



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Dissertação de Mestrado

**Proposta de Tecnologia para Controle de
Velocidade de *PIGs* Instrumentados Utilizando
*Lógica Fuzzy***

Gustavo Fernandes de Lima

Orientador: Prof. Dr. Andrés Ortiz Salazar

Natal, RN, dezembro de 2014



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Proposta de Tecnologia para Controle de Velocidade de *PIGs* Instrumentados Utilizando Lógica *Fuzzy*

Gustavo Fernandes de Lima

Orientador: Prof. Dr. Andrés Ortiz Salazar

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFRN (área de concentração: Automação e Sistemas) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem PPgEEC: M437
Natal, RN, dezembro de 2014

UFRN / Biblioteca Central Zila Mamede.
Catalogação da Publicação na Fonte.

Lima, Gustavo Fernandes de.

Proposta de tecnologia para controle de velocidade de PIGS instrumentados utilizando lógica fuzzy / Gustavo Fernandes de Lima. - Natal, RN, 2014.

91 f. : il.

Orientador: Andrés Ortiz Salazar

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Computação.

1. Lógica fuzzy - Dissertação. 2. Diferencial de pressão - Dissertação. 3. Controle de velocidade - Dissertação. 4. Válvula by-pass - Dissertação. 5. PIG instrumentado - Dissertação. 6. Arduino - Dissertação. I. Salazar, Andrés Ortiz. II. Universidade Federal do Rio Grande do Norte. III. Título.

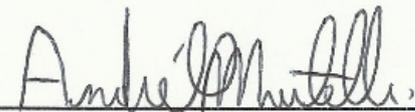
RN/UF/BCZM

CDU 004

Proposta de Tecnologia para Controle de Velocidade de *PIGs* Instrumentados Utilizando Lógica *Fuzzy*

Gustavo Fernandes de Lima

Dissertação de Mestrado aprovada em 16 de dezembro de 2014 pela banca examinadora composta pelos seguintes membros:


Prof. Dr. Andrés Ortiz Salazar (orientador)	DCA/UFRN

Prof. Dr. André Laurindo Maitelli	DCA/UFRN

Prof. Dr. Francisco de Assis Oliveira Fontes	DEM/UFRN

Prof. Dr. Oscar Gabriel Filho	UnP

*Para minha esposa, Diana Lídice, e
meus filhos, Mateus e Heitor, pelas
alegrias de uma vida em família.*

Agradecimentos

Ao Deus, todo poderoso, pela saúde e força em vencer os desafios e dificuldades pelos quais passei.

Aos meus pais, Francisco Gomes e Maria das Dôres, pelo apoio incondicional em todas as minhas decisões.

À minha amada esposa, Diana Lídice, pela paciência e horas abdicadas em prol deste projeto.

Ao professor Andrés Ortiz pelos ensinamentos, por ter acreditado em mim e pela oportunidade de trabalhar no projeto *PIG*.

Ao professor Francisco Fontes pelas orientações de como utilizar corretamente a bancada de testes nos ensaios de “tiro”.

Aos amigos Paulo Luzardo e Victor Galvão, que conheci no projeto *PIG*, pelas melhorias sugeridas na bancada de testes, e aos amigos Diego Moura, Glauco George e José Bernadino, que conheci no LAMP, pela parceria e momentos de descontração.

À todos pelas contribuições diretas e indiretas, pelas críticas e sugestões, durante esta jornada.

Resumo

O controle de velocidade de *PIGs* instrumentados é necessário, dentro da indústria de petróleo e gás, para permitir uma inspeção de dutos de melhor qualidade e maior segurança. Isto só é possível quando esses *PIGs* mantêm uma velocidade baixa durante a operação de inspeção de dutos, na procura de defeitos como corrosão, trincas e amassamentos. Este trabalho tem por objetivo propor uma tecnologia para o controle de velocidade de *PIGs* instrumentados utilizando uma válvula *by-pass* acionada por uma solenóide. Essa válvula tem a capacidade de controlar o diferencial de pressão atuante no corpo do *PIG*, e controlar o diferencial de pressão significa controlar a velocidade desse *PIG*. Uma bancada de testes, em escala reduzida, foi montada para simulação de situações de “tiro” e de aceleração sofridos pelos *PIGs*. A bancada permitiu estudar as reações das pressões a montante e a jusante da válvula *by-pass*. A lógica *Fuzzy* foi utilizada para determinar os tempos de abertura e fechamento dessa válvula, com objetivo de controlar o diferencial de pressão. Os testes na bancada mostraram uma redução substancial no valor do diferencial de pressão atuante no *PIG* após a ocorrência do “tiro”. Também ocorreu uma redução no tempo em que o diferencial de pressão vai ao máximo e retorna ao valor de referência predefinido. Os resultados obtidos permitem concluir que é possível controlar o diferencial de pressão atuante no *PIG*, assim também a velocidade, utilizando a lógica *Fuzzy*, atuando na abertura e fechamento da válvula *by-pass* proposta.

Palavras-chave: *PIG* instrumentado, diferencial de pressão, controle de velocidade, válvula *by-pass*, lógica *Fuzzy*, Arduino.

Abstract

The speed control of smart *PIGs* is necessary, in oil and gas industry, for allow a better and safer pipeline inspection. This is possible when those smart it *PIGs* maintain a low speed while performs a pipeline inspection, in the search for defects as corrosion, cracks and dents. Our objective propose a technology for speed control of smart pigs using a by-pass valve triggered by a solenoid. This valve is able to control the pressure differential applied in *PIG*'s body. Once we control the pressure differential, it is possible to control the speed of this *PIG*. A experimental benchmark was mounted in reduced scale for simulation of the *PIG* speed excursion situation and acceleration suffered by the *PIGs*. The experimental benchmark allowed to studying the reactions of the upstream and downstream pressure over the by-pass valve. The Fuzzy logic was used for determine the opening and closing period this valve to control pressure differential. The experimental tests show substantial reduction in pressure differential value over *PIG* after happening speed excursion. Also, the time necessary for the pressure differential go to maximum and returns to default reference value. The results obtained allow conclude which is possible to controlling pressure differential applied in *PIG*, thus also the speed, using Fuzzy logic, acting opening and closing period of proposed by-pass valve.

Keywords: Smart *PIG*, pressure differential, speed control, by-pass valve, *Fuzzy* logic, Arduino.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Símbolos e Abreviaturas	vii
1 Introdução	1
1.1 Breve Histórico	3
1.2 Objetivos	4
1.3 Organização do Texto	4
2 Fundamentação Teórica	5
2.1 Ferramenta para Inspeção de Dutos	5
2.1.1 Introdução	5
2.1.2 O Movimento do <i>PIG</i>	9
2.1.3 Revisão Bibliográfica	11
2.1.4 Publicações do Grupo de Pesquisa <i>PIG</i> da UFRN	16
2.2 Lógica <i>Fuzzy</i>	18
2.2.1 Introdução	18
2.2.2 Sistemas de Controle Baseados em Lógica <i>Fuzzy</i>	20
2.2.3 <i>Fuzzificação</i> ou Codificação	21
2.2.4 Regras (Base de Conhecimento)	22
2.2.5 Inferência	23
2.2.6 <i>Defuzzificação</i> ou Decodificação	24
2.3 Conclusões	25
3 Implementação do Sistema	27
3.1 Plataforma Arduino	27
3.2 Dispositivo de Aquisição de Dados (USB-6008)	33

3.3	Transdutor de Pressão	34
3.4	Válvula Solenóide e seu Acionamento	37
3.5	Válvula <i>By-pass</i>	39
3.5.1	Coeficiente de Vazão da Válvula <i>By-pass</i> Proposta	41
3.6	Bancada de Testes	42
3.7	Metodologia Experimental	44
3.7.1	Simulação de “Tiro”	45
3.7.2	Simulação de Aceleração	46
3.8	Conclusões	46
4	Testes e Resultados	47
4.1	Nomenclatura da Bancada de Testes	47
4.2	Testes Iniciais	48
4.3	Testes sem o Controlador <i>Fuzzy</i>	49
4.3.1	Simulação de “Tiro” e Resultados	49
4.3.2	Simulação de Aceleração e Resultados	50
4.4	Configuração do Controlador <i>Fuzzy</i>	51
4.4.1	Variáveis de Entrada do Sistema	52
4.4.2	Variável de Saída do Sistema	53
4.4.3	Bloco de Regras	53
4.4.4	Implicação e Agregação de Regras	54
4.5	Testes com Controlador <i>Fuzzy</i>	54
4.5.1	Simulação de “Tiro” e Resultados	54
4.5.2	Simulação de Aceleração e Resultados	56
4.6	Comparação dos Resultados Obtidos	57
4.6.1	Simulação de “Tiro”	57
4.6.2	Simulação de Aceleração	59
4.7	Conclusões	60
5	Conclusões e Trabalhos Futuros	61
5.1	Conclusões	61
5.2	Trabalhos Futuros	62
	Referências	63
A	Produção Científica Publicada	69

Lista de Figuras

2.1	Tipos de <i>PIGs</i> de limpeza.	5
2.2	Exemplo de <i>PIG</i> instrumentado inserido no duto.	6
2.3	Exemplo de <i>PIG</i> instrumentado dividido em dois módulos.	6
2.4	Exemplos de <i>PIGs</i> instrumentados.	7
2.5	Fluxo magnético uniforme pela parede do duto.	8
2.6	Detecção de anomalia pela fuga de fluxo magnético.	8
2.7	Balanço de forças que agem no <i>PIG</i>	9
2.8	Vista lateral do flexpig.	11
2.9	<i>PIG</i> palito com 250 sensores.	13
2.10	Protótipo do <i>PIG</i> self-drive.	13
2.11	<i>PIG</i> cobra palito.	14
2.12	Válvulas <i>by-pass</i>	16
2.13	Tipos de válvulas <i>by-pass</i>	17
2.14	Válvula com pistão acionada por solenóide.	17
2.15	Tipos de passagem das pertinências.	18
2.16	Modelo para um sistema de controle.	20
2.17	Operador humano uma analogia do controlador <i>Fuzzy</i>	20
2.18	Blocos do controlador <i>Fuzzy</i>	21
2.19	Exemplo de função de pertinência.	22
2.20	Base de conhecimento.	23
3.1	Diferentes placas da plataforma Arduino.	28
3.2	Blocos identificados do Arduino Uno.	29
3.3	Ethernet shield	30
3.4	Tela para programação do Arduino.	30
3.5	Configuração da porta serial.	31
3.6	Configuração da placa Arduino Uno.	31
3.7	Exemplo de odulação por largura de pulso.	32
3.8	Dispositivo de aquisição de dados USB-6008.	33
3.9	Transdutor A2X utilizado.	34

3.10	Esquema elétrico implementado.	34
3.11	Linearização de pontos do transdutor PT-01.	35
3.12	Linearização de pontos do transdutor PT-02.	36
3.13	Vista lateral da válvula solenóide.	37
3.14	Posições da válvula solenóide.	38
3.15	Transistor NPN TIP 122.	39
3.16	Vista em corte da válvula <i>by-pass</i>	40
3.17	Imagem real da válvula <i>by-pass</i>	40
3.18	Bancada de testes para simulações de “tiro” e de aceleração.	42
3.19	Detalhe da bancada de testes.	43
3.20	Bancada montada com os equipamentos eletrônicos.	43
3.21	Diagrama de blocos da bancada de testes.	44
3.22	Conexões realizadas na bancada de testes.	45
4.1	Nomenclatura dos componentes pneumáticos da bancada de testes.	47
4.2	Ensaio com V2 aberto 100% e VP fechada 100%.	48
4.3	Ensaio com V2 e VP fechados 100%.	48
4.4	Ensaio com V2 realizando “o Tiro” e VP fechada 100%.	49
4.5	Estimação de velocidade do <i>PIG</i>	49
4.6	Velocidade estimada do <i>PIG</i> depois do “tiro”.	50
4.7	Diferencial de pressão obtido depois da aceleração.	51
4.8	Velocidade estimada do <i>PIG</i> depois da aceleração.	51
4.9	Variável de entrada “deltaP”.	52
4.10	Variável de entrada “taxadeltaP”.	53
4.11	Variável de saída “PWM”.	53
4.12	Base de regras utilizada.	54
4.13	Diferencial de pressão obtido depois de um “tiro”.	55
4.14	Velocidade do <i>PIG</i> com controle <i>Fuzzy</i>	55
4.15	Sinal PWM aplicado na válvula <i>by-pass</i>	56
4.16	Diferencial de pressão atuante no <i>PIG</i> depois da aceleração.	56
4.17	Controle de velocidade do <i>PIG</i> durante a aceleração.	57
4.18	Sinal PWM aplicado na válvula <i>By-pass</i>	57
4.19	Resultados sem e com controle <i>Fuzzy</i> para o evento “tiro”.	58
4.20	Tendência de velocidade do <i>PIG</i> depois do “tiro”.	58
4.21	Resultados sem e com controle <i>Fuzzy</i> para o evento de aceleração.	59
4.22	Tendência de velocidade do <i>PIG</i> depois da aceleração.	60

Lista de Tabelas

2.1	Algumas tecnologias utilizadas nos <i>PIGs</i> instrumentados.	9
2.2	Vantagens e desvantagens da Lógica <i>Fuzzy</i>	19
3.1	Calibração do transdutores de pressão.	35
3.2	Especificações técnicas da válvula solenóide.	38
4.1	Comparação de resultados obtidos nas simulações de “tiro”.	58
4.2	Comparação de resultados obtidos durante as simulações de aceleração.	59

Lista de Símbolos e Abreviaturas

A	Área do <i>PIG</i>
A_{duto}	Área do duto
B	Coefficiente de atrito viscoso
C_v	Coefficiente de vazão para válvula <i>by-pass</i>
F_a	Força de atrito entre o <i>PIG</i> e a parede do duto
F_g	Força exercida pelo diferencial de pressão ΔP , que move o <i>PIG</i>
F_p	Força peso que age quando o duto tem inclinação
F_s	Atrito seco
M	Massa do <i>PIG</i>
P_1	Pressão a montante do <i>PIG</i>
P_2	Pressão a jusante do <i>PIG</i>
Q	Vazão de fluido
V	Velocidade do fluido
ΔP	Diferencial de pressão que age no <i>PIG</i>
β	Ângulo de inclinação do duto
ρ	Densidade relativa do gás em relação ao ar
a	Aceleração do <i>PIG</i>
g	Aceleração da gravidade
pt_{01}	Pressão no transdutor PT-01

<i>pt</i> ₀₂	Pressão no transdutor PT-02
<i>v</i>	Velocidade do <i>PIG</i>
<i>v</i> ₁	Sinal de tensão produzido pelo transdutor PT-01
<i>v</i> ₂	Sinal de tensão produzido pelo transdutor PT-02
ANP	Agência Nacional do Petróleo
CENPES	Centro de Pesquisas da Petrobrás
DAQ	Data Acquisition (Aquisição de Dados)
EEPROM	Electrically Erasable Programmable Read-Only Memory (Memória Programável Somente para Leitura Apagável Eletricamente)
EMAT	Electro Magnetic Acoustic Transducers (Transdutores Acústicos Eletro Magnéticos)
IDE	Integrate Development Environment (Ambiente de Desenvolvimento Integrado)
LAMP	Laboratório de Avaliação e Medição em Petróleo
MCI	Monitoramento de Corrosão Interna
MFL	Magnetic Flux Leakage (Fuga de Fluxo Magnético)
NC	Normal Close (Normal Fechado)
PID	Proporcional-Integrativo-Derivativo
PIG	Pipeline Inspection Gauge (Ferramenta para Inspeção de Dutos)
PWM	Pulse Width Modulation (Modulação por Largura de Pulso)
RAM	Random Access Memory (Memória de Acesso Aleatório)
RISC	Reduced Instruction Set Computing (Computador com um Conjunto Reduzido de Instruções)
RTI	Robô de Inspeção de Tubulações
UFRN	Universidade Federal do Rio Grande do Norte
UMI	Unidade de Medidas Inercial

Capítulo 1

Introdução

A malha dutoviária brasileira, segundo a Agência Nacional do Petróleo (ANP), citado por Alves (2007), tem uma extensão de 15.069 km, sendo 7.404 km de oleodutos e os outros 7.665 km de gasodutos. Este modal de transporte tem como vantagens (de Araújo 2012): alta economia no transporte, baixo consumo de energia e maior segurança para o meio ambiente e para a população.

O transporte de petróleo e derivados por dutos, nos quais a manutenção não se realiza periodicamente, provoca dois problemas indesejados. O primeiro é a incrustação ou formação de sedimentos, como a parafina, e a conseqüente redução da área útil desse duto. O segundo diz respeito a corrosão ou desgaste da parede interna da tubulação, que pode culminar com seu rompimento e posterior vazamento do produto transportado. Em ambos os casos é possível utilizar uma ferramenta, que tem por objetivo a desobstrução e/ou inspeção da integridade interna dessa tubulação, conhecida pelo nome de *PIG* (do inglês *Pipeline Inspection Gauge*) ou Ferramenta para Inspeção de Dutos.

Os *PIGs* instrumentados, segundo Pereira (2012), normalmente se deslocam no interior dos dutos de forma autônoma, impulsionados pela pressão produzida pelo fluido e percorrem toda a extensão do duto, podendo atingir centenas de quilômetros e são capazes de armazenar os registros da operação em curso para análise e avaliação posteriores.

Essa análise de dados permite que as empresas operadoras de inspeção de dutos tomem a melhor decisão quanto à necessidade de manutenção corretiva ou preventiva no duto, conforme o caso. Tais operações de inspeção, com *PIGs* instrumentados, são justificadas por duas razões (Pereira 2012): malha dutoviária mundial estar envelhecida e a preocupação crescente com o meio ambiente.

Uma malha dutoviária envelhecida significa uma malha sujeita a defeitos, principalmente, a corrosão. Para um duto a corrosão representa a perda de material do tubo, isto é, a diminuição da espessura de sua parede de contenção, o que aumenta a possibilidade de falha (de Araújo 2012). Uma falha ou acidente em dutos pode causar grandes prejuízos

financeiros, sociais e ambientais.

Assim, para que um *PIG* instrumentado equipado com sensores próprios para detecção de corrosão ou outro defeito em dutos possa identificar e quantificar essas anomalias é necessário que ele mantenha uma baixa velocidade de deslocamento, dentro de limites aceitáveis. Isso para que os sensores possam coletar dados com segurança e confiabilidade. E, segundo Nguyen, Yoo, Rho & Kim (2001), essa velocidade é geralmente na faixa de 1 a 5 m/s em dutos de líquido e de 2 a 7 m/s em dutos de gás.

Atualmente, está em desenvolvimento no Laboratório de Avaliação e Medição em Petróleo (LAMP), na Universidade Federal do Rio Grande do Norte (UFRN), o projeto *PIG*. Uma das ações deste projeto foi a montagem de uma planta ou bancada de testes constituída de uma tubulação de 4", uma válvula de controle do tipo *by-pass* e instrumentos eletrônicos para coleta de dados. O uso dessa bancada permite o estudo e simulação de um *PIG* em movimento dentro do duto, em função do comportamento das pressões a montante e a jusante atuantes do corpo desse *PIG*.

O *PIG* é um processo não-linear e sem um modelo matemático (descrição de um sistema de controle usando equações diferenciais) bem definido. Estas características dificultam a sintonia de controladores clássicos como o Proporcional-Integrativo-Derivativo (PID) ou controladores modernos como o Preditivo. Para utilizar estes controladores é preciso uma função de transferência, que relacione pressão do fluido e velocidade alcançada pelo *PIG*.

Por causa dessa complexidade, inerente do sistema, foi importante buscar por uma alternativa viável para o controle de velocidade de *PIGs* instrumentados. A resposta encontrada foi estudar e utilizar a lógica *Fuzzy* para o desenvolvimento de um controlador inteligente, que operasse sem a necessidade de um modelo matemático do processo.

Este é um diferencial do controle *Fuzzy*, uma vez que permite modelar as ações a partir de informações fornecidas pelo operador, ou seja, conhecimento especialista, em vez de modelar o processo em si (Carvalho et al. 2010). O controle e a modelagem *Fuzzy* são técnicas utilizadas para manusear informações imprecisas ou aproximadas a respeito do processo. Possibilita a inclusão da experiência humana em controle computadorizado, tornando possível decisões em problemas complexos (Shaw & Simões 1999).

Com base na descrição acima, este trabalho visa implementar um controle de diferencial de pressão, que implicará no controle de velocidade, utilizando microcontrolador para aquisição de dados e acionamento de uma válvula solenóide. O controlador implementado baseado na técnica de controle *Fuzzy*, utilizou a ferramenta *Fuzzy Toolbox* do software Matlab^{TM1} para calcular os tempos de abertura e fechamento da válvula de

¹Matlab é propriedade da Mathworks

controle tendo como sinais de entrada o diferencial de pressão e sua derivada. O sinal de saída gerado modulou a válvula de controle de maneira similar a um sinal PWM (do inglês *Pulse Width Modulation*) ou Modulação por Largura de Pulso, obtendo assim, um sinal médio para abertura da válvula.

1.1 Breve Histórico

Em 1865, foi construído na Pensilvânia (EUA) o primeiro oleoduto com duas polegadas de diâmetro feito de ferro fundido com extensão de 8 km ligando um campo de produção a uma estação de carregamento de vagões (Barros 2009).

Segundo Diaz (2008) a utilização dos primeiros *PIGs* ocorreu por volta de 1870, quando as empresas transportadoras de petróleo observaram que a vazão das linhas de transporte começava a decrescer e a pressão nas bombas a aumentar, indicando que depósitos de parafina estariam se formando na parede dos dutos.

Por causa disso, vários artifícios para remoção da parafina foram testados, mas sem sucesso. Até que surgiu a idéia de se bombear alguma substância por dentro do duto, como um feixe de tecidos, e os resultados foram positivos. Mais tarde, os tecidos foram substituídos por couro (Souza 2003).

Ainda segundo Souza (2003), um outro relato sobre a utilização de *PIG* aconteceu em 1904, em que uma bola de borracha foi lançada numa linha de 4" para verificar se um deslizamento de rocha, ocorrido durante a construção do duto, teria causado alguma restrição na área útil do duto.

O desenvolvimento dos *PIGs* instrumentados começou na década de 1960 e seguiu a seguinte cronologia como relata Nestleroth e Porter, citado por Souza (2003):

- 1964 - A empresa Tuboscope apresenta o primeiro *PIG* MFL (do inglês *Magnetic Flux Leakage*) ou Fuga de Fluxo Magnético, para inspecionar a parte inferior do tubo;
- 1966 - A Tuboscope constrói o primeiro *PIG* do tipo MFL para inspecionar toda a circunferência do tubo;
- 1971 - Outras empresas disponibilizam o *PIG* MFL de baixa resolução;
- 1978 - A British Gas desenvolve o primeiro *PIG* de alta resolução;
- 1986 - Primeiro *PIG* ultra-sônico para linha de líquidos;
- 1986 a 1996 - Outras empresas disponibilizam o *PIG* de alta resolução;
- 1992 - Protótipo do *PIG* ultra-sônico para detecção de trinca;
- 1997 - A Pipetronix desenvolve um *PIG* de detecção de trinca;

- 1998 - Primeiro *PIG* de fluxo magnético transversal;
- 1998 - Protótipo do EMAT (do inglês *Electro Magnetic Acoustic Transducers*) ou Transdutores Acústicos Eletro Magnéticos - *PIG* ultrasônico sem necessidade de líquido acoplante.

