

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

**Desenvolvimento de um Sistema de Controle de Posição Utilizando a
Abordagem de Conjuntos Aproximados**

Genival Alves dos Anjos Filho

Itajubá, Dezembro de 2017

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

Genival Alves dos Anjos Filho

**Desenvolvimento de um Sistema de Controle de Posição Utilizando a
Abordagem de Conjuntos Aproximados**

**Dissertação submetida ao Programa de Pós-Graduação
em Engenharia Elétrica como parte dos requisitos para
obtenção do Título de Mestre em Ciências em
Engenharia Elétrica.**

**Área de Concentração: Automação e Sistemas Elétricos
Industriais**

**Orientador: Prof. Dr. Paulo César Crepaldi
Co-Orientador: Prof. Dr. Luiz Lenarth Gabriel Vermaas**

**Dezembro de 2017
Itajubá-MG**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

Genival Alves dos Anjos Filho

**Desenvolvimento de um Sistema de Controle de Posição Utilizando a
Abordagem de Conjuntos Aproximados**

Dissertação aprovada por banca examinadora em 15 de dezembro de 2017, conferindo ao autor o título de *Mestre em Ciências em Engenharia Elétrica*.

Banca examinadora:

Prof. Dr. Paulo César Crepaldi (Orientador)

Prof. Dr. Luiz Lenarth Gabriel Vermaas (Co-Orientador)

Prof. Dr. Robson Luiz Moreno

Prof. Dr. José Feliciano Adami

**Itajubá-MG
2017**

Aos meus pais, Genival e Sônia.

À minha querida tia Cerize.

Agradecimentos

A Deus, pela força e pela proteção para me amparar.

Aos meus pais, Genival e Sônia, pelo incentivo e por me apoiarem durante os anos de estudo.

Aos meus irmãos, Monalisa e Gabriel, pela motivação e amizade.

Um agradecimento especial aos Professores Crepaldi e Luiz Lenarth, pela paciência, pelo tempo dedicado e incentivo, que foram fundamentais para o desenvolvimento desse projeto de pesquisa.

Aos antigos e novos amigos, pelo companheirismo.

À CAPES, pelo apoio financeiro.

“O conhecimento e a informação são os recursos estratégicos para o desenvolvimento de qualquer país. Os portadores desses recursos são as pessoas.”

Peter Drucker

Resumo

Este projeto de pesquisa apresenta o estudo e desenvolvimento de controladores utilizando os conceitos dos conjuntos aproximados (*rough sets*) para serem aplicados em um sistema de controle de posição utilizando o motor CC. A Teoria dos Conjuntos Aproximados é uma ferramenta matemática desenvolvida no início da década de 80, e possibilita a obtenção de um conjunto de regras com base no conhecimento sobre determinado sistema. Esse conceito é propício para situações práticas, devido a possibilidade da sua aplicação em dados que contenham incertezas ou imprecisões.

Uma das contribuições refere ao desenvolvimento de estruturas de controle simplificadas de modelos baseado em regras, também denominado controladores aproximados. A inteligência desses sistemas é representada pela capacidade de classificar determinadas condições nas quais são submetidos para tomada de decisões adequadas. São apresentadas duas abordagens para o desenvolvimento do controlador aproximado, resultando em duas estruturas diferentes. Em ambos os casos, buscou-se pelo menor tempo de processamento e por condições estáveis para o sistema.

Para demonstrar a viabilidade da abordagem proposta, são apresentados resultados de simulações e ensaios experimentais envolvendo as malhas de controle. Além de comparações de desempenho entre os modelos aproximados e os sistemas que utilizam o controlador *fuzzy*.

Palavras-chave: Controle inteligente, *Fuzzy*, Conjuntos Aproximados, tempo de processamento, motor CC.

Abstract

This research project presents the study and development of controllers using rough set concepts to be applied in a position control system using a DC motor. The Rough Set is a mathematical tool developed in the 1980's that allows the obtaining of a set of rules based on knowledge regarding a certain system. This concept is conducive to practical situations, due to its application in data that contain uncertainties or inaccuracies.

One of the contributions is the development of rule-based simplified model control structures, also called rough controllers. The intelligence of these systems is represented by their ability to classify certain conditions under which they are submitted to for appropriate decision making. Two approaches are presented for the development of the rough controller, resulting in two different structures. In both cases, we searched for the shortest processing time and stable system conditions.

To demonstrate the feasibility of the proposed approach, simulation and experimental test results involving the control loops are presented, in addition to performance comparisons between the rough models and the systems that use the fuzzy controller.

Keywords: Intelligent control, Fuzzy, Rough Set, processing time, DC motor.

Lista de Figuras

Figura 2.1 – Modelo do circuito elétrico do motor CC.	20
Figura 2.2 – Diagrama de blocos de uma malha de controle.	23
Figura 2.3 – Curva da resposta transitória e de regime estacionário.	23
Figura 2.4 – Resposta em malha aberta.	25
Figura 2.5 – Sistema em malha fechada para a oscilação sustentada.	26
Figura 2.6 – Diagrama do modelo de inferência de Mamdani.	34
Figura 2.7 – Representação gráfica do método de inferência MAX-MIN.	35
Figura 2.8 – Controlador <i>Fuzzy</i> PD.	38
Figura 2.9 – Representação das regiões do conjunto O	42
Figura 2.10 – Identificação das regiões $P_1^{(m)}$	48
Figura 2.11 – Mapeamento de conjuntos <i>fuzzy</i> em conjuntos <i>crisp</i>	49
Figura 2.12 – Exemplo do mapeamento de conjuntos <i>fuzzy</i> em conjuntos <i>crisp</i>	50
Figura 2.13 – Ilustração das regras ativadas na resposta do sistema de controle.	52
Figura 3.1 – Resposta do sistema a entrada tipo degrau sem o controlador.	55
Figura 3.2 – Resposta da malha de controle no modo de oscilação.	56
Figura 3.3 – Resposta ao degrau com o segundo método de Ziegler-Nichols.	57
Figura 3.4 – Estrutura do controlador <i>Fuzzy</i>	58
Figura 3.5 – Funções de pertinência para a entrada E.	58
Figura 3.6 – Funções de pertinência para a entrada CE.	59
Figura 3.7 – Funções de pertinência para a saída u.	59
Figura 3.8 – Comparação da resposta ao degrau com o controlador PID e <i>Fuzzy</i> PD.	60
Figura 3.9 – Malha de controle com o controlador Aproximado.	61
Figura 3.10 – Funções de Pertinência no <i>ANFIS</i>	62
Figura 3.11 – Resposta ao degrau do controlador aproximado com o distúrbio externo.	64
Figura 3.12 – Resposta ao degrau do controlador aproximado com o ajuste de ganho.	66
Figura 4.1 – Sistema de controle de posição considerado no ensaio.	68
Figura 4.2 – Diagrama de blocos da malha de controle para o ensaio.	69
Figura 4.3 – Protótipo do sistema utilizado nos experimentos práticos.	69
Figura 4.4 – Sinais de entrada e saída usados na identificação do sistema.	71
Figura 4.5 – Resposta do modelo ARX e os dados do ensaio.	71
Figura 4.6 – Simulação da malha de controle com o controlador convencional PD.	72

