering gerado pelo chaveamento das leis de controle	28
Figura 2.9	Diagrama de blocos de um controlador por modos deslizantes.....	29
Figura 3.1	Escalonamento de tarefas.....	34
Figura 4.1	Bancada de testes no LAMECC	44
Figura 4.2	Esquema da bancada experimental	45
Figura 4.3	Unidade de tratamento de ar e representação esquemática do componente ...	47
Figura 4.4	Válvula direcional proporcional	49
Figura 4.5	Cilindro atuador dupla ação sem haste	50
Figura 4.6	Régua potenciométrica.....	52
Figura 4.7	Sensor de pressão.....	53
Figura 4.8	Fonte de tensão ajustável	54
Figura 4.9	Microcontrolador ESP32	57
Figura 4.10	Diagrama de blocos do fluxo de sinais	58
Figura 4.11	Diagrama de blocos simplificado do condicionador de sinais.....	58
Figura 4.12	MCP3208	61
Figura 4.13	Diagrama de blocos do driver de potência desenvolvido	62
Figura 5.1	Distribuição Gaussiana de amostragens em um conversor AD.....	66
Figura 5.2	Fluxograma das operações da tarefa de controle	67
Figura 5.3	Fluxograma referente à tarefa de envio das informações ao computador	68
Figura 5.4	Diagrama de blocos – controle PID	70
Figura 5.5	Diagrama de blocos – controle por modos deslizantes.....	72

Figura 5.6	Trajétoria de referência senoidal.....	73
Figura 5.7	Trajétoria de referência polinomial.....	74
Figura 6.1	Tempos de ciclos de controle e taxas de ocupação de processadores para o controlador PID.....	78
Figura 6.2	Tempos de ciclos de controle e taxas de ocupação de processadores para o controlador por Modos Deslizantes	79
Figura 6.3	Seguimento de posição para o controle PID com referência senoidal.....	80
Figura 6.4	Erros de seguimento de trajetória para o controle PID com referência senoidal	81
Figura 6.5	Sinal de controle aplicado à válvula de controle direcional gerado pelo controlador PID para referência senoidal	81
Figura 6.6	Seguimento de trajetória para o controle PID com referência polinomial de 7ª ordem	83
Figura 6.7	Erros de seguimento de trajetória para o controle PID com referência polinomial	83
Figura 6.8	Sinal de controle aplicado à válvula de controle direcional gerado pelo controlador PID para referência polinomial	84
Figura 6.9	Seguimento de posição para o controle por modos deslizantes com referência senoidal	85
Figura 6.10	Erros de seguimento de trajetória para o controle por modos deslizantes com referência senoidal	86
Figura 6.11	Sinal de controle aplicado à válvula de controle direcional gerado pelo microcontrolador para referência senoidal.....	86
Figura 6.12	Seguimento de posição para o controle por modos deslizantes com referência polinomial de 7ª ordem	87
Figura 6.13	Erros de seguimento de trajetória para o controle por modos deslizantes com referência polinomial	88
Figura 6.14	Sinal de controle aplicado à válvula de controle direcional gerado pelo microcontrolador para referência polinomial.....	88
Figura A.1	Estágio para regulação de tensão elétrica	103
Figura A.2	Circuito de atenuação de sinais.....	104
Figura A.3	Amplificador diferencial com rejeição de ruído em modo comum	106
Figura A.4	Circuito filtro Butterworth implementado	106

Figura A.5	Circuito de condicionamento de sinal para sensores de pressão.....	107
Figura A.6	Atraso no sinal pelo estágio de filtro	107
Figura A.7	Diagrama completo do condicionador de sinais	108
Figura A.8	Esquema do regulador DC-DC	109
Figura A.9	Isolação e amplificação do sinal gerado pelo microcontrolador.....	110
Figura A.10	Circuito de potência em arranjo de amplificador classe AB.....	111
Figura A.11	Linearidade do sinal de saída (gráfico superior) amplificado em relação ao sinal de entrada (gráfico inferior)	112

LISTA DE TABELAS

Tabela 2.1	Quadro comparativo entre diferentes protocolos de comunicação	22
Tabela 4.1	Principais características do vaso de pressão	47
Tabela 4.2	Principais características da unidade de tratamento de ar	48
Tabela 4.3	Principais características da servoválvula direcional	49
Tabela 4.4	Principais características dos silenciadores	50
Tabela 4.5	Principais características do cilindro de dupla ação	51
Tabela 4.6	Principais características do transdutor de posição	52
Tabela 4.7	Principais características do sensor de pressão	53
Tabela 4.8	Principais características da fonte	54
Tabela 4.9	Características dos microcontroladores analisadas	55
Tabela 4.10	Principais características do ESP32	56
Tabela 4.11	Filtro de condicionamento de sinais implementado	59
Tabela 4.12	Características técnicas do MCP3208	60
Tabela 5.1	Características do filtro digital para dados de posição	71
Tabela 5.2	Características do filtro digital para dados de velocidade e aceleração	72
Tabela 6.1	Tempos de execução dos <i>looping</i> de controle em milissegundos	77
Tabela 6.2	Tempos de execução das tarefas do algoritmo PID em milissegundos	78
Tabela 6.3	Tempos de execução das tarefas do algoritmo SMC em milissegundos	79
Tabela 6.4	Relação dos resultados no erro de seguimento de trajetória	89
Tabela A.1	Características do amplificador operacional MCP6004	105
Tabela A.2	Características do <i>ampop</i> LM324	110
Tabela A.3	Custos dos componentes do controlador digital	113

LISTA DE SIGLAS E ABREVIATURAS

ABS	Anti-lock Braking System
ACK	Acknowledgment that the packet has been successfully received
AD/DA	Analógico-Digital/Digital-Analógico
ADC	Conversor Analógico-Digital
ADDR	Address
AMP	Assymmetric Multi-Processing
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CCMR	Common Mode Rejection Ratio
CI	Circuito Integrado
CPU	Central Processing United
CRC	Cyclic Redundancy Checksums
DAC	Conversor Digital-Analógico
DIN	Deutsches Institut für Normung
D.O.S	Disk Operating System
DMIPS	Milhões de instruções por segundo
DSP	Digital Signal Processing
DZC	Dead Zone Compensation
ENDP	Endpoint
FF	FeedForward
FIFO	First in-First out
FP	Fixed Priority
HD	Hard Disk
HMOS	High Speed CMOS
HOST	Hosting
I/O	Input/Output
I2C	Inter-Integrated-Circuit
IDE	Integrated Developement Environment
IIR	Infinite Impulse Response
IoT	Internet of Things

IPC	Inter-Process Communication
ISO	International Organization for Standardization
ISR	Interrupt Service Routine
LAMECC	Laboratório de Mecatrônica e Controle
LSB	Less Significant Bit
MCU	Microcontrolador
MISRA	Motor Industry Reliability Association
NAK	Data cannot be received and should be retried
NYET	Split transaction is not yet complete (not yet)
PID	Proporcional-Integral-Derivativo
P.I.D.	Product Identification
PVA	Posição-Velocidade-Aceleração
POSIX	Portable Operating System Interface
PWM	Pulse With Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RM	Rate Monotonic
ROM	Read Only Memory
RR	Round Robin
RS	Recommended Standart
RTAI	Real Time Application Interface
RTOS	Real Time Operation System
RTS	Real Time Systems
SI	Sistema Internacional de Unidades de Medida
SMC	Slide-Mode Control
SMP	Symmetric Multi-Processing
SNR	Signal to Noise Ratio
SO	Sistema Operacional
SOPG	Sistema Operacional de Propósito Geral
SOTR	Sistema Operacional de Tempo Real
SPI	Serial Peripheral Interface
STALL	Packet in response to a setup packet
UFRGS	Universidade Federal do Rio Grande do Sul

USB	Universal Serial Bus
UCP	Unidade Central de Processamento
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup

LISTA DE SÍMBOLOS

F_a	Força de atrito, N
F	Força, N
F_e	Força Externa, N
K_d	Ganho derivativo
K_i	Ganho integral
K_p	Ganho proporcional
T_d	Tempo da constante derivativa
T_i	Tempo da constante integral
p_1	Pressão na câmara 1, Pa
p_2	Pressão na câmara 2, Pa
p_{atm}	Pressão atmosférica, Pa
p_s	Pressão de suprimento, Pa
q_{m1}	Vazão mássica na câmara 1, kg/s
q_{m2}	Vazão mássica na câmara 2, kg/s
t_0	Tempo inicial, s
x_v	Deslocamento do Carretel Controlador da servoválvula, m
A	Área do êmbolo, m^2
M	Massa, kg
S	Superfície de Deslizamento
U	Tensão, V
q	Vazão, Pa
t	Tempo, s
u	Sinal de controle, V
$V(max)$	Tensão máxima aplicada a válvula, V
y	Posição da massa, m
$y(t)$	Posição da massa requerida, m
$y_a(t)$	Posição da massa requerida, trajetória senoidal, m
$y_{dp}(t)$	Posição da massa requerida, trajetória polinomial, m
\dot{y}	Velocidade, m/s

\ddot{y}	Aceleração, m/s^2
\bar{y}	Ponto médio da trajetória, m
y_{max}	Amplitude da trajetória, m
$\varepsilon(t)$	Erro, m
φ	Largura da camada limite, m
ω	Velocidade Angular, rad/s

1 INTRODUÇÃO

Devido a diversas características, como a fácil disponibilidade e armazenamento, a segurança, a facilidade de manutenção, a confiabilidade e a ausência de campos magnéticos envolvidos, o ar comprimido é uma das fontes energéticas mais utilizadas no meio industrial. A pneumática é o ramo da engenharia que estuda a aplicação de ar comprimido para converter a energia disponível no fluido em escoamento em energia mecânica [Qiong e Jiao, 2011]. A utilização da pneumática se configura como uma eficiente alternativa à eletricidade ou à hidráulica em aplicações que requerem altas velocidades e média potência para a movimentação de dispositivos industriais [Li *et al.*, 2006].

Atuadores pneumáticos são amplamente utilizados em várias aplicações industriais, particularmente naquelas que envolvem sistemas de manufatura, como *pick and place*, manipuladores robóticos e máquinas de rebiteamento [Salim *et al.*, 2014]. Porém, a utilização da pneumática é dificultada em aplicações específicas dentro das áreas da robótica e mecatrônica em função do comportamento altamente não-linear relacionado à dinâmica dos gases no interior das câmaras dos atuadores e a efeitos de atrito entre êmbolo e camisa do cilindro. Tais fatores tornam o controle de sistemas de posicionamento pneumático uma tarefa bastante difícil e limitam a aplicação dos sistemas pneumáticos, apesar de seus méritos [Hai-Peng e Huang, 2013]. Em razão das características não lineares destes sistemas e a consequente dificuldade da atuação de controladores convencionais, os dispositivos hidráulicos tiveram ao longo dos anos maior empregabilidade na indústria. Este resultado implica em uma subvalorização dos servoposicionadores pneumáticos, uma vez que o desempenho destes dispositivos não deveria ser muito inferior ao de seus equivalentes hidráulicos, como normalmente acontece [Mannetje, 1981 *apud* Sobczyk Sobrinho, 2009].

O avanço da microeletrônica e da informática somado ao advento de controladores/processadores de baixo custo, flexíveis, de alta velocidade e com alto poder de processamento computacional tem permitido aos pesquisadores o desenvolvimento de novas técnicas de programação e utilização de recursos computacionais para implementação de algoritmos sofisticados que minimizam os efeitos das não linearidades. Vieira, 1998, Perondi, 2002, Li *et al.*, 2006 e Qi, Bone e Zhang, 2019, indicam a utilização de períodos de ciclo de controle em ordem igual ou inferior a 1 ms para operação segura de dispositivos pneumáticos em função das características dinâmicas dos atuadores. Para garantir o funcionamento adequado do controlador programado, o tempo de resposta pré-definido a um evento (externo

ou interno) deve ser garantido (ou seja, deve operar em *tempo real*) [Kunz, 2006]. Outras publicações decorrentes da implementação de técnicas de controle a servoposicionadores indicam a utilização de períodos de amostragem de 2 ms [Ning e Bone, 2005] e 1.8 ms [Pandian *et al.*, 1997], fato que corrobora com a necessidade de alto desempenho de processamento computacional dos componentes de controle (*hardware/software*).

Equipamentos comerciais desenvolvidos com propósito de aplicação em dispositivos pneumáticos como a dSPACE [Dsapce, 2019] e placas de E/S multifuncionais como a NI-PCI-6221 [National Instruments, 2019], em função de seu alto valor de aquisição, permitem a implantação de tais sistemas somente em um número reduzido de aplicações industriais.

A motivação deste trabalho é o desenvolvimento de um dispositivo de controle digital para aplicação juntamente à servoposicionadores pneumáticos, com características de:

- a) baixo custo;
- b) desempenho de posicionamento considerado aceitável quando comparado aos hidráulicos e elétricos;
- c) operação em tempo real sob ciclo de controle em período de 1 ms e;
- d) arquitetura de *software* flexível e com boa confiabilidade.

Para tanto, é necessária a estruturação de uma bancada contendo os dispositivos de sensoramento, processamento computacional e de atuação para implantação de estratégias de controle tradicionais que permitam satisfazer tais premissas.

A fim de definir claramente o problema, as seções a seguir descrevem os principais conceitos no tocante a controladores digitais e a servoposicionadores pneumáticos.

1.1 O Servoposicionador Pneumático

Atuadores pneumáticos são componentes responsáveis em transformar energia pneumática em mecânicas, produzindo movimento [Moreira, 2012], sendo também o elemento responsável pela aplicação da força em uma carga mecânica [Sobczyk Sobrinho, 2009]. Sua forma de acionamento é geralmente elétrica, hidráulica ou pneumática. Atuadores pneumáticos são elementos mecânicos que, por meio de movimentos lineares ou rotativos, transformam a energia cinética gerada pelo ar pressurizado e em expansão em energia mecânica, produzindo trabalho [Fialho, 2007]. Quando um sistema é realimentado com a posição (permitindo o controle da posição) ele é denominado servoposicionador pneumático [Perondi, 2002].

Um servoposicionador (também denominado seguidor) é um sistema que possibilita o posicionamento de uma carga mecânica em uma localização desejada [Sobczyk Sobrinho, 2009], a qual pode ser fixa ou variar ao longo do tempo. O sistema representado na Figura 1.1 consiste de uma servoválvula direcional e um atuador pneumático de formato cilíndrico, possuindo duas câmaras separadas pelo êmbolo acoplado à massa M . Servoposicionadores pneumáticos são compostos basicamente por uma servoválvula que controla a vazão do ar na direção de um cilindro cujo êmbolo está ligado à massa que se deseja posicionar [Kunz, 2006]. Utiliza-se o eixo coordenado y para denotar o posicionamento da massa, que por sua vez pode ser fixo ou variável no tempo, sendo a posição requerida representada pelo termo $y(t)$.

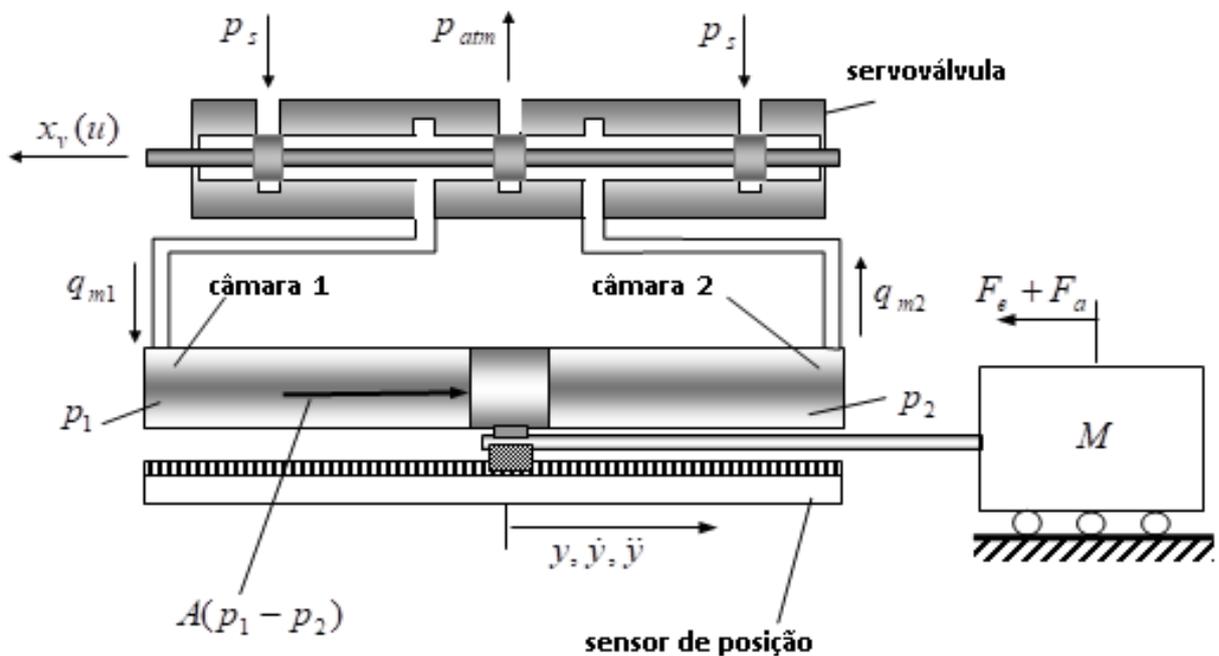


Figura 1.1 – Esquema de um servoposicionador pneumático acoplado à carga externa [Adaptado de Perondi, 2002].

No esquema representado na Figura 1.1, a servoválvula é utilizada para controlar o escoamento do ar sob pressão constante (p_s), fornecida por uma fonte de suprimento externa. Um deslocamento positivo (x_v) do carretel da válvula provocado pela aplicação de uma tensão elétrica (u) sobre um solenoide (em válvulas de acionamento elétrico), permite a conexão entre a pressão de suprimento (p_s) e a câmara 1, ao passo que a câmara 2 é conectada a atmosfera (p_{atm}). A vazão mássica (q_{m1}) na câmara 1 ocorre devido à diferença entre a pressão de suprimento (p_s) e a pressão na câmara 1 (p_1); situação análoga à vazão mássica (q_{m2}) que decorre da diferença da pressão na câmara 2 (p_2) e a atmosfera (p_{atm}). A diferença

das pressões p_1 e p_2 (p_1 tende a p_s e p_2 tende a p_{atm}) no interior do cilindro produz uma força F que resulta no deslocamento positivo $y(t)$ do êmbolo, e conseqüentemente da massa M . Esta força pode ser modelada pelo equilíbrio dinâmico entre as forças, a partir da 2ª Lei de Newton, conforme Equação 1.1:

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

onde F_a é a força de atrito, F_e é a força externa, A é a área do êmbolo, \ddot{y} é a aceleração da massa e $A(p_1 - p_2)$ é a força resultante da diferença de pressão entre as câmaras 1 e 2.

1.2 O Controle Digital

Para controlar um sistema físico a partir de um controlador digital, o *hardware* de controle deve receber medições do sistema, processá-las e enviar o sinal de controle para o atuador que efetua a ação de controle. Normalmente a planta é analógica, o controle é constante por partes e a ação de controle é atualizada periodicamente [Fadali e Visioli, 2013]. A Figura 1.2 representa o esquema clássico de um sistema de controle digital.

A precisão de movimentos em um sistema de seguimento de trajetórias exige a utilização de dispositivos processadores capazes de executar operações computacionais de aquisição e processamento de dados em um ciclo de controle, cujo período de tempo deve estar adequado à dinâmica do sistema físico controlado. Castrucci e Sales, 1990, afirmam que diminuindo o período de amostragem do sistema, o seu comportamento “tende” ao de sistemas analógicos, mas isto implica em maiores custos de conversão e computação.

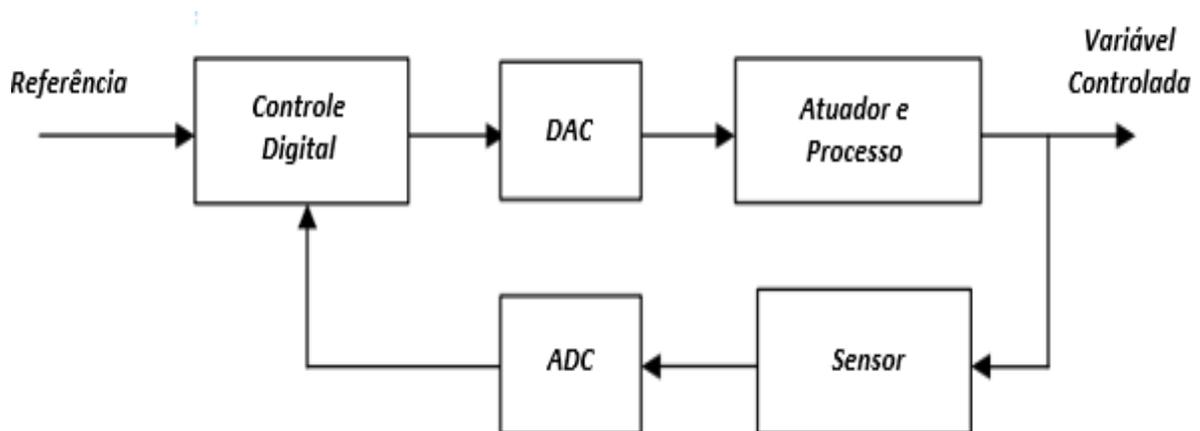


Figura 1.2 – Configuração de um sistema de controle digital [Adaptado de Fadali e Visioli, 2013].

Embora tenha por desvantagem a exigência de bons conhecimentos em desenvolvimento de *firmware* e/ou lógica digital por parte do programador/desenvolvedor, a utilização de controladores digitais agrega inúmeras vantagens, como melhor imunidade a ruídos, confiabilidade de execução controle em malhas complexas, comunicação entre diferentes dispositivos, interfaces gráficas, flexibilidade para implementação de controladores robustos baseados e estratégias de controle modernas, que por sua vez, permitem a compensação das não linearidades observadas nos sistemas pneumáticos.

1.3 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento e implementação de um controlador eletrônico digital para o controle de um servoposicionador pneumático. Este sistema deverá ser baseado no uso de um microcontrolador comercial de baixo custo, que tenha operação de controle que utiliza os recursos disponíveis em sistemas operacionais de tempo real de código aberto e gratuito e que possa ser programado (desenvolvimento de *firmware*) a partir de ambiente de desenvolvimento integrado (IDE – *Integrated Development Environment*) amigável.

1.4 Objetivos Específicos

Os principais objetivos específicos deste trabalho são:

- a) estruturar e instrumentar com sensores de pressão e posição, uma bancada de testes para um servoposicionador pneumático;
- b) projetar e desenvolver interfaces eletrônicas (*hardware*) de condicionamento de sinais para o microcontrolador, sob demanda do controle do servoposicionador;
- c) analisar e validar ferramentas computacionais de sistemas em tempo real para programação de estratégias de controle tradicionais;
- d) realizar testes de validação da capacidade do sistema em garantir a manutenção do tempo de execução do *looping* de controle para seguimento de trajetórias.

1.5 Justificativa

Embora o presente trabalho não tenha por objetivo o desenvolvimento ou aprimoramento de técnicas de controle avançadas para o controle de servoposicionadores pneumáticos, a implantação de um controlador digital desenvolvido sob regras de sistema de tempo real, baseado em algoritmos de controle tradicionais e de fácil programação, é importante no tocante à ampliação das possibilidades da utilização de sistemas pneumáticos em ambientes industriais.

1.6 Organização do Trabalho

Nos Capítulos 2 e 3 é apresentada uma revisão bibliográfica sobre os conhecimentos teóricos necessários para o controle de servoposicionadores pneumáticos. O Capítulo 4 são destacadas as implementações experimentais realizadas neste trabalho como o desenvolvimento do *hardware* para implementação da bancada pneumática. No Capítulo seguinte (5) é descrito o desenvolvimento de *software* em tempo real para aplicação de estratégias de controle lineares e robusta. No Capítulo 6 são apresentados os resultados e avaliação do desempenho dos algoritmos de controle desenvolvidos sob a interface do controlador digital. Por final, no Capítulo 7 são apresentadas as conclusões e as propostas de futuras implementações.

2 FUNDAMENTAÇÃO TEÓRICA

O grande número de pesquisas na área de controle para servoposicionadores pneumáticos reflete diretamente os avanços do desenvolvimento tecnológico ocorrido nas últimas décadas. Este capítulo inicia com o *estado da arte*, onde se procura contextualizar o presente trabalho quanto ao atual estágio do desenvolvimento de soluções para o controle destes sistemas, a partir da apresentação de uma breve descrição dos trabalhos importantes que empregam modelagem matemática, diferentes estratégias e arquiteturas de controle (*software* e instrumentação). Posteriormente, é apresentada a fundamentação teórica referente aos tópicos teóricos básicos para o entendimento dos capítulos seguintes.

2.1 O Estado da Arte

O objetivo geral do controle realimentado é a utilização do princípio de realimentação para que a variável de saída de um processo dinâmico siga a variável de referência desejada com precisão, independentemente da trajetória da variável de referência, perturbações externas ou de qualquer outra alteração na dinâmica do processo [Franklin, Powell e Emani-Naeini, 2013].

A maioria dos sistemas dinâmicos podem ser caracterizados por um conjunto de equações diferenciais que representam o modelo matemático do sistema [Ogata, 1985] podendo estas assumir muitas formas distintas, sendo que o modelo matemático envolve a aplicação criteriosa de leis físicas e de julgamento de Engenharia para obtenção de um conjunto de equações que irão (dentro de uma certa aproximação) descrever adequadamente o comportamento do sistema [Felício, 2010]. Conforme Suzuki, 2010, todo sistema físico é, em alguma medida, não-linear e difícil de obter um modelo que o represente por completo. Por causa disto, utilizam-se hipóteses que, sem comprometer a eficácia do modelo matemático desenvolvido, visam a simplificar a modelagem, diminuir o esforço do processamento computacional e facilitar o entendimento do comportamento do sistema físico. Felício, 2010, afirma que modelos mais simples geralmente produzem resultados menos precisos, entretanto a imprecisão relativa de um dado modelo é aceita por conta da contrapartida desejável, que é a obtenção rápida da visualização dos aspectos importantes do sistema. Modelos relativamente compactos são também geralmente aconselhados nas atividades de síntese de algoritmos de controle baseados em modelos, pois estes modelos

permitem a programação dos mesmos em *hardware* de controle com menor desempenho e disponibilidade de recursos (e, portanto, mais baratos).

Neste sentido, a modelagem matemática de servoposicionadores visando à aplicação em controle é abordada por diferentes autores, geralmente baseada no equacionamento da dinâmica da vazão mássica do gás comprimido através dos orifícios e da dinâmica do subsistema mecânico, expressa por meio da Segunda Lei de Newton (considerando a combinação de forças inerciais e de atrito). A modelagem das vazões pode ser obtida a partir leis de conservação de energia ou da continuidade [Maré, Colin e Geider, 2000], considerando usualmente condições isentrópicas no equacionamento do comportamento das vazões mássicas de entrada e saída durante a atuação da válvula, e isotérmicas para equacionamento das pressões nas câmaras de um atuador em função da vazão mássica e da velocidade translacional do pistão [Kain, 1973; Scavarda, 1994 *apud* Maré, Colin e Geider, 2000; Virvalo, 1995; Locateli, 2011; Richter e Hurmuzlu, 2000]. Segundo Cukla, 2012, em estratégias de controle baseadas em modelos, onde a precisão dos mesmos torna-se decisiva, para obtenção de um modelo próximo da realidade pode-se utilizar mapeamentos estáticos tridimensionais empíricos da vazão mássica em relação às pressões e à abertura da válvula em função do sinal de controle aplicado, como feito em Perondi, 2002. Ainda de acordo com Cukla, 2012, polinômios de 2º ou 3º grau são usados para interpolação do mapeamento estático e modelos matemáticos que utilizam essa abordagem para analisar o comportamento dos gases no interior das câmaras forneceram resultados experimentais aproximados aos das simulações computacionais baseadas nos modelos teóricos.

Em relação à utilização de estratégias de controle, os estudos realizados em diferentes trabalhos demonstraram uma variedade de abordagens com relação à aplicação de controladores baseados em técnicas lineares, como, por exemplo, a utilização do controlador PID (proporcional, integral e derivativo) [Cukla, 2012; Gentile *et al.*, 2002]. Conforme Sobczyk Sobrinho, 2009, controladores lineares são influenciados por variações paramétricas e efeitos dinâmicos não modelados, constituindo isso em uma limitação muito importante para a sua aplicação em sistemas de ordens elevadas e não lineares, como os sistemas servopneumáticos. Com o objetivo de contornar estes problemas, algumas aplicações de variações do controle linear podem ser observadas na bibliografia, como a utilização de controlador PD (proporcional-derivativo) com Compensação de Histerese [Chiu e Leung, 1988], ou de um Controlador PID não-linear (N-PID) [Salim *et al.*, 2014]. Somam-se à utilização de estratégias não lineares, os trabalhos desenvolvidos para

aplicação de Controle por Realimentação de Estados [Perondi; Guenter, 1999], Linearização por Realimentação [Suzuki, 2010] e Realimentação de Estados com Compensação de Zona Morta de válvula proporcional [Ning e Bone, 2005], Lógica *fuzzy* [Lara-Rojo et.al., 2001 *apud* Kunz, 2006; Najjari *et al.*, 2014], Controle por Modos Deslizantes [Perondi e Guenter, 2000; Perondi, 2002; Pandian *et al.*, 1997; Gyeviski *et al.*, 2005] e controle por Redes Neurais [Gervini, 2014; Gross e Rattan, 1997 *apud* Kunz, 2006].

Com relação à arquitetura de controle, diversos trabalhos apontam a utilização de plataformas comerciais baseadas em Digital Signal Processing (DSP - Processador Digital de Sinais, em português) de controle e aquisição, vinculada à plataforma de programação em alto nível, como a dSPACE [Perondi, 2002; Suzuki, 2010; Locateli, 2011; Schlüter, 2018; Rad e Hancu, 2017]. Visando à implementação e utilização de *hardware* de controle e aquisição alternativo, por motivos relacionados à redução de custos, compactação de sistemas ou melhoria em componentes do sistema de controle, trabalhos como o de Kunz, 2006 e Rios, 2009 focaram a utilização de computadores formato IBM-PC¹ como unidade de processamento.

No trabalho de Kunz, 2006, um computador de baixo desempenho foi utilizado para processamento de dados provindos de sensores de posição e de pressão, por meio de uma interface AD/DA de aquisição de dados e circuito microprocessado dedicado, além de enviar sinais analógicos de controle para uma servoválvula. Em conjunto com o sistema operacional nativo do computador (GNU/Linux) foi utilizado um subsistema para operação de rotinas programadas pelo usuário em tempo real, denominado Real Time Application Interface (RTAI), conforme será descrito em detalhes na Seção 3.4.1 [RTAI, 2019]. Destaca-se neste subsistema a ausência de interface gráfica, o que implica a necessidade de bons conhecimentos em linguagem C por parte do programador para a implementação de rotinas computacionais. A partir da execução de algoritmo para aquisição de dados com requisito temporal para execução de laço de processamento determinado em 1 ms, foi possível a construção de curvas estáticas experimentais para enchimento e para exaustão, referentes a vazão e pressão nas câmaras do atuador pneumático, em função da tensão aplicada à servoválvula e ao tempo, além da identificação do comportamento do atrito por imposição de velocidades ao atuador pneumático.

Rios, 2009, utiliza no seu trabalho algoritmo de controle programado em linguagem C, diretamente sobre o sistema operacional (DOS 6.2.2). A interface física é dada por três placas

¹ Sigla para computadores pessoais.

coletoras de dados para fechamento da malha de controle em uma aplicação de controle realimentado de posicionamento em um robô cartesiano pneumático controlado por válvulas solenoides rápidas. Considerando a proposta de composição de um equipamento de baixo custo de implementação, um computador já obsoleto na época dos testes foi escolhido para a execução de rotinas de controle (neste caso, controle por modos deslizantes) para cada um dos eixos do robô (X, Y e Z). Em função da frequência de atualização de dados proporcionada pelo computador ser de 1/13 avos de segundo, a utilização de placas de aquisição de dados com frequência de aquisição em 4 kHz, interligadas à unidade de processamento (CPU) por meio de protocolo RS422² possibilitou a aquisição de dados com frequência adequada à implementação. Os resultados de precisão de posicionamento dentro da faixa estabelecida pela camada limite, na ordem de milímetros, demonstram a qualidade do desempenho do sistema proposto.

Mais recentemente, Cukla, 2012, apresenta o desenvolvimento de uma plataforma de controle de um servoposicionador pneumático baseada em DSP-PIC para aquisição de dados, processamento do algoritmo de controle e acionamento da válvula de controle direcional para o controle de um servoposicionador pneumático linear. O desenho do dispositivo controlador proposto neste trabalho engloba interfaces de condicionamento de sinais para sensores e atuadores, além da utilização de um processador digital de sinais em formato de circuito integrado, de baixo custo e adequada capacidade de processamento computacional. Mesmo sem a utilização de sistemas de tempo real para garantia da rigidez de temporizações de laço de controle, a comparação dos resultados de erro de seguimento de trajetórias em diferentes referências, para duas diferentes estratégias de controle (neste caso, PID e controle por modos deslizantes) implantadas também em uma plataforma comercial (dSPACE), apresentam significativa similaridade em termos de comportamento dinâmico do sistema em malha fechada. Observa-se também significativa semelhança quanto aos sinais de controle gerados pelas duas diferentes plataformas. Os resultados obtidos correspondentes ao erro máximo de regime e dinâmico de posicionamento em relação à referência (na ordem de centímetros) validam a utilização de DSP para a aplicação em controle de sistemas pneumáticos.

² Especifica características elétricas para transmissão de sinais digitais.

2.2 Principais Componentes dos Sistemas Servopneumáticos

O sistema servopneumático de controle de posição é um sistema que possibilita o posicionamento de uma carga mecânica a partir do movimento de um atuador em função das diferenças das pressões em suas câmaras [Locateli, 2011]. Geralmente, o sistema (clássico) é constituído por um servoatuador pneumático, uma válvula de controle direcional, sensores/transdutores que medem a posição do cursor do atuador e opcionalmente as pressões internas das câmaras, além de um controlador. Nos tópicos a seguir são descritos os componentes mais importantes deste sistema.

2.2.1 Servoatuadores Pneumáticos

Atualmente, os servoatuadores pneumáticos desempenham um papel fundamental no segmento industrial, sendo amplamente utilizados para movimentos de partes móveis de máquinas, manipulação ponto-a-ponto de materiais e peças e em dispositivos de segurança, dentre diversas outras aplicações. São dispositivos que transformam a energia do ar ou gás sob pressão no interior das câmaras em energia mecânica de movimento (energia cinética). Sob as mais diversas especificações de carga e forma, os atuadores pneumáticos são construídos de acordo com normas técnicas DIN 24335³ e ISO 6431⁴, que garantem a segurança no manuseio e utilização [Parker, 2019].

No tocante à forma de operação destes dispositivos, os mesmos se dividem em três segmentos: atuadores lineares (produção de movimento linear), atuadores rotativos (através de momento torsor contínuo) e oscilantes (momento torsor limitado por posições angulares definidas). Os atuadores lineares se diferem com relação à sua forma de atuação: atuadores simples ação ou ação unidirecional, cujo movimento de ação é controlado a partir de uma válvula direcional com aplicação de ar sob pressão em uma única câmara, produzindo movimento em um único sentido, seja para o avanço ou retorno, sendo o movimento de retorno à posição de descanso realizado por meio de uma mola de reação interna ao componente; e atuadores de dupla ação, como os utilizados nos trabalhos desenvolvidos por Kunz, 2006 e Cukla, 2012, com movimentos de avanço e retorno controlados por uma válvula direcional pela aplicação de ar sob pressão em uma das duas câmaras do cilindro individualmente.

³ Pneumatic equipment; pneumatic cylinders, double acting, mounting dimensions, requirements.

⁴ Substituída pela ISO 15552:2004 - Potência de fluido pneumática - Cilindros com suportes removíveis, séries de 1 000 kPa (10 bar), furos de 32 mm a 320 mm - Dimensões básicas, de montagem e de acessórios.



Figura 2.1 – Exemplo de cilindro pneumático de dupla ação sem haste [Adaptado de Festo, 2019].

Os atuadores pneumáticos lineares são apresentados sob diversas formas de construção, adaptando-se à necessidade de operação, destacando-se: atuadores com haste (com ou sem amortecimento), de haste passante, de membrana, tandem ou *duplex*, *duplex* geminado, telescópico, dentre outros.

Servoatuadores sem haste (Figura 2.1) são utilizados para cursos de movimentação de carga relativamente longos e/ou quando é identificada a possibilidade de flambagem da haste de um atuador em operações de aplicação de força ou posicionamento.