No Brasil, segundo Oliveira e Camerini, citado por Diaz (2008), tem se registros de desenvolvimento de *PIGs* instrumentados a partir de 1992, por conta do CENPES (Centro de Pesquisa da Petrobras), apresentando o primeiro protótipo no ano de 1995.

1.2 Objetivos

Este trabalho tem como objetivo desenvolver um controlador de velocidade para *PIGs* instrumentados, que permita a realização da operação de inspeção segura e eficiente. Os objetivos específicos são:

- Implementar um sistema de aquisição de dados para obtenção de valores de pressão a montante e a jusante, a fim de calcular o diferencial de pressão atuante no *PIG* e estimar a velocidade de deslocamento do mesmo;
- Diminuir os efeitos do fenômeno conhecido por "tiro" sofrido pelos *PIGs* dentro do duto, por meio da abertura e fechamento de uma válvula *by-pass* comandada por solenóide;
- Aplicar a lógica *Fuzzy* no desenvolvimento de um controlador inteligente, para atuar nos tempos de abertura e fechamento da válvula *by-pass* proposta.

1.3 Organização do Texto

O Capítulo 2 aborda a fundamentação teórica apresentando, em detalhes, a tecnologia dos *PIGs* e, também, a uma abordagem sobre a lógica *Fuzzy*. O Capítulo 3 descreve todos os componentes e equipamentos utilizados na implementação do sistema ou bancada de testes para realização de simulação de situações de "tiro" e de aceleração em *PIGs*. O Capítulo 4 mostra os testes realizados com bancada montada e os resultados obtidos nos ensaios experimentais. Por último, o Capítulo 5 apresenta as conclusões deste trabalho e algumas sugestões para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo abordará a fundamentação teórica para o entendimento do trabalho, compreendendo uma revisão sobre o tema *PIG* e a teoria sobre lógica *Fuzzy*, aplicada no desenvolvimento de um controlador inteligente.

2.1 Ferramenta para Inspeção de Dutos

2.1.1 Introdução

Os *PIGs* podem ser classificados em duas categorias: *PIGs* de limpeza (*Utility PIGs*) e *PIGs* instrumentados (*In line inspection tools* ou *Smart PIGs*).

Os *PIGs* de limpeza realizam função de limpeza, separação de produtos, remoção de água e desamassamento dos dutos (Diaz 2008). Quando utilizados para remoção de detritos acumulados (p.ex. parafina) permitem restaurar a seção reta original da tubulação. A Figura 2.1 apresenta três tipos de *PIGs* de limpeza disponíveis no mercado.



(a) Copo.



(b) Disco.



(c) Espuma.

Figura 2.1: Tipos de *PIGs* de limpeza.

Fonte: de Oliveira Souza (2005).

Os *PIGs* instrumentados fornecem as informações das condições dos dutos, extensão e localização dos defeitos das tubulações, com a ajuda de equipamentos de inspeção embarcados nos *PIGs* (Diaz 2008). A construção básica de um *PIG* instrumentado, com apenas um módulo, pode ser visualizada na Fig. 2.2, onde existe uma cápsula apoiada por dois suportes de borracha. Dentro da cápsula estão os circuitos eletrônicos e as baterias do *PIG*. Os suportes de borracha mantêm a cápsula centralizada na tubulação (Mazzini 2009).

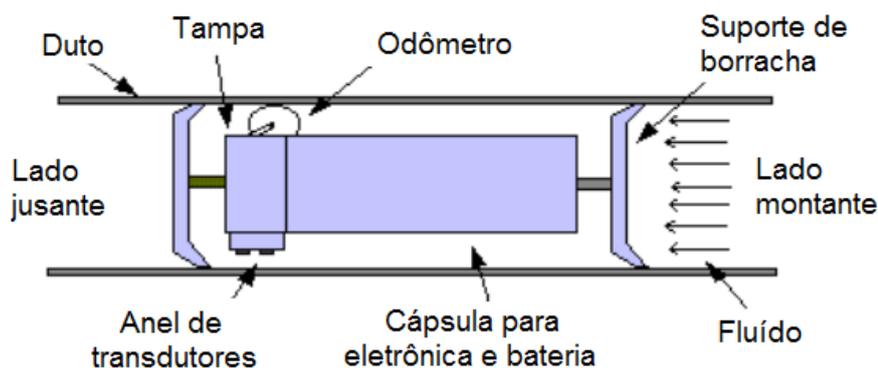


Figura 2.2: Exemplo de *PIG* instrumentado inserido no duto.

Fonte: Mazzini (2009).

Essa estrutura básica de um módulo pode ser expandida em vários módulos de comprimentos menores, com o objetivo de tornar o conjunto mais articulável, permitindo ao *PIG* realizar curvas mais fechadas durante as corridas. A Figura 2.3 mostra um *PIG* com dois módulos e um prolongador realizando a conexão entre eles.

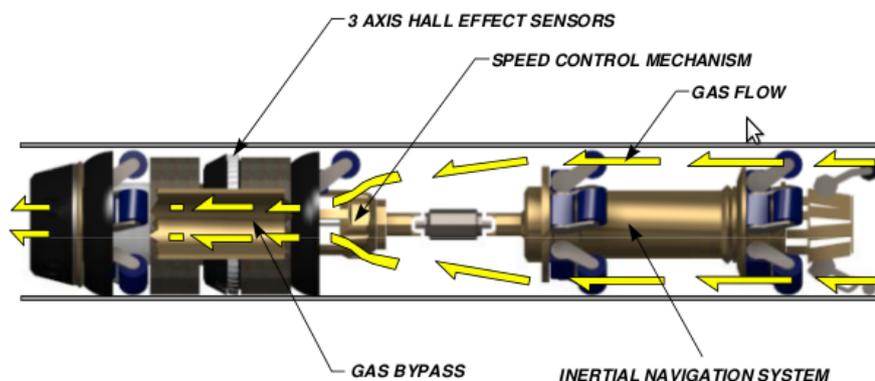


Figura 2.3: Exemplo de *PIG* instrumentado dividido em dois módulos.

Fonte: Sutherland & Paz (2000).

Nesta situação, de módulos separados, é importante observar a existência de orifícios

no módulo traseiro (direita), isso para permitir a passagem do fluxo de gás (*gas flow*) até o módulo dianteiro (ou cabeça). No módulo dianteiro pode existir uma válvula reguladora de fluxo (*gas bypass*) que limita a passagem de gás pelo *PIG*. Tal recurso permite controlar a velocidade do *PIG* (*speed control mechanism*).

Outra vantagem deste sistema modulado é a separação dos componentes, de modo a evitar interferência entre os sistemas. Por exemplo, um *PIG* instrumentado composto por três módulos poderá ter a seguinte divisão de componentes: na cabeça do *PIG* estão os sensores responsáveis pela inspeção do duto; no segundo módulo reside a eletrônica embarcada e no terceiro módulo estão as baterias, que fornecem energia para todos os outros módulos e deverão ter duração suficiente para toda a corrida. Além disso, cada módulo deverá ser vedado contra a entrada de fluídos, de modo a evitar possíveis danos aos componentes internos.

Para realizar a inspeção por defeitos dentro do duto, os *PIGs* instrumentados precisam transportar em suas cápsulas sensores capazes de detectar a falha ou descontinuidade que se deseja encontrar. São capazes de detectar e determinar as dimensões de pequenos defeitos, informando também a sua localização (Mazzini 2009).

Duas categorias de *PIGs* instrumentados são: *PIG* geométrico (Fig. 2.4(a)) e *PIG* de corrosão (Fig. 2.4(b)). O primeiro é uma ferramenta instrumentada de sensores múltiplos capaz de localizar e quantificar ovalizações, reduções de diâmetro e curvaturas. Operam em tubulações com diâmetros de 3” a 42” (Pipeway 2012). Já o segundo, é uma ferramenta instrumentada de alta resolução, com grande capacidade de registro de dados capaz de localizar, quantificar e discriminar perdas de espessura (interna e externa) da parede do duto, através da técnica MFL. Operam em dutos com diâmetros de 6” a 24” (Pipeway 2012).



(a) Pig geométrico.



(b) Pig de corrosão.

Figura 2.4: Exemplos de *PIGs* instrumentados.

Fonte: Pipeway (2012).

A técnica MFL consiste na aplicação de um fluxo magnético (transversal ou longitudinal) na parede do tubo até obter sua saturação magnética, à medida que a ferramenta se desloca, através do sistema de magnetização do *PIG* (ímãs) (de Araújo 2012). A Figura 2.5 apresenta um exemplo onde um duto íntegro é percorrido por campo magnético uniforme, com linhas paralelas à parede do referido duto.

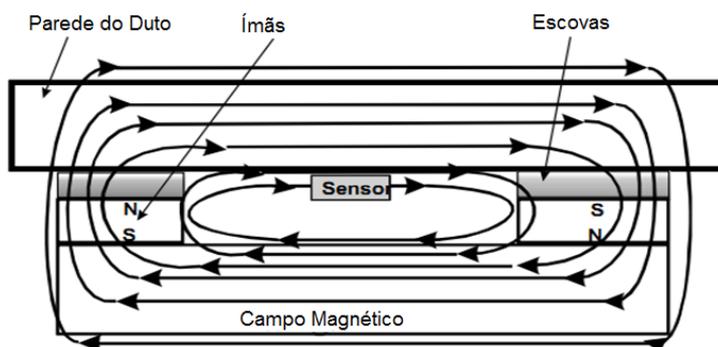


Figura 2.5: Fluxo magnético uniforme pela parede do duto.

Ainda segundo de Araújo (2012) na presença de anomalias ocorre a fuga do campo magnético, que é medida pelos sensores e armazenada pelo sistema de registro de dados. Assim, a anomalia só é detectada se existir uma fuga de campo magnético. A Figura 2.6 mostra uma anomalia no duto e em função disto o campo magnético sofre um desvio ou fuga (as linhas não ficam mais paralelas ao duto), sendo registrado pelo sistema de dados.

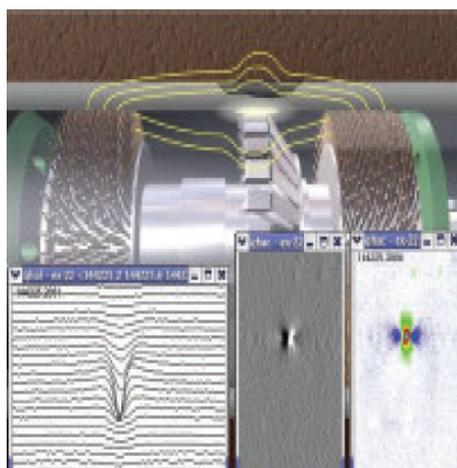


Figura 2.6: Detecção de anomalia pela fuga de fluxo magnético.

Atualmente, existem mais de 350 tipos diferentes para as diversas necessidades demandadas pelas operadoras de dutos (Souza 2003). Por isso, Bueno (2007) apresenta

uma tabela com diversos tipos de *PIGs* instrumentados e vários objetivos de inspeção. Uma parte dessa diversidade de *PIGs* foi transcrita na Tab. 2.1. Nela foram listados dez tipos de falhas possíveis em um duto e seis tecnologias capazes de detectar tais defeitos.

Tabela 2.1: Algumas tecnologias utilizadas nos *PIGs* instrumentados.

Objetivo Inspeção	Exemplos de tecnologias utilizadas					
	Geo-métrico	MFL	Inercial	Umbilical	Ultrasônico p/ trincas	Magnético p/ trincas
Amassamentos	•					
Mossas	•					
Ovalizações	•					
Entalhes, Sulcos		•		•		
Erosão interna		•		•		
Trincas					•	•
Raios de curvatura	•		•			
Corrosão interna		•		•		
Corrosão externa		•		•		
Mudança de traçado				•		

2.1.2 O Movimento do *PIG*

Segundo Pereira, de Oliveira, de Lima, Salazar, Maitelli & de Assis O. Fontes (2013) a dinâmica do movimento do *PIG* é baseada na Segunda Lei de Newton. O balanço de forças é mostrado na Fig. 2.7.

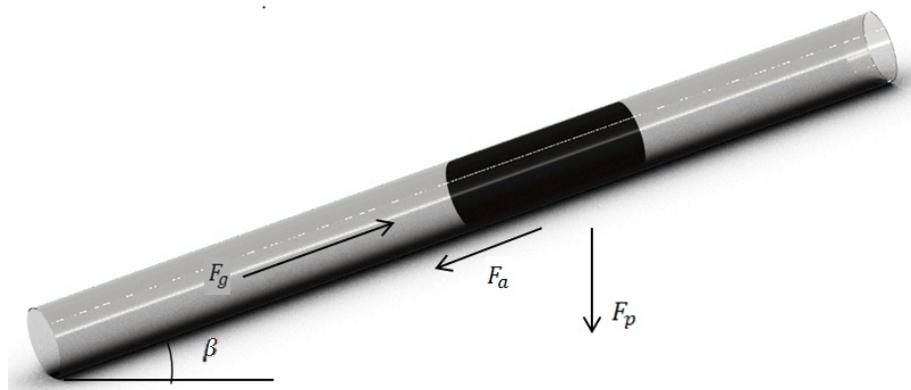


Figura 2.7: Balanço de forças que agem no *PIG*.

onde: F_g é a força exercida pelo diferencial de pressão ΔP , que move o *PIG*; F_a é a força de atrito entre o *PIG* e a parede do duto; F_p é a força peso que age quando o duto tem inclinação; e β é o ângulo de inclinação do duto.

Estas forças estão relacionadas pela Segunda Lei de Newton, da seguinte forma:

$$M \cdot a = F_g + F_a + F_p \quad (2.1)$$

$$F_g = \Delta P \cdot A \quad (2.2)$$

$$F_a = B \cdot v + F_s \quad (2.3)$$

$$F_p = M \cdot g \cdot \sin \beta \quad (2.4)$$

onde: M é a massa do *PIG*; a é a sua aceleração; ΔP é o diferencial de pressão no *PIG*; A é a área do *PIG* na qual é exercida a pressão do gás; B é o coeficiente de atrito viscoso; v é a velocidade do *PIG*; F_s é o atrito seco; g é a aceleração da gravidade.

Para obter o modelo linear do sistema foi desconsiderado, inicialmente, o atrito seco e a força peso, que serão modelados posteriormente para análise de seus efeitos no sistema. Dessa forma, temos que:

$$M \cdot a = \Delta P \cdot A + B \cdot v = (P_1 - P_2) \cdot A + B \cdot v \quad (2.5)$$

onde: P_1 é a pressão a montante do *PIG*; e P_2 é a pressão a jusante do *PIG*.

Para a Eq. (2.5) ficar em função da velocidade do *PIG*, a aceleração foi substituída por $\frac{dv}{dt}$, conforme a Eq. (2.6).

$$M \cdot \frac{dv}{dt} + B \cdot v = \Delta P \cdot A \quad (2.6)$$

Realizando a transformada de Laplace, considerando as condições iniciais iguais a zero, obtemos a função de transferência do sistema, como mostrado na Eq. (2.7).

$$\frac{V(s)}{\Delta P \cdot A} = \frac{1}{M \cdot s + B} \Rightarrow \frac{V(s)}{F_g(s)} = \frac{1}{M \cdot s + B} \quad (2.7)$$

Onde a entrada é a força que move o *PIG*, causada pelo diferencial de pressão e a saída é a velocidade do mesmo.

A Equação (2.7) será utilizada no Capítulo 4 para a estimação de velocidade alcançada pelo *PIG*, com base nos valores de diferencial de pressão ΔP obtidos na bancada de testes. Nos cálculos com a Eq. (2.7) foram empregados os seguintes valores: $M = 80$ kg; $B = 900$ N · s/m; $A = 0,0081$ m², que corresponde a um duto de 4" de diâmetro.

2.1.3 Revisão Bibliográfica

A utilização de *PIGs* instrumentados como ferramenta de aquisição de dados para estudos de corrosão, por exemplo, ou como objeto direto de pesquisa cresceu na última década. Muitos pesquisadores vem se dedicando ao estudo dessa ferramenta e realizando diversos trabalhos e publicações. A quantidade de publicações sobre *PIGs* é relativamente grande e por isso é apresentada uma síntese desses trabalhos.

Sutherland & Paz (2000) apresentaram os avanços tecnológicos projetados e construídos para uma ferramenta do tipo MFL de alta resolução. Este ferramenta incorpora tecnologias de: mapeamento inercial, através de sensores circunferencial; redes neurais para o dimensionamento da perdas metálicas nas paredes do duto inspecionado; e controle de velocidade por meio de *by-pass* de fluxo de gás, melhorando assim a qualidade dos dados coletados.

Franzoi et al. (2001) desenvolveram um *PIG* 100% em resina de poliuretano, chamado de FLEXPIG (ver Fig. 2.8), capaz de transpor restrições internas no duto. Foi instalada a eletrônica embarcada necessária para que se pudesse obter dados que atestassem a posição e magnitude das eventuais restrições internas.



Figura 2.8: Vista lateral do flexpig.

Nguyen, Yoo, Rho & Kim (2001) propuseram um controlador não-linear baseado no método *back-stepping* para controlar a velocidade de um *PIG*. O sistema em malha fechada era estável pelo critério de estabilidade de Lyapunov. Resultados de simulação mostraram que o controlador proposto podia ser usado para controlar a velocidade de *PIGs* com boa performance em tubulações de gás.

Nguyen, Kim, Yoo & Rho (2001) modelaram e simularam o controle de fluxo de gás através de um *PIG* em um duto de gás natural. Resultados de simulação mostraram que o

modelo matemático e o esquema computacional propostos foram eficazes para estimar a posição e velocidade do *PIG* com uma dada condição operacional da tubulação.

Souza (2003) descreveu a cronologia do desenvolvimento dos *PIGs* instrumentados, a necessidade de escolha do tipo de *PIG* em função do defeito a ser inspecionado, a avaliação dos defeitos de corrosão por níveis de complexidade e os principais métodos atualmente utilizados, ressaltando as limitações de cada um quanto à sua empregabilidade.

Kim et al. (2003) verificaram o modelo teórico para o comportamento dinâmico de um *PIG* viajando através de um duto de gás em alta pressão. Até aquele momento não se tinha certificação experimental para o comportamento dinâmico do *PIG*. Foi fabricado um *PIG* geométrico de 30" e realizada uma corrida real, cumprida com êxito. A comparação dos resultados de simulação com resultados experimentais mostraram eficácia na predição e velocidade do *PIG*.

Campos et al. (2004) projetaram um sistema para estimar a trajetória descrita por um *PIG* instrumentado a partir dos dados fornecidos pelos sensores de uma UMI (Unidade de Medidas Inercial) de baixo custo e por um odômetro digital. Os objetivos deste trabalho foram os de localizar onde as eventuais corrosões ocorrem e realizar o mapeamento dos oleodutos, uma vez que para muitos locais a documentação era insuficiente.

O trabalho de Tolmasquim (2004) obteve uma ferramenta eficiente para ajudar no controle e projeto das operações de passagem de *PIGs* em dutos. Tais operações para serem eficientes e seguras demandam que parâmetros operacionais (p.ex. pressões máximas e mínimas no duto) sejam bem avaliados durante a etapa de planejamento e mantidos dentro dos limites durante o acompanhamento da operação. A ferramenta proposta simula o escoamento mediante diferentes estratégias para alcançar uma operação eficiente.

Han et al. (2004) desenvolveram um *PIG* autônomo para determinar posição, curvatura e deformações dos dutos. Foram apresentados a concepção de projeto, a integração dos sistemas e o método de processamento/análise dos dados. Resultados de um experimento real em um duto de 58 km mostram erro de navegação menor que 0,2 m.

de Oliveira Souza (2005) realizou um levantamento de dados de laboratório sobre as forças de contato *PIG*-duto, para determinar as forças de remoção dos depósitos de parafina. Resultados mostraram que a remoção de parafina com o modelo disponível consegue estimar a força inicial para a quebra desse depósito dentro de uma faixa de incerteza de $\pm 30\%$.

O trabalho de Plácido (2005) analisou os principais sistemas robotizados de inspeção interna de dutos, sendo evidenciados e descritos suas características mais relevantes, tais como, sistema de sensoriamento, tipo de alimentação, tração e sistemas de controle.

Franzoi et al. (2005) propuseram a construção de um novo *PIG* instrumentado, o cha-

mado *PIG* Palito (ver Fig. 2.9), e o compararam com outros dois tipos de *PIG*, o MFL e o Ultrasônico. Na comparação entre o *PIG* MFL e o *PIG* Palito, o sistema MFL indicou regiões sem defeitos e o mesmo ocorreu com o *PIG* Palito. Nas regiões defeituosas indicadas pelo MFL, o mesmo ocorreu com o *PIG* Palito, mas não foi possível realizar comparações de profundidade. Quanto a comparação entre o *PIG* Ultrasônico e o *PIG* Palito, ambos os sistemas se propõem a indicar diretamente a medida dos defeitos.



Figura 2.9: *PIG* palito com 250 sensores.

Hu & Appleton (2005) apresentaram um modelo dinâmico para um novo robô auto-dirigível (ver Fig. 2.10) para dutos, que obtém sua energia do movimento cinético do fluxo de fluido via turbina. Para verificação do modelo proposto foi construído um protótipo e um duto de testes. Resultados mostraram que a turbina impulsionada pelo fluxo é uma fonte confiável de energia e o *PIG* pode ser vista como um sistema de primeira ordem com mudança de parâmetros em diferentes estágios do movimento.

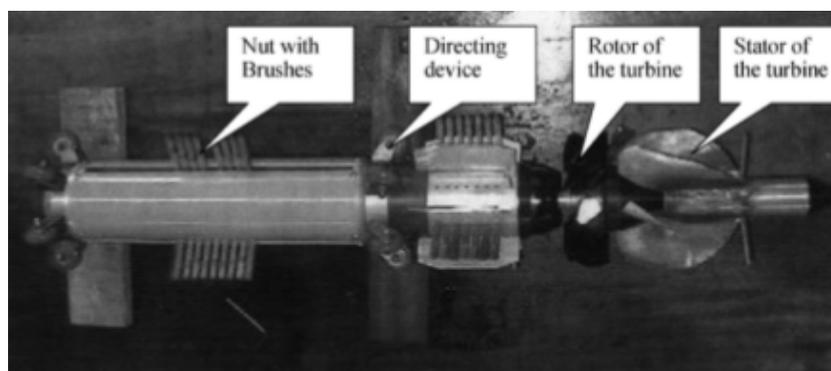


Figura 2.10: Protótipo do *PIG* self-drive.