Figura 4.7 – Resposta do controlador <i>Fuzzy</i> PD com 5 variáveis linguísticas.....	73
Figura 4.8 – Funções de pertinência da variável E do controlador <i>fuzzy</i>	74
Figura 4.9 – Funções de pertinência da variável CE do controlador <i>fuzzy</i>	74
Figura 4.10 – Funções de pertinência da variável u do controlador <i>fuzzy</i>	75
Figura 4.11 – Resposta do controlador <i>Fuzzy</i> PD com 7 variáveis linguísticas.....	76
Figura 4.12 – Exemplo do controlador <i>Fuzzy</i> PD em uma malha de controle.	76
Figura 4.13 – Resposta da malha de controle que define os dados do SI.....	77
Figura 4.14 – Estrutura do controlador Aproximado.	79
Figura 4.15 – Resposta do controle de posição para a referência em 90 graus.	80
Figura 4.16 – Resposta do controle de posição com variação no sinal de entrada.....	81
Figura 4.17 – Resposta do controle de posição com carga e variação no sinal de referência. .	83

Lista de Tabelas

Tabela 2.1 – Regras de ajuste manual.	25
Tabela 2.2 – Regras de Ziegler-Nichols.	26
Tabela 2.3 – Modelo discreto dos índices de desempenho baseados no erro.....	28
Tabela 2.4 – Representação da tabela genérica de um sistema de informação.	41
Tabela 2.5 – Sistema de informação associado ao exemplo.....	44
Tabela 2.6 – Matriz de discernibilidade do exemplo.....	45
Tabela 2.7 – Representação tabular numérica de um SI.	46
Tabela 2.8 – Sistema de Informação associado com o modelo estabelecido.	50
Tabela 3.1 – Parâmetros do motor CC.	54
Tabela 3.2 – Critério de Estabilidade de Routh.....	56
Tabela 3.3 – Base de regras do controlador <i>Fuzzy</i> PD usado na simulação.....	60
Tabela 3.4 – Sistema de Informação obtido com o controle <i>Fuzzy</i>	61
Tabela 3.5 – Coeficientes do conjunto de regras.....	63
Tabela 3.6 – Comparação do desempenho dos controladores.....	65
Tabela 4.1 – Base de regras do controlador <i>Fuzzy</i> PD usado no ensaio.	75
Tabela 4.2 – Sistema de Informação do modelo de controle.....	77
Tabela 4.3 – Comparação dos resultados dos controladores.	82

