



UNIVERSIDADE ESTADUAL PAULISTA

Programa de Pós-graduação em Engenharia Mecânica

Roger Oliva Felix

IDENTIFICAÇÃO AUTOMÁTICA DE ATRITO EM
ATUADORES PNEUMÁTICOS UTILIZANDO
EQUIPAMENTO DE BAIXO CUSTO

Bauru

2018

Roger Oliva Felix

IDENTIFICAÇÃO AUTOMÁTICA DE ATRITO EM
ATUADORES PNEUMÁTICOS UTILIZANDO
EQUIPAMENTO DE BAIXO CUSTO

Dissertação apresentada à Faculdade de Engenharia de Bauru, Programa de Pós-Graduação em Engenharia Mecânica na Área de Projeto Mecânico, como parte dos requisitos necessários à obtenção de Título de Mestre em Engenharia Mecânica.

Orientador: Prof. Dr. Marcos Silveira

Bauru

2018

Felix, Roger Oliva.

Identificação automática de atrito em atuadores
pneumáticos utilizando equipamento de baixo custo /
Roger Oliva Felix, 2018

71 f. : il.

Orientador: Marcos Silveira

Dissertação (Mestrado)- Universidade Estadual
Paulista. Faculdade de Engenharia, Bauru, 2018

1. Pneumática. 2. Dinâmica não linear.
3. Controlador de baixo custo. I. Universidade
Estadual Paulista. Faculdade de Engenharia. II.
Título.

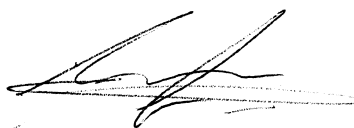
ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE Mestrado de Roger Oliva Felix, discente do Programa de Pós-graduação em Engenharia Mecânica, da Faculdade de Engenharia - Câmpus de Bauru.

Aos 27 dias do mês de julho do ano de 2018, às 14:00 horas, no(a) Anfiteatro da Pós-graduação / FEB, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Prof. Dr. MARCOS SILVEIRA - Orientador(a) do(a) Departamento de Engenharia Mecânica / Faculdade de Engenharia de Bauru, Prof. Dr. FABRICIO CESAR LOBATO DE ALMEIDA do(a) Departamento de Engenharia de Biosistemas / Faculdade de Ciências e Engenharia de Tupã, Prof. Dr. FRANSBER SANTADE do(a) Coordenadoria de Mecânica e Eletromecânica / Escola Técnica Estadual de São Paulo - ETEC Jacinto Ferreira de Sá, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE Mestrado de ROGER OLIVA FELIX, intitulada **IDENTIFICAÇÃO AUTOMÁTICA DE ATRITO EM ATUADORES PNEUMÁTICOS UTILIZANDO EQUIPAMENTO DE BAIXO CUSTO**. Após a exposição, o discente foi arguido oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

Prof. Dr. MARCOS SILVEIRA



Prof. Dr. FABRICIO CESAR LOBATO DE ALMEIDA



Prof. Dr. FRANSBER SANTADE



Dedicatória...

*Aos meus pais que me apresentaram à importância da família, o caminho da
honestidade, e a persistência na em busca do conhecimento.*

A todos os amigos que fizeram parte dessa pesquisa.

Agradecimentos

A realização desta dissertação de mestrado contou com importantes apoios e incentivos sem os quais não se teria tornado uma realidade e aos quais estarei eternamente grato.

Primeiramente agradeço a Deus pela minha existência, por me dar sabedoria, discernimento, saúde e muita força para superar todas as dificuldades, e por tudo que ele me proporcionou na minha vida pessoal e profissional, pois sem ele não conseguiria realizar meus sonhos.

Aos meus pais Mauro Felix e Elisabete Oliva Felix por acreditarem em mim, pelo apoio motivacional, incentivo desde sempre, amor, carinho, paciência e seus ensinamentos, e por não medirem esforços para que eu pudesse levar meus estudos adiante.

À minha futura esposa Laura Peres Zanetine, que sempre me apoiou e me ajudou nos momentos difíceis.

Ao meu orientador Professor Dr. Marcos Silveira pelo auxílio, dedicação, incentivo e pelo conhecimento transmitido durante o mestrado.

Aos docentes e membros do grupo de pesquisa NDE (Núcleo de Dinâmica em Engenharia) da UNESP-FEB, que contribuíram no desenvolvimento do trabalho.

Aos meus colegas de pesquisa que frequentam o Laboratório de Controle de Sistemas Mecânicos: Alana, Arthur, Matheus e Victor.

À empresa SMC Pneumáticos na qual trabalhei e obtive experiência profissional na área de automação pneumática, muito útil ao desenvolvimento do trabalho, agradecendo principalmente o funcionário Nure Soares pela ajuda técnica, incentivo, opiniões e críticas no desenvolvimento do trabalho.

À fundação CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) que auxiliou financeiramente no desenvolvimento da pesquisa.

Resumo

FELIX, R. O., IDENTIFICAÇÃO AUTOMÁTICA DE ATRITO EM ATUADORES PNEUMÁTICOS UTILIZANDO EQUIPAMENTO DE BAIXO CUSTO, Bauru: Faculdade de Engenharia, UNESP - Universidade Estadual Paulista, 2018, 71 p., Dissertação (Mestrado).

O presente trabalho aborda o estudo experimental das características não lineares de atrito interno em atuadores pneumáticos. Os atuadores pneumáticos contêm uma tecnologia limpa comparados com os atuadores hidráulicos, ótima relação entre potência e massa suportada, e baixo custo em relação a atuadores elétricos. Também possuem internamente características não lineares que influenciam no controle de um servoposicionador, incluindo principalmente a vedação do êmbolo com o perfil do atuador e a vedação da haste. Portanto, para obter a precisão de posicionamento é necessária uma caracterização do atrito interno. Os métodos utilizados para o desenvolvimento da pesquisa são: investigação do problema de atrito com o modelo *LuGre*; desenvolvimento do equipamento experimental com um controlador de baixo custo *Raspberry Pi*; e realização dos experimentos com a identificação automática de atrito em atuadores pneumáticos. Obtendo assim, a visualização gráfica da força de atrito em função da velocidade em regime permanente. A força de atrito em relação aos parâmetros do modelo é calculada em tempo real, possibilitando, dessa forma, a visualização do comportamento da força ao longo do deslocamento do êmbolo do atuador. Os resultados têm a finalidade de verificar as características de atrito do atuador pneumático através dos parâmetros identificados. Desse modo, é possível classificar o atuador pneumático mais adequado para aplicação em sistemas de posicionamento e monitorar o seu desgaste.

Palavras-chave: pneumática. dinâmica não linear. controlador de baixo custo.

Abstract

FELIX, R. O., AUTOMATIC IDENTIFICATION OF FRICTION IN PNEUMATIC ACTUATORS USING RIG OF LOW COST, Faculty of Engineering, UNESP - São Paulo State University, 2018, 71 p., Dissertation (Master degree).

This work presents an experimental procedure for determination of the nonlinear characteristics of internal friction of pneumatic actuators. Pneumatic actuators are a clean technology compared to hydraulic actuators, have good relationship between power and supported mass, and low cost compared to electric actuators. They also have nonlinear characteristics which influence the control of the servo positioner, including mainly the piston seal with the intern surface of the pneumatic actuators and rod seal, therefore to get the positioning accuracy it is necessary a good characterization of internal friction. The methods used for the development of the research are: investigation of the friction problem with the *LuGre* model; development of experimental equipment with a low cost controller *Raspberry Pi*; and achievement of the experiments with the automatic identification of friction in pneumatic actuators. Thus, it is obtained the graphical visualization of the friction force as a function of the velocity in steady state. The friction force in relation to model parameters is calculated in real time, enabling the visualization of behaviour of pneumatic force along of displacement piston of the actuator. The results have as purpose verify the friction characteristics of the pneumatic actuator using the identified parameters, thus, it is possible to classify the best pneumatic actuator for application in positioning systems and to monitor the your wear.

Keywords: pneumatic. nonlinear dynamics. low cost controller.

LISTA DE SÍMBOLOS

Símbolos	Descrição
Símbolos latinos	
A_a	área do êmbolo (Câmara a)
A_b	diferença entre a área do êmbolo e da haste (Câmara b)
b	número de <i>bits</i>
B	coeficiente de amortecimento
c_1	constante da equação da reta
c_2	constante da equação da reta
D	distância medida pelo sensor ultrassônico
D_p	diâmetro do êmbolo
D_r	diâmetro da haste
F_c	força de atrito de <i>Coulomb</i>
F_e	força de atrito externo
F_f	força de atrito
F_l	força da carga externa
F_n	força normal
F_p	força pneumática
F_s	força de atrito estático
P	ponto de avanço
P_a	pressão instantânea da câmara a
P_b	pressão instantânea da câmara b
P_p	pressão pneumática
P_r	pressão do regulador proporcional
P_s	pressão instantânea no momento em que o êmbolo se desloca

P_t	pressão medida do transdutor de pressão
r	raio do pneu
S	curso do atuador pneumático
t	tempo
v	velocidade do som
V_a	volume instantâneo da câmara a
V_b	volume instantâneo da câmara b
V_{da}	volume da câmara a com o atuador na zona morta
V_{db}	volume da câmara b com o atuador na zona morta
V_{out}	tensão de saída do conversor DA
V_r	tensão de entrada do regulador proporcional
V_{ref}	tensão de referência do conversor DA
V_t	tensão de saída do transdutor de pressão
x	coeficiente angular da equação da reta
y	deslocamento
\dot{y}_s	velocidade <i>Stribeck</i>
z	microdeformação
z_{ba}	deslocamento de força de quebra
z_{max}	máximo valor da microdeformação
z_{ss}	microdeformação em regime permanente

Símbolos gregos

α	característica da curva de tensão e deformação
ζ_0	relaxamento longitudinal
ζ_1	deformação longitudinal
θ	ângulo de inclinação da regressão linear
σ_0	coeficiente de rigidez
σ_1	coeficiente de amortecimento
σ_2	coeficiente de arraste
ω	velocidade angular

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Introdução ao problema	1
1.2	Revisão bibliográfica	3
1.3	Justificativa	17
1.4	Objetivos	17
1.5	Estrutura da dissertação	18
2	EQUIPAMENTO EXPERIMENTAL	19
2.1	Montagem do equipamento experimental	19
2.2	Dispositivos de baixo custo	23
2.3	Calibração dos instrumentos de medição	27
2.4	Funcionamento do equipamento experimental	28
3	CARACTERIZAÇÃO DE ATRITO	31
3.1	Procedimento para caracterização de atrito	31
3.2	Ensaio para extensão e compressão	33
3.3	Identificação dos parâmetros estáticos e dinâmicos	37
4	CONCLUSÕES FINAIS E TRABALHOS FUTUROS	39
4.1	Conclusões finais	39
4.2	Trabalhos futuros	40
5	TRABALHOS PUBLICADOS	41
	REFERÊNCIAS BIBLIOGRÁFICAS	42

APÊNDICES	47
A Circuitos eletrônicos e diagrama pneumático	47
A.1 Conexões elétricas de entrada e saída	48
A.2 Circuitos da placa intermediária - 1	49
A.3 Circuitos da placa intermediária - 2	50
A.4 Diagrama pneumático	51
A.5 Lista de equipamentos	52
B Programação do <i>Raspberry Pi</i> para obter os resultados experimentais	53
B.1 Programação em linguagem C	53

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo apresenta a introdução geral sobre caracterização de atrito em atuadores pneumáticos, utilizando um equipamento experimental com dispositivos de baixo custo para aquisição de dados. Também apresenta a revisão bibliográfica sobre a temática da pesquisa, com a finalidade de demonstrar alguns estudos sobre atuadores pneumáticos e seus componentes; modelos de atrito estático e dinâmico; influência e identificação de atrito; e dispositivos de baixo custo. A justificativa e os objetivos deste trabalho também são abordados neste capítulo.

1.1 INTRODUÇÃO AO PROBLEMA

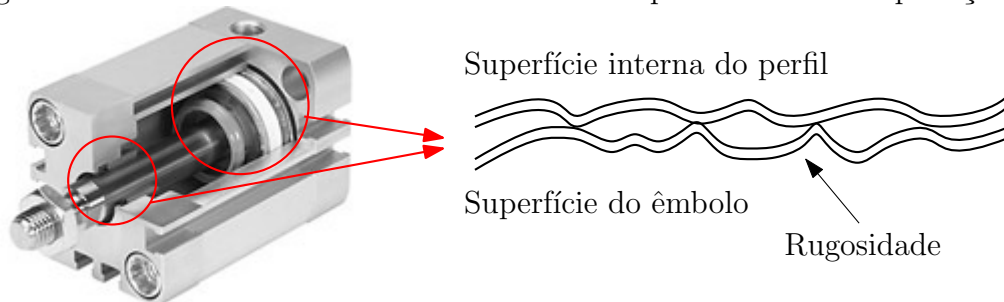
O estudo de atrito em atuadores pneumáticos está relacionado a uma dificuldade de controle em sistemas de posicionamento, devido à influência do atrito interno do atuador pneumático, que é representada por um modelo matemático não linear. A pesquisa objetiva desenvolver a análise do comportamento da força de atrito em função da velocidade do êmbolo, em regime permanente, além de identificar os parâmetros estáticos e dinâmicos através da caracterização de atrito, que é realizada por um processo automático utilizando um equipamento experimental.

O termo atrito surgiu no século XV, por *Leonardo da Vinci*, e foi identificado através do deslizamento de blocos retangulares sobre superfícies planas. Posteriormente, *Charles de Coulomb* identificou, por meio do movimento de um corpo, que a força de atrito independe da velocidade. Existem diversos modelos de atrito, dentre eles o está-

tico e o dinâmico. Os principais modelos de atrito estático são: viscoso e *Coulomb*; e os principais modelos de atrito dinâmico são: *Dahl* e *LuGre*, considerados os de maior importância para sistemas com precisão. Portanto, cada modelo possui características distintas que podem contribuir ou prejudicar o sistema de posicionamento em um processo industrial.

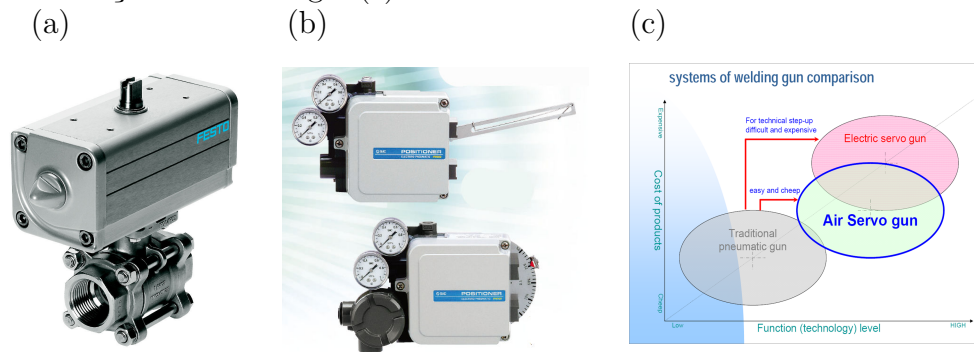
O atrito interno de um atuador pneumático ocorre em função do deslocamento do êmbolo, tanto em relação ao contato da vedação do êmbolo com a superfície interna do perfil, quanto ao da haste com a vedação da superfície interna do cabeçote. O atuador é projetado para operar em baixas pressões entre as câmaras internas, e possui uma vedação entre o êmbolo e a superfície interna do perfil, assim, não há queda de pressão ou vazamento de ar comprimido. Para superar as limitações da força de atrito, Wit *et al.* (1995b) propõem o modelo de atrito dinâmico, conhecido como *LuGre* (*Lund-Grenoble*), baseado na flexão média das reentrâncias existentes entre os materiais de contato e muito utilizado em sistemas não lineares para compensação de atrito. Através da identificação do atrito interno do atuador pneumático ou do atrito existente na carga acoplada pode-se desenvolver o controle de compensação adequado para o sistema de posicionamento e apresentar a solução para o problema de atrito. Sendo assim, com os resultados de cada experimento em regime permanente torna-se possível caracterizar a força de atrito em função da velocidade para extensão e compressão do atuador pneumático, bem como, identificar os parâmetros do sistema. A Figura 1.1 mostra os pontos de contato interno do atuador pneumático de dupla ação, a Figura 1.2-a e a Figura 1.2-b mostram alguns modelos dos servoposicionadores utilizados no meio industrial e a Figura 1.2-c mostra graficamente o custo dos equipamentos posicionadores em função da tecnologia.

Figura 1.1: Pontos de contato interno do atuador pneumático de dupla ação



Fonte: Adaptado de Sun *et al.* (2016)

Figura 1.2: Modelos de servoposicionadores (a) e (b), e custo de equipamentos posicionadores em função da tecnologia (c)



Fonte: <https://www.festo.com> (a); <http://www.smcbr.com.br> (b) e (c)

1.2 REVISÃO BIBLIOGRÁFICA

Os atuadores pneumáticos aplicados a sistemas de posicionamento não correspondem, de forma satisfatória, a algumas aplicações industriais, devido ao atrito interno existente no atuador. Diversos estudos, relacionados ao atrito em atuadores, já foram realizados com a finalidade de solucionar esse problema e melhorar a performance de controle em sistemas de precisão. Este trabalho está baseado nos artigos de Andrighetto, Valdiero e Carlotto (2006) e Lafmejani, Masouleh e Kalhor (2016), e tem como finalidade aprofundar o estudo da identificação do comportamento de atrito em atuadores pneumáticos, realizando ensaios experimentais para análise do comportamento do atrito de forma automática e utilizando dispositivos de baixo custo.

A presente pesquisa não visa realizar uma análise complexa de todos os modelos de atrito existentes, e sim, abordar de forma geral alguns dos modelos de atrito estático e dinâmico, dando ênfase ao modelo de atrito *LuGre*.

A concepção do modelo de atrito *LuGre*, utilizada na pesquisa, é apresentada em alguns trabalhos de Dahl (1968) e de Wit *et al.* (1995b), os quais contribuíram significativamente para a modelagem da força de atrito por meio de um modelo dinâmico, com fundamentações experimentais do fenômeno de atrito e esclarecendo detalhes do problema de atrito.

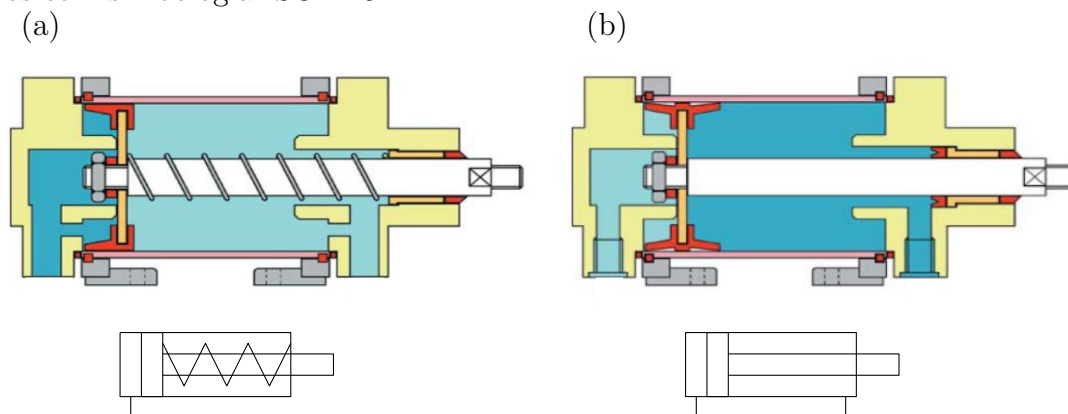
1.2.1 ATUADORES PNEUMÁTICOS

O atuador pneumático é o componente responsável pela transformação da energia pneumática em energia cinética gerada pelo ar pressurizado e em expansão (FIALHO, 2004). Conforme Schoenmeier *et al.* (2014) o atuador corresponde a um sistema que possibilita posicionar uma carga em um certo ponto do curso ou realizar uma trajetória variável em função do tempo. Segundo Alciatore e Hestand (2014) após o compressor liberar o ar comprimido para o sistema pneumático, os excessos de umidade e calor são removidos do ar com o auxílio de uma unidade de tratamento, de maneira oposta das bombas hidráulicas, que transmitem deslocamento positivo de fluido (óleo) a alta pressão. Portanto, os sistemas pneumáticos comparados aos sistemas hidráulicos possuem algumas características positivas, tais como: fonte de energia limpa e sistemas abertos, que estão sempre processando ar novo, eliminando a necessidade de uma rede de linhas de retorno.

Os atuadores possuem movimentos retilíneos, giratórios e angulares, contendo baixa rigidez devido à compressibilidade do fluido (ar comprimido) porém, essa característica corresponde a um sistema não preciso quanto ao controle de posicionamento. O movimento dos atuadores retilíneos possui duas condições: extensão e compressão, que implica em um controle simples e de baixo custo. Para o funcionamento adequado do atuador são necessários: compressor, com a finalidade de fornecer o ar comprimido; instalação de unidade de preparação (filtro e regulador de pressão); e válvulas direcionais (ROMANO; DUTRA, 2002).

Existem dois tipos de atuadores pneumáticos com movimentos retilíneos: atuadores de ação simples e atuadores de ação dupla, conforme mostrados na Figura 1.3. Os atuadores de ação simples são acionados apenas em um sentido de movimento, por meio do ar comprimido. No entanto, o movimento oposto do atuador pode ser realizado por mola, que se movimenta através de uma força elástica e externa, ou pela ação da gravidade, caso não possua mola. Já os atuadores de ação dupla têm seus movimentos de extensão e compressão acionados pneumaticamente. Os atuadores normalizados possuem um limite de curso devido às possibilidades de deformação da haste por flexão e flambagem (MOREIRA, 2012).

Figura 1.3: Atuador pneumático de ação simples com retorno mola (a) e dupla ação (b), ambos com simbologia ISO1219-1



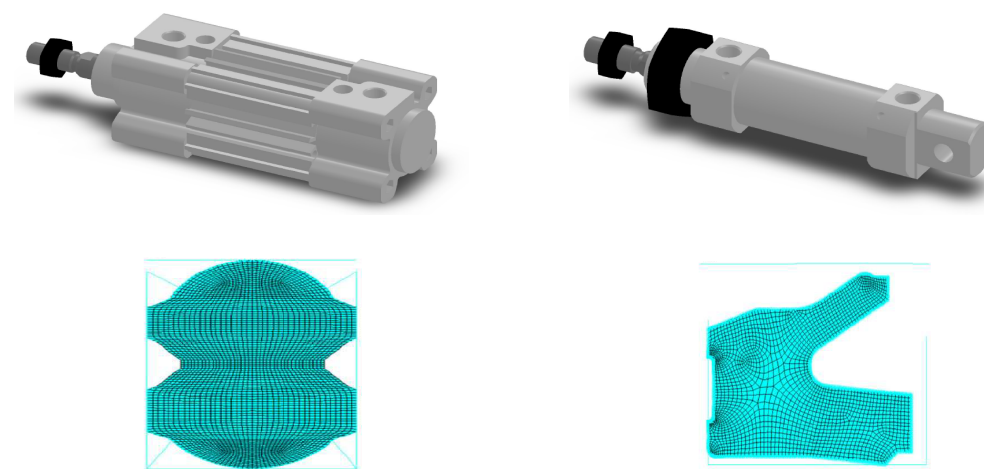
Fonte: Adaptado de Moreira (2012)

1.2.2 TIPOS DE VEDACÕES

O atuador pneumático possui a vedação do êmbolo e do cabeçote, que são os principais componentes para o funcionamento correto e preciso. O material mais utilizado nas vedações é o NBR (*Nitrile Butadiene Rubber*) porém, mesmo com as propriedades elásticas possui atrito, ocasionado devido aos contatos do êmbolo e da haste com a superfície interna do atuador. A maioria dos atuadores pneumáticos possui modelos de vedações idênticos aos dos atuadores normalizados ISO15552 e ISO6432. A Figura 1.4 mostra os atuadores normalizados e suas correspondentes vedações.

Belforte *et al.* (2009) analisaram dois tipos de vedações elastoméricas de atuadores pneumáticos sem carga acoplada, realizando uma análise numérica com utilização de elementos finitos e uma análise experimental para validar a performance das vedações. Os dois tipos de vedações analisados correspondem as vedações utilizadas em atuadores normalizados. A vedação do ISO15552 obteve a menor pressão de contato comparada à vedação do ISO6432. Os experimentos foram realizados a uma pressão de 6 bar nas condições de uso a seco e lubrificado, e através desses experimentos concluíram que as duas condições possuem coeficiente de atrito menor e o ISO15552 corresponde ao atuador ideal para sistemas de posicionamento. Sob outra perspectiva Belforte, Manuello e Mazza (2006) apresentaram a otimização da vedação do ISO15552 para diminuir a força de atrito interna do atuador pneumático, utilizando um modelo numérico de elementos finitos que mostra a não linearidade do material. Desse modo, desenvolveram uma nova geometria da vedação que é validada experimentalmente.