Panta (2005) monitorou a operação do G.I.R.I.N.O. (Gabarito Interno Robotizado de Incidência Normal ao Oleoduto) na procura por vias menos arriscadas no processo de inspeção interna de dutos. O robô desenvolvido pelo CENPES, para obter informações sobre suas partes a fim de certificar o funcionamento normal desta ferramenta, com objetivo de reconhecer possíveis falhas de operação.

Junior & Lourenço (2006) propuseram a implementação de um robô, provido de câmeras de alta resolução distribuídas ao redor de sua estrutura, para abranger toda a circunferência do duto. As câmeras fornecem imagens, que são processadas digitalmente e gravadas em memória não-volátil. Um *software* realiza a identificação das não-conformidades presentes nas imagens.

O trabalho de Silva & Lima (2006) propôs a identificação, descrição e avaliação das principais técnicas de detecção de vazamentos em gasodutos enterrados. Dentre essas técnicas foi apresentado um *PIG* instrumentado equipado com microfones especiais capazes de detectar sons de 1 kHz a 1 MHz (faixa de frequência onde se encontra o ruído produzido pelo vazamento no gasoduto). Como vantagem deste sistema em relação aos demais: fácil instalação e operação; baixo custo; boa precisão na localização do vazamento, etc.

Salcedo et al. (2007) apresentaram o *PIG* cobra palito (ver Fig. 2.11) como uma ferramenta capaz de superar diversas limitações encontradas nos *PIGs* tradicionais. Montado em uma estrutura flexível, que permitiu a inspeção de dutos com grande variação de diâmetros e curvas de pequeno raio. Os resultados experimentais em campo consolidaram a tecnologia do *PIG* cobra palito como boa alternativa tecnológica para perfilagem da corrosão interna em dutos.



Figura 2.11: *PIG* cobra palito.

Bueno (2007), fazendo uso de dados coletados por *PIGs* instrumentado, desenvolveu uma metodologia que avaliou os diferentes recursos tecnológicos para monitoramento e controle do processo de corrosão em dutos.

Thiago (2008) relata a primeira campanha de inspeção interna de gasoduto com *PIGs* instrumentados tipo MFL, Geométrico e Inercial. Foram inspecionados 2.593 km de dutos de 32", 24", 20", 18" e 16", em aproximadamente 60 corridas de *PIGs*. O trabalho descreveu as etapas de inspeção interna, as indicações registradas pelos *PIGs*, o processo de avaliação de integridade e os reparos executados no gasoduto.

Diaz (2008) apresentou o estudo e o projeto de um robô para inspeção de linhas de serviço. Utilizando uma metodologia de projeto mecânico e mecatrônico, propôs o projeto mecânico e de controle de um novo RIT (Robô de Inspeção de Tubulações) com a capacidade de se descolar com velocidade diferente do fluido de trabalho.

Ainda segundo Diaz (2008) existe um sistema de controle de velocidade para *PIGs* em que um motor de passo aciona um disco perfurado, este permite um maior ou menor fluxo de fluido pelo corpo do *PIG*, controlando assim sua velocidade.

Emmendoerfer et al. (2008) desenvolveram, implementaram e testaram um reconstrutor de trajetórias utilizando medidas de giroscópios e odômetro. Como resultado essa técnica pode ser utilizada como forma alternativa às técnicas comumente encontradas, que derivam de aplicações espaciais e utilizam acelerômetros inerciais.

O trabalho de Addor (2009) investigou o controle de corrosão na parede interna de dutos rígidos em um oleoduto em operação na Bacia de Campos. As taxas de corrosão resultante do Monitoramento de Corrosão Interna (MCI) foram comparados com as inspeções realizadas por *PIGs* instrumentados do tipo MFL e Ultra-som.

Sabino (2009) pesquisou sobre o desempenho da tecnologia de *PIG* de perfilagem na detecção e na exatidão de medição de comprimento, largura e profundidade de perdas de espessura na superfície interna da parede de um oleoduto, em condições normais de inspeção de oleodutos com *PIG*.

Salcedo (2009) apresentou a análise do sistema de sensores utilizados pelo *PIG* instrumentado do tipo Palito usado para detecção e dimensionamento de perda de espessura em dutos. Os resultados obtidos da análise experimental de medição com o sensor de microgeometria indicaram a viabilidade do uso desta técnica como alternativa na medição do ganho ou perda de espessura na superfície interna dos dutos.

Guibin et al. (2011) apresentam um sistema de controle ativo de velocidade inovador para *PIGs*, uma válvula *by-pass*. Foi apresentada a característica estrutural de duas válvulas *by-pass*, uma linear (ver Fig. 2.12(a)) e outra rotativa (ver Fig. 2.12(b)). Análise de resultados teóricos e experimentais mostram tendência semelhante.

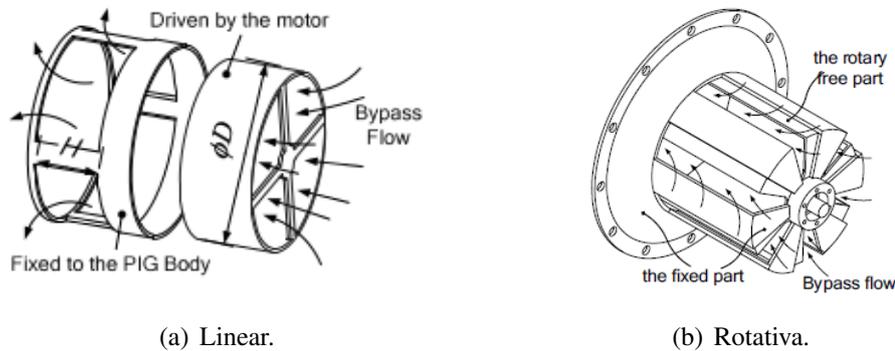


Figura 2.12: Válvulas *by-pass*.

Assim, pela revisão bibliográfica apresentada é possível perceber que muita pesquisa já foi desenvolvida na área de *PIGs*, mas que ainda existe muito a se fazer objetivando melhorar ainda mais tais ferramentas. Dessa forma este trabalho foi desenvolvido com o propósito de estudar o comportamento do diferencial de pressão de modo a simular o movimento de um *PIG*, quando submetido aos eventos de “tiro” e de aceleração, algo diferente do apresentado até o momento na literatura.

2.1.4 Publicações do Grupo de Pesquisa *PIG* da UFRN

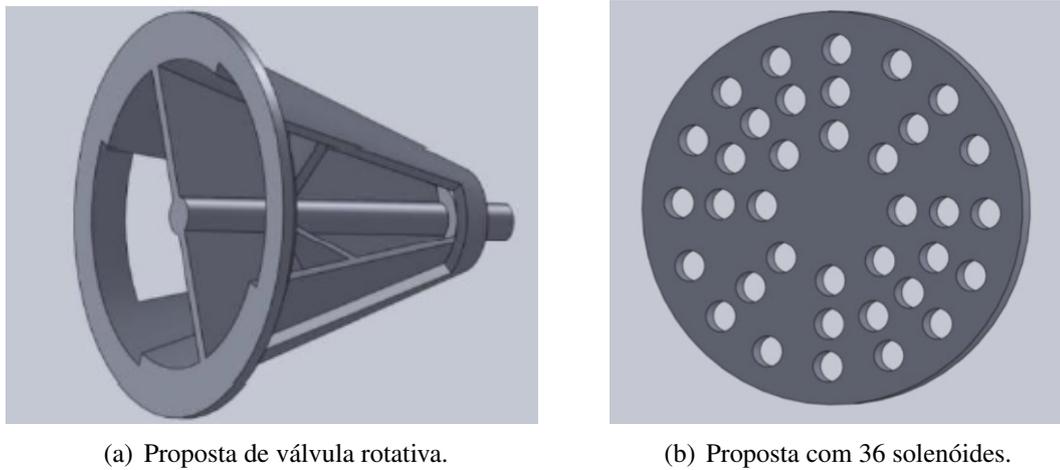
Para contribuir com o estudo de controle de velocidade de *PIGs* instrumentados, esta seção tem por objetivo apresentar os trabalhos desenvolvidos pelo grupo *PIG* da UFRN e apresentados em congressos.

Silva et al. (2012) propuseram um modelo de controle de velocidade para *PIG* utilizando uma válvula *by-pass* (Fig. 2.13(a)) que limita o fluxo de gás fazendo com que o diferencial de pressão mudasse de valor e, assim, a velocidade pudesse ser controlada.

O trabalho de Pereira, Maitelli & Salazar (2013) apresentou o estudo e simulação do controle de velocidade de *PIGs*. Nele a lógica *Fuzzy* foi utilizada para controlar a abertura e fechamento de 36 válvulas solenóides distribuídas simetricamente e atuadas aos pares. A Figura 2.13(b) mostra uma vista frontal da referida válvula.

Em Pereira, de Oliveira, de Lima, Salazar, Maitelli & de Assis O. Fontes (2013) foi proposto a utilização da lógica *Fuzzy* para controlar os tempos de abertura e fechamento de uma nova válvula *by-pass* acionada por solenoide (ver Fig. 2.14) de forma a obter um diferencial de pressão que garante a velocidade desejada.

Em de Lima, de Freitas, Salazar, Maitelli & de Assis O. Fontes (2014) foi proposto a utilização da plataforma Arduino como ferramenta para aquisição de pressões a montante e a jusante da válvula *by-pass* em uma bancada de testes, cálculo do diferencial de pressão

Figura 2.13: Tipos de válvulas *by-pass*.

atuante no *PIG* estático e o controle de abertura e fechamento da válvula *by-pass* proposta.

Em de Lima, de Freitas, de Oliveira, Salazar, Maitelli & de Assis O. Fontes (2014) foram apresentados os resultados de uma estratégia de controle em que o acionamento da válvula *by-pass* teve seus tempos de abertura e fechamento controlado em função do valor do diferencial de pressão. A idéia foi abrir por mais tempo a válvula *by-pass* para valores altos de ΔP e a medida que este ia diminuindo, os tempos de abertura também foram diminuindo.

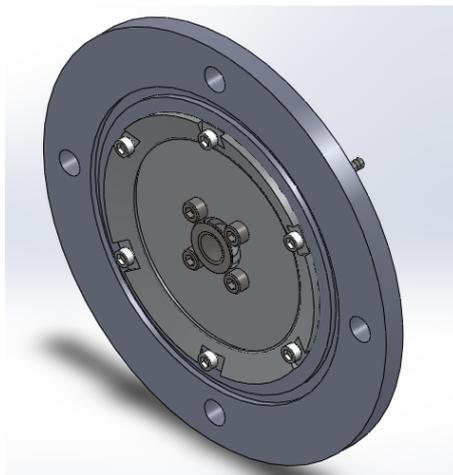


Figura 2.14: Válvula com pistão acionada por solenóide.

Por fim, em de Freitas et al. (2014) foram apresentados mais resultados da viabilidade do controle de velocidade para *PIGs* instrumentados, utilizando uma heurística de controle baseada na modulação PWM.

2.2 Lógica *Fuzzy*

2.2.1 Introdução

Em 1965, o professor Lofti Asker Zadeh da Universidade de Berkeley (USA) publicou o trabalho intitulado “*Fuzzy sets*” (Zadeh 1965) no jornal acadêmico “*Information and Control*”. Ele propôs uma nova teoria de conjuntos, onde a passagem da pertinência para a não pertinência fosse feita de uma forma lenta e gradual e não abrupta como na teoria usual de conjuntos (Bilobrovec et al. 2004). Daí surgiram os Conjuntos *Fuzzy* (nebuloso ou difuso).

A Figura 2.15(a) ilustra um exemplo de passagem abrupta da pertinência para a não pertinência. Um elemento está dentro ou fora desse conjunto, sem existir intervalos entre os extremos. O meio é excluído, de forma a estabelecer uma bivalência (verdadeiro ou falso, quente ou frio, alto ou baixo, pertence ou não pertence, etc).

Já, a Figura 2.15(b) mostra uma forma lenta e gradual de passagem da pertinência para não pertinência. Nessa nova teoria de conjuntos, um elemento pode transitar entre os extremos passando pelos intervalos. O meio não é mais excluído, de forma a estabelecer uma multivalência.

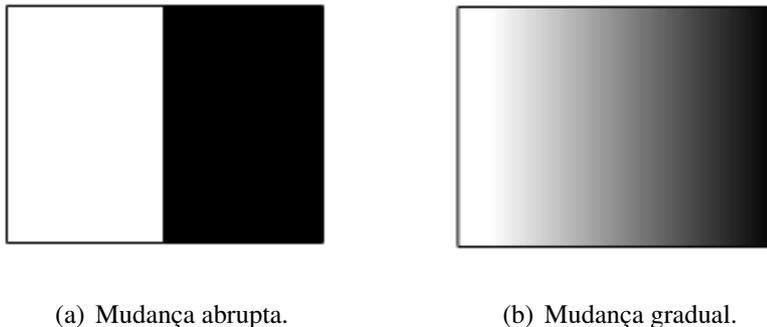


Figura 2.15: Tipos de passagem das pertinências.

Do exposto acima é possível concluir que o mundo real não é bivalente. Nem sempre ter apenas duas possibilidades é suficiente para resolver problemas. Em determinadas situações, valores intermediários seriam soluções melhores (Pereira 2012). O mundo real é analógico, altamente não-linear, com um infinitas possibilidades de opções, ao invés de somente duas.

Assim, a Lógica *Fuzzy*, segundo Filho et al. (2002), é uma técnica de Inteligência Artificial, disciplina que procura maneiras de máquinas simularem o raciocínio humano na solução de problemas diversos. E, como todo método, a Lógica *Fuzzy* também pos-

sui suas vantagens e desvantagens, tais características foram transcritas de Vale (2007) e podem ser visualizadas na Tab. 2.2.

Tabela 2.2: Vantagens e desvantagens da Lógica *Fuzzy*.

Vantagens	Desvantagens
O uso de variáveis lingüísticas (heurísticas)	Necessita de mais experiências e regras precisas
Menor esforço computacional	Necessita do conhecimento de um especialista humano
Simplifica a solução de problemas usando a experiência do especialista humano	Dificuldades de estabelecer regras precisas
Não requer cálculos e equações complicadas	Não há uma definição matemática precisa

Dentro da área de Sistemas de Controle, um dos primeiros relatos da utilização da lógica *Fuzzy* aconteceu, segundo Reyero e Nicolas, citado por Ivanqui (2005), em 1975, quando o professor Mamdani, do Queen Mary College, Universidade de Londres, após inúmeras tentativas frustradas em controlar uma máquina a vapor com tipos distintos de controladores, incluindo o controle proporcional, integral e derivativo, mais conhecido como controlador PID, somente conseguiu fazê-lo através da aplicação do raciocínio *Fuzzy*.

Esse sucesso serviu de alavanca para muitas outras aplicações, por exemplo, os controladores *Fuzzy* de plantas nucleares, refinarias, processos biológicos e químicos, trocador de calor, máquina diesel, tratamento de água e sistema de operação automática de trens (Ivanqui 2005).

Alguns exemplos de aplicações industriais que o controlador *Fuzzy* vem obtendo sucesso são (Vale 2007): simuladores de vôo; máquinas de lavar roupa; geladeira; ar condicionados; injeção eletrônica; e elevadores.

Em algumas dessas tarefas é observado um alto nível de atenção ou tomada de decisão por parte do operador. Nisso reside uma grande vantagem da lógica *Fuzzy* implementada nos controladores, que é a de ser projetado para comportar-se conforme o raciocínio *dedutivo*, isto é, o processo que as pessoas utilizam para inferir conclusões baseadas em informações que elas já conhecem (Shaw & Simões 1999).

Em resumo, a Lógica *Fuzzy* objetiva fazer com que as decisões tomadas pela máquina se aproximem cada vez mais das decisões humanas, principalmente ao trabalhar com uma

grande variedade de informações, vagas e incertas (Vale 2007).

2.2.2 Sistemas de Controle Baseados em Lógica *Fuzzy*

Um sistema de controle de uma forma geral pode ser descrito por um controlador e uma planta ou processo que está sendo controlado, conforme a Fig. 2.16.

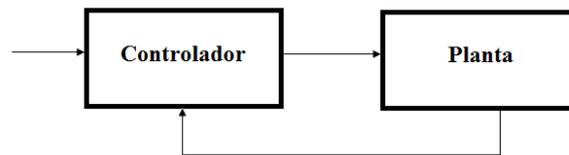


Figura 2.16: Modelo para um sistema de controle.

Segundo Ivanqui (2005) para a confecção de um controlador *Fuzzy*, pode se imaginar que ao invés de usar apenas parâmetros de modelos matemáticos para construir o controlador, pode se visualizar este controle como ilustrado na Fig. 2.17, onde o operador humano, um especialista, teria a responsabilidade de controlar os parâmetros da planta.

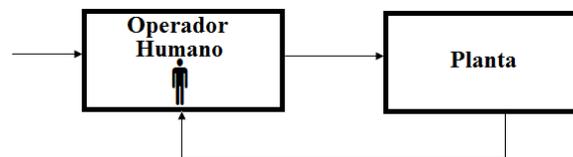


Figura 2.17: Operador humano uma analogia do controlador *Fuzzy*.

Desta maneira, o controlador *Fuzzy* é desenvolvido para automatizar, como um especialista, o gerenciamento do processo. Com isto, o primeiro passo na construção de um sistema de controle *Fuzzy* consiste em adquirir conhecimentos sobre o processo que se quer controlar (Maniçoba 2013). A começar pela definição das variáveis de entrada e saída desse processo.

As variáveis de entrada são aquelas utilizadas pelo operador da planta para realizar uma análise de desempenho do processo e para tomar decisões sobre os próximos passos a seguir, em geral, a escolha dessas variáveis é feita pela experiência desse operador.

As variáveis de saída são as variáveis controladas do processo. Estas são de mais fácil identificação já que na maioria dos casos elas estão relacionadas aos objetivos de controle e são as mesmas utilizadas nos controladores convencionais (Ivanqui 2005).

Depois disso, é preciso especificar a base de conhecimento que forma o núcleo do controlador *Fuzzy*. Quanto mais informações sobre o processo estiver disponível, é grande

a possibilidade de uma boa base de regras ser projetada e, com isso, esperar um bom desempenho do controlador *Fuzzy*.

Em conclusão, para um bom desenvolvimento do controlador *Fuzzy* é necessário que a planta a ser controlada seja bem compreendida e a escolha das variáveis de entrada e saída é parte fundamental desse projeto.

Um sistema baseado em Lógica *Fuzzy* pode ter sua ação esquematizada pelos seguintes elementos constituintes (Filho et al. 2002): *Fuzzificador*; Regras (base de conhecimento); Inferência (lógica de tomada de decisões); e *Defuzzificador*. Esta estrutura pode ser visualizada na Fig. 2.18.

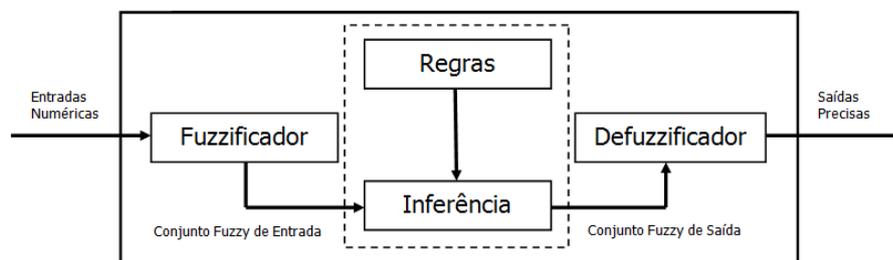


Figura 2.18: Blocos do controlador *Fuzzy*.

A seguir são apresentados os passos que descrevem cada um desses blocos. De certa forma, é um roteiro para o desenvolvimento de um controlador *Fuzzy* baseado em regras descritas por um especialista, cujo objetivo é fazer uso do conhecimento humano, em como controlar a planta, a fim de usar a lógica *Fuzzy* para automatizá-lo.

2.2.3 Fuzzificação ou Codificação

O estágio de *fuzzificação* tem por função transformar os valores numéricos (*crisp*) das entradas em suas respectivas variáveis linguísticas. As entradas numéricas são fornecidas por sensores ou dispositivos computadorizados. Tais entradas são normalizadas em um universo de discurso padronizado, mapeando-os em expressões linguísticas, como “muito quente” ou “pouco quente”, ou ainda, “pressão alta” ou “pressão baixa”, representando um conjunto *Fuzzy*.

Cada conjunto *Fuzzy* é caracterizado por sua função de pertinência, e é através dela que será determinado o quanto um determinado elemento pertence ao conjunto. O fator de pertinência pode então assumir qualquer valor entre “0” e “1”, sendo que o “valor 0” indica uma completa exclusão e um “valor 1” representa uma completa pertinência (Maniçoba 2013).

A Figura 2.19 apresenta uma função de pertinência em que no eixo y os valores variam de “0” a “1”, podendo então uma variável qualquer transitar dentro desse intervalo. E no eixo x o intervalo de valores vai de “0” a “10”, caso fosse uma variável como pressão, esta poderia variar de 0 a 10 psi ou de 0 a 10 bar, por exemplo.

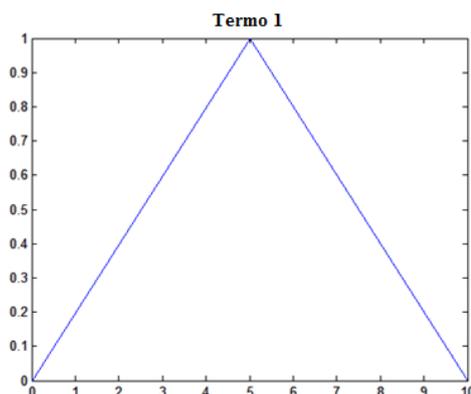


Figura 2.19: Exemplo de função de pertinência.

Em resumo, a interface de *fuzzificação* recebe as variáveis numéricas de entrada, realiza uma normalização para um universo de discurso, transformando-os em valores *fuzzificados* de um conjunto *Fuzzy*.

2.2.4 Regras (Base de Conhecimento)

A *base de conhecimento* representa o modelo do sistema a ser controlado, consistindo em uma base de dados e uma base de regras *Fuzzy* linguísticas, conforme a Fig. 2.20. A base de dados fornece definições numéricas e a base de regras caracteriza os objetivos do controlador e sua estratégia usada, geralmente fornecida por pessoas especialistas no sistema (Filho et al. 2002).

A base de regras tem a função representar de forma organizada a maneira como o controlador gerenciará o sistema. A forma mais geral de uma regra linguística é:

SE premissa **ENTÃO** consequência

As premissas são associadas com as entradas do controlador *Fuzzy*, e formam a parte das regras representada à esquerda, enquanto as consequências estão associadas às saídas dos controladores, e são conhecidas como ações de controle, conforme Campos Filho citado por Maniçoba (2013). As premissas também podem ser chamadas de antecedentes. Enquanto que as consequências podem ser chamadas também de consequentes.