Lista de Abreviaturas

A/D - conversor analógico/digital

ANFIS - *Adaptive Network Based Fuzzy Inference System*

ARX - *AutoRegressive model with eXogenous inputs*

D/A - conversor digital-analógico

IAE - *Integrated absolute error*

IE - *Integrated Error*

ISE - *Integrated square error*

ITAE - *Integrated of the time multiplied by absolute error*

Motor CC - Motor de corrente contínua

PD - Proporcional e Derivativo

PID - Proporcional, Integral e Derivativo

PRBS - *Pseudo Random Binary Signal*

TCA - Teoria dos Conjuntos Aproximados

USB - *Universal Serial Bus*

Lista de Símbolos

R_a - Resistência de armadura

L_a - Indutância de armadura

I_a - Corrente de armadura

V_a - Tensão de armadura

E_b - Força contra-eleromotriz

ω - Velocidade angular

T_m - Torque do motor

θ - Posição angular do eixo do rotor

J_m - Momento de inércia

B_m - Coeficiente de amortecimento

K_T - Constante de torque

K_b - Constante da força eletromotriz

s - Variável de Laplace

T_a - Período de amostragens

P_{cr} - Período de oscilações

T_i - Tempo integrativo

T_d - Tempo derivativo

T_r - Tempo de subida

T_s - Tempo de acomodação

M_p - *Overshoot*

K_{cr} - Ganho crítico

K_p - Ganho proporcional

K_i - Ganho integral

K_d - Ganho derivativo

e - Erro da malha de controle

r - Referência ou *set point*

y - Saída do sistema

N - Número de amostras

q^{-1} - Operador de atraso das amostras

$\hat{\theta}$ - Parâmetro estimado pelo método dos mínimos quadrados

\hat{y} - Saída estimada do modelo

J - Função de custo

r_i - Regras

μ_{r_i} - Grau da função de pertinência

GE - Ganho que multiplica o erro de um controlador *fuzzy*

GCE - Ganho da variação do erro de um controlador *fuzzy*

GU - Ganho absoluto de um controlador *fuzzy*

$B_*(O)$ - Aproximação Inferior de O

$B^*(O)$ - Aproximação Superior de O

$NEG(O)$ - Região negativa de O

$RF(O)$ - Região de fronteira de O

$\alpha_B(O)$ - Coeficiente de Imprecisão de O

$P_n^{(m)}$ - Conjuntos poliedrais

$y^{(k)}$ - Valores de x_i que representam os limites inferiores

$y^{(m)}$ - Valores de x_i que representam os limites superiores

Sumário

1	Introdução	16
1.1	Definição do problema	16
1.2	Objetivos	16
1.3	Justificativa	17
1.4	Estrutura da dissertação.....	18
2	Revisão Bibliográfica	19
2.1	Motores de corrente contínua.....	19
2.1.1	Modelagem matemática	20
2.1.2	Técnicas de controle empregadas.....	21
2.2	Sistemas de Controle.....	23
2.2.1	Sintonia do Controlador PID.....	24
2.2.2	Método de Ziegler-Nichols	25
2.2.3	Crítérios de desempenho baseados no erro	27
2.2.4	Desenvolvimento baseado em modelo.....	28
2.3	Identificação de Sistemas	29
2.3.1	Representação geral para o modelo polinomial	30
2.3.2	Modelo ARX.....	31
2.3.3	Estimador de mínimos quadrados	31
2.4	Controle <i>Fuzzy</i>	32
2.4.1	Modelo de Mamdani	33
2.4.2	Modelo de Takagi-Sugeno	36
2.4.3	Sintonia do Controlador <i>Fuzzy</i>	37
2.4.4	Controlador <i>Fuzzy</i> Proporcional-Derivativo	38
2.5	A Teoria dos Conjuntos Aproximados.....	40
2.5.1	Conceitos básicos	40
2.5.2	Modelos baseados em regras.....	46
2.5.3	Ferramenta Computacional	47
2.5.4	Conversão de modelos <i>Fuzzy</i> em modelos Aproximados.....	47
2.5.5	Comparação com Sistemas Fuzzy.....	50
3	Conversão entre os Modelos Baseados em Regras.....	53
3.1	Caracterização do problema	53
3.2	Projeto do compensador PID	54
3.3	Projeto do compensador <i>Fuzzy</i> PD	57

3.4	Obtenção do compensador Aproximado.....	60
3.5	Conclusão.....	66
4	Desenvolvimento de um controlador Aproximado	67
4.1	Caracterização do problema.....	67
4.2	Construção do protótipo do sistema de posicionamento.....	67
4.3	Identificação de um modelo ARX.....	70
4.4	Projeto do controlador <i>Fuzzy</i> PD	72
4.5	Projeto do controlador Aproximado.....	76
4.6	Conclusão.....	83
5	Conclusão Geral	84
	Apêndice A – Diagramas de Blocos usados no Sistema de Controle	86
	Apêndice B – Códigos Fonte do Sistema de Controle	88
	Referências	93

1 Introdução

As máquinas de corrente contínua caracterizam-se por sua versatilidade e são projetadas para atuar no regime permanente e em operações dinâmicas (FITZGERALD, KINGSLEY e UMAS, 2006). Devido a sua simplicidade de acionamento, dimensões reduzidas e compatibilidade com sistemas digitais, percebe-se a contínua utilização dos motores CC em sistemas industriais, manipuladores robóticos, eletrodomésticos e em muitas aplicações que requerem o controle de posição.

1.1 Definição do problema

Existe uma complexidade nos sistemas reais, especificamente nos motores CC que possuem efeitos não lineares. Essa característica contribui para a redução do desempenho dos controladores convencionais quando aplicados em máquinas CC. Para o controle de posição ser exato é necessário um controlador robusto que seja pouco afetado por fatores externos, realize o rastreamento da referência e possua uma característica de resposta rápida. A utilização de abordagens envolvendo técnicas de inteligência artificial representam alternativas eficientes nessas aplicações.

Os controladores que possuem a lógica *fuzzy* são bem aplicados em sistemas controle devido a sua capacidade de lidar com entradas imprecisas, vagas ou com ruídos (VERMAAS, 2010). Além disso, o modelo *fuzzy* possui uma melhor exatidão nos modelos resultantes em virtude das regiões nas quais os conjuntos difusos se sobrepõem. Essas vantagens acabam gerando um maior custo computacional (GUARACY, PINHEIRO e FERREIRA, 2014).

Um dos principais problemas do controlador *fuzzy* é o tempo de computação que é muito maior em relação aos controladores convencionais, por exemplo. Isso ocorre devido os procedimentos complexos como a fuzificação e defuzificação (AMLASHI, 2012). Demonstra-se nesse projeto o desenvolvimento de um compensador com o desempenho semelhante ao controlador *fuzzy*, e que não necessita dos procedimentos citados.

1.2 Objetivos

O principal objetivo no desenvolvimento deste projeto de pesquisa é utilizar os conceitos dos conjuntos aproximados no desenvolvimento de compensadores baseados em

regras para serem aplicados em um sistema de controle de posição. As características que permitem esse tipo de aplicação são descritas em Pinheiro *et al.* (2010), na qual mostra como esses conceitos podem ser usados na aproximação de funções e obtenção de modelos de sistemas dinâmicos complexos com características não lineares e/ou parâmetros variáveis no tempo.

O desenvolvimento de controladores baseados na Teoria dos Conjuntos Aproximados (TCA) é tratado nesta dissertação em duas abordagens. A primeira consiste no processo de conversão do modelo de um controlador *fuzzy* em um controlador aproximado. A segunda abordagem utiliza o modelo funcional de um sistema com o controlador *fuzzy* para determinar a base de conhecimento que define o controlador aproximado. Uma das características em comum nas duas abordagens é a utilização do sistema de informação para a geração das regras aproximadas.

1.3 Justificativa

A capacidade de transformar um conjunto de dados em conhecimento é uma das maiores contribuições da teoria dos conjuntos aproximados. Dificilmente a base de conhecimento é caracterizada por informações completas ou precisas, tornando difícil a obtenção de conclusões válidas sobre o processo. A TCA pode ser muito útil em aplicações práticas devido a possibilidade de lidar com essas imperfeições do conhecimento (DUBOIS e PRADE, 1990).

Os modelos baseados em regras constituem uma ferramenta importante na representação de sistemas dinâmicos e modelos de controle que utilizam técnicas de inteligência artificial (PINHEIRO, CAMATTA e REZEK, 2012). Este é um ponto em comum entre os sistemas *fuzzy* e os conjuntos aproximados, onde as regras estabelecem relações entre as variáveis de entrada e permitem a representação de um determinado modelo através de procedimentos computacionais.