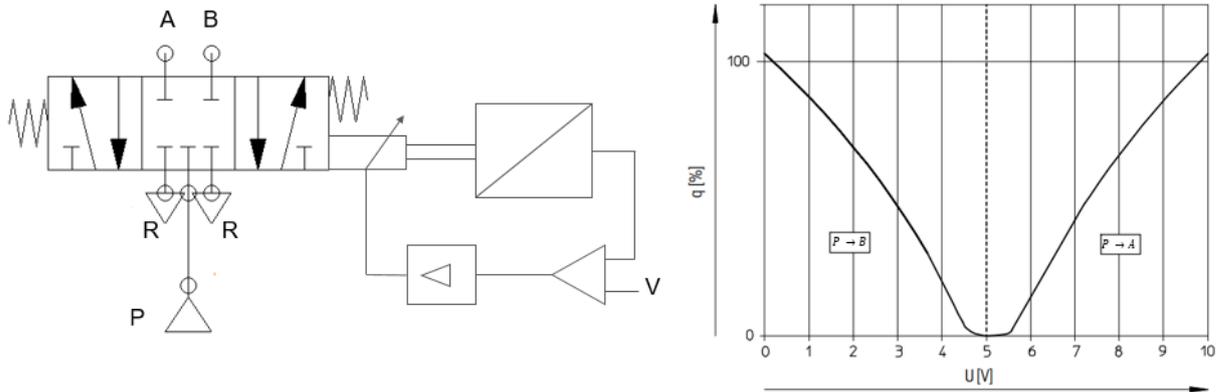
Também são utilizados em situações de restrição de espaço, ou seja, nas situações em que a haste apresentaria interferência com relação aos demais componentes do sistema mecânico.

2.2.2 Servoválvulas Pneumáticas

Os sistemas servopneumáticos de controle de posição são, na sua grande parte, acionados por meio de válvulas proporcionais direcionais. Podem ser citadas diversas aplicações para estes sistemas, como a manipulação e movimentação de materiais, na robótica, nas linhas de montagem, dentre outros [Virvalo 1995 *apud* Locateli, 2011].

As válvulas de controle proporcionais são elementos destinados para comandos de sistemas de alta performance, em função de sua rápida resposta e precisão de controle [Fialho, 2007]. A servoválvula proporcional é projetada para efetuar um controle preciso da vazão de ar que passa às câmaras do cilindro, através de um sinal de tensão ou corrente aplicado à válvula [Saravanakumar, 2017]. Basicamente, o funcionamento de uma válvula proporcional consiste na conversão de um sinal analógico (de corrente ou tensão) em uma determinada vazão proporcional ao sinal elétrico de entrada.

A Figura 2.2a mostra o esquema de funcionamento de uma válvula direcional 5 vias e 3 posições, que permite o controle da direção da vazão e a curva característica (praticamente simétrica, porém não linear) de vazão q (%) pela tensão elétrica aplicada U (V).



(a) Representação simbólica da válvula proporcional

(b) Curva característica das válvulas proporcionais de controle de vazão

Figura 2.2 – Esquema de funcionamento de uma válvula [Adaptado de Fialho, 2007; Festo, 2019].

O gráfico da Figura 2.2b indica a posição de repouso da válvula (vazão nula em função do bloqueio das vias) quando alimentada com 5 V. Com a aplicação de um sinal de tensão em rampa de 5 a 0 V, ocorre a abertura da válvula para a conexão dos pontos de P (pressão) para o ponto de consumo B (vazão proporcional ao sinal até seu valor máximo) ao passo da conexão do ponto A para R (atmosfera).

De forma análoga, a aplicação de sinal em rampa de tensão de 5 a 10 V ocasiona a abertura da válvula para a conexão dos pontos de P (pressão) para o ponto de consumo A e a conexão do ponto B para R (atmosfera).

2.2.3 Sensores / Transdutores

De acordo com Thomazini e Albuquerque, 2005, sensor é o termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente. Os sensores são elementos que executam a conversão de uma grandeza medida (pressão, temperatura, vazão, por exemplo) em uma grandeza elétrica (corrente ou tensão).

Um transdutor é um dispositivo utilizado para transformar uma forma de energia em outra, podendo essa segunda ser efetivamente utilizada por um dispositivo de controle.

Figliola e Beasley, 2011, afirmam que um transdutor converte informações obtidas de um sinal detectável. O sinal pode ser mecânico, elétrico, óptico ou pode assumir qualquer outra forma que possa ser gravada de maneira significativa.

2.2.3.1 Sensores de Posicionamento

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 [Kunz, 2006]. Neste sentido, diversas tecnologias foram desenvolvidas para o sensoriamento de posição linear. Nos trabalhos listados na bibliografia, as tecnologias mais utilizadas são a da régua potenciométrica e da régua óptica (*encoder óptico*). Em trabalhos como os de Ning e Bone, 2005; Pandian *et al.*, 1997 e Kunz, 2006, utilizou-se *encoder* óptico para obtenção da posição do atuador pneumático. Basicamente, tais componentes são subdivididos em duas categorias: os *encoders* absolutos - que indicam a posição real (absoluta) do objeto a partir de uma referência, sendo à saída do sensor um sinal digital em padrão de código binário; e os *encoders* incrementais, que indicam a posição relativa ao ponto de ativação do sensor (*encoder* é zerado a cada desligamento), sendo o sinal de saída um trem de pulsos, que necessita de um contador digital de pulsos para determinação do sentido do movimento e a distância total percorrida. Dentre as vantagens da utilização deste sensor, destacam-se a obtenção valores de velocidade e posição de um deslocamento linear em formato digital; destaca-se como desvantagem o alto custo do componente. No caso de sensores rotativos existe a necessidade de transformação de movimento translacional em torcional, e, no caso de sensores ópticos lineares, existe, por sua vez, a necessidade de uma estrutura de base com alta rigidez e um alinhamento de alta precisão com relação ao deslocamento translacional do pistão.

Sensores potenciométricos lineares, como os utilizados nos trabalhos de Cukla, 2012, Perondi, 2002 e Vieira, 1998, funcionam como um potenciômetro eletroeletrônico deslizante, o qual proporciona um sinal de saída de tensão proporcional à posição do cursor acoplado ao pistão (princípio da divisão de tensão resistiva). A principal desvantagem da utilização deste componente implica na necessidade de técnicas de conversão analógico-digital (conversão ADC) passíveis de ruídos e distorções provindos da perda de sinal (durante o processo de quantização numérica), sendo que as vantagens englobam o reduzido custo quando comparado aos *encoders ópticos* (estimado em torno de 30% do valor de um *encoder* absoluto) e a não necessidade de conjunto de acoplamentos por polia - correia ao atuador pneumático. A Figura 2.3 apresenta um sensor potenciométrico linear.



Figura 2.3 – Sensor potenciométrico linear [Adaptado de Festo, 2019].

2.2.3.2 Sensores de Pressão

Os sensores de pressão utilizados em aplicações industriais podem ser fabricados a partir de diferentes técnicas que medem a força aplicada sobre uma unidade de área. Tais tecnologias podem envolver a utilização de células de carga (*strain gauges*), cristais piezoresistivos e propriedades elétricas como a capacitância, por exemplo.

Segundo Thomazini e Albuquerque, 2005, transdutores de pressão piezoelétricos têm seu princípio de funcionamento baseado nas propriedades piezoelétricas do cristal de quartzo que, quando deformados elasticamente, geram um potencial elétrico em seus terminais por meio de determinado plano cristalográfico (disco de quartzo). Ainda de acordo com os autores, a tensão elétrica fornecida por este dispositivo é muito pequena ($1mV$ por kg/cm^2), por isso o elemento piezoelétrico é sempre conectado à um amplificador eletrônico. A Figura 2.4 apresenta alguns modelos de sensores de pressão industriais.



Figura 2.4 – Sensores de pressão [Adaptado de Wika do Brasil, 2019].

2.3 Controladores Digitais

A utilização de sistemas digitais para implantação de controle de processos tem ganhado amplo espaço frente aos controladores analógicos tradicionais em função de diversos aspectos, dentre os quais se destacam a flexibilidade de programação, que facilita o desenvolvimento; a possibilidade de reutilização de recursos e o uso de interface amigáveis com usuário. Segundo Carvalho, 2000, a estrutura de um sistema controlado por computador não difere daquela de um controlador analógico. Nise, 2002, afirma que em muitos sistemas modernos, o controlador (ou compensador) é um computador digital.

2.3.1 Computadores Pessoais (PC)

A partir da década de 1980, a comercialização em escala global dos primeiros computadores pessoais em formato IBM-PC implementados sob arquitetura *Reduced Instruction Set Computer* (RISC - computação com conjunto reduzido de operações) culminou na popularização dos sistemas de informática em larga escala, atingindo também o âmbito industrial. A possibilidade de implantação de sistemas sofisticados de aquisição, monitoramento e gerenciamento de dados a partir de computadores (unidades de controle local) que interagem diretamente com o processo, interligados a uma unidade de processamento central (estação de operações) em um barramento de dados comum (rede – *bus*) e com protocolo de comunicação de alta velocidade, deu origem à arquitetura denominada controle distribuído, que, conforme Gordillo, 2004 *apud* Cukla, 2012, é uma arquitetura recomendada para supervisão de plantas em diferentes processos.

Diversos trabalhos com o objetivo de controle de posicionamento apontam para a utilização de sistemas baseados em microprocessamento através de computadores pessoais (compatíveis com formato IBM-PC). Por exemplo, Rios, 2009, utiliza um computador como unidade central de controle acoplado a uma interface eletrônica de potência (*driver*) para acionamento de válvulas direcionais pneumáticas do tipo ON-OFF, objetivando o controle de posição em um robô cartesiano pneumático. Conforme já comentado, a programação é realizada diretamente no sistema operacional (D.O.S. 6.22) com resolução temporal de 77 milissegundos. Ning e Bone, 2005, utilizam um computador para execução de dois diferentes algoritmos de controle (PVA+FF+DZC, sigla de controle pressão/velocidade/aceleração, realimentação de estados e compensação de zona-morta,

respectivamente e SMC, sigla de controle por modos deslizantes ou *sliding mode control*), programados em linguagem C em laço de controle com intervalo de tempo de 2 ms, acoplado a uma interface eletrônica (*driver*) de controle para uma servoválvula proporcional, a fim de comparação dos resultados de precisão de posicionamento de um cilindro pneumático orientado na direção vertical, sob diferentes aplicações de carga mecânica. Em Pandian *et al.*, 1997, um algoritmo de controle baseado em modos deslizantes é implementado em um computador IBM-PC compatível programado em linguagem C com laço de controle com intervalos de 0,18 ms.

A partir da década de 2000, formatos de computadores miniaturizados de custo reduzido, chamados de *singleboards*, dotados de sistema operacional livre (baseado em *kernel* Linux) e com poucos recursos em comparação a um computador formato IBM-PC, porém com grande capacidade de processamento computacional (múltiplos núcleos de processamento) e periféricos comuns aos de microcontroladores para manipulação de dispositivos (entradas/saídas digitais, interfaces de comunicação, conversão analógico-digital, *Pulse With Modulation* (PWM)) ganharam espaço como unidade de processamento computacional no desenvolvimento de sistemas de controle. Com o objetivo de determinar das características não lineares do atrito em atuadores pneumáticos, Felix, 2018, utiliza uma placa em formato de *singleboard* denominada *Raspberry Pi* modelo 3 como interface eletrônica para aquisição de dados. No trabalho de Deimel, Radke e Brock, 2016, é utilizada uma unidade de controle de tempo real em placa única (*singleboard*), chamada *Beaglebone Black*, para o controle de manipuladores robóticos pneumáticos do tipo *soft-hand* com ciclo de controle de 2 ms, análogos a mãos humanas, utilizados em robôs colaborativos para manipulação de objetos.

No presente trabalho, a possibilidade de implantação de uma arquitetura de controle baseada em *singleboard* como unidade de processamento foi avaliada e testada; porém as diversas incompatibilidades relacionadas às versões de sistemas operacionais destas placas com versões de *kernel* de sistema operacional de tempo real dedicado (Seção 3) em relação às versões de *hardware* disponíveis hoje no comércio nacional inviabilizaram tal opção. Assim, em trabalhos futuros, dependendo da atualização do mercado nacional, esta possibilidade deverá ser considerada para seguimento do desenvolvimento de uma plataforma em tempo real adequada ao controle de servossistemas pneumáticos.

2.3.2 Microcontroladores

A escolha de uma solução que envolva controle digital pode ser muito distinta, de acordo com o resultado esperado. Fatores impactantes, como a adequação de custos de projeto, capacidade de processamento computacional em operação com restrição temporal ou diminuição/miniaturização do *hardware* necessário, devem, geralmente, ser considerados. Os microprocessadores são geralmente utilizados em aplicações de alto desempenho onde o custo e tamanho do *hardware* não são críticos [Kunz, 2006]. A tendência natural no desenvolvimento de projetos com soluções que englobem em um mesmo circuito integrado (CI) diversas funcionalidades necessárias ao projeto, tornam os microcontroladores uma alternativa atraente dentre as diversas opções de *hardware* disponíveis.

Um microcontrolador é, basicamente, um computador de um único CI, proveniente da evolução tecnológica dos circuitos lógicos digitais em função do aumento de sua complexidade e necessidade de miniaturização. O dispositivo pode ser programado a partir de um conjunto de operações organizadas e sequenciais para execução de funções específicas voltadas ao controle; dentre elas, destaca-se o controle de periféricos e dispositivos físicos. Segundo Haupt e Dachi, 2016, microcontroladores são circuitos integrados que contêm funções de unidade central de processamento (UCP), memória, entrada e saída de dados, além de possuírem incorporados à sua arquitetura interna, memória (RAM e ROM) e alguns periféricos (interface serial, PWM, contadores, temporizadores, etc.). A velocidade de *clock* é geralmente na faixa de 4 a 48 MHz, as quais, embora muito menores do que a de processadores digitais encontrados em computadores pessoais é adequada para diversas aplicações de controle, pois muitas vezes nestas aplicações não é necessária uma grande capacidade de processamento, precisando apenas da utilização de instruções simples que executem operações rapidamente.

Para suprir exigências de projetos, existe uma ampla gama de microcontroladores disponíveis no mercado, diferenciando-se principalmente no número de entradas e saídas Input/Output (I/O) e pelos recursos periféricos do dispositivo [Oliveira Neto, Monteiro e Queiroga, 2012]. Um dos primeiros microcontroladores comerciais de uso genérico, o 8051, fabricado originalmente pela Intel[®] no início da década de 1980 é um controlador de 40 pinos e tecnologia HMOS, dotado de um CPU de 8 bits de arquitetura RISC, com velocidade de operação de 12 MHz, 4 KB de memória de programa e 128 bytes de memória de dados, cujo encapsulamento agrupa recursos como conversores analógico-digital e digital-analógico (AD/DA), geradores de PWM, 4 portas de comunicação serial, *timers* internos de 16 bits,

além das tradicionais saídas e entradas digitais. Este componente passou por diversos aprimoramentos desde a sua criação, podendo ser encontrado atualmente em versões comerciais com velocidades de operação de até 60 MHz e com maior capacidade de memória. Outros fabricantes, como a Microchip[®], Zilog[®], Atmel[®] (atualmente, é subsidiária da Microchip), Philips[®] e Motorola[®] também dispõem de componentes com o mesmo propósito, mais avançados tecnologicamente e com operação de alto desempenho, que com barramentos de 16, 32 e até 64 bits, somado a recursos extras como conversores ADC de até 24 bits, interface serial e Universal Serial Bus (USB), suporte à protocolos I²C, Controller Area Network (CAN) e Serial Peripheral Interface (SPI), além de *whatchdog timers*, combinam baixo custo e grande disponibilidade comercial como atrativos para sua utilização.

Nos últimos anos, o desenvolvimento de microcontroladores tipo *multi-core*, que possuem em um mesmo encapsulamento dois ou mais núcleos (CPU) para multiprocessamento assimétrico (AMP) ou simétrico (SMP) emergiram associados ao crescente desenvolvimento de aplicações que envolvem execução de rotinas periódicas, com processamento intenso de dados e baixos tempos de resposta, além dos limites físicos relacionados ao desempenho, como densidade de potência ou limitações para aumento de frequência de operação. Estes modelos de microcontroladores realizam execuções de rotinas síncronas, de forma direcionada e paralela, de tarefas distintas que podem ser programadas de acordo com a especificidade de cada arquitetura de núcleo. Neste contexto, se destaca o microcontrolador *dual-core* baseado em arquitetura Harvard, da fabricante *Espressif Systems* modelo ESP32 (Figura 2.5), que alia baixo consumo de energia, dimensões reduzidas e baixo custo de aquisição com amplos recursos periféricos integrados como *Wi-fi*, Bluetooth, 14 portas ADC, frequência de *clock* em 240 MHz (além dos recursos descritos na Seção 4.3.5).



Figura 2.5 – ESP32 [Adaptado de Espressif, 2019].

Este dispositivo é amplamente utilizado em desenvolvimento de aplicações em *internet of things* (IoT). A sincronia de operação de tarefas dos microcontroladores com mais de uma CPU (*multi-core*) utiliza um sistema de memória compartilhada, na qual as informações são inseridas em fila, podendo ser acessadas em qualquer núcleo de maneira independente. Este mecanismo de troca de mensagens por fila, denominado *inter-processor messaging* (IPC) é característico de sistemas operacionais de tempo real (Seção 3.3), comumente utilizado para implementação de rotinas operacionais nestes componentes.

Embora uma solução baseada em microcontroladores possa ser considerada conservadora sob o ponto de vista de inovação, a escolha por estes dispositivos conduz a enormes vantagens do ponto de vista operacional, dentre as quais se destacam o baixo custo e consumo de energia, a facilidade de programação, possibilidade de reprogramação de *firmware*, redução da necessidade de desenvolvimento de circuitos para componentes eletrônicos auxiliares, além da facilitada manutenção. O uso de microcontroladores na área de controle consiste em uma solução adequada aos servoposicionadores pneumáticos devido à sua flexibilidade, miniaturização de componente e capacidade adequada para algumas tarefas específicas [Cukla, 2012]. Tais equipamentos, quando implementados como *controladores dedicados*, atuam isoladamente como uma única central de processamento e realizam, além do gerenciamento do sistema, operações pré-programadas de controle (realimentação, decisões de segurança, etc.).

Dentre os trabalhos pesquisados no âmbito de controle de servoposicionadores a partir de microcontroladores, se destaca o trabalho de Lara-Rojo *et al.*, 2001 *apud* Kunz, 2006, onde é utilizado um microcontrolador do fabricante Microchip, modelo PIC16C711 na implementação de um controlador baseado em técnica de controle baseada em lógica *fuzzy*. No trabalho de Najjari *et al.*, 2014, é utilizado um microcontrolador do fabricante Atmel AVR modelo ATMEGA32 para execução da tarefa de posicionamento do cursor de um cilindro pneumático de dupla ação, a partir do chaveamento por PWM (*pulse with modulation*) em válvulas de controle direcional de duas vias por duas posições, sob duas diferentes estratégias de controle: PID e lógica *fuzzy*.

2.3.3 Processador Digital de Sinais (DSP)

Digital signal processor (DSP) são microprocessadores de arquitetura otimizada para o processamento, manipulação matemática de sinais digitais e gerenciamento de dados com

elevadas velocidades, oferecendo capacidades de conversão de sinais (analógico-digital/digital-analógico) com precisão. Estes componentes, assim como os microcontroladores, são compostos de uma CPU, memória e periféricos (interfaces de comunicação, conversores ADC, pinos de I/O) agrupados em um único CI. A estrutura interna é baseada na topologia Harvard modificada possuindo barramentos internos independentes para programas e dados de entrada/saída, permitindo acesso simultâneo a instruções de programa e dados, dobrando o fluxo de informações para aplicações matemáticas, possibilitando realizar múltiplas ações em um mesmo ciclo de operação [Silva, 2008].

As aplicações típicas de DSP consistem em situações que exigem intensa execução de operações numéricas de alto nível, como equações matriciais, acumulação de valores parciais resultante de múltiplos produtos, além de operações de tempo real que requerem processamento permanente de sinais de entrada e saída, como equipamentos de áudio e vídeo com redutores de ruído, equipamentos de telecomunicações, de diagnóstico médico e em sistemas de controle flexíveis e adaptativos.

Conforme Cardoso, 2006, as características dos DSP permitem a implementação de algoritmos complexos de controle (*fuzzy*, adaptativo, robusto, *sensorless*, etc.) com flexibilidade de alteração dos parâmetros de controle por *software* dispensando a interferência e parâmetros físicos dos componentes do processamento, fazendo que os custos desta tecnologia passem a ser atrativos.

Neste sentido, Cukla, 2012, implementou uma solução digital baseada em DSP para o controle de servoposicionador pneumático a partir da utilização de um processador digital de sinais modelo *dsPIC30F3013* do fabricante Microchip®; com programação lógica de execução de operações realizada em linguagem “C” e gravada diretamente em *hardware* (*firmware*).

Equipamentos comerciais para aquisição de dados de instrumentação e prototipagem de algoritmos de controle hospedada em microcomputador, como os produtos da empresa dSPACE®, fazem o uso comercial de DSP para estas operações. Vários trabalhos de projeto de controladores de servoposicionadores pneumáticos, como em Perondi, 2002, Sobczyk Sobrinho, 2009 e Suzuki, 2010, utilizaram como unidade de processamento um equipamento da dSPACE. Nesse tipo de sistema, a placa de aquisição e processamento de dados pode permanecer hospedada em um PC-IBM comum ou em módulos autônomos; a programação das rotinas de aquisição de dados e o controle da planta são realizados em interfaces gráficas

computacionais amigáveis de linguagem de alto nível (programação realizada em esquemas baseados em blocos e fluxo de sinais).

2.4 Comunicação de Dados

Em sistemas de controle baseados em dispositivos digitais, é necessária a utilização de um modelo de intercomunicação de dados entre diferentes dispositivos que compõem a instrumentação e unidade de processamento com interfaces homem-máquina ou outros equipamentos. Em arquiteturas de tempo real onde é necessário transmitir dados para sistemas supervisórios ou de armazenamento de informações (*data-loggers*) durante o ciclo de controle, é importante especificar um protocolo de comunicação rápido e confiável. Neste contexto, diversos protocolos normatizados são utilizados para comunicação de equipamentos, com diferentes características que os tornam adequados de acordo com a aplicação, dentre os quais se destacam o RS232⁵, RS485⁶, IEEE488⁷ e CAN, dentre outros. A Tabela 2.1 apresenta um quadro comparativo entre as principais características dos principais protocolos.

Tabela 2.1 – Quadro comparativo entre diferentes protocolos de comunicação

Protocolo	RS232	RS485	USB 2.0	IEEE488	CAN
Tipo	serial	serial	serial	paralelo	serial
Linha de dados	-	-	-	8	-
Taxa de dados	115,2 Kbps	100 Kbps	1,5 / 12 Mb/s	1 Mb/s	125 Kbps
Nº de Dispositivos	1	32	127	14	100
Distância Máxima de Transmissão	12 m	1200 m	5 m	20 m	40 m

Devido às características do presente projeto, o protocolo USB é utilizado para transmissão dos dados gerados pelo controlador digital a um computador compatível com IBM-PC, em função da disponibilidade de conexão USB na placa de prototipagem e computador, da facilidade de uso por sua característica *plug-and-play* e por apresentar as maiores taxas de transmissão de dados dentre as pesquisadas.

⁵ É um padrão de protocolo para troca série de dados binários entre um DTE (Data Terminal Equipment) e um DCE (Data Communication Equipment).

⁶ É um padrão que define as características elétricas de *drivers* e receptores para uso em sistemas de comunicação serial.

⁷ É um padrão para barramento de comunicações digitais de curto alcance.

O protocolo de comunicação USB (sigla para universal serial bus ou barramento serial universal) foi criado ao longo da década de 1990 por um conjunto de empresas da área de desenvolvimento de *hardware* e *software*, com o objetivo de padronizar a interconexão de periféricos em computadores pessoais baseado na comunicação de dados seriais e assíncronos (sincronismo é obtido por mecanismos característicos do protocolo), aliando funcionalidades e vantagens frente aos outros protocolos (p.e. RS-232) como *plug-and-play*, autoconfiguração de dispositivo a quente (*hot-pluggable*) além de transmissão de energia elétrica no mesmo cabo de conexão de dados.

A topologia USB é formada por até 7 camadas, nas quais podem ser interconectados até 127 diferentes dispositivos. Na camada principal (primeira) se encontra o HOST (dispositivo hospedeiro; computador, por exemplo) que efetua o controle direto de periféricos (*devices*) e HUBs. A função do HOST consiste na identificação, enumeração e coordenação do fluxo de dados entre os diferentes dispositivos conectados ao barramento. Um *device* tem por função o recebimento de dados endereçados a ele e checagem de erros e retorno da informação. Já, o HUB é um dispositivo que contém uma ou mais portas USB para outros periféricos, cujas funções básicas são a repetição do fluxo de dados, sinalização de status e gerenciamento de potência. Somente pode existir um HOST em uma arquitetura USB.

O protocolo USB se baseia em protocolo de transferência baseado em um conceito chamado de *Token Ring*; na qual três pacotes de dados são enviados a cada 1 ms. A Figura 2.6 mostra a composição dos pacotes deste modelo.

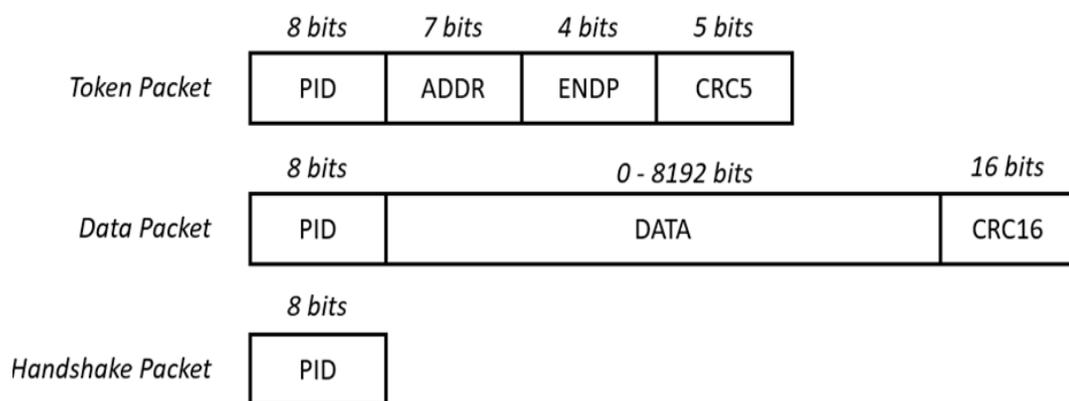


Figura 2.6 – Pacote de dados protocolo USB [Adaptado de Axelson, 2009].

A transferência de dados ocorre a partir do envio de um pacote de dados denominado *Token Packet* no qual no campo *Packet Identifier* (P.I.D.) constam informações de tipo de pacote que será transmitido, o formato e o tipo de detecção de erro; o campo *address* (ADDR)

contém o endereço do *device* ao qual está destinada a mensagem e o campo ENDP representa o *endpoint*. O campo *cyclic redundancy checks* (CRC) tem como objetivo passar informação ao destinatário de erros ocasionados durante o processo de transmissão [Axelson, 2009]. O segundo pacote enviado é o pacote de dados (*data packet*) que contém a repetição do envio do P.I.D. e CRC somado ao envio dos dados (campo DATA). O terceiro pacote consiste em somente um campo P.I.D. que contém internamente quatro pacotes: ACK, NAK, STALL e NYET, que correspondem, respectivamente, à seguinte comunicação: aceite de mensagem livre de erro; emissor não pode aceitar pacotes; solicitação de controle não suportada; e dispositivo não pode aceitar nova mensagem livre de erros [Axelson, 2009]. Durante as trocas de dados entre *host-device*, ambos assumem a condição de transmissor e receptor de dados de maneira não simultânea (comunicação *half-duplex*).

Floyd, 2007, afirma que o protocolo USB suporta duas taxas de transferência de dados, sendo uma de alta velocidade de 12 Mb/s (USB 2.0) e uma de baixa taxa de velocidade (1,5 Mb/s - USB 1.0). A versão mais moderna (USB 3.0) permite a transferência de dados em até 480 Mb/s. O protocolo USB aceita 4 tipos diferentes de transferência de dados [Axelson, 2009]:

- a) *Control*: utilizado basicamente na configuração e identificação de dispositivos;
- b) *Bulk*: transferência que possui maior compromisso com a integridade dos dados enviados, com rotinas de retransmissão de dados em caso de identificação de erro, utilizado em equipamentos com baixa taxa de transferência de dados;
- c) *Interrupt*: utilizado em condições onde há a necessidade de envio periódico de notificações ao hospedeiro sob a necessidade de confirmação de recebimento de pacotes em tempo adequado e livre de erros, como em periféricos de computadores pessoais;
- d) *Isochronous*: utilizado em situações onde a alta velocidade de transmissão de dados é um quesito fundamental; neste processo não há o envio de *handshake packet*, o que interfere diretamente na garantia de integridade das informações; muito utilizado em equipamentos de áudio e vídeo.

2.5 Técnicas de Controle Implementadas

Conforme Kunz, 2006, existem dois tipos básicos de tarefas de posicionamento para servoposicionadores pneumáticos: problema de controle de posição e problemas de controle

de trajetória. Para controle de posição, trabalha-se somente com o erro que é definido pela diferença da posição atual e a posição desejada (final). No controle de trajetória, segue-se uma curva com sucessivas posições, o que demanda maior quantidade de informações transmitidas ao longo do tempo (por exemplo, posição, velocidade e aceleração medidos e desejados).

Sendo o principal objetivo deste trabalho o desenvolvimento de uma plataforma de controle para aplicação em sistemas de servoposicionamento pneumático, para validação do desempenho do controlador digital em seguimento de trajetórias e controle de posicionamento optou-se pela implementação de duas diferentes, porém tradicionais, estratégias de controle: uma linear, denominadas PID (com ações de controle Proporcional, Integral e Derivativa) e uma não linear, chamada de controle por modos deslizantes (SMC – *Slide-Mode Control*).

Apesar de o controle PID ser considerado pouco eficiente para o controle de servoposicionadores pneumáticos em termos de precisão, em função das não linearidades inerentes à planta [Pandian *et al.*, 1997; Song, Bao e Ishida, 1997; Tanaka *et al.*, 1996], a implementação do controle PID ocorre em função da ampla utilização desta estratégia para o controle de diversas malhas de processos em ambientes industriais com resultados satisfatórios, sendo, portanto, um controlador que serve muitas vezes para obter dados de comparação de desempenho de controladores, como, por exemplo, em Perondi, 2002.

O Controle por Modos Deslizantes (SMC) é um controlador (não-linear) de estrutura variável que possui a característica de apresentar geralmente baixa sensibilidade a variações paramétricas do sistema [Perondi e Guenter, 2000]. Esta característica o torna atrativo para controle de servoposicionadores pneumáticos, sendo que na bibliografia vários trabalhos contemplam a aplicação dos controladores por estrutura variável a estes sistemas, tais como Perondi e Guenter, 2000, Gyeviki *et al.*, 2005, Li *et al.*, 2006 e Cukla, 2012.

2.5.1 Controle Linear Proporcional-Integral-Derivativo (PID)

A técnica de controle PID consiste em aplicar um sinal de correção à ser aplicado à planta de controle $u(t)$ calculado a partir do valor do erro $\varepsilon(t)$, obtido através da diferença entre o valor desejado e o valor da variável controlada, somado ao valor de sua taxa de variação em função do tempo (derivada) e sua integral com relação ao último período de tempo computado.

Este valor é transformado em um sinal adequado à planta e tem por objetivo a garantia de um controle estável e preciso.

As Equações 2.1 e 2.2 representam a estrutura do controlador, sendo K_p um valor constante que representa o ganho proporcional, K_i é o ganho integral (ou taxa integral) e K_d é o ganho derivativo (ou tempo derivativo) e definem a intensidade de cada ação. O termo $\varepsilon(t)$ representa o erro instantâneo do sistema, gerado pela diferença entre o valor da posição atual $y(t)$ e o valor de referência $y_d(t)$. O resultado da Equação 2.1 é o sinal de controle aplicado à planta $u(t)$.

$$u(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(t) dt + K_d \frac{d\varepsilon(t)}{dt} \quad (2.1)$$

$$\varepsilon(t) = y(t) - y_d(t) \quad (2.2)$$

A Figura 2.7 representa o diagrama de blocos de um controlador PID.

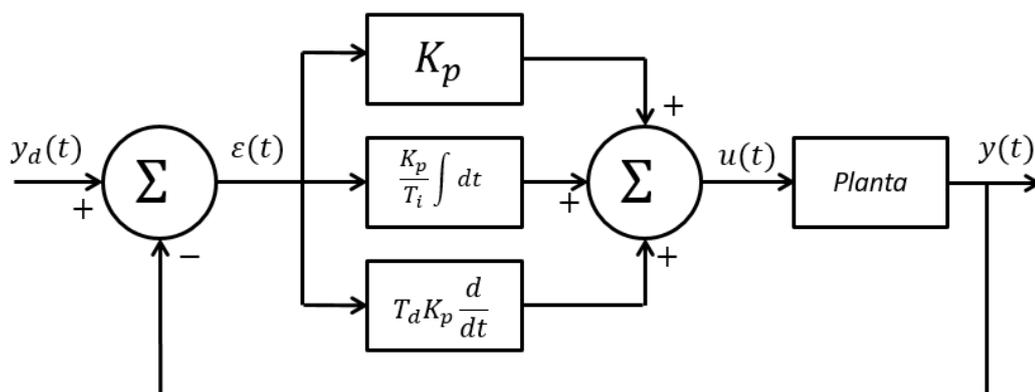


Figura 2.7 – Diagrama de blocos de um controlador PID

Na Figura 2.7, K_p é um valor constante que representa o ganho proporcional, K_p/T_i é o ganho integral e $T_d K_p$ é o ganho derivativo (ou tempo derivativo) e definem a intensidade de cada ação sobre a planta.

Aström e Hagglund, 1995, *apud* Sobczyk Sobrinho, 2009, recomendam o uso de controladores PID para sistemas cuja dinâmica seja de ordem até 2, isto é, para sistemas cuja função e transferência possui até dois polos. Conforme Perondi, 2002, o sistema de posicionamento pneumático apresenta dois polos pouco amortecidos em malha aberta e um sobre a origem. As oscilações pouco amortecidas e o risco de instabilidade, segundo o mesmo autor, limitam o ganho de um controlador proporcional, resultando em um posicionamento lento. Uma redução de erro pode ocorrer com a aplicação de uma parcela integral, mas não

produz efeitos sobre oscilações e instabilidades devido a não alteração dos polos de malha fechada. Esta característica pode ser melhorada com o acréscimo de uma ação integral, porém, também com ganhos limitados pela possibilidade de estabilidade.

Devido às dificuldades associadas à sintonia de dos três ganhos independentes dos controladores PID, diversos autores incorporam outras estratégias de controle ao PID, como PID não linear autorregulado (automático) ou SN-PID [Salim *et al.*, 2014]; ajuste adaptativo de ganhos com compensação de atrito [Azis e Bone, 1998 *apud* Perondi, 2002]; adaptação via redes neurais [Song, Bao e Ishida, 1997].

2.5.2 Controle Robusto por Modos Deslizantes (SMC)

As limitações relacionadas aos sistemas pneumáticos, como compressibilidade do ar, vazamentos e alta sensibilidade ao atrito se caracterizam em efeitos não-lineares que comprometem a eficiência dos controladores lineares. A limitação dos controladores clássicos levou os pesquisadores a buscarem soluções baseadas em controladores que fossem capazes de lidar com as não linearidades do sistema [Borges, 2017].

O controlador de estrutura variável por modos deslizantes (*slide-mode control*) possui a característica de fácil implementação e apresenta característica de robustez a variações paramétricas dos sistemas. Um sistema é dito robusto, quando ele apresenta mudanças aceitáveis no desempenho, mesmo na presença de alterações na planta e imprecisões do modelo [Dorf, 1989].

De acordo com Perondi, 2002, o controlador por modos deslizantes se baseia no chaveamento das leis de realimentação com o objetivo de conduzir a trajetória dos estados do sistema com o objetivo de conduzir a trajetória dos estados do sistema em direção a uma superfície arbitrária no espaço de estados definida em projeto (conhecida como *superfície de deslizamento* ou *superfície de chaveamento*). A superfície, uma vez atingida, deve confinar o vetor de estados de forma que ele deslize sobre ela até o equilíbrio [Locateli, 2011]. Este comportamento caracteriza o regime de deslizamento ou de modo deslizante. Na implantação de um controlador para problemas de regulação [Cukla, 2012], superfície S que o sistema deve seguir é expressa pela Equação 2.3, onde ε é o valor de erro, \dot{y} é o valor da velocidade e \ddot{y} é o valor da aceleração do sistema, enquanto S_1 , S_2 e S_3 são determinados para imposição da convergência do sistema e garantir a estabilidade.

$$S = -S_1\varepsilon + S_2\dot{y} + S_3\ddot{y} \quad (2.3)$$

Em controladores por modos deslizantes é possível definir uma região em torno da superfície de deslizamento, conhecida como *camada limite*. Os controladores que utilizam o chaveamento de leis de realimentação não garantem a existência do modo de deslizamento para todo o espaço de estados, sendo, neste caso, necessário utilizar um valor relativamente grande de largura de camada limite [Gyeviki *et al.*, 2005]. A troca contínua (chaveamento) das leis de controle para manutenção da resposta dentro do valor de largura da camada limite φ , que corresponde a um valor de erro dentro de um limite especificado em projeto, pode originar sinais de alta frequência (fenômeno conhecido como *chattering*), causando problemas como a diminuição da vida útil dos componentes móveis do sistema. A Figura 2.8 apresenta um exemplo de *chattering* de um sistema associado ao chaveamento das leis de controle no intuito de manter o sistema em modo deslizante. Os termos y e \dot{y} são a posição e a velocidade (variáveis de estado) e $S(x) = 0$ é a trajetória de estados relativa à superfície de deslizamento.