Figura 1.4: Vedação do ISO15552 (a) e vedação do ISO6432 (b)



Fonte: Adaptado de Belforte *et al.* (2009)

1.2.3 MODELOS DE ATRITO ESTÁTICO

Os modelos de atrito estático apresentam diferentes fenômenos de atrito que modificam a característica da força de atrito, assim, o atrito estático pode ser considerado uma função descontínua entre a força de atrito e a velocidade relativa das superfícies de contato (WONDIMU, 2006). A Figura 1.5 mostra a curva da força de atrito em função da velocidade de modo histerese, apresentando os seguintes parâmetros: F_f é a força de atrito e \dot{y} é a velocidade. A força de atrito é maior para aceleração crescente e menor para aceleração decrescente. Esse fenômeno de histerese ocorre principalmente devido às características das superfícies de contato e a viscosidade do lubrificante. (OLSSON *et al.*, 1998).

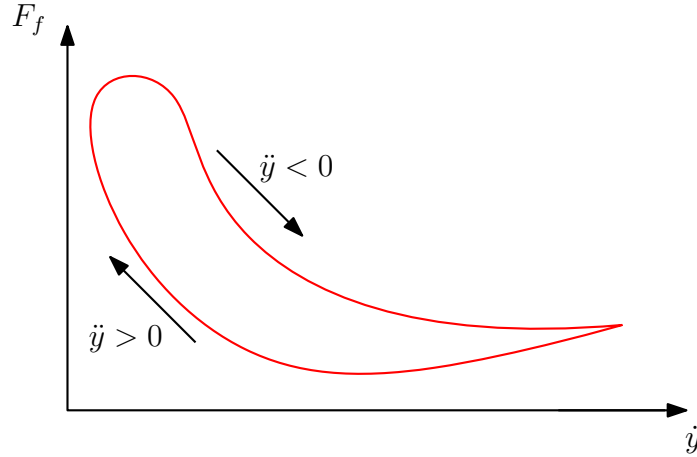
O modelo de *Coulomb* é o mais difundido dentre os modelos de atrito estático e sua curva característica é mostrada na Figura 1.6-a. Esse modelo não possui a força de atrito para velocidade zero e independe da velocidade (ARMSTRONG; WIT, 1996). A força de atrito do modelo de *Coulomb* é dada por:

$$F_f = F_c \operatorname{sgn}(\dot{y}) \quad (1.1)$$

na qual F_c é a força de *Coulomb* e $\operatorname{sgn}()$ é a função sinal.

O modelo de atrito estático tem uma força de atrito que ocorre quando um corpo

Figura 1.5: Histerese da força de atrito em função da velocidade



Fonte: Adaptado de Olsson *et al.* (1998)

se encontra em repouso, conforme mostrado na Figura 1.6-b. A força de atrito estático ocorre no momento de pré-deslizamento entre as duas superfícies de contato e depende da força externa para definir se é positiva ou negativa. A força de atrito também é caracterizada através de uma função impulso para auxiliar nas implementações computacionais (VALDIERO, 2005). A intensidade da força de atrito estático é sempre maior que a força de atrito de *Coulomb* (SÁNCHEZ-MAZUCA; CAMPA, 2013). O modelo de atrito estático é dado por:

$$F_f = F_s \operatorname{sgn}(F_e) \quad (1.2)$$

no qual F_s é a força de atrito estático e F_e é a força de atrito externo.

O modelo de atrito viscoso é linearmente proporcional a velocidade, conforme mostrado na Figura 1.6-c. O atrito viscoso corresponde a uma situação de boa lubrificação e tem a função de realizar o amortecimento (VALDIERO, 2005). O modelo de atrito viscoso é dado por:

$$F_f = B \dot{y} \quad (1.3)$$

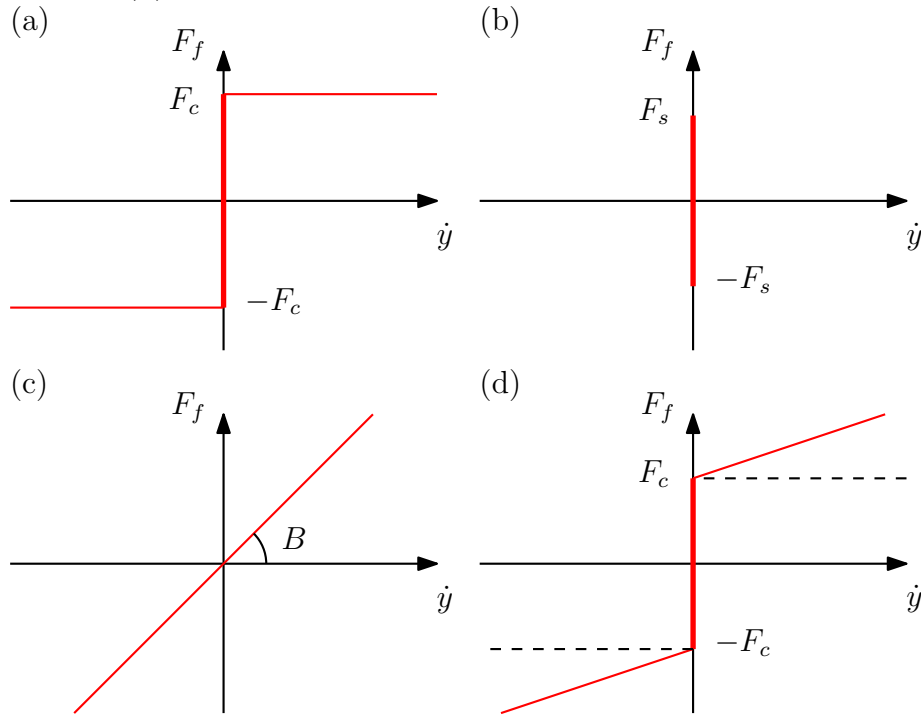
no qual B é o coeficiente de amortecimento ou coeficiente de atrito viscoso.

O modelo de atrito de *Coulomb* combinado com o modelo viscoso, mostrado na Figura 1.6-d, é muito utilizado em problemas dinâmicos. Esse modelo possui facilidade para ser tratado matematicamente, com isso é possível realizar uma aproximação do

sistema real (ANDERSSON; SÖDERBERG; BJÖRKLUND, 2007). Nesse caso, o atrito de *Coulomb* combinado ao atrito viscoso é dado por:

$$F_f = F_c \operatorname{sgn}(\dot{y}) + B\dot{y} \quad (1.4)$$

Figura 1.6: Modelos de atrito: *Coulomb* (a), estático (b), viscoso (c) e *Coulomb* combinado com viscoso (d)



Fonte: Adaptado de Olsson *et al.* (1998)

1.2.4 MODELOS DE ATRITO DINÂMICO

Entre os modelos mais utilizados para a compensação de atrito estão os modelos estático: *Coulomb* e viscoso. No entanto, esses modelos submetidos a aplicações com alta precisão e baixa velocidade apresentam um desempenho insatisfatório (WIT *et al.*, 1995a). Dessa forma, com a alta demanda por sistemas mais precisos, estudos relacionados aos aspectos dinâmicos de atrito começaram a se intensificar. Os modelos de atrito dinâmico descritos nesta subseção são: *Dahl* e *LuGre*.

O modelo descrito por Dahl (1968) tem a finalidade de caracterizar o atrito em sistemas de posicionamento. Esse modelo iniciou-se a partir de várias experiências de atrito em sistemas de deslocamento, incluindo as esferas de aço dos rolamentos. Essas experi-

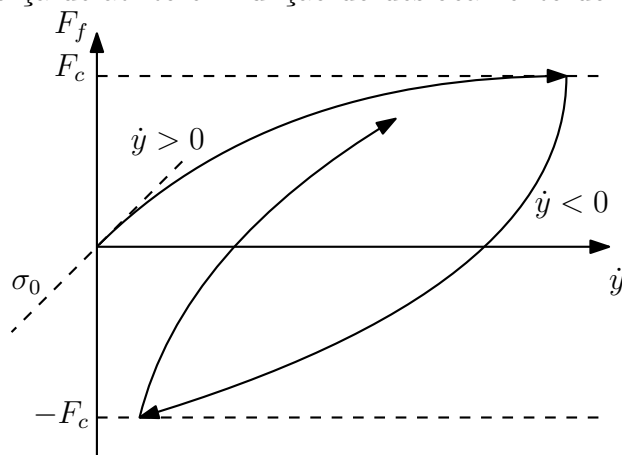
ências mostraram que o atrito do rolamento tem comportamento semelhante ao atrito sólido e que possui contatos metálicos entre as superfícies de contato. Desse modo, foi desenvolvido o modelo de *Dahl*, comparativamente simples, que é usado extensivamente para simular sistemas com atrito de rolamento de esferas (OLSSON *et al.*, 1998). A modelagem da curva de tensão e deformação é realizada por uma equação diferencial, dada por:

$$\dot{F}_f(y) = \sigma_0 \left(1 - \frac{F_f}{F_c} \text{sgn}(\dot{y})\right)^\alpha \quad (1.5)$$

na qual σ_0 é o coeficiente de rigidez das microdeformações, $\text{sgn}()$ é a função sinal e α é o parâmetro que determina a característica da curva de tensão e deformação. Wit *et al.* (2003) demonstraram que quanto maior o valor do expoente, conseqüentemente, mais acentuada é a curva do modelo, lembrando que a força de atrito jamais é maior que a força de *Coulomb*, conforme mostra a Figura 1.7. Assim, é necessário obter o modelo $\dot{F}_f(y)$ em domínio do tempo, resultando em:

$$\dot{F}_f(t) = \sigma_0 \left(1 - \frac{F_f}{F_c} \text{sgn}(\dot{y})\right)^\alpha \dot{y} \quad (1.6)$$

Figura 1.7: Força de atrito em função do deslocamento do modelo de *Dahl*



Fonte: Adaptado de Olsson *et al.* (1998)

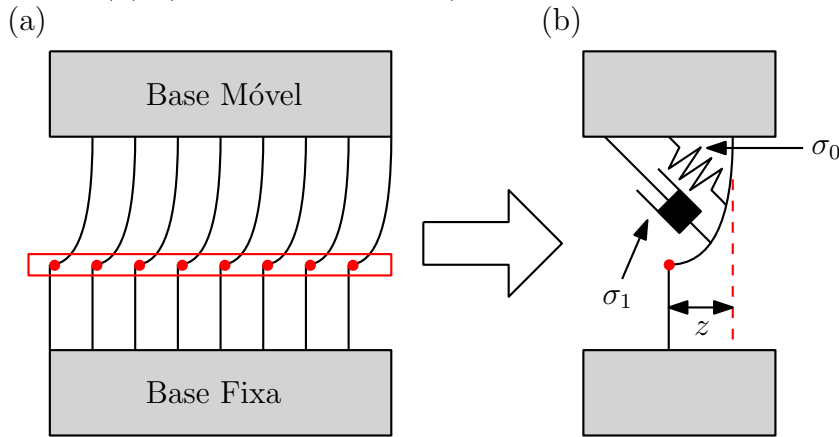
Olsson *et al.* (1998) mostraram que o modelo de *Dahl* origina-se a partir do atrito de *Coulomb*, dessa forma, não possui o efeito *Stribeck* e nem o atrito estático. Portanto, conhecendo que $F_f = \sigma_0 z$ e assumindo que $\alpha = 1$, a dinâmica da microdeformação z é

dada por:

$$\dot{z} = \dot{y} - \frac{\sigma_0 |\dot{y}|}{F_c} z \quad (1.7)$$

O modelo de atrito utilizado na pesquisa é o modelo *LuGre*, que possui uma força de atrito composta por três partes: a primeira refere-se à média das deformações $\sigma_0 z$, a segunda proporcional à taxa de variação das deformações $\sigma_1 \dot{z}$, ambas mostradas na Figura 1.8, e a terceira ao atrito de arraste causado pela resistência ao movimento do corpo através de um fluido. Esse modelo é uma continuação do modelo de *Dahl*, que possui o efeito *Stribeck* e o atrito estático, ambos inclusos na modelagem de atrito. (WIT *et al.*, 1995a).

Figura 1.8: Área de contato entre duas superfícies (a) e média das microdeformações representada por z (b); (• ponto de contato)



Fonte: Adaptado de Sun *et al.* (2016)

Guenther *et al.* (2006) apresentaram em suas pesquisas a força de atrito do modelo *LuGre*, que é causada pela microdeformação z e descrita por:

$$F_f = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{y}^2 \text{sgn}(\dot{y}) \quad (1.8)$$

na qual σ_1 é o coeficiente de amortecimento e σ_2 é o coeficiente de arraste. A dinâmica das microdeformações é caracterizada pela variável não mensurável z , modelada pela Equação (1.9) e incorporada ao modelo *LuGre*, dada por:

$$\dot{z} = \dot{y} - a(z, \dot{y}) \frac{\sigma_0}{g_{ss}(\dot{y})} |\dot{y}| z \quad (1.9)$$

na qual $a()$ é a função para obter o atrito estático em velocidades baixas, $|\dot{y}|$ é o valor absoluto da velocidade, e g_{ss} é a função positiva das características de atrito em regime permanente, que resulta em:

$$g_{ss}(\dot{y}) = F_c + (F_s - F_c)e^{-\left(\frac{\dot{y}}{\dot{y}_s}\right)} \quad (1.10)$$

De acordo com Dupont, Armstrong e Hayward (2000), a função $a(z, \dot{y})$ é incorporada ao modelo *LuGre* e empregada para obter a representação do atrito estático em velocidades menores. Assim, a função é definida por:

$$a(z, \dot{y}) = \begin{cases} 0 & , \text{se } |z| \leq z_{ba} \\ 0 < \frac{1}{2} \sin \left(\pi \frac{z - \left(\frac{z_{max}(\dot{y}) + z_{ba}}{2} \right)}{z_{max}(\dot{y}) - z_{ba}} \right) + \frac{1}{2} < 1 & , \text{se } z_{ba} < |z| \leq z_{max}(\dot{y}) \\ 1 & , \text{se } |z| \geq z_{max}(\dot{y}) \\ 0 & , \text{se } \text{sgn}(\dot{y}) \neq \text{sgn}(z) \end{cases} \quad (1.11)$$

$$0 < z_{ba} < z_{max}(\dot{y}) = \frac{g_{ss}(\dot{y})}{\sigma_0} \text{ para } \forall \dot{y} \in \mathfrak{R}; \quad (1.12)$$

na qual \dot{y}_s é a velocidade de *Stribeck* e z_{ba} deslocamento de força de quebra, pois para $z \leq z_{ba}$ qualquer movimento na interface de atrito é composto apenas de comportamentos elásticos e z_{max} é o valor máximo da microdeformação que está em função da velocidade. Em regime permanente \dot{y} é constante, $a(z, \dot{y}) = 1$ e $\dot{z} = 0$. Substituindo o valor dessas condições na Equação (1.9) é possível obter a força de atrito para movimentos em velocidades constantes (JOHANASTROM; WIT, 2008). Carneiro e Almeida (2015) apresentaram a aproximação do desvio z , que é dada por:

$$z_{ss} = \frac{\dot{y} g_{ss}(\dot{y})}{\dot{y} \sigma_0} = \text{sgn}(\dot{y}) \frac{F_c + (F_s - F_c) e^{-\left(\frac{\dot{y}}{\dot{y}_s}\right)^2}}{\sigma_0} \quad (1.13)$$

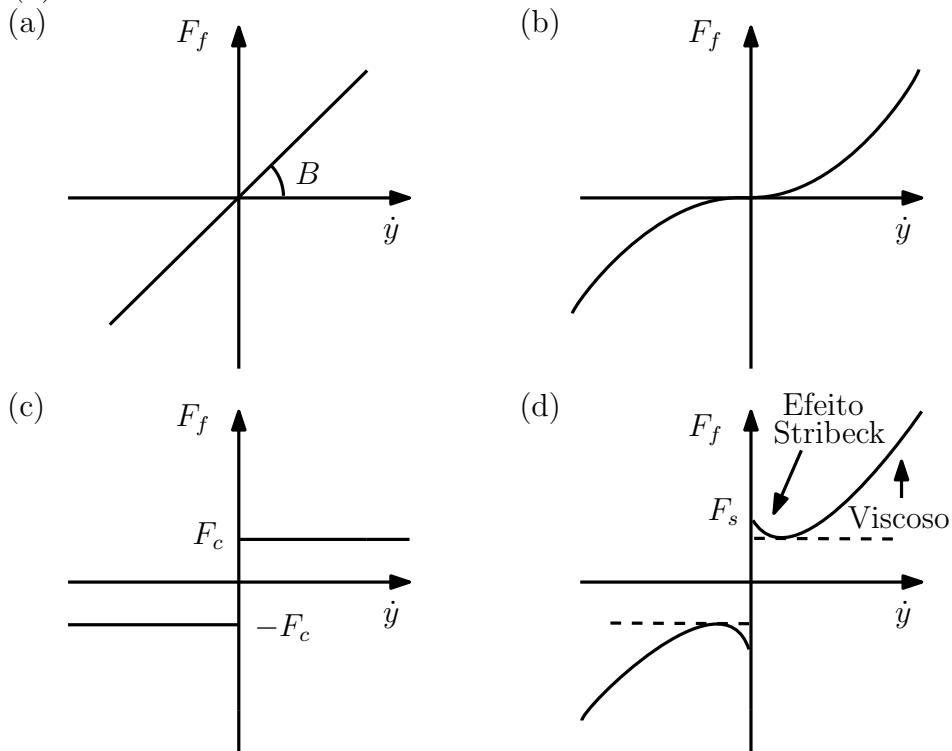
na qual z_{ss} é a microdeformação em regime permanente. Assim, é possível obter a equação representativa do modelo e identificar os parâmetros estáticos do atrito (F_c , F_s ,

\dot{y}_s e σ_2), dada por:

$$F_f = \text{sgn}(\dot{y})(F_c + (F_s - F_c))e^{-\left(\frac{\dot{y}}{\dot{y}_s}\right)^2} + \sigma_2\dot{y}^2\text{sgn}(\dot{y}) \quad (1.14)$$

Fung, Han e Chang (2008) ressaltaram as características de atrito para cada fenômeno de atrito do modelo *LuGre*, conforme é mostrado na Figura 1.9.

Figura 1.9: Curva característica da força de atrito viscoso (a), arraste (b), *Coulomb* (c) e *LuGre* (d)



Fonte: Adaptado de Ha *et al.* (2006)

Além das forças de atrito dos modelos de atrito estático (Viscoso e *Coulomb*), está também presente no modelo *LuGre*, a força de atrito de arraste, dada por:

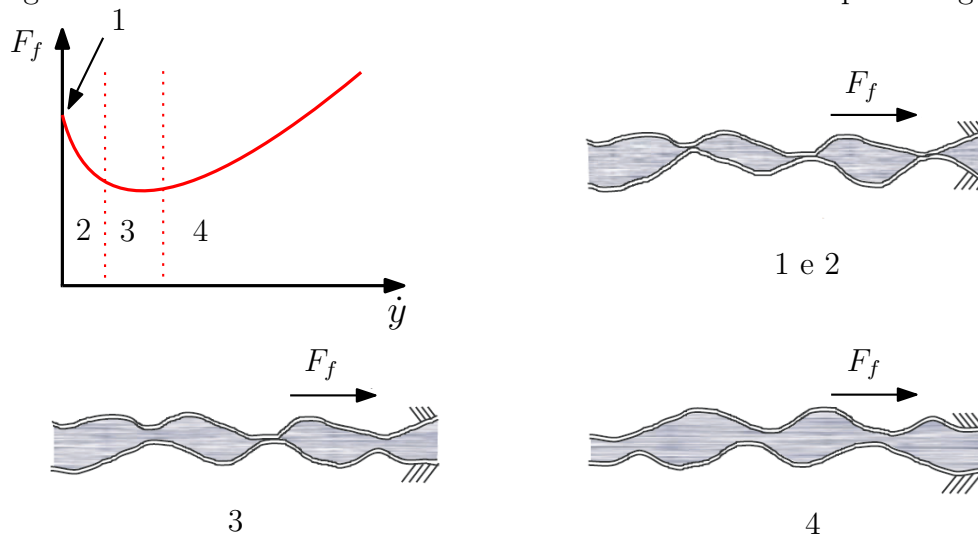
$$F_f = \sigma_0\dot{y}^2\text{sgn}(\dot{y}) \quad (1.15)$$

assim, é possível observar que a força de atrito do modelo *LuGre* em velocidades maiores se aproxima da força de atrito viscoso.

A curva do modelo *LuGre* está dividida em quatro regimes, apresentando o comportamento do atrito em regime permanente, conforme mostrado na Figura 1.10. Do primeiro ao terceiro regime é mostrado ao efeito *Stribeck*. O primeiro regime indica o atrito está-

tico, que gera uma deformação elástica na vedação, resultando em $F_f \leq F_s$. O segundo regime corresponde à condição de lubrificação, que pode interferir no comportamento da força de atrito, devido à variação da geometria do êmbolo, às vedações do atuador e ao tipo de lubrificação (óleo ou graxa). O terceiro regime aponta a condição parcial de lubrificação, que proporciona o deslizamento na área de contato entre o êmbolo e a superfície interna do atuador. O quarto regime corresponde ao atrito de arraste ou viscoso e mostra a condição de lubrificação completa, que ocorre no momento em que a lubrificação é suficiente para eliminar o máximo de contato da vedação do êmbolo com a superfície interna do atuador. (HVOLDAL; OLESEN, 2011).

Figura 1.10: Curva de atrito com efeito *Stribeck* dividido em quatro regimes



Fonte: Adaptado de Márton e Lantos (2006)

1.2.5 INFLUÊNCIA E IDENTIFICAÇÃO DE ATRITO EM ATUADORES PNEUMÁTICOS E HIDRÁULICOS

O termo atrito em atuadores pneumáticos ou hidráulicos é idealizado somente em sistemas de translação, porém existem também em sistemas de rotação. Para determinar o modelo matemático de atrito inerente a uma garra pneumática de forma experimental é preciso desenvolver o controle de precisão em malha fechada, solucionando o problema de controle de precisão no processo robótico. O modelo matemático proposto é formado por blocos de *feedback* e *feedforward*, o que possibilita identificar uma redução significativa no comportamento do sistema de posicionamento, otimizando a velocidade e a

precisão do deslocamento (JOHNSON; LORENZ, 1992). Num simulador de movimento de vôo são utilizados atuadores de rotação hidráulico, no entanto ocorre uma falha na região de velocidade baixa, que é afetada pela força de atrito não linear, apesar de existir compensação de atrito do diagrama completo do simulador. Essa aplicação, referente ao atuador rotativo hidráulico, não possui um modelo de atrito caracterizado. O modelo matemático do sistema em completo é desenvolvido na prática, aplicando uma lei adaptativa para ensinar o processo a se modificar lentamente e compensar o atrito existente utilizando um controlador robusto adaptativo, que assume o sistema em regime permanente. Assim, apresentando os efeitos de atrito composto na região de velocidade baixa e identificando o erro da modelagem (YAO; JIAO; HAN, 2013).

Os atuadores hidráulicos são aplicados, por exemplo, em estruturas civis para isolar as vibrações transmitidas pelo solo. É possível identificar os parâmetros do atuador hidráulico utilizado numa estrutura civil por meio do modelo dinâmico e com o intuito de linearizar o modelo não linear do comportamento do atuador hidráulico, através de uma análise experimental. O método proposto é baseado em algoritmos genéricos, com a finalidade de realizar interações comparadas ao modelo base identificado, assim os algoritmos genéricos têm a vantagem de obter alta eficiência em interações para encontrar a solução ótima (QIAN *et al.*, 2014). Outra aplicação de atuadores hidráulicos, que também possuem fatores que dificultam o seu funcionamento, seria a de um simulador de turbina eólica que, através do movimento da turbina, é possível determinar um comportamento diferente em função da resposta em frequência, podendo corresponder a um modelo não linear ou linear. Dessa forma, causando um mau funcionamento ou uma ótima condição de simulação. Com a análise do comportamento do atuador hidráulico, o problema do mau funcionamento do simulador pode ser ajustado pelos seguintes fatores: porcentagem de água e óleo internamente no atuador, atrito externo e instabilidade do sensor de falha (RAKOTO; SCHORSCH; KINNAERT, 2015).