O diferencial do método proposto como estratégia de controle está na possibilidade de desenvolver um compensador que possua uma estrutura simples, com o menor tempo de processamento e que apresente respostas dinâmicas adequadas para o sistema. Essas características contribuem para aplicações com motores CC que requerem um controle de posição rápido, exato e que possa ser programado em processadores de baixo custo.

É importante salientar que os conceitos dos conjuntos aproximados e lógica *fuzzy* não foram combinados para o desenvolvimento de uma técnica de controle, ou seja, cada teoria

define seu respectivo controlador, sendo possível a comparação dos seus resultados através de simulações computacionais e avaliações dos resultados experimentais.

1.4 Estrutura da dissertação

Esta dissertação está dividida em cinco capítulos. O segundo capítulo apresenta uma breve revisão dos conceitos sobre o modelo dinâmico do motor CC e das teorias de controle envolvidas.

A metodologia de conversão do modelo de um controlador *fuzzy* em um controlador aproximado é abordada no terceiro capítulo. O modelo matemático do processo e as simulações computacionais fazem parte do desenvolvimento das estratégias de controle.

O capítulo 4 apresenta a parte experimental de um sistema de controle de posição e utiliza o controlador *fuzzy* como modelo operacional para o projeto do controlador aproximado. Finalmente, no quinto capítulo são discutidos os resultados obtidos e deixa as sugestões para projetos futuros.

2 Revisão Bibliográfica

Neste capítulo são abordados os conceitos básicos envolvidos e de fundamental importância para compreensão da metodologia de controle utilizada.

2.1 Motores de corrente contínua

A alta confiabilidade, compatibilidade com os sistemas digitais e o baixo consumo de energia mantêm a contínua utilização dos motores CC em sistemas industriais, manipuladores robóticos, eletrodomésticos e nas demais aplicações que requerem o controle de posição do motor (AHMED *et al.*, 2013; AKBARI-HASANJANI, JAVADI e SABBAGHI-NADOOSHAN, 2015).

As vantagens das máquinas CC provêm da variedade de características de operação que podem ser obtidas a partir do método de excitação dos enrolamentos de campo que influencia nas características de regime permanente e, no caso de sistemas de controle, o comportamento dinâmico da máquina. As principais partes que compõem o aspecto construtivo dos motores CC (FITZGERALD, KINGSLEY e UMAS, 2006; GUIMARÃES, 2013):

- Estator: É a parte fixa do motor. O estator gera o campo magnético fixo a partir do enrolamento de campo ou ímã permanente.
- Rotor: É onde encontra-se o enrolamento de armadura. Trata-se da parte rotativa do motor CC. Quando a corrente contínua circula na bobina é gerado um campo magnético que interage com o campo magnético do estator, criando um torque proporcional a corrente elétrica aplicada.
- Comutador: Permite manter a rotação do motor. Tem a função de inverter o sentido da corrente de armadura, logo antes da armadura se alinhar com o campo, assim é gerado um torque que mantém o movimento.
- Escovas: Enquanto o motor gira as escovas deslizam sobre o comutador, e realizam o contato elétrico da parte fixa com a parte móvel (armadura). Ou seja, a energia fornecida à armadura é levada aos segmentos do comutador através das escovas.

2.1.1 Modelagem matemática

A principal diferença entre as máquinas CC está na fonte fixa de fluxo, gerado pelo ímã permanente. Dessa forma, o circuito equivalente de um motor CC com excitação externa é semelhante ao de ímã permanente, exceto que não existe conexões de enrolamento de campo (FITZGERALD, KINGSLEY e UMAS, 2006). A Figura 2.1 apresenta o circuito elétrico equivalente do motor CC.

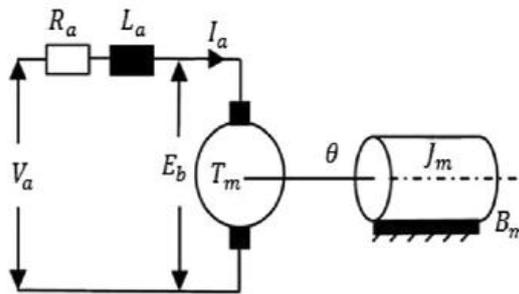


Figura 2.1 – Modelo do circuito elétrico do motor CC.

Fonte: Adaptado de Akbari-Hasanjani, Javadi e Sabbaghi-Nadooshan (2015).

O modelo matemático do motor de corrente contínua é obtido a partir da análise do circuito de armadura da Figura 2.1. Com a aplicação da lei das malhas de Kirchhoff tem-se a Equação diferencial (2.1).

$$V_a = R_a I_a + L_a \frac{dI_a}{dt} + E_b \quad (2.1)$$

Como os condutores na armadura estão atravessando o campo magnético a rotação da armadura produz uma tensão induzida nos condutores que se opõe a tensão aplicada. Essa tensão induzida é denominada força contra-eletromotriz E_b (2.2). Em (2.3), T_m é o torque eletromagnético dado em $N.m$. Essas relações eletromecânicas possuem as constantes K_b e K_T dadas em $V.s/rad$ e $N.m/A$, respectivamente.

$$E_b = K_b \omega \quad (2.2)$$

$$T_m = K_T I_a \quad (2.3)$$

A Equação para o movimento do motor (2.4) é obtida com a soma dos conjugados. A primeira parcela refere-se ao conjugado devido a inércia do eixo, onde J_m é o momento de inércia dado em $Kg.m^2/rad$. B_m é o coeficiente de amortecimento dado em $Nm.s/rad$ e é proporcional a rotação.

$$T_m = J_m \frac{d\omega}{dt} + B_m \omega \quad (2.4)$$

As Equações (2.5) e (2.6) representam a parte elétrica e mecânica, respectivamente. Com a aplicação da transformada de Laplace, considerando as condições iniciais nulas pode-se escrever (2.7) e (2.8).

$$V_a = R_a I_a + L_a \frac{dI_a}{dt} + K_b \omega \quad (2.5)$$

$$K_T I_a = J_m \frac{d\omega}{dt} + B_m \omega \quad (2.6)$$

$$V_a(s) = R_a I_a(s) + L_a s I_a(s) + K_b \omega(s) \quad (2.7)$$

$$K_T I_a(s) = J_m s \omega(s) + B_m \omega(s) \quad (2.8)$$

A corrente de armadura da Equação (2.8) é substituída em (2.7), resultando em (2.9).