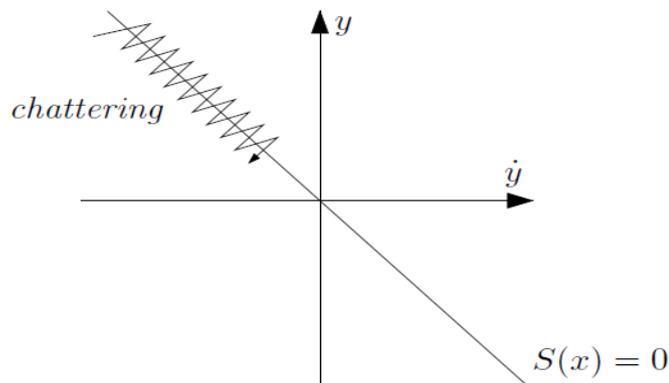


Figura 2.8 – Chattering gerado pelo chaveamento das leis de controle [Adaptação de Gyeviki *et al.*, 2005].

O *chattering* inerente a esta técnica de controle, somando ao ruído proveniente da derivação do sinal de velocidade para obtenção da aceleração, consiste em uma das maiores dificuldades para aplicação desta técnica para o controle de servoposicionadores pneumáticos [Perondi e Guenther, 2000]. Tanto o *chattering* quanto a presença de ruído eletroeletrônico nos sinais medidos, podem ocasionar excitação de dinâmicas não modeladas do sistema, o que pode comprometer a qualidade do resultado do controle.

Assim, a camada limite definida pela Equação 2.4 pode ser utilizada para diminuir o efeito de *chattering*.

$$u = -V_{max}Sat(S) \quad (2.4)$$

A função $Sat(S)$ pode ser definida como $Sgn(S/\varphi)$ quando $|S| > \varphi$ e como S/φ quando $|S| \leq \varphi$. O valor de V_{max} é o valor de tensão de saturação (tensão máxima aplicada). Na condição $|S| \leq \varphi$, ou seja, com os estados no interior da camada limite, o controlador se comporta como um controlador de estados [Perondi, 2002], com vetor de ganho $(V_{max}/\varphi)(S_1 \ S_2 \ S_3)$ e vetor de estados $(\varepsilon \ \dot{y} \ \ddot{y})$. O diagrama de blocos de um controlador tipo *slide-mode control* é apresentado na Figura 2.9.

O projeto e a implementação destes controladores em sistemas de tempo real serão apresentados na Seção 5.

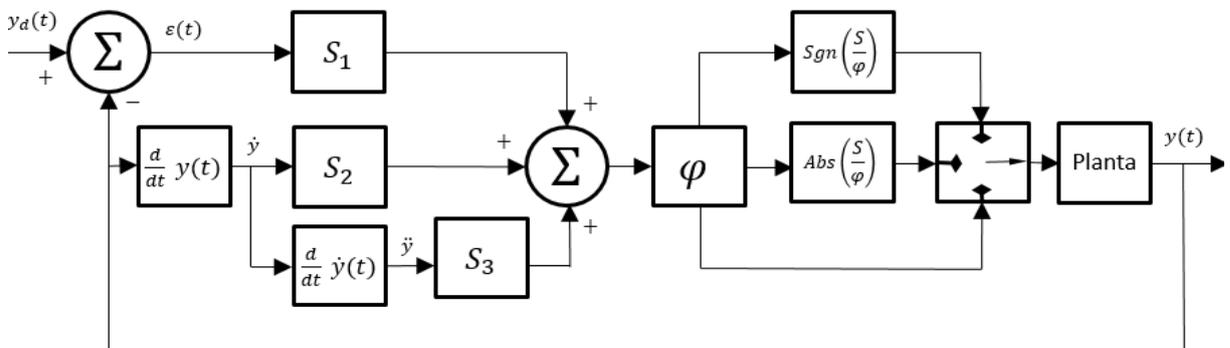


Figura 2.9 – Diagrama de blocos de um controlador por modos deslizantes [Adaptado de Perondi e Guenther, 2000].

3 SISTEMAS DE CONTROLE EM TEMPO REAL

Controladores eletrônicos digitais utilizados para o controle de sistemas físicos devem possuir capacidade de aquisição e processamento de dados em um ciclo de controle, cujo período de tempo deve estar adequado à dinâmica do processo físico controlado. Segundo Kunz, 2006, tanto o sistema de aquisição e acionamento quanto a técnica de controle necessitam de garantias temporais para eficiência de seu funcionamento. No entanto, muitas das plataformas de computação comumente utilizadas para implementar sistemas de controle não são capazes de dar quaisquer garantias determinísticas e de equidistância da amostragem [Oliveira, 2011]. Desta forma, o desenvolvimento de algoritmos em sistemas operacionais que contenham a propriedade de estabelecer um limite temporal rígido, com alto grau de confiabilidade para obtenção de uma resposta em relação ao momento de um estímulo, denominados sistemas de tempo real (RTS do inglês *real-time systems*) é útil e necessário para garantia da estabilidade e o desempenho de malhas de controle.

Um sistema operacional de tempo real (*RTOS* do termo em inglês *real-time operating system*) pode ser definido como um conjunto de *softwares* que efetuam o gerenciamento de recursos computacionais como memória *RAM*, periféricos compartilhados, gestão de tempo de processos e de tarefas, dentro dos quais o *kernel* é a camada de abstração entre *hardware* e *software*, que tem por objetivo a garantia de que todos eventos sejam executados em função das restrições temporais impostas pelos códigos programados. Estes sistemas são destinados à execução de múltiplas tarefas que interagem sincronamente com o mundo real, executam tarefas de processamento e controle as quais devem ser realizadas em um tempo suficientemente hábil ou dentro de um intervalo de tempo previamente estipulado de forma a manter o sistema operando de forma consistente. Conforme Macêdo e Farines, 2018, sistemas de tempo real podem ser informalmente definidos como aqueles que interagem sincronamente com o mundo real, percebendo nele a ocorrência de eventos e reagindo aos mesmos em intervalos de tempo previamente estipulados. De acordo com o padrão POSIX 1003.1, tempo real de sistemas operacionais consiste na habilidade do sistema em prever uma determinada quantidade de operações em um tempo pré-definido de resposta [Monfret e Linimon, 2000 *apud* Oliveira, 2005]. A norma alemã DIN 44300⁸ especifica que um sistema de tempo real é aquele onde os

⁸ Processando informação; vocabulário.

resultados das entradas estão sempre disponíveis em períodos de tempo pré-determinados [Amianti, 2008; Timmerman, 2011 *apud* Aroca, 2008]. Sistemas de tempo real são por padrão desenvolvidos a partir de uma estrutura conjunta de *hardware* e *software*, na qual o suporte de tempo real é formado por componentes de *software* que gerenciam as aplicações e os recursos de *hardware* [Mello e Sarcinelli-Filho, 2014].

O objetivo da utilização de programação em tempo real na implementação de sistemas de controle é alocar recursos computacionais limitados de tal forma que a estimação do estado e os algoritmos de controle possam garantir a estabilidade e a otimização do desempenho do sistema de controle como um sistema de tempo real [Oliveira, 2011]. Segundo Aroca, 2008, para sistemas em que alta confiabilidade é um requisito, os sistemas operacionais de tempo real devem ser utilizados por serem sistemas projetados e preparados para funcionarem de forma determinística, em contraposição aos sistemas operacionais convencionais de propósito geral.

A área de pesquisa e desenvolvimento de sistemas de controle em tempo real é interdisciplinar e engloba conhecimentos sobre computação aplicada e engenharia de controle e automação, sendo uma área importante no contexto do desenvolvimento de controladores digitais em processadores e microcontroladores eletrônicos. De acordo com Oliveira, 2011, a implementação bem-sucedida de um sistema controlado por computador requer uma boa compreensão tanto da teoria de controle quanto da teoria de sistemas de tempo real.

3.1 Classificação de Sistemas de Tempo Real

Os sistemas de tempo real podem ser classificados de acordo com as características dos sistemas aos quais se deseja controlar, como:

- a) *Soft real-time*: também conhecidos como sistemas não críticos de tempo real, quando uma reação tardia, realizada após o *deadline* (tempo máximo para realização de uma tarefa) mantém a sua utilidade – apesar dos impactos na diminuição do desempenho; atrasos apenas diminuem a qualidade de resposta do sistema, o tempo de execução é probabilístico, onde boa parte das tarefas é executada no prazo;
- b) *Hard real-time*: também conhecidos como sistemas críticos de tempo real. Neste caso, quando a reação à tarefa ultrapassa o *deadline*, pode levar à falha

- ou a condições catastróficas. Nesta classificação, não existe tolerância a atrasos, sendo que uma tarefa que não cumpra o *deadline* é considerada inútil ao sistema; desta forma, o tempo de execução deve ser garantido pelo sistema;
- c) *Firm real-time*: sistema intermediário de tempo real no qual o acúmulo de falhas ou perdas de prazo para execução de tarefas em um sistema *soft real-time* pode transformá-lo em um *hard real-time*. Nestes sistemas, as tarefas realizadas após o *deadline* são toleráveis, porém a frequência de novas ocorrências destas tarefas pode degradar o sistema como um todo, tornando-o inutilizável.

Os programas embarcados em eletrodomésticos, como micro-ondas e aparelhos de reprodução de vídeos, cujo atraso não representa condicionamento adverso do sistema físico controlado, são exemplos de aplicação de sistemas *soft real-time*. Sistemas considerados *firm real-time* são utilizados na transmissão de dados via radiofrequência e em algoritmos comunicação através de protocolos de comunicação serial, dentre outros. Equipamentos médicos, sistemas de comutação de canais de dados para telecomunicações, sistemas de gerenciamento de freios *anti-lock braking system* (ABS) e sistemas utilizados para controle de instrumentos em aviação representam sistemas onde ocorre a necessidade de utilização de sistemas tipo *hard real-time*, por conta da possibilidade de ocorrência de resultados possivelmente desastrosos, sejam eles financeiros, materiais ou de perdas humanas.

3.2 Tarefas e Políticas de Escalonamento

Em sistemas operacionais, as aplicações (programas computacionais), de tempo real ou não, podem ser abstraídas em unidades lógicas chamadas de tarefas ou processos [Mello e Sarcinelli-Filho, 2014]. Tarefas (ou *tasks*) podem ser definidas como um segmento ou trecho de código sequencial cuja execução possui um identificador, estado de execução, recurso utilizado e uma dada prioridade de acordo com a sua importância no contexto da aplicação e um atributo temporal próprio. A característica temporal das tarefas em um algoritmo implica na categorização em dois grupos:

- a) tarefas periódicas ou síncronas: ativações ocorrem em um intervalo de tempo cíclico, regular em uma sequência permanente, enquanto ocorre a execução do laço de controle;

- b) tarefas aperiódicas ou assíncronas: processamento de tarefa relacionado a interrupções ou eventos externos.

Farines, Fraga e Oliveira, 2000, afirmam que tarefas ou processos formam unidades de processamento sequencial que concorrem sobre um ou mais recursos computacionais de um sistema. São consideradas restrições de tempo para tarefas periódicas, o tempo de computação, o tempo de ativação da tarefa, o *deadline* e o *jitter* de liberação de tarefa (tempo necessário para a liberação de execução da tarefa pelo sistema operacional). O *jitter* pode ocorrer por vários motivos, dentre eles uma variação no relógio, *branches* (saltos) no código e decisões de escalonamento [Cinkelj *et al.*, 2005 *apud* Aroca, 2008].

O controle do tempo em dispositivos digitais, como computadores e microcontroladores, é realizado a partir de temporizadores (*timers*) implementados por *hardware* e muitas vezes controlados por *software*, que têm por objetivo a geração de interrupções com frequência determinada. As interrupções tipo *interrupt service routine* (ISR – ou rotina de serviço de interrupção) podem ser definidas como uma sinalização de finalização de uma operação, como por exemplo, o estouro de contagem (*overflow*) de um temporizador; o sinal de interrupção é enviado a um *controlador de interrupções* que executa uma rotina lógica para tratamento do evento. O mecanismo de interrupção de *timers* possibilita que o sistema operacional atribua fatias de tempo (*time-slices*) para cada um dos processos em um sistema multitarefas. A cada interrupção do relógio (*tick*), o gerenciador do sistema verifica se o tempo decorrido de execução de uma dada tarefa está esgotado, e em caso afirmativo, aciona o escalonador para seleção do processo subsequente para processamento. Um *tick* é a menor unidade de tempo que pode ser medida em um sistema operacional, seu tamanho otimizado depende da frequência de *clock* do sistema [Oliveira, 2011].

Para garantia de execução de todas as tarefas em restrição temporal, a camada central de gerenciamento do sistema (*kernel*) realiza a função de escalonamento (*scheduling* – classificação e ordenação) das tarefas de acordo com seu grau de prioridade indicando a ordem de ocupação do processador por um conjunto de tarefas. O escalonador é um programa que determina qual tarefa será executada a seguir, segundo algum algoritmo de escalonamento [Mello e Sarcinelli-Filho, 2014]. Moreira, 2007, afirma que o termo *escalonamento de processos* se refere à ordem de execução dos processos no estado de pronto que irão utilizar recursos do processador. Nesse processo, a rotina central é o escalonador, o qual implementa

uma escala de execução de acordo com uma política específica de escalonamento. Estes conceitos estão representados e relacionados na Figura 3.1.

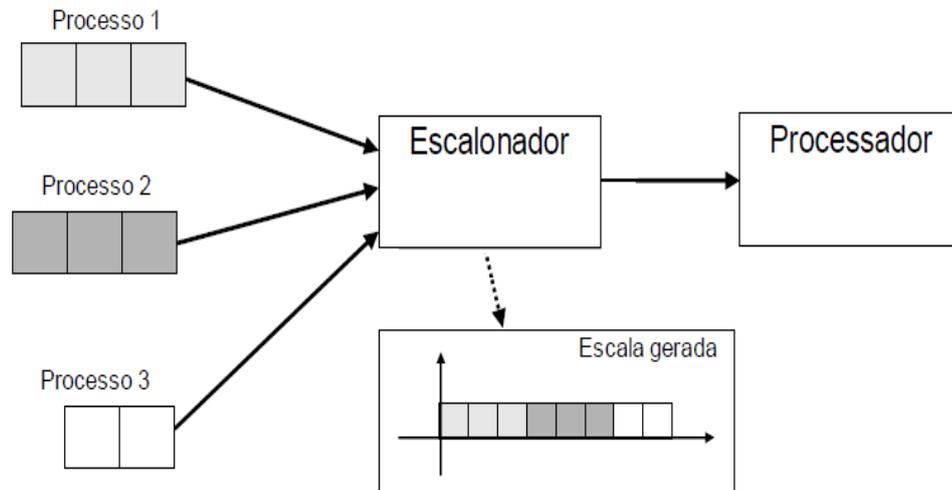


Figura 3.1– Escalonamento de tarefas [Adaptado de Moreira, 2007].

Sistemas de tempo real geralmente apresentam algoritmos de políticas de escalonamento definidas, determinando qual a ordem de execução de tarefas. As principais classificações encontradas na literatura destas políticas de escalonamento de tarefas são:

- a) Preemptivo: modo de escalonamento em que uma tarefa em execução pode sofrer interrupção antes da finalização de sua execução, para que outra tarefa de maior prioridade possa ser executada. Os algoritmos não-preemptivos não permitem a interrupção de tarefas;
- b) Estático: modo de escalonamento na qual não é permitida a interrupção de uma tarefa, sendo a ordem de execução baseada em parâmetros definidos quando da ativação da tarefa;
- c) Dinâmico: algoritmo que efetua o ajuste da prioridade e ordem de execução das tarefas de acordo com o estado ou condição do sistema;
- d) Ótimos: algoritmos de escalonamento baseados na premissa de diminuição de custo computacional para sua classe de problema.
- e) Prioridade Fixa (FP): cada tarefa periódica é associada a um período fixo, um deadline, um tempo e execução de pior caso e uma prioridade fixa.

Um exemplo de algoritmo ótimo denominado *Rate Monotonic* (RM), cujo esquema de prioridade produz escalas em tempo de execução através de escalonadores preemptivos, dirigidos a prioridades. A escalabilidade de um modelo RM define um modelo de tarefas

bastante simples com as premissas de: tarefas periódicas e independentes, *deadline* de cada tarefa igual ao período, tempo de computação conhecido e constante e tempo de chaveamento entre tarefas é considerado nulo [Oliveira, 2011]. Outra extensão muito utilizada em sistemas e tempo real comerciais é o *Round Robin* (RR), cujo conceito se assemelha ao escalonamento de prioridade fixa. Neste tipo de algoritmo, a CPU é compartilhada entre as diferentes tarefas utilizando o conceito de fatiamento de tempo (*time-slice*). Cada tarefa em um grupo de tarefas com a mesma prioridade é executada por um determinado *time-slice* antes de liberar a CPU para a próxima tarefa no grupo. Nenhuma das tarefas pode se apoderar do processador enquanto ele estiver bloqueado [Oliveira, 2011]. Outras políticas de escalonamento disponíveis em sistemas de tempo real comerciais são: *first-in first out* (FIFO), no qual uma tarefa ocupa a CPU até o seu término, bloqueio ou até que um processo de maior prioridade esteja pronto para execução; *Sporadic*: nesta política há apenas dois tipos de prioridade, alta e baixa, quando uma tarefa entra em execução é atribuída à mesma um ‘saldo’ que decresce à medida do tempo de uso da CPU. Quando o saldo é zerado, a prioridade é diminuída, recuperando a prioridade e o saldo após um determinado período de tempo, repetindo o ciclo.

Conforme Macêdo e Farines, 2018, o escalonamento é um problema fundamental em sistemas de tempo real, cujas soluções determinam a ordem de execução das tarefas do sistema. Como há prazos de execução que devem ser cumpridos pelas tarefas, ordenar a execução das mesmas têm implicações diretas na correção do sistema. O escalonador depende de um procedimento de troca de contexto para direcionar o processamento à próxima tarefa a ser executada. Segundo Santos, 2017, a troca de contexto é o procedimento pelo qual um processo A, que está em execução no processador, é pausado, e um processo B toma o seu lugar para iniciar ou continuar sua execução. Ainda de acordo com Santos, 2017, o contexto de um processo é formado pelo conjunto de dados internos ao processador: seus registradores, ponteiro de programa, ponteiro de pilha, códigos de condição de execução, além das variáveis e estados do próprio processo.

3.3 Relações de Precedência e Exclusão e Mecanismos de Sincronização e Comunicação

Políticas de escalonamento definem critérios ou regras para a ordenação de tarefas de tempo real. Os escalonadores utilizando então estas políticas produzem escalas que se forem realizáveis garantem o cumprimento das restrições temporais impostas às tarefas de tempo real [Farines, Fraga e Oliveira, 2000].

Contudo em diversas aplicações, não é possível realizar o processamento computacional das tarefas em ordem arbitrária, mas sim em ordens previamente definidas. As relações de precedência implicam na realização de ordens parciais de execução de tarefas no contexto de interdependência de sinais de sincronização ou envio/recebimento de informações produzidas em diferentes tarefas, necessárias à correta execução da rotina descrita no código, enquanto as relações de exclusão definem a forma de relação entre tarefas pela ótica do compartilhamento de recursos computacionais. As relações de precedência podem ser definidas através das necessidades de comunicação e sincronização entre as tarefas [Farines, Fraga e Oliveira, 2000].

Em sistemas de tempo real multitarefas, nas quais programas concorrentes são executados de maneira simultânea, pode haver ainda situações de concorrência (necessidade de uso de um mesmo periférico, por exemplo) entre diferentes processos programados. As principais classes de solução adotadas em SOTR para finalidade de dirimir tais problemas em programas concorrentes são:

- a) Semáforos ou Mutex: o semáforo (mecanismo de sincronização disponível em praticamente todos os RTS) é um tipo de variável abstrata que tem por objetivo a sincronização de tarefas e a restrição de acesso ou comunicação dentre as tarefas. Um semáforo binário permite controlar o acesso a um único recurso, a partir do valor de uma variável, cujo valor varia entre “1” na condição de uso (*lock*) e “0” quando na condição de livre (*unlock*). O sincronismo de tarefas, provido de semáforos, tem como objetivo manter a integridade dos recursos compartilhados, de forma a assegurar uma cooperação entre os processos [Mello e Sarcinelli-Filho, 2014]. Muitos autores também chamam os semáforos binários de *Mutual Exclusion (mutex)* para enfatizar sua característica de exclusão mútua entre duas tarefas [Laplante, 2004; Tanenbaum, 2001 *apud* Aroca, 2008]. A exclusão mútua é necessária em recursos que não permitem a preempção, como a tarefa de escrita em displays gráficos e comunicação serial, por exemplo;
- b) Troca de Mensagens (*Message Passing*): O processo de troca de mensagens se restringe a duas operações: envio e recebimento, sendo a sincronização realizada através das mesmas operações. As tarefas tipicamente recebem mensagens, executam seus processamentos e, por fim, enviam seus resultados ou sinais de sincronização na forma de mensagens [Farines, Fraga e Oliveira, 2000]. A comunicação entre processos ou *Inter-Process Communication (IPC)* é um

sistema que permite que tarefas sendo executadas troquem informações entre si [Aroca, 2008].

A intercomunicação provê mecanismos de comunicação entre tarefas, como a troca de mensagens e filas, comuns entre processos que dependem uns dos outros para a própria execução [Mello e Sarcinelli-Filho, 2014].

A escolha por um dos métodos acima leva em consideração aspectos relacionados à construção das tarefas programadas. Em programas nos quais há ocorrência de execução de tarefas em diferentes computadores ou dois diferentes núcleos em um mesmo processador, a tendência natural pela escolha de sincronismo através da troca de mensagens é geralmente observada. Conforme Aroca, 2008, qualquer programa que deseje mandar mensagem para outro pode disponibilizar esta informação na memória compartilhada e outro programa irá obter dados desta região de memória compartilhada. Ainda de acordo com o autor, como esta técnica pode causar corrompimento de dados e falha, já que não se tem controle sobre o que os programas estão escrevendo, é comum utilizar mecanismos de sincronização, como semáforos, para controlar o acesso a estas regiões de memória. A utilização de variáveis compartilhadas (semáforos binários) com permissão de acesso simultâneo entre várias tarefas gera menor custo de computação, uma vez que não há necessidade de cópia de dados de memória do remetente ao destinatário da informação, mas há alteração direta nas variáveis compartilhadas.

3.4 Sistemas Operacionais de Tempo Real (RTOS)

Para o desenvolvimento de aplicações que requerem condições temporais restritas, é necessária a avaliação das exigências qualitativas dos RTOS a fim de atendimento aos requisitos necessários. O sistema operacional deve fornecer funcionalidade mínima suficiente para que possibilite a execução completa de aplicações em tempo real de acordo com suas restrições temporais [Moreira, 2007].

A escolha de um *real time operating system* (RTOS) é importante para dar suporte a prioridades, interrupções, *timers*, comunicação entre tarefas, sincronização, gerenciamento de memória e multiprocessamento [Baskiyar e Meghanathan, 2005 *apud* Aroca, 2008]. Existem diversos projetos que provêm suporte a sistemas de tempo real com licença não-proprietária, muitos do quais estão focados em *kernel Linux* [Santos e Lima, 2009]; outras propostas de SOTR são destinadas a aplicações com sistemas embarcados com capacidades reduzidas de

processamento e quantidade reduzida de memória. Dentre os trabalhos pesquisados para implantação de controle digital para servoposicionadores pneumáticos, diversos apresentam a aplicação sistemas de tempo real.

Kunz, 2006, apresenta em seu trabalho a construção de uma bancada experimental para um sistema de servoposicionamento pneumático, com a utilização de um microcomputador IBM-PC para processamento de dados de sensores de pressão e de posição (encoder) advindos de conversores AD/DA e geração do sinal de controle para acionamento de uma servoválvula para o controle direcional. Destaca-se a utilização do sistema operacional de *kernel* Linux denominado RTAI (sigla de *real time application interface*, descrito na Seção 3.4.1) para geração de laço de controle com intervalos fixos de 1 ms. Os resultados obtidos com a aquisição a esta taxa de aquisição de dados permitiram o mapeamento das curvas estáticas tridimensionais de enchimento e exaustão, referentes a vazão e pressão nas câmaras internas do atuador pneumático em função da tensão aplicada à servoválvula e ao tempo, além da identificação do comportamento da atuação do atrito neste sistema por imposição de velocidade de deslocamento ao êmbolo do pistão.

Entre os trabalhos que relacionam a aplicação de sistemas de tempo real para controle para posicionamento e seguimento de trajetória em atuadores pneumáticos, destaca-se o trabalho apresentado por Perondi e Guenter, 2000, no qual se utilizam como aparato de controle e aquisição de dados uma placa de controle rápida dSPACE GmbH – DS1102 acoplada a um computador IBM-PC DX266MHz. Para implementação de algoritmo de controle foi utilizado o *software Matlab/Simulink* que conta com módulos e interfaces de aquisição de dados de sensores piezoelétricos para aquisição dos dados de pressão e um sensor linear para posição, em tempo real, neste caso com intervalos de tempo de amostragem fixos de 1 ms. Os ensaios realizados, utilizando um atuador de dupla ação sem haste e uma servoválvula proporcional na aplicação de controle por modos deslizantes (*slide mode control*) e em comparação a resultados de um controle tipo PVA (posição, velocidade e aceleração) apresentaram bons resultados, com baixo tempo de assentamento e erros de regime na ordem de $\pm 2,4$ mm (PVA) e $\pm 0,2$ mm (SMC) para uma massa acoplada ao atuador de 0,3 Kg. Quando da aplicação da massa de 1,4 Kg, foi verificada uma reduzida perda de desempenho (SMC) e resposta excessivamente oscilatória e perda de desempenho (PVA).

A utilização do *software Matlab / Simulink* para implementação controladores baseado em sistemas de tempo real pode ser observada em diversos outros artigos. Takosoglu, Dindorf e Laski, 2008, apresentam os resultados de simulação e de testes experimentais realizados

para o controle de um servoposicionador pneumático, onde é utilizado um algoritmo de controle baseado em lógica *fuzzy* (controle Fuzzy-PD). Este sistema, composto de um atuador linear de com 224 mm de comprimento, válvula proporcional 5/3 vias e transdutor de posição sem contato, além de placas comerciais de resolução de 16 bits (*Measurement Computing - AD/DA PCI-DAS1602/16*) como interface de computadores, utilizou o sistema *Matlab-Simulink* para prototipação do controlador em tempo real, além de recursos do mesmo sistema (*toolboxes*) para construção da modelagem matemática do sistema físico utilizado e realização da sintonia do controlador. Os resultados de testes práticos realizados com a aplicação de sinal de trajetória do tipo degrau unitário, rampa e sinal senoidal (frequência de 0.5Hz) com a aplicação de diferentes cargas mostraram-se semelhantes aos resultados previstos no modelo matemático proposto. Respostas oscilatórias durante a aplicação de cargas ao atuador, observadas nos resultados dos ensaios para o controle de posicionamento têm como causa, segundo os autores, o comportamento não modelado das forças de atrito (*Stribeck*) e efeitos como o *stick-slip* (adere-desliza), além da amplificação de ruídos provenientes do sistema de medição e aquisição de dados.

Por fim, o trabalho de Salim *et al.*, 2014, apresenta a validação experimental de uma modelagem matemática proposta para um controlador denominado SN-PID (*Self-Regulation Nonlinear PID* (PID não-linear com ganho automático)) que consiste em uma aplicação de uma função adicional ao controlador PID de forma a compensar as não linearidades do sistema. Este algoritmo foi aplicado a um controlador pneumático linear de dupla ação com curso de 500 mm. Como elementos de instrumentação, destacam-se o uso de placa de aquisição comercial (NI-PCI-6221 *card*) acopladas a um microcomputador, além de válvula proporcional 5/3 vias e sensor sem contato para aquisição do valor de posição e sensores de pressão. A construção do modelo matemático proposto e a implementação da rotina de controle sob regras de SOTR com intervalo de amostragem de 0.01 segundo foram realizadas usando o *software Matlab/Simulink*. Os ensaios experimentais permitem comparar as respostas à aplicação de um degrau unitário com passo de 400 mm para diversas cargas acopladas ao atuador (3.2 kg, 8.4 kg, 13.5 kg, 23 kg e 28 kg). Para um ensaio realizado com massa de 13.5 kg, utilizando diferentes estratégias de controle (PID convencional, N-PID (PID não linear) e controle por modos deslizantes – SMC), verifica-se uma melhor performance associada ao controlador SN-PID para cargas de até 28 kg, em relação às respostas do controle N-PID (menor velocidade) PID convencional e SMC (mais oscilatórias). Os erros encontrados no estado estacionário para os controladores SN-PID e N-PID são da

ordem de 0.01 mm enquanto para os controladores PID e SMC concentram-se na faixa de 0.1 a 0.5 mm.

Segundo Macêdo *et al.*, 2004, uma das principais atividades na implementação de sistemas de tempo real consiste no mapeamento dos requisitos temporais das aplicações e suporte de execução. Um dos meios para facilitar essa tarefa consiste na adoção de linguagens de programação e ambientes de execução com características especiais capazes de traduzir e atender a esses requisitos. Nas subseções a seguir estão resumidos os principais sistemas operacionais de tempo real pesquisados que podem ser aplicados nessa tarefa.

3.4.1 RTAI / Xenomai

O *real time application interface for Linux* (RTAI) é um sistema operacional de tempo real de código aberto e licença não-proprietária, cuja execução se realiza de forma cooperativa junto ao Sistema Operacional de Propósito Geral (SOPG) Linux nativo. Essa estratégia garante alto grau de previsão temporal de tarefas e possibilita a execução de processamento multitarefas de tempo real do tipo *hard* (restrições temporais críticas). A implementação destes sistemas na forma de *microkernel* (sistema operacional com número reduzido de recursos/serviços) é baseada em uma camada de abstração denominada *adaptive domain environment for operating systems* (ADEOS) cuja função é a de realizar a otimização do sistema para o acesso a recursos de *hardware*.

Segundo Santos e Lima, 2009, para execução de tarefas de tempo real, este sistema utiliza a abordagem de redirecionamento de interrupções, conhecida como *interrupt abstraction*, onde o *kernel* de tempo real é executado independentemente do *kernel* do SOPG. Ainda de acordo com os mesmos autores, o mecanismo de funcionamento do RTAI consiste em modificar o *kernel* do SOPG para que as tarefas executadas pelo *kernel* nativo tenham baixa prioridade em detrimento das gerenciadas pelo SOTR. Assim, as interrupções são tratadas primeiro pelo SOTR e, caso não sejam destinadas a ele, passam para o SOPG. Durante a execução das tarefas de tempo real, os processos próprios do *kernel* Linux são suspensos.

Conforme Regnier, Lima e Barreto, 2008, quando uma requisição de interrupção acontece, o *nanokernel* identifica se esta é relativa a uma tarefa de tempo real ou se é destinada a um processo Linux. No primeiro caso, o tratador de interrupção é executado

imediatamente. Caso não identifique, a tarefa é enviada para o *kernel* Linux para execução durante a ausência de tarefas de tempo real.

O *real time framework for Linux* (Xenomai) é um *framework* de tempo real que trabalha em cooperação com o *kernel* do Linux nativo visando a fornecer um amplo suporte *hard real-time* a aplicações em espaço de usuário [Pinheiro, 2009]. Implementado a partir do mesmo projeto RTAI, este SOTR se baseia na abordagem de *dual kernel*, na qual um *nanokernel* é executado ao lado do *kernel* padrão do Linux sobre o mesmo *hardware* de maneira concomitante. Ocorre aplicação de uma camada de abstração ao sistema padrão, baseada no ADEOS, que tem o objetivo de modificar as tarefas de Linux nativo para operações de baixa prioridade, além de inibir quaisquer interrupções provindas de processos Linux nativo para garantia de execução das tarefas de tempo real em modo prioritário, sem interferências. Fundamentalmente, o Xenomai segue uma filosofia de *co-kernel*, tendo um escalonador de tempo real para tarefas de tempo real e o Linux para tarefas sem regras temporais restritas [Viegas, 2015]. O *co-kernel* é responsável pelo gerenciamento das tarefas de tempo real que são executadas no ambiente Linux [Yaghmour, 2009 *apud* Mello; Sarcinelli-Filho, 2014].

De acordo com Pinheiro, 2009, os sistemas RTAI e Xenomai fornecem suporte para mecanismos de sincronização inter-tarefas, semáforos, *mutexes*, filas, *mailboxes* e serviços de tempo real a aplicações rodando em espaço de usuário, além da possibilidade de atribuição de tarefas a CPUs específicos. Ambos sistemas são portáveis a arquiteturas como *PowerPC* e *ARM*.

3.4.2 QNX Neutrino

O sistema operacional de tempo real QNX Neutrino (compatível com padrão *portable operating system interface* (POSIX) e certificado pelas normas DO-278⁹ e MIL-STD-1553¹⁰) é um *microkernel* voltado para aplicações de sistemas embarcados em plataformas *ARM* e *x86*. As políticas de escalonamento disponíveis neste SOTR se dividem em: *FIFO*, *RoundRobin*, *Sporadic*, *Adaptative Partitioning Scheduler*, o que permite adaptação do SO de

⁹ Norma regulamentadora de software para aplicação em sistemas de comunicações, navegação, vigilância e gerenciamento de tráfego aéreo.

¹⁰ Normatização Americana que define características mecânicas, elétricas e funcionais de barramento de comunicação de dados.

acordo com as características mais adequadas à aplicação, garantindo tempo de resposta adequada e altamente previsível (determinístico).

Nesta arquitetura, o *microkernel* implementa apenas quatro serviços essenciais: escalonamento, comunicação entre tarefas, comunicação de rede e detecção de interrupções. Todo restante do sistema (inserção de *drivers*, arquivos e bibliotecas) são implementados como tarefa de usuário, tornando o *kernel* rápido, confiável e relativamente pequeno [Barabanov, 1997 *apud* Aroca, 2008]. Há suporte para as características essenciais de um SOTR, como: semáforos binários (*mutex*), modelagem simplificada para tarefas de tempo real através da passagem síncrona de mensagens, além de suporte a ambientes *multi-core*, interface gráfica e integração com aplicações Java[®].

O QNX possui uma versão gratuita de código aberto para utilização em sistemas com finalidade não comercial. Para equipamentos comerciais, são disponibilizadas licença com diferentes níveis de privilégios. É amplamente difundido no meio industrial para aplicações em controle de operações críticas, como usinas nucleares, equipamentos robóticos, automotivos, instrumentos médicos, além de sistemas aeronáuticos e de defesa aeroespacial.

3.4.3 FreeRTOS

O FreeRTOS é um sistema multitarefas, de código aberto, gratuito (livre de licenças), mantido pela Real Time Engineers Ltd., desenvolvido de acordo com o padrão MISRA C (MISRA é a sigla de Motor Industry *Software* Reliability Association, que é uma associação de empresas, principalmente da área automotiva, cujo objetivo é promover boas práticas para o desenvolvimento de produtos eletrônicos automotivos) [Denardin e Barrichello, 2019].

Uma característica importante deste SOTR é o tamanho do seu *kernel* (em torno de 6 kB) o que torna possível a sua aplicação em sistemas embarcados que utilizam controladores digitais onde existe restrição operacional em função da quantidade de memória disponível e capacidade de processamento (*microkernel* baseado em linguagem C cujo código fonte possui pouco mais de 2000 linhas). Este sistema possui portabilidade para mais de 30 diferentes arquiteturas (ARM, Atmel, ESP e PIC, de 8 a 32 bits), além de suporte a funções essenciais de sistemas de tempo real, como mecanismos de sincronização baseado em semáforos binários (*mutex*), semáforos de contagem e filas de comunicação entre tarefas (*Message Queue*), suporte para processamento paralelo e aplicações de baixo consumo de energia.

Neste SOTR, o escalonamento de tarefas é selecionável pelo usuário nos modos preemptivo (*round-robin*), cooperativo ou híbrido. As tarefas são implementadas pelo usuário (em linguagem "C") no formato de pequenos programas que possuem um ponto de entrada e que é executado sem interrupção [Frigieri *et al.*, 2014]. O sistema permite que as aplicações sejam organizadas em forma de tarefas (*tasks* ou *threads*) de forma independente. A troca de mensagens entre tarefas é realizada através de um sistema baseado em fila (*queue*); as quais podem armazenar valores finitos para utilização conforme estratégia FIFO (*first-in, first-out*). As filas são objetos do sistema que não fazem parte da tarefa específica. Dados inseridos nestas estruturas podem ser acessados por qualquer tarefa, permitindo a sincronização do sistema [Frigieri *et al.*, 2014]. O FreeRTOS pode ser utilizado em aplicações onde requisitos de tempo são críticos (*hard real time*).

Existe boa disponibilidade de documentação de referência para utilização e implementação deste sistema, que incluem diversos tipos de projetos, o que facilita o processo de aprendizagem do desenvolvedor de sistemas. Neste sentido, o FreeRTOS foi utilizado no presente projeto para implementação de um algoritmo de controle com operação em tempo real, conforme descrito na Seção 5.1.