Para identificação de atrito em atuadores hidráulicos de forma experimental, é aplicado um algoritmo genérico com o intuito de identificar a aproximação do erro do modelo matemático de atrito. Apresentando, assim, um modelo de atrito mais simples e rápido em relação a simulação computacional, que pode ser utilizado em sistemas hidráulicos industriais (MÁRTON; FODOR; SEPEHRI, 2011). O mesmo procedimento pode ser

aplicado para identificar o atrito em atuadores pneumáticos, caracterizando os parâmetros de atrito aplicando algoritmo genérico. Pelo fato de ser impossível de se obter a medição direta dos parâmetros de atrito não lineares, o método de identificação proposto apresenta a estimação do atrito dinâmico ao longo do curso do atuador pneumático, aplicando um algoritmo genérico para alcançar a solução ótima e, após esse algoritmo ser testado, é realizada a simulação computacional para validar o método proposto (WANG *et al.*, 2004). Outra forma de identificar as características não lineares do atuador hidráulico seria realizar uma comparação entre dois modelos propostos e avaliar por meio dos resultados experimentais. O primeiro modelo *Hammerstein* é utilizado para a dinâmica de atrito, que a partir da modelagem matemática tem a possibilidade de mapear o sistema não linear para desenvolver o segundo modelo paralelo mais aproximado. Como os dados dos parâmetros físicos são constantes, o modelo paralelo proporciona uma constante relação não linear entre a força de atrito no retentor e a velocidade do deslocamento do êmbolo (KWAK; YAGLE; LEVITT, 1998). A linearização de sistemas não lineares é comum para sistemas mecânicos, facilitando a resolução do problema com uma simplificação. A linearização do modelo matemático não linear do sistema de posicionamento em atuadores hidráulicos pode ser realizada através de estimativas de estado, baseadas em coeficientes de estados dependentes, assim linearizando o sistema não linear com a equação algébrica de *Riccati* juntamente com o filtro de *Kalman*. (STRANO; TERZO, 2016).

Um dos métodos de solução para identificação de atrito em atuadores pneumáticos e hidráulicos é a utilização do modelo de *Karnopp*, que considera a descontinuidade de estática para o atrito de *Coulomb*, deixando o sistema de posicionamento com menor velocidade. Para a identificação do modelo de atrito são utilizados três passos: a regressão linear para estimar a massa; o gráfico para encontrar a velocidade limite na região *stick-slip*; e o método de otimização não linear para identificar o parâmetro *stick* (RAVANBOD-SHIRAZI; BESANÇON-VODA, 2003).

Uma forma de melhorar a precisão de posicionamento de um atuador pneumático sem o modelo de atrito caracterizado seria obter o modelo matemático de uma servoválvula próximo ao linear e encontrar a solução ótima, mostrando uma melhoria para o sistema de posicionamento pneumático. Essa melhoria ocorre devido ao controle reali-

zado na mudança de estado do carretel da válvula. Portanto, aplicando o controle teórico moderno no sistema de posicionamento em regime permanente, o controle PID (Proporcional Integrativo e Derivativo) alcança a melhor performance do critério ótimo, obtendo o melhor desempenho do sistema pneumático (SHIH; TSENG, 1995). Do mesmo modo, Hamiti, Voda-Besancon e Roux-Buisson (1996) apresentam uma aplicação de um controle PI (proporcional integrativo) para ajustar a precisão do atuador pneumático, devido à presença do atrito *stick-slip* na carga acoplada na haste do atuador, que exige uma pressão maior que a pressão de trabalho. O método proposto tem a finalidade de obter a simplicidade de cálculos, facilidade de interpretação, precisão suficiente e fácil implementação do sistema de controle.

Para um modelo de controle de um servomotor hidráulico que possui incertezas (atrito), ocorre a resolução do problema robusto de um sistema com controlador clássico PI (Proporcional Integrativo), com o intuito de apresentar o modelo linear do sistema de posicionamento, realizando a linearização desse sistema. (PUGLISI *et al.*, 2015).

1.2.6 DISPOSITIVOS DE BAIXO CUSTO PARA ANÁLISE EXPERIMENTAL

Os dispositivos de baixo custo estão aplicados em diversas áreas, como agricultura, industrial e acadêmica. Entre as áreas citadas esses dispositivos são mais utilizados na área acadêmica, para o desenvolvimento de protótipos. Os dispositivos de baixo custo estão sendo muito utilizados devido à vinda da internet das coisas IOT(Internet of Things) e indústria 4.0.

Todos os dispositivos de baixo custo possuem alguns protocolos de comunicações em comum, como USB, RS-232 e RS-485. Assim, possibilitando a comunicação entre esses dispositivos (KRAUSS, 2016). Além de se comunicarem entre eles, também é possível realizar acesso remoto, que é muito importante quando tem a necessidade de controlar ou monitorar as variáveis a longa distância. Esse monitoramento e controle das variáveis são realizados por meio da interface gráfica aplicada a um equipamento de imagem, como interface homem máquina (IHM) ou até mesmo pelo celular com funções de *touchscreen* (PINEÑO, 2014).

Existem diversos trabalhos na área de projetos mecânicos que utilizam os dispositivos

de baixo custo. Gonçalves e Silveira (2017) apresentaram uma solução experimental de monitoramento a integridade de estruturas mecânicas utilizando dispositivos de baixo custo, como o controlador arduino e o sensor piezoelétrico. Assim, demonstrando a possibilidade de utilizar dispositivos de baixo custo para essa aplicação, obtendo resultados efetivos. Do mesmo modo, Elfasakhany *et al.* (2011) apresentaram o desenvolvimento de um robô competitivo utilizando dispositivos de baixo custo, porém com objetivo de controlar seus movimentos e monitorar os ângulos de cada braço do robô. Assim, obtendo resultados de precisão ótimos.

1.3 JUSTIFICATIVA

A caracterização de atrito tem sua importância em todos os sistemas de precisão, auxiliando assim, no desenvolvimento do controle desses sistemas e na identificação do atuador mais adequado para determinada aplicação. Essa caracterização não se aplica apenas aos atuadores pneumáticos, podendo ser utilizada em cargas acopladas ao eixo do atuador, válvulas industriais com abertura de precisão, entre outros. Este trabalho poderá contribuir, principalmente, na elaboração de projetos mecânicos com sistema de precisão e beneficiar os departamentos de pesquisa e desenvolvimento (P&D) dos fabricantes de atuadores pneumáticos.

1.4 OBJETIVOS

O objetivo principal deste trabalho é realizar a caracterização de atrito em atuador pneumático utilizando equipamento de baixo custo, considerando as características não lineares ocasionadas devido aos tipos de lubrificação, condições físicas das vedações, dentre outros fatores. O objetivo principal pode ser dividido em três partes:

- (a) Projeto e montagem do equipamento experimental, utilizando dispositivos eletrônicos de baixo custo;
- (b) Realização de ensaios automáticos para caracterização de atrito;
- (c) Identificação dos parâmetros estáticos e dinâmicos da caracterização de atrito.

1.5 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está estruturada em capítulos sequenciais de modo que: o capítulo 2 mostra os dispositivos utilizados no equipamento experimental e a metodologia para encontrar a força pneumática e estática do atuador pneumático. O capítulo 3 apresenta os resultados parciais para expansão e compressão do atuador, com o objetivo de caracterizar o atrito inerente ao atuador e encontrar os parâmetros estáticos e dinâmicos. E, por fim, o capítulo 4 mostra as conclusões obtidas por meio deste trabalho.

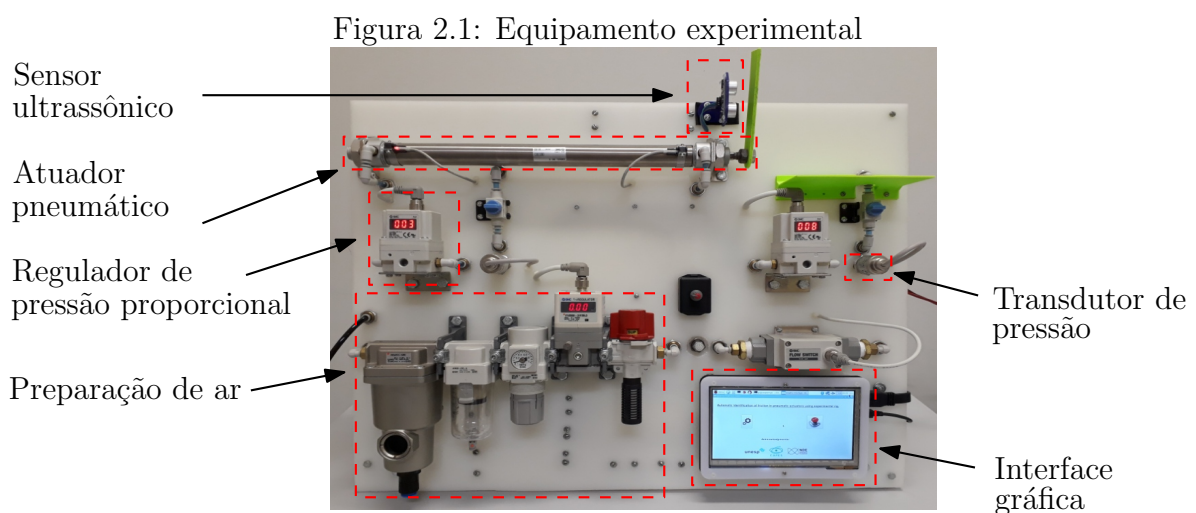
CAPÍTULO 2

EQUIPAMENTO EXPERIMENTAL

Neste capítulo apresenta a montagem, da parte mecânica e elétrica, do equipamento experimental, desde a fase inicial até o seu término. Também mostra os dispositivos de baixo custo utilizados e a forma como funcionam. Além disso, apresenta a metodologia utilizada para realizar a aquisição de dados, ou seja, obter os resultados da caracterização de atrito do atuador pneumático.

2.1 MONTAGEM DO EQUIPAMENTO EXPERIMENTAL

A montagem do equipamento experimental é mostrada na Figura 2.1.



Essa montagem foi desenvolvida com base no diagrama pneumático, mostrado no Apêndice A.4.

O atuador pneumático, mostrado na Figura 2.2, refere-se ao modelo ISO6432 e foi utilizado na parte superior do equipamento experimental para facilitar a visualização do deslocamento da haste e para funcionar juntamente com o sensor de distância, descartando possíveis interferências na medição do deslocamento da haste. A entrada e saída do fluxo de ar comprimido do atuador estão conectadas na válvula de fechamento, que está conectada na reguladora de pressão proporcional.

Figura 2.2: Atuador pneumático ISO6432



Fonte: <http://www.smcbr.com.br>

A Tabela 2.1 mostra os parâmetros básicos do atuador pneumático utilizados no equipamento experimental. Para os volumes V_{da} e V_{db} das câmaras do atuador, o êmbolo encontra-se no fim de curso.

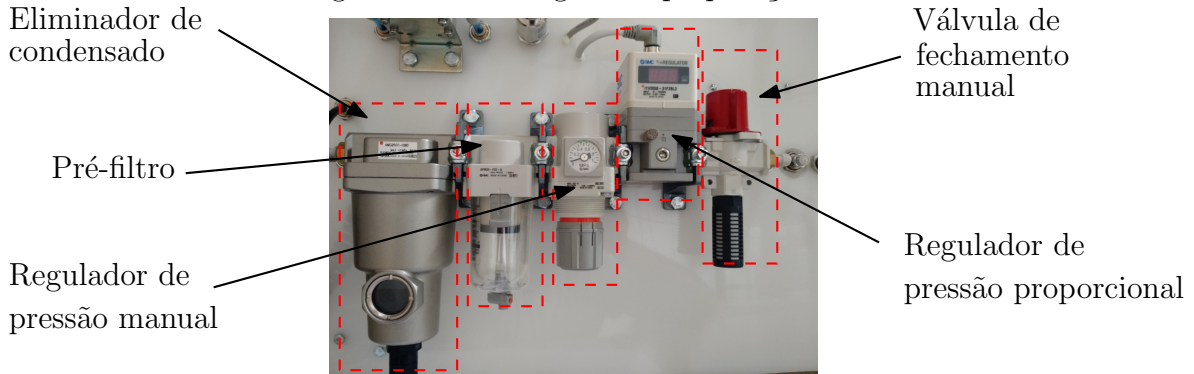
Tabela 2.1: Parâmetros básicos do atuador pneumático

Descrição dos parâmetros	Simbologia	Valor	Unidade
Diâmetro do êmbolo	D_p	0,025	m
Diâmetro da haste	D_r	0,010	m
Curso	S	0,300	m
Área do êmbolo	A_a	$4,909 \times 10^{-4}$	m^2
Área do êmbolo - Área da haste	A_b	$4,123 \times 10^{-4}$	m^2
Volume da câmara a	V_{da}	$1,473 \times 10^{-4}$	m^3
Volume da câmara b	V_{db}	$1,237 \times 10^{-4}$	m^3

A preparação de ar comprimido foi montada na forma de conjunto, conforme a seguinte sequência: eliminador de condensado, pré-filtro, regulador de pressão manual, regulador de pressão proporcional e válvula de fechamento manual. A Figura 2.3 mostra o conjunto da preparação de ar do equipamento experimental seguindo a ordem de montagem mencionada anteriormente.

O eliminador de condensado tem o objetivo de eliminar a água que pode se acumular dentro da tubulação de ar comprimido. Ele foi instalado na parte inferior do projeto experimental para evitar que o condensado atinja as partes mais frágeis do equipamento

Figura 2.3: Montagem da preparação de ar



experimental, como por exemplo: a eletrônica e os dispositivos eletropneumáticos. O eliminador de condensado possui uma eficiência em 99%, segundo o fabricante (SMC, 2011b).

O pré-filtro possui duas camadas de filtragem, a primeira filtra objetos maiores, realizando uma limpeza mais grosseira, e a segunda, realiza uma filtragem mais precisa, impedindo que qualquer objeto maior ao limite de filtragem ultrapasse o sistema de tratamento de ar. A primeira camada possui um filtro de $0,3 \mu m$, e a segunda, possui um filtro de $0,01 \mu m$, conforme consta no manual do fabricante (SMC, 2011a). O pré-filtro apresenta uma eficiência de 99,9%, segundo o fabricante.

O regulador de pressão proporcional tem como objetivo controlar as pressões desejadas de forma automática através de um sinal analógico. Foram utilizados dois reguladores para controlar as câmaras a e b do atuador pneumático, um em cada lado da câmara, a fim de facilitar a visualização do funcionamento do sistema pneumático. Também foi utilizado um regulador na entrada do sistema de preparação de ar para controlar a pressão de entrada do equipamento experimental.

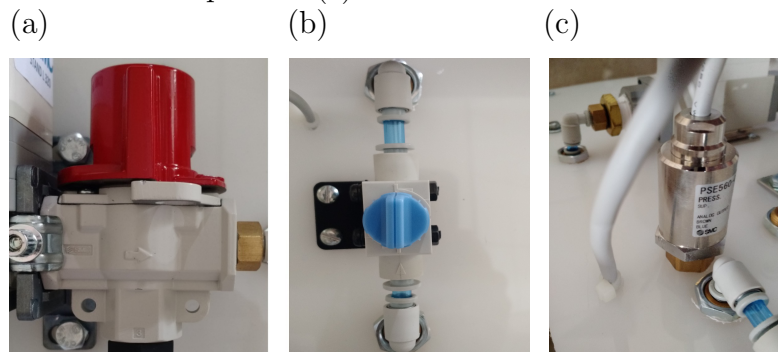
O regulador de pressão manual tem a função de controlar a pressão de entrada da preparação de ar através do ajuste de uma manopla, que determina a altura da agulha interna e a pressão de saída do regulador. Esse regulador foi utilizado antes do regulador de pressão proporcional da preparação de ar, a fim de realizar o ajuste grosso da pressão de entrada.

A válvula de fechamento manual tem a função de direcionar o ar comprimido, ou de bloquear a passagem de ar, descarregando a pressão existente no sistema pneumático

para a atmosfera. A válvula também possui um sistema de segurança para bloquear a entrada de ar comprimido, contendo um orifício na manopla e outro, no corpo da válvula para a utilização do cadeado, assim, impossibilitando a movimentação da manopla para liberar a passagem do ar. A válvula de fechamento manual utilizada na entrada da preparação de ar tem a finalidade de bloquear ou liberar o ar comprimido para o equipamento experimental. Já a válvula de fechamento manual da câmara tem a finalidade de bloquear ou liberar o fluxo de ar da saída do regulador de pressão proporcional. A Figura 2.4-a mostra a válvula de fechamento manual utilizada na preparação de ar e a Figura 2.4-b mostra a válvula de fechamento manual da câmara do atuador.

O transdutor de pressão, mostrado na Figura 2.4-c, tem a função de converter a variável de pressão instantânea em variável de medida elétrica, na forma de corrente ou de tensão elétrica. Essa variável de medida possui uma saída analógica com coeficiente angular proporcional. O transdutor é montado diretamente na saída de ar comprimido do regulador de pressão proporcional, que está interligado à entrada e à saída de ar do atuador pneumático.

Figura 2.4: Válvula de fechamento manual da preparação de ar (a) e da câmara do atuador (b), e transdutor de pressão (c)

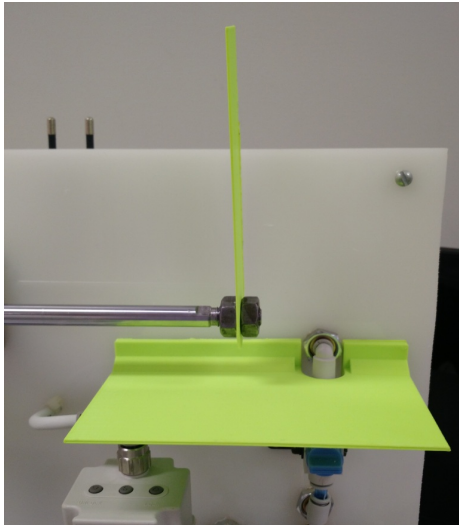


O suporte da haste e o suporte de proteção, mostrados na Figura 2.5-a, foram impressos na impressora 3D. O suporte da haste tem a finalidade de rebater as ondas ultrassônicas, direcionando-as ao receptor do sensor ultrassônico, obtendo assim, a distância do deslocamento da haste. Já o suporte de proteção objetiva proteger a área de medição do ultrassônico, sem interferir na medição do deslocamento da haste. Com esse suporte fixado não existe a possibilidade de interferências.

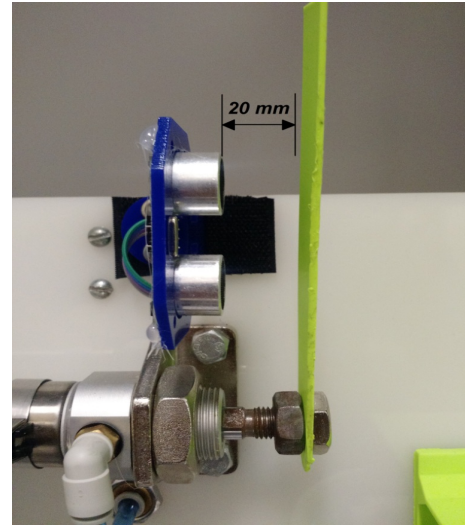
O suporte do sensor ultrassônico, mostrado na Figura 2.5-b, foi fixado a 20 mm do suporte da haste, pois o sensor ultrassônico inicia a medição a partir dessa distância.

Figura 2.5: Suporte da haste e suporte de proteção (a); e suporte de fixação do ultrassônico (b)

(a)



(b)



2.2 DISPOSITIVOS DE BAIXO CUSTO

A Figura 2.6 mostra o sensor ultrassônico utilizado no projeto experimental. O sensor ultrassônico HC-SR04 é capaz de medir distâncias de 2 cm a 4 m, com precisão de 2 mm. Esse sensor possui um circuito pronto com emissor e receptor acoplados e 4 pinos (Vcc, Trigger, Echo e Gnd) para alimentação e aquisição de dados.

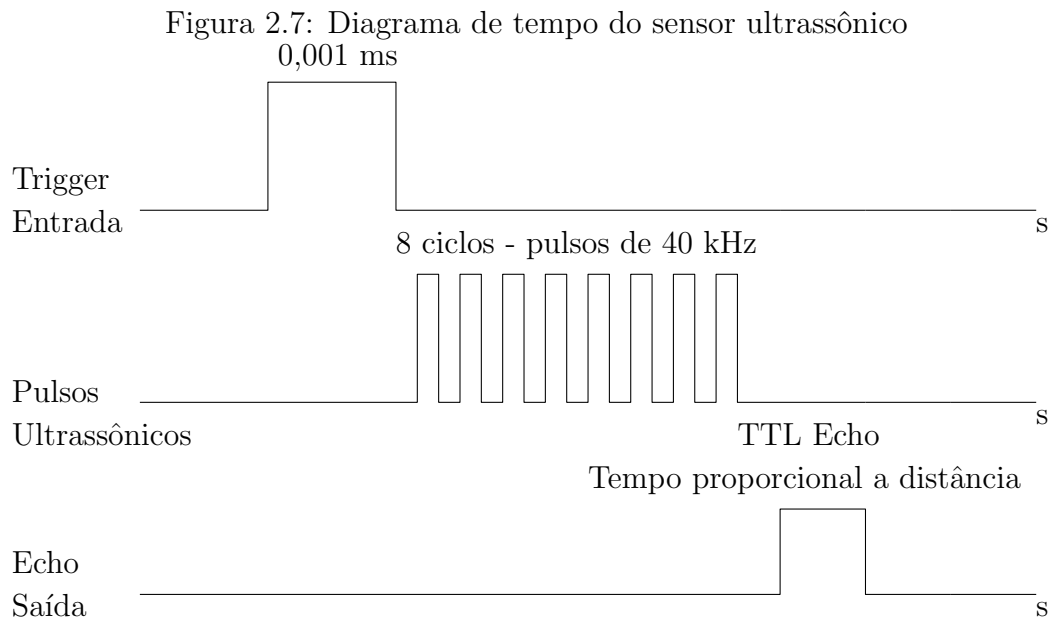
Figura 2.6: Sensor ultrassônico



Fonte: <http://www.electfreaks.com>

Os pinos Gnd (negativo) e Vcc (positivo) alimentam o sensor. Os outros dois são utilizados para a aquisição de dados e obtenção da distância. O pino Trigger, quando acionado, emite ondas sonoras que ao se depararem com um obstáculo retornam em direção ao receptor. Já o pino Echo emite um sinal de tensão após as ondas ultrassônicas serem emitidas, rebatidas e recebidas no receptor, possibilitando calcular o deslocamento da haste. O funcionamento do sensor ultrassônico é exemplificado através da Figura 2.7.

A distância medida pelo sensor ultrassônico é calculada considerando a velocidade



do som, $v = 340 \text{ m/s}$, e é obtida em metros, dada por:

$$D = \frac{t v}{2} \quad (2.1)$$

na qual v é a velocidade aproximada do som, t é o tempo em nível alto do pino Echo, e a divisão por 2 é devido ao fato da onda percorrer duas vezes a distância medida, na emissão e recepção.

O controlador *Raspberry Pi 3*, utilizado no trabalho, vem trazendo avanços em termos de processamento e performance em projetos. O objetivo do controlador é realizar a aquisição de dados das variáveis de medida em tempo real. Para a melhor precisão dessas variáveis é necessário um controlador com uma alta performance. Esse controlador possui uma desvantagem relacionada às variáveis analógicas, pois possui somente entradas e saídas digitais. Para realizar a aquisição de variáveis analógicas é necessário uma placa eletrônica externa com conversores AD/DA (Analógico - Digital e Digital - Analógico). O controlador *Raspberry Pi* e a placa de conversores AD/DA são mostrados na Figura 2.8.

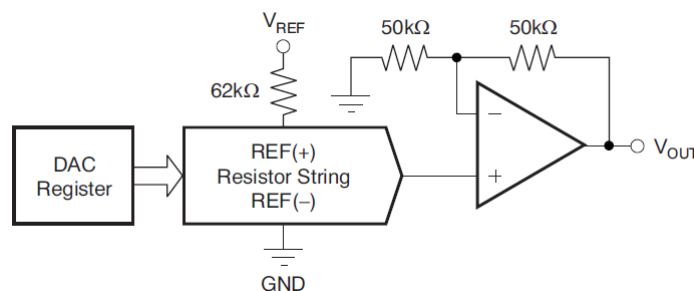
Figura 2.8: Controlador *Raspberry Pi 3* (a) e placa de conversores AD/DA (b)



Fonte: <https://www.raspberrypi.org> (a); <http://www.waveshare.com> (b)

A placa de conversores AD/DA possui conversores AD de 24 *bits* de resolução e conversores DA de 16 *bits* de resolução. A arquitetura básica do conversor DA, mostrada na Figura 2.9, utiliza o chipset DAC8532 junto com um registrador para guardar os valores digitais recebidos e convertê-los em níveis de tensão variáveis; um resistor *string* que realiza a divisão da tensão de referência; e um amplificador com a finalidade de gerar, na saída, a tensão de 0 a 5 V.

Figura 2.9: Arquitetura DAC8532



Fonte: <http://www.ti.com>

A equação, que determina a quantidade de estados do conversor DA, é dada por:

$$2^b = 2^{16} = 65536 \quad (2.2)$$

na qual b é a quantidade de *bits* do conversor DA.