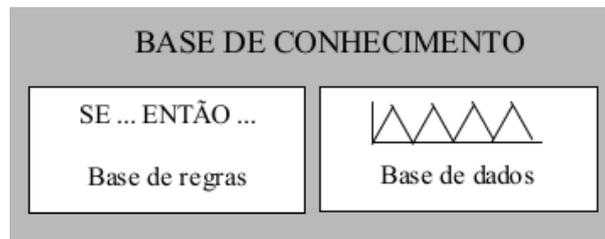


Figura 2.20: Base de conhecimento.

Com relação às variáveis linguísticas do processo é possível associar as premissas com ações de controle. Um exemplo da utilização de regras é o controle de iluminação. É praticável estabelecer uma relação entre iluminação de um *LED*, com o brilho desse mesmo *LED*, o que produz as seguintes regras:

SE iluminacao = escuro **ENTÃO** Brilho do LED = elevado

SE iluminacao = normal **ENTÃO** Brilho do LED = medio

SE iluminacao = Iluminado **ENTÃO** Brilho do LED = baixo

Esse exemplo, aparentemente simples, é capaz de manter estável a iluminação de um *LED*, uma vez que os conjuntos *Fuzzy* envolvidos foram definidos de maneira condizente com a aplicação.

Este conjunto demonstra que a base de regras *Fuzzy* é bastante intuitiva, do ponto de vista humano. Não existe uma formulação matemática explícita envolvida na solução do problema, mas sim a representação de um conhecimento adquirido pelo operador. Portanto é este conhecimento - o da experiência - que as regras *Fuzzy* representam. (Ivanqui 2005).

Em resumo, a Base de Conhecimento é formado por uma base de dados e uma base de regras. Estes contem todo o conhecimento de como controlar o processo.

2.2.5 Inferência

Segundo Shaw & Simões (1999) a lógica de tomada de decisões, incorporada na estrutura da *inferência* da base de regras, usa implicações *Fuzzy* para simular tomadas de decisão humana. Ela gera ações de controle - consequentes - inferidas a partir de um conjunto de condições de entrada - antecedentes.

Como no exemplo, SE a iluminação está escura ENTÃO o brilho do *LED* deve ser elevado. Como o objetivo é manter a iluminação normal, então foi realizada uma inferência

para determinar a ação que será realizada para se alcança a iluminação desejada, que foi aumentar o brilho do *LED* para elevado.

Existem basicamente dois modelos de máquina de inferência: a Mamdani (Mamdani e Assilan citado por de Lima (2011) e Takagi-Sugeno (Takagi e Sugeno, citado por de Lima (2011)). Elas são diferentes nas habilidades de representar diferentes tipos de informação. No primeiro tipo, a base de regras é estritamente linguística, enquanto que, no segundo tipo são equações paramétricas que relacionam as entradas e saídas do processo.

No modelo Mamdani, os antecedentes como os consequentes são mapeados por conjuntos linguísticos. Para cada regra de inferência é preciso aplicar uma técnica de agregação dos conjuntos antecedentes, com o objetivo de gerar um conjunto consequente. Se existir “n” regras, serão produzidos “n” conjuntos consequentes, que são combinados.

A técnica mais comum na composição dos vários conjuntos *Fuzzy* de entrada para cada regra é o método de inferência MAX-MIN (de Lima 2011). O “MIN” implica em um conectivo “E” e o “MAX” em um conectivo “OU”. O conectivo “E”, chamado de operação de agregação, resulta na interseção *Fuzzy* dos termos de entrada. O conectivo “OU”, chamado de operação de composição, resulta na união dos termos de saída.

2.2.6 Defuzzificação ou Decodificação

Segundo Filho et al. (2002) o *defuzzificador* mapeia valores linguísticos em valores numéricos de saída. Esta função é realizada por uma interface de *defuzzificação*, obtendo-se um valor discreto que possa ser usado numa ação de controle no mundo real. Assim, a *defuzzificação*, segundo (Shaw & Simões 1999), é uma transformação inversa que traduz a saída do domínio *Fuzzy* para o domínio discreto.

Os seguintes métodos são muito utilizados (Shaw & Simões 1999) :

- **Centro-da-Área (C-o-A):** Este método também é chamado de método *Centro-de-Gravidade*, pois ele calcula o *centróide* da área composta que representa o termo de saída *Fuzzy* (μ_{OUT}), esse termo é composto pela união de todas as contribuições de regras. O centróide é um ponto que divide a área de μ_{OUT} em duas partes iguais. A Equação 2.8 é a fórmula utilizada para esse cálculo.

$$u^* = \frac{\sum_{i=1}^N u_i \mu_{out}(u_i)}{\sum_{i=1}^N \mu_{out}(u_i)} \quad (2.8)$$

onde: u_i é a posição do centróide da função de pertinência individual; e μ_{OUT} é a área de uma função de pertinência.

- **Centro-do-Máximo (C-o-M):** Neste método são considerados os picos das funções de pertinência, enquanto que, as áreas dessas funções são ignoradas. A saída discreta é calculada como uma média ponderada dos máximos, cujos pesos são os resultados da inferência e realizado através da seguinte Eq. 2.9. Este método também pode ser chamado de *defuzzificação* pelas alturas.

$$u^* = \frac{\sum_{i=1}^N u_i \cdot \sum_{k=1}^n \mu_{o,k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{o,k}(u_i)} \quad (2.9)$$

onde: $\mu_{o,k}(u_i)$ indicam os pontos em que ocorrem os máximos das funções de pertinências de saída.

- **Média-do-Máximo (M-o-M):** Neste método é preciso encontrar o meio dos valores da função pertinência que são máximos e pode ser calculado através da Eq. 2.10.

$$u^* = \frac{1}{M} \sum_{m=1}^M u_m \quad (2.10)$$

onde: u_m é o m-ésimo elemento onde se obtenha o máximo da função de pertinência; e M é o número total desses elementos.

Após aplicar a fórmula do método escolhido, será obtida a saída real, ou seja, o valor que deverá ser aplicado na planta para conseguir o resultado esperado no controle do sistema (Vale 2007).

2.3 Conclusões

Neste capítulo foram apresentadas as principais características sobre as ferramentas *PIGs* para inspeção de dutos, seus tipos e aplicações, também. Foram apresentados, também, os artigos científicos do Grupo *PIG* da UFRN apresentados em congressos. E por fim, a Lógica *Fuzzy* foi apresentada como opção para implementação de controlador inteligente para atuar no controle de velocidade de *PIGs* instrumentados.

Capítulo 3

Implementação do Sistema

Neste capítulo são abordados os principais componentes e sub-sistemas para implementação/montagem de uma bancada de testes, com o objetivo de realizar simulações de situações de conhecidas por “tiro” e aceleração sofridos pelos *PIGs* instrumentados.

3.1 Plataforma Arduino

O Arduino surgiu em 2005, no *Interaction Design Institute* na cidade de Ivrea, na Itália, com um professor chamado Massimo Banzi, que queria ensinar eletrônica e programação de computadores a seus alunos de *design*, para que eles usassem em seus projetos de arte, interatividade e robótica (de Robótica 2012). Porém, dois grandes desafios existiam pela frente: ensinar eletrônica e programação para pessoas que não eram dessas áreas; e a inexistência de placas baratas e poderosas disponíveis no mercado.

O professor Massimo Banzi discutiu esse problema com David Cuartielles e juntos decidiram criar sua placa própria, o Arduino. Segundo Evans et al. (2013) as principais exigências eram que fosse barato (o preço almejado não poderia ser mais do que o que um estudante gastaria se saísse para comer uma pizza) e que fosse uma plataforma que qualquer pessoa pudesse utilizar.

Eles, ainda, contaram com a ajuda de um aluno de Massimo, David Mellis, que ficou responsável por criar a linguagem de programação que controlaria a placa Arduino. E Massimo contratou um engenheiro local, Gianluca Martino, para produzir uma tiragem inicial de duzentas placas.

As placas eram vendidas em forma de kit para que os alunos fizessem seus próprios projetos. E, segundo Evans et al. (2013), sua popularidade cresceu rapidamente quando o grande público percebeu que o Arduino era um sistema de fácil utilização, de baixo custo e que poderia ser usado em seus próprios projetos, bem como era uma excelente introdução para programação de microcontroladores.

Atualmente, a plataforma Arduino conta mais de 11 modelos diferentes de placas, alguma delas são: Arduino Uno, Arduino Duemilanove, Arduino Ethernet, Arduino Mega, LilyPad Arduino, Arduino Nano, entre outras. A Figura 3.1 mostra alguns exemplos de placas diferentes que constituem a plataforma Arduino.

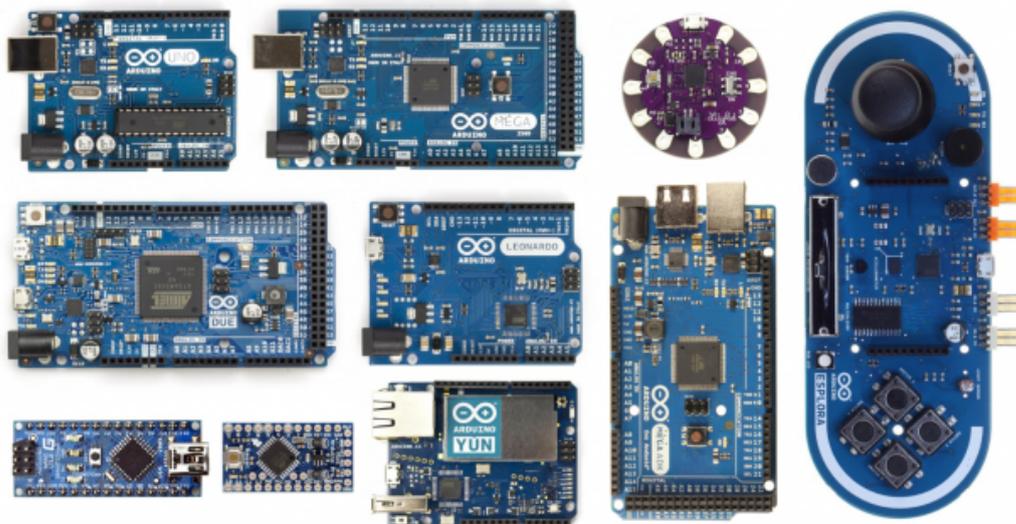


Figura 3.1: Diferentes placas da plataforma Arduino.

Fonte: <http://blog.filipeflop.com/>.

Então, para reforçar a idéia, Arduino é uma plataforma de prototipagem eletrônica destinada a artistas, *designers*, *hobbistas*, ou qualquer pessoa interessada em criar objetos ou ambientes interativos (Banzi et al. 2005). É uma pequena placa de circuito impresso e seu projeto é baseado nas filosofias de *hardware* e *software* livres, *open sources*.

Os projetos *open sources* segundo Gomes (2011) são aqueles criados para o domínio público, ou seja, que podem ser copiados e modificados por outras pessoas conforme suas necessidades e depois colocados de volta ao domínio público de modo que outros usuários possam usufruir dessas mudanças em seus próprios projetos.

Segundo Banzi et al. (2005) a placa Arduino pode interagir com o ambiente recebendo em suas entradas sinais dos mais variados tipos de sensores e afetar ao seu redor acionando luzes, motores ou outros atuadores.

Para o presente projeto foi escolhido o modelo Arduino Uno pelos seguintes motivos: possui conector USB tipo B para conexão com um computador, conector jack 4 para alimentação externa, programação simples, preço baixo e fácil manuseio. Uma placa Arduino Uno, também, conta com um microcontrolador ATmega328P (fabricado pela empresa Atmel), 14 entradas e saídas digitais, 6 entradas analógicas, 1 oscilador de cristal

de 16 MHz e pinos de alimentação com 3,3 V, 5 V e Terra (GND). Toda essa estrutura pode ser visualizada na Fig. 3.2, onde os principais blocos da placa foram identificados.

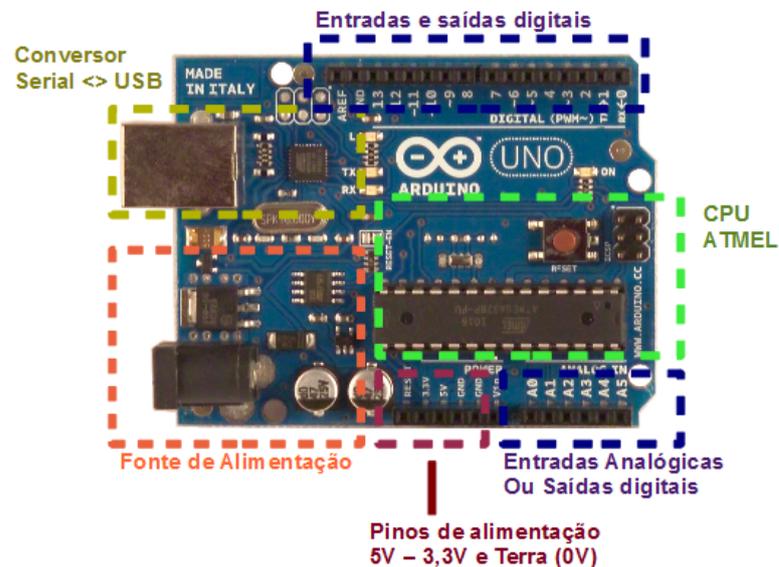


Figura 3.2: Blocos identificados do Arduino Uno.

Fonte: www.robotizando.com.br.

Para Fonseca et al. (2010) um microcontrolador (também denominado MCU) é um computador em um *chip*, que contém processador, memória e periféricos de entrada/saída. É um microprocessador que pode ser programado para funções específicas, em contraste com outros microprocessadores de propósito geral (como os utilizados nos computadores pessoais).

O ATmega328P é um microcontrolador AVR 8 bits de baixa potência e arquitetura RISC (*Reduced Instruction Set Computing*) avançada, que executa a maioria das 131 instruções em um único ciclo (Gomes 2011). Ele possui 28 pinos, 32 kB de memória Flash, 2 kB de memória RAM e 1 kB de EEPROM. E conta com um *bootloader*, que dispensa o uso de programadores para o *chip*, facilitando ainda mais o seu uso uma vez que não exige compiladores ou *hardware* adicional (Souza et al. 2011).

Outra vantagem da plataforma Arduino reside na utilização de placas adicionais que expandem suas funções. Elas são chamadas de *shields* (do inglês escudo) e são instaladas sobre a placa Arduino. Um exemplo de expansão de funcionalidade do Arduino é o *Ethernet Shield* mostrado na Fig. 3.3. Ele permite que o Arduino se conecte à Internet por meio de um roteador, utilizando o conector RJ45. Este modelo de *shield* ainda conta com um conector para cartão de memória micro SD, que permite a funcionalidade de armazenamento de dados para o Arduino.



Figura 3.3: Ethernet shield

A programação do Arduino é feita por meio de uma linguagem de programação própria, chamada de *Wiring* baseada em C/C++, que é implementada em um ambiente de desenvolvimento integrado (IDE), também próprio, baseado em *Processing*, e que pode ser utilizada em vários sistemas operacionais (Gioppo et al. 2009).

A Figura 3.4 mostra uma tela da IDE de programação do Arduino. Nela é possível visualizar uma barra de *menus*, outra barra com botões e uma área para digitação de códigos fontes.