A seguir, no Capítulo 4 são apresentados os dispositivos (sensoriamento, condicionamento, atuação e computação) utilizados na estruturação da bancada para um servoposicionador pneumático, a montagem física dos componentes e as soluções implementadas no tocante ao desenvolvimento de interfaces de condicionamento de sinais dos sensores para realização dos testes experimentais.

4 IMPLANTAÇÕES EXPERIMENTAIS

Este capítulo aborda os tópicos relacionados ao aparato utilizado nos testes experimentais, englobando a descrição da montagem física do servoposicionador pneumático, as etapas relacionadas à seleção de componentes e as soluções desenvolvidas para aquisição e condicionamento de sinais.

4.1 Introdução

A Figura 4.1 apresenta uma fotografia da bancada experimental utilizada para a realização dos testes experimentais no âmbito do presente trabalho. Este sistema está instalado no Laboratório de Mecatrônica e Controle (LAMECC) no Departamento de Engenharia Mecânica da Universidade Federal do Rio Grande do Sul (UFGRS).

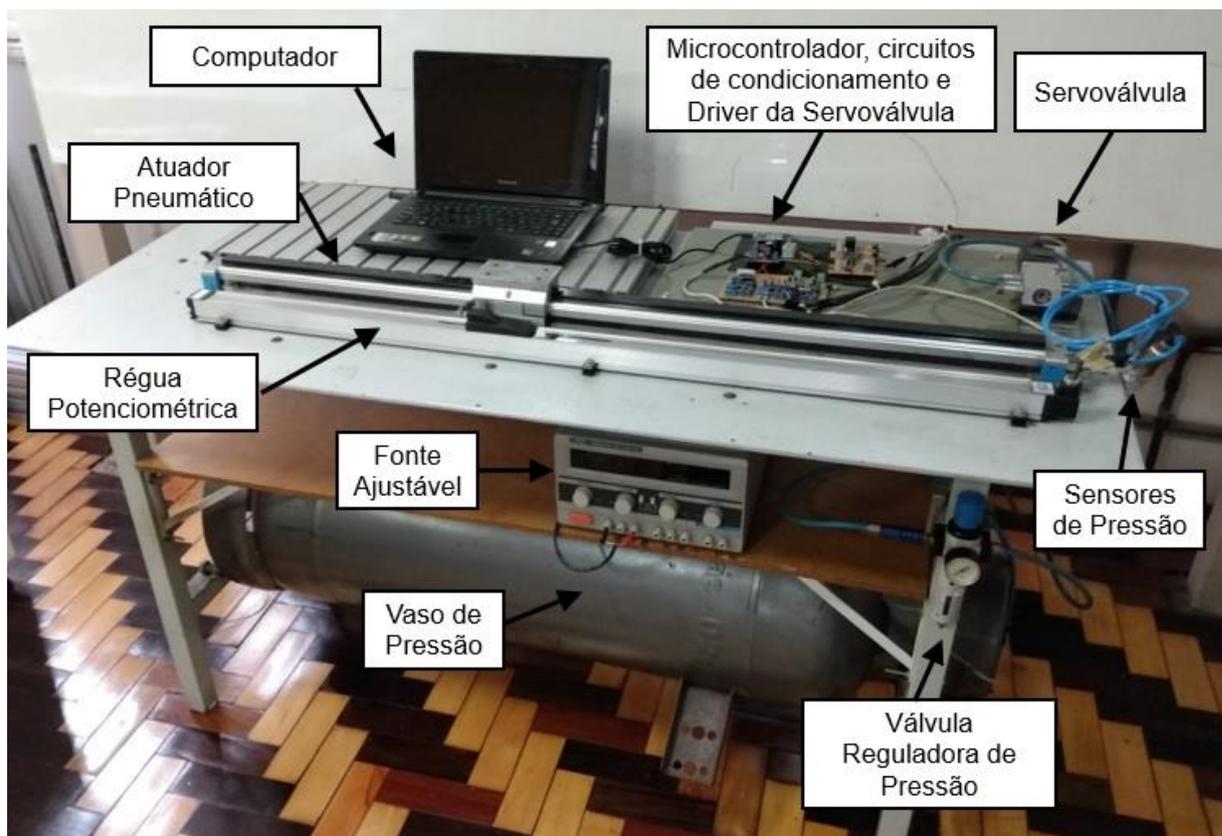


Figura 4.1 – Bancada de testes no LAMECC

4.2 Descrição Geral do Sistema

A Figura 4.2 representa o arranjo de construção do sistema, composto por uma válvula reguladora de pressão com filtro acoplado, um vaso de pressão, uma servoválvula direcional, dois sensores de pressão, um cilindro atuador de dupla ação sem haste, uma sensor resistivo linear (régua potenciométrica), um microcontrolador (MCU), dois circuitos eletrônicos: um para condicionamento de sinais dos sensores e um driver para sinal de acionamento da servoválvula direcional e um microcomputador formato IBM-PC. Este sistema se comporta de acordo com as características de operação e funcionamento descritas na Seção 1.1.

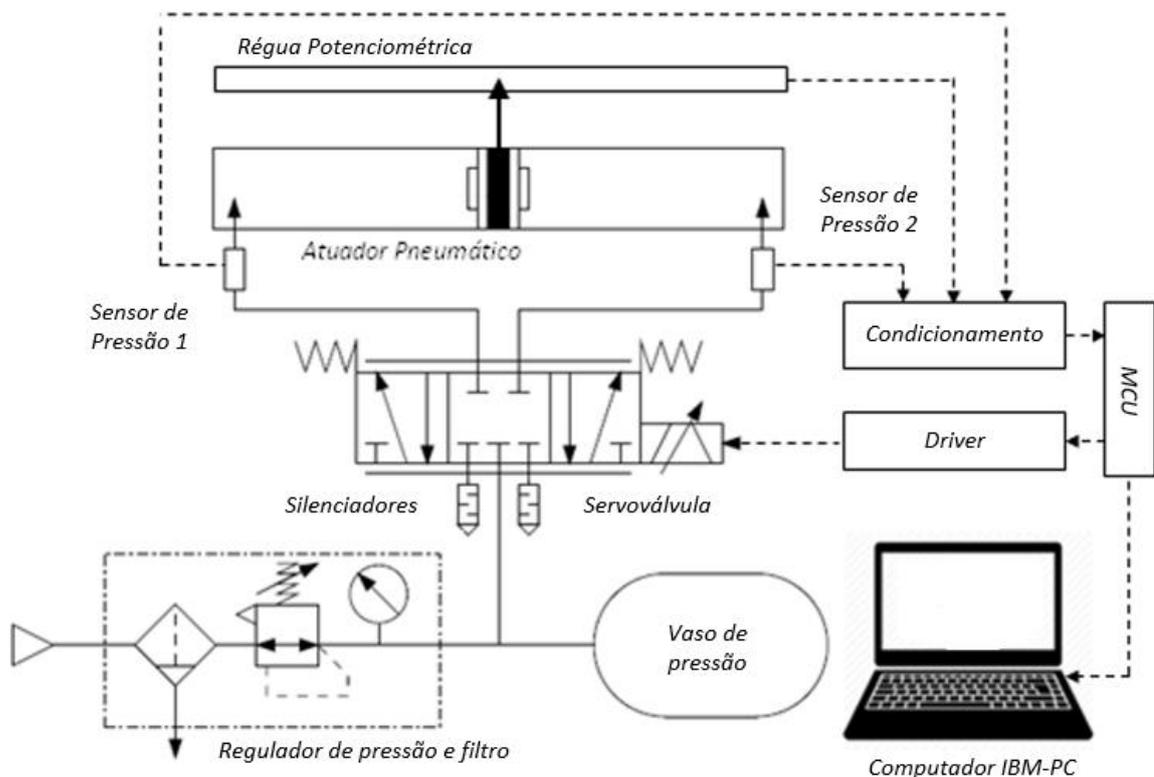


Figura 4.2 – Esquema da bancada experimental

Com o objetivo de facilitar a descrição do sistema, divide-se o conjunto em 5 partes de acordo com a sua funcionalidade: tratamento de fluido; atuador pneumático; transdução/sensoriamento de sinais; controle e driver.

O conjunto relativo à etapa de tratamento de fluido é composto por um vaso de pressão e por uma unidade de regulagem de pressão e filtro. O vaso de pressão, ligado em paralelo com a servoválvula, funciona como um filtro que reduz flutuações de pressão no sistema por utilização do ar comprimido na operação. A unidade de regulagem de pressão

permite o ajuste e manutenção da pressão de trabalho em valores aproximadamente constantes e adequados à operação segura dos atuadores, ao passo que o filtro tem por responsabilidade a retenção de particulados contaminantes presentes no ar comprimido, como poeira e água oriunda da própria atmosfera, presentes na rede de suprimento de ar comprimido.

A parte correspondente ao atuador pneumático compreende uma servoválvula direcional e um cilindro atuador. A servoválvula direcional atua no controle da força e direção do movimento do cilindro pneumático conforme o sinal de controle aplicado. Os sensores de posição e pressão das câmaras atuam no fornecimento de informações relativas ao estado da planta para o sistema de controle. A posição do êmbolo móvel do atuador pneumático é obtida a partir de uma régua potenciométrica, enquanto que as pressões internas às câmaras do atuador pneumático são adquiridas através de transdutores de pressão conectados dentre a saída da válvula proporcional e as entradas do atuador pneumático.

Para utilização das informações dos sensores pelo sistema de controle digital, a etapa de condicionamento de sinais objetiva o ajuste a valores adequados, compatíveis com o sistema de aquisição, dos sinais dos sensores em níveis de tensão e impedância, além da filtragem ativa de ruídos. Os sinais são entregues ao microcontrolador que executa as operações em tempo real, tais como de conversão analógico-digital (ADC), geração do sinal elétrico de controle (DAC/PWM) e envio de dados por meio de comunicação padrão USB para um computador formato IBM-PC. O sinal de controle em tensão elétrica gerado pelo microcontrolador é amplificado por um driver eletrônico de potência, também com o objetivo de adequar o sinal a valores adequados para atuação na servoválvula.

4.3 Componentes

São descritos a seguir os componentes que integram cada um dos quatro subsistemas acima apresentados.

4.3.1 Componentes do Sistema de Tratamento de Fluido

Para interconexão dos diferentes componentes do conjunto (vaso de pressão, regulador de pressão e filtro, atuador pneumático, transdutores de pressão e válvula direcional) foram utilizadas conexões com engate rápido [Festo, 2019], além de tubos pneumáticos com capacidade de operação em até 10 bar.

4.3.1.1 Vaso de Pressão

O vaso de pressão posicionado entre o regulador de pressão e filtro e a servoválvula tem por objetivo reduzir variações de pressão de trabalho decorrentes do consumo em operação. As principais características estão descritas na Tabela 4.1.

Tabela 4.1 – Principais características do vaso de pressão [Adaptado de Kunz, 2006].

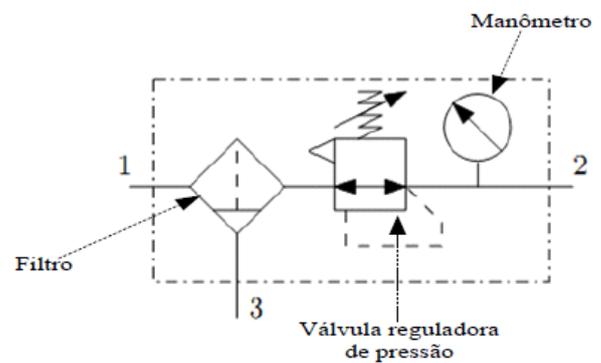
	NBR8460
Pressão de serviço	1,7 MPa
Massa	45 kg
Volume total	108 litros

4.3.1.2 Regulador de Pressão e Filtro

A unidade de tratamento de ar (Festo, LFR-1/4-D-MIDI), apresentada na Figura 4.3, realiza as funções de regulagem de pressão e filtragem do ar comprimido em um único componente.



(a) Aspecto físico do componente



(b) Representação esquemática

Figura 4.3 – Unidade de tratamento de ar e representação esquemática do componente

[Adaptado de Festo, 2019].

A Figura 4.3a mostra o aspecto físico do componente enquanto a Figura 4.3b apresenta a representação esquemática do componente, onde 1 é a entrada de ar comprimido

(pressão primária), 2 é a saída do ar para processo (pressão secundária) e 3 é a saída de água (purgador) resultante do processo de desumidificação. A regulagem da pressão é realizada através do ajuste manual de (manômetro) acoplado ao componente.

As informações referentes à unidade de tratamento de ar estão apresentadas na Tabela 4.2.

Tabela 4.2 – Principais características da unidade de tratamento de ar [Adaptado de Festo, 2019].

Grau de Filtragem	40 μm
Volume máximo do condensado	22 cm^3
Indicador de pressão	Manômetro
Faixa de Regulagem de Pressão	0,5 – 12 bar
Pressão primária (entrada)	1 – 16 bar
Histerese de pressão máxima	1400 ℓ/min
Temperatura do meio	-10 a 60°C
Temperatura ambiente	-10 a 60°C
Massa	460 g

4.3.2 Componentes do Atuador Pneumático

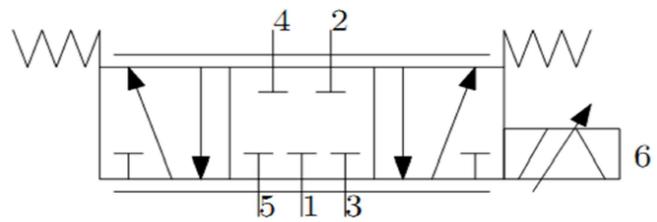
Os dois principais componentes do subgrupo do atuador pneumático são a servoválvula direcional e o cilindro atuador de dupla ação sem haste.

4.3.2.1 Servoválvula Direcional

Uma válvula de controle direcional proporcional (Festo-MPYE-5-1/8-HF-010-B), apresentada na Figura 4.4, foi utilizada para comandar o pistão, pois permite o controle do movimento do cursor do atuador pneumático em ambos os sentidos de movimento. A Figura 4.4a mostra o aspecto físico do componente enquanto que a Figura 4.4b mostra uma representação esquemática, sendo o ponto indicado pelo algarismo 1 a entrada de ar comprimido do sistema, os pontos 2 e 4 são os de utilização (saída balanceada de ar) conforme aplicação do sinal de controle em 6. Os pontos 3 e 5 representam as saídas para a atmosfera do ar exaurido durante a operação de esvaziamento das câmaras internas do atuador pneumático.



(a) Aspecto físico do componente



(b) Representação esquemática

Figura 4.4 – Válvula direcional proporcional [Adaptado de Festo, 2019].

Na Tabela 4.3 se encontram as principais características funcionais da válvula de controle direcional utilizada.

Tabela 4.3 – Principais características da servoválvula direcional [Adaptado de Festo, 2019].

Diâmetro nominal	6 mm
Tipo de acionamento	Elétrico
Princípio construtivo	Válvula corredeira do êmbolo
Tipo de reposição	Mola magnética
Tipo de Pilotagem	Direto
Sentido da vazão	Não reversível
Pressão de operação	0 – 10 bar
Vazão nominal padrão	700 ℓ/min
Frequência limite	100 Hz
Histerese máxima	0,4 %
Tensão de alimentação	17 – 30 V _{DC}
Ondulação residual	5%
Valores teórico / real	Tipo de tensão 0 – 10 V _{DC}
Meio operacional	Grau de filtragem em 5 μm
Massa	330 g

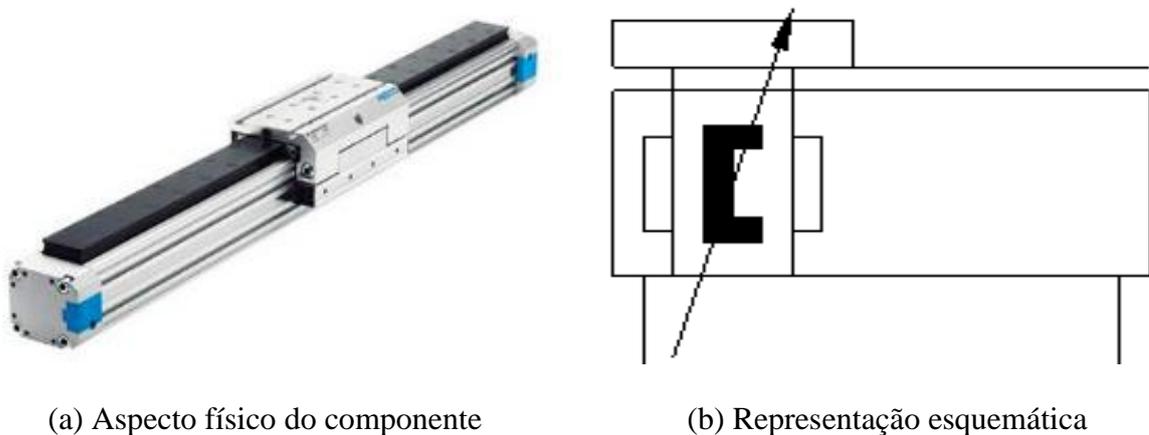
Nas conexões 3 e 5 foram utilizados silenciadores (Festo-U-1/8-B) cujas características se encontram na Tabela 4.4.

Tabela 4.4 – Principais características dos silenciadores [Adaptado de Festo, 2019].

Pressão de trabalho	0 – 10 bar
Vazão para atmosfera	1204 ℓ/min
Pressão sonora	74 dB
Massa	8 g

4.3.2.2 Cilindro Atuador

Utilizou-se o cilindro atuador de dupla ação, sem haste (Festo DGPL-25-1000-PP-V-A-KF-B), apresentado na Figura 4.5.



(a) Aspecto físico do componente

(b) Representação esquemática

Figura 4.5 – Cilindro atuador dupla ação sem haste [Adaptado de Festo, 2019].

Neste cilindro, há transmissão direta do movimento do êmbolo do atuador para o carro móvel por meio de uma guia conectora. Na Figura 4.5a é apresentada uma imagem do componente físico, enquanto que a Figura 4.5b representa a sua simbologia técnica.

As principais características construtivas e de operação referentes ao cilindro sem haste utilizado neste trabalho (Festo DGPL-25-1000-PP-V-A-KF-B) estão apresentadas na Tabela 4.5.

Tabela 4.5 – Principais características do cilindro de dupla ação [Adaptado de Festo, 2019].

Curso	1000 mm
Diâmetro do êmbolo	25 mm
Amortecimento	Ajustável em ambos limites
Pressão de trabalho	2 a 8 bar
Modo de operação	Dupla ação
Temperatura de operação	-10 a 60 °C
Curso de amortecimento	18 mm
Velocidade máxima de operação recomendada	1 m/s
Carga de flexão no eixo X (Fig. 3.3c)	430 N
Carga de flexão no eixo Y (Fig. 3.3c)	430 N
Momento longitudinal Mx (Fig. 3.3c)	5,4 Nm
Momento longitudinal My (Fig. 3.3c)	14 Nm
Momento longitudinal Mz (Fig. 3.3c)	14 Nm
Força teórica a 6 bar, avanço / retorno	295 N
Peso básico para curso de 0 mm	1,52 kg
Fator de massa adicional por 10 mm de curso	0,053 kg

4.3.3 Sensores e Transdutores

Este subsistema é composto pela régua potenciométrica linear e pelos sensores de pressão.

4.3.3.1 Régua Potenciométrica

Para medição do estado instantâneo da posição do êmbolo do atuador pneumático, utilizou-se uma régua potenciométrica (Festo MLO-POT-1000-TLF). Optou-se pela utilização deste sensor em função do comprimento de curso de medição serem os mesmos do atuador e por não necessitar de dispositivos extras, como engrenagens ou correias para acoplamento ao cursor móvel da régua ao atuador pneumático.

A Figura 4.6 apresenta a imagem comercial do componente (18^a) e a simbologia esquemática (4.6b). A régua potenciométrica é alimentada com 24 V_{DC} entre os pinos 1 e 2,

com saída proporcional à posição do cursor, variando entre 0 e 24 V_{DC} entre os pinos 3 e 2. A Tabela 4.6 apresenta as principais características deste dispositivo.

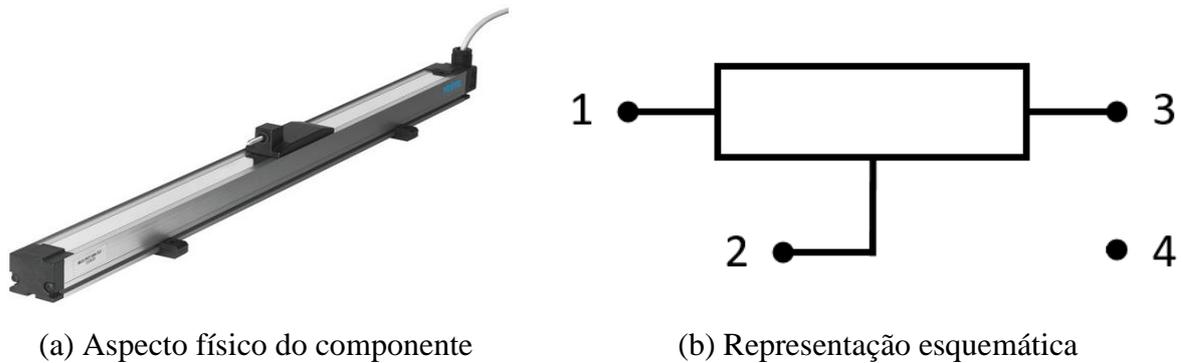


Figura 4.6 – Régua potenciométrica [Adaptado de Festo, 2019].

Tabela 4.6 – Principais características do transdutor de posição [Adaptado de Festo, 2019].

Curso máximo	1000 mm
Princípio de medição	Analógico
Resolução	0.01 mm
Peso	2,2 kg
Tensão mínima de alimentação	10 V _{DC}
Resistência ôhmica máxima	10 kΩ
Corrente máxima	4 mA
Linearidade	0,04%
Conector (tipo)	Plug 4 pinos, tipo A DIN43 650

4.3.3.2 Sensores de Pressão

Para aquisição da pressão interna de cada uma das câmaras do cilindro atuador, foram utilizados dois sensores de pressão absoluta (Wika S-10, Figura 4.7). Apesar de não serem utilizados diretamente na implantação dos algoritmos de controle utilizados no corrente projeto, os sensores de pressão fazem parte da estrutura tradicional de uma bancada de desenvolvimento de algoritmos de controle, seja, por conta da necessidade de utilização das informações de pressão medidas diretamente no algoritmos de controle (como em Perondi, 2002, por exemplo), seja para avaliação de modelos matemáticos desenvolvidos, ou mesmo

para verificar o funcionamento adequado da bancada experimental (avaliar o comportamento da pressão de suprimento, por exemplo).

Cada sensor foi acoplado entre a saída da servoválvula (conexões 2 e 4 da Figura 4.6b) e a entrada do atuador pneumático. Estes sensores foram selecionados por apresentarem range de medição de pressão dentro da faixa de valores utilizados durante a realização dos ensaios experimentais. As principais especificações técnicas deste sensor estão relacionadas na Tabela 4.7.

Tabela 4.7 – Principais características do sensor de pressão [Adaptado de Wika do Brasil, 2019].

Escala	0 – 10 bar
Sinal de Saída	0 – 10 V _{DC} ou 4 – 20 mA
Alimentação	10 – 30 V _{DC}
Tempo de resposta	≤ 1 ms
Peso	300 g
Proteção Dielétrica	500 V _{DC}
Precisão	25 mbar
Não linearidade	≤ ± 0,2%



Figura 4.7 – Sensor de pressão [Adaptado de Wika do Brasil, 2019].

4.3.4 Fonte de Tensão Elétrica

A alimentação elétrica para o funcionamento dos sensores, servoválvula, condicionadores de sinal e driver de potência foi promovida por uma fonte chaveada de

tensão ajustável (Marca Politerm – modelo POL-16E), conforme Figura 4.8. Esta fonte de tensão possui duas saídas com ajuste de tensão contínua que podem operar de forma independente, em paralelo ou em série, o que torna possível a utilização em sistemas que necessitam potências mais elevadas ou em sistemas eletrônicos que operam com tensões de alimentação distintas.



Figura 4.8 – Fonte de tensão ajustável [Adaptado de Politerm, 2019].

As principais especificações elétricas e de funcionamento da fonte de alimentação utilizadas estão descritas na Tabela 4.8.

Tabela 4.8 – Principais características da fonte [Adaptado de Politerm, 2019].

Alimentação	110 / 220 V
Operação em modo independente	0 a 32 V _{DC} – 5 A
Operação em modo paralelo	0 a 64 V _{DC} – 5 A
Operação em modo série	0 a 32 V _{DC} – 10 A
Peso aproximada	11 kg
Consumo máximo	580 W
Regulação tensão	≤ 0,01% + 3 mV (corrente < 5 A)
Ripple e ruído	≤ 1 mV _{RMS} (5 Hz a 1 MHz, ≤ 5 A)
Temperatura de operação	0 a 40 °C

4.3.5 Microcontrolador

Para a aplicação no controle da bancada pneumática, utilizou-se um microcontrolador ESP32 da fabricante Espressif, especificado por conta dos critérios e características descritas nas seções que seguem.

4.3.5.1 Seleção do Microcontrolador

Devido à grande disponibilidade no mercado de diferentes modelos de microcontroladores dotados dos mais diversos recursos e periféricos e com custo atrativo, é necessário o estabelecimento de premissas que possam adequar à correta escolha de um componente para o atendimento das necessidades previstas para a aplicação desejada. Na Tabela 4.9 são elencadas as principais características analisadas dentre três diferentes microcontroladores de diferentes fabricantes (Espressif Systems[®], Microchip[®] e Atmel[®]) para a escolha de um microcontrolador, sendo considerado o microcontrolador Node MCU ESP32 da fabricante Espressif o mais adequado à aplicação.

Tabela 4.9 – Características dos microcontroladores analisadas [Adaptado de Espressif, 2019; Microchip, 2008; Atmel, 2011].

Características analisadas	Microcontrolador		
	ESP32	PIC18F4550	ATMEGA2560
Quantidade de núcleos de processamento	2	1	1
Arquitetura	32 bits	8 bits	8 bits
Frequência de operação (<i>clock</i>)	240 MHz	48 MHz	16 MHz
Compilador + IDE gratuitos	Sim	Sim	Sim
Montado em plataforma de prototipagem	Sim	Não	Sim
Gravação <i>firmware</i> via USB	Sim	Sim	Sim
Programação em linguagem C	Sim	Sim	Sim
Suporte e documentação técnica acessível	Sim	Sim	Sim
Suporte simulador (Proteus)	Não	Sim	Sim
Suporta sistema operacional tempo real	Sim	Sim	Sim

4.3.5.2 Características do Microcontrolador Escolhido

O microcontrolador ESP32 é um dispositivo relativamente novo no mercado de microcontroladores, porém em função de seu relativamente alto desempenho associado ao baixo custo e baixo consumo de energia, pode ser aplicado em processos de manufatura industrial e em aplicações embarcadas para controle de médio e baixo nível de complexidade.

Dentre as principais propriedades que o tornam atrativo para a aplicação em controle, encontra-se:

- a) dois processadores (dual core) Xtensa 32-bit LX6, arquitetura Harvard;
- b) velocidade de até 600 DMIPS (milhões de instruções por segundo);
- c) capacidade de operação em multiprocessamento (processamento paralelo);
- d) código aberto com suporte a SOTR;
- e) montado em placa de prototipagem (Node MCU ESP32).

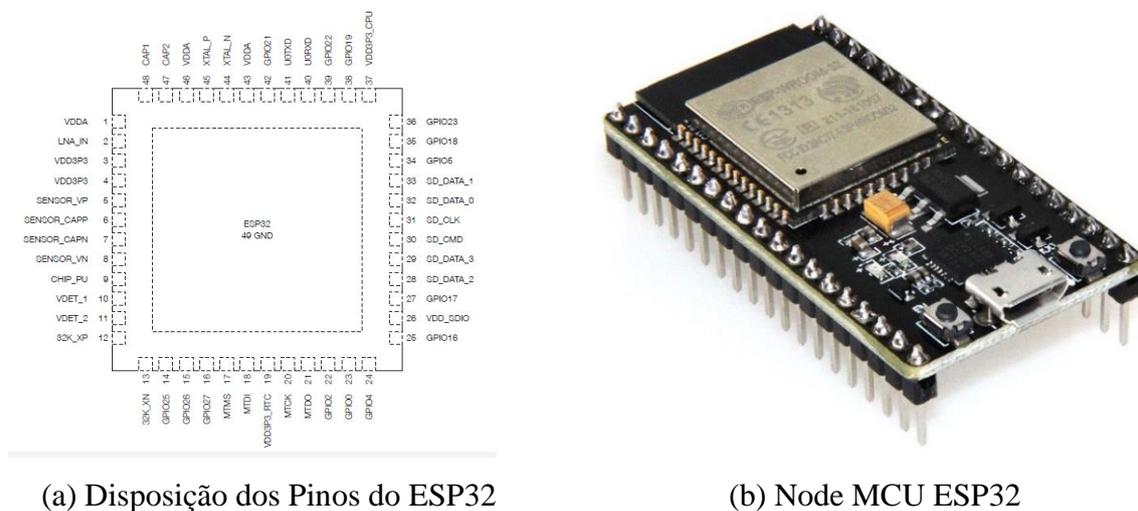
A Tabela 4.10 apresenta a relação das principais características técnicas e operacionais do dispositivo.

Tabela 4.10 – Principais características do ESP32 [Adaptado de Espressif, 2019].

Alimentação / Consumo	2,2 a 3,6 V _{DC} / 80 mA
Nº GPIO / tensão operação / corrente máxima	34 / 3,3 V _{DC} / 12 mA
Nº pinos ADC integrado / resolução	14 / 12 bits
Nº pinos DAC integrado / resolução	2 / 8 bits
Memória Flash	4 MB
Memória RAM	520 KB
Memória ROM	448 KB
Conversores ADC / Resolução	18 / 12-bits
Conversores DAC / Resolução	2 / 8-bits
Nº temporizadores / resolução máxima	4 / 64 bits
Saída PWM / resolução	16 / 12 bits
Interfaces I ² C / USART / CAN / SPI	2 / 2 / 1 / 4
Interface USB	2.0
Wi-Fi	2,4 GHz WPA/WPS
Bluetooth	BLE

A Figura 4.9 apresenta a disposição dos pinos do dispositivo juntamente com as funcionalidades previstas para cada porta, e mostra o aspecto comercial do componente, montado em uma placa de prototipação padrão, conforme utilizado no presente trabalho.

Na bancada, o ESP32 é dedicado às funções de aquisição e filtragem dos ruídos de dados dos sensores de posição e pressão das câmaras, processamento do algoritmo de controle, geração de sinal para atuação na válvula e envio dos dados resultantes do processamento a um computador IBM-PC. Conforme Perondi, 2002, em aplicação ao comando de servoposicionadores pneumáticos com algoritmos sofisticados de controle, estas tarefas devem ser realizadas dentro de um ciclo de controle de, no máximo, 1 ms.



(a) Disposição dos Pinos do ESP32

(b) Node MCU ESP32

Figura 4.9 – Microcontrolador ESP32 [Adaptado de Espressif, 2019].

4.3.6 Sistema de Aquisição de Dados

O cálculo do sinal de controle a ser aplicado à válvula direcional proporcional depende do monitoramento constante do estado da planta. As informações das variáveis físicas referentes ao valor numérico e ordem de grandeza das variáveis podem ser obtidas através dos sensores/transdutores agregados ao sistema; se convertidos em dados numéricos digitais, podem ser processados pelo algoritmo de controle implantado.

Nas seções que seguem, são apresentados os dispositivos desenvolvidos e utilizados para aquisição dos sinais, como conversores AD, interface de condicionamento e filtragem eletrônica de sinais dos sensores, conversores DA, interface de potência para amplificação em

tensão e corrente do sinal gerado pelo controlador e o processo de comunicação dos dados de processamento com um computador pessoal.

4.3.6.1 Condicionamento de Sinais para o Microcontrolador

Em razão da necessidade de adequação nos níveis de tensão e corrente dos sinais elétricos adquiridos a partir dos sensores de pressão e de posição em relação aos limites de operação previstos em componentes eletrônicos, foi necessário o desenvolvimento e implementação de uma topologia de circuito para o condicionamento de sinais (Figura 4.10).

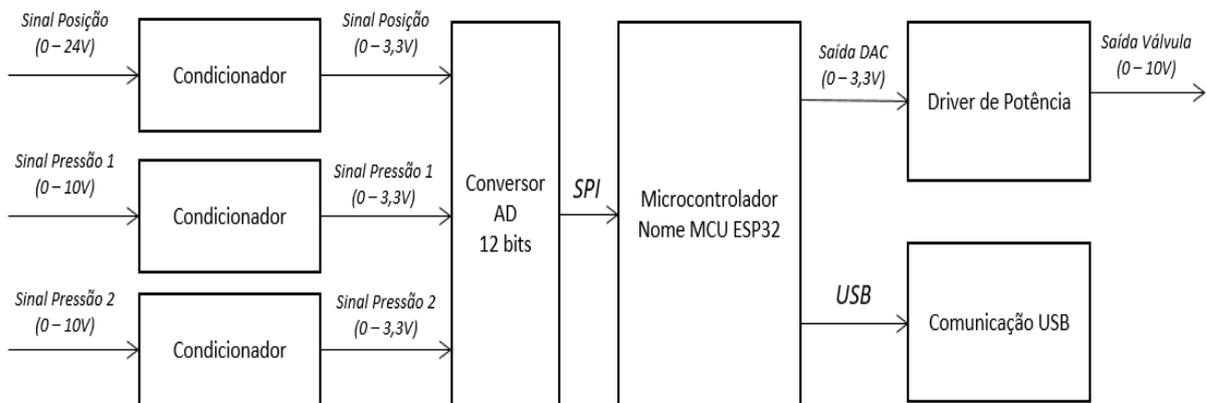


Figura 4.10 – Diagrama de blocos do fluxo de sinais

Na Figura 4.11 é representado o diagrama de blocos do circuito condicionador de sinais desenvolvido para esta aplicação.

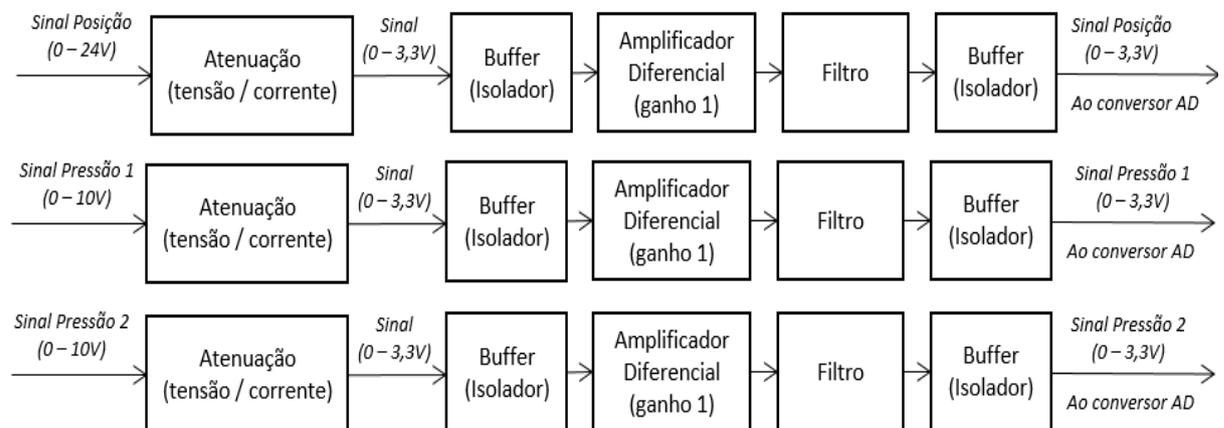


Figura 4.11 – Diagrama de blocos simplificado do condicionador de sinais

No condicionador de sinais, a etapa inicial de isolamento tem por objetivo efetuar o desacoplamento e o casamento de impedâncias entre os estágios de saída do sensor e de entrada do circuito. A aplicação de um estágio de topologia diferencial e ganho unitário somado a um filtro eletrônico analógico tipo passa-baixas objetiva a rejeição de tensões transientes ocasionadas por interferências ou ruídos elétricos provenientes do ambiente que poderiam ser prejudiciais ao bom desempenho do algoritmo de controle. As características do filtro, representadas na Tabela 4.11 foram obtidas com a utilização do *software FilterLab* versão 2.0.0.0 da fabricante Microchip Technology Inc.

Tabela 4.11 – Filtro de condicionamento de sinais implementado

Tipo	Analógico – ativo
Desenho	Buttherworth
Ordem	2
Frequência de corte	200 Hz
Resposta em Frequência	Passa-baixas
Ordem de atenuação	0,5 dB

O diagrama esquemático eletrônico completo e os devidos cálculos, estratégias utilizadas no desenvolvimento do circuito de aquisição de sinais e demais detalhes construtivos estão apresentados no Apêndice A.