$$V_a(s) = \omega(s) \frac{1}{K_T} \left[L_a J_m s^2 + (R_a J_m + L_a B_m) s + (R_a B_m + K_b K_T) \right] \quad (2.9)$$

A relação entre a velocidade do eixo do rotor e a tensão aplicada na armadura é expressa em (2.10). E a relação entre a posição dada em *rad* e velocidade em *rad/s* é definida por (2.11).

$$\frac{\omega(s)}{V_a(s)} = \frac{K_T}{L_a J_m s^2 + (R_a J_m + L_a B_m) s + (R_a B_m + K_b K_T)} \quad (2.10)$$

$$\theta(s) = \frac{1}{s} \omega(s) \quad (2.11)$$

Enfim, a função de transferência a qual se objetiva é dada por (2.12), e relaciona a tensão na armadura com a posição do eixo do motor. Há de se considerar que esta função de transferência deve ser utilizada no estudo das características dinâmicas do motor CC e possibilita o desenvolvimento do controlador proposto no capítulo 3.

$$\frac{\theta(s)}{V_a(s)} = \frac{K_T}{L_a J_m s^3 + (R_a J_m + L_a B_m) s^2 + (R_a B_m + K_b K_T) s} \quad (2.12)$$

2.1.2 Técnicas de controle empregadas

Os motores CC possuem efeitos não lineares como o atrito de Coulomb, folgas, zona morta, entre outros. Esses fatores muitas vezes influenciam no desempenho do controlador, por

isso diferentes estratégias de controle robusto são aplicadas em máquinas CC. Vale considerar o amplo interesse em pesquisas relacionadas com o controle por modos deslizantes, além de outros métodos baseados em inteligência artificial como redes neurais e controle *fuzzy*.

O sistema de controle com estrutura variável (*Variable Structure Control*) utilizando o modo deslizante (*Sliding Mode*) consiste na mudança da estrutura do controlador em resposta a mudança do estado do sistema. O modo deslizante acontece quando a trajetória dos estados da planta atinge a superfície de deslizamento e permanece na mesma. A lei de controle fornece um meio eficaz para controlar plantas lineares e não lineares com uma rápida resposta dinâmica, insensibilidade à variação de parâmetros da planta e rejeição a distúrbios externos (DECARLO *et al.*, 1988).

Os controladores por modos deslizantes possuem um efeito indesejável conhecido como trepidação (*chattering*). Isto ocorre devido ao atraso de resposta causado pelo elemento que realiza o chaveamento. Em sistemas físicos a eliminação do *chattering* deve ser levada em consideração, pois isso pode ocasionar problemas práticos como a excitação de dinâmicas não modeladas, causando a degradação da ação de controle e um desgaste em partes mecânicas móveis do atuador (LORDELO e FAZZOLARI, 2012). Sinha *et al.* (2015) apresentaram uma abordagem utilizando o controlador por modos deslizantes em que um robô móvel de pêndulo invertido é capaz de rejeitar perturbações para distúrbios limitados e realizar o rastreamento em condições variáveis.

Na literatura existem muitas abordagens envolvendo técnicas de controle artificial como alternativa em sistemas não lineares complexos ou sistemas físicos de aplicações reais para o controle de processos. Dursun e Durdu (2016) utilizaram a lógica *fuzzy* para desenvolver o controle de posição em um motor de corrente contínua comparando em seguida com o controlador PID convencional.

Existem poucos trabalhos que abordam os conceitos de conjuntos aproximados em sistemas de controle e utilizem variáveis contínuas ou amostradas. Pinheiro, Camatta e Rezek (2012) apresentaram a teoria dos conjuntos aproximados como uma alternativa eficiente no projeto de controladores baseados em regras para compensar sistemas dinâmicos complexos. Jiang *et al.*, (2015) aplicaram os conceitos de conjuntos aproximados no processo de controle de um carro inteligente em um percurso com obstáculos. O sistema considera uma série de fatores como a velocidade e direção do carro para gerar uma base de regras que proporciona uma resposta de controle rápida e com a trajetória adequada. Machado e Pinheiro (2013) utilizaram a TCA para compensar uma planta de nível com o comportamento não linear.

Esta dissertação utilizou o controlador *fuzzy* como modelo para o desenvolvimento de um controlador considerado recente e que envolve a teoria dos conjuntos aproximados na modelagem de conhecimento baseado em regras.

2.2 Sistemas de Controle

No controle em malha fechada a diferença entre o valor de referência e a variável medida determina o sinal de erro. O controlador utiliza o sinal de erro para calcular o sinal de controle a ser aplicado no processo. A representação da malha de controle em diagrama de blocos é apresentada na Figura 2.2.

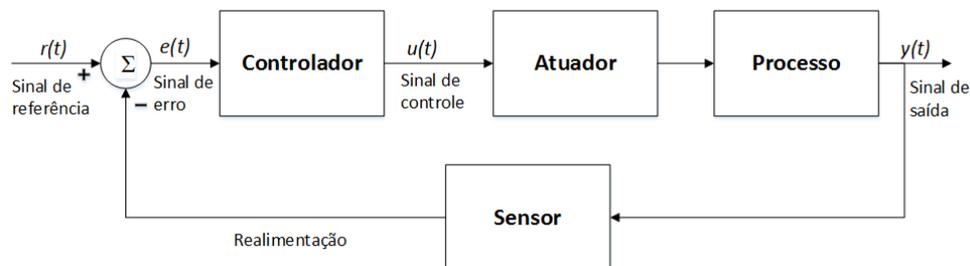


Figura 2.2 – Diagrama de blocos de uma malha de controle.

A realimentação em malhas de controle permite desempenhar características importantes, tais como: melhora a dinâmica do sistema, estabiliza sistemas instáveis em malha aberta, rejeita os distúrbios externos e efeitos de ruídos.

Em muitos problemas práticos as características de desempenho de um sistema de controle são expressas em termos de grandezas no domínio do tempo a uma entrada tipo degrau, conforme a Figura 2.3.