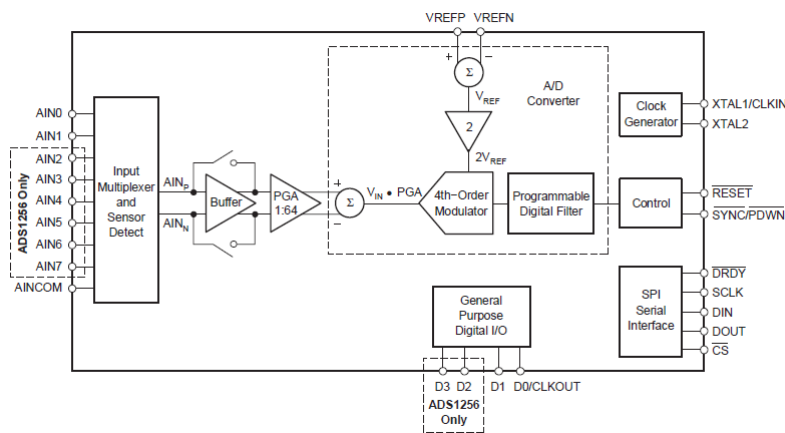
A equação, utilizada na programação do *Raspberry Pi* para determinar a tensão de saída (V_{out}) do conversor DA, é dada por:

$$V_{out} = V_{ref} \left(\frac{D}{65536} \right) \quad (2.3)$$

na qual V_{ref} é a tensão de referência.

A arquitetura básica do conversor AD, mostrada na Figura 2.10, utiliza o chipset ADS1256 e possui as seguintes características: seleção de entradas com multiplex; sensor para detectar a entrada utilizada; amplificador de ganho programável (PGA) para modificar a quantidade de amostra, por segundo, com o padrão de 30k sps (*samples per second*); e um modulador de quarta ordem juntamente com um filtro digital programável para configurar o PGA. A comunicação compatível utilizada é SPI (*Serial Peripheral Interface*) para controlar os conversores por meio do controlador *Raspberry Pi*.

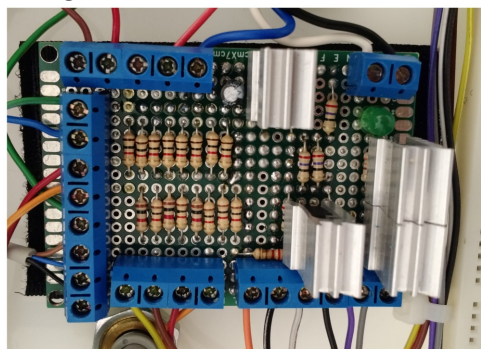
Figura 2.10: Arquitetura ADS1256



Fonte: <http://www.ti.com>

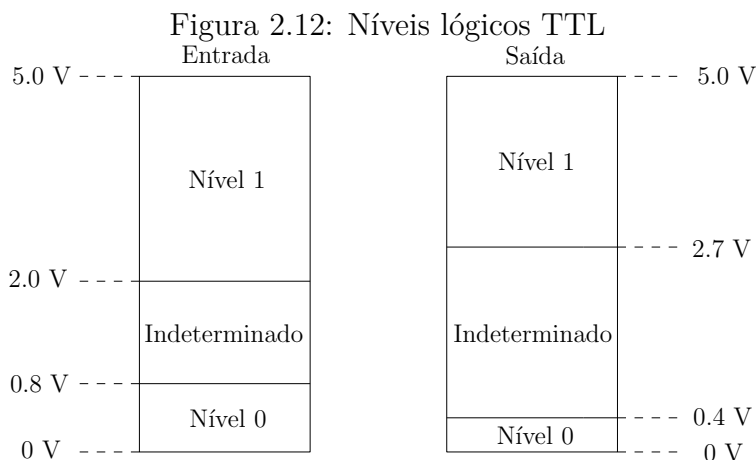
Os circuitos eletrônicos, mostrados nos Apêndices A.2 e A.3, foram montados na placa intermediária, como mostra a Figura 2.11. Essa placa foi desenvolvida com o intuito de unificar os circuitos eletrônicos que fazem os tratamentos de sinais conforme a necessidade de cada instrumento e do controlador *Raspberry Pi*.

Figura 2.11: Placa intermediária



Os dispositivos de baixo custo são projetados em tecnologia TTL (*Transistor-Transistor Logic*) com níveis de tensão específico. A Figura 2.12 mostra esses níveis de tensão. O

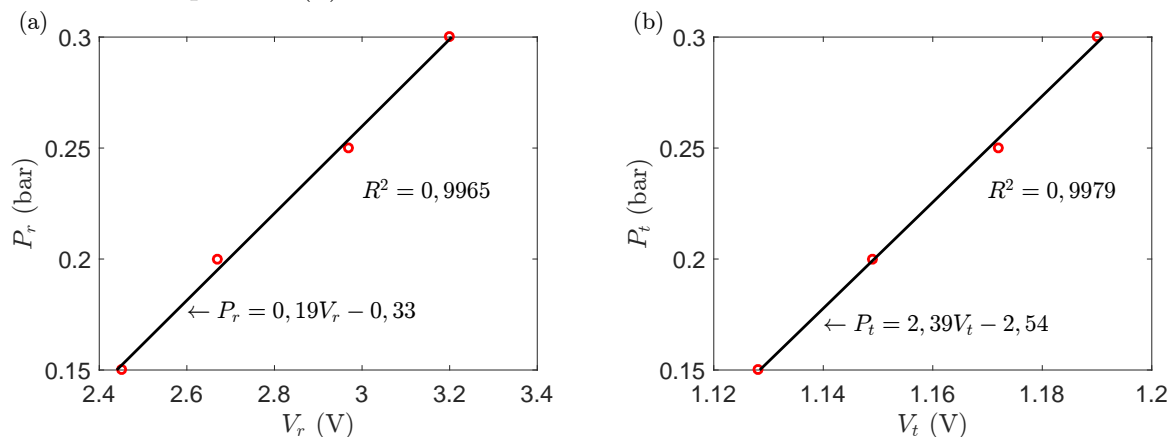
nível alto 1 (binário) é definido como aceitável dentro da escala; o nível indefinido pode ser alto ou baixo, sem previsão; e o nível baixo pode ser considerado 0 (binário) dentro da escala.



2.3 CALIBRAÇÃO DOS INSTRUMENTOS DE MEDIÇÃO

Para realizar o controle do regulador de pressão proporcional e a aquisição de dados do diferencial de pressão do atuador foi necessário encontrar as curvas características do regulador de pressão proporcional e do transdutor de pressão, conforme mostra a Figura 2.13. A tensão de entrada do regulador de pressão proporcional e a pressão correspondente a essa tensão são dadas, respectivamente, por V_r e P_r . Já a tensão de saída do transdutor de pressão e a pressão correspondente a essa tensão são dadas, respectivamente, por V_t e P_t .

Figura 2.13: Curva característica do regulador de pressão (a) e curva característica do transdutor de pressão (b)



2.4 FUNCIONAMENTO DO EQUIPAMENTO EXPERIMENTAL

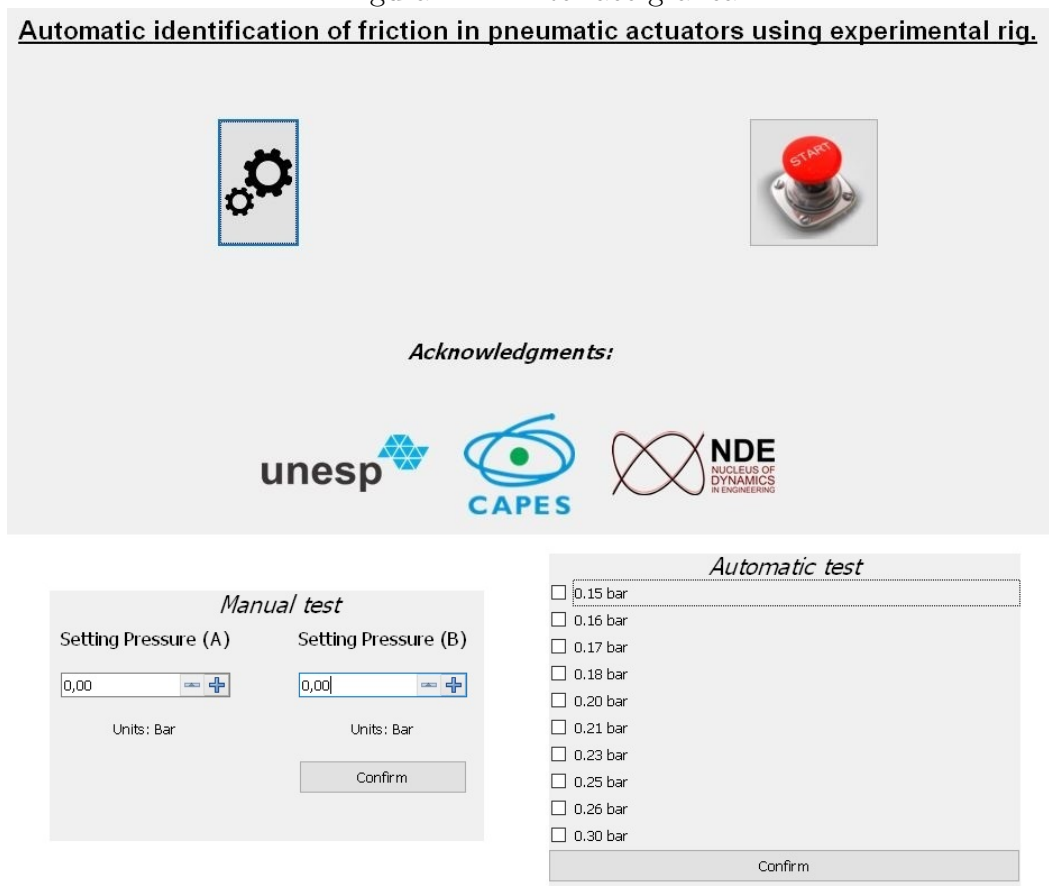
O sistema eletrônico do equipamento experimental foi desenvolvido com a finalidade de obter a comunicação entre os instrumentos e o controlador *Raspberry Pi*. Esse sistema possui circuitos eletrônicos que realizam o tratamento do sinal adequado para cada instrumento utilizado no equipamento experimental. E, para o funcionamento desse equipamento experimental foi desenvolvido uma programação em linguagem C, mostrada no Apêndice B.1.

A interface do usuário é realizada por meio de uma tela *touchscreen*, mostrada na Figura 2.14. Essa interface comunica-se com o *Raspberry Pi*, podendo configurar as pressões das câmaras do atuador pneumático para o ensaio manual e ensaio automático. Essa interface do usuário foi desenvolvida por meio de uma biblioteca gráfica GTK+, que tem uma flexibilidade na licença do aplicativo a ser desenvolvido e facilidade na comunicação com a linguagem C. Já na caracterização de atrito foi utilizada a biblioteca GSL, possibilitando realizar a regressão linear (*gsl_fit*) e a regressão não linear (*gsl_multifit_nlin*), estimando os parâmetros estáticos para calcular os parâmetros dinâmicos através das propriedades do modelo *LuGre*.

A caracterização de atrito em atuadores pneumáticos é realizada a partir do usuário, que insere todas as informações necessárias na interface gráfica. Essa interface gráfica possui uma tela primária com dois botões, o primeiro botão refere-se à configuração de cada ensaio no atuador pneumático e o segundo botão inicia-se a caracterização de atrito.

A configuração dos parâmetros de entrada para a caracterização de atrito em atuadores pneumáticos de simples ação e dupla ação é realizada por meio da interface gráfica. Em uma tela secundária é possível escolher o ensaio manual ou o ensaio automático. Na tela terciária é configurado o ensaio manual, que possui apenas um ensaio de pressão no atuador pneumático, sendo que é possível escolher o valor de pressão para a câmara a e para câmara b em atuadores de dupla ação. Caso o atuador pneumático seja de simples ação também é possível caracterizar o atrito, necessita apenas inserir o valor zero no campo da pressão referente à câmara que não será utilizada no ensaio manual. Em outra tela terciária é realizado o ensaio automático, que possui dez ensaios de pressão na ex-

Figura 2.14: Interface gráfica



tensão e na compressão do atuador pneumático, sendo que é possível escolher cada ensaio por meio de botões na interface gráfica. Caso o atuador pneumático seja simples ação também é possível caracterizar o atrito interno automaticamente, necessitando apenas escolher a opção de simples ação por meio do botão.

A caracterização de atrito para o atuador pneumático utilizado nesse equipamento experimental possui um tempo médio de 3,30 minutos, para identificar a força de atrito estático que ocorre quando o êmbolo se desloca e realizar o ensaios automáticos para identificar a força de atrito em função da velocidade em regime permanente. Os resultados da caracterização de atrito são mostrados na interface gráfica, com os parâmetros estáticos e dinâmicos, o coeficiente de determinação R^2 e o gráfico da força de atrito em função da velocidade.

A Figura 2.15 mostra a vista frontal e a vista posterior do equipamento experimental, com numeração dos principais dispositivos eletrônicos; e também, os instrumentos eletropneumáticos e pneumáticos utilizados para a aquisição de dados. Essa numeração é essencial para compreender o diagrama de aquisição de dados. O diagrama de aquisi-

ção de dados é mostrado na Figura 2.16. Esse diagrama possui a mesma numeração dos principais dispositivos eletrônicos e dos instrumentos eletropneumáticos e pneumáticos, mostrados da Figura 2.15. Através do diagrama é possível observar a forma como foram realizadas as ligações em cada dispositivo e instrumento, sendo que RX é a recepção do sinal e TX é a transmissão do sinal.

Figura 2.15: Vista frontal (a) e vista posterior (b), ambos do equipamento experimental

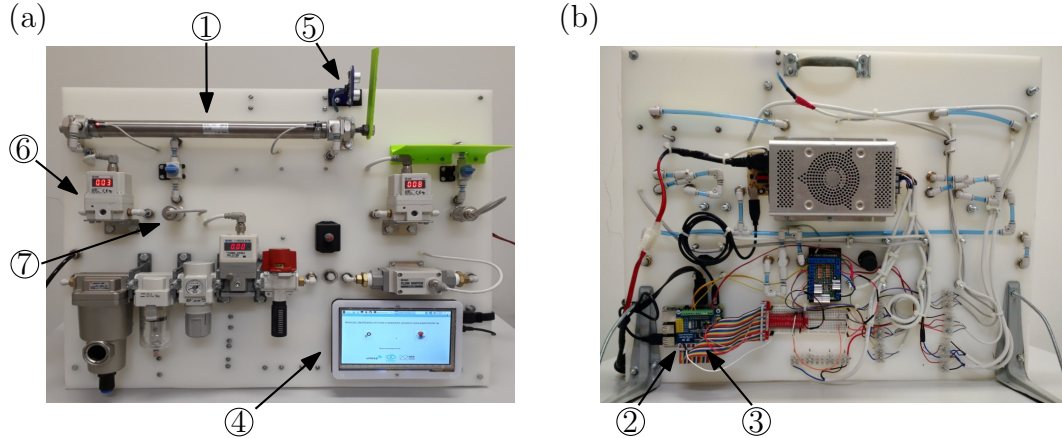
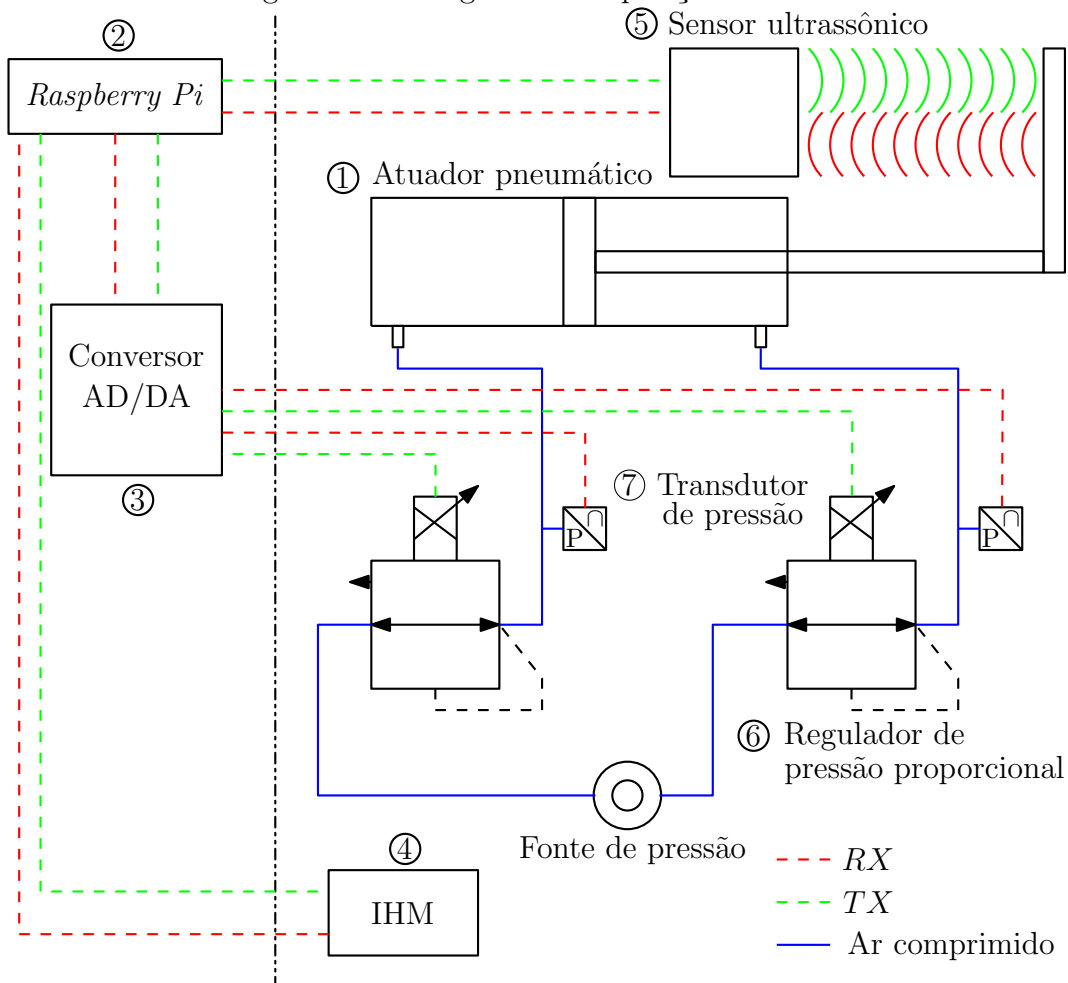


Figura 2.16: Diagrama de aquisição de dados



CAPÍTULO 3

CARACTERIZAÇÃO DE ATRITO

Neste capítulo apresenta a identificação experimental das características de atrito do atuador pneumático. Essa identificação foi realizada na extensão e compressão do atuador, sendo que os experimentos foram realizados por meio da variação de pressão e, conseqüentemente, da velocidade. A partir da caracterização de atrito foi possível obter o mapa estático de atrito, mostrando a força de atrito em função da velocidade em regime permanente; e identificar os parâmetros estáticos e dinâmicos.

3.1 PROCEDIMENTO PARA CARACTERIZAÇÃO DE ATRITO

Para obter os resultados da caracterização de atrito em atuadores pneumáticos na extensão e compressão é necessário seguir alguns passos. Primeiramente, é necessário obter o deslocamento do êmbolo em função do tempo por meio do sensor ultrassônico e, posteriormente, obter a pressão pneumática em função do tempo através dos transdutores de pressão. Após a aquisição de dados das variáveis de deslocamento e de pressão pneumática é preciso realizar uma regressão linear em cada uma dessas variáveis, numa faixa de tempo. Essa faixa de tempo é obtida através do deslocamento de todos ensaios, identificando graficamente a faixa de tempo que possui a melhor tendência linear. Porém, cada atuador pneumático obterá faixas de tempo diferentes. A tendência linear ocorre devido à pressão pneumática de cada ensaio ser constante. Assim, a velocidade em regime permanente de cada ensaio é encontrada por meio do deslocamento em função da faixa de tempo, obtida pela regressão linear.

Portanto, com a regressão linear realizada na pressão pneumática é possível encontrar a força pneumática, através da segunda lei de *Newton*, referente ao equilíbrio dinâmico, obtido através das forças exercidas no êmbolo, resultando em:

$$M\ddot{y} + F_f = F_p - F_l \quad (3.1)$$

no qual F_f é a força de atrito que ocorre principalmente nas superfícies de contato entre a haste e as vedações; F_l representa a força da carga externa; F_p é a força pneumática; M refere-se a massa do êmbolo mais a massa da haste; e \ddot{y} é a aceleração do deslocamento da haste. Essa força pneumática corresponde a diferença do produto da pressão e da área das câmaras do atuador pneumático, no momento em que a velocidade é constante e a aceleração é nula. Assim, a força de atrito é igual a força pneumática, que é dada por:

$$F_p = A_a P_a - A_b P_b \quad (3.2)$$

em que A_a , P_a , A_b e P_b correspondem, respectivamente, a área da câmara a, a pressão da câmara a, a área da câmara b e a pressão da câmara b.

Após identificar a força pneumática é encontrada a força de atrito estático, que foi obtida em dez ensaios no momento que êmbolo começa a deslocar. Assim, realizando a média das forças de atrito estático encontradas. A força de atrito estático é dada por:

$$F_s = P_s A_a \operatorname{sgn}(\dot{y}) \quad (3.3)$$

na qual P_s é a pressão instantânea no momento em que o êmbolo se desloca.

E, por fim, obtém a caracterização de atrito por meio do mapa estático, que corresponde a força de atrito em função da velocidade em regime permanente. Para obtenção do mapa estático foram realizados vinte ensaios em um processo automático, variando a pressão de entrada do atuador pneumático e, conseqüentemente, a velocidade em regime permanente para cada ensaio, obtendo os pontos experimentais. Cada ponto experimental representa um ensaio realizado em regime permanente, identificando a força de atrito em velocidades positivas e negativas. Após a realização dos ensaios, o modelo *LuGre* foi

ajustado através dos pontos experimentais, com a utilização da regressão não linear dos pontos experimentais. Assim, possibilitando visualizar graficamente o mapa de atrito estático com os pontos experimentais e a regressão não linear desse pontos, e identificar os parâmetros estáticos e dinâmicos referente ao atrito do atuador pneumático, através das propriedades do modelo *LuGre*.

3.2 ENSAIOS PARA EXTENSÃO E COMPRESSÃO

A Figura 3.1-a mostra o deslocamento do êmbolo y em função do tempo t para a extensão do atuador, nesse caso é possível visualizar a zona morta do sensor ultrassônico no início da medição, e a Figura 3.1-b mostra o mesmo deslocamento, sem a zona morta. Essa zona morta do sensor indica a maior variação do deslocamento sem provocar mudança no sinal de saída do sensor. A Figura 3.1-c mostra o diferencial de pressão nas câmaras do atuador em função do tempo para extensão, nota-se que o diferencial de pressão corresponde à pressão pneumática P_p ; e a Figura 3.1-d mostra o mesmo diferencial de pressão, mas sem o pico inicial de pressão gerado pelo regulador de pressão proporcional. A Figura 3.1-e mostra a força pneumática F_p em função do tempo para a extensão do atuador e a Figura 3.1-f mostra a mesma força pneumática, porém sem o transiente causado pelo pico de pressão inicial gerado através do regulador de pressão proporcional. Essa força é calculada pela Equação (3.2).

Para a identificação do atrito na extensão do atuador foram realizados dez ensaios, variando a pressão de entrada do atuador pneumático. Cada ensaio foi realizado à pressão constante, em um intervalo de deslocamento de 10 a 25 cm, que apresentou velocidade constante, conforme mostra a Figura 3.1-g. A Figura 3.1-h mostra o mapa estático do atrito para a extensão do atuador pneumático.

Os ensaios realizados na compressão do atuador ocorrem do mesmo modo que os realizados na extensão. A Figura 3.2 mostra o deslocamento do êmbolo, o diferencial de pressão e a força pneumática. A Figura 3.3-a mostra a regressão linear do deslocamento em função do tempo para compressão e a Figura 3.3-b mostra o mapa estático do atrito.