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

Done compiling.
C:\Users\gustavo\AppData\Local\Temp\build4304263987998654201.tmp\Blink.cpp.hex
Binary sketch size: 1018 bytes (of a 32256 byte maximum)
3
```

Figura 3.4: Tela para programação do Arduino.

A configuração da IDE acontece em dois passos. No primeiro, a porta de comunicação (COM) na qual a placa está conectada deve ser escolhida, utilizando os seguintes comandos: clicar em *Tools*, depois em *Serial Port* e, por fim, em COM3 (ver Fig. 3.5).

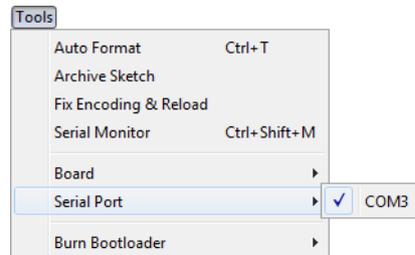


Figura 3.5: Configuração da porta serial.

No segundo passo o modelo de placa Arduino que está conectada na porta COM deve ser selecionada, usando os seguintes comandos: clicar em *Tools*, apontar para *Board* e depois escolher a placa Arduino Uno, como na Fig. 3.6.

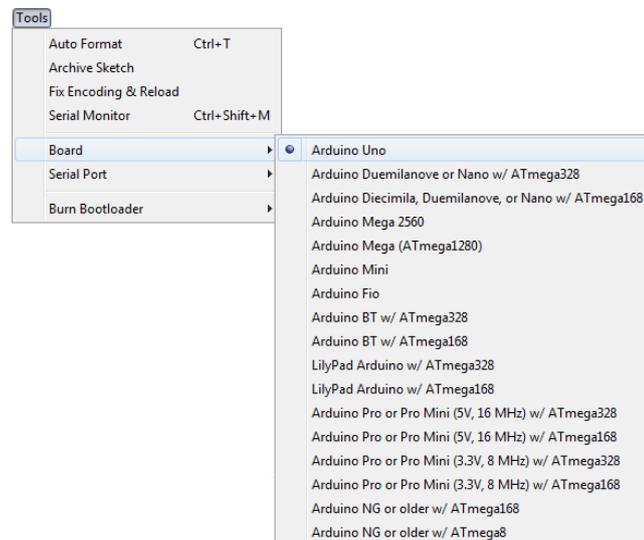


Figura 3.6: Configuração da placa Arduino Uno.

Depois da IDE estar configurada é possível digitar o código do programa que se deseja enviar para o Arduino. A programação do Arduino é feita por meio de duas funções básicas: a *void setup()* e a *void loop()*. A primeira é executada somente uma vez e carrega todos os periféricos que deverão funcionar durante a execução do programa e a segunda será executada infinitamente.

Como exemplo de programação a Fig. 3.4 mostra o programa *Blink*. Nele é possível ver o *setup* configurando a porta digital 13 como saída. Depois, o laço *loop* tem uma

sequência de comandos que realiza uma escrita na porta 13 com valor HIGH (nível lógico um), é realizada uma espera de 1.000 milissegundos, depois ocorre uma escrita na porta 13 com valor LOW (nível lógico zero) e outra espera de 1.000 milissegundos. De maneira tal que o *LED* liga um segundo e depois apaga por um segundo, o que o faz piscar.

E por fim, um último destaque da plataforma Arduino diz respeito ao uso das portas PWM. Segundo Souza (2014) a técnica PWM é utilizada para variar o valor médio de uma forma de onda periódica. Ela consiste em manter a frequência de uma onda quadrada fixa e variar o tempo que o sinal fica em nível lógico alto. Esse tempo é chamado de *duty cycle* ou ciclo ativo, da forma de onda.

No Arduino, uma porta PWM assume valores de 8 bits ($2^8 = 256$), ou seja, seu acionamento pelo comando `analogWrite('argumento')` necessita que seu argumento seja um valor de 0 a 255. Esta faixa de valores corresponde aos valores de 0 a 5 V. A tensão de saída de uma porta PWM pode ser obtida utilizando a Eq. (3.1).

$$V_{out} = \frac{duty\ cycle}{100} \cdot V_{cc} \quad (3.1)$$

Em resumo, a modulação PWM funciona da seguinte forma: quando o valor do *duty cycle* varia, o valor médio da saída também varia. A Figura 3.7 apresenta algumas modulações PWM. Nela é possível observar três momentos distintos de um sinal PWM. No início, o *duty cycle* assume o valor de 50%, então a tensão de saída assume o valor de 50% da tensão máxima, ou 2,5 V. Depois, o *duty cycle* foi modificado para 10%, então o valor médio da saída foi para 0,5 V. E por fim, quando o *duty cycle* assumiu o valor de 90% a saída assumiu o valor 90% da tensão máxima, neste caso 4,5 V.

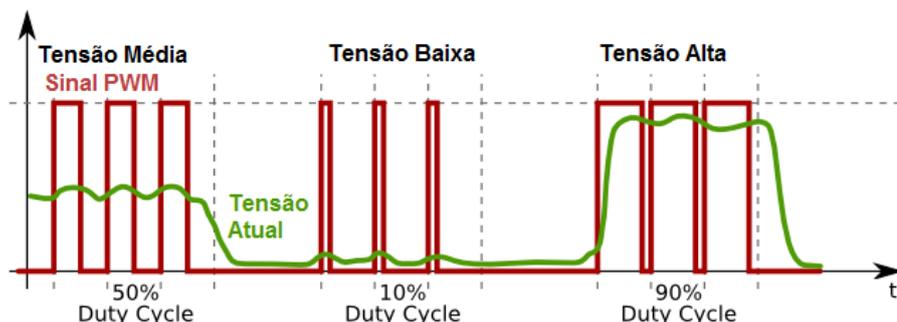


Figura 3.7: Exemplo de modulação por largura de pulso.

Fonte: <https://www.wayneandlayne.com/projects/video-game-shield/design/>.

Um sinal PWM pode ser usado em inúmeras aplicações, como por exemplo (Souza 2014): controle de velocidade de motores, variação da luminosidade de *LEDs*, geração de

sinais analógicos, geração de sinais de áudio, etc.

Neste projeto, uma porta PWM foi utilizada para controlar o tempo em que a válvula *by-pass* permaneceu aberta, com o objetivo de obter uma saída média variável. A válvula *by-pass* possui somente dois estágios de funcionamento: aberto (ou 100%) e fechado (ou 0%). Dessa forma, o uso de uma porta PWM possibilitou a obtenção de valores entre 0 e 100% para a abertura da válvula.

3.2 Dispositivo de Aquisição de Dados (USB-6008)

Um dispositivo de aquisição de dados ou dispositivo DAQ (do inglês, Data Acquisition), segundo Gesteira (2014), atua como a interface entre o computador e os sinais do mundo exterior. Ele digitaliza os sinais analógicos de entrada, de forma que o PC possa lê-los ou interpretá-los.

Os três principais componentes de um dispositivo DAQ são (NI 2013): os circuitos eletrônicos internos de condicionamento de sinal, o conversor analógico digital e o barramento de comunicação com o computador.

Para a aquisição de dados deste projeto foi selecionado o dispositivo NI USB-6008 da empresa *National Instruments*, conforme a Fig. 3.8. A NI USB-6008 é um dispositivo DAQ de baixo custo, fácil conexão com PC utilizando uma porta USB e de dimensões reduzidas (NI 2013).



Figura 3.8: Dispositivo de aquisição de dados USB-6008.

Um dispositivo NI USB-6008 é simples o suficiente para medições rápidas e versátil

o suficiente para aplicações complexas. Ele conta com (NI 2013): 8 entradas analógicas *single-ended* (11-bit, até 10 Hz e ± 10 V); 4 entradas analógicas diferenciais (12-bit, 10 kHz e ± 20 V, ± 10 V, ± 5 V, ± 4 V, $\pm 2,5$ V, ± 2 V, $\pm 1,25$ V, ± 1 V); 2 saídas analógicas (12-bit, até 150 Hz e 0 a 5 V); 12 entradas e saídas digitais; 1 contador de 32 bits; 1 porta USB; e compatibilidade com o *software* LabVIEW™.

3.3 Transdutor de Pressão

O termo transdutor, conforme da Silva (2013), é definido como um dispositivo que realiza a conversão de uma forma de energia (pressão, por exemplo) em outra (tensão elétrica, por exemplo). O transdutor deve responder somente a variações da forma de energia que se pretende medir, com a exclusão de outras.

Neste projeto foram utilizados dois transdutores de pressão, identificados como PT-01 e PT-02. Um a montante da válvula *by-pass* e outro a jusante dela. Eles são instrumentos da marca Ashcroft do modelo A2X e medem pressões de 0 a 1.000 psi (ver Fig. 3.9).



Figura 3.9: Transdutor A2X utilizado.

Fonte: www.ashcroft.com.

Os transdutores foram alimentados com tensão de 15 V e produziram em suas saídas sinais de tensão compatíveis com o módulo de aquisição de dados. O esquema de ligação elétrica do transdutor pode ser visualizado na Fig. 3.10.

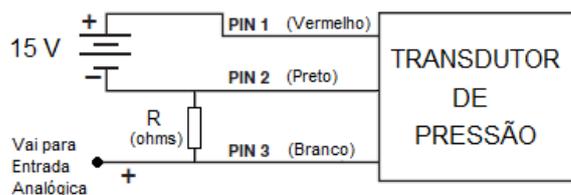


Figura 3.10: Esquema elétrico implementado.

A calibração realizada nos transdutores ocorreu no Laboratório de Metrologia da UFRN . Nos testes foram aplicadas pressões de 0 a 450 psi e as tensões lidas variaram de 0 a 2,255 V. Tais resultados podem ser visualizados na Tab. 3.1.

Tabela 3.1: Calibração do transdutores de pressão.

P (psi)	20	40	60	80	100	...	450	R (ohms)
PT-01(volts)	0,103	0,2047	0,301	0,402	0,506	...	2,255	470
PT-02(volts)	0	0,092	0,194	0,291	0,394	...	2,150	570

Para conversão de tensão em pressão foi necessário linearizar os pontos da Tab. 3.1. Foi utilizado para isso o *software* Matlab™ rodando o comando *cftool*.

A Figura 3.11 apresenta o resultado da linearização para o transdutor PT-01. Nela é possível visualizar na janela 'Results' os valores $p_1 = 199,9$ e $p_2 = -0,5472$.

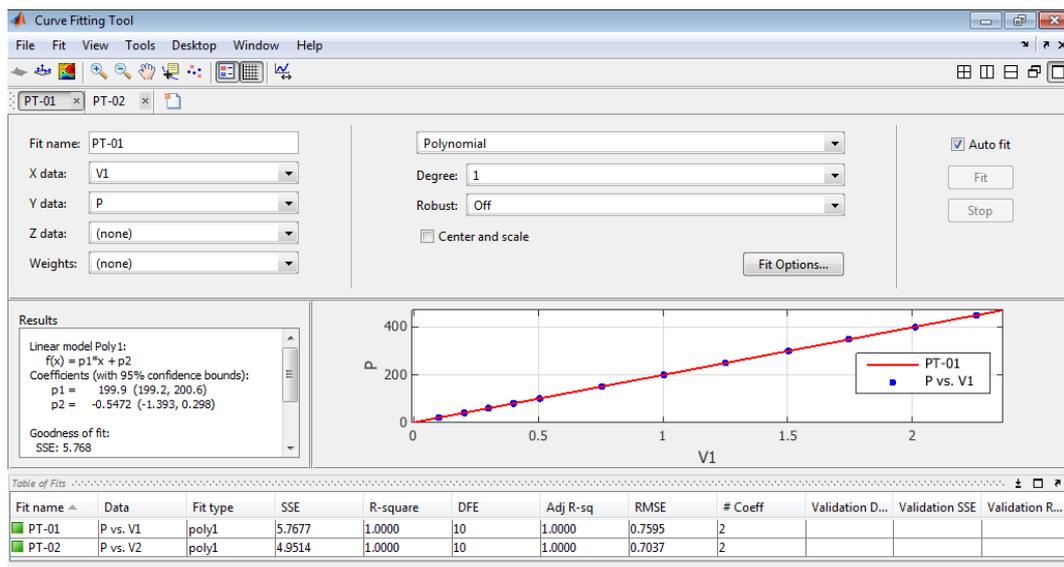


Figura 3.11: Linearização de pontos do transdutor PT-01.

Dessa forma a conversão de tensão para pressão utilizou a Eq. (3.2).

$$pt_{01}(v_1) = 199,9 \cdot v_1 - 0,5472 \quad (3.2)$$

Da Equação (3.2), a variável v_1 foi isolada para obtenção da faixa de tensões que seriam produzidas em função das pressões lidas no transdutor PT-01, conforme a Eq. (3.3).

$$v_1(pt_{01}) = 0,005 \cdot pt_{01} + 0,00274 \quad (3.3)$$

A faixa de trabalho utilizada neste projeto foi de 0 a 100 psi, o que produziu uma faixa de tensões de 0,00274 a 0,502 V, para o transdutor PT-01.

Em seguida, a Fig. 3.12 apresenta o resultado da linearização para o transdutor PT-02. Nela é possível visualizar na janela 'Results' os valores $p_1 = 199,4$ e $p_2 = 21,09$.

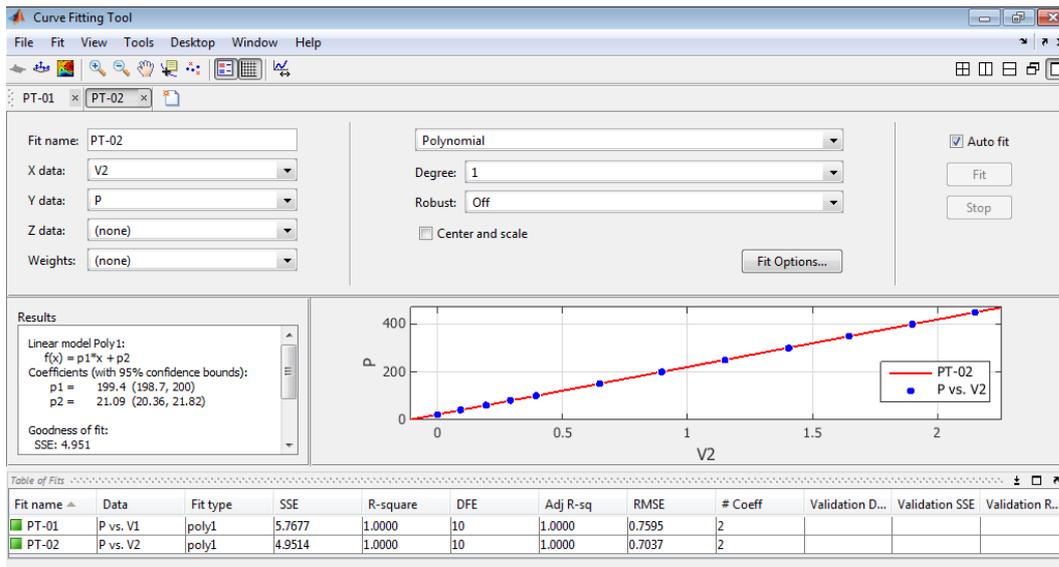


Figura 3.12: Linearização de pontos do transdutor PT-02.

Dessa forma a conversão de tensão para pressão utilizou a Eq. (3.4).

$$pt_{02}(v_2) = 199,4 \cdot v_2 + 21,09 \quad (3.4)$$

Da Equação (3.4), a variável v_2 foi isolada para obtenção da faixa de tensões que seriam produzidas em função das pressões lidas no transdutor PT-02, conforme a Eq. (3.5).

$$v_2(pt_{02}) = 0,005 \cdot pt_{02} - 0,105 \quad (3.5)$$

A faixa de trabalho utilizada neste projeto foi de 0 a 100 psi, o que produziu uma faixa de tensões de -0,105 a 0,395 V, para o transdutor PT-02.

Em função desta faixa de valores negativos apresentados pelo transdutor PT-02 foi necessário a utilização do dispositivo DAQ USB-6008, uma vez que este realiza a medição de tensão com valores negativos, lembrando que ele mede de - 10 V até + 10 V, nas suas entradas analógicas. Ao contrário da placa Arduino, que só lê tensões positivas.

3.4 Válvula Solenóide e seu Acionamento

A válvula solenóide é um componente eletromecânico responsável pela abertura e fechamento da passagem de gás pela válvula *by-pass*, conforme Fig. 3.13. Ela é responsável por transferir parte da pressão a montante da válvula *by-pass* para uma câmara interna que empurra um pistão. Este movimento permite ou não a passagem de gás pela válvula *by-pass*.

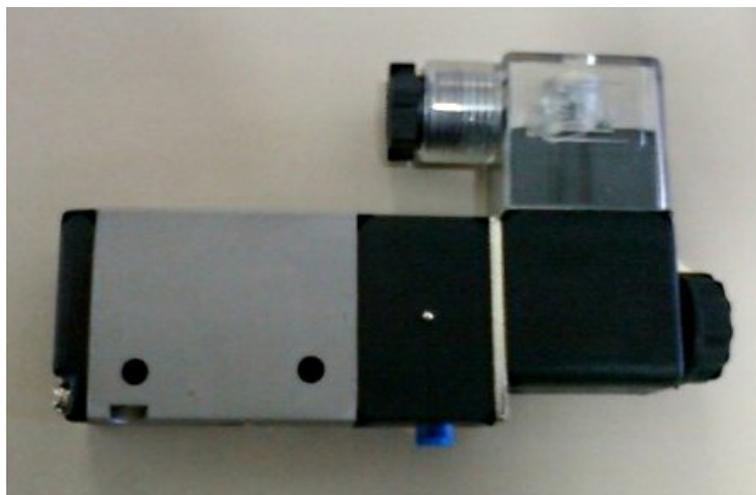


Figura 3.13: Vista lateral da válvula solenóide.

Fonte: www.jelpc.com.

Essa válvula é do tipo 3 vias e 2 posições com acionamento por solenóide, retorno por mola e normal fechada (do inglês Normal Close ou NC). A condição NC garante o fechamento da válvula *by-pass* num momento de falta de energia ou pane eletrônica. E estando fechada garantirá o movimento do *PIG* no momento de um “tiro”.

O funcionamento da válvula solenóide pode ser visualizado na Fig. 3.14. Nela a válvula está inicialmente em repouso, com isso não passa pressão de 'P' para 'A' (Fig. 3.14(a)). No momento da energização com $12 V_{DC}$, o eletroímã existente em sua estrutura movimenta um êmbolo interno para esquerda (Fig. 3.14(b)). Isso possibilita a passagem de pressão de 'P' para 'A', permitindo o acionamento do pistão interno da válvula *by-pass*. No momento da desenergização da solenóide entra em ação uma mola que empurra de volta o êmbolo para a direita ou posição de repouso, cortando a passagem de pressão entre 'P' e 'A' (Fig. 3.14(c)). Com isso, a pressão residual existente em 'A' retorna por 'R'.

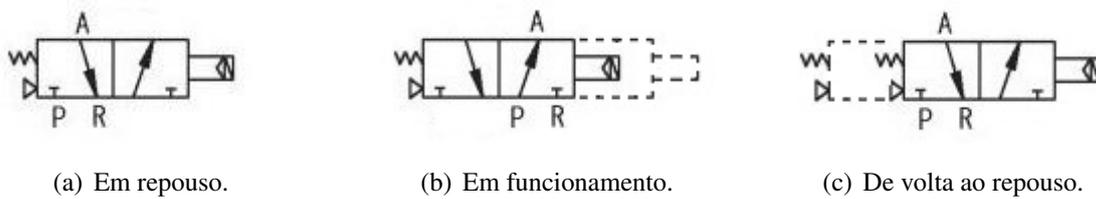


Figura 3.14: Posições da válvula solenóide.

As especificações técnicas desta válvula podem ser visualizadas na Tab. 3.2.

Tabela 3.2: Especificações técnicas da válvula solenóide.

Modelo	3V210-08
Entrada/Saída	1/4"
Escape	1/8"
Área Equivalente	16 mm ²
Coefficiente de Vazão	0,89
Fluído	Ar Filtrado (40 μm)
Pressão	0,15 a 0,8 MPa
Pressão Máxima	1,2 MPa
Temperatura	5 a 50 °C
Classe de Proteção	IP 65
Tensão	12 Vdc
Corrente Elétrica	400 mA
Potência Consumida	4,8 W
Frequência	5 Hz
Tempo de Resposta	0,05 s

Para o acionamento da válvula solenóide foi preciso primeiro consultar a Tab. 3.2. Nela foi possível notar que a corrente consumida pela válvula solenóide é 400 mA. Tal valor de corrente é oito vezes superior a máxima corrente fornecida pela porta digital da placa Arduino, aproximadamente 50 mA. Isto poderia provocar um defeito na placa, podendo até queimá-la.

Para que isso não acontecesse foi utilizado um circuito externo ao Arduino, com a finalidade de ligar e desligar a solenóide, mas que ainda fosse comandado pelo Arduino. Para tanto foi montado um circuito com transistor Darlington NPN modelo TIP 122, conforme

Fig. 3.15(a). Para Boylestad & Nashelsky (2009) a característica principal da conexão Darlington é que a composição de transistores atua como uma unidade única, com um ganho de corrente que é o produto dos ganhos de corrente dos transistores individuais.

A Figura 3.15(b) apresenta o esquema elétrico de ligação da válvula solenóide. Nela a bobina da solenóide foi representada pelo símbolo de relé (RL1) que só será acionada quando existir corrente na base do transistor Darlington (Q1). Nesse momento também ocorrerá o acendimento do LED (D2) como indicação visual do acionamento da solenóide.

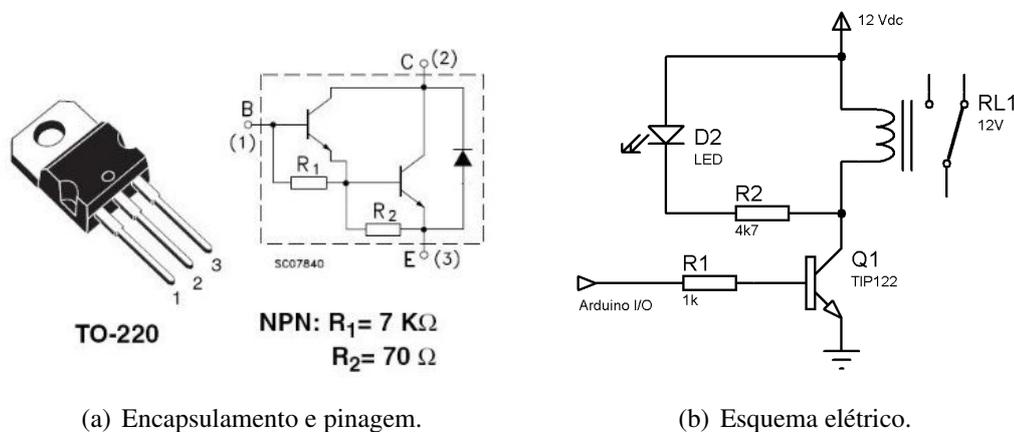


Figura 3.15: Transistor NPN TIP 122.

E para proteção contra altas tensões em decorrência da desenergização da bobina da solenóide é que existe internamente no TIP 122 um diodo de roda livre, ligando os terminais coletor C e emissor E, conforme Fig. 3.15(a).

3.5 Válvula *By-pass*

A válvula *by-pass* utilizada neste projeto tem por função controlar a passagem de fluxo de gás pelo corpo do *PIG*. Significa que ela pode controlar o diferencial de pressão a jusante (na frente) do mesmo. Como consequência deste controle é possível controlar a velocidade de um *PIG*.

A Figura 3.16 apresenta um esquema em corte da válvula *by-pass* proposta. É possível visualizar uma passagem principal de fluxo e o pistão que controla a abertura e fechamento dessa passagem principal. Na montagem da válvula foram deixados 4 orifícios de 2,5 mm de diâmetro e outros 2 de 3,5 mm, como passagens principais de gás.

Outra característica sobre a válvula *by-pass* proposta, é que ela aproveita uma pequena parte de pressão do próprio duto (lado montante da válvula) para o acionamento do pistão.

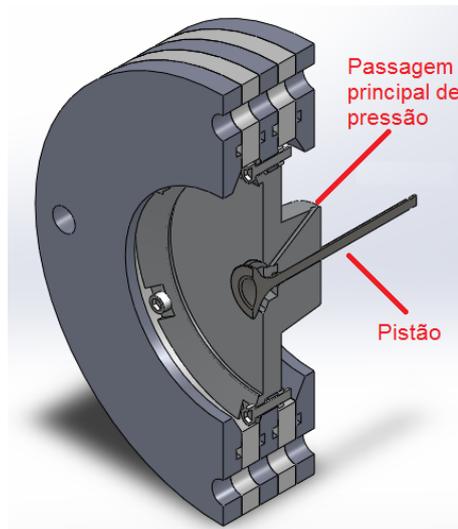


Figura 3.16: Vista em corte da válvula *by-pass*.

A válvula *by-pass* proposta compõe um conjunto de componentes formado por: um pistão, que abre e fecha a passagem principal de gás; um prolongador, onde fica a mola que empurra de volta o pistão para sua posição de repouso; uma solenóide (de cor preta) que controla o movimento do pistão; e por mangueiras (cor azul), conforme a Fig. 3.17.

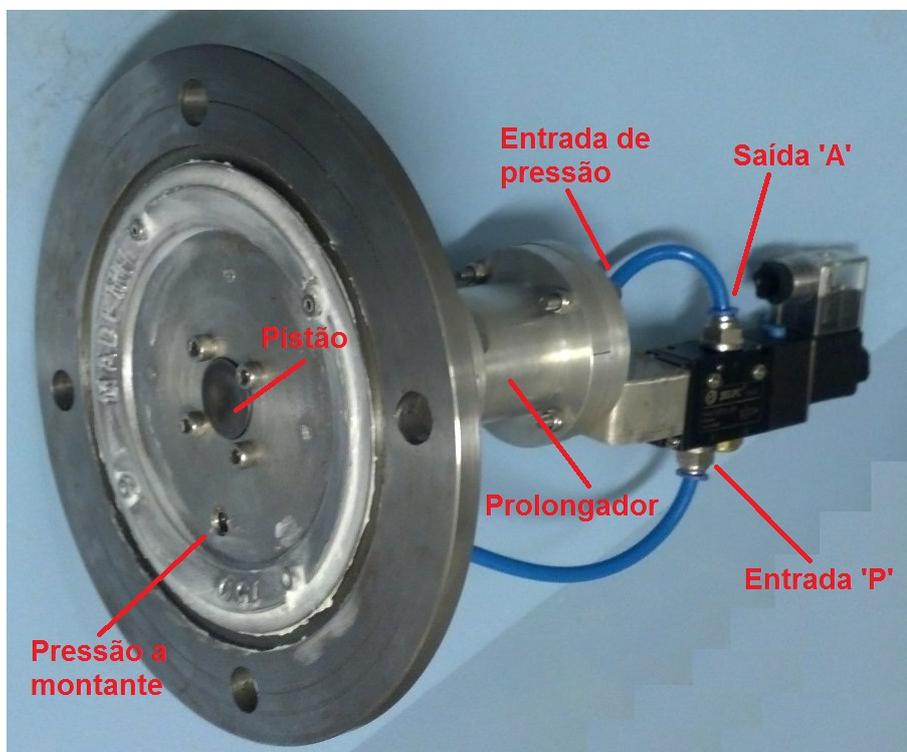


Figura 3.17: Imagem real da válvula *by-pass*.

O funcionamento do conjunto acontece quando a pressão a montante entra pelo orifício indicado na Fig. 3.17, percorre a mangueira até chegar na entrada 'P' da válvula solenóide. Quando a solenóide está desligada não existe passagem de pressão da entrada 'P' para a saída 'A'. Com isso a válvula *by-pass* permanece fechada.

Quando ocorre a energização da solenóide, então acontece a conexão da entrada 'P' com a saída 'A', permitindo a passagem de pressão pela mangueira até a entrada de pressão, conforme a Fig. 3.17. Com isso o pistão é empurrado para frente, abrindo a passagem principal de pressão, conforme a Fig. 3.16, na página 40. Se a energia da solenóide for cortada, provoca o corte de pressão no pistão, com isso este retorna para posição inicial, fechando a passagem principal de pressão.

3.5.1 Coeficiente de Vazão da Válvula *By-pass* Proposta

Para comprovação de que é possível controlar a velocidade de um *PIG* a partir dessa válvula será deduzido uma relação entre velocidade e diferencial de pressão aplicado ao corpo do *PIG*, com base no Coeficiente de Vazão (C_v) da válvula proposta.

Segundo de Lima, de Freitas, Salazar, Maitelli & de Assis O. Fontes (2014) o regime de escoamento através do orifício de descarga da válvula *by-pass* durante o controle de velocidade do *PIG* pode ser obtido a partir do Coeficiente de Vazão (C_v) dessa válvula, conforme a Eq. (3.6):