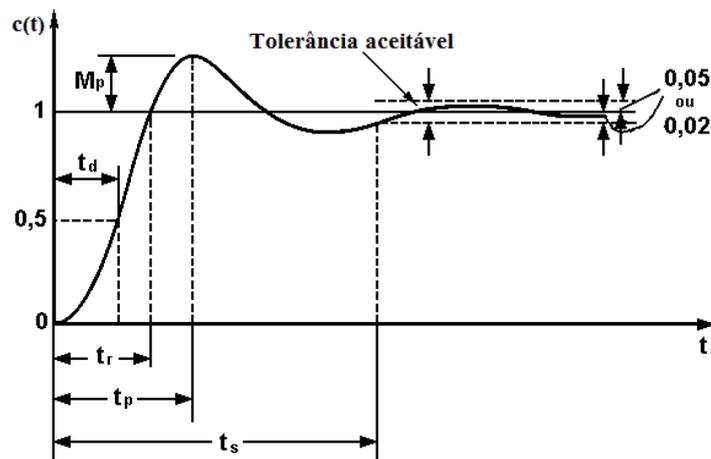


Figura 2.3 – Curva da resposta transitória e de regime estacionário.
Fonte: Ogata (2003).

As especificações das respostas transitórias apresentadas na Figura 2.3 são definidas em Ogata (2003) da seguinte forma:

- Tempo de subida (T_r): É o tempo requerido para que a resposta passe na faixa usual de 10% a 90% do seu valor final.
- Tempo de atraso (T_d): É o tempo necessário para que a resposta alcance a metade do seu valor final.
- Overshoot (M_p): É o valor máximo de pico da curva de resposta, medido a partir da referência durante o regime transitório.
- Tempo de acomodação (T_s): É o tempo para que a resposta alcance uma faixa de 2% a 5% em torno do seu valor final.

2.2.1 Sintonia do Controlador PID

Os controladores PID são comumente utilizados em aplicações industriais devido a sua estrutura simples, baixo custo e algoritmos de controle intuitivamente compreensíveis (AHMED *et al.*, 2013; DURSUN e DURDU, 2016). A Equação (2.13) representa o controlador PID no domínio do tempo, com o sinal de controle $u(t)$ obtido a partir da combinação linear do erro, sua integral e derivada. Também é possível definir $K_i = K_p/T_i$ e $K_d = K_p T_d$. Desse modo, a ação de controle é escrita de acordo (2.14), sendo T_i o tempo integrativo e T_d o tempo derivativo.

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.13)$$

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.14)$$

No controlador digital a ação de controle pode ser aproximada, conforme a Equação (2.15). Neste caso, n indica o instante de tempo, $u(n)$ é a ação de controle discretizada e $e(n)$ é o erro discretizado. O termo integral é substituído por uma soma e o termo derivativo é substituído pela equação diferença de primeira ordem.

$$u(n) = K_p \left(e(n) + \frac{1}{T_i} \sum_{j=1}^n e(j) T_a + T_d \frac{e(n) - e(n-1)}{T_a} \right) \quad (2.15)$$

Existem diferentes métodos para a sintonia de malhas de controle, tais como: o Lugar das Raízes, Resposta em Frequência, ajuste manual, Ziegler-Nichols, entre outros. Os parâmetros devem ser ajustados da melhor forma para obter um controlador com o desempenho

desejado (FRANKLIN, POWEL e EMAMI-NAENI, 2013). O ajuste manual permite reduzir o *overshoot* para um valor tolerável, e pode ser aplicado com base nas regras mostradas na Tabela 2.1. Convém ressaltar que essas regras nem sempre são válidas. Por exemplo, se existir um integrador na planta, então o aumento de K_p o torna mais estável (AHMED *et al.*, 2013; NAMAZOV e BASTURK, 2010).

Tabela 2.1 – Regras de ajuste manual.

	Tempo de subida	Overshoot	Estabilidade
$K_P \uparrow$	Reduz	Aumenta	Degrada
$T_D \uparrow$	Reduz pouco	Reduz	Melhora
$1/T_I \uparrow$	Reduz	Aumenta	Degrada

O ajuste manual do controlador PID é realizado primeiro aumentando o valor do ganho proporcional K_P até obter uma resposta desejável. O ganho derivativo K_d e o ganho integral K_i são então ajustados para melhorar e otimizar a resposta do sistema (AMLASHI, 2012). A desvantagem desse método é o maior tempo necessário para ajustar os parâmetros.

2.2.2 Método de Ziegler-Nichols

Em 1942, Ziegler e Nichols propuseram dois métodos clássicos para a sintonia do controlador PID. O primeiro, consiste na resposta ao degrau do sistema de controle que deve apresentar a curva de reação do processo conforme a Figura 2.4.

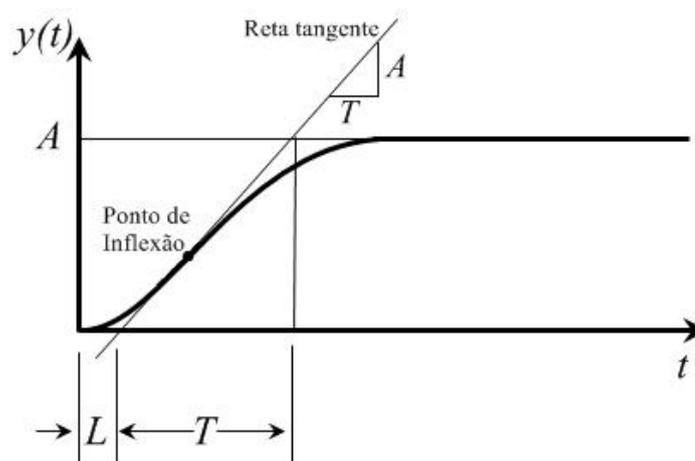


Figura 2.4 – Resposta em malha aberta.
Fonte: Adaptado de Ogata (2003).

A constante de tempo e o atraso de transporte podem ser determinados com a resposta ao degrau. Uma reta tangente é traçada no ponto de inflexão da curva de reação, assim a

inclinação da reta será $R = A/T$ e a intersecção da reta tangente com o eixo do tempo identifica o atraso de transporte L . A curva em formato de “S” pode ser aproximada pela função de transferência (2.16), obtida pela resposta ao degrau e que representa um sistema de primeira ordem com atraso de transporte.