Figura 3.1: Posição do êmbolo com zona morta (a) e sem zona morta (b), pressão pneumática com transiente (c) e sem transiente (d), força pneumática com transiente (e) e sem transiente (f), posição do êmbolo em função do tempo (g) e força de atrito em função da velocidade (h); todas para extensão do atuador pneumático (* pontos experimentais; — Ajuste dos pontos experimentais)

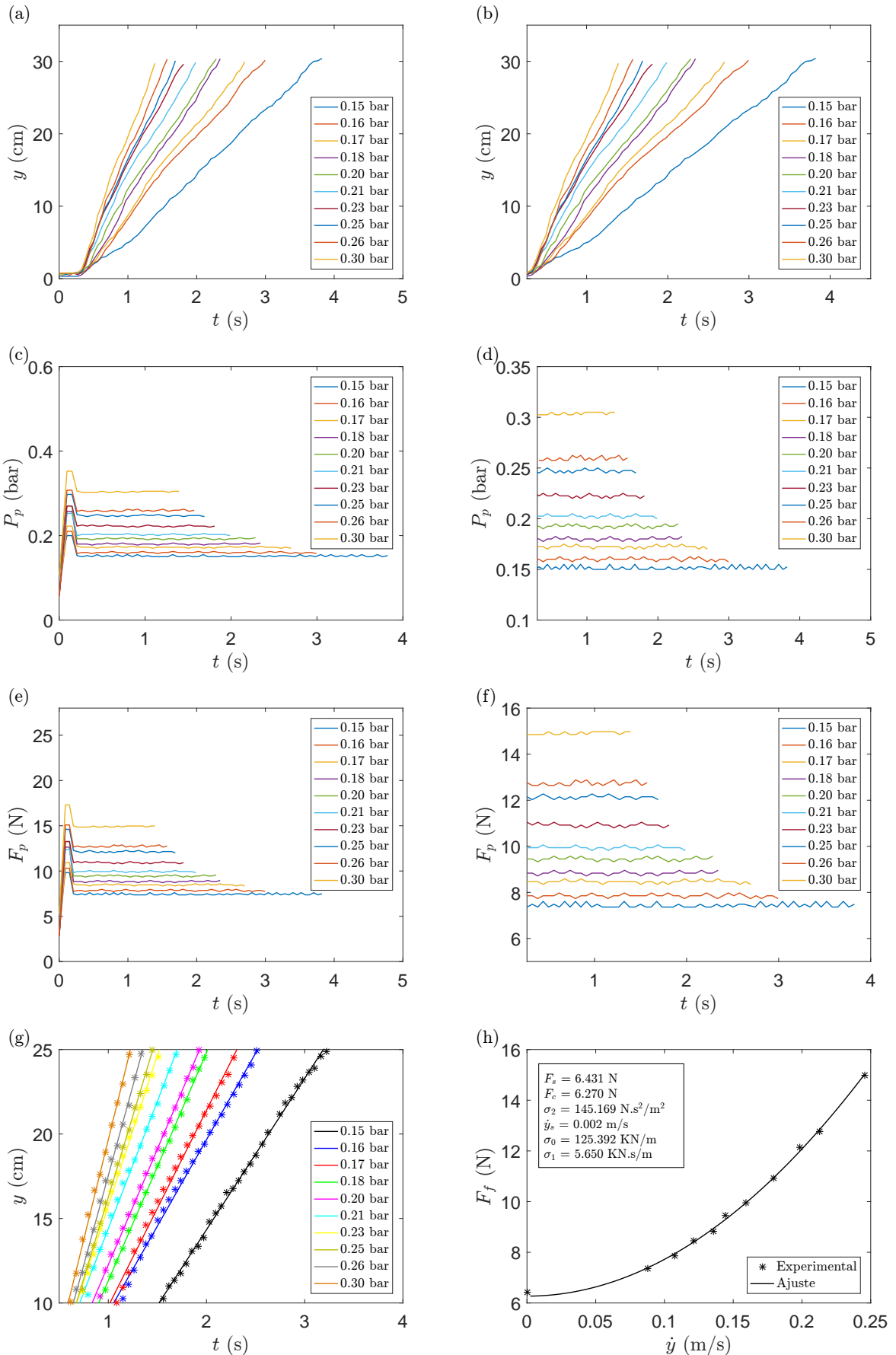


Figura 3.2: Posição do êmbolo sem zona morta (a), pressão pneumática sem transiente (b) e força pneumática sem transiente (c), todas em função do tempo

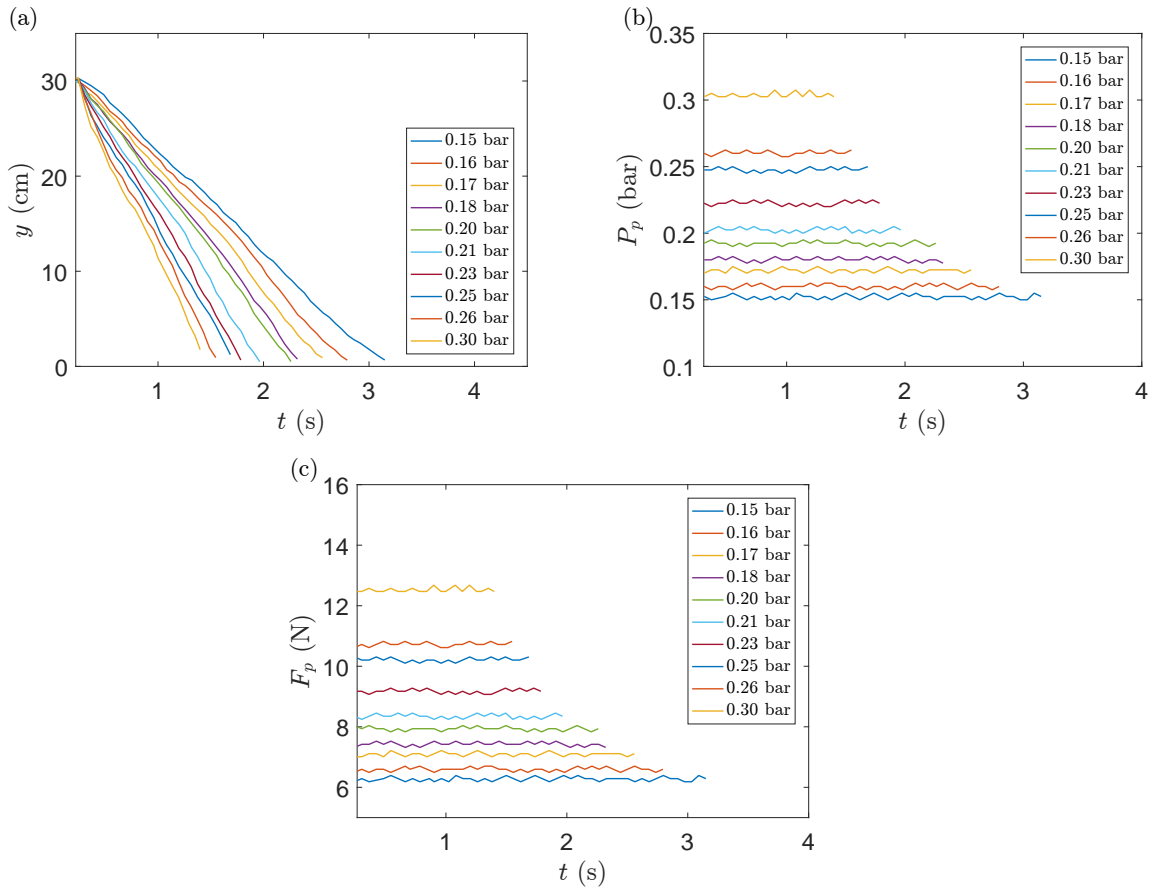
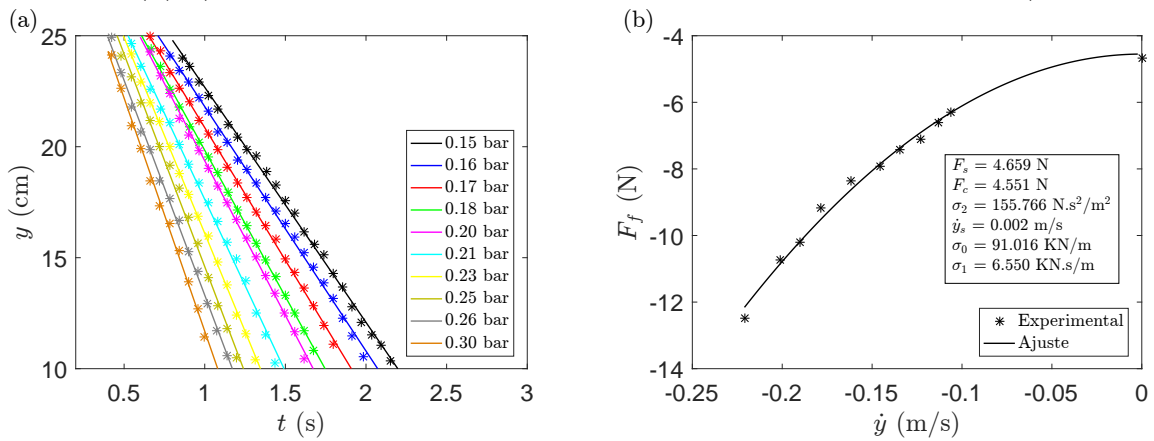


Figura 3.3: Posição do êmbolo em função do tempo (a) e força de atrito em função da velocidade (b); (* pontos experimentais; — Ajuste dos pontos experimentais)



A caracterização de atrito do atuador pneumático não é realizada no início e nem no fim de curso, pois com o modelo *LuGre* é possível obter a força de atrito em velocidades constantes sem afetar a precisão de um sistema de posicionamento.

A Tabela 3.1 e a Tabela 3.2 mostram os intervalos de confiança do deslocamento em função do tempo, referentes a cada regressão linear na extensão e compressão do atuador pneumático para encontrar os parâmetros c_1 e c_2 da equação da reta ($c_1x + c_2 = 0$), informando os valores mínimos e máximos de cada parâmetro.

Tabela 3.1: Intervalos de confiança para regressão linear da extensão

Intervalos de confiança para 95%				
$P_a - P_b$	c_1		c_2	
	mínimo	máximo	mínimo	máximo
0,15	8,8199	9,1035	-3,9337	-3,2446
0,16	9,9674	10,4941	-1,2899	-0,3027
0,17	11,1455	12,0596	-2,5872	-1,0147
0,18	13,2588	13,7485	-2,5551	-1,8172
0,20	13,3771	14,0387	-1,9314	-0,9716
0,21	14,4826	15,6745	-1,4365	0,0769
0,23	17,1942	18,5164	-2,8666	-1,3497
0,25	18,7008	19,8244	-3,6346	-2,3822
0,26	20,4255	21,9887	-4,4666	-2,8662
0,30	22,5062	24,8187	-5,0853	-2,9255

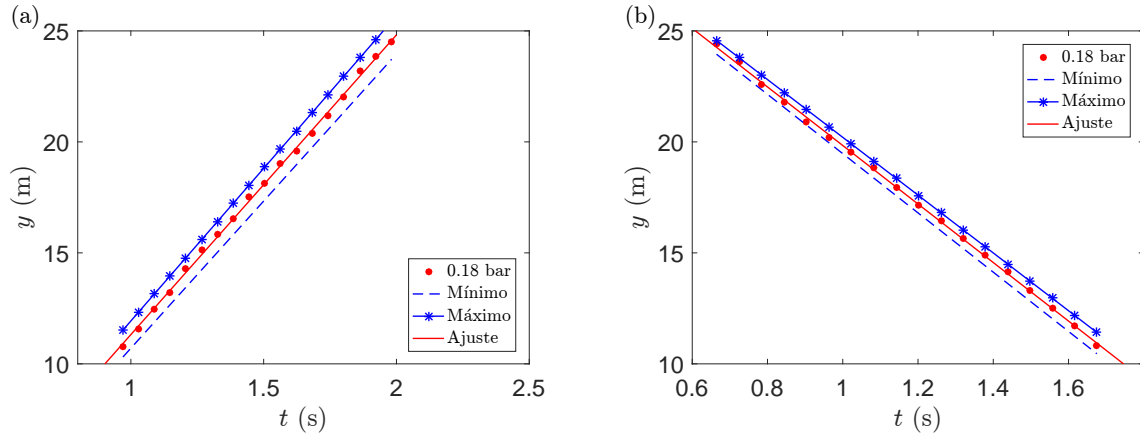
Tabela 3.2: Intervalos de confiança para regressão linear da compressão

Intervalos de confiança para 95%				
$P_a - P_b$	c_1		c_2	
	mínimo	máximo	mínimo	máximo
0,15	-10,7912	-10,3907	32,9579	33,5788
0,16	-11,2591	-10,7656	32,4671	33,1707
0,17	-12,1155	-11,7810	32,5931	33,0204
0,18	-13,3505	-13,0188	32,8217	33,2234
0,20	-14,2210	-13,6372	32,9512	33,6380
0,21	-16,0572	-14,9648	32,5751	33,6980
0,23	-17,9640	-17,3282	33,4368	34,0476
0,25	-20,0063	-18,2846	32,9934	34,4909
0,26	-20,0367	-18,6699	32,1107	33,2195
0,30	-22,0059	-20,0029	31,9476	33,4359

Desses vinte ensaios realizados a pressão inferior corresponde a 0,15 bar, devido a limitação física do atuador pneumático, pois a haste do atuador somente se desloca com uma pressão mínima de 0,15 bar. Já a pressão superior corresponde a 0,30 bar, devido a limitação da frequência de amostragem do conversor A/D, pois os ensaios com valores superiores 0,30 bar possuem poucos pontos experimentais coletados, tornando-se inviável.

A Figura 3.4 mostra graficamente dois dos ensaios realizados do deslocamento em função do tempo, observando que estão dentro do intervalo de confiança de 95%.

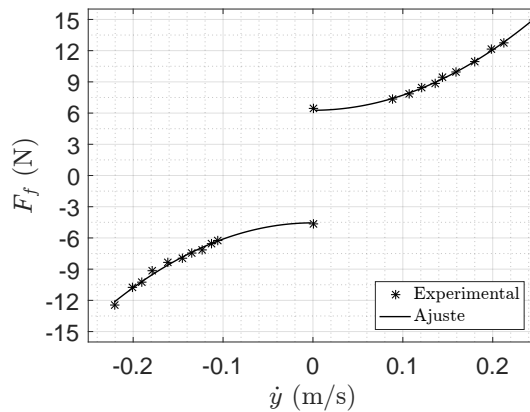
Figura 3.4: Dispersão do ensaio de 0,18 bar para extensão (a) e para compressão (b)



3.3 IDENTIFICAÇÃO DOS PARÂMETROS ESTÁTICOS E DINÂMICOS

Os parâmetros estáticos (F_c , F_s , \dot{y}_s e σ_2) são encontrados através da regressão não linear realizada na curva experimental, conforme mostra a Figura 3.5. A força de atrito para a extensão e compressão do atuador pneumático possui uma assimetria, devido às áreas das câmaras presentes no atuador serem diferentes ($A_a \neq A_b$) e também por outros fatores, como: mal lubrificação e condições físicas da vedação. Os coeficientes de determinação para velocidades positivas e velocidades negativas são, respectivamente, 0,998 e 0,983. Os parâmetros dinâmicos (σ_0 e σ_1) são encontrados através de simulações

Figura 3.5: Força de atrito em função da velocidade para extensão e compressão; (* pontos experimentais; — Ajuste dos pontos experimentais)



e da propriedade de passividade, referentes ao modelo *LuGre*. Para identificar o parâmetro σ_0 foi utilizada a metodologia proposta por Valdiero (2005). Esse parâmetro foi

ajustado por meio de simulações, conforme duas orientações: a microdeformação z , que ocorre na região de pré-deslizamento, e a implementação de um observador de atrito, sem perda da estabilidade numérica por limitações do tempo de amostragem. Conforme Armstrong e Wit (1996), as microdeformações na região de pré-deslizamento estão na faixa de 1 a 50 μm . O parâmetro σ_0 é dado por:

$$\sigma_0 = \frac{F_c}{|z|} \quad (3.4)$$

no qual o valor de z é considerado 50 μm , pelo fato de ser a condição mais desfavorável em relação ao atrito. De acordo com Barahanov e Ortega (2000), para identificar o parâmetro dinâmico σ_1 é utilizada a propriedade de passividade do modelo *LuGre*. Assim, o parâmetro σ_1 é dado por:

$$\sigma_1 \leq \frac{\sigma_2}{\left(\frac{F_s}{F_c} - 1\right)} \quad (3.5)$$

A Tabela 3.3 mostra os parâmetros estáticos e dinâmicos do atrito existentes no atuador pneumático.

Tabela 3.3: Parâmetros estáticos e dinâmicos

Descrição dos parâmetros	Simbologia	$\dot{y} > 0$	$\dot{y} < 0$
Força de atrito estático	F_s	6,431 N	4,659 N
Força de atrito de <i>Coulomb</i>	F_c	6,270 N	4,551 N
Coefficiente de arraste	σ_2	145,169 N.s ² /m ²	155,766 N.s ² /m ²
Velocidade <i>Stribeck</i>	\dot{y}_s	0,002 m/s	0,002 m/s
Coefficiente de rigidez	σ_0	125,392 KN/m	91,016 KN/m
Coefficiente de amortecimento	σ_1	5,650 KN.s/m	6,550 KN.s/m

CAPÍTULO 4

CONCLUSÕES FINAIS E TRABALHOS FUTUROS

Nesse capítulo são apresentados os comentários finais pertinentes a caracterização de atrito interno em atuadores pneumáticos. Também é apresentada uma sugestão de trabalho futuro, com o objetivo de continuar às análises mostradas nesse trabalho e aprimorar o equipamento experimental.

4.1 CONCLUSÕES FINAIS

A dissertação trata-se da identificação do atrito interno do atuador pneumático, modelo ISO6432, de haste simples e dupla ação. Com o equipamento experimental desenvolvido foi possível realizar a caracterização de atrito de forma automática, utilizando dispositivos de baixo custo. Desse modo, pode-se afirmar que, mesmo com dispositivos de baixo custo, é possível alcançar resultados desejáveis que podem contribuir para compensação de atrito em um sistema de servoposicionamento.

Por meio do modelo *LuGre* pôde-se visualizar as características não lineares do atrito dinâmico e, através do mapa estático de atrito, foi possível analisar algumas características não lineares do atrito, encontrando os parâmetros estáticos e dinâmicos com o ajuste de regressão não linear do modelo matemático. Esse mapa estático de atrito corresponde à força de atrito em função da velocidade, que é mostrado após o ensaio automático com seus respectivos parâmetros. Assim, realizando a identificação automática de atrito em

atuadores pneumáticos.

A força de atrito caracterizada no atuador pneumático do equipamento experimental possui uma assimetria em relação à extensão e compressão. Essa assimetria ocorre, principalmente, devido à diferença das áreas das câmaras e a outros fatores, como: mau lubrificação e geometria da vedação.

Portanto, com esse equipamento experimental de baixo custo poderá proporcionar utilidades importantes no meio industrial, por exemplo, no monitoramento do desgaste das vedações do atuador pneumático, e na identificação dos parâmetros de atrito estáticos e dinâmicos para inclusão nos manuais dos fabricantes.

4.2 TRABALHOS FUTUROS

Um próximo passo para essa pesquisa seria a realização de uma análise estatística do erro de cada parâmetro identificado e realização do controle de posicionamento do atuador pneumático, utilizando os parâmetros da caracterização de atrito obtida nesse trabalho.

O objetivo principal seria obter o controle do sistema de posicionamento pneumático com a maior precisão possível, utilizando os parâmetros estáticos e dinâmicos identificados na caracterização de atrito.

Esse objetivo principal poderia ser dividido em três partes:

- (a) Obter o modelo matemático que descreva o sistema de posicionamento do atuador pneumático;
- (b) Realizar o controle do sistema de posicionamento pneumático utilizando os parâmetros identificados da caracterização de atrito;
- (c) Validar o modelo matemático do sistema de posicionamento pneumático com o equipamento experimental, através de ensaios experimentais e simulações.

CAPÍTULO 5

TRABALHOS PUBLICADOS

Neste capítulo apresenta os trabalhos publicados durante o período do mestrado em Engenharia Mecânica. Os trabalhos publicados são:

FELIX, R. O.; SILVEIRA, M. Experimental characterization of the nonlinear characteristics of friction in pneumatic actuators. In: *Proceedings of the 24th ABCM International Congress of Mechanical Engineering*. ABCM, 2017. Disponível em: <<https://doi.org/10.26678/abcm.cobem2017.cob17-0477>>.

FELIX, R. O.; SILVEIRA, M. Low cost measurement of friction force characteristics of pneumatic actuator. In: SEMINÁRIO DA PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA, XI., 2017. Bauru: FEB-UNESP.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALCIATORE, D. G.; HISTAND, M. B. *Introdução à Mecatrônica e aos Sistemas de Medições*. 4. ed. Porto Alegre: AMGH Editora, 2014.
- ANDERSSON, S.; SÖDERBERG, A.; BJÖRKLUND, S. Friction models for sliding dry, boundary and mixed lubricated contacts. *Tribology international*, Elsevier, v. 40, n. 4, p. 580–587, 2007. Disponível em: <<https://doi.org/10.1016/j.triboint.2005.11.014>>.
- ANDRIGHETTO, P. L.; VALDIERO, A. C.; CARLOTTO, L. Study of the friction behavior in industrial pneumatic actuators. In: ABCM SYMPOSIUM SERIES IN MECHATRONICS, 2005, Ouro Preto. *Anais*. Ouro Preto: ABCM, 2006. v. 2, p. 369–376.
- ARMSTRONG, B.; WIT, C. C. D. Friction modeling and compensation. *The control handbook*, CRC Press Boca Raton, FL, v. 77, p. 1369–1382, 1996.
- BARAHANOV, N.; ORTEGA, R. Necessary and sufficient conditions for passivity of the lugre friction model. *IEEE Transactions on Automatic Control*, IEEE, v. 45, n. 4, p. 830–832, 2000. Disponível em: <<https://dx.doi.org/10.1109/9.847131>>.
- BELFORTE, G.; CONTE, M.; BERTETTO, A. M.; MAZZA, L.; VISCONTE, C. Experimental and numerical evaluation of contact pressure in pneumatic seals. *Tribology international*, Elsevier, v. 42, n. 1, p. 169–175, 2009. Disponível em: <<https://dx.doi.org/10.1016/j.triboint.2008.04.010>>.
- BELFORTE, G.; MANUELLO, A.; MAZZA, L. Optimization of the cross section of an elastomeric seal for pneumatic cylinders. *Journal of Tribology*, American Society of Mechanical Engineers, v. 128, n. 2, p. 406–413, 2006. Disponível em: <<https://dx.doi.org/10.1115/1.2162915>>.
- CARNEIRO, J. F.; ALMEIDA, F. G. de. Lugre friction model: Application to a pneumatic actuated system. In: CONTROLO'2014–PROCEEDINGS OF THE PORTUGUESE CONFERENCE ON AUTOMATIC CONTROL, 11., 2014, Chicago. *Anais eletrônicos...* Porto: Springer, 2015. p. 459–468. Disponível em: <https://dx.doi.org/10.1007/978-3-319-10380-8_44>.
- DAHL, P. R. *A solid friction model*. Los Angeles, 1968.
- DUPONT, P.; ARMSTRONG, B.; HAYWARD, V. Elasto-plastic friction model: contact compliance and stiction. In: PROCEEDINGS OF THE AMERICAN CONTROL CONFERENCE, 2000, Chicago. *Anais eletrônicos...* Chicago: IEEE, 2000. p. 1072–1077. Disponível em: <<https://dx.doi.org/10.1109/ACC.2000.876665>>.
- ELFASAKHANY, A.; YANEZ, E.; BAYLON, K.; SALGADO, R. Design and development of a competitive low-cost robot arm with four degrees of freedom. *Modern Mechanical Engineering*, Scientific Research Publishing, v. 1, n. 02, p. 47, 2011.

FELIX, R. O.; SILVEIRA, M. Low cost measurement of friction force characteristics of pneumatic actuator. In: SEMINÁRIO DA PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA, XI., 2017. Bauru: FEB-UNESP.

FELIX, R. O.; SILVEIRA, M. Experimental characterization of the nonlinear characteristics of friction in pneumatic actuators. In: *Proceedings of the 24th ABCM International Congress of Mechanical Engineering*. ABCM, 2017. Disponível em: <<https://doi.org/10.26678/abcm.cobem2017.cob17-0477>>.

FIALHO, A. B. *Automação Pneumática - Projetos, Dimensionamento e Análise de Circuitos*. 2. ed. São Paulo: Editora Érica, 2004.

FUNG, R.-F.; HAN, C.-F.; CHANG, J.-R. Dynamic modeling of a high-precision self-moving stage with various frictional models. *Applied Mathematical Modelling*, Elsevier, v. 32, n. 9, p. 1769–1780, 2008. Disponível em: <<https://dx.doi.org/10.1016/j.apm.2007.06.012>>.

GONÇALVES, P. J. P.; SILVEIRA, M. Condition monitoring of structures under non-ideal excitation using low cost equipment. In: *Probabilistic Prognostics and Health Management of Energy Systems*. Springer International Publishing, 2017. p. 241–261. Disponível em: <https://doi.org/10.1007/978-3-319-55852-3_15>.

GUENTHER, R.; PERONDI, E. A.; DEPIERI, E. R.; VALDIERO, A. C. Cascade controlled pneumatic positioning system with lugre model based friction compensation. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, SciELO Brasil, v. 28, n. 1, p. 48–57, 2006. Disponível em: <<http://dx.doi.org/10.1590/S1678-58782006000100006>>.