$$C_v = \frac{47,2 \cdot Q}{\sqrt{\Delta P \cdot (P_1 + P_2)} \cdot \rho} \quad (3.6)$$

em que: C_v é o coeficiente de vazão para válvula, ρ é a densidade relativa do gás em relação ao ar, Q é a vazão, ΔP é a diferencial de pressão ou perda de carga, P_1 é a pressão a montante da restrição, e P_2 pressão a jusante da restrição.

Agora, isolando a vazão Q na Eq. (3.6) obtém-se a Eq. (3.7):

$$Q = \frac{C_v \cdot \sqrt{\Delta P \cdot (P_1 + P_2)} \cdot \rho}{47,2} = 0,0212 \cdot C_v \cdot \sqrt{\Delta P \cdot (P_1 + P_2)} \cdot \rho \quad (3.7)$$

A vazão de um fluido em um duto é dada pela Eq. (3.8):

$$Q = V \cdot A_{duto} \quad (3.8)$$

em que: Q é a vazão do fluido, V é a velocidade do fluido, e A_{duto} é a área do duto.

Substituindo a Eq. (3.8) na Eq. (3.7) obtém-se uma relação entre diferença de pressão

ΔP e a velocidade do fluido V , conforme Eq. (3.9):

$$V \cdot A_{duto} = 0,0212 \cdot C_v \cdot \sqrt{\Delta P \cdot (P_1 + P_2) \cdot \rho} \quad (3.9)$$

Por fim, considerando-se que a velocidade do *PIG* é a mesma velocidade do fluido (quando a válvula estiver fechada), então pode-se expressar a velocidade do *PIG* pela Eq. (3.10):

$$V_{PIG} = \frac{0,0212 \cdot C_v \cdot \sqrt{\Delta P \cdot (P_1 + P_2) \cdot \rho}}{A_{duto}} \Rightarrow V_{PIG} = f(\Delta P, P_1) \quad (3.10)$$

Então, pode-se dizer que existe uma relação entre o diferencial de pressão aplicado ao *PIG* e a velocidade alcançada pelo mesmo, conforme demonstração apresentada acima.

3.6 Bancada de Testes

A bancada de testes montada para simulação de situações de “tiro” em *PIGs* foi construída em aço galvanizado, na bitola de 4” de diâmetro e aproximadamente 2 metros de comprimento. Na Figura 3.18 é possível visualizar a válvula *by-pass* proposta VP ao centro da tubulação de 4”, representando um *PIG* estático. Em cada lado da válvula *by-pass* foi instalado um manômetro junto com um transdutor de pressão. E por último, um registro esfera de acionamento manual em cada extremidade da tubulação, identificada por V1 e V2, respectivamente .



Figura 3.18: Bancada de testes para simulações de “tiro” e de aceleração.

O lado direito da válvula *by-pass*, chamado de montante, recebe a pressão que vem de um compressor. A pressão máxima possível a montante é 80 psi, sendo limitada por um regulador de pressão e a válvula V1 permanece aberta durante todo o ensaio. Já do lado esquerdo da válvula *by-pass*, chamado de jusante, tem em sua extremidade o registro esfera V2. A abertura desse registro esfera permite a simulação das situações de "tiro" e de aceleração em um *PIG* dentro de um duto.

A Figura 3.19 apresenta um detalhe dos principais componentes da bancada de testes, para uma melhor visualização dos mesmos.

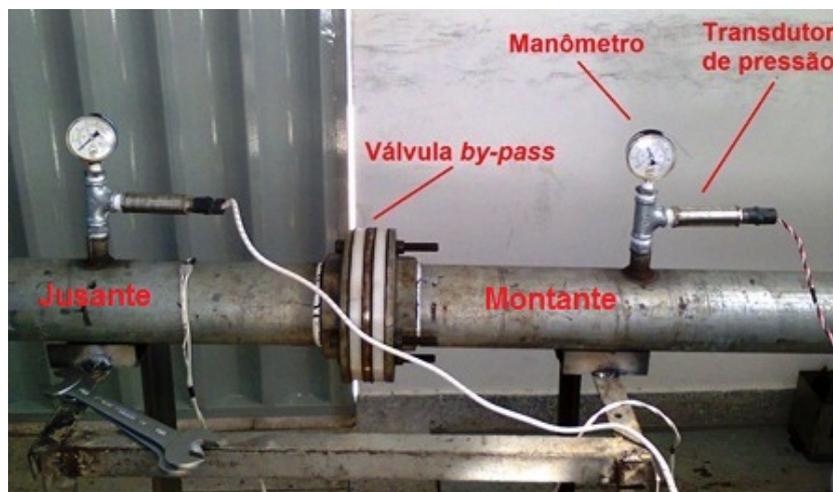


Figura 3.19: Detalhe da bancada de testes.

Próximo dessa bancada foi instalada uma mesa para acomodação da parte eletrônica, necessária para o pleno funcionamento do sistema proposto. A Figura 3.20 apresenta os componentes eletrônicos utilizados para realização dos testes.



Figura 3.20: Bancada montada com os equipamentos eletrônicos.

A bancada de instrumentos eletrônicos conta com: 01 fonte dupla de tensão contínua e regulável para alimentação dos transdutores de pressão em 15 V e alimentação da válvula solenóide em 12 V; 01 *protoboard* para condicionamento de sinais dos transdutores; 01 placa de aquisição de dados USB-6008; 01 placa Arduino Uno para acionamento da válvula solenóide, utilizando uma porta de saída PWM; e 01 computador para coleta de dados.

Para um melhor entendimento das ligações entre os diversos componentes da bancada de testes foi produzido um diagrama de blocos e apresentado na Fig. 3.21.

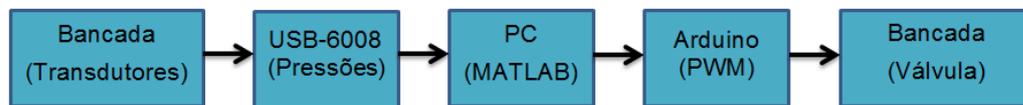


Figura 3.21: Diagrama de blocos da bancada de testes.

Então, a bancada produziu dois sinais analógicos que foram captados pela placa USB-6008. Esta os enviou para o *software* Matlab™ que processou os sinais e retornou um valor de saída para o Arduino. E este acionou a abertura ou fechamento da válvula *by-pass*.

3.7 Metodologia Experimental

A metodologia empregada nesta Dissertação foi a de realizar a simulação de duas situações básicas e distintas: a de “tiro” e a de aceleração do *PIG* dentro do duto. Em ambas, foi observado as reações das pressões a montante e a jusante da válvula *by-pass*, como também, do diferencial de pressão, que é responsável pela movimentação do *PIG* dentro do duto.

Para obtenção dos resultados foi necessário realizar a montagem dos componentes apresentados no entorno da bancada de testes. Primeiro, os sinais de tensão dos transdutores de pressão foram conectados nas entradas analógicas AI0 e AI1 do dispositivo DAQ USB-6008 e este foi conectado ao computador via cabo USB. Depois, o Arduino foi conectado ao computador vica cabo USB, também. E por fim, a saída da porta digital 09 da placa Arduino, com um sinal PWM, foi conectada ao resistor da base do transistor Darlington TIP 122, que aciona a válvula *by-pass*.

A Figura 3.22 mostra as interligações realizadas no entorno da bancada de testes.

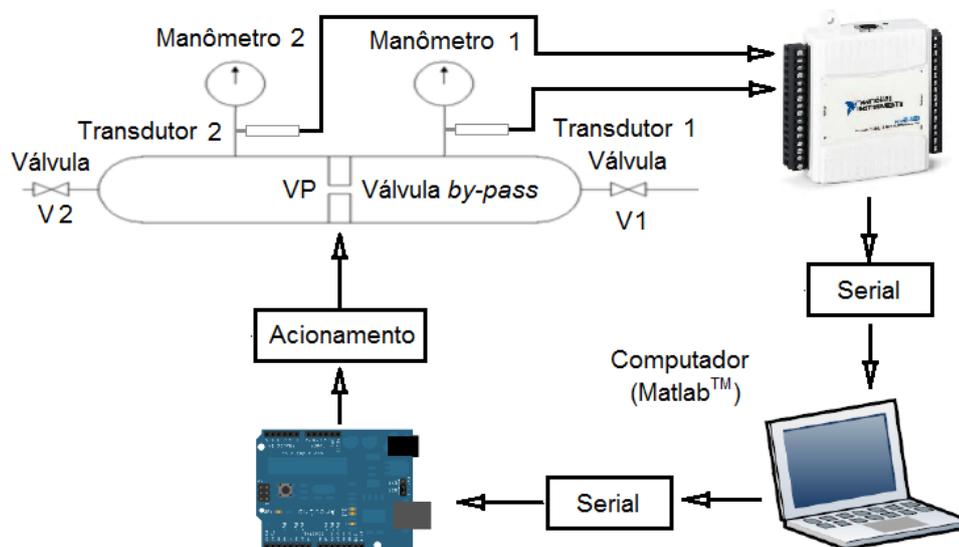


Figura 3.22: Conexões realizadas na bancada de testes.

3.7.1 Simulação de “Tiro”

Em uma situação real, o momento do “tiro” acontece depois que o *PIG* ficou emperrado na tubulação. Quando isso ocorre, a pressão a jusante cai praticamente a zero, enquanto que a pressão a montante cresce até romper a força estática que faz o *PIG* permanecer parado no duto. No momento do “tiro”, o diferencial de pressão sobe drasticamente provocando o disparo sem controle do *PIG* dentro da tubulação.

Este tipo de evento foi simulado na bancada de teste. A válvula *by-pass* foi mantida aberta para que as pressões a montante e a jusante ficassem iguais em aproximadamente 60 psi, por exemplo. Quando isso aconteceu, a válvula *by-pass* foi fechada. Com as duas pressões estando praticamente iguais, o diferencial de pressão foi praticamente zero, com isso, foi simulado a situação do *PIG* parado dentro da tubulação.

A situação de “tiro” foi simulada com a abertura manual de forma rápida do registro esfera identificado por V1 (ver Fig. 3.18) que foi instalado à jusante da válvula *by-pass*. No momento de abertura desse registro ocorreu uma rápida queda de pressão do lado jusante da válvula *by-pass*, conseqüentemente um rápido aumento do diferencial de pressão ΔP , simulando assim, o momento do “tiro” dentro do duto.

Ocorrido o “tiro” pela abertura do registro esfera, quase que ao mesmo tempo foi comandado pelo Arduino a abertura da válvula *by-pass*. Isso refletiu na queda acentuada do diferencial de pressão, permitindo assim alcançar valores baixos, para uma simulação de movimentação segura do *PIG* dentro da tubulação.

3.7.2 Simulação de Aceleração

Um segundo tipo de evento ao qual os *PIGs* instrumentados podem sofrer durante uma inspeção de dutos é de acelerar, por um aumento da pressão a montante. Quando a pressão montante cresce em relação à pressão jusante provoca um aumento do diferencial de pressão ΔP , acarretando assim no aumento da velocidade alcançada pelo *PIG*.

Esta situação de aceleração foi simulada na bancada de testes com a abertura parcial do registro esfera V1. E este ficou aberto por curto espaço de tempo de maneira a permitir o funcionamento da válvula *by-pass*. Esta modulou, durante o tempo que V1 permaneceu aberta, com o objetivo de aumentar a pressão jusante e, assim, diminuir o diferencial de pressão ΔP atuante no corpo do *PIG*.

3.8 Conclusões

Neste capítulo foram apresentados os principais componentes utilizados na montagem da bancada de testes e suas funcionalidades. O primeiro deles foi o dispositivo de aquisição de dados o USB-6008 responsável por captar os sinais dos transdutores de pressão instalados na bancada de testes. Estes sinais foram processados pelos *software* Matlab™ e depois enviou um sinal de acionamento para a placa Arduino. Este comandou a abertura e fechamento da válvula *by-pass*, instalada na bancada de testes. E por fim, a metodologia empregada para testar a válvula *by-pass* proposta.

Capítulo 4

Testes e Resultados

Neste capítulo são apresentados os resultados dos ensaios experimentais realizados na bancada de testes. Foram testados duas situações distintas em que os *PIGs* podem ser submetidos: a simulação de “tiro” e a simulação de aceleração. Cada situação foi testada sem o funcionamento da válvula *by-pass* e depois com ela funcionando. Isso permitiu uma comparação do tipo antes e depois do efeito do diferencial de pressão aplicado ao corpo do *PIG*.

4.1 Nomenclatura da Bancada de Testes

A Figura 4.1 apresenta de forma simplificada a nomenclatura dos componentes pneumáticos utilizados na bancada de testes, onde: V1 e V2 são registros esfera de 1/2” (PN 50) e 3/4”(PN 40) de diâmetro, respectivamente; P1 e P2 são as pressões a montante e a jusante em relação à válvula *by-pass*, respectivamente; e VP é a válvula *by-pass* proposta.

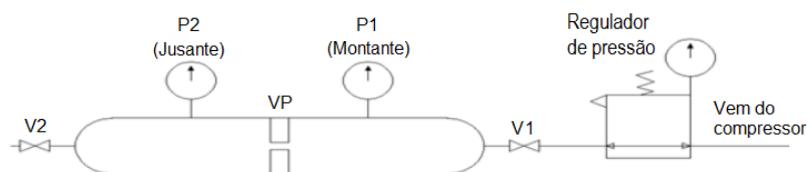


Figura 4.1: Nomenclatura dos componentes pneumáticos da bancada de testes.

O regulador de pressão conectado ao compressor limita a pressão máxima do lado montante da bancada de testes. Este valor de pressão pode variar de 0 a 80 psi. Por uma questão de segurança, os ensaios realizados tiveram sua pressão limitada em 60 psi.

Outro dado importante é que em todos os testes realizados, o registro V1 permaneceu aberto 100%, enquanto que V2 e VP ficaram abertos ou fechados conforme a necessidade de cada teste realizado.

4.2 Testes Iniciais

Para o primeiro ensaio experimental, na bancada de testes, foi realizada uma aquisição de pressões com o objetivo de testar os manômetros. Para isso o registro V2 ficou aberto 100% e a válvula *by-pass* permaneceu fechada 100%. Com o registro V2 aberto a câmara jusante ficou submetida à pressão atmosférica, que vale zero psi manométrico.

Como resultado, é apresentada a Fig. 4.2. Nela é possível ver a coerência das pressões, uma vez que, na montante (curva azul) a pressão ultrapassou o limite de 60 psi. Enquanto que a pressão na jusante (curva vermelha) foi próximo de zero psi, como esperado. A duração do ensaio foi de 100 s.

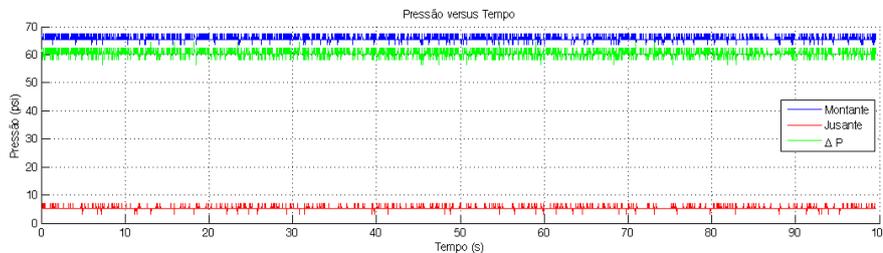


Figura 4.2: Ensaio com V2 aberto 100% e VP fechada 100%.

Para o segundo ensaio, na bancada de teste, a válvula *by-pass* VP foi mantida fechada. Enquanto que o registro V2 foi fechado 100% após o tempo de 10 s. No início a câmara da jusante estava vazia e depois de V2 fechado, foi observado um escape (ou descarga) de pressão de montante para jusante. A câmara da jusante começou a receber pressão da montante e isso foi registrado pelos transdutores.

A Figura 4.3 mostra o resultado deste ensaio. Numa situação real, o escape de pressão representa a passagem de gás da montante para jusante entre o suporte de borracha e a parede do duto (ver Fig. 2.2).

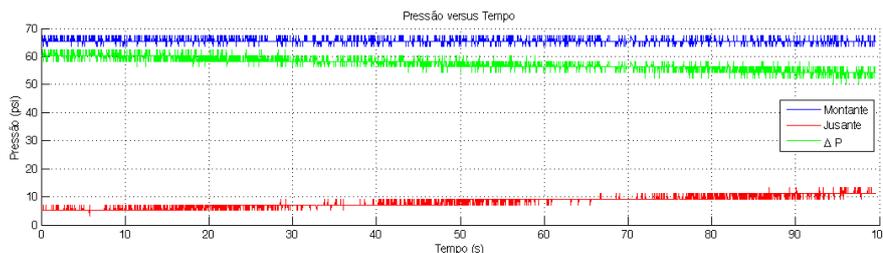


Figura 4.3: Ensaio com V2 e VP fechados 100%.

Depois destes dois ensaios experimentais iniciais foi dado início aos testes de “tiro” e depois de aceleração em *PIGs*.

4.3 Testes sem o Controlador *Fuzzy*

4.3.1 Simulação de “Tiro” e Resultados

A situação de “tiro”, como descrita anteriormente, é o momento em que o *PIG* fica sujeito a um alto valor de diferencial de pressão e, neste caso, por não ter uma válvula *by-pass*, que regule esse diferencial de pressão, desenvolve uma velocidade alta dentro de duto. Isto impossibilita uma leitura de qualidade pelos sensores, além de ser inseguro para realização da operação de inspeção.

Para uma visualização deste evento, foi produzido o gráfico da Fig. 4.4, em que foi realizado um “tiro” após o tempo de 10 s e VP foi mantida fechada 100%. Dessa forma, a redução do diferencial de pressão foi muito lenta, demorando quase 90 s para a pressão cair de 58 psi para 45 psi, o que significa uma redução de 22%.

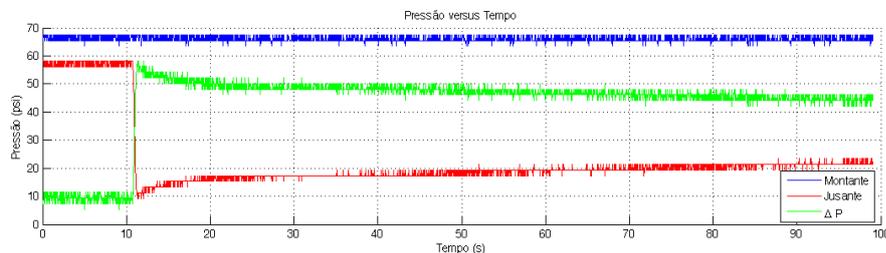


Figura 4.4: Ensaio com V2 realizando “o Tiro” e VP fechada 100%.

Para uma estimativa de velocidade alcançada pelo *PIG* depois deste “tiro”, foi aplicada a Eq. (2.7) em uma simulação do *software* Matlab™, conforme diagrama de blocos da Fig. 4.5.

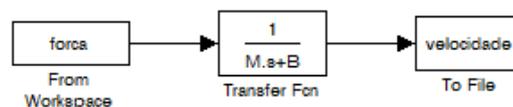


Figura 4.5: Estimativa de velocidade do *PIG*.

A Figura 4.6 apresenta o resultado dessa estimativa de velocidade. Nela a velocidade

alcançou um pico de 3,5 m/s e demorou quase 90 s para seu valor alcançar a marca de 2,7 m/s, aproximadamente, o que significa uma redução de 23%.

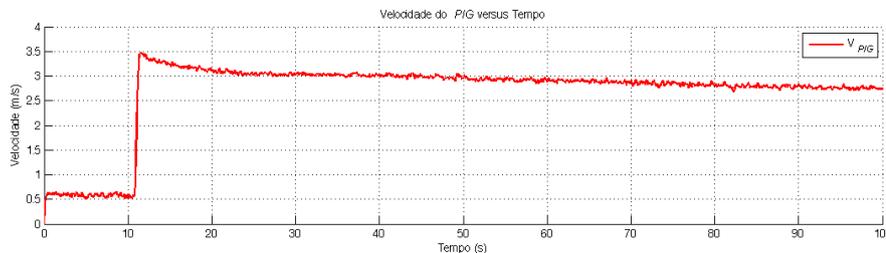


Figura 4.6: Velocidade estimada do *PIG* depois do “tiro”.

Esse é um intervalo de valores aceitável para o deslocamento do *PIG* dentro do duto, mas neste ensaio, o *PIG* não apresentou controle, ficando sujeito ao diferencial de pressão aplicado. Por isso, foi realizado o cálculo de espaço percorrido pelo *PIG* durante todo o ensaio de “tiro”. Com base no gráfico apresentado, foi possível determinar o valor de 268,15 m. Este resultado foi obtido pela integração numérica utilizando o *software* Matlab™ rodando o comando *trapz*.

Assim, fica confirmado a necessidade de desenvolver uma tecnologia que permita o controle dessa velocidade, e também, do espaço percorrido sem controle pelo *PIG*. Que inicialmente minimize os efeitos do evento “tiro” e depois mantenha a velocidade dentro de uma faixa predefinida.

4.3.2 Simulação de Aceleração e Resultados

O evento de aceleração é caracterizado pelo aumento da pressão montante em relação à pressão jusante. Isso provoca um aumento no diferencial de pressão, que implica na aceleração do *PIG*. Esse aumento acontece quando as empresas transportadoras de gás precisam atender a demanda dos clientes, aumentando a pressão dos compressores.

Na bancada de testes a aceleração foi caracterizada pela diminuição da pressão jusante em relação à pressão montante que permaneceu quase constante. Esse efeito foi obtido pela abertura parcial do registro esfera V2 por um tempo de aproximadamente 11 s.

A Figura 4.7 apresenta o resultado deste ensaio de aceleração sofrido pelo *PIG*. A diminuição de pressão jusante ocorreu dentro do intervalo de 11 a 22 s. Depois disso, o registro V2 foi fechado e, em seguida, começou o acúmulo de pressão na câmara jusante devido ao escape, fazendo o diferencial de pressão reduzir de valor, na duração de quase 80 s. Seu valor caiu de 50 psi para 42 psi, representando uma redução de 16%.

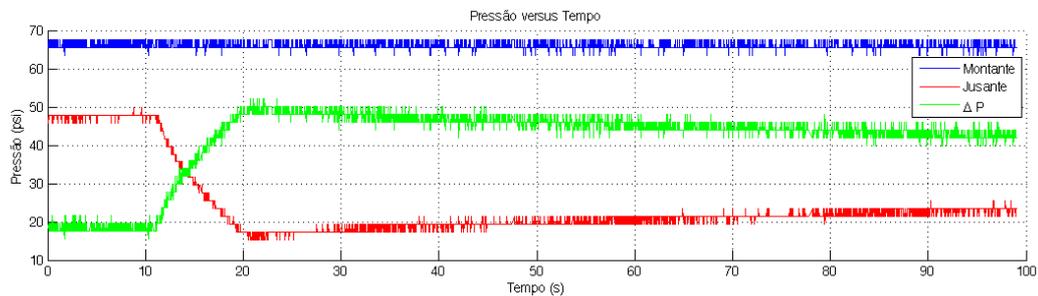


Figura 4.7: Diferencial de pressão obtido depois da aceleração.

A Figura 4.8 mostra o resultado da aceleração sofrida pelo *PIG*. Nela a velocidade alcançou um de 3,1 m/s e demorou quase 80 s para seu valor reduzir para 2,6 m/s, aproximadamente, o que significou uma redução de 16%.

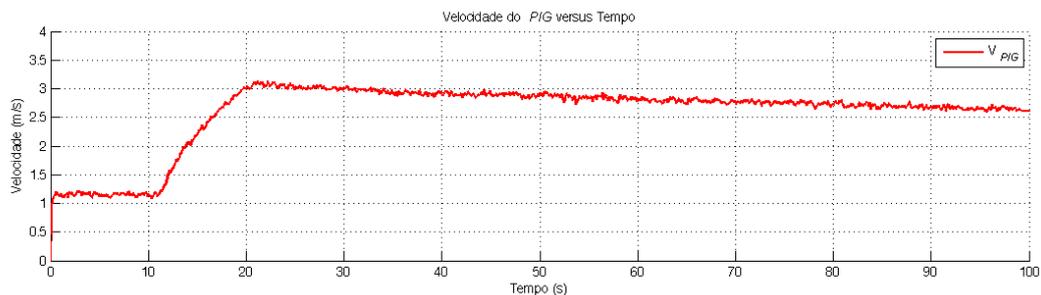


Figura 4.8: Velocidade estimada do *PIG* depois da aceleração.