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \quad (2.16)$$

O segundo método proposto é o da sensibilidade crítica, onde os ajustes dos parâmetros são baseados na avaliação da amplitude e a frequência do sistema no limite da estabilidade. Neste caso, o ganho proporcional é aumentado até que o sistema se torne marginalmente estável com oscilações persistentes. O ganho que provoca a oscilação é definido como ganho crítico K_{cr} , e o período de oscilação P_{cr} (FRANKLIN, POWEL e EMAMI-NAENI, 2013). A Figura 2.5 mostra o sistema em malha fechada com o controlador proporcional para a determinação desses parâmetros.

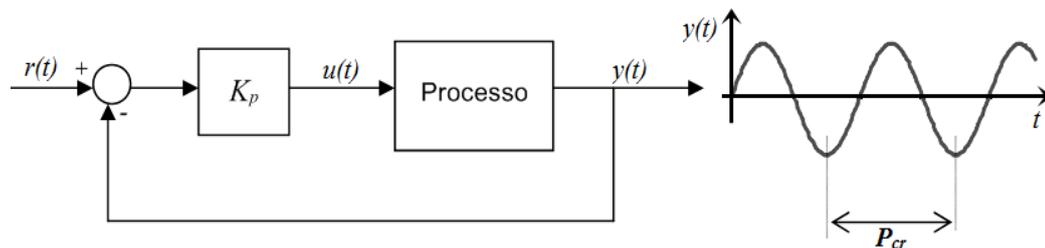


Figura 2.5 – Sistema em malha fechada para a oscilação sustentada.

Fonte: Adaptado de Ogata (2003).

A aplicação das regras de Ziegler-Nichols ilustradas na Tabela 2.2 permite gerar respostas aceitáveis para muitos sistemas em malha fechada. Embora seja comum a sua aplicação quando as dinâmicas das plantas não são conhecidas, estas regras também podem ser utilizadas nas situações em que as dinâmicas das plantas sejam conhecidas, como é o caso da sua aplicação no capítulo 3 desta dissertação.

Tabela 2.2 – Regras de Ziegler-Nichols.

Controle	K_P	T_I	T_D
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$P_{cr}/1,2$	0
PID	$0,6K_{cr}$	$P_{cr}/2$	$P_{cr}/8$

2.2.3 Critérios de desempenho baseados no erro

O índice de desempenho é definido como uma medida quantitativa do desempenho da malha de controle. Esses índices podem ser baseados nas integrais do sinal de erro. O erro $e(t)$ é definido como:

$$e(t) = r(t) - y(t) \quad (2.17)$$

Onde $r(t)$ e $y(t)$ são a referência e saída do sistema, respectivamente. De acordo com Pinto (2014) existem alguns critérios para quantificar o erro ocorrido devido uma perturbação. O primeiro critério é chamado integral do erro (*Integrated Error* – IE).

$$IE = \int_0^T e(t) dt \quad (2.18)$$

Entretanto, em processos oscilatórios este critério não é viável, pois o erro varia entre valores negativos e positivos podendo anular-se. Uma alternativa para esse problema é a aplicação do critério da integral absoluta do erro (*Integrated absolute error* – IAE), conforme a expressão (2.19).

$$IAE = \int_0^T |e(t)| dt \quad (2.19)$$

O critério integral do erro quadrático (*Integrated square error* – ISE) proporciona grandes ponderações para erros grandes e menores ponderações para erros pequenos. A desvantagem desse critério é quantificar em maior escala os erros iniciais dos sistemas oscilatórios.

$$ISE = \int_0^T e(t)^2 dt \quad (2.20)$$

Quando o erro absoluto é ponderado pelo tempo é possível lidar com o problema da quantificação dos erros iniciais em malhas oscilatórias. Para um erro inicial grande tem-se uma pequena ponderação, enquanto os erros que ocorrem após a resposta transitória existe uma maior ponderação. Isto é realizado com o critério integral do tempo multiplicado pelo erro absoluto (*Integrated of the time multiplied by absolute error* – ITAE), dado pela Equação (2.21). Este critério possui uma melhor seletividade em relação ao critério IAE.

$$ITAE = \int_0^T t|e(t)| dt \quad (2.21)$$

Para analisar os sistemas de controle no tempo discreto as integrais devem ser substituídas pelo somatório e o erro ser representado em forma de erro discreto $e(k)$, conforme a Tabela (2.3). Onde N é o número de amostras.

Os critérios baseados no erro também podem ser utilizados como função objetivo em problemas de otimização de ajuste dos controladores. No entanto, este projeto utiliza esses critérios apenas para a medida de desempenho dos controladores.

Tabela 2.3 – Modelo discreto dos índices de desempenho baseados no erro.

Índice de desempenho	Modelo computacional
IE	$\sum_{k=1}^N e(k)$
IAE	$\sum_{k=1}^N e(k) $
ISE	$\sum_{k=1}^N e^2(k)$
ITAE	$\sum_{k=1}^N k e(k) $

2.2.4 Desenvolvimento baseado em modelo

O desenvolvimento baseado em modelo consiste na utilização de ferramentas de desenvolvimento gráfico para a construção do sistema de controle. Stella (2015) afirma que a principal característica e vantagem desse método é o desenvolvimento em uma única plataforma, ou seja, é possível criar a planta do sistema desejado e o seu controlador utilizando a mesma ferramenta computacional. O resultado é um sistema funcional de fácil compreensão e que proporciona uma melhor visualização do comportamento do sistema.

A utilização de ferramentas que possuem uma interface gráfica contribui para o rápido desenvolvimento de modelos de sistemas dinâmicos. Em vez de se utilizar estruturas complexas com todo o sistema representado por códigos extensos, são empregados diagramas de blocos específicos. Atualmente existem ferramentas computacionais avançadas como o Matlab/Simulink[®] que auxiliam na simulação e projeto do controlador através de diagramas de blocos. Esta ferramenta simplifica o desenvolvimento do processo e a avaliação de sistemas complexos.