4.3.6.2 Conversor Analógico-Digital

Após a correção do sinal dos sensores para níveis adequados de potência (tensão e corrente), filtragem e isolamento elétrica, para que seja possível o processamento dos dados pelo microcontrolador, é necessária a conversão do sinal analógico em um valor numérico correspondente, o que é feito por um conversor AD. A função de um conversor AD é traduzir o sinal analógico que varia continuamente no tempo em uma série de valores discretos em uma quantidade suficiente de pontos de aquisição (amostragem) de forma que possa ser definido o formato do sinal analógico.

Para a função de conversão analógico-digital, muitos microcontroladores possuem um bloco específico e especializado para a conversão analógico-digital de um sinal elétrico integrado ao seu *hardware* interno. De acordo com o teorema de *Nyquist*, a frequência de amostragem tem de ser pelo menos duas vezes a componente de maior frequência do sinal

analógico [Floyd, 2007]. Uma vez o sinal traduzido em uma série de valores discretos, os mesmos são quantizados em códigos binários que representam a amplitude do sinal analógico em cada instante amostrado. As características mais importantes de um processo de conversão analógico-digital são: o range de entrada, a resolução, o número de bits, tensão do fundo de escala, tempo de conversão, taxa de amostragem, erro de conversão, linearidade e relação sinal/ruído.

O microcontrolador ESP32 possui 14 canais destinados para a conversão analógico digital com resolução de 12 bits (Tabela 4.12) com taxa de amostragem de 100 ksps (cem mil amostras por segundo), entretanto, foi verificado em folha de dados do componente, a não linearidade no processo de conversão AD na ordem de $\pm 6\%$ na medida realizada entre diferentes CIs de mesma nomenclatura [Espressif, 2019], em toda a faixa de entrada. Em função do custo computacional para processamento de operações algébricas de correção e linearização do sinal que comprometiam a execução do algoritmo em ciclo de controle em tempo programado, utilizou-se a alternativa de um componente eletrônico externo com a mesma função para realização deste objetivo. Logo, o conversor analógico-digital MCP3208 da fabricante Microchip foi escolhido devido à suas características operacionais condizentes com a necessidade deste trabalho, apresentadas na Tabela 4.12.

Tabela 4.12 – Características técnicas do MCP3208 {Adaptado de Microchip, 2019}.

Resolução / Método de Conversão	12 bits / SAR
Nº de canais de entrada	8
Taxa de amostragem	100 ksps
Alimentação	2,7 a 5,5 V _{DC}
Fundo de escala	3,3 V
Potência sinal entrada	0 a 3,3V / 100 μ A
Interface de Comunicação	SPI
Não linearidade Integral	± 1 LSB
Não linearidade Diferencial	± 1 LSB

A Figura 4.12 mostra o aspecto comercial do componente e a referência de pinos de operação.

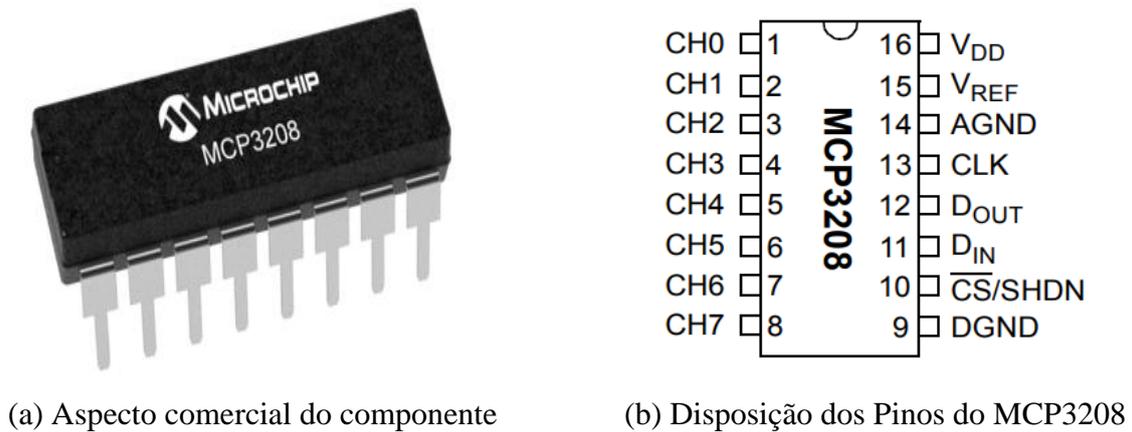


Figura 4.12 – MCP3208 [Adaptado de Microchip, 2019].

Os dados resultantes da conversão do sinal analógico pelo componente são enviados ao microcontrolador já em formato digital através do protocolo SPI (do inglês *serial peripheral interface* ou interface de dados periféricos).

4.3.6.3 Condicionamento de Sinais de Saída

Para o acionamento de válvulas direcionais por meio do sinal de controle gerado pelo ESP32, já que este opera com potência bastante reduzida há necessidade de utilização de uma interface eletrônica entre a saída do microcontrolador e a entrada da válvula proporcional, cujo objetivo é a aplicação de um ganho em potência (amplificação em tensão e corrente elétrica) ao sinal de controle provindo do sistema lógico.

No circuito de condicionamento de sinais de saída implementado (*driver*), as variáveis de interesse são os limites de potência elétrica (valores de tensão e corrente) necessários para o correto acionamento da servoválvula direcional e o formato do sinal de controle.

Neste sentido foram consideradas as seguintes premissas:

- a) a válvula proporcional pneumática utilizada neste trabalho (descrição na Seção 2.2.2) é acionada por dois níveis de tensão elétrica em dois terminais distintos; uma de $+24V_{DC}$ fixa, que alimenta o circuito interno à válvula, e outra ajustável de 0 a $+10V_{DC}$, responsável pelo controle proporcional e direcional da vazão de ar no componente;
- b) os sinais gerados pelos controladores digitais podem ser de tensão (típico de conversores digital analógico), cuja amplitude depende das características do conversor, ou sinal *pulse width modulation* (PWM – modulação por largura de

pulso), que consiste em um trem de pulsos de onda quadrada, de frequência constante e com possibilidade de modificação nos períodos de tempo de permanência do sinal em nível lógico alto (ligado) e nível lógico baixo (desligado).

O circuito condicionador de sinais desenvolvido para esta aplicação tem como diagrama de blocos a Figura 4.13.

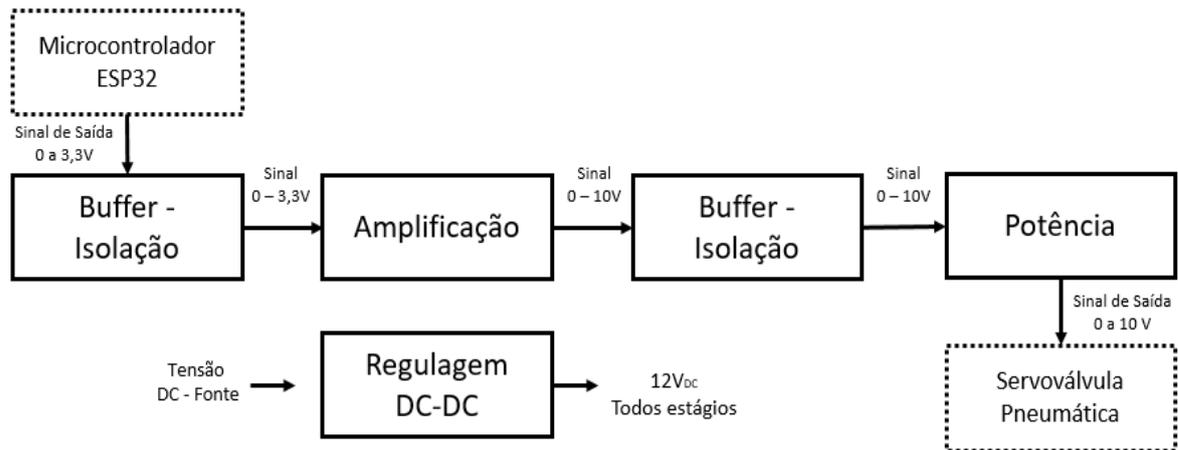


Figura 4.13 – Diagrama de blocos do driver de potência desenvolvido

Destaca-se neste circuito a implementação de estágios de isolação e casamento de impedâncias entre microcontrolador e driver, com o intuito de garantir os níveis de tensão de saída adequados para acionamento da válvula. A etapa de potência utiliza topologia de amplificadores classe AB para geração de sinal com potência adequada ao acionamento da válvula proporcional.

O diagrama eletrônico esquemático completo e os devidos cálculos, estratégias utilizadas no desenvolvimento e demais detalhes construtivos do circuito do driver de potência descrito nesta seção estão apresentados no Apêndice B.

4.3.7 Comunicação USB

A comunicação do microcontrolador com o computador foi realizada através da conexão USB, disponível em ambos equipamentos. O microcontrolador (*device*) envia ao computador (*host*), em ciclos com intervalo de 1 ms, informações relativas ao processamento de dados realizados pelo algoritmo de controle implementado. Os dados coletados na planta

enviados pelo microcontrolador (pressões, posição, referência, tempo, etc.) estão configurados para processamento de acordo com o Sistema Internacional de Unidades (SI). Os algoritmos implementados não utilizam para o controle os dados dos sensores de pressão interna às câmaras do atuador pneumático, contudo servem para o monitoramento da operação do sistema e sugere a aplicação de um trabalho futuro na qual se utilizem estas informações para um controlador robusto de estados.

4.3.8 Microcomputador

O microcomputador utilizado neste trabalho é um notebook do tipo Intel Core i3-4005U, 8Gb de memória RAM e HD de 500Gb. A CPU foi utilizada para recebimento dos dados de processamento do microcontrolador e geração de gráficos. A coleta dos dados do microcontrolador facilita a rotina da análise estatística de desempenho do controlador implementado, bem como permite a manipulação dos dados para geração de gráficos bidimensionais permitindo a visualização do comportamento das variáveis durante a operação do sistema.

No Capítulo 5, que segue, será abordada a implementação dos controladores do servoposicionador no tocante ao desenvolvimento de algoritmos de controle utilizando o sistema operacional em tempo real.

4.3.9 Custos

A Tabela A.3, no Apêndice C, apresenta os valores de mercado dos componentes utilizados para a implementação do controlador digital e suas interfaces.

5 ROTINAS COMPUTACIONAIS DO SISTEMA DE CONTROLE

Neste capítulo serão analisadas e discutidas as funcionalidades necessárias para a implantação do controle para o servoposicionador em tempo real com o objetivo de realização de testes no sistema desenvolvido, além dos algoritmos implementados para o acionamento da servoválvula e atualização das variáveis medidas da planta.

5.1 Rotinas Computacionais para o Controlador Digital

Conforme mencionado anteriormente (Seção 1.2), para controlar um sistema físico a partir de um controlador digital, o controlador deve receber medições do sistema, processá-las e enviar o sinal de controle para o atuador que afeta a ação de controle. Conforme Perondi, 2002, para garantir o uso de controladores sofisticados com segurança, em sistemas pneumáticos de servoposicionamento, o intervalo de tempo de um ciclo de controle deve ser de, no máximo 1 ms (frequência de 1 kHz).

Para garantia de execução das tarefas acima relacionadas pelo microcontrolador utilizado em um tempo conforme adequado ao sistema (1 ms), optou-se pela utilização do sistema operacional de tempo real FreeRTOS que atua como base para o desenvolvimento de um sistema de tempo real. O FreeRTOS fornece ao usuário serviços de gerenciamento de tarefas, controle de memória gerenciamento do tempo e controle dos dispositivos de entrada e saída, tornando o desenvolvimento de aplicações de tempo real mais simples e prático [Barry, 2009 *apud* Galvão, 2010].

A estrutura comum observada para o processamento computacional pelo controlador digital dentre as diferentes estratégias de controle, consiste na execução sequencial das seguintes tarefas:

- a) aquisição do sinal do sensor de posição;
- b) aquisição dos sinais dos sensores de pressão das câmaras do atuador pneumático;
- c) processamento numérico dos dados dos sensores pelo algoritmo de controle;
- d) atualização e aplicação do nível de sinal de tensão aplicado à saída, resultante do algoritmo de controle;
- e) envio digital das informações relativas aos estados instantâneos para um sistema de monitoramento de informações, externo ao microcontrolador.

No FreeRTOS, as tarefas são unidades de trabalho independentes que compõem um sistema multitarefa [Galvão, 2010]. Cada unidade de trabalho (tarefa) é construída com o mesmo formato de uma função em linguagem C, que opera em um ciclo contínuo. Considerando esta característica funcional do sistema, optou-se pela criação de duas tarefas (*tasks*): a primeira referente ao laço de controle da planta (Seção 5.1.1), e a segunda com o laço de comunicação de dados com o computador externo via porta USB (Seção 5.1.2).

5.1.1 Rotina de Controle

A tarefa denominada “tarefa de controle” engloba os processos referentes à aquisição de dados dos sensores, processamento da estratégia de controle e geração de sinal a ser aplicado à planta. Para aquisição do sinal de posição do êmbolo, o sinal de tensão elétrica deve ser primeiramente convertido em um valor numérico análogo pela interface de conversão AD (ver Seção 4.3.6). Tais valores numéricos devem ser transmitidos ao microcontrolador para processamento pelo algoritmo de controle. O circuito integrado MCP3208 possui o tempo de aquisição de dados para cada amostra sendo determinado pelo inverso da frequência de amostragem, conforme Equação 2.5:

$$T_{aquisição} = \frac{1}{f_{amostragem}} = \frac{1}{100000 \text{ (sps)}} = 1\mu s \quad (2.5)$$

Para a aquisição do dado de posição com supressão de ruídos providos do sensor potenciométrico, o sinal da posição instantânea utilizado para processamento é o resultado da média aritmética de dez aquisições sequenciais do sinal da régua potenciométrica. A utilização do valor de posição resultante do valor médio das aquisições de posição tem por objetivo estimar com uma maior precisão o valor da posição provido pelo conversor AD, além de eliminar ruídos introduzidos pelo processo de quantização binária da conversão AD.

Se o sinal de entrada mudar aleatoriamente de amostra para amostra, em quantidades (amplitudes) comparáveis à resolução da conversão (1 LSB, sigla de bit menos significativo) e o sinal de entrada tem igual probabilidade de ser qualquer valor entre dois bits de resolução adjacentes, o ruído pode ser modelado como aproximado ao ruído branco [Silicon Labs, 2019]. O ruído branco é, por definição, um sinal que possui a característica de distribuição uniforme de sua potência em todo espectro de frequência (mesma densidade espectral).

O efeito relativo à aquisição de várias amostras do mesmo sinal remete ao efeito de que a distribuição destes sinais em um histograma segue a Função Densidade de

Probabilidade e se mostra associado a uma curva de distribuição gaussiana. A Figura 5.1 representa um exemplo de distribuição de um conjunto de leituras de um conversor AD para uma aquisição de dados em um conversor de 12 bits na qual o eixo horizontal representa o valor digital e o eixo vertical representa o número de amostras do sinal.

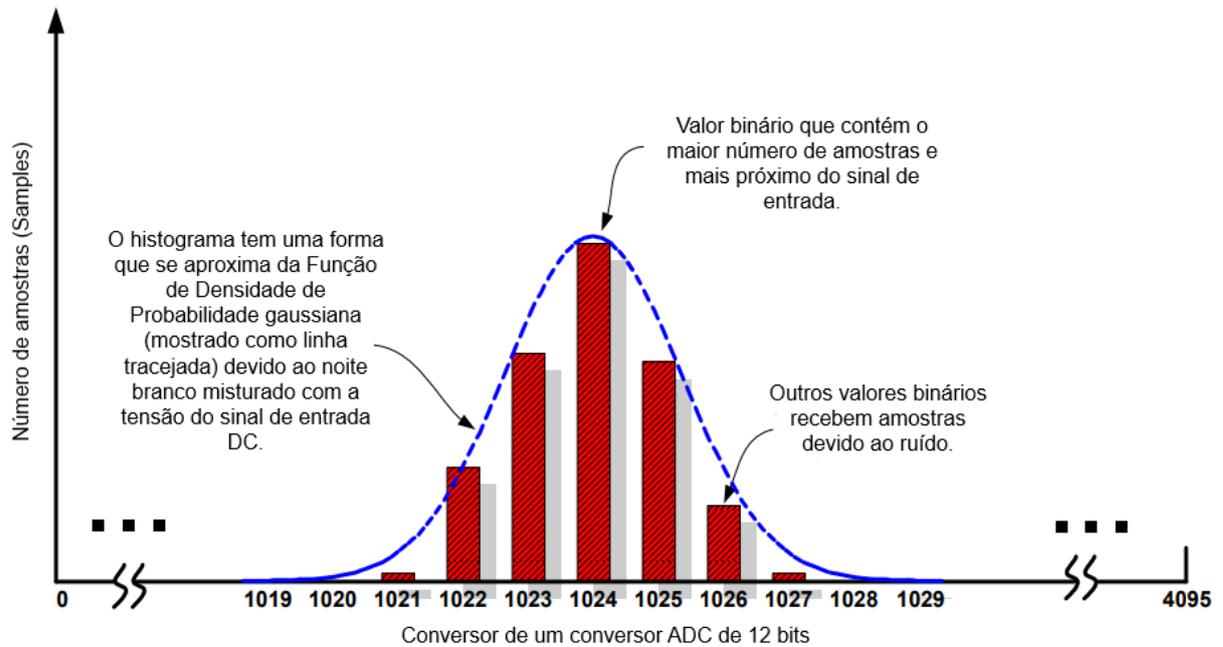


Figura 5.1 – Distribuição Gaussiana de amostragens em um conversor AD [Adaptado de Silicon Labs, 2019].

O histograma apresentado na Figura 5.1 indica que um número expressivo de leituras terá o seu valor próximo ou igual a 1024 (neste exemplo), sendo que as leituras próximas (1023 e 1025) estão contaminadas com ruído branco (± 1 LSB). No presente trabalho, em função da tensão elétrica proporcional à posição medida pelo sensor ser praticamente estática (sinal DC) no intervalo de tempo medido somada ainda a possíveis ruídos oriundos do processo de conversão, a sobreamostragem e a média dos dados de posição contribuem para a melhoria na relação de razão sinal ruído (SNR, sigla de *signal to noise ratio*) e para a resolução efetiva da conversão de dados.

O valor numérico resultante da média aritmética das aquisições dos valores de posição é submetido à sub-rotina de filtragem digital (IIR com frequência de corte de 120 Hz, Seção 5.4.1.1) e enviado, por meio do recurso de fila de mensagens (*queues*) para a rotina de envio de dados USB. Os sinais dos sensores de pressão acoplados às câmaras do atuador

pneumático foram obtidos através do processo de conversão analógico-digital do CI MCP3208 não sendo realizados processos de filtragem digital dos sinais.

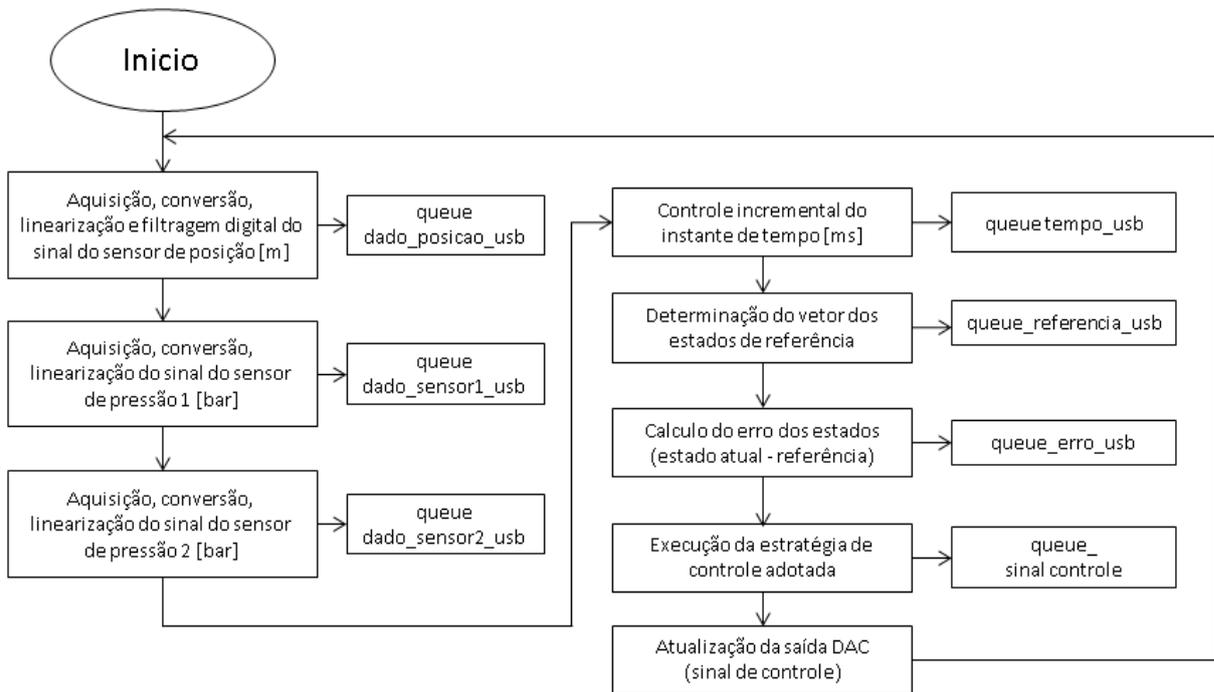


Figura 5.2 – Fluxograma das operações da tarefa de controle

Após a aquisição, filtragem e adequação dos dados de posição e pressões, são executadas as operações referentes à estratégia de controle adotada, como determinação dos incrementos de tempo, do vetor de estados de referência e do vetor de erros, além da execução do processamento do algoritmo de controle e realização da atualização dos sinais da saída do microcontrolador a ser aplicado à planta e do envio de informações para a rotina de comunicação de dados pelas portas USB a partir das filas de trocas de mensagens (*queue*). A Figura 5.2 apresenta um esquema representativo do fluxo de operações a serem executadas no âmbito desta tarefa.

5.1.2 Rotina de Envio de Dados via USB

A rotina de comunicação USB tem por objetivo receber os dados das diferentes tarefas programadas e enviar de maneira ordenada a um computador. As informações exibidas em uma interface gráfica facilitam a interpretação dos estados dinâmicos do sistema e a sua observação. A Figura 5.3 apresenta a sequência ordenada de operações para recebimento de mensagens das diferentes filas.

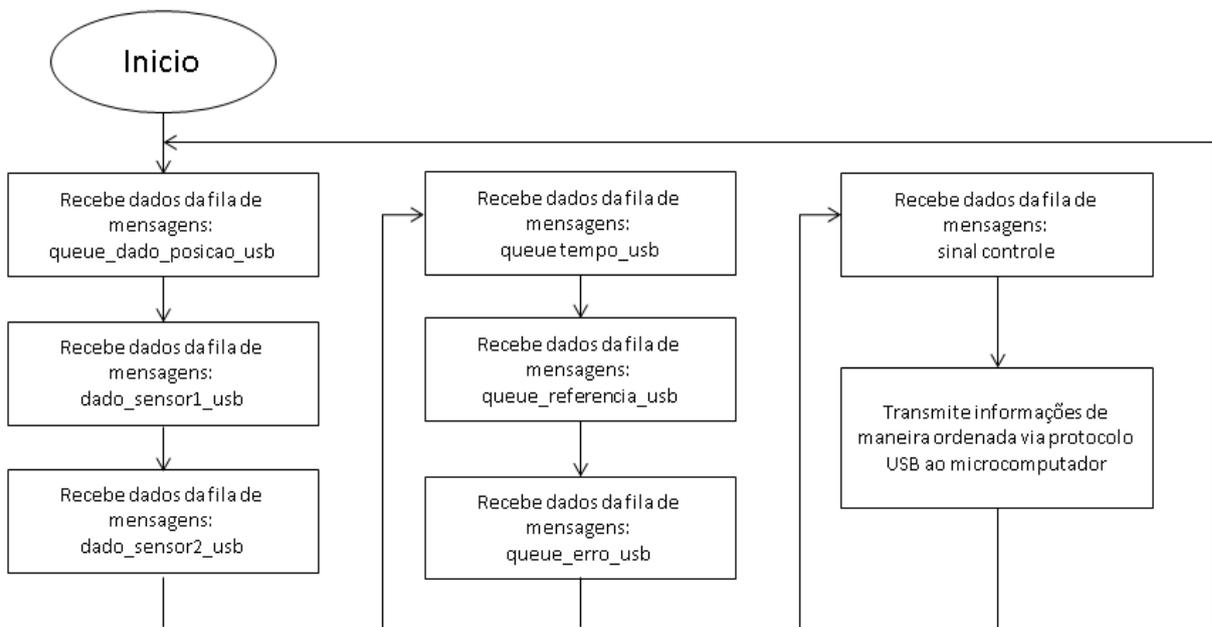


Figura 5.3 – Fluxograma referente à tarefa de envio das informações ao computador

Os valores enviados pelo microcontrolador ao computador (posição, pressões, unidade de tempo, erro, etc.) encontram-se calibrados e convertidos em unidades do Sistema Internacional (SI). Conforme já comentado, no corrente projeto, os dados referentes às pressões internas das câmaras não são utilizados nos algoritmos de controle implementados, servindo apenas para o monitoramento do sistema.

5.2 Configuração do Sistema Operacional e Organização das Rotinas Computacionais no Controlador Digital

Tendo em vista o compromisso das garantias temporais exigidas pelo SOTR, neste trabalho optou-se pela configuração padrão do sistema operacional FreeRTOS para operação conjunta com o microcontrolador ESP32, no qual é utilizado o escalonador de tarefas preemptivo baseado na prioridade tipo Round Robin, com *time-slicing* de 1 ms (que corresponde a uma frequência de operação de 1 kHz).

Com o objetivo de evitar qualquer efeito relativo a sobrecarga nos processadores ou de erros provindos de tarefas não finalizadas no prazo estabelecido, tornado assim sua utilização segura, foi considerado o direcionamento das tarefas para execução paralela em diferentes núcleos específicos (lembrando que o ESP32 possui duas CPU independentes), porém com a atribuição de mesma prioridade. A tarefa referente à ‘rotina de controle’ foi atribuída ao

núcleo “0”, enquanto a tarefa referente à rotina de ‘envio de dados via USB’ foi atribuída para execução ao núcleo “1”, ambas com prioridade de grau 2. O sincronismo entre as diferentes tarefas atribuídas aos diferentes processadores é garantido pela utilização de filas de mensagens. Conforme Mello e Sarcinelli-Filho, 2014, a intercomunicação provê mecanismos de comunicação entre tarefas, como a troca de mensagens e filas, comuns entre processos que dependem uns dos outros para a própria execução.

5.3 Ambiente de Programação

Para programação do conjunto de instruções relativo às rotinas das diferentes estratégias de controle, foi utilizada a IDE Arduino, versão 1.8.9. A opção pela utilização desta interface de desenvolvimento integrada foi baseada em quesitos como a estabilidade de operação adequada, recursos integrados de utilização de bibliotecas para acesso a periféricos e configuração de registradores, suporte para transferência e gravação do *firmware* diretamente no *hardware* do microcontrolador via porta USB, gratuidade, reduzido tempo de aprendizado, farta disponibilidade de informações em comunidade de desenvolvedores e literatura especializada.

5.4 Descrição do Ensaio Padrão

A descrição do procedimento utilizado para os ensaios experimentais para medição da robustez e do desempenho do controlador desenvolvido é um ponto importante deste trabalho. Basicamente, consiste em três operações: programação do algoritmo de controle, sintonia dos ganhos que mais se aproximam de um desempenho esperado e a análise dos dados. Os algoritmos programados estão descritos nas Seções 5.4.1 (PID), e 5.4.2 (*slide mode*), sendo as trajetórias de referência descritas na Seção 5.4.3.

5.4.1 Controle PID

O projeto deste controlador foi desenvolvido utilizando as regras do *software* FreeRTOS, operando em tempo real com laço de controle em repetições a cada 1 ms. As características de operação deste tipo de controlador foram descritas na Seção 2.5.1. Para a estimativa de relação de ganhos para o controlador PID foi utilizado o segundo método de Ziegler-Nichols [Ogata, 2010] resultando a partir da aplicação do ganho crítico $K_{cr} = 36,04$

ao sistema o período de oscilação de $0,209s$ (frequência $4,784 Hz$). Entretanto foi realizada de maneira empírica a sintonia fina do controlador, sendo os ganhos $K_p = 38,01$, $K_i = 3,22$, $K_d = 0,89$ ajustados pela medição dos melhores resultados, no controle de seguimento de trajetória, obtidos através dos ensaios realizados.

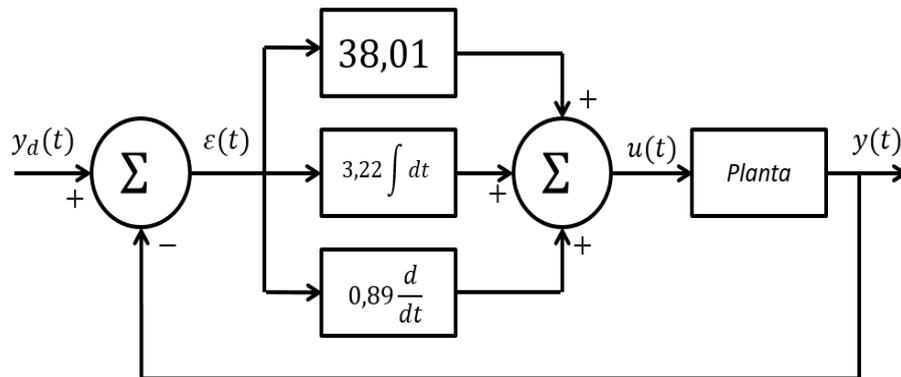


Figura 5.4 – Diagrama de blocos – controle PID

A exemplo de Cukla, 2012, a saída do controlador foi limitada aos níveis máximos de mínimos de tensão aplicados à válvula de controle direcional, definidos entre $+1$ e -1 . Estes valores foram aplicados a um bloco de multiplicação do sinal para ajuste dos valores de tensão entre 0 e 10 (V), sendo a referência em $5,1$ (V), correspondente ao ponto de centragem da válvula. A Figura 5.4 apresenta o diagrama de blocos do controlador PID.

5.4.1.1 Filtro Digital IIR - Posição

O sinal da posição do êmbolo conta, além dos filtros analógicos implantados da etapa de condicionamento de sinais, com mais um estágio de filtro digital, implantado por *software*. Para o sinal de posição, utilizou-se um filtro digital do tipo IIR implementado a partir da técnica da transformada bilinear para frequência de corte de $120 Hz$ considerando taxa de amostragem de $1kHz$. A Tabela 5.1 especifica as características do filtro implementado.

O valor estabelecido para a frequência de corte e a ordem deste filtro digital tem por objetivo diminuir o efeito de ruídos indesejados provindas do sistema de alimentação do microcontrolador, do sensor de posição e de outras fontes eventuais. Conforme Cukla, 2012, este valor leva em consideração o compromisso entre a atenuação do ruído e o atraso introduzido no sinal de entrada, o qual pode comprometer o desempenho dos controladores, principalmente quanto à capacidade de rejeição de perturbações.

Tabela 5.1 – Características do filtro digital para dados de posição

Tipo	Digital - IIR
Desenho	Butterworth
Ordem	1
Frequência de corte	120 Hz
Resposta em frequência	Passa-baixos

5.4.2 Controle SMC

Assim como no controlador PID, as mesmas técnicas de desenvolvimento de *software* foram utilizadas para implementação do controle por modos deslizantes, cujas características operacionais se encontram descritas na Seção 2.5.2. Na Figura 5.5, a superfície de deslizamento (Equação 5.1) utilizada para o problema de seguimento de uma trajetória de ordem 3, onde $\varepsilon(t) = y(t) - y_d(t)$, é o erro de posicionamento, \dot{y} é a velocidade e \ddot{y} é a aceleração do sistema.

Para a estimativa inicial de valores de ganhos para o controlador por modos deslizantes, definiu-se os valores de $S_1 \equiv \omega_n^2$ e $S_2 \equiv 2\zeta\omega_n$, onde ω_n é a frequência natural do sistema e ζ é o índice de amortecimento. De acordo com Perondi e Guenther, 1999, estes valores podem ser definidos por qualquer critério de projeto, a partir do tempo de assentamento e máximo sobrepasso aceitável. Desta forma, como critérios de projeto, a exemplo de Perondi e Guenther, 1999 e Surgenor, 1995, foram adotados os valores de 0,2s para o tempo de assentamento (critério de 2%) e 1,5% para o máximo sobrepasso, resultando em $S_1 = 866,7$ e $S_2 = 47,1$ e $S_3 = 1$.

Conforme Perondi, 2002, o projeto de um controlador por modos deslizantes pode resultar em um controlador inadequado devido ao excessivo *chattering* no sinal de controle, sendo uma alternativa modificar a lei de controle e incluir a camada limite. Desta forma o valor da camada limite, assim como os valores de ganhos (S_1 , S_2 e S_3) foram determinados através de ensaios experimentais, tendo como base os melhores resultados para um compromisso entre os menores erros de posição, a redução do efeito de *chattering* e a redução da excitação de dinâmicas não modeladas por efeitos de ruídos, provenientes do sistema de aquisição de dados e de derivação do sinal de posição; resultado nos ganhos $S_1 = 1350$, $S_2 = 10$, $S_3 = 0.5$ sendo a largura da camada limite (*boundary layer*) implementada

conforme descrito na Equação 5.2 com valor estabelecido como $\varphi = 10$. A Figura 5.5 apresenta o diagrama de blocos do controlador por modos deslizantes (SMC).

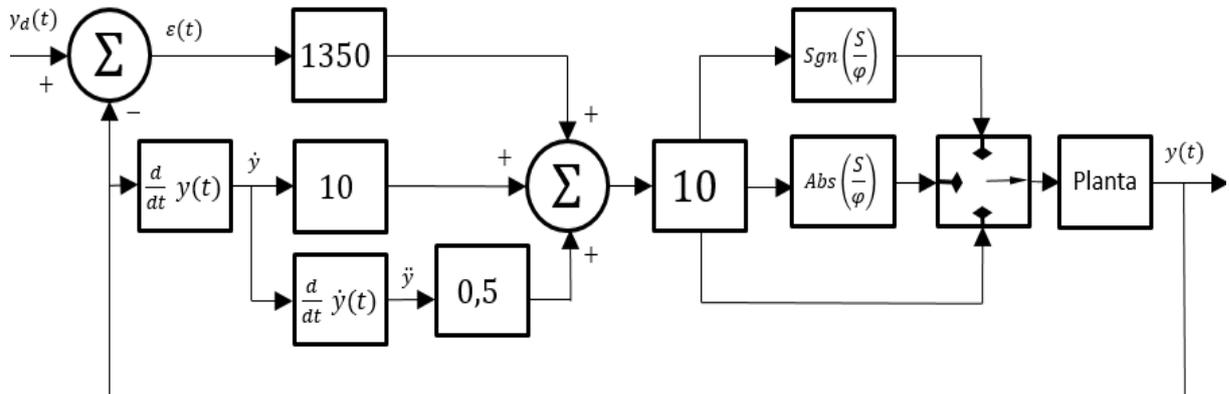


Figura 5.5 – Diagrama de blocos – controle por modos deslizantes

5.4.2.1 Filtro Digital IIR – Velocidade e Aceleração

A obtenção dos valores de velocidade e aceleração por processos indiretos de cálculo numérico (ou seja, por derivações do sinal de posição) é outra fonte de ruído que impacta diretamente na qualidade do controle. Neste sentido, a aplicação de filtros digitais IIR para os sinais obtidos tem por objetivo a redução destes ruídos, melhorando a qualidade do sinal aplicado ao algoritmo de controle de forma a suavizar a atuação do controlador sobre a válvula proporcional. A Tabela 5.2 especifica as características dos filtros implementados.

Tabela 5.2 – Características do filtro digital para dados de velocidade e aceleração

Parâmetro	Velocidade	Aceleração
Tipo	Digital - IIR	Digital - IIR
Desenho	Buttherworth	Buttherworth
Ordem	1	1
Frequência de corte	30 Hz	20 Hz
Resposta em frequência	Passa-baixas	Passa-baixas

Os valores de frequência de corte foram escolhidos através da realização de ensaios experimentais com a utilização de diferentes frequências. Para o ajuste foi considerado, a

exemplo de Cukla, 2012, o compromisso entre o nível de ruído aceitável e a instabilidade (oscilações) introduzida no sistema por atrasos relativos aos filtros na medida dos sinais.

5.4.3 Referências de Trajetória Utilizadas

Como trajetórias de referência nos experimentos realizados foram utilizadas duas curvas ao longo do tempo: uma trajetória senoidal e uma polinomial, a exemplo de Perondi, 2002 e Sobczyk Sobrinho, 2009. A trajetória senoidal é representada pela Equação 5.3:

$$y_d(t) = \bar{y} + y_{max} \text{sen}(\omega t) \quad (5.3)$$

onde \bar{y} é o valor médio da trajetória desejada, y_{max} é a amplitude e ω é a frequência de oscilação. Os valores numéricos adotados são $\bar{y} = 0,5m$, $y_{max} = 0,3m$ e $\omega = 2rad/s$, respectivamente. Na Figura 5.6 o gráfico que apresentada a representação da trajetória de referência senoidal.