HA, J. L.; FUNG, R. F.; HAN, C. F.; CHANG, J. R. Effects of frictional models on the dynamic response of the impact drive mechanism. *Journal of vibration and acoustics*, American Society of Mechanical Engineers, v. 128, n. 1, p. 88–96, 2006. Disponível em: <<http://dx.doi.org/10.1115/1.2128641>>.

HAMITI, K.; VODA-BESANCON, A.; ROUX-BUISSON, H. Position control of a pneumatic actuator under the influence of stiction. *Control Engineering Practice*, Elsevier, v. 4, n. 8, p. 1079–1088, 1996. Disponível em: <[https://dx.doi.org/10.1016/0967-0661\(96\)00106-2](https://dx.doi.org/10.1016/0967-0661(96)00106-2)>.

HVOLDAL, M.; OLESEN, C. *Friction modelling and parameter estimation for hydraulic asymmetrical cylinders*. 100 f. Dissertação (Mestrado em Projeto de Sistema Eletro-Mecânico) — Universidade de Aalborg, Aalborg, 2011.

JOHANASTROM, K.; WIT, C. C. Revisiting the lugre friction model. *IEEE control Systems*, IEEE, v. 28, n. 6, p. 101–114, 2008. Disponível em: <<http://dx.doi.org/10.1109/MCS.2008.929425>>.

JOHNSON, C. T.; LORENZ, R. D. Experimental identification of friction and its compensation in precise, position controlled mechanisms. *IEEE Transactions on Industry Applications*, IEEE, v. 28, n. 6, p. 1392–1398, 1992. Disponível em: <<https://dx.doi.org/10.1109/28.175293>>.

KRAUSS, R. Combining raspberry pi and arduino to form a low-cost, real-time autonomous vehicle platform. In: IEEE. *American Control Conference (ACC), 2016*. [S.l.], 2016. p. 6628–6633.

KWAK, B.-J.; YAGLE, A. E.; LEVITT, J. A. Nonlinear system identification of hydraulic actuator. friction dynamics using a hammerstein model. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1998, Seattle. *Anais eletrônicos...* Seattle: IEEE, 1998. p. 1933–1936. Disponível em: <<https://dx.doi.org/10.1109/ICASSP.1998.681441>>.

LAFMEJANI, A. S.; MASOULEH, M. T.; KALHOR, A. An experimental study on friction identification of a pneumatic actuator and dynamic modeling of a proportional valve. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND MECHATRONICS, 4., 2016, Tehran. *Anais eletrônicos...* Tehran: IEEE, 2016. p. 166–172. Disponível em: <<https://dx.doi.org/10.1109/ICRoM.2016.7886840>>.

MÁRTON, L.; FODOR, S.; SEPEHRI, N. A practical method for friction identification in hydraulic actuators. *Mechatronics*, Elsevier, v. 21, n. 1, p. 350–356, 2011. Disponível em: <<https://dx.doi.org/10.1016/j.mechatronics.2010.08.010>>.

MÁRTON, L.; LANTOS, B. Identification and model-based compensation of striebeck friction. *Acta Polytechnica Hungarica*, Óbudai Egyeten, v. 3, n. 3, p. 45–58, 2006.

MOREIRA, I. D. S. *Sistemas pneumáticos*. São Paulo: SESI SENAI Editora, 2012.

OLSSON, H.; ÅSTRÖM, K. J.; WIT, C. C.; GÄFVERT, M.; LISCHINSKY, P. Friction models and friction compensation. *European journal of control*, Elsevier, v. 4, n. 3, p. 176–195, 1998. Disponível em: <[https://dx.doi.org/10.1016/S0947-3580\(98\)70113-X](https://dx.doi.org/10.1016/S0947-3580(98)70113-X)>.

PINEÑO, O. Arduipod box: A low-cost and open-source skinner box using an ipod touch and an arduino microcontroller. *Behavior research methods*, Springer, v. 46, n. 1, p. 196–205, 2014.

PUGLISI, L. J.; SALTAREN, R. J.; GARCIA, C.; BANFIELD, I. A. Robustness analysis of a pi controller for a hydraulic actuator. *Control Engineering Practice*, Elsevier, v. 43, p. 94–108, 2015. Disponível em: <<https://dx.doi.org/10.1016/j.conengprac.2015.06.010>>.

QIAN, Y.; OU, G.; MAGHAREH, A.; DYKE, S. J. Parametric identification of a servo-hydraulic actuator for real-time hybrid simulation. *Mechanical Systems and Signal Processing*, Elsevier, v. 48, n. 1, p. 260–273, 2014. Disponível em: <<https://dx.doi.org/10.1016/j.ymsp.2014.03.001>>.

RAKOTO, L.; SCHORSCH, J.; KINNAERT, M. Modelling hydraulic pitch actuator for wind turbine simulation under healthy and faulty conditions. *IFAC-PapersOnLine*, Elsevier, v. 48, n. 21, p. 577–582, 2015. Disponível em: <<https://dx.doi.org/10.1016/j.ifacol.2015.09.588>>.

RAVANBOD-SHIRAZI, L.; BESANÇON-VODA, A. Friction identification using the karnopp model, applied to an electropneumatic actuator. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, SAGE Publications Sage UK: London, England, v. 217, n. 2, p. 123–138, 2003. Disponível em: <<https://dx.doi.org/10.1177/095965180321700206>>.

ROMANO, V.; DUTRA, M. *Robótica Industrial Aplicação na Indústria de Manufatura e de Processos*: Introdução à robótica industrial. São Paulo: Edgard Blücher, 2002. 1–19 p.

SÁNCHEZ-MAZUCA, S.; CAMPA, R. An improvement proposal to the static friction model. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, v. 2013, n. 946526, p. 1–8, 2013. Disponível em: <<http://dx.doi.org/10.1155/2013/946526>>.

SCHOENMEIER, C.; HAMMES, G.; VALDIERO, A. C.; RASIA, L. A. Modelagem matemática e validação experimental de um atuador pneumático para uma bancada de ensaios de estruturas. *Revista Saberes e Sabores Educacionais*, n. 1, 2014.

SHIH, M.-C.; TSENG, S.-I. Identification and position control of a servo pneumatic cylinder. *Control Engineering Practice*, Elsevier, v. 3, n. 9, p. 1285–1290, 1995. Disponível em: <[https://dx.doi.org/10.1016/0967-0661\(95\)00127-G](https://dx.doi.org/10.1016/0967-0661(95)00127-G)>.

SMC. *Mist Separator*. Noblesville, 2011. Disponível em: <https://content2.smcetech.com/pdf/AC_A.pdf>.

SMC. *Water Separator*. Noblesville, 2011. Disponível em: <http://content2.smcetech.com/pdf/AM_AFF.pdf>.

STRANO, S.; TERZO, M. Accurate state estimation for a hydraulic actuator via a sdre nonlinear filter. *Mechanical Systems and Signal Processing*, Elsevier, v. 75, p. 576–588, 2016. Disponível em: <<https://dx.doi.org/10.1016/j.ymsp.2015.12.002>>.

SUN, Y.-H.; CHEN, T.; WU, C. Q.; SHAFAI, C. Comparison of four friction models: Feature prediction. *Journal of Computational and Nonlinear Dynamics*, American Society of Mechanical Engineers, v. 11, n. 3, p. 031009, 2016. Disponível em: <<https://dx.doi.org/10.1115/1.4031768>>.

VALDIERO, A. C. *Controle de robôs hidráulicos com compensação de atrito*. 158 f. Tese (Doutorado em Engenharia Mecânica) — Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2005.

WANG, J.; WANG, J.; DAW, N.; WU, Q. Identification of pneumatic cylinder friction parameters using genetic algorithms. *IEEE/ASME transactions on mechatronics*, IEEE, v. 9, n. 1, p. 100–107, 2004. Disponível em: <<https://dx.doi.org/10.1109/TMECH.2004.823883>>.

WIT, C. C.; OLSSON, H.; ASTROM, K. J.; LISCHINSKY, P. A new model for control of systems with friction. *IEEE Transactions on automatic control*, IEEE, v. 40, n. 3, p. 419–425, 1995. Disponível em: <<https://dx.doi.org/10.1109/9.376053>>.

WIT, C. C.; OLSSON, H.; JOHANASTROM, K.; LISCHINSKY, P. A new model for control of systems with friction. *IEEE Transactions on automatic control*, IEEE, v. 40, n. 3, p. 419–425, 1995. Disponível em: <<http://dx.doi.org/10.1109/9.376053>>.

WIT, C. C.; TSIOTRAS, P.; VELENIS, E.; BASSET, M.; GISSINGER, G. Dynamic friction models for road/tire longitudinal interaction. *Vehicle System Dynamics*, Taylor & Francis, v. 39, n. 3, p. 189–226, 2003. Disponível em: <<https://dx.doi.org/10.1076/vesd.39.3.189.14152>>.

WONDIMU, N. A. *Simulated and experimental sliding mode control of a hydraulic positioning system*. 100 f. Dissertação (Mestrado em Engenharia Mecânica) — University of Akron, Akron, 2006.

YAO, J.; JIAO, Z.; HAN, S. Friction compensation for low velocity control of hydraulic flight motion simulator: a simple adaptive robust approach. *Chinese Journal of Aeronautics*, Elsevier, v. 26, n. 3, p. 814–822, 2013. Disponível em: <<https://dx.doi.org/10.1016/j.cja.2013.04.001>>.

APÊNDICE A

CIRCUITOS ELETRÔNICOS E DIAGRAMA PNEUMÁTICO

O Apêndice A.1 mostra os terminais de conexões da parte posterior do equipamento experimental, com objetivo de facilitar a ligação elétrica dos instrumentos.

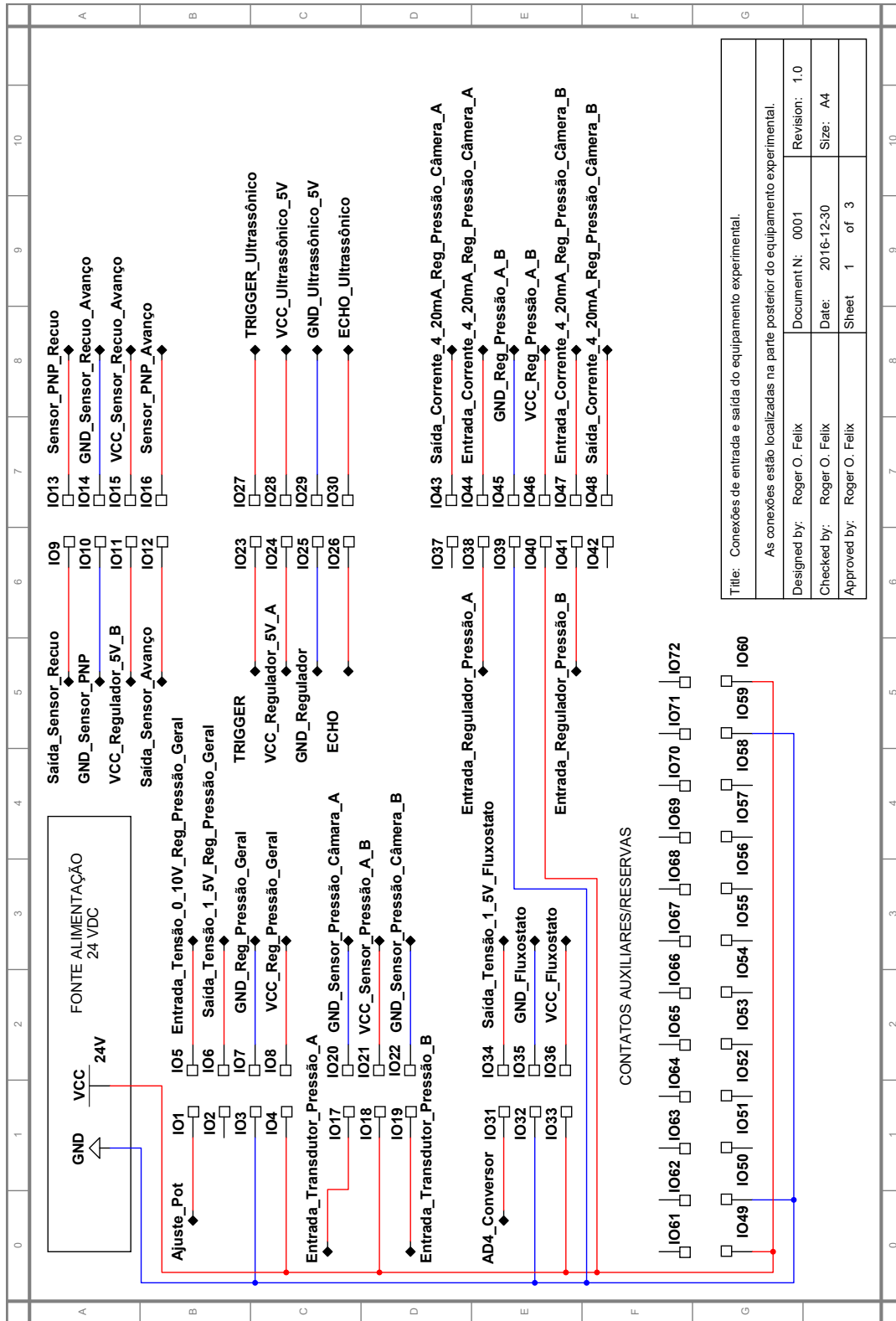
O Apêndice A.2 mostra o circuito eletrônico dos sensores de fim de curso, o circuito eletrônico dos reguladores de pressão proporcional e o circuito eletrônico do botão de segurança. O circuito eletrônico dos sensores de fim de curso possui uma divisão de tensão para enviar o sinal de saída ao *Raspberry Pi* na tensão adequada, assim, diminuindo a tensão de saída de 5 volts em 3,3 volts. O circuito eletrônico dos reguladores de pressão proporcional possui uma divisão de tensão do sinal recebido do conversor DA, convertendo esse sinal de tensão de 1 a 5 V em 4 a 20 mA. Também mostra o circuito do botão de segurança.

O Apêndice A.3 mostra o circuito eletrônico dos transdutores de pressão, o circuito eletrônico de ajuste manual da pressão principal e o circuito eletrônico do sensor ultrassônico. O circuito eletrônico dos transdutores de pressão possui uma divisão de tensão do sinal recebido dos transdutores, convertendo esse sinal de corrente de 4 a 20 mA em 1 a 5 V. Assim, utilizando o conversor A/D com entrada de 1 a 5 V. O circuito eletrônico de ajuste manual da pressão principal possui uma divisão de tensão do regulador de 12 V, contendo uma resistência variável em série com uma fixa, assim, podendo variar a tensão de saída com o limite de 10 V, esse limite é necessário para ajustar o instrumento de regulação proporcional de pressão. E, por fim, o circuito eletrônico do sensor ultrassônico, que possui uma divisão de tensão do sinal recebido do sensor. Assim, diminuindo a tensão de saída do sensor de 5 V em 3,3 V. Esse circuito eletrônico é importante para proteção das entradas do *Raspberry Pi*.

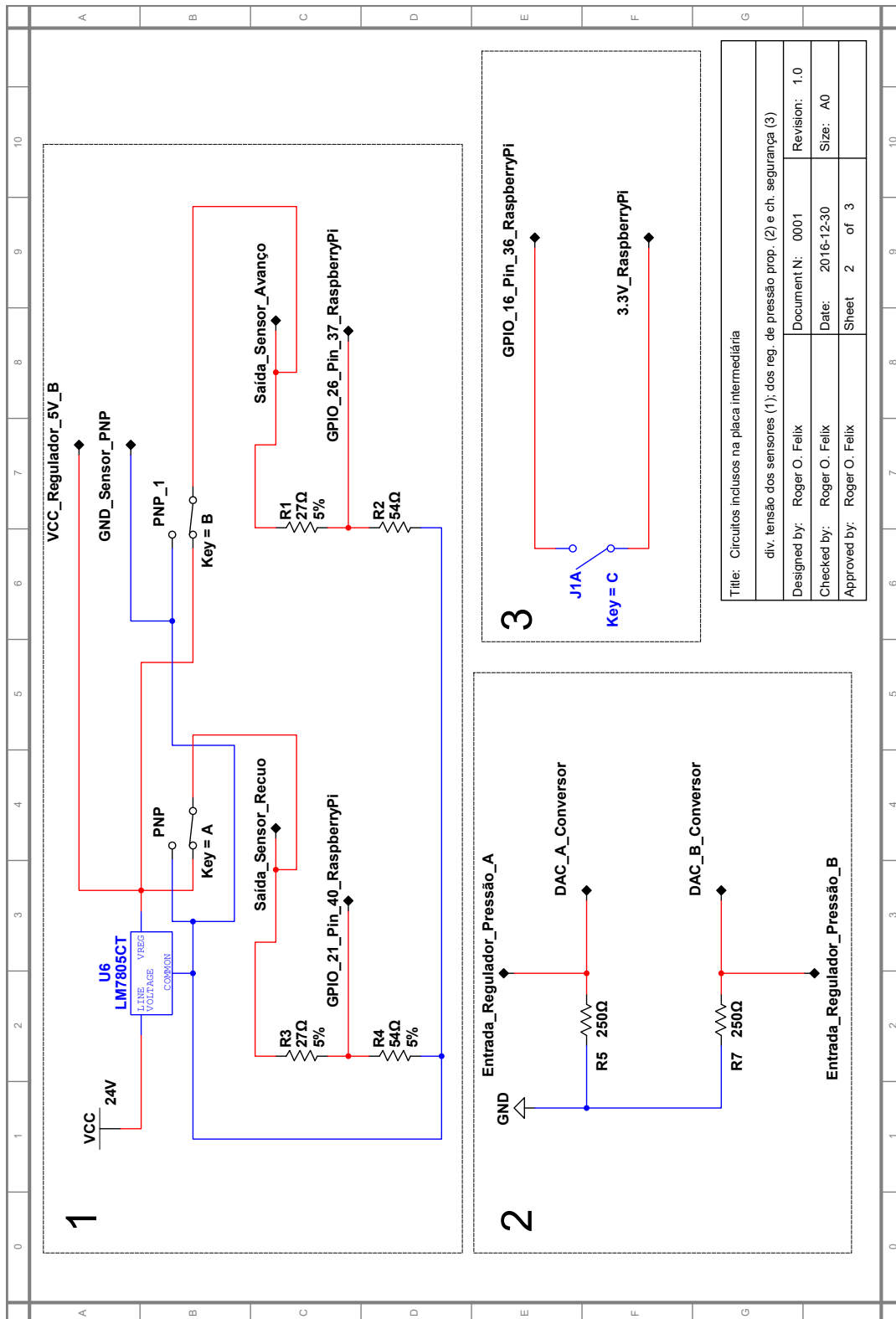
O apêndice A.4 mostra o diagrama pneumático utilizado para o desenvolvimento do equipamento experimental.

O apêndice A.5 mostra a lista de equipamentos para o desenvolvimento do equipamento experimental.

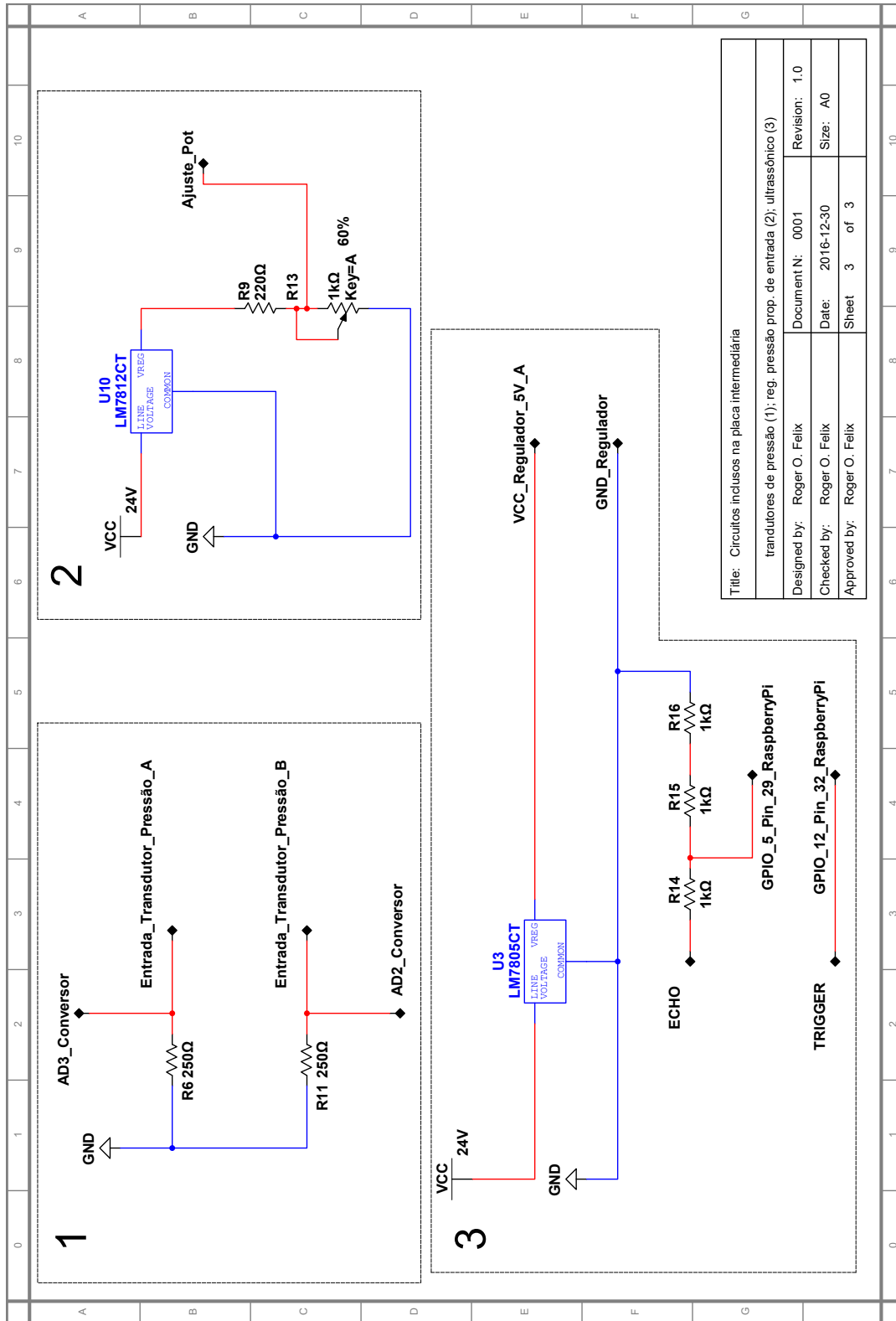
A.1 CONEXÕES ELÉTRICAS DE ENTRADA E SAÍDA



A.2 CIRCUITOS DA PLACA INTERMEDIÁRIA - 1



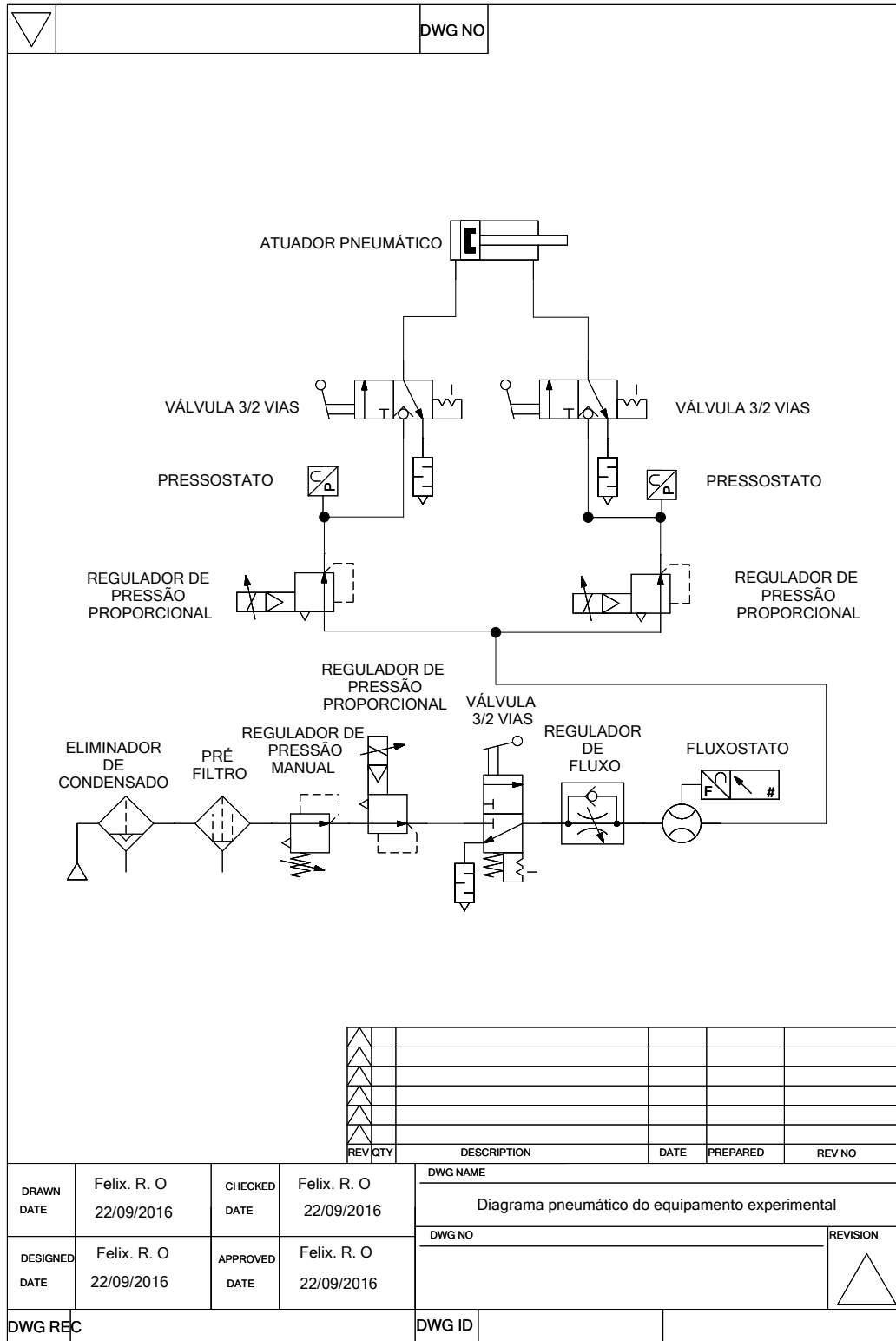
A.3 CIRCUITOS DA PLACA INTERMEDIÁRIA - 2



Title: Circuitos inclusos na placa intermediária

transdutores de pressão (1); reg. pressão prop. de entrada (2); ultrassônico (3)			
Designed by:	Roger O. Felix	Document N°:	0001
Checked by:	Roger O. Felix	Date:	2016-12-30
Approved by:	Roger O. Felix	Sheet	3 of 3

A.4 DIAGRAMA PNEUMÁTICO



A.5 LISTA DE EQUIPAMENTOS

Tabela A.1: Equipamentos pneumáticos e eletropneumáticos

Descrição	Modelo/Marca	Quantidade
Atuador pneumático	C85/SMC	1
Eliminador de condensado	AMG/SMC	1
Fluxostato	PF/SMC	1
Pré-filtro	AF/SMC	1
Regulador de pressão proporcional	ITV/SMC	3
Regulador de pressão manual	AR/SMC	1
Regulador de vazão manual	AS/SMC	1
Transdutor de pressão	IS/SMC	2
Válvula de fechamento manual	VH/SMC	3

Tabela A.2: Equipamentos eletrônicos

Descrição	Modelo/Marca	Quantidade
Fonte de alimentação 24V	MW	1
Raspberry Pi	Raspberry Pi 3	1
Sensor ultrassônico	Hc-sr04	1
Tela 7" touchscreen	Waveshare	1
Conversor AD/DA	Waveshare	1

APÊNDICE B

PROGRAMAÇÃO DO *Raspberry Pi* PARA OBTER OS RESULTADOS EXPERIMENTAIS

O Apêndice B.1 mostra a programação do controlador *Raspberry Pi*. Para aquisição de dados foi utilizada uma biblioteca em linguagem C, que foi fornecida pelo fabricante do conversor AD/DA. O fabricante do conversor é *Waveshare* e sua biblioteca está localizada no seguinte endereço: <http://www.waveshare.com>.