Mais uma vez, a velocidade do *PIG* ficou dentro de uma faixa aceitável apresentada pela literatura, mas sem controle. O *PIG* percorreu o duto sujeito a um alto diferencial de pressão. Isso também motivou o cálculo do espaço percorrido pelo *PIG* durante todo o ensaio de aceleração. Com base no gráfico apresentado, foi possível determinar o valor de 258,33 m.

Os dois testes apresentados aqui servem de parâmetro de comparação com o que será mostrado mais adiante. E são semelhantes às situações reais, uma vez que, ainda não se tem o controle de velocidade implementado nos *PIGs* instrumentados hoje em dia.

4.4 Configuração do Controlador *Fuzzy*

Nesta seção são apresentadas as especificações do controlador *Fuzzy*. Para isso foi utilizada a ferramenta *Fuzzy Logical Toolbox* do software Matlab™.

Para poder regular a pressão utilizando a válvula *by-pass* foi definida uma faixa de oscilação desejada para o diferencial de pressão, após o “tiro”, entre 20 e 30% do valor de pressão da montante. A pressão montante foi regulada inicialmente em 60 psi, o que produziu uma faixa para oscilação do diferencial de pressão entre 12 e 18 psi.

4.4.1 Variáveis de Entrada do Sistema

As variáveis numéricas são convertidas em variáveis linguísticas. Aqui é apresentada a especificação das duas entradas do sistema, transformadas em conjuntos *Fuzzy*. Primeiramente, a variável “deltaP”. Esta entrada foi definida como podendo assumir valores de 0 a 80 psi, uma vez que a pressão máxima aplicada no sistema foi de 80 psi e por uma questão de segurança, os ensaios foram realizados com a pressão montante em 60 psi. Sendo “deltaP” dado pela Eq. (4.1).

$$\Delta P = P_{montante} - P_{jusante} \quad (4.1)$$

A Figura 4.9 ilustra o conjunto *Fuzzy* para a entrada “deltaP”. Os valores linguísticos definidos para “deltaP” foram baixo, referência, médio e alto.

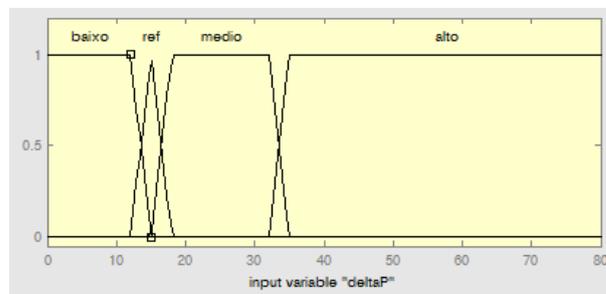


Figura 4.9: Variável de entrada “deltaP”.

A segunda variável de entrada foi chamada de “taxadeltaP”. Esta entrada foi definida como podendo assumir valores de -80 a 80 psi/s. Valores negativos indicam que a pressão está caindo e valores positivos indicam que a pressão está subindo. A taxa de variação do diferencial de pressão é dada pela Eq. (4.2).

$$\dot{\Delta P} = \frac{d}{dt} \Delta P = \frac{\Delta P_{atual} - \Delta P_{ant}}{t_{atual} - t_{ant}} \quad (4.2)$$

Os valores linguísticos definidos para “taxadeltaP” foram: caindo, estável e subindo. A Figura 4.10 mostra as pertinências utilizadas na variável “taxadeltaP”.

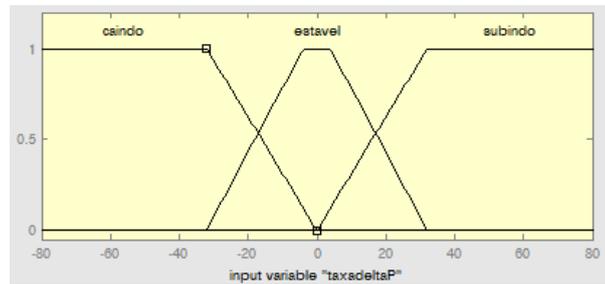


Figura 4.10: Variável de entrada “taxadeltaP”.

4.4.2 Variável de Saída do Sistema

A variável de saída do sistema foi definida como “PWM” e representa o sinal necessário para o acionamento da válvula solenóide. Esta é responsável por abrir/fechar o pistão da válvula *by-pass*. Esta saída foi definida como podendo assumir valores de 0 a 255 e não na forma usual por meio de porcentagem. A utilização de valores de 0 a 255 foi por uma necessidade de compatilizar com a porta digital da placa Arduino. Um sinal “PWM” é gerado no Arduino com valores de 8 *bits* ($2^8 = 256$).

Os valores linguísticos previstos para a variável de saída “PWM” foram: fechado, pouco, médio e aberto. A Figura 4.11 apresenta a configuração da pertinências da variável de saída “PWM”.

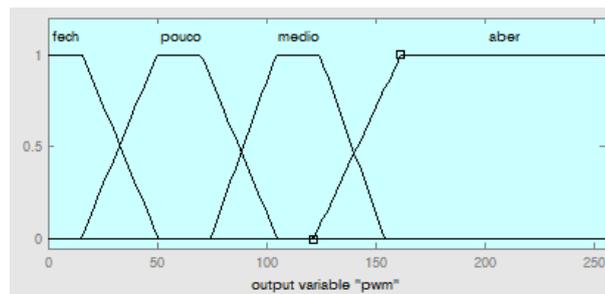


Figura 4.11: Variável de saída “PWM”.

4.4.3 Bloco de Regras

A Lógica *Fuzzy* necessita de regras que definam seu comportamento. Estas regras descrevem as condições esperadas durante o processo, e quais atitudes serão tomadas para cada condição. Elas substituem as fórmulas matemáticas normalmente utilizadas. Estas regras devem cobrir todas as situações possíveis (Maniçoba 2013). Por este motivo o bloco de regras desta implementação possui doze regras, que cobrem todas as combina-

ções das duas entradas. A Figura 4.12 mostra as regras definidas dentro da ferramenta *Fuzzy Logical Toolbox* do Matlab™.

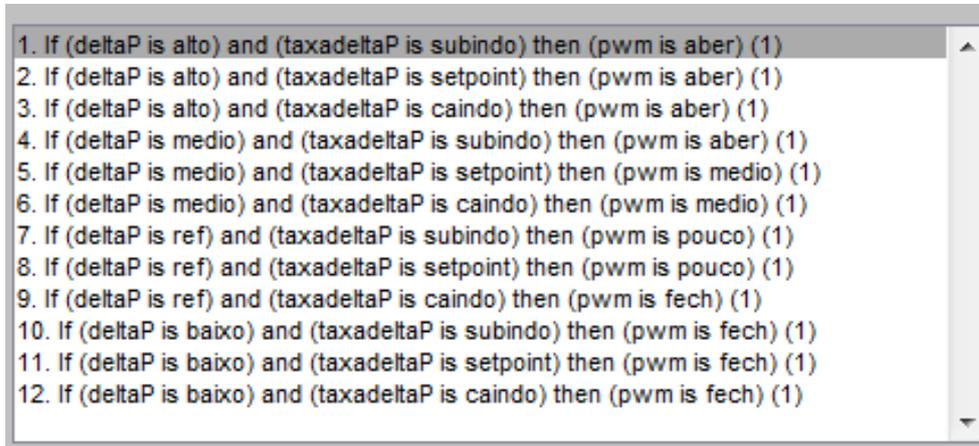


Figura 4.12: Base de regras utilizada.

4.4.4 Implicação e Agregação de Regras

Foi usada como máquina de inferência de Mamdani que utiliza como implicação a t-norma mínimo [min] e como agregação a sua co-norma máximo [max]. Inicialmente esta base de regras foi simulada várias vezes com o *software* Matlab™, a fim de testar sua coerência/consistência. A *defuzzificação* foi calculada utilizando o centróide.

4.5 Testes com Controlador *Fuzzy*

4.5.1 Simulação de “Tiro” e Resultados

Para mostrar o funcionamento do controlador *Fuzzy* atuando no acionamento da válvula *by-pass* foi necessário repetir o ensaio de simulação de “tiro” e registrar o comportamento das pressões para posterior análise. A Figura 4.13 apresenta a ocorrência de um “tiro” depois do tempo de 10 s e o diferencial de pressão atingiu um pico de aproximadamente 47 psi, menor do os 58 psi alcançado no ensaio anterior sem o controlador *Fuzzy*, sendo considerado bastante significativo.

Os efeitos do evento “tiro” foram minimizados muito rápido. Sua duração foi menor do que 3 s onde o diferencial de pressão caiu de 47 psi para 10 psi. Isto representa uma queda de 79%. Melhor do que os 22% de redução obtidos no ensaio de “tiro” sem o controle *Fuzzy*.

Por fim, neste ensaio foi possível visualizar que o diferencial de pressão permaneceu dentro da faixa predefinida de 12 psi a 18 psi de pressão. Isso foi considerado como satisfatório para o movimento seguro do *PIG*.

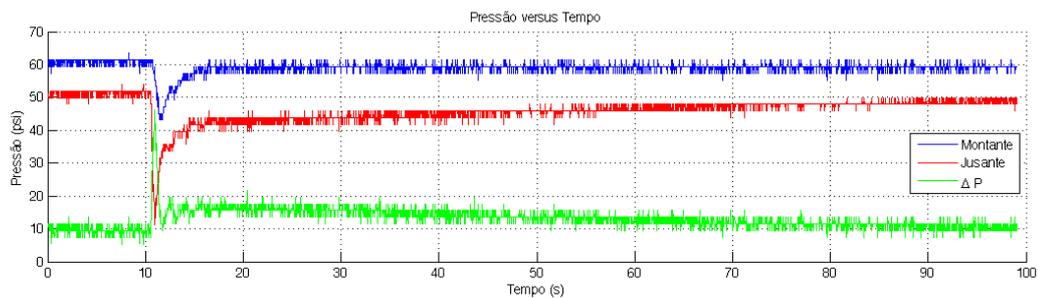


Figura 4.13: Diferencial de pressão obtido depois de um “tiro”.

Na sequência foi produzido o gráfico de velocidade estimada do *PIG*. A Figura 4.14 mostra o controle que aconteceu para reduzir no menor tempo possível os efeitos do “tiro” sobre o corpo do *PIG*. O pico de velocidade alcançou o valor máximo de 2,5 m/s e depois caiu para 0,8 m/s, o que representou uma redução de 68%. Melhor do que os 23% de redução obtidos no ensaio de “tiro” sem o controle *Fuzzy*.

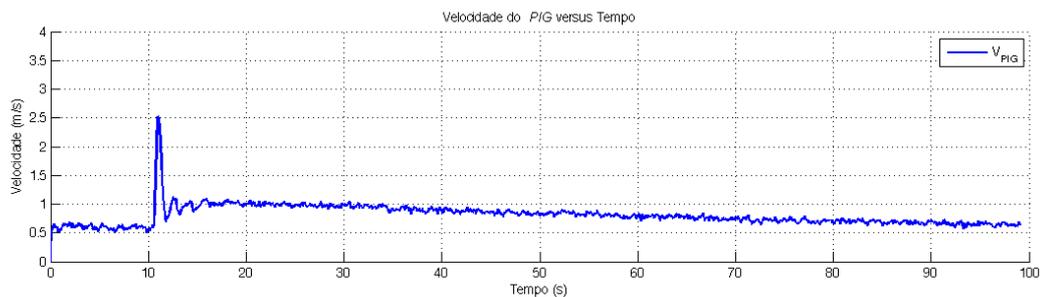


Figura 4.14: Velocidade do *PIG* com controle *Fuzzy*.

Depois foi calculado o espaço percorrido pelo *PIG* durante todo o ensaio de “tiro” com controle *Fuzzy*. Com base no gráfico apresentado, foi possível determinar o valor de 79,75 m. Melhor do que os 268,15 m apresentados no ensaio de “tiro” sem controle *Fuzzy*, o que significou uma redução de 70%.

Para comprovar o funcionamento do controlador *Fuzzy* na Fig. 4.15 é apresentada a variação de sinal de saída PWM aplicado à válvula *by-pass*.

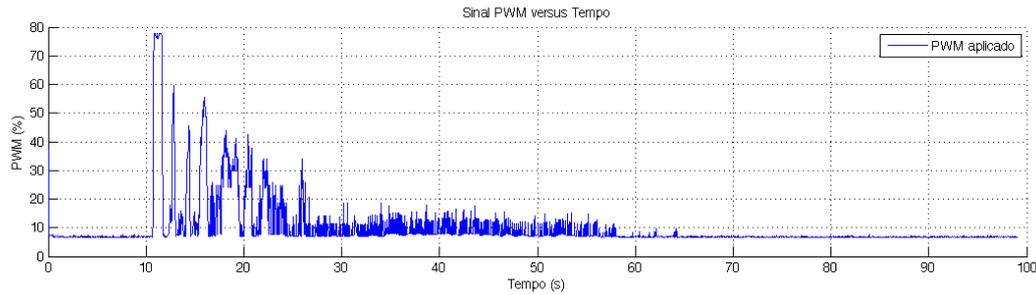


Figura 4.15: Sinal PWM aplicado na válvula *by-pass*.

4.5.2 Simulação de Aceleração e Resultados

Agora, utilizando a mesma configuração do controlador *Fuzzy*, apresentado anteriormente, foi produzido um ensaio em que o controle do diferencial de pressão atuou contra a aceleração do *PIG*. Neste teste o registro esfera V2 foi aberto depois do tempo de 11 s e fechado em 22 s, aproximadamente. Nesse intervalo de tempo o diferencial de pressão não aumentou, como aconteceu no ensaio anterior sem o controlador *Fuzzy*.

A Figura 4.16 apresenta o resultado do controle *Fuzzy* atuando para impedir um aumento do diferencial de pressão, que provoca aceleração do *PIG*. No intervalo de tempo de 10 s a 20 s, o diferencial de pressão oscilou dentro da faixa de 12 psi a 18 psi como estabelecido inicialmente.

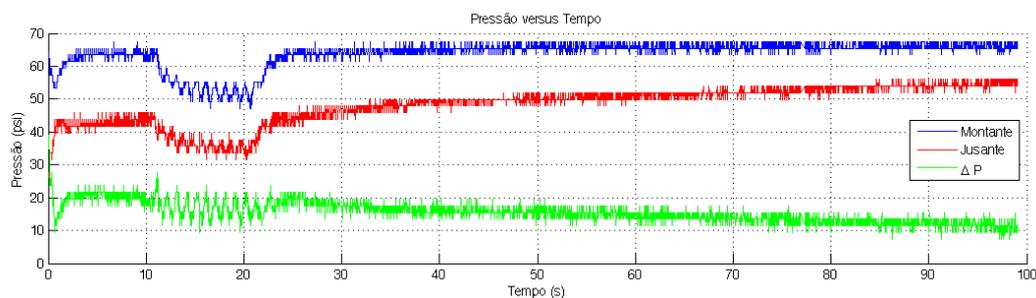


Figura 4.16: Diferencial de pressão atuante no *PIG* depois da aceleração.

A velocidade alcançada pelo *PIG* durante o evento de aceleração pode ser visualizada na Fig. 4.17. Nela, é possível visualizar, no trecho de 11 s a 22 s, o alto chaveamento ocorrido com o objetivo de manter a velocidade sob controle, dentro da faixa predefinida, dessa forma, evitando a aceleração do *PIG* como aconteceu no ensaio sem controle *Fuzzy*.

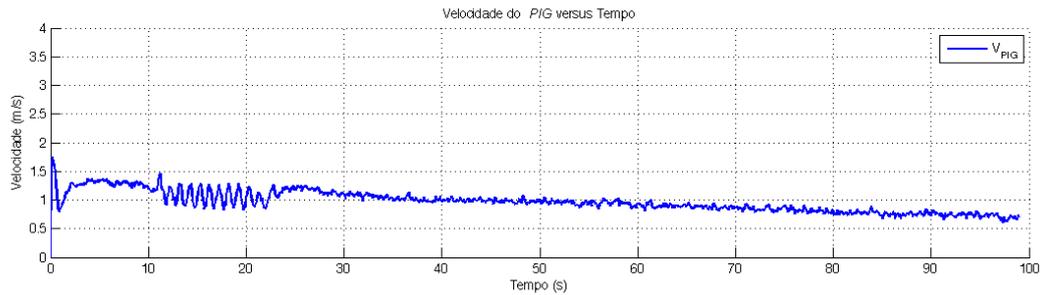


Figura 4.17: Controle de velocidade do *PIG* durante a aceleração.

Depois foi calculado o espaço percorrido pelo *PIG* durante todo o ensaio de aceleração com controle *Fuzzy*. Com base no gráfico apresentado, foi possível determinar o valor de 95,75 m. Melhor do que os 258,33 m apresentados no ensaio de aceleração sem controle *Fuzzy*, o que significou uma redução de 63%.

Para comprovar o funcionamento do controlador *Fuzzy* na Fig. 4.18 é apresentada a variação de sinal de saída PWM aplicado à válvula *by-pass*.

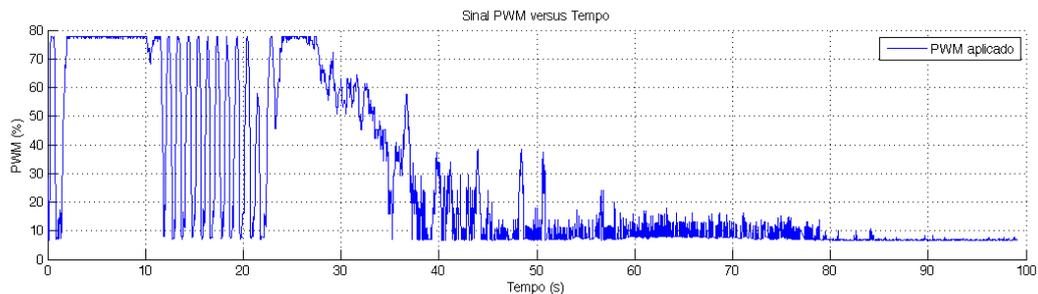


Figura 4.18: Sinal PWM aplicado na válvula *By-pass*.

4.6 Comparação dos Resultados Obtidos

Nesta seção são reapresentados os resultados obtidos nos ensaios experimentais utilizando a bancada de testes. Os gráficos antes do controle e depois do controle foram superpostos para uma melhor análise dos dados.

4.6.1 Simulação de “Tiro”

A Figura 4.19 apresenta os dois ensaios realizados para simulação do evento “tiro”. O ensaio sem controle foi representado na cor vermelha e com controle na cor azul.

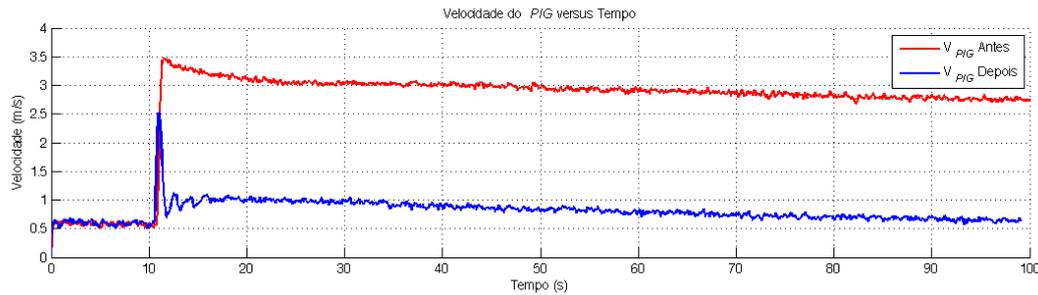


Figura 4.19: Resultados sem e com controle *Fuzzy* para o evento “tiro”.

Para auxiliar na análise, os dados de velocidade de pico, duração do evento e espaço percorrido pelo *PIG* antes sem controlador e depois com controlador *Fuzzy* foram organizados na Tab. 4.1.

Tabela 4.1: Comparação de resultados obtidos nas simulações de “tiro”.

	Sem controlador	Com controlador
Velocidade de pico	3,5 m/s	2,5 m/s
Redução de velocidade	23%	68%
Espaço percorrido	268,25 m	79,75 m

Outro parâmetro estabelecido para comparação foi o de estimar quanto tempo levaria para a curva vermelha alcançar a curva azul na Fig. 4.19. Para isso foi gerada uma linha de tendência após a velocidade de pico no valor de 3,5 m/s, conforme a Fig. 4.20.

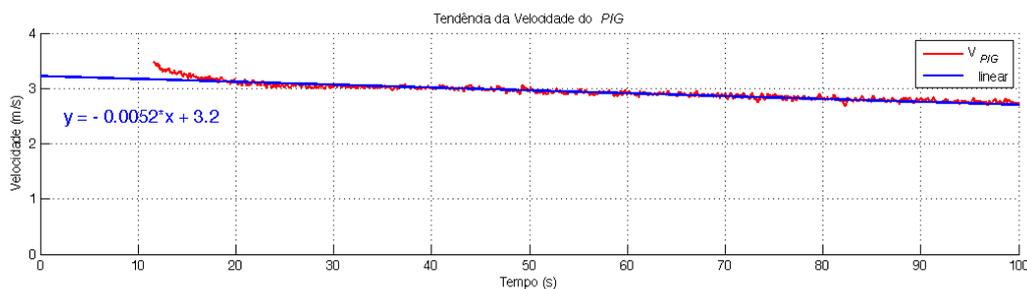


Figura 4.20: Tendência de velocidade do *PIG* depois do “tiro”.

A equação da reta obtida foi $y = -0,0052 \cdot x + 3,2$, ou ainda, em função de velocidade e tempo seria $v(t) = -0,0052 \cdot t + 3,2$. Para realizar essa comparação foi escolhido o

primeiro valor mínimo depois do “tiro” igual a 0,8 m/s. Assim, $0,8 = -0,0052 \cdot t + 3,2 \Rightarrow 0,8 - 3,2 = -0,0052 \cdot t \Rightarrow t \approx 461,5$ s.

De onde é possível concluir que sem o controle *Fuzzy* o tempo para reduzir a velocidade somente com o escape de gás entre o suporte de borracha e a parede do duto levaria aproximadamente 462 s, enquanto, que utilizando o controle *Fuzzy* a redução foi em 3 s.

4.6.2 Simulação de Aceleração

A Figura 4.21 apresenta os dois ensaios realizados para simulação de aceleração. O ensaio sem controle foi representado na cor vermelha e com controle na cor azul.

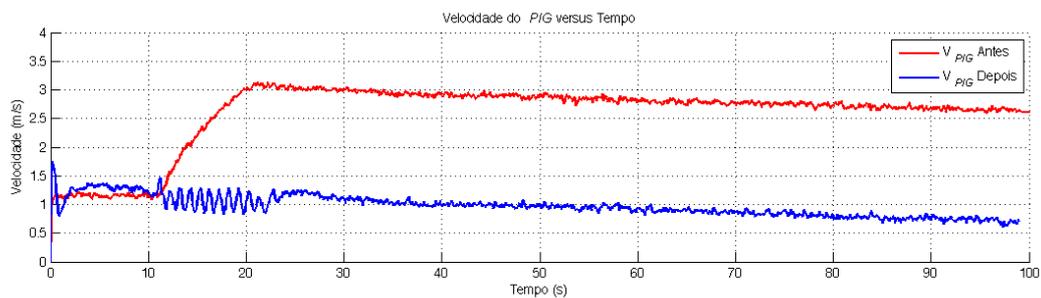


Figura 4.21: Resultados sem e com controle *Fuzzy* para o evento de aceleração.

Para auxiliar na análise, os dados de velocidade de pico, duração do evento e espaço percorrido pelo *PIG* antes sem controlador e depois com controlador *Fuzzy* foram organizados na Tab. 4.2.

Tabela 4.2: Comparação de resultados obtidos durante as simulações de aceleração.

	Sem controlador	Com controlador
Velocidade de pico	3,1 m/s	1,5 m/s
Espaço percorrido	258,33 m	95,75 m

Mais uma vez foi aplicada a idéia de estimar quanto tempo levaria para a curva vermelha alcançar a curva azul na Fig. 4.21. Para isso foi gerada uma linha de tendência após de velocidade de pico no valor de 3,1 m/s, conforme a Fig. 4.22.

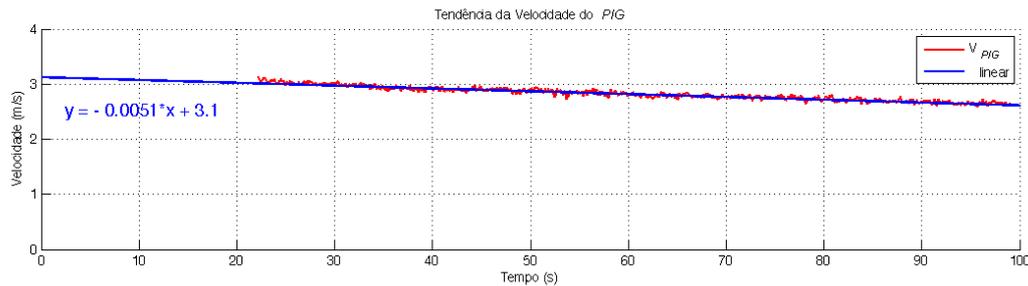


Figura 4.22: Tendência de velocidade do *PIG* depois da aceleração.

A equação da reta obtida foi $y = -0,0051 \cdot x + 3,1$, ou ainda, em função de velocidade e tempo seria $v(t) = -0,0051 \cdot t + 3,1$. Para realizar essa comparação foi escolhido o valor médio de velocidade durante o evento de aceleração no valor de 1 m/s. Assim, $1 = -0,0051 \cdot t + 3,1 \Rightarrow 1 - 3,1 = -0,0051 \cdot t \Rightarrow t \approx 411,76$ s.

Feito isso, é possível concluir que sem o controle *Fuzzy* o tempo para reduzir a velocidade somente com o escape de gás entre o suporte de borracha e a parede do duto levaria aproximadamente 411 s, enquanto que utilizando o controlador *Fuzzy* a velocidade oscilou em torno de 1 m/s durante 11 s, sem sinal de crescimento.

Então, ao analisar as Figs. 4.19 e 4.21 e as Tabs. 4.1 e 4.2 podemos verificar o sucesso da implementação do controlador *Fuzzy* para realizar o controle de velocidade de *PIGs*.

4.7 Conclusões

Neste capítulo foram apresentados os resultados dos ensaios experimentais realizados utilizando a bancada de testes. Foram mostrados os testes iniciais para aferição dos manômetros instalados na montante e na jusante da válvula *by-pass*. Depois, foram apresentados os testes de “tiro” e aceleração sem o funcionamento do controlador *Fuzzy*, de maneira a serem parâmetros de comparação. Em seguida, foram realizadas as configurações do controlador *Fuzzy*. Por fim, os ensaios de “tiro” e aceleração foram repetidos, mas com o controlador *Fuzzy* funcionando. Estes resultados mostram o controle efetivo do diferencial de pressão, como consequência, o controle da velocidade do *PIG* dentro do duto. Esta é a principal contribuição desta Dissertação, que apresenta uma implementação de um controlador inteligente por Lógica *Fuzzy* para regular a velocidade de um *PIG* instrumentado. Em resumo, para que não fiquem dúvidas, o movimento do *PIG* foi simulado, neste trabalho, pelas reações das pressões montante e jusante. Em momento algum foi lançado qualquer tipo de *PIG* dentro da tubulação da bancada de testes.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste capítulo são apresentados algumas considerações sobre os resultados obtidos ao longo deste trabalho. Além disso, são apresentadas algumas sugestões para trabalhos futuros de pesquisa.

5.1 Conclusões

O objetivo principal desta dissertação foi propor uma tecnologia capaz de controlar a velocidade de *PIGs* instrumentados. O movimento desses *PIGs* é altamente não-linear e sem um modelo matemático bem definido, o que dificulta a utilização de controladores tradicionais como PID ou Preditivo. A estratégia proposta, neste trabalho, conseguiu contornar esse problema através da implementação de um controlador inteligente implementado com lógica *Fuzzy*.

Uma bancada de testes, em escala reduzida, foi construída para simulação de situações de “tiro” e de aceleração sobre esses *PIGs*, isso permitiu uma melhor compreensão do que acontece com as pressões montante e jusante sobre a válvula *by-pass*. Uma placa NI USB-6008 foi empregada para realizar a aquisição dessas pressões e enviá-las para o Matlab™. Este foi responsável pelo processamento dos dados e obtenção de um valor de PWM para controlar a válvula *by-pass*, transmitido via serial para a placa Arduino. E por fim, este comandou a abertura e fechamento dessa válvula, por meio de um sinal PWM de 8 bits.

Os resultados obtidos, com a utilização da bancada de testes, mostraram um bom desempenho para o controle proposto. O mesmo conseguiu manter o diferencial de pressão atuante no corpo do *PIG* dentro de uma faixa predefinida, mesmo sendo submetido ao “tiro” ou ao evento de aceleração. A velocidade do *PIG* foi estimada por um modelo linearizado e foi observado que a mesma foi controlada com sucesso, permanecendo dentro de uma faixa aceitável de 1 a 5 m/s, sem prejuízos ou danos para os sensores, instalados no corpo do *PIG*.