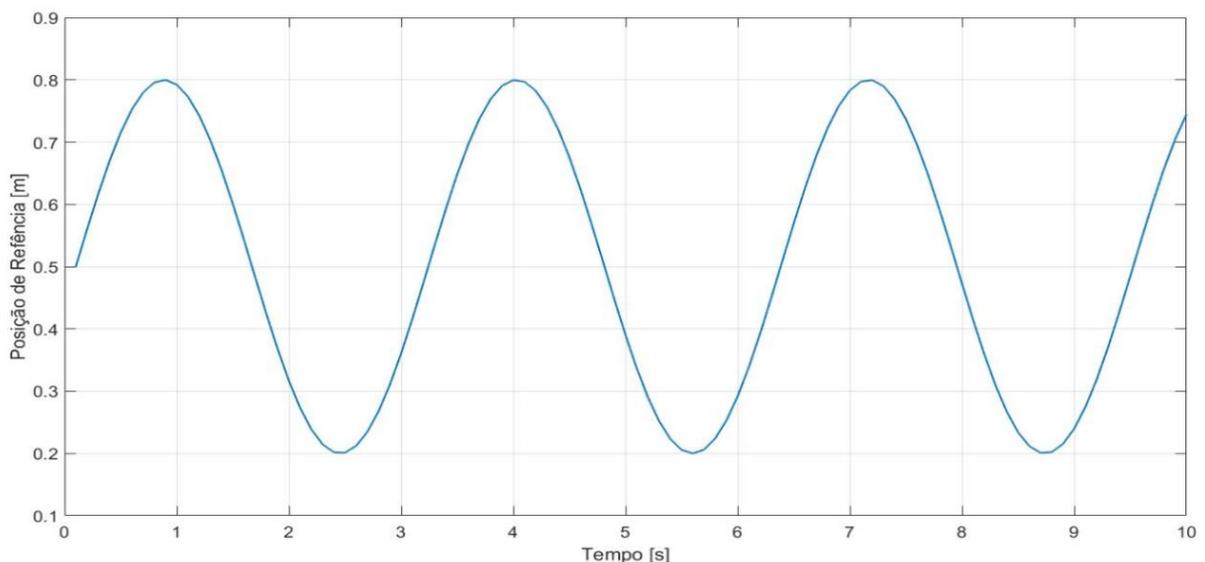


Figura 5.6 – Trajetória de referência senoidal

A trajetória polinomial foi utilizada para aproximar o comportamento de uma função do tipo salto unitária por meio de uma curva contínua [Sobczyk Sobrinho, 2009], de forma a garantir a parada nos fins de curso para que as características de regime pudessem ser analisadas [Cukla, 2011]. Nos experimentos, a trajetória polinomial é representada pela Equação 5.4:

$$y_d(t) = \begin{cases} 0,2 + y_{dp}(t/2) & t < 2 \\ 0,8 & 2 \leq t < 4 \\ 0,8 + y_{dp}((t-4)/2) & 4 \leq t < 6 \\ 0,2 & 6 \leq t < 8 \end{cases} \quad (m) \quad (5.4)$$

onde, o termo y_{dp} é dado por:

$$y_{dp}(t) = -12t^7 + 42t^6 - 50,4t^5 + 21t^4 \quad (5.5)$$

Neste caso, a posição inicial do pistão é definida em $0,2m$ a partir de uma das extremidades ($y_d(0) = 0,2m$). O pistão deve permanecer nesta posição durante um período de 2 segundos e, a seguir, em um intervalo de 0,5 segundos, ocorre um deslocamento para a posição $y_d(1) = 0,8m$ (posição a $0,2m$ da outra extremidade) de acordo com a curva polinomial descrita por meio da Equação 5.5. Novamente, o pistão deve permanecer nesta posição durante 2 segundos, quando é iniciado o movimento de retorno, também respeitando a curva polinomial anterior, repetindo o ciclo em intervalos de dois segundos entre um deslocamento e outro. Na Figura 5.7 está o gráfico que representa a trajetória polinomial de referência.

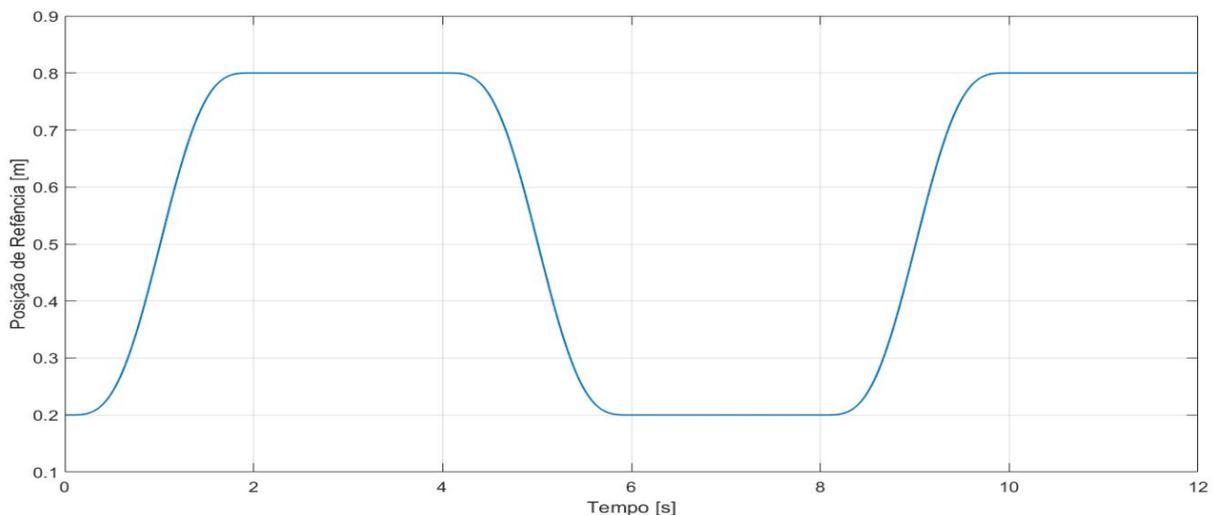


Figura 5.7 – Trajetória de referência polinomial

Os coeficientes do polinômio de sétima ordem, bem como a frequência da trajetória senoidal foram escolhidos por Perondi, 2002, de forma a se requerer a velocidade de $0,9m/s$, próxima ao limite máximo estabelecido pelo fabricante do cilindro (ver Tabela 4.5).

5.4.4 Compensação da Zona Morta da Servoválvula Proporcional

A zona morta é caracterizada como uma situação onde uma faixa de valores de sinal de entrada não produz qualquer efeito na saída [Salim *et al.*, 2014]. Para redução de erros no seguimento de trajetória e de posicionamento, ocasionados por efeitos de assimetria no controle de vazão da servoválvula proporcional, a zona morta foi compensada para o controlador PID. A compensação foi realizada através da soma de valores fixos de compensação nos sinais de controle de acordo com a direção de abertura da válvula [Perondi e Guenter, 2000], considerando os diferentes regimes (estacionário e transiente) das trajetórias de referência.

Para o regime transiente das trajetórias senoidal e polinomial, o sinal de controle efetivo (u_{efet}) é descrito pelas Equações 5.6 e 5.7:

$$u_{efet} = k + u(t) + z_{m1} \rightarrow u > 0 \quad (5.6)$$

$$u_{efet} = k + u(t) - z_{m2} \rightarrow u < 0 \quad (5.7)$$

onde k é o valor de tensão de centro da válvula, $u(t)$ é o sinal de controle originado pelo algoritmo, z_{m1} e z_{m2} são os valores de tensão estabelecidos como zona morta. Neste trabalho os valores utilizados para os parâmetros são: $k = 5,1V$, $z_{m1} = 0,78V$ e $z_{m2} = 0,95V$. Estes valores foram obtidos experimentalmente por meio dos ensaios realizados para a sintonia de malha no controlador PID, sendo que a aplicação destes valores à servoválvula proporcional permite a manutenção de uma velocidade mínima de movimento do êmbolo, a fim de minimizar respostas oscilatórias ocasionadas por efeito *stick-slip*, de atrito e por ruídos do sistema de instrumentação.

Para o regime estacionário, onde há a permanência do pistão em uma posição fixa durante um intervalo de tempo (trajetória polinomial, Seção 5.4.3), o sinal de controle efetivo (u_{efet}) é descrito pelas Equações 5.8, 5.9, 5.10 e 5.11.

$$u_{efet} = 5,35V \rightarrow \text{se } 5,0V \geq u(t) \leq 5,35V \text{ e } y(t) = 0.20 \quad (5.8)$$

$$u_{efet} = 4,65V \rightarrow \text{se } 5,0V \leq u(t) \leq 4,65V \text{ e } y(t) = 0.20 \quad (5.9)$$

$$u_{efet} = 5,35V \rightarrow \text{se } 5,0V \geq u(t) \leq 5,35V \text{ e } y(t) = 0.80 \quad (5.10)$$

$$u_{efet} = 4,65V \rightarrow \text{se } 5,0V \leq u(t) \leq 4,65V \text{ e } y(t) = 0.80 \quad (5.11)$$

onde, $u(t)$ é o sinal de controle originado pelo algoritmo e $y(t)$ é o valor da posição do êmbolo. Da mesma forma que os valores de zona morta, estes valores foram obtidos de maneira experimental e representam a tensão mínima aplicada a servoválvula para iminência do movimento do êmbolo, sendo que proporcionam resposta de regime com menor oscilação nos períodos de baixa velocidade do êmbolo durante a troca do regime estacionário para o transiente.

O capítulo que segue apresenta os resultados obtidos na medição do tempo de execução de cada *task* programada no *looping* de controle além dos resultados obtidos no seguimento de trajetória para as duas estratégias de controle nas referências adotadas.

6 RESULTADOS

Neste capítulo são apresentados os resultados obtidos através dos ensaios realizados na bancada experimental. Primeiramente, são apresentados os resultados referentes à medição dos tempos de execução dos laços de controle.

Posteriormente, são apresentados os resultados experimentais em malha fechada, sendo inicialmente considerados os resultados obtidos com a aplicação do controle PID, seguidos do controle SMC, para as trajetórias senoidal e polinomial. Por fim são apresentadas as análises comparativas dos resultados obtidos em termos de amplitudes de erro com relação a trabalhos similares que se valem das mesmas estratégias de controle, porém, utilizando diferentes *hardwares* de controle.

6.1 Tempo de Execução do Laço de Controle

Os resultados referentes aos tempos de execução das tarefas programadas (controle e comunicação de dados), medidos com um analisador lógico modelo AL24 da marca Seleae, que opera com frequência de amostragem de 24 MHz e possui precisão de ± 24 ns, permitiram a verificação da manutenção e estabilidade das repetições dos ciclos de controle em intervalos regulares de 1 ms, durante o processo de execução de ambos algoritmos de controle programados, evidenciando a efetividade da atuação do sistema operacional de tempo real no controle e manutenção dos tempos programados de execução dos algoritmos.

A Tabela 6.1 apresenta os valores estatísticos referentes aos tempos de *looping* de controle medidos para as estratégias PID e SMC.

Tabela 6.1 – Tempos de execução dos *looping* de controle em milissegundos

Tempo (ms)	PID	SMC
Mínimo	1,000125	1,000167
Máximo	1,000417	1,000417
Média ciclo controle	1,000276	1,000276
Desvio padrão	$\pm 8.9 \times 10^{-5}$	$\pm 2.1 \times 10^{-5}$

6.1.1 Controlador PID

A Figura 6.1 apresenta o resultado da medição do tempo de execução em sequência de dois ciclos de controle pelo controlador, além de indicar as taxas de ocupação de cada núcleo do processador para execução de cada uma das tarefas programadas, sendo para a amostra, a ocupação de 29,35% do ciclo de 1 ms para a tarefa de controle, enquanto que a tarefa de envio de dados utiliza 21,85%.

É possível também identificar o sincronismo entre as duas diferentes tarefas, relacionado à finalização de envio de informações por filas de mensagens do laço de processamento da tarefa de controle em relação à tarefa de comunicação USB.



Figura 6.1 – Tempos de ciclos de controle e taxas de ocupação de processadores para o controlador PID

A Tabela 6.2 apresenta os valores estatísticos dos tempos de execução das tarefas programadas.

Tabela 6.2 – Tempos de execução das tarefas do algoritmo PID em milissegundos

Tempo (ms)	task_controle	task_usb
Mínimo	0,285624	0,217542
Máximo	0,293669	0,24450
Média Ciclo Controle	0,289464	0,228325
Desvio Padrão	$\pm 3,7234 \times 10^{-3}$	$\pm 9,0356 \times 10^{-3}$

6.1.2 Controlador por Modos Deslizantes (SMC)

A Figura 6.2 apresenta os resultados da medição dos ciclos de controle e taxas de ocupação de processadores para o controlador por Modos Deslizantes. Para este controlador, os valores encontrados na amostra são de 30,07% para o processamento do algoritmo de controle e 25,18% para o envio de dados via USB. Da mesma forma como identificado para o controle PID, é verificado o sincronismo entre as diferentes tarefas.

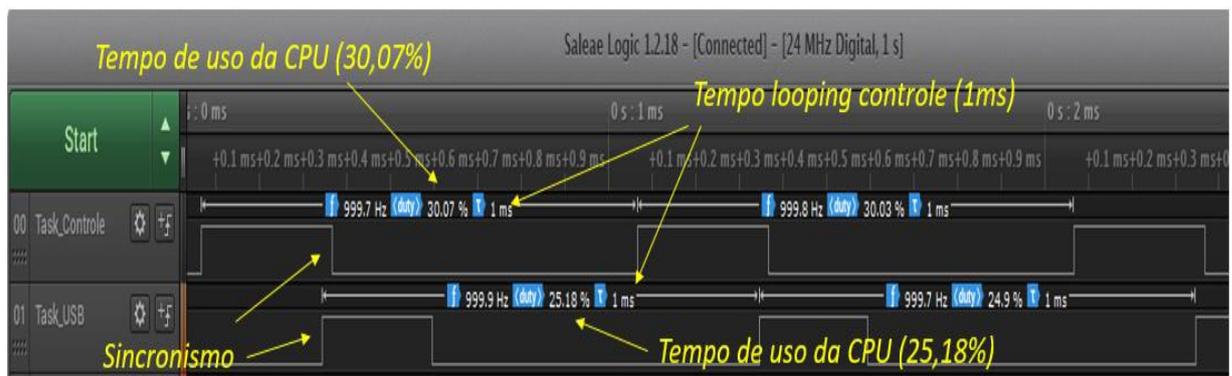


Figura 6.2 – Tempos de ciclos de controle e taxas de ocupação de processadores para o controlador por Modos Deslizantes

A Tabela 6.3 apresenta a os valores estatísticos dos tempos de execução das tarefas programadas.

Tabela 6.3 – Tempos de execução das tarefas do algoritmo SMC em milissegundos

Tempo (ms)	task_controle	task_usb
Mínimo	0,302542	0,240125
Máximo	0,300291	0,296208
Média Ciclo Controle	0,301353	0,270647
Desvio Padrão	$\pm 8,32 \times 10^{-4}$	$\pm 2,08 \times 10^{-2}$

6.2 Resultados Obtidos no Seguimento de Trajetórias

Os resultados apresentados nas próximas seções correspondem aos ensaios realizados com a plataforma de controle desenvolvida neste trabalho. Para esta tarefa, as estratégias de controle PID e SMC descritas nas seções 4.4.1 e 4.4.2 foram testadas utilizando as trajetórias senoidal e polinomial de 7ª ordem descritas na Seção 5.4.3.

6.2.1 Controle PID

Para realização dos ensaios de controle por meio da técnica PID, os ganhos utilizados são: $K_p = 38,01$, $K_i = 3,22$, $K_d = 0,89$. Estes valores foram sintonizados experimentalmente a partir dos resultados de diversos ensaios e correspondem aos menores erros de seguimento de trajetória para as diferentes referências utilizadas. A pressão de trabalho ajustada na válvula de controle de pressão e utilizada para todos os ensaios foi de 0,6 MPa (6 bar).

6.2.1.1 Trajetória Senoidal - PID

A Figura 6.3 apresenta os resultados típicos obtidos nos experimentos, onde pode-se observar a posição e a trajetória de referência em função do tempo.

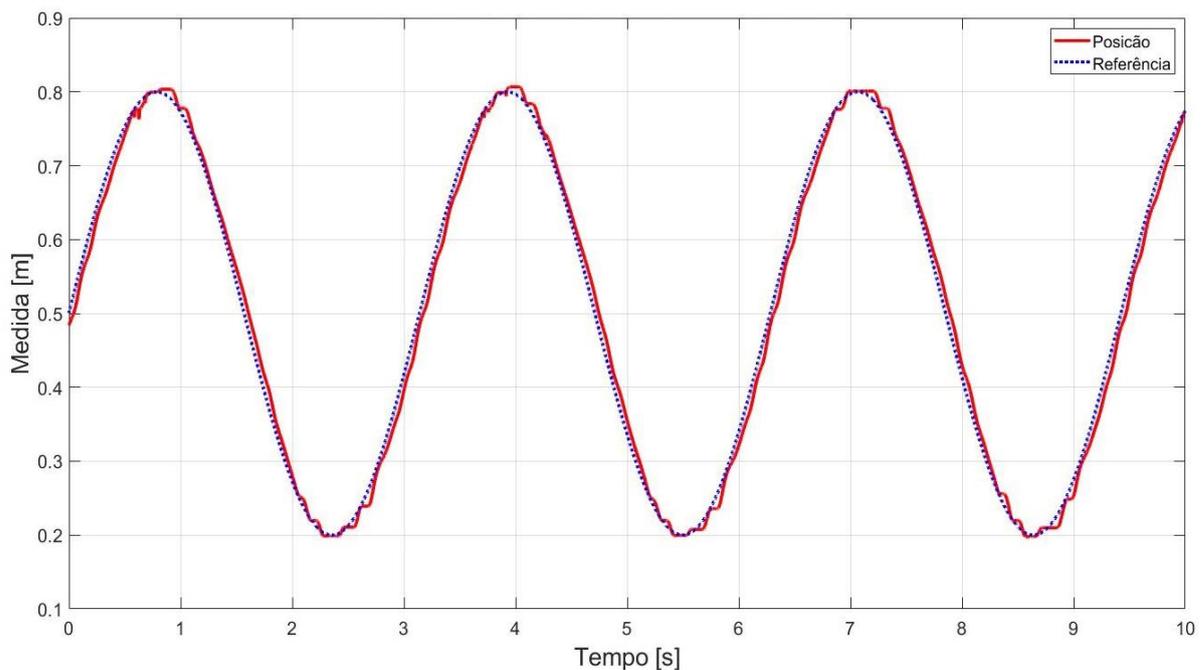


Figura 6.3 – Seguimento de posição para o controle PID com referência senoidal

A Figura 6.4 apresenta o erro de trajetória entre a referência e a posição medida enquanto que sinal de controle correspondente aplicado à válvula pode ser visualizado na Figura 6.5.

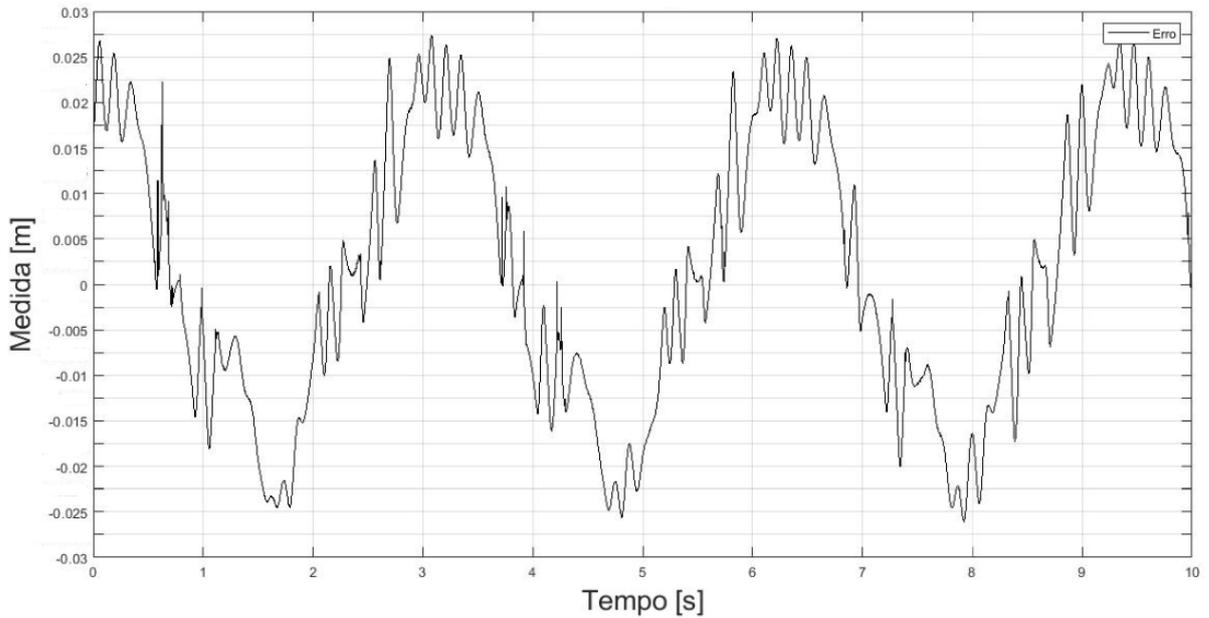


Figura 6.4 – Erros de seguimento de trajetória para o controle PID com referência senoidal

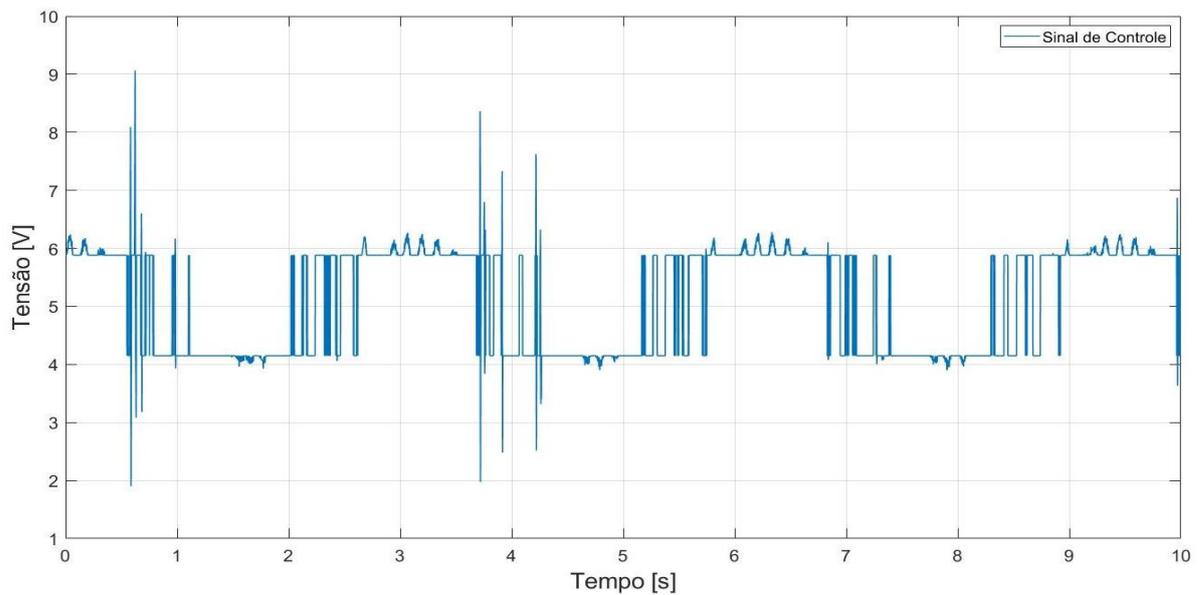


Figura 6.5 – Sinal de controle aplicado à válvula de controle direcional gerado pelo controlador PID para referência senoidal

Observa-se na Figura 6.4 que os valores de erro máximos entre +2,75 cm e -2,61 cm. Nota-se também que os erros máximos ocorrem nas extremidades das trajetórias, próximo às regiões de troca de sentido de movimento, locais onde a velocidade de deslocamento é relativamente baixa. Isto indica a forte influência da força de atrito estático no comportamento do sistema.

Na Figura 6.5, é possível verificar, através do formato do sinal aplicado, a presença de ruídos provindos do sistema de instrumentação e/ou de origem eletrostática, os quais não foram completamente suprimidos pelo sistema de filtros incorporados no sistema. Nota-se também o efeito da compensação da zona-morta da válvula proporcional no sinal de controle conforme descrito na Seção 5.4.4, onde são aplicados valores de tensão à válvula proporcional que objetivam a manutenção mínima de movimento do êmbolo do pistão, com o objetivo de dirimir efeitos transitórios ocasionados por atrito no sistema, embora estes efeitos não tenham sido completamente eliminados por esta estratégia.

6.2.1.2 Trajetória Polinomial de 7ª Ordem - PID

Para os ensaios com a trajetória polinomial de 7ª ordem, foram mantidas as mesmas características de sintonia (ganhos do controlador e valores para zona-morta) utilizados nos ensaios de trajetória senoidal, descritos na seção anterior. Os resultados para o seguimento de trajetória obtidos podem ser observados na Figura 6.6.

Os erros máximos de trajetória medidos entre a referência polinomial e a posição estão contidos na faixa de valores entre 2,77 cm e -3,04 cm durante o regime transiente e na faixa de 0,638 cm a 0,145 cm no regime estacionário. A Figura 6.7 apresenta os resultados em termos de erro de seguimento de posição enquanto que a Figura 6.8 apresenta o respectivo sinal de controle.

Nota-se que, da mesma forma como ocorrido no caso da trajetória senoidal, os picos de erro ocorrem nos trechos de início e fim de movimento, onde as velocidades de deslocamento são reduzidas. Desta forma, caracteriza-se a forte influência do atrito no desempenho para o seguimento de trajetórias.

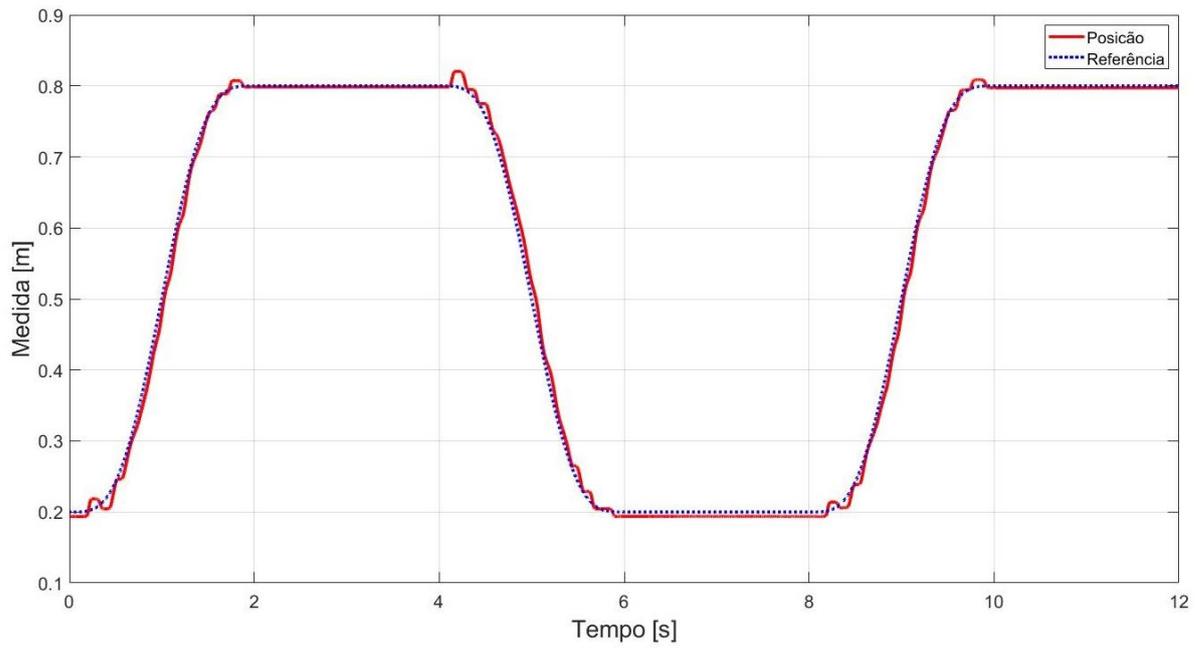


Figura 6.6 – Seguimento de trajetória para o controle PID com referência polinomial de 7ª ordem

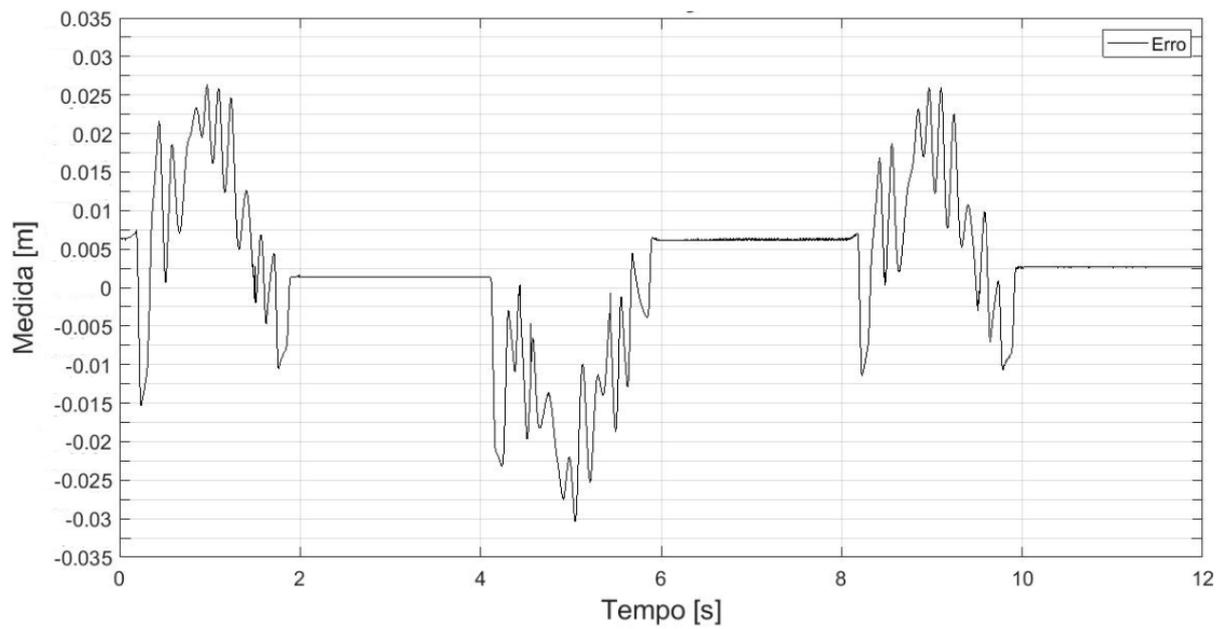


Figura 6.7 – Erros de seguimento de trajetória para o controle PID com referência polinomial

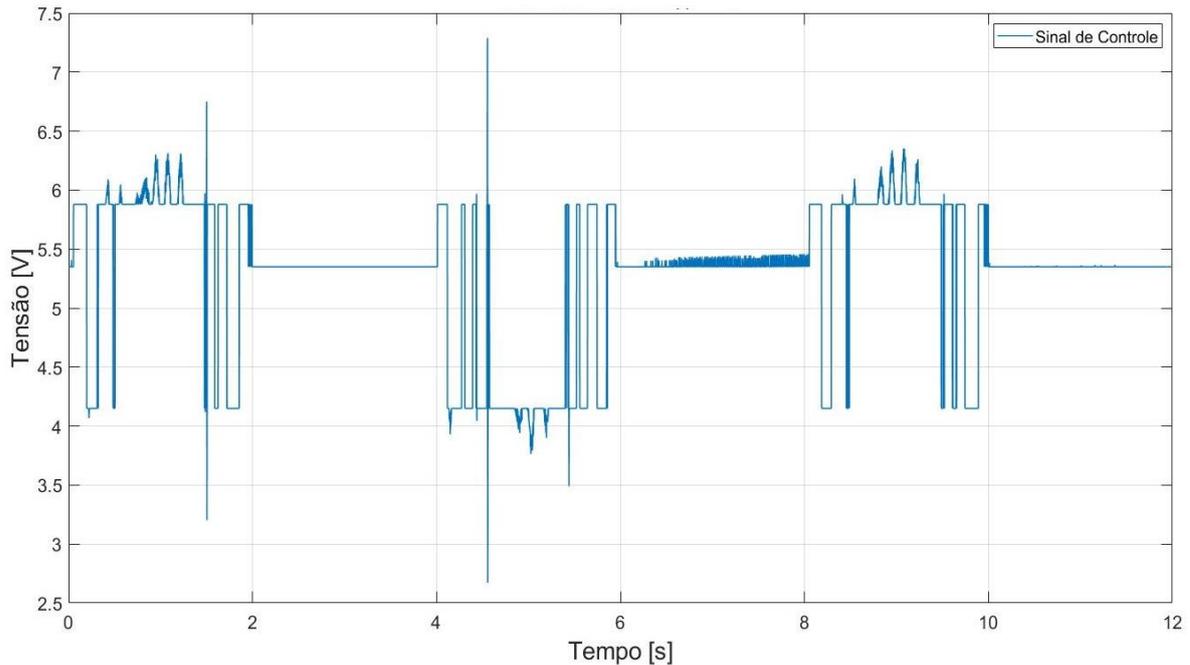


Figura 6.8 – Sinal de controle aplicado à válvula de controle direcional gerado pelo controlador PID para referência polinomial

Observa-se no sinal de controle (Figura 6.8) a faixa de tensão relacionada à compensação da zona morta da válvula. Nos períodos de tempo relacionados ao regime estacionário, o valor aplicado à válvula de controle gerado pelo algoritmo é semelhante para ambos posicionamentos do êmbolo à extremidade do atuador (5,35 V). Na faixa dos 6 a 8 segundos, é registrada a presença de transições de sinal de saída, causadas pela presença de ruído (proveniente do sensoriamento de posição, proveniente de natureza estática), o qual não foi completamente eliminado pelo sistema de filtros analógicos e digitais implementados.

6.2.2 Controle por Modos Deslizantes (SMC)

Para realização dos ensaios de controle por meio da estratégia de controle por modos deslizantes através do *hardware* implementado, foram utilizados os ganhos $S_1 = 1350$, $S_2 = 10$, $S_3 = 0.5$ e largura da camada limite com o valor $\varphi = 10$.

Da mesma forma como realizado no controlador PID, estes valores foram sintonizados experimentalmente a partir dos resultados de diversos ensaios e correspondem aos menores erros de seguimento de trajetória para as diferentes referências utilizadas.

6.2.2.1 Trajetória Senoidal – SMC

A Figura 6.9 apresenta o resultado para o seguimento de trajetória para a referência senoidal para o controle por modos deslizantes, enquanto que a Figura 6.9 apresenta os respectivos erros máximos de seguimento de trajetória.

Observa-se na Figura 6.8, quando comparada aos resultados do controlador PID, a ocorrência de uma redução significativa dos efeitos do atrito no comportamento do sistema nos pontos de troca de sentido de movimento e nas zonas de baixa velocidade.

Esta ocorrência corrobora a característica do controlador por modos deslizantes de apresentar menor sensibilidade às variações paramétricas da planta, uma vez que por conta das não-linearidades presentes nos sistemas pneumáticos, os seus efeitos alteram a dinâmica da planta de forma não-linear em função dos valores que seus estados assumem ao longo do tempo.

Nota-se através da Figura 6.10 que a maioria dos valores do erro estão contidos na faixa de ± 1 cm; sendo os valores de pico entre +1,26 cm e -0,86 cm, neste caso, não relacionados ao erro de seguimento e sim, possivelmente relacionados a ruídos provindos do sistema de sensoriamento de posição.