B.1 PROGRAMAÇÃO EM LINGUAGEM C

```
/*
*****
* Program for automation of the experimental rig      *
* Using Raspberry Pi and program in C                *
* MS Roger Felix - UNESP Bauru/SP                   *
* Date: 2017-01-25                                   *
*****
*/
#include <bcm2835.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <math.h>
#include <errno.h>
#include <gtk/gtk.h>
#include <glib.h>
#include <stdlib.h>
#include <gsl/gsl_fit.h>

/* Define variable */
#define SEN_AVANCO RPI_V2_GPIO_P1_37
#define SEN_RECUO RPI_V2_GPIO_P1_40
#define TRIG RPI_V2_GPIO_P1_32
#define ECHO RPI_V2_GPIO_P1_29
#define DRDY RPI_V2_GPIO_P1_11
#define RST RPI_V2_GPIO_P1_12
#define DRDY_IS_LOW() ((bcm2835_gpio_lev(DRDY)==0))
#define RST_1() bcm2835_gpio_write(RST,HIGH);
#define RST_0() bcm2835_gpio_write(RST,LOW);
#define G_MINDOUBLE DBL_MIN
#define G_MAXDOUBLE DBL_MAX

#define GNUPLOT "gnuplot -persist"

int a=-1;
int v=0;
int cont1[10]={1,1,1,1,1,1,1,1,1,1};
int cont=-1;
float volts1=0;
float volts2=0;
float bar1=0;

float bar2=0;
float test=0;
float ext[10]={0,0,0,0,0,0,0,0,0,0};
float compr[10]={0,0,0,0,0,0,0,0,0,0};

GtkWidget *main_window;
GtkWidget *window1;
GtkWidget *window2;
GtkWidget *window3;
GtkWidget *window4;
GtkWidget *window5;
GtkBuilder *gtkBuilder;

#define SPICS RPI_V2_GPIO_P1_15
#define SPICS1 RPI_V2_GPIO_P1_16
#define CS_1() bcm2835_gpio_write(SPICS,HIGH)
#define CS_0() bcm2835_gpio_write(SPICS,LOW)
#define CS1_1() bcm2835_gpio_write(SPICS1,HIGH)
#define CS1_0() bcm2835_gpio_write(SPICS1,LOW)
#define uint8_t unsigned char
#define uint16_t unsigned short
#define uint32_t unsigned long
#define channel_A 0x30
#define channel_B 0x34

GtkWidget *window1;
GtkWidget *window;
GtkBuilder *gtkBuilder;

typedef double gdouble;
typedef float gfloat;
float volts1=0;
float volts2=0;
float bar1=0;
float bar2=0;

typedef enum {false=0, true=!false} bool;
/* gain channel */
typedef enum
{
    ADS1256_GAIN_1 = (0),/*GAIN 1 */

```

```

    ADS1256_GAIN_2 = (1), /*GAIN 2 */
    ADS1256_GAIN_4 = (2), /*GAIN 4 */
    ADS1256_GAIN_8 = (3), /*GAIN 8 */
    ADS1256_GAIN_16 = (4), /*GAIN 16 */
    ADS1256_GAIN_32 = (5), /*GAIN 32 */
    ADS1256_GAIN_64 = (6), /*GAIN 64 */
} ADS1256_GAIN_E;

typedef enum
{
    ADS1256_3000SPS = 0,
    ADS1256_1500SPS,
    ADS1256_750SPS,
    ADS1256_375SPS,
    ADS1256_2000SPS,
    ADS1256_1000SPS,
    ADS1256_500SPS,
    ADS1256_100SPS,
    ADS1256_60SPS,
    ADS1256_50SPS,
    ADS1256_30SPS,
    ADS1256_25SPS,
    ADS1256_15SPS,
    ADS1256_10SPS,
    ADS1256_5SPS,
    ADS1256_2d5SPS,
    ADS1256_DRATE_MAX
} ADS1256_DRATE_E;

#define ADS1256_DRAE_COUNT = 15;

typedef struct
{
    ADS1256_GAIN_E Gain; /* GAIN */
    ADS1256_DRATE_E DataRate; /* DATA output speed*/
    int32_t AdcNow[8]; /* ADC Conversion value */
    uint8_t Channel; /* The current channel*/
    uint8_t ScanMode;
} ADS1256_VAR_T;

/*Register Map --- ADS1256 datasheet Page 30*/
enum
{
    /*Register address, followed by reset the default values */
    REG_STATUS = 0, // x1H
    REG_MUX = 1, // 01H
    REG_ADCON = 2, // 20H
    REG_DRATE = 3, // FOH
    REG_IO = 4, // EOH
    REG_OFPC0 = 5, // xxH
    REG_OFPC1 = 6, // xxH
    REG_OFPC2 = 7, // xxH
    REG_FSCO = 8, // xxH
    REG_FSC1 = 9, // xxH
    REG_FSC2 = 10, // xxH
};

/* Command Definitions --- ADS1256 datasheet Page 34 */
enum
{
    CMD_WAKEUP = 0x00,
    // Completes SYNC and Exits Standby Mode 0000 0000 (00h)
    CMD_RDATA = 0x01,
    // Read Data 0000 0001 (01h)
    CMD_RDATAC = 0x03,
    // Read Data Continuously 0000 0011 (03h)
    CMD_SDATAC = 0x0F,
    // Stop Read Data Continuously 0000 1111 (0Fh)
    CMD_RREG = 0x10,
    // Read from REG rrr 0001 rrrr (1xh)
    CMD_WREG = 0x50,
    // Write to REG rrr 0101 rrrr (5xh)
    CMD_SELFCAL = 0xF0,
    // Offset and Gain Self-Calibration 1111 0000 (F0h)
    CMD_SELFOCAL = 0xF1,
    // Offset Self-Calibration 1111 0001 (F1h)
    CMD_SELFGCAL = 0xF2,
    // Gain Self-Calibration 1111 0010 (F2h)
    CMD_SYSCCAL = 0xF3,
    // System Offset Calibration 1111 0011 (F3h)
    CMD_SYSGCAL = 0xF4,
    // System Gain Calibration 1111 0100 (F4h)
    CMD_SYNC = 0xFC,
    // Synchronize the A/D Conversion 1111 1100 (FCh)
    CMD_STANDBY = 0xFD,
    // Begin Standby Mode 1111 1101 (FDh)
    CMD_RESET = 0xFE,
    // Reset to Power-Up Values 1111 1110 (FEh)
};

ADS1256_VAR_T g_tADS1256;
static const uint8_t s_tabDataRate[ADS1256_DRATE_MAX] =
{
    0xF0, /*reset the default values */
    0xE0,
    0xD0,
    0xC0,
    0xB0,
    0xA1,
    0x92,
    0x82,
    0x72,
    0x63,
    0x53,
    0x43,
    0x33,
    0x20,
    0x13,
    0x03
};

void bsp_DelayUS(uint64_t micros);
void Write_DAC8532(uint8_t channel, uint16_t Data);
uint16_t Voltage_Convert(float Vref, float voltage);
void ADS1256_StartScan(uint8_t _ucScanMode);
static void ADS1256_Send8Bit(uint8_t _data);
void ADS1256_CfgADC(ADS1256_GAIN_E _gain, ADS1256_DRATE_E _drate);
static void ADS1256_DelayDATA(void);
static uint8_t ADS1256_Receive8Bit(void);
static void ADS1256_WriteReg(uint8_t _RegID, uint8_t _RegValue);
static uint8_t ADS1256_ReadReg(uint8_t _RegID);
static void ADS1256_WriteCmd(uint8_t _cmd);
uint8_t ADS1256_ReadChipID(void);
static void ADS1256_SetChannel(uint8_t _ch);
static void ADS1256_SetDiffChannel(uint8_t _ch);
static void ADS1256_WaitDRDY(void);
static int32_t ADS1256_ReadData(void);
int32_t ADS1256_GetAdc(uint8_t _ch);
void ADS1256_ISR(void);
uint8_t ADS1256_Scan(void);
void Write_DAC8552(uint8_t channel, uint16_t Data);
uint16_t Voltage_Convert(float Vref, float voltage);

void bsp_DelayUS(uint64_t micros)
{
    bcm2835_delayMicroseconds (micros);
}

/*
*****
* name: bsp_InitADS1256
* function: Configuration of the STM32 GPIO and SPI
* parameter: NULL
* The return value: NULL
*****
*/

void bsp_InitADS1256(void)
{
#ifdef SOFT_SPI
    CS_1();
    SCK_0();
    DI_0();
#endif
}

/*
*****
* name: ADS1256_StartScan
* function: Configuration DRDY PIN for external interrupt
* parameter: _DiffMode : 0 Single-ended input 8 channel
* 1 Differential input 4 channel
* the return value: NULL
*****
*/

void ADS1256_StartScan(uint8_t _ucScanMode)
{
    g_tADS1256.ScanMode = _ucScanMode;
    {
        uint8_t i;

        g_tADS1256.Channel = 0;

        for (i = 0; i < 8; i++)
        {
            g_tADS1256.AdcNow[i] = 0;
        }
    }
}

/*
*****
* name: ADS1256_Send8Bit
* function: SPI bus to send 8 bit data
* parameter: _data: data
* the return value: NULL
*****
*/

static void ADS1256_Send8Bit(uint8_t _data)
{
    bsp_DelayUS(2);
    bcm2835_spi_transfer(_data);
}

/*
*****
* name: ADS1256_CfgADC
* function: The configuration parameters of ADC
* parameter: _gain:gain 1-64 _drate: data rate
* the return value: NULL
*****
*/

void ADS1256_CfgADC(ADS1256_GAIN_E _gain, ADS1256_DRATE_E _drate)
{
    g_tADS1256.Gain = _gain;
    g_tADS1256.DataRate = _drate;
    ADS1256_WaitDRDY();
}

```

```

{
    * Storage ads1256 register configuration parameters */
    uint8_t buf[4]; /

    /*
    Bit 3 ORDER: Data Output Bit Order
    0 = Most Significant Bit First (default)
    1 = Least Significant Bit First

    Bit 2 ACAL : Auto-Calibration
    0 = Auto-Calibration Disabled (default)
    1 = Auto-Calibration Enabled

    Bit 1 BUFEN: Analog Input Buffer Enable
    0 = Buffer Disabled (default)
    1 = Buffer Enabled

    Bit 0 DRDY : Data Ready (Read Only)
    This bit duplicates the state of the DRDY pin.

    ACAL=1 enable calibration
    */
    //enable the internal buffer
    //buf[0] = (0 << 3) | (1 << 2) | (1 << 1);
    //buffer is prohibited
    buf[0] = (0 << 3) | (1 << 2) | (0 << 1);

    //ADS1256_WriteReg...
    (REG_STATUS, (0 << 3) | (1 << 2) | (1 << 1));

    buf[1] = 0x08;

    /* ADCON: A/D Control Register (Address 02h)
    Bit 7 Reserved, always 0 (Read Only)
    Bits 6-5 CLK1, CLK0 : DO/CLKOUT Clock Out Rate Setting
    00 = Clock Out OFF
    01 = Clock Out Frequency = fCLKIN (default)
    10 = Clock Out Frequency = fCLKIN/2
    11 = Clock Out Frequency = fCLKIN/4

    Bits 4-3 SDCS1, SDCS0: Sensor Detect Current Sources
    00 = Sensor Detect OFF (default)
    01 = Sensor Detect Current = 0.5 A
    10 = Sensor Detect Current = 2 A
    11 = Sensor Detect Current = 10 A

    Bits 2-0 PGA2, PGA1, PGA0:
    Programmable Gain Amplifier Setting
    000 = 1 (default)
    001 = 2
    010 = 4
    011 = 8
    100 = 16
    101 = 32
    110 = 64
    111 = 64
    */
    buf[2] = (0 << 5) | (0 << 3) | (_gain << 0);
    buf[3] = s_tabDataRate[_drate]; // DRATE_10SPS;
    CS_0();
    ADS1256_Send8Bit(CMD_WREG | 0);
    /* Write and send the register address */
    ADS1256_Send8Bit(0x03);
    /* Register number 4, Initialize -1*/
    ADS1256_Send8Bit(buf[0]);
    /* Set the status register */
    ADS1256_Send8Bit(buf[1]);
    /* Set the input channel parameters */
    ADS1256_Send8Bit(buf[2]);
    /* Set the ADCON control register, gain */
    ADS1256_Send8Bit(buf[3]);
    /* Set the output rate */
    CS_1(); /* SPI cs = 1 */
}
    bsp_DelayUS(50);
}

/*
*****
* name: ADS1256_DelayDATA
* function: delay
* parameter: NULL
* the return value: NULL
*****
*/
static void ADS1256_DelayDATA(void)
{
    bsp_DelayUS(10); /* The minimum time delay 6.5us */
}

/*
*****
* name: ADS1256_Recive8Bit
* function: SPI bus receive function
* parameter: NULL
* the return value: NULL
*****
*/
static uint8_t ADS1256_Recive8Bit(void)
{
    uint8_t read = 0;
    read = bcm2835_spi_transfer(0xff);
    return read;
}

}

/*
*****
* name: ADS1256_WriteReg
* function: Write the corresponding register
* parameter: _RegID: register ID
* regValue: register Value
* the return value: NULL
*****
*/
static void ADS1256_WriteReg(uint8_t _RegID, uint8_t _RegValue)
{
    CS_0(); /* SPI cs = 0 */
    ADS1256_Send8Bit(CMD_WREG | _RegID);
    /*Write command register */
    ADS1256_Send8Bit(0x00);
    /*Write the register number */
    ADS1256_Send8Bit(_RegValue);
    /*send register value */
    CS_1(); /* SPI cs = 1 */
}

/*
*****
* name: ADS1256_ReadReg
* function: Read the corresponding register
* parameter: _RegID: register ID
* the return value: read register value
*****
*/
static uint8_t ADS1256_ReadReg(uint8_t _RegID)
{
    uint8_t read;
    CS_0();
    /* SPI cs = 0 */
    ADS1256_Send8Bit(CMD_RREG | _RegID);
    /* Write command register */
    ADS1256_Send8Bit(0x00);
    /* Write the register number */
    ADS1256_DelayDATA();
    /*delay time */
    read = ADS1256_Recive8Bit();
    /* Read the register values */
    CS_1(); /* SPI cs = 1 */
    return read;
}

/*
*****
* name: ADS1256_WriteCmd
* function: Sending a single byte order
* parameter: _cmd : command
* the return value: NULL
*****
*/
static void ADS1256_WriteCmd(uint8_t _cmd)
{
    CS_0(); /* SPI cs = 0 */
    ADS1256_Send8Bit(_cmd);
    CS_1(); /* SPI cs = 1 */
}

/*
*****
* name: ADS1256_ReadChipID
* function: Read the chip ID
* parameter: _cmd : NULL
* the return value: four high status register
*****
*/
uint8_t ADS1256_ReadChipID(void)
{
    uint8_t id;
    ADS1256_WaitDRDY();
    id = ADS1256_ReadReg(REG_STATUS);
    return (id >> 4);
}

/*
*****
* name: ADS1256_SetChannal
* function: Configuration channel number
* parameter: _ch: channel number 0--7
* the return value: NULL
*****
*/
static void ADS1256_SetChannal(uint8_t _ch)
{
    /*
    Bits 7-4 PSEL3, PSEL2, PSEL1, PSEL0:
    Positive Input Channel (AINP) Select
    0000 = AINO (default)
    0001 = AIN1
    0010 = AIN2 (ADS1256 only)
    0011 = AIN3 (ADS1256 only)
    0100 = AIN4 (ADS1256 only)
    0101 = AIN5 (ADS1256 only)
    0110 = AIN6 (ADS1256 only)
    0111 = AIN7 (ADS1256 only)
    1xxx = AINCOM

    Bits 3-0 NSEL3, NSEL2, NSEL1, NSEL0:
    Negative Input Channel (AINN)Select
    */
}

```

```

0000 = AINO
0001 = AIN1 (default)
0010 = AIN2 (ADS1256 only)
0011 = AIN3 (ADS1256 only)
0100 = AIN4 (ADS1256 only)
0101 = AIN5 (ADS1256 only)
0110 = AIN6 (ADS1256 only)
0111 = AIN7 (ADS1256 only)
1xxx = AINCOM
*/
if (_ch > 7)
{
    return;
}
ADS1256_WriteReg(REG_MUX, (_ch << 4) | (1 << 3));
/* Bit3 = 1 */
}

/*
*****
* name: ADS1256_SetDiffChannel
* function: The configuration difference channel
* parameter: _ch: channel number 0--3
* the return value: four high status register
*****
*/
static void ADS1256_SetDiffChannel(uint8_t _ch)
{
    /*
    Bits 7-4 PSEL3, PSEL2, PSEL1, PSEL0:
    Positive Input Channel (AINP) Select
    0000 = AINO (default)
    0001 = AIN1
    0010 = AIN2 (ADS1256 only)
    0011 = AIN3 (ADS1256 only)
    0100 = AIN4 (ADS1256 only)
    0101 = AIN5 (ADS1256 only)
    0110 = AIN6 (ADS1256 only)
    0111 = AIN7 (ADS1256 only)
    1xxx = AINCOM

    Bits 3-0 NSEL3, NSEL2, NSEL1, NSEL0:
    Negative Input Channel (AINN)Select
    0000 = AINO
    0001 = AIN1 (default)
    0010 = AIN2 (ADS1256 only)
    0011 = AIN3 (ADS1256 only)
    0100 = AIN4 (ADS1256 only)
    0101 = AIN5 (ADS1256 only)
    0110 = AIN6 (ADS1256 only)
    0111 = AIN7 (ADS1256 only)
    1xxx = AINCOM
    */
    if (_ch == 0)
    {
        ADS1256_WriteReg(REG_MUX, (0 << 4) | 1);
    }
    else if (_ch == 1)
    {
        ADS1256_WriteReg(REG_MUX, (2 << 4) | 3);
    }
    else if (_ch == 2)
    {
        ADS1256_WriteReg(REG_MUX, (4 << 4) | 5);
    }
    else if (_ch == 3)
    {
        ADS1256_WriteReg(REG_MUX, (6 << 4) | 7);
    }
}

/*
*****
* name: ADS1256_WaitDRDY
* function: delay time wait for automatic calibration
* parameter: NULL
* the return value: NULL
*****
*/
static void ADS1256_WaitDRDY(void)
{
    uint32_t i;

    for (i = 0; i < 400000; i++)
    {
        if (DRDY_IS_LOW())
        {
            break;
        }
    }
    if (i >= 400000)
    {
        printf("ADS1256_WaitDRDY() Time Out ...r\n");
    }
}

/*
*****
* name: ADS1256_ReadData
* function: read ADC value
* parameter: NULL
* the return value: NULL
*****
*/

```

```

static int32_t ADS1256_ReadData(void)
{
    uint32_t read = 0;
    static uint8_t buf[3];

    CS_0(); /* SPI cs = 0 */

    ADS1256_Send8Bit(CMD_RDATA); /* read ADC command */

    ADS1256_DelayDATA(); /*delay time */

    /*Read the sample results 24bit*/
    buf[0] = ADS1256_Recive8Bit();
    buf[1] = ADS1256_Recive8Bit();
    buf[2] = ADS1256_Recive8Bit();

    read = ((uint32_t)buf[0] << 16) & 0x00FF0000;
    read |= ((uint32_t)buf[1] << 8);
    read |= buf[2];

    CS_1();

    /* Extend a signed number*/
    if (read & 0x800000)
    {
        read |= 0xFF000000;
    }

    return (int32_t)read;
}

/*
*****
* name: ADS1256_GetAdc
* function: read ADC value
* parameter: channel number 0--7
* the return value: ADC vaule (signed number)
*****
*/
int32_t ADS1256_GetAdc(uint8_t _ch)
{
    int32_t iTemp;
    if (_ch > 7)
    {
        return 0;
    }

    iTemp = g_tADS1256.AdcNow[_ch];

    return iTemp;
}

/*
*****
* name: ADS1256_ISR
* function: Collection procedures
* parameter: NULL
* the return value: NULL
*****
*/
void ADS1256_ISR(void)
{
    if (g_tADS1256.ScanMode == 0)
    {
        ADS1256_SetChannel(g_tADS1256.Channel);
        bsp_DelayUS(5);
        ADS1256_WriteCmd(CMD_SYNC);
        bsp_DelayUS(5);
        ADS1256_WriteCmd(CMD_WAKEUP);
        bsp_DelayUS(25);
        if (g_tADS1256.Channel == 0)
        {
            g_tADS1256.AdcNow[7] = ADS1256_ReadData();
        }
        else
        {
            g_tADS1256.AdcNow[g_tADS1256.Channel-1] = ADS1256_ReadData();
        }
        if (++g_tADS1256.Channel >= 8)
        {
            g_tADS1256.Channel = 0;
        }
    }
    else /*DiffChannel*/
    {
        ADS1256_SetDiffChannel(g_tADS1256.Channel);
        bsp_DelayUS(5);
        ADS1256_WriteCmd(CMD_SYNC);
        bsp_DelayUS(5);
        ADS1256_WriteCmd(CMD_WAKEUP);
        bsp_DelayUS(25);
        if (g_tADS1256.Channel == 0)
        {
            g_tADS1256.AdcNow[3] = ADS1256_ReadData();
        }
        else
        {
            g_tADS1256.AdcNow[g_tADS1256.Channel-1] = ADS1256_ReadData();
        }
        if (++g_tADS1256.Channel >= 4)
        {
            g_tADS1256.Channel = 0;
        }
    }
}