5.2 Trabalhos Futuros

Como forma de aprofundar este trabalho e de uma forma geral para o desenvolvimento da área de *PIGs* instrumentados, são apresentados alguns tópicos para trabalhos futuros:

- Construir um *loop* com lançador e receptor de *PIGs* na bitola de 4" ou 6" nas dependências do LAMP para realização de testes práticos, em escala reduzida;
- Reavaliar e melhorar a bancada de testes para construção de um módulo didático;
- Montar um *PIG* protótipo para realização de testes sob condições reais de uma tubulação;
- Realizar o aperfeiçoamento da modelagem do movimento do *PIG*, incluindo variáveis não consideradas no presente modelo, como, o atrito seco e a força peso;
- Utilizar uma rede neural artificial para obtenção de modelo que relacione o diferencial de pressão e velocidade alcançada pelo *PIG*;
- Realizar a comparação com outras técnicas de controle, com base no modelo matemático obtido;
- Estudar e simular em *software* apropriado o fluxo de gás pelo corpo do *PIG*, para um entendimento melhor de como reagem as pressões montante e jusante, envolvidas no movimento do *PIG*.

Referências Bibliográficas

- Addor, Pedro N. (2009), *Avaliação da Corrosão em Dutos Rígidos Submarinos em Operação: Comparação entre as Técnicas de Monitoramento da Corrosão Interna e Inspeção com Pigs Instrumentados*, Dissertação de mestrado, Escola de Engenharia. Universidade Federal do Rio Grande do Sul, Porto Alegre, RS.
- Alves, Vanessa R. F. M. (2007), *Programação de Transferência de Derivados de Petróleo em Rede Dutoviária usando Algoritmo Genético*, Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.
- Banzi, Massimo, David Cuartielles, Tom Igoe, Gianluca Martino & David Mellis (2005), 'Site Arduino', <http://www.arduino.cc>. Acesso em: 26 abr. 2013.
- Barros, Pedro P. S. (2009), *Dimensionamento de um Suporte para Lançadores de PIGs e para Proteção de Válvulas*, Projeto de graduação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.
- Bilobrovec, Marcelo, Rui Francisco M. Marçal & João Luis Kovaleski (2004), *Implementação de um Sistema de Controle Inteligente Utilizando a Lógica Fuzzy*, em 'XI SIMPEP', Bauru, SP.
- Boylestad, Robert L. & L Nashelsky (2009), *Dispositivos eletrônicos e teoria de circuitos*, 8ª edição, Pearson, São Paulo, SP.
- Bueno, Alysson H. S. (2007), *Avaliação Integrada de Mecanismos de Falha por Corrosão em Dutos*, Tese de doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.
- Campos, Victor A. F., Douglas D. S. Santana, Newton Maruyama & Celso M. Furukawa (2004), *Trajectory Estimation of a PIG Using a Low Cost Inertial Measurement Unit and Odometry*, em 'VI Induscon 2004', Joinville, SC.
- Carvalho, Adelson S., Adriellen L. de Sousa, Larissa dos S. Martins & Philippe M. Bringhamti (2010), *Modelo Autorregressivo e Controlador Fuzzy para um Sistema de Controle Digital de Velocidade do Kit Digital Servo*, em 'VII SEGeT'.

- da Silva, Janicleide T., Josiane T. da Silva & Gustavo F. de Lima (2014), *Controle e Monitoramento de Nível de Líquido Utilizando Arduino Uno*, em 'IX Congresso Norte Nordeste de Pesquisa e Inovação - CONNEPI 2014', São Luiz, MA.
- da Silva, Thiago H. G. (2013), *Análise da Mecânica Respiratória de Traqueas Isoladas de Ratos*, Dissertação de mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, SP.
- de Araújo, Anne A. (2012), *Gerenciamento de Falhas por Corrosão em Dutos*, Dissertação de mestrado, Pontifícia Universidade Católica, Rio de Janeiro, RJ.
- de Freitas, Victor C. G., Gustavo F. de Lima, Ralyson R. G. de Oliveira, Andrés O. Salazar, André L. Maitelli & Francisco de A. O. Fontes (2014), *Plataforma Arduino no Controle de Velocidade de PIGs*, em 'SAS 2014', Natal, RN.
- de Lima, Fábio Araújo (2011), *Sistema Automatizado para o Aproveitamento de Água Cinza e Azul com Controle de PH Utilizando Lógica Fuzzy*, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, Natal, RN.
- de Lima, Gustavo F. (2013), *Utilização da Placa Arduino no Controle de Temperatura de um Sistema de Baixo Custo*, em 'VIII Congresso Norte Nordeste de Pesquisa e Inovação - CONNEPI 2013', Salvador, BA.
- de Lima, Gustavo F., Glauco George C. Maniçoba & Andrés O. Salazar (2013), *Controle de Temperatura de uma Tocha de Plasma Térmico com Acoplamento Indutivo Utilizando a Placa Arduino*, em 'XI Simpósio Brasileiro de Automação Inteligente - SBAI 2013', Fortaleza, CE.
- de Lima, Gustavo F., Glauco George C. Maniçoba & Andrés O. Salazar (2014), *Temperature Control of a Thermal Plasma Torch with Inductive Coupling Using the Arduino Board*, em 'Journal of Mechanics Engineering and Automation', Vol. 4, pp. 499–504, No. 6.
- de Lima, Gustavo F., Victor C. G. de Freitas, Andrés O. Salazar, André L. Maitelli & Francisco de Assis O. Fontes (2014), *Proposta de Controle de Velocidade para PIGs Instrumentados Utilizando a Plataforma Arduino*, em 'VIII Congresso Nacional de Engenharia Mecânica -CONEM 2014', Uberlândia, MG.
- de Lima, Gustavo F., Victor C. G. de Freitas, Ralyson R. G. de Oliveira, Andrés O. Salazar, André L. Maitelli & Francisco de Assis O. Fontes (2014), *Proposta para Con-*

- trole de Velocidade de PIGs Utilizando a Placa Arduino*, em 'Rio Oil & Gas 2014', Rio de Janeiro, RJ.
- de Oliveira Souza, Robson (2005), *Avaliação de Modelos Para a Remoção de Depósitos de Parafina em Dutos Utilizando Pigs*, Dissertação de mestrado, Departamento de Engenharia Mecânica. Pontifícia Universidade Católica, Rio de Janeiro, RJ.
- de Robótica, Grupo (2012), *Introdução ao Arduino*, Notas de aula, Universidade Federal do Mato Grosso do Sul.
- Diaz, John F. A. (2008), *Estudo e Projeto Conceitual de um Robô para Inspeção de Linhas de Serviço*, Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.
- Emmendoerfer, Gustavo, Ricardo A. Langer, Winderson E. dos Santos & Roger Gules (2008), *Sistema de Reconstrução de Trajetórias de Pig's Baseado em Medidas Gps/Giroscópios/Odômetros*, em 'XVII Congresso Brasileiro de Automática - CBA 2008', Juiz de Fora, MG.
- Evans, Martin, Joshua Noble & Jordan Hochenbaum (2013), *Arduino em Ação*, 1ª edição, Novatec Editora Ltda, São Paulo, SP.
- Filho, Fernando M. L., Hugo Leonardo Gosmann & Adolfo Bauchspiess (2002), *Controle Fuzzy para Sistema de Nível de Líquidos*, em 'XIV Congresso Brasileiro de Automática', Natal, RN, pp. 3017–3022.
- Fonseca, Erika G. P., Mathyan M. Beppu & Alexandre S. de la Vega (2010), *Apostila Arduino*, Notas de aula, Universidade Federal Fluminense, Niterói, RJ.
- Franzoi, Aldo R., Claudio S. Camerini, Sergio I. O. Bueno, José A. P. Silva & Ivan V. J. Miranda (2001), *Pig Especial para Inspeção de Oleoduto na Bacia de Campos*, em '3rd Seminar on Pipeline', Rio de Janeiro, RJ.
- Franzoi, Aldo R., Sergio I. O. Bueno, Claudio S. Camerini, Carla Marinho, Fabiana Nazario, Miguel Freitas & Jean P. von der Weid (2005), *Comparação do Resultado de 3 Técnicas de Inspeção com Pigs Instrumentados em um Mesmo Oleoduto da Bacia de Campos*, em 'Rio Pipeline Conference & Exposition 2005', Rio de Janeiro, RJ.
- Gesteira, Luis G. G. K. (2014), *Sistema de Aquisição de Dados Baseado em Labview para um Dinamômetro de Chassi*, Dissertação de mestrado, Universidade Federal da Bahia, Salvador, BA.

- Gioppo, Lucas L., Marcelo M. K. Higaskino, Ricardo F. da Costa & William H. T. Meira (2009), *Robô Seguidor de Linha*, Projeto de graduação, Universidade Tecnológica Federal do Paraná, Curitiba, PR.
- Gomes, Samyr A. (2011), *Chuveiro Automático*, Projeto de graduação, Centro Universitário de Brasília, Brasília, DF.
- Guibin, Tan, Zhang Shimin, Zhu Xiaoxiao, Shi Liyun & Zhang Qingbao (2011), *Research on Bypass-valve and its Resistance Characteristic of Speed Regulating PIG in Gas Pipeline*, em '2011 Third International Conference on Measuring Technology and Mechatronics Automation', pp. 1114–1117.
- Han, Hyung S., Jae J. Yu, Chan G. Park & Jang G. Lee (2004), *Development of Inspection Gauge System for Gas Pipeline*, em 'KSME International Journal', Vol. 18, pp. 370–378, No. 3.
- Hu, Zheng & Ernest Appleton (2005), *Dynamic Characteristics of a Novel Self-Drive Pipeline Pig*, em 'IEEE Transactions on Robotics', Vol. 21, pp. 781–789, No. 5.
- Ivanqui, Josmar (2005), *Esteira Eletrônica com Velocidade Controlada por Lógica Fuzzy*, Dissertação de mestrado, Centro Federal de Educação Tecnológica do Paraná, Curitiba, PR.
- Junior, Carlos A. B. & Sérgio R. Lourenço (2006), *Sistema de Monitoramento da Corrosão Interna de Gasoduto por Meio de Processamento Digital de Sinais*, em 'Exacta', Vol. 4, São Paulo, SP, pp. 391–400.
- Kim, Dong K., Sung H. Cho, Seoung S. Park, Yong W. Rho, Hui R. Yoo, Tan T. Nguyen & Sang B. Kim (2003), *Verification of the Theoretical Model for Analyzing Dynamic Behavior of the PIG from Actual Pigging*, em 'KSME International Journal', Vol. 17, pp. 1357–1357, No. 9.
- Maniçoba, Glauco G. C. (2013), *Controle de Temperatura do Sistema de Refrigeração a Água de uma Tocha de Plasmática Indutiva Utilizando Lógica Fuzzy*, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, Natal, RN.
- Mazzini, Carlos E. (2009), *Comparação de Métodos de Inspeção de Integridade de Dutos: Método PIG e CIS/DCVG*, Projeto de graduação, Departamento de Engenharia Mecânica. Universidade Federal do Espírito Santo, Vitória.

- Nguyen, Tan T., Hui R. Yoo, Yong W. Rho & Sang B. Kim (2001), *Speed Control of PIG Using Bypass Flow in Natural Gas Pipeline*, em 'Industrial Electronics, 2001. IEEE International Symposium on', Vol. 2, pp. 863 –868.
- Nguyen, Tan T., Sang B. Kim, Hui R. Yoo & Yong W. Rho (2001), *Modeling and Simulation for PIG Flow Control in Natural Gas Pipeline*, em 'KSME International Journal', Vol. 15, pp. 1165–1173, No. 8.
- NI (2013), *User Guide and Specifications - NI USB-6008/6009*, Manual do usuário, National Instruments.
- Panta, Pedro E. G. (2005), *Monitoramento de Robô de Inspeção Interna de Oleodutos - GIRINO*, Dissertação de mestrado, UFRJ, Rio de Janeiro, RJ.
- Pereira, Cinthia G. (2012), *Estudo e Simulação de um Sistema de Controle de Velocidade de PIGs*, Projeto de graduação, UFRN, Natal, RN.
- Pereira, Cinthia G., André L. Maitelli & Andrés O. Salazar (2013), *Estudo e Simulação de Tecnologia para Controle de Velocidade de PIGs Utilizando Lógica Fuzzy*, em 'NAVCOMP 2013', Rio Grande, RS.
- Pereira, Cinthia G., Ralyson R. G. de Oliveira, Gustavo F. de Lima, Andrés O. Salazar, André L. Maitelli & Francisco de Assis O. Fontes (2013), *Proposta de Tecnologia para Controle de Velocidade de PIGs Instrumentados*, em 'INTERCON 2013', Trujillo, Perú.
- Pipeway (2012), 'Site PipeWay', <http://www.pipeway.com.br>. Acesso em: 03 jul. 2012.
- Plácido, Marisol E. B. (2005), *Sistemas Robotizados de Inspeção Interna de Dutos*, Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.
- Sabino, João M. (2009), *Avaliação por Pig de Perfilagem de Danos Superficiais nos Materiais das Paredes de Dutos de Petróleo*, Dissertação de mestrado, Centro de Tecnologia. Universidade Federal do Rio Grande do Norte, Natal, RN.
- Salcedo, Thiago (2009), *Análise do Sensor de um Pig Instrumentado do Tipo Palito*, Dissertação de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.
- Salcedo, Thiago, Jean P. von der Weid, Miguel Freitas, Claudio S. Camerini & Ricardo Moraes (2007), *Pig Cobra Palito: Um Método Simples e Eficiente para a Detecção e Dimensionamento de Corrosão em Dutos*, em 'Rio Pipeline Conference & Exposition 2007', Rio de Janeiro, RJ.

- Shaw, Ian S. & Marcelo G. Simões (1999), *Controle e Modelagem Fuzzy*, 1ª edição, Edgard Blücher Ltda, São Paulo, SP.
- Silva, Ciro R. S., Cinthia G. Pereira, Paulo L. B. Silva, Andrés O. Salazar, André L. Maitelli & Carla W. S. P. Maitelli (2012), *Desenvolvimento de Tecnologia de Controle de Velocidade para Pigs*, em 'Rio Oil & Gas Expo and Conference 2012', Rio de Janeiro, RJ.
- Silva, Ricardo H. & Valdir L. Lima (2006), *Avaliação das Principais Técnicas de Detecção de Vazamentos em Gasodutos Enterrados*, Projeto de graduação, Departamento de Engenharia Química. Universidade Federal da Bahia, Salvador.
- Souza, Anderson R., Alexsander C. Paixão, Diego D. Uzêda, Marco A. Dias, Sergio Duarte & Helio S. de Amorim (2011), *A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC*, em 'Revista Brasileira de Ensino de Física', Vol. 33, pp. 1702-1 – 1702-5.
- Souza, Fábio (2014), 'Site Embarcados', <http://www.embarcados.com.br/arduino-saidas-pwm/>. Acesso em: 22 dez. 2014.
- Souza, Ricardo D. (2003), *Avaliação Estrutural de Dutos com Defeitos de Corrosão Reais*, Dissertação de mestrado, Departamento de Engenharia Mecânica. Pontifícia Universidade Católica, Rio de Janeiro, RJ.
- Sutherland, Jeff & Hernan Paz (2000), *Advances in In-line Inspection Technology for Pipeline Integrity*, em 'V Annual International Pipeline Congress', Morelia, Mexico.
- Thiago, Vicente (2008), *Avaliação de Integridade no Gasoduto Bolívia-Brasil*, em 'Conferência de Tecnologia de Soldagem e Inspeção', Rio de Janeiro, RJ.
- Tolmasquim, Sueli T. (2004), *Projeto e Controle da Operação de Passagem de Pigs em Dutos*, Dissertação de mestrado, Departamento de Engenharia Mecânica. Pontifícia Universidade Católica, Rio de Janeiro, RJ.
- Vale, Marcelo R. B. G. (2007), *Análise Comparativa do Desempenho de um Controlador Fuzzy Acoplado a um PID Neural Sintonizado por um Algoritmo Genético com Controladores Inteligentes Convencionais*, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, Natal, RN.
- Zadeh, Lotfi A. (1965), *Fuzzy Sets*, em 'Information and Control', Universidade da Califórnia - Berkeley, pp. 338 – 354.

Apêndice A

Produção Científica Publicada

Para evidenciar a divulgação dos trabalhos não relacionados com esta dissertação de mestrado, seguem abaixo alguns artigos publicados ou submetidos até o momento:

1. Uma aplicação da placa Arduino para controle de temperatura de uma tocha de plasma térmico foi apresentado em de Lima et al. (2013). A configuração de um controlador *Fuzzy* e os resultados de ensaios experimentais foram apresentados.
2. Outra utilização para a placa Arduino foi no controle de temperatura de um ambiente de baixo custo apresentado em de Lima (2013). A montagem do sistema proposto e os resultados de um controle do tipo liga/desliga foram apresentados.
3. de Lima, Maniçoba & Salazar (2014) apresentaram os resultados de uma estratégia *Fuzzy* no controle de temperatura de uma tocha de plasma com acoplamento indutivo. Os resultados apresentados permitiram concluir que é possível controlar a temperatura da tocha utilizando o Arduino, uma vez que o controle de temperatura não necessita de uma grande velocidade na ação de controle.
4. Em da Silva et al. (2014) foi proposto a utilização da plataforma Arduino Uno no controle e monitoramento de nível de líquido de um sistema de tanques de baixo custo. O trabalho descreve a montagem do sistema e da régua de *LEDs*. Os resultados de um controle liga/desliga foram apresentados também.