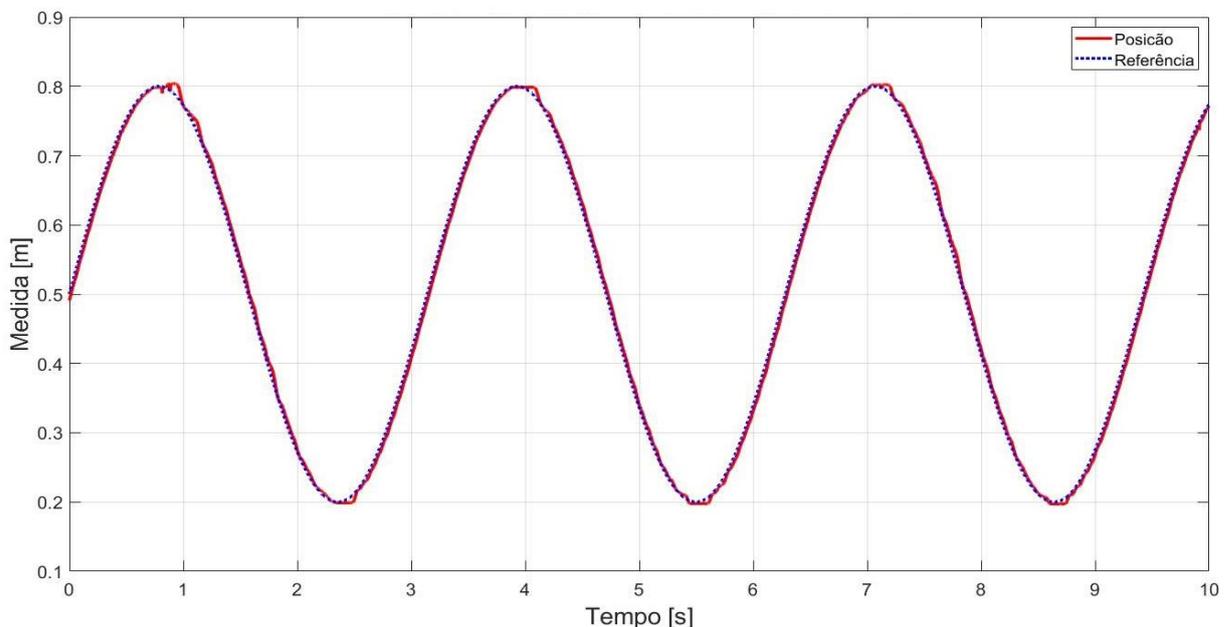


Figura 6.9 – Seguimento de posição para o controle por modos deslizantes com referência senoidal

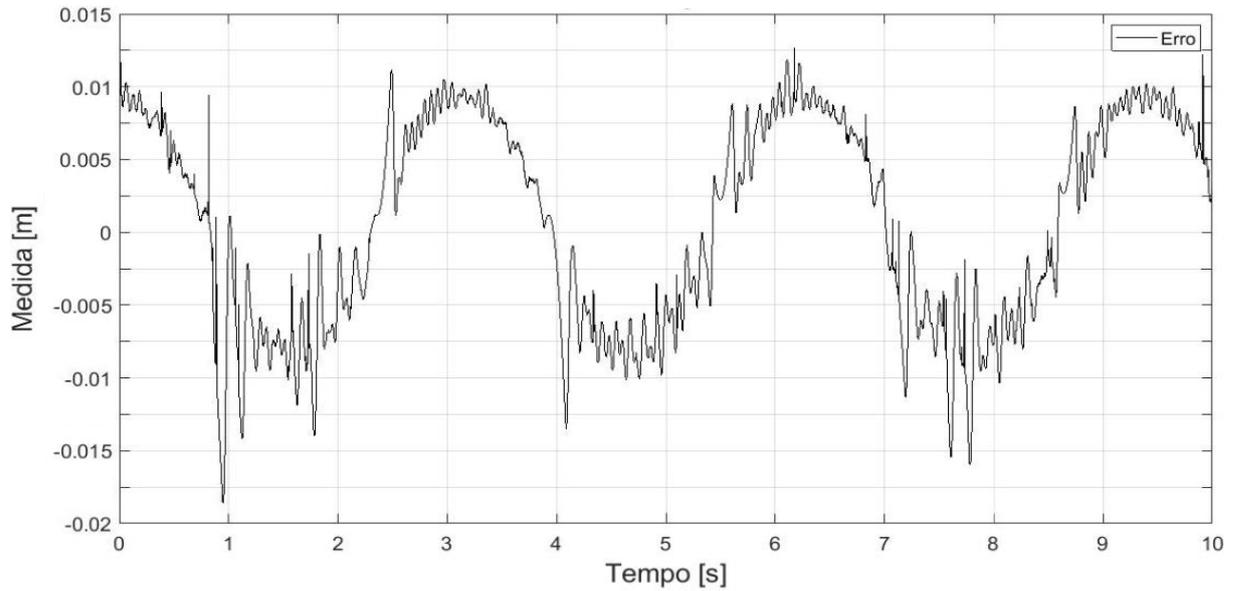


Figura 6.10 – Erros de seguimento de trajetória para o controle por modos deslizantes com referência senoidal

Na Figura 6.11 é apresentado o sinal de controle aplicado à válvula de controle. Nesta é possível observar as oscilações associadas ao chaveamento do controlador, característica relacionada ao SMC.

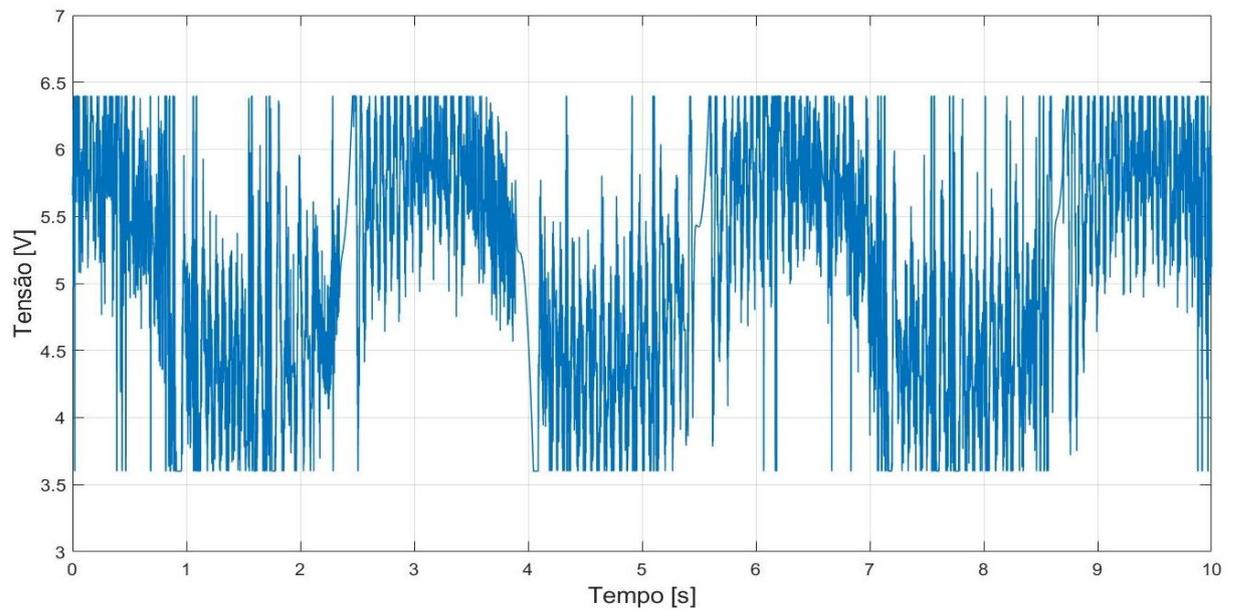


Figura 6.11 – Sinal de controle aplicado à válvula de controle direcional gerado pelo microcontrolador para referência senoidal

6.2.2.2 Trajetória Polinomial de 7ª Ordem - SMC

A Figura 6.12 apresenta o resultado para o seguimento de trajetória para a referência polinomial para o controle por modos deslizantes.

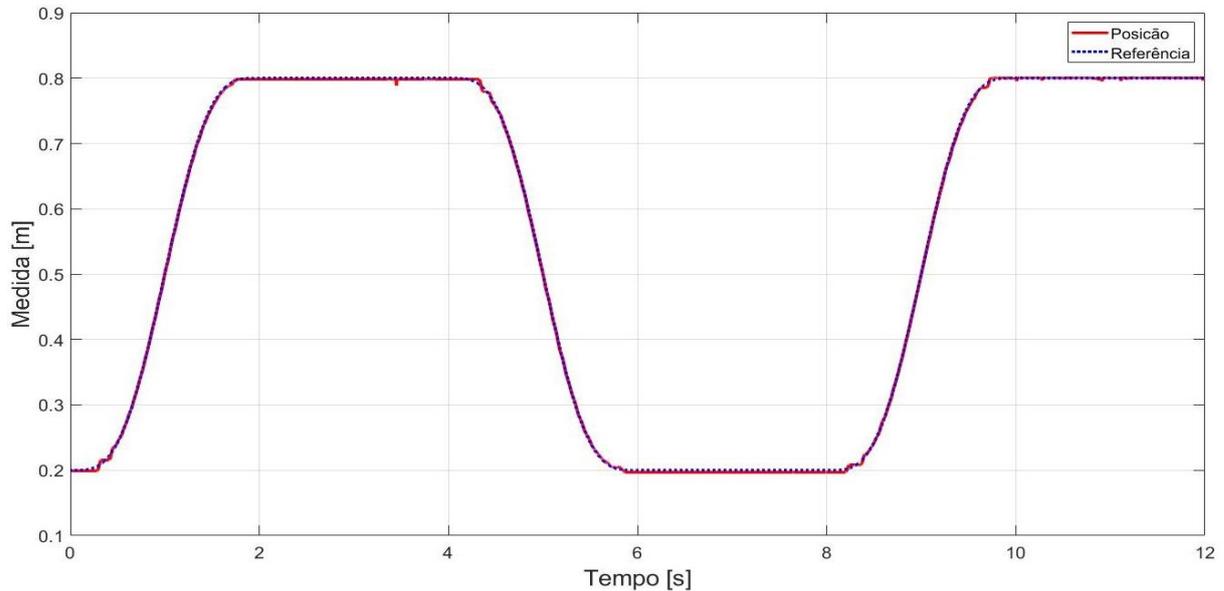


Figura 6.12 – Seguimento de posição para o controle por modos deslizantes com referência polinomial de 7ª ordem

Observa-se na Figura 6.12, da mesma forma como para a trajetória senoidal, a redução dos efeitos do atrito no comportamento do sistema, nos pontos de troca de sentido de movimento e nas zonas de baixa velocidade.

A Figura 6.13 apresenta os erros de seguimento de trajetória. Os valores de erro medidos durante o regime estacionário encontram-se na faixa de faixa de +0,31 cm e -0,05 cm enquanto que durante o regime transiente, os valores possuem picos de +1,15 cm e -0,88 cm. Estes valores máximos observados também podem ser atribuídos à ruídos estáticos provindos do sistema de aquisição de dados e a ruídos elétricos de natureza ambiental.

Por fim, a Figura 6.14 apresenta o sinal de controle aplicado à válvula. Nesta imagem é possível observar os efeitos do ruído da aquisição de dados para a válvula de controle; no trecho entre os tempos 2s e 4s (Figura 6.13) é possível observar um valor de pico máximo (+1,15 cm) provindo do sistema de medição da posição e a consequente ação do sinal de controle para correção do erro (Figura 6.14).

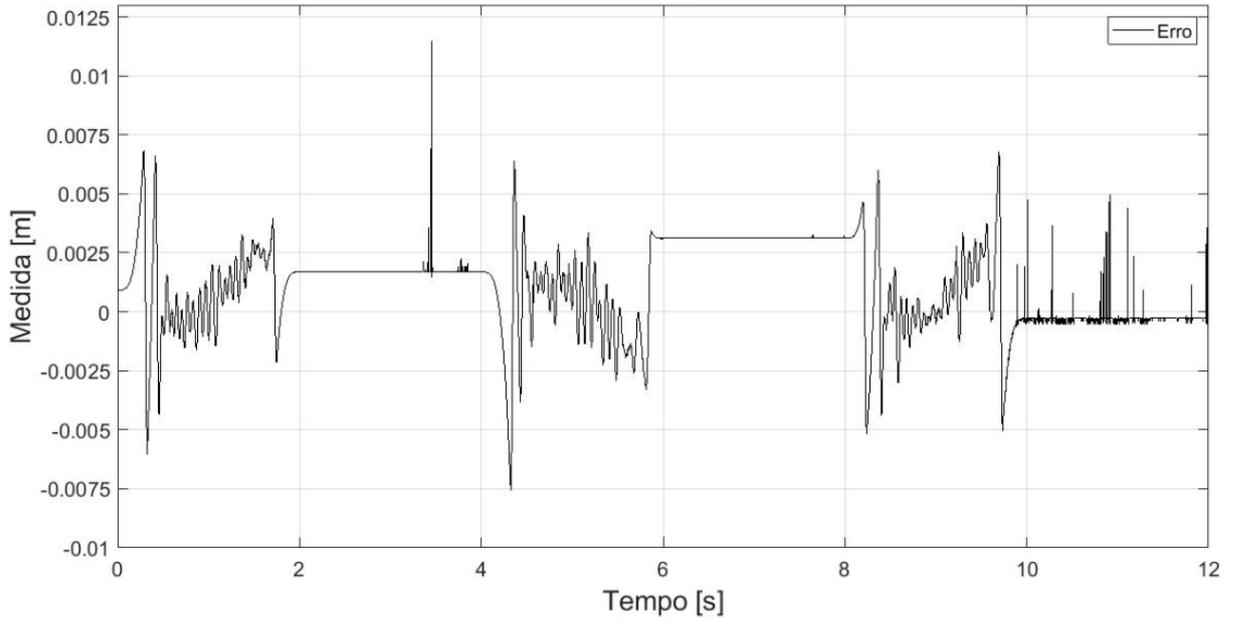


Figura 6.13 – Erros de seguimento de trajetória para o controle por modos deslizantes com referência polinomial

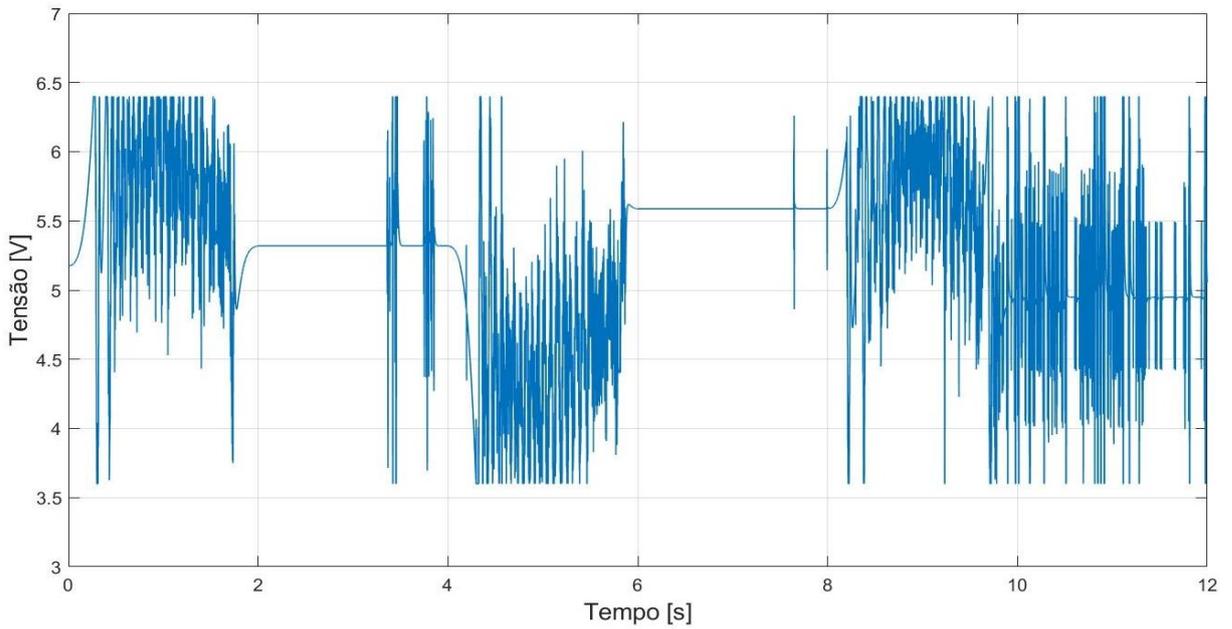


Figura 6.14 – Sinal de controle aplicado à válvula de controle direcional gerado pelo microcontrolador para referência polinomial

6.3 Discussões sobre os Resultados Obtidos

Os resultados referentes à manutenção do tempo de ciclo de 1 milissegundo, apresentados na Seção 6.1, para ambos algoritmos de controle implementados, além do sincronismo interdependente entre as duas tarefas, demonstram a eficiência e eficácia do sistema operacional de tempo real FreeRTOS como base para a implementação de algoritmos de controle cuja necessidade de temporizações restritas quanto ao tempo de resposta do controlador são essenciais.

Da mesma forma, a medida dos tempos de utilização dos núcleos de processamento do microcontrolador ESP32, para execução das tarefas programadas (controle e comunicação) totalizando tempos inferiores ao tempo de ciclo total programado, conforme apresentado nas Tabelas 6.2 e 6.3, indicam a capacidade e velocidade computacional do dispositivo, validando o seu uso para aplicações de controle.

Neste conjunto, a programação de diferentes estratégias de controle na plataforma desenvolvida permitiu a avaliação do desempenho do sistema, considerando valores erro de posicionamento no controle de uma planta de testes. A comparação entre os resultados obtidos, a partir da sintonia realizada experimentalmente, possibilita identificar as características típicas dos controladores, como a maior suscetibilidade a efeitos não lineares (como a força de atrito em regiões de baixas velocidades) no controlador PID e a maior robustez do controlador por modos deslizantes (SMC) contra variações paramétricas da planta. Para melhor visualização dos dados de erro de seguimento obtidos entre os diferentes experimentos, a Tabela 6.4 apresenta um resumo destes valores.

Tabela 6.4 – Relação dos resultados no erro de seguimento de trajetória

Controlador	Trajectoria de referência	Faixa de erro máximo no período transiente (cm)	Faixa de erro máximo no período estacionário (cm)
PID	Senoidal	+2,75 a -2,61	-
PID	Polinomial 7 ^a ordem	+2,77a -3,04	0,63 a 0,14
SMC	Senoidal	+1,26 a -1,86	-
SMC	Polinomial 7 ^a ordem	1,15 e -0,88	+0,31 e -0,05

Entretanto, os gráficos apresentados na Seção 6 indicam a presença de ruídos provenientes do sistema de instrumentação utilizado para aquisição do sinal de posição, os

quais não foram suprimidos pelo sistema. Por serem intermitentes, verifica-se a necessidade de estudos mais detalhados sobre as frequências e origem dos ruídos e conseqüentemente a aplicação de estratégias analógicas e digitais de atenuação, além de utilização de técnicas de proteção de circuitos e instrumentos contra interferências eletromagnéticas. Estes estudos deverão ser realizados em trabalhos futuros visando à melhoria das características de aquisição de dados e supressão de ruídos nas medidas de posição. Com isso, espera-se um conseqüente aumento da precisão de seguimento de trajetória do sistema.

7 CONCLUSÕES

Diante do desafio inicialmente proposto (apresentado na Seção 1.3) para o trabalho de controlar um servoposicionador pneumático por meio de um *hardware* de baixo custo, a partir das pesquisas realizadas durante a revisão bibliográfica, cujas soluções apontavam para a aplicação de técnicas de controle por algoritmos de média e alta complexidade, foi possível perceber a necessidade de desenvolvimento de um dispositivo de controle digital para servoposicionadores que, além do baixo custo, agregasse características de flexibilidade e desempenho adequados, bem como alto poder de processamento computacional e capacidade de fornecer suporte para operações em ciclos de controle com alta precisão temporal, de acordo com os conceitos de controle em tempo real.

Neste sentido, a opção pela utilização de um microcontrolador, neste caso, o microcontrolador ESP32, em operação conjunta ao sistema operacional de tempo real FreeRTOS, além de fornecer a precisão temporal necessária para o ciclo de controle desejado de 1ms, apresenta vantagens associada à disponibilidade de uma interface de gravação de *firmware* via USB, à possibilidade de utilização de *software* livre e gratuito e também referentes à utilização compiladores, ferramentas e bibliotecas compatíveis com outras plataformas de desenvolvimento, fornecidos tanto pelo fabricante quanto por uma grande comunidade de desenvolvedores de *software* independentes.

Para integralização dos demais objetivos (apresentados na Seção 1.4), foi realizada a instrumentação de uma bancada experimental de um servoposicionador pneumático. Para tanto, foi necessária a construção de interfaces de condicionamento de sinais para os sensores de posição e de pressão, necessitando de aplicação de técnicas específicas para atenuação e adequação da potência do sinal provindo dos sensores, além da redução de ruídos provenientes do ambiente e de instrumentação por filtros e estágios de amplificação diferencial. Foi necessário também realizar a calibração das faixas de operação para compatibilização com conversores AD integrados. Para o acionamento da servoválvula direcional foi necessário o desenvolvimento de um driver de potência, com o objetivo de realizar a amplificação e adequação do sinal de controle à potência necessária para o acionamento da servoválvula.

Posteriormente à montagem e interligação dos dispositivos desenvolvidos à bancada, foi realizado o projeto e aplicação de dois algoritmos de controle sob regras de programação do sistema FreeRTOS, um PID e outro por modos deslizantes com o objetivo de verificação

do desempenho para o seguimento de trajetórias. Para avaliação comparativa do desempenho foram testadas trajetórias senoidais e polinomiais. Os resultados apresentados referente aos erros máximos obtidos através dos testes experimentais evidenciaram a qualidade do desempenho controlador por modos deslizantes frente ao desempenho do controlador PID, porém, ambos mostraram a capacidade e eficácia da arquitetura de *hardware* proposta para aplicação no controle de um servoposicionador pneumático. A partir dos resultados apresentados no capítulo 6, é possível concluir que todas as tarefas propostas foram realizadas com êxito.

Pode-se destacar os seguintes pontos com relação à solução desenvolvida:

- a) **Tempo de ciclo:** as medidas apresentadas na Seção 6.1 referentes ao ciclo de controle mostram que o sistema executa as instruções relativas às tarefas de laços (ciclos) de controle em intervalos fixos e rígidos de 1ms, conforme definido no sistema via *software*. Esta característica é baseada nos resultados para os dois controladores implantados com os respectivos tratamentos de dados. Evidentemente, novas avaliações são necessárias para o caso de outros controladores mais exigentes em termos de necessidade de processamento;
- b) **Flexibilidade:** a possibilidade de pronta reprogramação das características de funcionamento por *firmware* e ajustes de *software* de diferentes estratégias de controle, via interface USB.

Finalmente, conclui-se que os resultados do presente trabalho colaboram com a aplicação de tecnologias que permitam a ampliação e popularização do uso de servoposicionadores pneumáticos para aplicações industriais.

7.1 Trabalhos Futuros

Como sugestão de trabalhos futuros para o desenvolvimento de plataformas de controle, tem-se os seguintes pontos para serem abordados:

- a) **Técnicas de redução de ruídos provenientes da instrumentação ou meio ambiente:** realização de estudos de compatibilidade e blindagem eletromagnética visando à redução de ruídos. Isso permitirá uma melhor leitura das variáveis analógicas, com conseqüente redução da frequência de corte de filtros analógicos e digitais, diminuindo o atraso dos sinais em um ciclo de controle;

- b) Utilização de sensores digitais para posicionamento:** utilização de sensores de posição digitais, como um *encoder óptico* ou de sistemas sem contato, como as técnicas eletro e magneto strictivas ou laser na determinação de distâncias pode ser uma estratégia adequada para obtenção de medidas de posicionamento com maior precisão e em formato digital, atenuando a presença de ruídos ou incertezas referentes ao processo de conversão analógico-digital;
- c) Desenvolvimento de interfaces gráficas para visualização simultânea dos valores associados às variáveis de interesse no seguimento de trajetória:** desenvolvimento de uma interface visual para a avaliação instantânea das variáveis de interesse do processo, como o seguimento de trajetória e erros máximos resultará em maior rapidez no processamento dos resultados e no processo de sintonia da malha;
- d) Modelagem matemática do sistema servopneumático a ser controlado incluindo modelagem do atrito:** existem diversos trabalhos desenvolvidos no LAMECC (e encontrado também na bibliografia) sobre a modelagem matemática de servoposicionadores pneumáticos. Uma vez caracterizados os valores dos parâmetros da bancada desenvolvida, estes modelos poderão ser utilizados para a síntese e auxílio à sintonia de controladores mais sofisticados que poderão ser programados e avaliados na plataforma de *hardware* desenvolvida;
- e) Análise estatística para os dados de erro de trajetória:** utilização de técnicas estatísticas para a determinação do erro médio e desvio padrão para os erros de seguimento de trajetória, para as referências senoidal e polinomial;
- f) Análise do sistema de medição:** estudo do sistema de medição desenvolvido sob a ótica de análise de propagação de erros e de cálculo da precisão da cadeia de medição.

REFERÊNCIAS BIBLIOGRÁFICAS

Aroca, R. V. **Análise de Sistemas Operacionais de Tempo Real para Aplicações de Robótica e Automação**. Dissertação de mestrado, Universidade de São Paulo, São Paulo, 2008.

Atmel. **ATmega32**. <http://ww1.microchip.com/downloads/en/devicedoc/2503s.pdf>, 2011, Acesso em 15-10-2019.

Axelson, J. **USB Complete: The Developer's Guide**. Lakeview Research, Madison, 4rd edition, 2009.

Borges, F. A. P. **Controle em Cascata de um Atuador Hidráulico Utilizando Redes Neurais**. Tese de doutorado, Universidade Federal do Rio Grande do Sul, 2017.

Cardoso, F. L. **Projeto de um Retificador Bidirecional com Elevado Fator de Potência com Controle por Valores Médios Instantâneos Implementado no DSP TMS320F2812**. Dissertação de mestrado, Universidade Federal de Santa Catarina, 2006.

Carvalho, J. L. M. **Sistemas de Controle Automático**. LTC, Rio de Janeiro, 2000.

Castrucci, P., e Sales, R. M. **Controle Digital**. Edgard Blücher, São Paulo, 1990.

Chiu, C., e Leung, T. P. Modeling and Microcomputer Control of a Nonlinear Pneumatic Servomechanism. **Trans. Institute of Measurement & Control**, vol. 10, p. 71-78, 1988.

Cukla, A. R. **Arquitetura Microcontrolada Programável Aplicada ao Controle de um Servoposicionador Pneumático**. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul, 2012.

Deimel, R., Radke, M. e Brock, O. Mass Control of Pneumatic Soft Continuum Actuators with Commodity Components. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, p. 774-779, 2016.

Denardin, G. W., e Barriquello, C. H. **Sistemas Operacionais de Tempo Real e sua Aplicação em Sistemas Embarcados**. Blucher, São Paulo, 2019.

Dorf, R. C. **Sistemas de Controle Moderno**. LTC, Rio de Janeiro, 7rd edition, 1989.

Dspace. **dSPACE GmbH**. <http://www.dspace.com> Acesso em: 15-10-2019.

Endler, L. **Modelagem da Vazão Mássica de uma Servoválvula Pneumática e sua Aplicação no Controle Ótimo de um Servoposicionador Pneumático**. Dissertação de mestrado, Universidade Regional do Noroeste do Rio Grande do Sul, 2009.

Espressif. **ESP32**. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf Acesso em: 20 out. 2019.

Fadali, M. S., e Visioli A. **Digital Control Engineering: Analysis and Control**. Elsevier, Oxford, 2rd edition, 2013.

Farines, J., Fraga, J., e Oliveira, R. **Sistemas de Tempo Real**. UFSC, Florianópolis, 2000.

Felício, L. C. **Modelagem da Dinâmica de Sistemas e Estudo da Resposta**. Rima, São Carlos, 2rd. edition, 2010.

Felix, R. O. **Identificação Automática de Atrito em Atuadores Pneumáticos Utilizando Equipamento de Baixo Custo**. Dissertação de mestrado, Universidade Estadual de São Paulo, 2018.

Festo. **Festo Corporation**. <http://www.festo.com> Acesso em 15-11-2019.

Fialho, A. B. **Automação Pneumática: Projetos, Dimensionamento e Análise de Circuitos**. Érica, São Paulo, 5rd. edition, 2007.

Figliola, R. S., e Beasley, D. E. **Mechanical Measurements.**: John Wiley e Sons, New Jersey, 5rd. edition, 2011.

Floyd, T. L. **Sistemas Digitais: Fundamentos e Aplicações.** Bookman, Porto Alegre, 9rd. edition, 2007.

Franklin, G. F., Powell, J. D., e Emami-Naeini, A. **Sistemas de Controle para Engenharia.** Bookman, Porto Alegre, 6rd. edition, 2013.

Freertos. **Real-time Operating System for Microcontrollers.** <http://www.freertos.org>
Acesso em 15-10-2019.

Frigieri, E. P. *et al.* Portando uma Aplicação de Sistema Embarcado com Arquitetura Superloop para Operar como Sistema Operacional de Tempo Real. **ForScienci**, vol. 2(1), p. 47-67, 2014.

Galvão, S. S. L. **Especificação do Micronúcleo FreeRTOS Utilizando o Método B.** Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, 2010.

Gentile, A., Giannoccaro, N. I., e Reina, G. Experimental Tests on Position Control of a Pneumatic Actuator Using on/off Solenoid Valves. **IEEE International Conference on Industrial Technology**, p. 555-559, 2002.

Gervini, V. I. **Modelagem e Controle de um Servoposicionador Pneumático via Redes Neurais.** Tese de doutorado, Universidade Federal do Rio Grande do Sul, 2014.

Gyeviki, J. *et al.* Sliding Modes Application in Pneumatic Positioning. **IEEE International Conference on Mechatronics**, p. 62-68, 2005.

Hai-Peng, R., e Huang, C. Adaptive Backstepping Control of Pneumatic Servo System **IEEE International Symposium on Industrial Electronics**, p. 1-6, 2013.

Haupt, A. G., e Dachi, E. P. **Eletrônica Digital.** Blucher, São Paulo, 2016.

Kunz, G. O. **Desenvolvimento de um Sistema de Controle em Tempo Real para um Servoposicionador Pneumático**. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul, 2006.

Li, R. *et al.* A Sliding Mode Variable Structure Control Approach for a Pneumatic Force Servo System. **Proceedings of the 6th World Congress of Intelligent Control and Automation**, vol. 2, p. 8173-8177, 2006.

Locateli, C. C. **Modelagem e Desenvolvimento de um Sistema de Controle de Posição Pneumático com Acionamento por Válvulas ON/OFF**. Dissertação de mestrado, Universidade Federal de Santa Catarina, 2011.

Macêdo, R. *et al.* **Tratando a Previsibilidade em Sistemas de Tempo Real Distribuídos: Especificação, Linguagens, Middleware e Mecanismos Básicos**. <http://www.lasid.ufba.br/publicacoes/artigos/tempoReal2004.pdf>, 2004. Acesso em 15-09-2019.

Macêdo, R. J. A.; Farines, J. A. **Projeto de Sistemas Distribuídos e de Tempo Real para Automação**. EDUFBA, Salvador, 2018.

Maré, J., Colin, S., e Geider, O. An Improved Dynamic Model Of Pneumatic Actuators. **International Journal of Fluid Power**, vol. 1(2), p. 39-47, 2000.

Mello, B. T., e Sarcinelli-Filho, M. **Desenvolvimento de um Sistema Embarcado De Tempo Real para Aplicações em Robótica**. <http://www.swge.inf.br/CBA2014/anais/PDF/1569935289.pdf>, 2014, Acesso em: 20-10-2019.

Microchip. **MCP3208**. <http://ww1.microchip.com/downloads/en/DeviceDoc/21298e.pdf>, 2008, Acesso em 15-10-2019.

Microchip. **MCP6001**. <http://ww1.microchip.com/downloads/en/DeviceDoc/20001733K.pdf>, 2019, Acesso em 15-10-2019.

Moreira, A. L. S. **Análise de Sistemas Operacionais de Tempo Real**. Dissertação de mestrado, Universidade Federal de Pernambuco, 2007.

Moreira, I. S. **Sistemas Pneumáticos**. SENAI-SP, São Paulo, 2rd.edition, 2012.

Najjari, B. *et al.* Position Control of an Electro-Pneumatic System Based on PWM Technique and FLC. **ISA Transactions**, vol. 53, p. 647-657, 2014.

National Instruments. **Líder Global em Sistemas de Teste e Medição Automatizados**. <https://www.ni.com/pt-br.html> Acesso em 15-10-2019.

Ning S., e Bone, G. M. Experimental Comparison of two Pneumatic Servo Position Control Algorithms. **IEEE**, Niagara Falls, 2005.

Nise, N. S. **Engenharia de Sistemas de Controle**. LTC, Rio de Janeiro, 2002.

Ogata, K. **Engenharia de Controle Moderno**. Prentice-Hall do Brasil, Rio de Janeiro, 1985.

Ogata, K. **Engenharia de Controle Moderno**. Pearson Prentice Hall, São Paulo, 5rd. edition, 2010.

Oliveira Neto, B. B. de, Monteiro, P. F., e Queiroga, S. L. M. **Aplicabilidade dos Microcontroladores em Inovações Tecnológicas**. <http://propi.ifto.edu.br/ocs/index.php/connepi/vii/paper/viewFile/2433/2526>, 2012, Acesso em 15-10-2019.

Oliveira, T. B. **Sistemas Operacionais de Tempo Real: Características e Principais Implementações Existentes**. https://www.researchgate.net/publication/267805789_Sistemas_Operacionais_de_Tempo_Real_-_Caracteristicas_e_Principais_Implementacoes_Existentes. 2005, Acesso em 20-10-2019.

Oliveira, W. P. **Arquitetura de Software para Sistemas de Tempo Real**. Dissertação de mestrado, Universidade de São Paulo, 2011.

Pandian, S. R. *et al.* Practical Design of a Sliding Mode Controller for Pneumatic Actuators. **Journal of Dynamic Systems, Measurement, and Control**, vol. 119, p. 666-674, 1997.

Parker. **This is Parker**. <https://www.parker.com/>, Acesso em 15-10-2019.

Perondi, E. A. **Controle Não-Linear em Cascata de um Servoposicionador Pneumático com Compensação do Atrito**. Tese de doutorado, Universidade Federal de Santa Catarina, 2002.

Perondi, E. A.; Guenter, R. Controle de um Servoposicionador Pneumático por Modos Deslizantes, **Anais do Congresso Nacional de Engenharia Mecânica**, 2000.

Pinheiro, B. C. **Sistema de controle tempo real embarcado para automação de manobra de estacionamento**. Dissertação de mestrado, Universidade Federal de Santa Catarina, 2009.

Politerm. **Instrumentos de Medição**. <http://www.politerm.com.br>, Acesso em 14-08-2019.

Qi, H., Bone, G.M., e Zhang, Y. Position Control of a Pneumatic Actuators Using Discrete-Valued Model Predictive Control. **Actuators**, vol. 8(56), 2019.

Qiong, W., e Jiao, Z. Modeling and Analysis of Pneumatic Loading System. **Proceedings of IEEE International Conference On Control Applications**, p. 642- 646, 2011.

QNX Neutrino. **Qnx Neutrino Realtime Operating System**. <http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino%2Fbookset.html>, Acesso em 15-10-2019.

Rad, C., e Hancu, O. An Improved Nonlinear Modelling and Identification Methodology of a Servo-Pneumatic Actuating System With Complex Internal Design for High-Accuracy Motion Control Applications. **Simulation Modelling Practice and Theory**, vol. 75, p. 29-47, 2017.

Regnier, P., Lima, G., e Barreto, L. **Avaliação do Determinismo Temporal no Tratamento de Interrupções em Plataformas de Tempo Real linux**. <http://www.lasid.ufba.br/publicacoes/artigos/wso2008-final.pdf>, 2008, Acesso em 15-09-2019.

Richter, E., e Hurmuzlu, Y. A High-Performance Pneumatic Force Actuator System. Part I – Nonlinear Mathematical Model. **Journal of Dynamic Systems, Measurement, and Control**, vol. 122, p. 416-425, 2000.

Rios, C. F. **Projeto e Construção de um Robô Cartesiano com Acionamento Pneumático**. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul, 2009.

RTAI. **Real Time Application Interface: Oficial Website**. <http://www.rtai.org>. Acesso em 15-10-2019.

Salim, S. N. S. *et al.* Position Control of a Pneumatic Actuator Using Self-Regulation Nonlinear PID. **Mathematical Problems in Engineering**, vol. 1, p. 1-12, 2014.

Santos, C. A. M. dos. **Sistema Dinâmico de Economia de Energia em RTOS**. Dissertação de mestrado, Universidade Federal de Itajubá, 2017.

Santos, J. A., e Lima, G. **Implementação da política EDF no Xenomai**. <http://www.lasid.ufba.br/publicacoes/artigos/Implementa%E7%E3o+da+pol%EDtica+EDF+no+Xenomai.pdf>, 2009, Acesso em 20-10-2019.

Saravanakumar, D., Mohan, B., e Muthuramalingam, T. A Review on Recent Research Trends in Servo Pneumatic Positioning Systems. **Precision Engineering**, vol. 49, p. 481-492, 2017.

Schuler, C. **Eletrônica II**. AMGH, Porto Alegre, 7rd. edition, 2013.

Schlüter, M. S. **Controle não Linear Adaptativo com Compensação de Atrito de um Manipulador Scara com Acionamento Pneumático**. Tese de doutorado, Universidade Federal do Rio Grande do Sul, 2018.

Silicon Lab. **ANP6001**. Data sheet, <https://www.silabs.com/documents/public/application-notes/an118.pdf>, 2019, Acesso em 25-02-2020.

Silva, A. P. M. **Uso do DSP na Instrumentação Associada a Medição Distribuída de Temperatura**. Monografia de especialização, Universidade Federal de Pernambuco, 2008.

Sobczyk Sobrinho, M. R. **Controle em cascata e à estrutura variável com adaptação de parâmetros e compensação de atrito de um servoposicionador pneumático**. 2009. Tese (Doutorado em Engenharia Mecânica) – Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

Song, J., Bao, X., e Ishida, Y. An Application of MNN Trained by Meka for the Position Control of Pneumatic Cylinder. **International Conference on Neural Network**, vol. 2, p. 829-833, 1997.

Surgenor, B. W., Vaughan, N. D., e Uebeling, M. Continuous Sliding Mode Control of a Pneumatic Positioning. **Proceedings of 8th Bath International Fluid Power Workshop**, p. 270-285, 1995.

Suzuki, R. M. **Controle Baseado em Linearização por Realimentação dos Estados Aplicado a um Servoposicionador Pneumático**. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul, 2010.