```

```

}

/*
*****
* name: ADS1256_Scan
* function:
* parameter: NULL
* the return value: 1
*****
*/
uint8_t ADS1256_Scan(void)
{
    if (DRDY_IS_LOW())
    {
        ADS1256_ISR();
        return 1;
    }
    return 0;
}

/*
*****
* name: Write_DAC8532
* function: DAC send data
* parameter: channel : output channel number
* data : output DAC value
* the return value: NULL
*****
*/
void Write_DAC8532(uint8_t channel, uint16_t Data)
{
    CS1_1();
    CS1_0();
    bcm2835_spi_transfer(channel);
    bcm2835_spi_transfer((Data>>8));
    bcm2835_spi_transfer((Data&0xff));
    CS1_1();
}

/*
*****
* name: Voltage_Convert
* function: Voltage value conversion function
* parameter: Vref : The reference voltage 3.3V or 5V
* voltage : output DAC value
* the return value: NULL
*****
*/
uint16_t Voltage_Convert(float Vref, float voltage)
{
    uint16_t _D;
    _D = (uint16_t)(65536 * voltage / Vref);
    return _D;
}

/*
*****
* name: Program of friction test
*****
*/

void on_main_window_destroy()
{
    gtk_main_quit();
}

void on_window3_delete_event()
{
    gtk_widget_hide_on_delete(window3);
}

void on_window1_delete_event()
{
    gtk_widget_hide_on_delete(window1);
}

void on_window2_delete_event()
{
    gtk_widget_hide_on_delete(window2);
}

void on_window4_delete_event()
{
    gtk_widget_hide_on_delete(window4);
}

void on_window5_delete_event()
{
    gtk_widget_hide_on_delete(window5);
}

void on_button1_clicked(GtkButton *button1)
{
    gtk_widget_show(window3);
}

void on_adjustment1_value_changed(GtkAdjustment *adjustment1)
{
    volts1 = gtk_adjustment_get_value(adjustment1);
    volts1= (0.1954*bar1)-0.3265;
    printf("%f\n",volts1);
}

void on_adjustment2_value_changed(GtkAdjustment *adjustment2)
{
    bar2 = gtk_adjustment_get_value(adjustment2);
    volts2= (0.1954*bar1)-0.3265;
    printf("%f\n",volts2);
}

void on_button2_clicked(GtkButton *button2)
{
    gtk_widget_hide(window1);
    gtk_widget_hide(window2);
    gtk_widget_hide(window3);
}

void on_checkbutton5_toggled(GtkToggleButton *toggle_button5)
{
    cont1[0]=cont1[0]+1;
    if (gtk_toggle_button_get_active(toggle_button5)==TRUE)
    {
        cont=cont+1;
        ext[cont]=2.45;
        compr[cont]=2.45;
        printf("Tensao A %f\n",ext[cont]);
        printf("Tensao B %f\n",compr[cont]);
    }
    else
    {
        cont=cont-1;
        printf("Tensao A \n");
        printf("Tensao B \n");
    }
}

void on_checkbutton1_toggled(GtkToggleButton *toggle_button1)
{
    cont1[1]=cont1[1]+1;
    if (gtk_toggle_button_get_active(toggle_button1)==TRUE)
    {
        cont=cont+1;
        ext[cont]=2.50;
        compr[cont]=2.50;
        printf("Tensao A %f\n",ext[cont]);
        printf("Tensao B %f\n",compr[cont]);
    }
    else
    {
        cont=cont-1;
        printf("Tensao A \n");
        printf("Tensao B \n");
    }
}

void on_checkbutton6_toggled(GtkToggleButton *toggle_button6)
{
    cont1[2]=cont1[2]+1;
    if (gtk_toggle_button_get_active(toggle_button6)==TRUE)
    {
        cont=cont+1;
        ext[cont]=2.55;
        compr[cont]=2.55;
        printf("Tensao A %f\n",ext[cont]);
        printf("Tensao B %f\n",compr[cont]);
    }
    else
    {
        cont=cont-1;
        printf("Tensao A \n");
        printf("Tensao B \n");
    }
}

void on_checkbutton2_toggled(GtkToggleButton *toggle_button2)
{
    cont1[3]=cont1[3]+1;
    if (gtk_toggle_button_get_active(toggle_button2)==TRUE)
    {
        cont=cont+1;
        ext[cont]=2.60;
        compr[cont]=2.60;
        printf("Tensao A %f\n",ext[cont]);
        printf("Tensao B %f\n",compr[cont]);
    }
    else
    {
        cont=cont-1;
        printf("Tensao A \n");
        printf("Tensao B \n");
    }
}

void on_checkbutton3_toggled(GtkToggleButton *toggle_button3)
{
    cont1[4]=cont1[4]+1;
    if (gtk_toggle_button_get_active(toggle_button3)==TRUE)
    {
        cont=cont+1;
        ext[cont]=2.67;
        compr[cont]=2.67;
        printf("Tensao A %f\n",ext[cont]);
        printf("Tensao B %f\n",compr[cont]);
    }
    else
    {
        cont=cont-1;
        printf("Tensao A \n");
        printf("Tensao B \n");
    }
}

```



```

bcm2835_gpio_set_pud(DRDY, BCM2835_GPIO_PUD_UP);

id = ADS1256_ReadChipID();
printf("\r\n");
printf("ID=\r\n");

if (id != 3)
{
printf("Error, ASD1256 Chip ID = 0x%d\r\n", (int)id);
}
else
{
printf("Ok, ASD1256 Chip ID = 0x%d\r\n", (int)id);
}

ADS1256_CfgADC(ADS1256_GAIN_1, ADS1256_10SPS);
ADS1256_StartScan(0);

for(f=0;f<20;f++)
{
while((ADS1256_Scan() == 0));
adc1 = ADS1256_GetAdc(5);
adc2 = ADS1256_GetAdc(6);
}

inicio_tempo_avanco = bcm2835_st_read();
Write_DAC8532(0x34, Voltage_Convert(5.0,(float)volts1));
//Write channel A buffer (0x34)
Write_DAC8532(0x30, Voltage_Convert(5.0,(float)0.0));
//Write channel A buffer (0x30)
while(bcm2835_gpio_lev(SEN_AVANCO) == LOW)
{
i++;
D[i]=getCM();
tempo_avanco[i]=(bcm2835_st_read()-...
inicio_tempo_avanco)/1000000;
while((ADS1256_Scan() == 0));
adc1 = ADS1256_GetAdc(5);
volt1 = (adc1 * 100.0) / 167.0;
iTemp1[i] = volt1;
printf("Distance: %f\n", D[i]);
}
Write_DAC8532(0x34, Voltage_Convert(5.0,(float)0.0));
//Write channel A buffer (0x34)
bsp_DelayUS(5000000);
inicio_tempo_recuo = bcm2835_st_read();
Write_DAC8532(0x30, Voltage_Convert(5.0,(float)volts2));
//Write channel A buffer (0x30)
while(bcm2835_gpio_lev(SEN_RECUCO) == LOW)
{
w++;
F[w]=getCM();
tempo_recuo[w]=(bcm2835_st_read()-...
inicio_tempo_recuo)/1000000;
while((ADS1256_Scan() == 0));
adc2 = ADS1256_GetAdc(6);
volt2 = (adc2 * 100.0) / 167.0;
iTemp2[w] = volt2;
printf("Distance: %f\n", F[w]);
}
bsp_DelayUS(300);
Write_DAC8532(0x30, Voltage_Convert(5.0,(float)0.0));
//Write channel A buffer (0x30)
bcm2835_spi_end();
bcm2835_close();

for(d=0;d<=i;d++)
{
if(D[d]>=10.0 && D[d]<=25.0)
{
c++;
S[c]=D[d];
te[c]=tempo_avanco[d];
iTemp1n[c]=iTemp1[d];
if (te[c]< smate)
{
smate=te[c];
}
if (te[c]> bigte)
{
bigte=te[c];
}
}
}

for(d=0;d<=w;d++)
{
if(F[d]>=10.0 && F[d]<=25.0)
{
g++;
G[g]=F[d];
tc[g]=tempo_recuo[d];
iTemp2n[g]=iTemp2[d];
if (tc[g]< smatc)
{
smatc=tc[g];
}
if (tc[g]> bigtc)
{
bigtc=tc[g];
}
}
}

gsl_fit_linear (te, 1, S, 1, c, &c0,...
&c1, &cov00, &cov01, &cov11, &sumsq0);
printf ("# best fit extension displacement:...
Y = %g + %g X\n", c0, c1);
printf ("# sumsq0 = %g\n", sumsq0);
printf ("\n");

gsl_fit_linear (tc, 1, G, 1, g, &c2,...
&c3, &cov100, &cov101, &cov111, &sumsq1);
printf ("# best fit compression displacement:...
Y = %g + %g X\n", c2, c3);
printf ("# sumsq1= %g\n", sumsq1);
printf ("\n");

gsl_fit_linear (te, 1, iTemp1n, 1, c,...
&c4, &c5, &cov200, &cov201, &cov211, &sumsq2);
printf ("# best fit extension pressure:...
Y = %g + %g X\n", c4, c5);
printf ("# sumsq2 = %g\n", sumsq2);
printf ("\n");

gsl_fit_linear (tc, 1, iTemp2n, 1, g,...
&c6, &c7, &cov300, &cov301, &cov311, &sumsq3);
printf ("# best fit compression pressure:...
Y = %g + %g X\n", c6, c7);
printf ("# sumsq3= %g\n", sumsq3);
printf ("\n");

for(p=0;p<=n-1;p++)
{
tef0[p]=((bigte-smate)/(n))*p+smate;
Disp0[p]=(c1*tef0[p])+c0;
Pres0[p]=c4;
}
for(p=0;p<=n-1;p++)
{
tef1[p]=((bigtc-smatc)/(n))*p+smatc;
Disp1[p]=(c3*tef1[p])+c2;
Pres1[p]=c6;
}

FILE *filem1;
filem1=fopen("manual_extension.txt","w");
fprintf(filem1,"%s\n","Distância do Avanço...
<----> Tempo do Avanço <----> Tensão do Avanço");
int j=0;
for(j=0;j<=c;j++)
{
fprintf(filem1,"%f %f %f \n",S[j],te[j], iTemp1n[j]);
}
fclose(filem1);

FILE *filem2;
filem2=fopen("manual_compression.txt","w");
fprintf(filem2,"%s\n","Distância do Recuo...
<---->Tempo do Recuo <----> Tensão de Recuo");
int s=0;
for(s=0;s<=g;s++)
{
fprintf(filem2,"%f %f %f \n", G[s], tc[s],...
iTemp2n[s]);
}
fclose(filem2);

FILE *filem3;
filem3=fopen("fit_manual_extension.txt","w");
fprintf(filem3,"%s\n","Distância do Avanço...
<----> Tempo do Avanço <----> Pressao do Avanço");
for(j=0;j<=n-1;j++)
{
fprintf(filem3,"%f %f %f \n", Disp0[j],...
tef0[j], Pres0[j]);
}
fclose(filem3);

FILE *filem4;
filem4=fopen("fit_manual_compression.txt","w");
fprintf(filem4,"%s\n","Distância do Recuo...
<---->Tempo do Recuo <----> Pressao de Recuo");
for(s=0;s<=n-1;s++)
{
fprintf(filem4,"%f %f %f \n", Disp1[s],...
tef1[s], Pres1[s]);
}
fclose(filem4);

system("gnuplot -p -e \load 'scriptmanual.gnu'\");

}

if (test==2)
{
// if none test choose
if (cont==-1)
{
gtk_widget_show(window4);
return;
}

// variables for automatic test
uint8_t id;
int32_t adc1=0; int32_t adc2=0;
int32_t volt1=0; int32_t volt2=0;

```

```

int32_t iTemp1[10000][10]={0};
int32_t iTemp2[10000][10]={0};
int tec=0; int q=0; int f=0; int j=0;
int d=0; int at=0;
int n=-1; int i=-1; int w=-1; int c;
int g; int N=500;
int ci[10]={0}; int cw[10]={0}; int p=0;
int ncont[10]={0};
float D[1000][10]={0}; float F[1000][10]={0};
float inicio_tempo_avanco; float inicio_tempo_recuo;
float tempo_avanco[1000][10]={0};
float tempo_recuo[1000][10]={0};
float bigtc; float smate; float bigte; float smatc;
float bigG; float smaG; float bigS; float smaS;
float Vel0[10]={0}; float Vel1[10]={0};
float Ff0[10]={0}; float Ff1[10]={0};
double G[1000]={0};
double S[1000]={0};
double c0, c1, c2, c3, c4, c5, c6, c7;
double sumsq0, sumsq1, sumsq2, sumsq3;
double cov00, cov01, cov11, cov100;
double cov101, cov111, cov200, cov201;
double cov211, cov300, cov301, cov311;
double te[1000]={0};
double tc[1000]={0};
double iTempin[10000]={0};
double iTemp2n[10000]={0};
float tef0[501]={0};
float tef1[501]={0};
float Disp0[501]={0};
float Displ[501]={0};
float Pres0[501]={0};
float Pres1[501]={0};

char LE[10][20]={"avanco_015.txt","avanco_016.txt",...
"avanco_018.txt","avanco_020.txt",...
"avanco_021.txt","avanco_023.txt",...
"avanco_025.txt","avanco_026.txt",...
"avanco_030.txt"};
char LC[10][20]={"recuo_015.txt","recuo_016.txt",...
"recuo_018.txt","recuo_020.txt",...
"recuo_021.txt","recuo_023.txt",...
"recuo_025.txt","recuo_026.txt","recuo_030.txt"};
char fileAE[10][20]={"file1","file2","file3",...
"file4","file5","file6","file7",...
"file8","file9","file10"};
char fileAC[10][20]={"file11","file12","file13",...
"file14","file15","file16",...
"file17","file18","file19","file20"};
char FLE[10][20]={"fit_avanco_015.txt",...
"fit_avanco_016.txt","fit_avanco_018.txt",...
"fit_avanco_020.txt","fit_avanco_021.txt",...
"fit_avanco_023.txt","fit_avanco_025.txt",...
"fit_avanco_026.txt","fit_avanco_030.txt"};
char FLC[10][20]={"fit_recuo_015.txt",...
"fit_recuo_016.txt","fit_recuo_018.txt",...
"fit_recuo_020.txt","fit_recuo_021.txt",...
"fit_recuo_023.txt","fit_recuo_025.txt",...
"fit_recuo_026.txt","fit_recuo_030.txt"};
char fileFAE[10][20]={"file21","file22","file23",...
"file2","file25","file26","file27",...
"file28","file29","file30"};
char fileFAC[10][20]={"file31","file32","file33",...
"file34","file35","file36","file37",...
"file38","file39","file40"};

// start library bcm2835
bcm2835_init();
bcm2835_spi_begin();
bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_LSBFIRST);
bcm2835_spi_setDataMode(BCM2835_SPI_MODE1);
bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_8192);
bcm2835_gpio_fsel(SPICS, BCM2835_GPIO_FSEL_OUTP);
bcm2835_gpio_write(SPICS, HIGH);
bcm2835_gpio_fsel(SPICS1, BCM2835_GPIO_FSEL_OUTP);
bcm2835_gpio_write(SPICS1, HIGH);
bcm2835_gpio_fsel(TRIG, BCM2835_GPIO_FSEL_OUTP);
bcm2835_gpio_fsel(ECHO, BCM2835_GPIO_FSEL_INPT);
bcm2835_gpio_fsel(SEN_RECUCU, BCM2835_GPIO_FSEL_INPT);
bcm2835_gpio_fsel(SEN_AVANCO, BCM2835_GPIO_FSEL_INPT);
bcm2835_gpio_fsel(DRDY, BCM2835_GPIO_FSEL_INPT);
bcm2835_gpio_set_pud(DRDY, BCM2835_GPIO_PUD_UP);

// check if ADC it is OK
id = ADS1256_ReadChipID();
printf("\r\n");
printf("ID=\r\n");
if (id != 3)
{
printf("Error, ASD1256 Chip ID = 0x%d\r\n", (int)id);
}
else
{
printf("Ok, ASD1256 Chip ID = 0x%d\r\n", (int)id);
}

// configure gain and sample per second
ADS1256_CfgADC(ADS1256_GAIN_1, ADS1256_10SPS);
ADS1256_StartScan();

// loop for verify ADC
for(f=0;f<20;f++)
{
while((ADS1256_Scan() == 0));

adc1 = ADS1256_GetAdc(5);
adc2 = ADS1256_GetAdc(6);
}

// start automatic test
for (tec=0;tec<=cont;tec++)
{
// start for extension
i=-1;
w=-1;
bsp_DelayUS(3000000);
inicio_tempo_avanco = bcm2835_st_read();
//Write channel A buffer (0x34)
Write_DAC8532(0x34, Voltage_Convert(5.0, (float)ext[tec]));
//Write channel A buffer (0x30)
Write_DAC8532(0x30, Voltage_Convert(5.0, (float)0.0));
while(bcm2835_gpio_lev(SEN_AVANCO) == LOW)
{
ci[tec]=i++;
D[i][tec]=getCM();
tempo_avanco[i][tec]=(bcm2835_st_read()-...
inicio_tempo_avanco)/1000000;
while((ADS1256_Scan() == 0));
adc1 = ADS1256_GetAdc(5);
volt1 = (adc1 * 100) / 167;
iTemp1[i][tec] = volt1;
printf("Distance: %f\n", D[i][tec]);
}
//Write channel A buffer (0x34)
Write_DAC8532(0x34, Voltage_Convert(5.0, (float)0.0));
bsp_DelayUS(5000000);
// start for compression
inicio_tempo_recuo = bcm2835_st_read();
//Write channel A buffer (0x30)
Write_DAC8532(0x30, Voltage_Convert(5.0, (float)compr[tec]));
while(bcm2835_gpio_lev(SEN_RECUCU) == LOW)
{
cw[tec]=w++;
F[w][tec]=getCM();
tempo_recuo[w][tec]=(bcm2835_st_read()-inicio_tempo_recuo)/1000000;
while((ADS1256_Scan() == 0));
adc2 = ADS1256_GetAdc(6);
volt2 = (adc2 * 100) / 167;
iTemp2[w][tec] = volt2;
printf("Distance: %f\n", F[w][tec]);
}
bsp_DelayUS(300);
//Write channel A buffer (0x30)
Write_DAC8532(0x30, Voltage_Convert(5.0, (float)0.0));
}
// close spi and library
bcm2835_spi_end();
bcm2835_close();

// loop for identify numbers of test
for (q=0;q<=cont; q++)
{
ncont[q]=q;
}

// loop for
for (at=0;at<=9;at++)
{
if (cont1[at]%2==0)
{
smate=30.0;
bigte=0.0;
smatc=30.0;
bigtc=0.0;
n++;
c=-1;
g=-1;

for(d=0;d<=ci[ncont[n]];d++)
{
if (D[d][ncont[n]]>=10.0 && D[d][ncont[n]]<=25.0)
c++;
S[c]=D[d][ncont[n]];
te[c]=tempo_avanco[d][ncont[n]];
iTempin[c]=(iTemp1[d][ncont[n]])/1000000;
if (te[c]< smate)
{
smate=te[c];
}
if (te[c]> bigte)
{
bigte=te[c];
}
}
}
for(d=0;d<=cw[ncont[n]];d++)
{
if (F[d][ncont[n]]>=10.0 && F[d][ncont[n]]<=25.0)
{
g++;
G[g]=F[d][ncont[n]];
tc[g]=tempo_recuo[d][ncont[n]];
iTemp2n[g]=(iTemp2[d][ncont[n]])/1000000;
if (tc[g]< smatc)
{
smatc=tc[g];
}
if (tc[g]> bigtc)

```



```

{
bigtc=tc[g];
}
}
}

// constant velocity: c1 and c3
// constant pressure: c4 and c6
// linear regression of displacement and pressure
gsl_fit_linear (te, 1, S, 1, c, &c0,...
&c1, &cov00, &cov01, &cov11, &sumsq0);p
rintf ("# best fit extension displacement:...
Y = %g + %g X\n", c0, c1);
printf ("# sumsq0 = %g\n", sumsq0);
printf ("\n");

gsl_fit_linear (tc, 1, G, 1, g,...
&c2, &c3, &cov100, &cov101, &cov111, &sumsq1);
printf ("# best fit compression displacement:...
Y = %g + %g X\n", c2, c3);
printf ("# sumsq1 = %g\n", sumsq1);
printf ("\n");

gsl_fit_linear (te, 1, iTemp1n, 1, c,...
&c4, &c5, &cov200, &cov201, &cov211, &sumsq2);
printf ("# best fit extension pressure:...
Y = %g + %g X\n", c4, c5);
printf ("# sumsq2 = %g\n", sumsq2);
printf ("\n");

gsl_fit_linear (tc, 1, iTemp2n, 1, g,...
&c6, &c7, &cov300, &cov301, &cov311, &sumsq3);
printf ("# best fit compression pressure:...
Y = %g + %g X\n", c6, c7);
printf ("# sumsq3 = %g\n", sumsq3);
printf ("\n");

// vector of time, displacement and pressure...
for the linear regression with n points
for(p=0;p<=N-1;p++)
{
tef0[p]=(bigtc-smate)/(N)*p+(smate);
Disp0[p]=(c1*tef0[p])+c0;
Pres0[p]=c4;
}

Vel0[at]=c1/100;
Ff0[at]=c4*(3.14159265*((0.025/2)*(0.025/2)));

for(p=0;p<=N-1;p++)
{
tef1[p]=(bigtc-smatc)/(N)*p+(smatc);
Disp1[p]=(c3*tef1[p])+c2;
Pres1[p]=c6;
}

Vel1[at]=c3/100;
Ff1[at]=-c6*(3.1415*(0.025/2)*(0.025/2));

// save the points experimental
FILE *fileAE[at];
fileAE[at]=fopen(LE[at],"w");
fprintf(fileAE[at],"%s\n","Distância do Avanço...
<----> Tempo do Avanço <----> Pressão do Avanço");
for(j=0;j<=c;j++)
{
fprintf(fileAE[at],"%f %f %f \n",...
S[j],te[j], iTempin[j]);
}
fclose(fileAE[at]);

FILE *fileAC[at];
fileAC[at]=fopen(LC[at],"w");
fprintf(fileAC[at],"%s\n","Distância do Recuo...
<----> Tempo do Recuo <----> Pressão do Recuo");
for(j=0;j<=g;j++)
{
fprintf(fileAC[at],"%f %f %f \n",...
G[j],tc[j], iTemp2n[j]);
}
fclose(fileAC[at]);

// save the points linear regression
FILE *fileFAE[at];
fileFAE[at]=fopen(FLE[at],"w");
fprintf(fileFAE[at],"%s\n","Distância do Avanço...
<----> Tempo do Avanço <----> Pressão do Avanço");
for(j=0;j<=N-1;j++)
{
fprintf(fileFAE[at],"%f %f %f \n",...
Disp0[j], tef0[j], Pres0[j]);
}
fclose(fileFAE[at]);

FILE *fileFAC[at];
fileFAC[at]=fopen(FLC[at],"w");
fprintf(fileFAC[at],"%s\n","Distância do Recuo...
<---->Tempo do Recuo <----> Pressão do Recuo");
for(j=0;j<=N-1;j++)
{
fprintf(fileFAC[at],"%f %f %f \n",...
Disp1[j], tef1[j], Pres1[j]);
}
fclose(fileFAC[at]);
}
}

// save the static map
FILE *filemap;
filemap=fopen("mapa_estatico.txt","w");
fprintf(filemap,"%s\n","For. Atrito <----> Velocidade");
for(i=0;i<=cont;i++)
{
fprintf(filemap,"%f %f \n", Ff0[i], Vel0[i]);
}
for(i=0;i<=cont;i++)
{
fprintf(filemap,"%f %f \n", Ff1[i], Vel1[i]);
}
fclose(filemap);

system("gnuplot -p -e \"load 'scriptautomatic.gnu'\"");
}
}

int main(int argc, char *argv[])
{
//init gtk lib
gtk_init(&argc, &argv);

//add glade files
gtkBuilder = gtk_builder_new();
gtk_builder_add_from_file(gtkBuilder, "test.glade", NULL);

// get main_windows widget
main_window = GTK_WIDGET(gtk_builder_get_object...
(gtkBuilder,"main_window"));
window1 = GTK_WIDGET(gtk_builder_get_object(gtkBuilder,"window1"));
window2 = GTK_WIDGET(gtk_builder_get_object(gtkBuilder,"window2"));
window3 = GTK_WIDGET(gtk_builder_get_object(gtkBuilder,"window3"));
window4 = GTK_WIDGET(gtk_builder_get_object(gtkBuilder,"window4"));
window5 = GTK_WIDGET(gtk_builder_get_object(gtkBuilder,"window5"));

gtk_builder_connect_signals(gtkBuilder,NULL);
g_object_unref(G_OBJECT(gtkBuilder));

// show window
gtk_widget_show(main_window);
gtk_main();

return 0;
}

```