Tanaka, K. *et al.* Multi-rate Adaptative Pole-Placement Control for Pneumatic Servo System with Additive External Forces. **Proceedings of 4th IEEE International Workshop on Advanced Motion Control**, p. 213-218, 1996.

Takosoglu, J. E.; Dindorf, R. F.; Laski, P. A. Rapid Prototyping of Fuzzy Controller Pneumatic Servo System. **International Journal of Advanced Manufacturing Technology** **4.0**, p. 349-361, 2009.

Texas Instruments. **LM324-MIL Quadruple Operational Amplifier**. Data sheet, <http://www.ti.com/lit/ds/symlink/lm324-mil.pdf>, 2017, Acesso em 15-10-2019.

Thomazini, D., e Albuquerque, P. U. B. de. **Sensores Industriais: Fundamentos e Aplicações**. Érica, São Paulo, 4rd. edition, 2005.

Viegas, H. C. F. **Estudo, Implementação e Avaliação de um Sistema de Tempo Real na Plataforma RaspberryPi**. Dissertação de mestrado, Universidade de Lisboa, 2015.

Vieira, A. D. **Análise Teórico Experimental de Servoposicionadores Lineares Pneumáticos**. Dissertação de mestrado, Universidade Federal de Santa Catarina, 1998.

Xenomai. **Xenomai real time core**. <https://gitlab.denx.de/Xenomai/xenomai> Acesso em 15-10-2019.

Wika do Brasil. **Grupo Wika**. www.wika.com.br Acesso em 12-10-2019.

APÊNDICE A – Circuito de Condicionamento de Sinais para o MCP3208 / ESP32

No circuito de condicionamento de sinais elaborado para adequação dos níveis de potência elétrica (tensão e corrente) provindo dos dois transdutores de pressão e da régua potenciométrica em relação aos valores admitidos para operação do conversor analógico digital MCP3208, foi necessária a inclusão de blocos de circuitos eletrônicos adicionais com o objetivo de melhorar o processo de aquisição dos sinais com a maior estabilidade, linearidade e redução de ruídos no processo de aquisição de dados.

Para garantia de estabilidade de alimentação de tensão elétrica para os circuitos e sensores ligados à planta, foi inserido estágio de regulação de tensão elétrica a partir de reguladores de tensão à circuito integrado, para as tensões de $24V_{DC}$, que alimenta os sensores de pressão e régua potenciométrica, e $5V_{DC}$ para alimentação dos circuitos ativos de isolamento de sinais, filtragem e demais componentes do condicionador. A Figura A.1 apresenta o diagrama do estágio de regulação de tensão DC-DC.

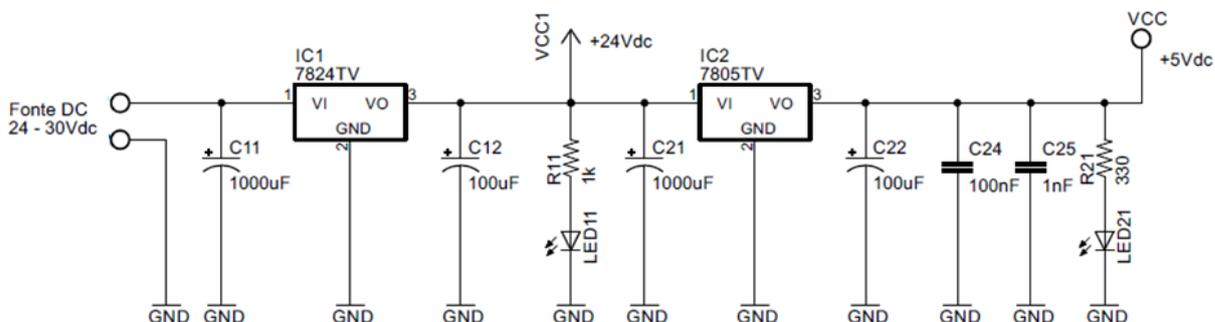


Figura A.1 – Estágio para regulação de tensão elétrica

O segundo estágio, cuja função é de atenuação (redução) do sinal do sinal de entrada a níveis adequados, é implementado por meio de um circuito denominado divisor resistivo. Para este foram consideradas as seguintes premissas:

- os níveis de tensão elétrica de saída da régua potenciométrica (Festo MLO-POT-1000-TLF) variam de 0 a 24V conforme a posição do cursor sobre a régua, com corrente elétrica máxima de 4mA [Festo, 2019];

b) os sensores de pressão (Wika S-10) fornecem saída também proporcional ao sinal medido com limites entre 0 e 10V com corrente elétrica de 1mA [Wika do Brasil, 2019].

Na Figura A.2 é apresentada a configuração de circuito eletrônico proposta para a adequação do sinal de tensão elétrica provindo da régua potenciométrica. Para permitir um ajuste da atenuação do sinal de entrada, em função das possíveis variações de resistência por conta da faixa de tolerância de resistores comerciais comuns (5%), um resistor variável (R_{303}) foi colocado em paralelo com o resistor R_{302} a fim de se obter a possibilidade de correção de tensão elétrica por ajuste de resistor via *hardware*.

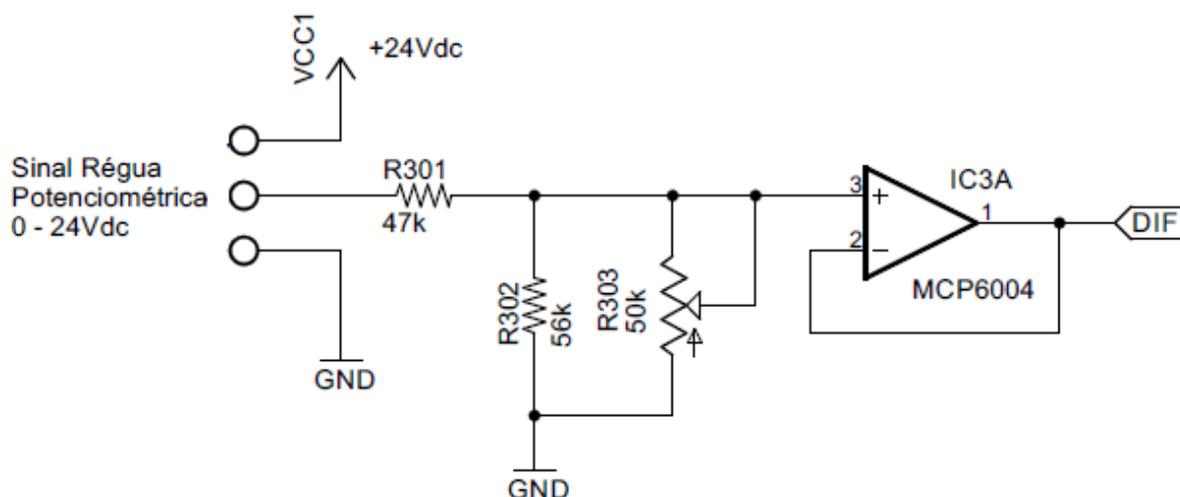


Figura A.2 – Circuito de atenuação de sinais

A tensão elétrica obtida na saída deste divisor resistivo pode ser obtida a partir da Equação A.1.

$$V_{out} = \frac{R_{302} \cdot R_{303}}{R_{301}(R_{302} + R_{303}) + R_{302} \cdot R_{303}} \cdot V_{in} \quad (\text{A.1})$$

Ao todo, foram dimensionadas três entradas individuais para condicionamento dos sinais de forma independente. Os resistores escolhidos para esta relação foram $R_{301} = 47k\Omega$, $R_{302} = 47k\Omega$ e $R_{303} = 50k\Omega$ em formato de resistor variável (*trimpot* tipo multivoltas). Estes valores de resistência elétrica no conjunto foram determinados de forma a garantir níveis de corrente elétrica adequada para aplicação ao próximo estágio. Ao final do circuito divisor resistivo, o sinal é aplicado a um de *buffer* ou amplificador isolador, formado pelo componente IC3A, consistindo na utilização de um amplificador operacional na configuração

de *seguidor de tensão*. Esta configuração é utilizada para conectar um estágio de alta impedância a uma carga de baixa impedância, onde o sinal de saída apresenta exatamente as mesmas características do sinal de entrada (valores de tensão à medida do tempo) sem atenuações ou atrasos consideráveis no sinal. Amplificadores com *buffer* são utilizados para isolar fontes de sinal de quaisquer efeitos de carga, sendo também úteis quando se trabalha com fontes de sinal que possuem impedâncias internas elevadas [Schuler, 2013]. A relação de sinais de entrada e saída em um seguidor de sinal obedece à Equação A.2.

$$V_{out} = V_{in} \quad (\text{A.2})$$

O amplificador operacional utilizado nos estágios de isolamento e filtragem é o MCP6004 da fabricante Microchip Technology Inc. [Microchip, 2019], em função de suas características elétricas e de operação compatíveis com as interfaces do conversor analógico-digital MCP3208 [Microchip, 2008] e do microcontrolador ESP32 [Espressif, 2019], além da facilidade de obtenção do componente no mercado nacional. A Tabela A.1 apresenta as características observadas para a escolha deste componente.

Tabela A.1 – Características do amplificador operacional MCP6004 [Adaptado de Microchip, 2019]

Alimentação	1,8 a 6,0 V _{DC} / 100 μA
Nº amplificadores em um CI	4
Tensão mínima de entrada	4,5 mV
Corrente mínima de entrada	1 pA
<i>Slew rate</i>	0,6 V/μs
Largura de banda	1 MHz
Corrente saída por pino	30 mA
CMRR	80 dB

Por serem analógicos, os valores de tensão elétrica provindos dos sensores de posição e de pressão podem estar contaminados por sinais interferentes ao circuito de diferentes origens, como ruídos conduzidos (cabos, aterramento), irradiados ou artificiais (harmônicos de rede elétrica, emissores de RF, comutação de circuitos de potência), naturais (descargas atmosféricas), etc. Por estes motivos, foi incluído ao circuito um estágio de amplificador diferencial em modo comum, que tem por objetivo de atenuar a entrada de sinais indesejados

de mesma amplitude, frequência e fase atuantes em ambas entradas do amplificador operacional. A Figura A.3 apresenta o circuito de amplificação diferencial.

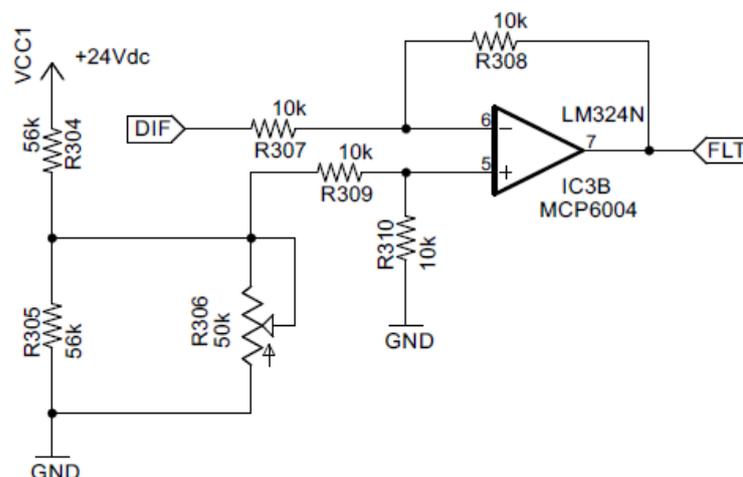


Figura A.3 – Amplificador diferencial com rejeição de ruído em modo comum

O quarto estágio é composto de um filtro analógico passa-baixas com frequência de corte de 200Hz, desenho Butherworth, com o intuito de redução do ruído produzido pelo atrito entre componentes mecânicos da régua potenciométrica. A Tabela 4.11 (ver Seção 4.3.6) apresenta as características do filtro projetado, enquanto a Figura A.4 mostra o circuito implementado.

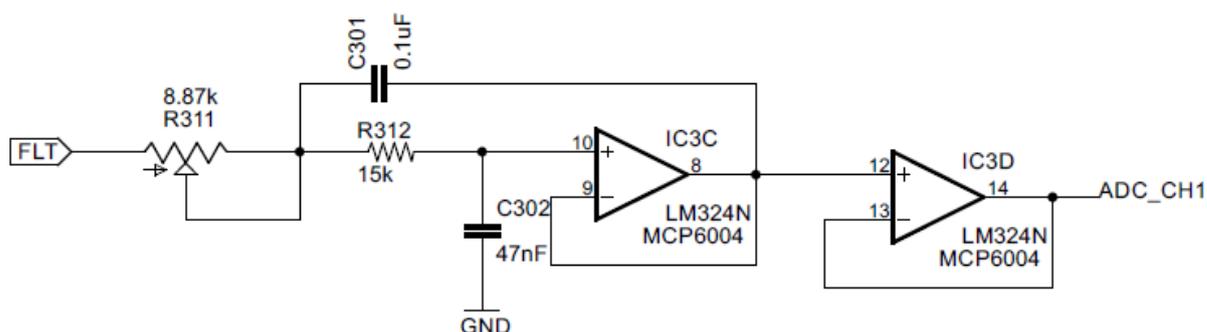


Figura A.4 – Circuito filtro Butterworth implementado

O estágio de filtro analógico foi atribuído somente ao sinal provindo do sensor de posição; para os sinais referentes às pressões das câmaras não foi atribuído o processo de filtragem. O circuito representado pela Figura A.5 representa o circuito de condicionamento implementado. Para este bloco, foram utilizados componentes comerciais (tolerâncias de

capacitores e resistores na ordem de 5%), resistores variáveis tipo trimpot multivoltas, além de uma unidade de amplificador operacional disponível no mesmo CI do MCP6004. Os resistores variáveis foram calibrados para os valores de referência calculados pelo *software*.

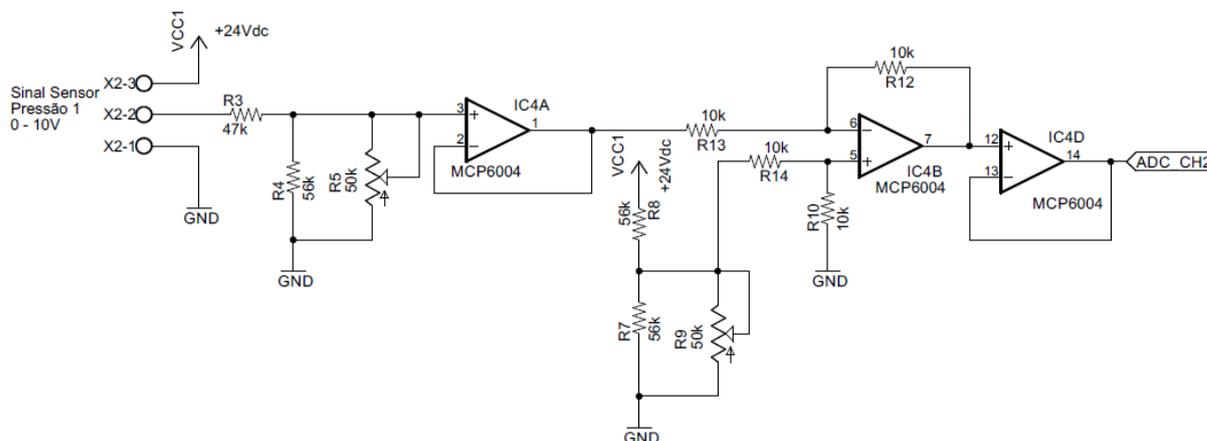


Figura A.5 – Circuito de condicionamento de sinal para sensores de pressão

A frequência de corte e características do filtro utilizada nesta etapa consideraram o compromisso entre o ruído aceitável e o atraso proporcionado ao sinal. A Figura A.6 apresenta a medição do atraso de tempo introduzido ao sistema pelo filtro analógico na ordem de 500 μ s. O atraso no sinal deve ser preferencialmente inferior a 1 ms em função da frequência do ciclo de controle utilizado. Para realização desta medida foi utilizado o osciloscópio Agilent, modelo DSO3062A – 60 MHz.

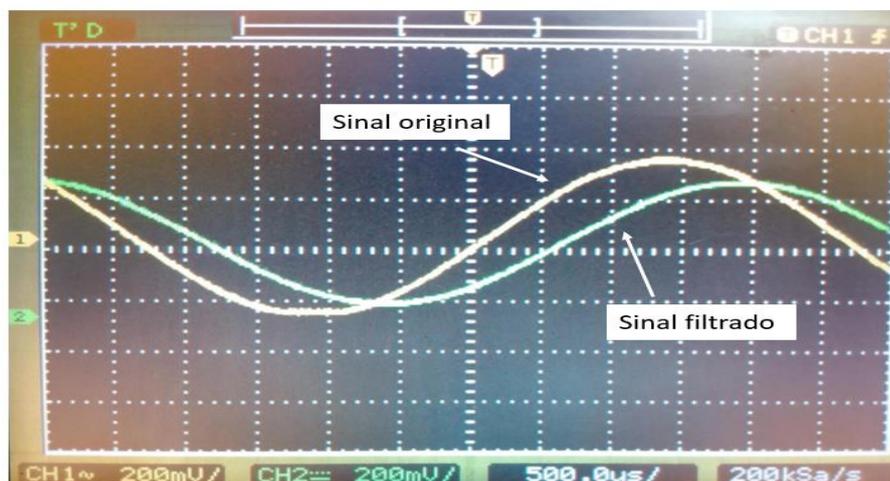


Figura A.6 – Atraso no sinal pelo estágio de filtro

O último estágio, antes do envio do sinal ao componente de conversão analógico-digital, corresponde identicamente ao amplificador isolador, já descrito anteriormente. O diagrama completo implementado do condicionador desenvolvido pode ser verificado na Figura A.7.

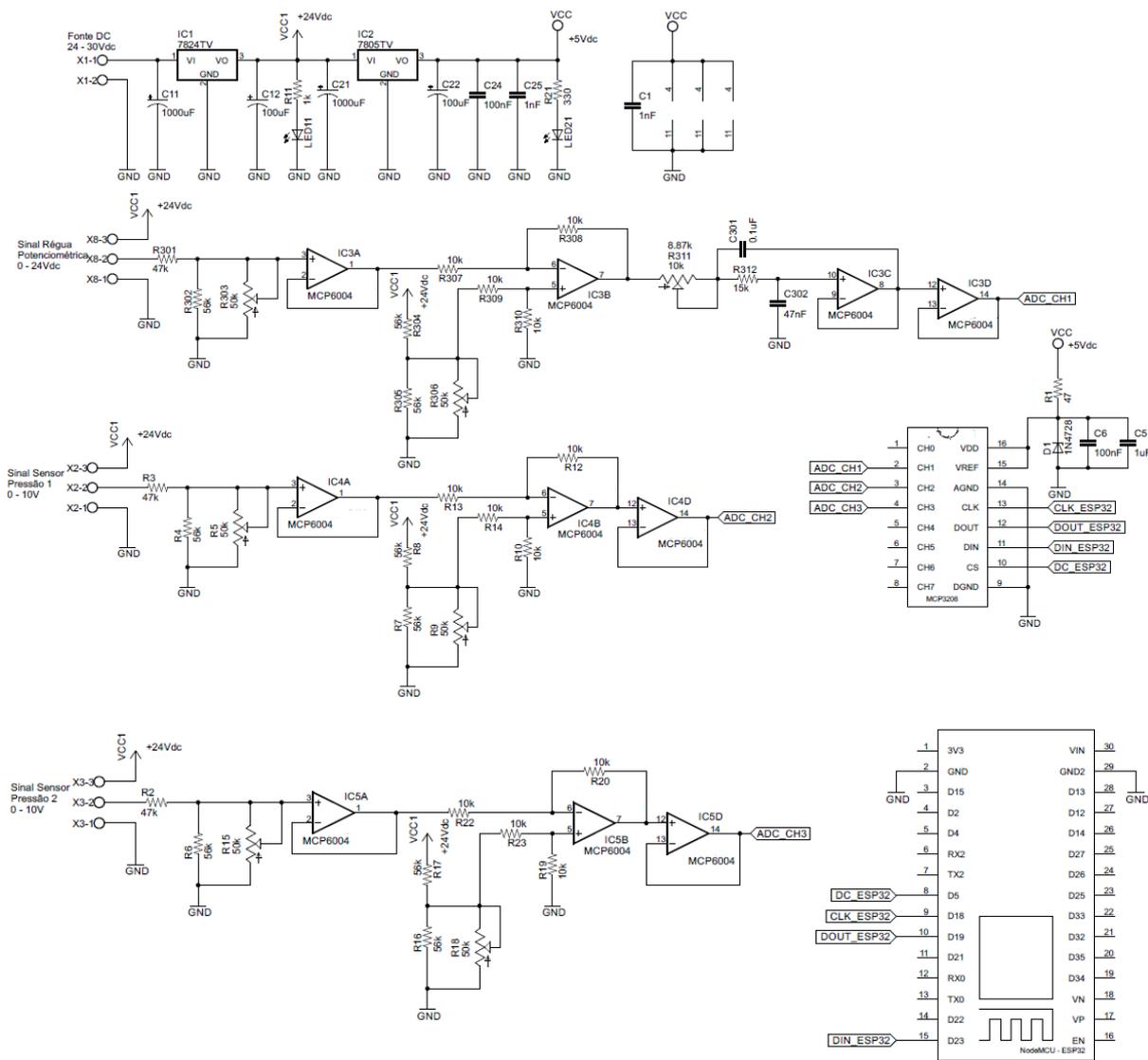


Figura A.7 – Diagrama completo do condicionador de sinais

APÊNDICE B – Circuito Driver para Acionamento da Servoválvula

Com o objetivo de garantir a qualidade e estabilidade da alimentação de energia para os componentes eletrônicos do circuito e também com o intuito de diminuir ruídos elétricos por chaveamento foi inserido no projeto dois estágios de regulação de tensão elétrica (regulagem DC-DC) a partir de reguladores de tensão em circuito integrado (LM7824 e LM7812), para as tensões de 24 V_{DC} e 12 V_{DC} respectivamente. Os capacitores de filtro (0.33 μF e 0.1 μF) para estabilização do sinal da fonte são recomendados pelo fabricante, conforme informações de *datasheet* (folha de dados) dos reguladores. Um *led* também foi adicionado à fonte de 12 V_{DC} como indicador visual do funcionamento do circuito. A Figura A.8 mostra o diagrama esquemático do regulador de tensão de alimentação da placa.

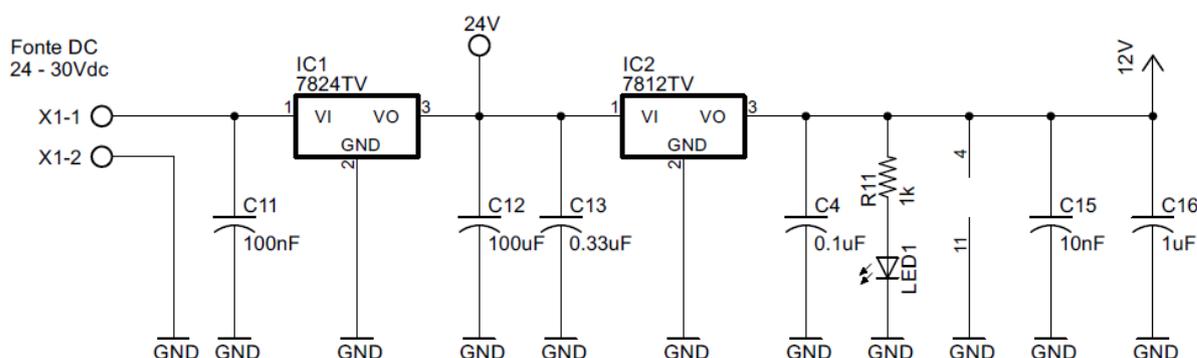


Figura A.8 – Esquema do regulador DC-DC

O segundo e o quarto estágio do driver consistem no arranjo de amplificador de ganho unitário (seguidor de tensão ou *buffer*), que tem o objetivo de isolar eletricamente o sinal provindo do controlador eletrônico, além de garantir o casamento de impedância entre sinais de saída do controlador digital e o driver. No terceiro estágio, o *ampop* (amplificador operacional) é configurado como amplificador não inversor, sendo que o *trimpot* (resistor variável R_{32}) permite o ajuste do ganho deste estágio de acordo com o valor desejado na saída. O amplificador operacional sob a configuração de amplificador não inversor apresenta comportamento de acordo com a Equação A.3.

$$V_{out} = \left(1 + \frac{R_{32}}{R_{31}}\right) V_{in} \quad (\text{A.3})$$

Neste trabalho, o valor do resistor fixo $R_{31} = 100\Omega$ e do resistor variável $R_{32} = 35,8k\Omega$ foi determinado para que fosse possível o ajuste manual do ganho para valores aproximados na ordem de $\sim 3,3 \times$, de forma a garantir amplitude de tensão elétrica proporcional de 0 a 10V na saída do bloco. A Figura A.9 apresenta os estágios implementados para adequação e ajuste do sinal antes da etapa de amplificação de potência.

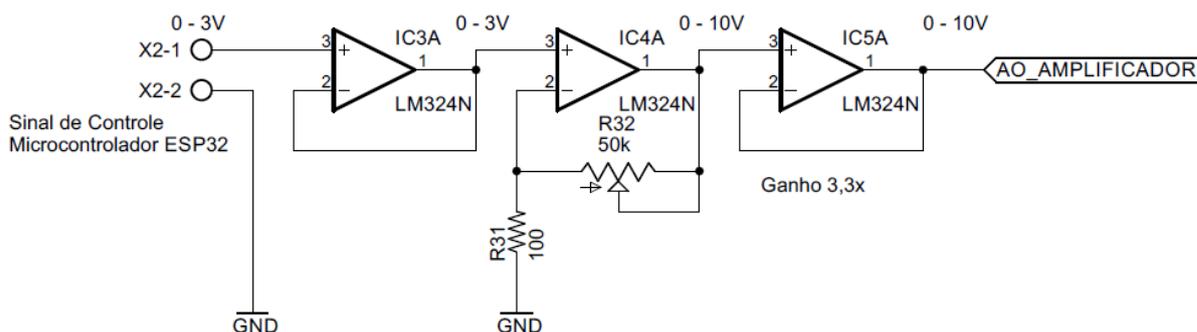


Figura A.9 – Isolação e amplificação do sinal gerado pelo microcontrolador

Para as etapas de amplificação e isolamento de sinal / *buffer*, foi utilizado o *ampop* LM324 da fabricante Texas Instruments Inc., em função de oferecer características de funcionamento adequadas ao projeto em diferentes configurações, além de possuir 4 amplificadores em um mesmo CI, o que colabora com a redução do tamanho físico necessário para acomodação dos componentes em placas de circuito impresso [Texas Instruments, 2017]. A Tabela A.2 descreve as principais características deste componente.

Tabela A.2 – Características do *ampop* LM324 [Adaptado de Texas Instruments, 2017].

Alimentação	3 a 32 V _{DC} / 700 μA
Nº amplificadores em um CI	4
Tensão mínima de entrada	2 mV
Corrente mínima de entrada	5 nA
<i>Slew rate</i>	0,5 V/μs
Largura de banda	1 MHz
Corrente saída por pino	30 mA

O sinal do microcontrolador amplificado e isolado eletricamente é aplicado a quinta e última etapa, denominada etapa de potência, na qual se utiliza de um circuito em arranjo de

amplificadores classe AB. Os amplificadores classe AB produzem uma distorção mínima e rendimento razoável [Schuler, 2013] e são apropriados para produzir na saída, sobre a carga, o efeito combinado das correntes de coletor, de forma a garantir níveis de corrente elétrica necessários para o acionamento da servoválvula. Os dois transistores TIP31 e TIP32 são dispositivos complementares (NPN e PNP) que possuem boa simetria operacional. O diagrama esquemático do estágio de potência pode ser visualizado na Figura A.10.

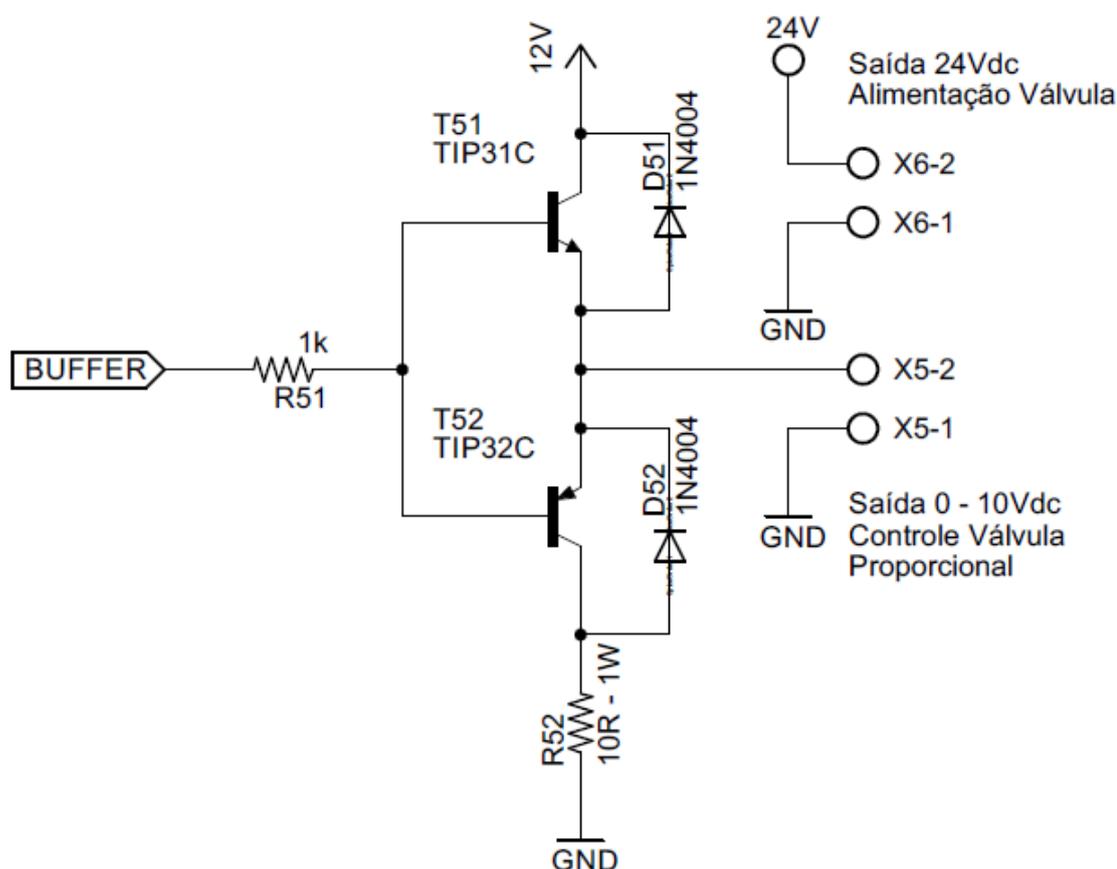


Figura A.10 – Circuito de potência em arranjo de amplificador classe AB

Os demais componentes utilizados nesta implementação (resistores, capacitores, diodos, *trimpots*) são de uso comum no segmento eletrônico, de baixa potência e com tolerância nominal a ordem de 5%.

Para realização de testes no driver desenvolvido, foi aplicado um sinal de entrada em formato tipo dente de serra em rampa ascendente, range de tensão entre 0 e +3 Vdc, com frequência de 1 kHz, permitiram a observação da linearidade correspondente da etapa de amplificação. A Figura 55 demonstra a plotagem do sinal de saída amplificado em razão de 0 a +10 Vdc, enquanto no quadro inferior da mesma imagem é verificado o sinal aplicado à

entrada. Em análise dos dados adquiridos sobre os valores numéricos dos sinais de tensão, permitiram a determinação da correlação média do ganho deste estágio em um valor de $3,15\times$. Foi identificado um erro médio de $+0,14 V_{DC}$ (cerva de 1,45%) referente a não linearidade da rampa de subida após a amplificação. É possível observar a estimativa da distorção do sinal nos regimes de ciclo ativo / inativo em 3,5%; dado que pode ser analisado em relação a flutuações médias causadas por eventuais ruídos na geração de sinal de entrada. Para realização desta medida foi utilizado o osciloscópio Agilent, modelo DSO3062A – 60 MHz além de um gerador de funções marca Rigol, modelo DG2041A de 40 MHz.

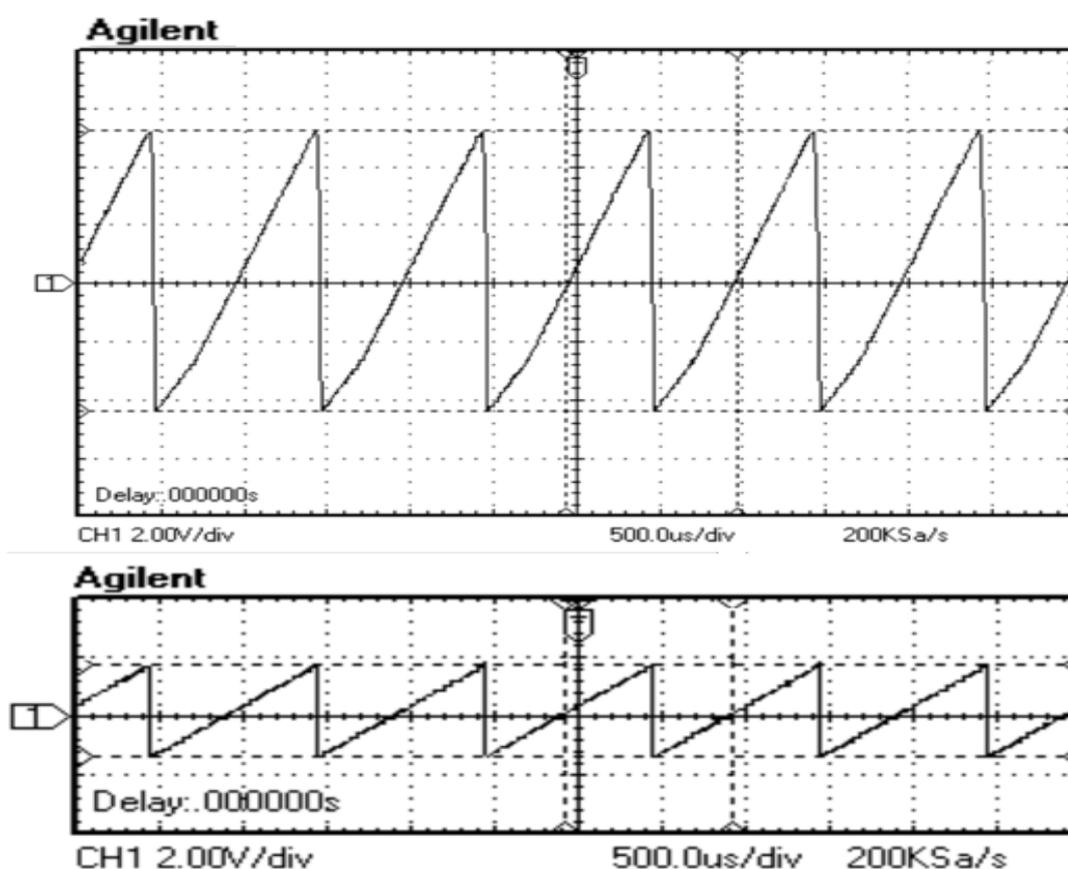


Figura A.11 – Linearidade do sinal de saída (gráfico superior) amplificado em relação ao sinal de entrada (gráfico inferior)

APÊNDICE C – Custos de Aquisição de Componentes

A Tabela A.3 apresenta os valores de aquisição dos componentes para a implementação do controlador digital. Não serão incluídos nesta tabela os valores relacionados a sensores de pressão, régua potenciométrica e demais componentes da banca (atuador, regulador de pressão, etc.).

Tabela A.3 – Custos dos componentes do controlador digital

Componente	Qtde	R\$ Unit.	R\$ Total
Microcontrolador - ESP32	1 PC	R\$ 49,90	R\$ 49,90
Conversor analógico digital - MCP3208	1 PC	R\$ 23,00	R\$ 23,00
Amplificador operacional - MCP6004	3 PC	R\$ 12,00	R\$ 36,00
Amplificador operacional - LM324	2 PC	R\$ 3,60	R\$ 7,20
Reg. Tensão – (LM7805 / LM7812 / LM7824)	4 PC	R\$ 2,50	R\$ 10,00
Transistores <i>Darlington</i> - TIP31C / TIP32C	2 PC	R\$ 2,80	R\$ 5,60
Diodos <i>Zener</i> , de sinal e retificadores	3 PC	R\$ 0,20	R\$ 0,60
Resistores diversos	32 PC	R\$ 0,10	R\$ 3,20
Capacitores eletrolíticos / poliéster	17 PC	R\$ -	R\$ 16,60
Potenciômetros / Trimpot	8 PC	R\$ 0,80	R\$ 6,40
Diodos LED 5 mm	3 PC	R\$ 0,20	R\$ 0,60
Conectores / Borne para PCI	8 PC	R\$ 0,90	R\$ 7,20
Placa cobreada circuito impresso (10x15 cm)	2 PC	R\$ 4,90	R\$ 9,80
Soquetes DIL (14 e 16 pinos)	5 PC	R\$ 0,50	R\$ 2,50
Cabo blindado 2 vias	2 m	R\$ 2,80	R\$ 5,60
		Total	R\$ 186,20