A etapas para o desenvolvimento baseado em modelo nos sistemas de controle fundamenta-se nos requisitos necessários para o funcionamento do sistema, no modelo funcional e a arquitetura física ou protótipo do sistema a ser controlado. A partir do modelo funcional deve-se definir o controlador desejado e realizar os testes necessários via *software* ou

hardware. A depender dos resultados obtidos a última etapa permite validar o sistema (STELLA, 2015).

Esse procedimento tem sido bastante utilizado em sistemas de controle de movimento, equipamentos industriais, aplicações automotivas e sistemas de comunicação. Um exemplo pode ser observado em Hat *et. al* (2015) que apresentaram o controle de velocidade do motor CC *brushless* utilizando um controlador PID e o desenvolvimento baseado em modelo realizado no Matlab/Simulink®.

2.3 Identificação de Sistemas

O sistemas dinâmicos encontrados na prática são, em última análise, não lineares. Contudo, em muitos casos as aproximações lineares são suficientes para estas aplicações (AGUIRRE, 2004). A representação de um sistema dinâmico geralmente é realizada com a sua modelagem matemática, pois através dela é possível realizar estudos das características dinâmicas, análise de estabilidade, verificação de desempenho, estratégias de controle, etc (PINHEIRO, 2000).

De acordo com Antunes (2007) dentre as técnicas para determinar o modelo matemático existe a modelagem caixa branca, também conhecida por modelagem fenomenológica, sendo baseada nas propriedades físicas dos fenômenos observados. Este modelo depende do conhecimento do sistema a ser modelado. Quando o acesso a estrutura física do sistema é complexo, geralmente é empregada a modelagem empírica ou em caixa preta. A área do conhecimento que estuda esse tipo de técnica de modelagem é conhecida como identificação de sistemas. Nesse caso, é necessário apenas os dados de entrada e saída, o que dispensa o conhecimento prévio do sistema. Os procedimentos básicos para a identificação de sistema é composto pelas seguintes etapas:

- Coleta de dados: Consiste na obtenção dos dados de entrada e saída do sistema. Desse modo, é importante a escolha adequada do sinal de excitação;
- Escolha da representação do modelo: É definida de acordo as características do sistema a ser modelado, observando se ele é linear ou não linear e as condições de variações de parâmetros, por exemplo;
- Escolha da estrutura do modelo: Determinação do número de coeficientes do modelo. O menor número de coeficientes contribui para uma computação mais rápida;

- Estimação de parâmetros: Nesta etapa ocorre a aplicação de algoritmos para determinar a estimação dos coeficientes. Muitos destes algoritmos são baseados no método dos mínimos quadrados e suas variações;
- Teste de validação do modelo: Após a obtenção do modelo, é necessário verificar se a sua característica corresponde ao sistema original. Isso é possível comparando as simulações do modelo obtido com os dados coletados do sistema.

2.3.1 Representação geral para o modelo polinomial

A forma generalizada de algumas representações matemáticas são adequadas à identificação de sistemas. Nessa perspectiva, são usados alguns algoritmos conhecidos para a estimação de parâmetros. O modelo geral linear do tipo discreto é definido pela Equação (2.22) e pode ser reescrito conforme as Equações (2.23) e (2.24).

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k) \quad (2.22)$$

$$y(k) = \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}v(k) \quad (2.23)$$

$$y(k) = H(q)u(k) + G(q)v(k) \quad (2.24)$$

Sendo $u(k)$ a entrada do sistema e $y(k)$ a saída do sistema. A variável $v(k)$ representa erros de medidas e ruídos que podem ser incluídos as medições práticas. E o operador q^{-1} caracteriza os atrasos decorrentes as amostragens das informações, de forma que $y(k)q^{-1} = y(k-1)$ (AGUIRRE, 2004). Os polinômios $A(q)$, $B(q)$, $C(q)$, $D(q)$ e $F(q)$, são definidos como:

$$A(q) = 1 - a_1q^{-1} - \dots - a_{ny}q^{-ny} \quad (2.25)$$

$$B(q) = b_1q^{-1} + \dots + b_{nu}q^{-nu} \quad (2.26)$$

$$C(q) = 1 + c_1q^{-1} + \dots + c_{nv}q^{-nv} \quad (2.27)$$

$$D(q) = 1 + d_1q^{-1} + \dots + d_{nd}q^{-nd} \quad (2.28)$$

$$F(q) = 1 + f_1q^{-1} + \dots + f_{nf}q^{-nf} \quad (2.29)$$

2.3.2 Modelo ARX

O modelo ARX, ou seja, o modelo auto-regressivo com entradas externas (*AutoRegressive model with exogenous inputs*) é obtido a partir do modelo geral. Neste caso, $C(q)$, $D(q)$ e $F(q)$ são considerados iguais a 1. Sendo $A(q)$ e $B(q)$ polinômios arbitrários, a Equação (2.30) representa o modelo ARX. Este modelo também pode ser expresso de forma alternativa por (2.31).

$$A(q)y(k) = B(q)u(k) + v(k) \quad (2.30)$$

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}v(k) \quad (2.31)$$

Convém observar que as funções de transferência $H(q) = B(q)/A(q)$ e $G(q) = 1/A(q)$ representam o comportamento dinâmico do sistema e o ruído, respectivamente. O estimador de mínimos quadrados pode ser aplicado na obtenção de modelos paramétricos como o ARX.

2.3.3 Estimador de mínimos quadrados

O método dos mínimos quadrados é um dos mais conhecidos na estimação de parâmetros de funções nas diversas áreas de ciência e tecnologia, sendo proposto por Gauss em seus estudos sobre astronomia (AGUIRRE, 2004). Assim, a solução se inicia considerando $\hat{\theta}$ o vetor de parâmetros dado por (2.32):

$$\hat{\theta} = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_n \end{bmatrix} \quad (2.32)$$

A partir da estimativa $\hat{\theta}$, e dos regressores ϕ que são as medidas de entradas e saídas com atrasos semelhantes aos apresentados no modelo geral, tem-se a previsão de saída \hat{y} dada na forma matricial em (2.33).

$$\hat{y} = \phi \hat{\theta} \quad (2.33)$$

Sendo y a saída real e \hat{y} a saída estimada, é possível determinar o erro de predição em (2.34).

$$\varepsilon(i) = y(i) - \hat{y}(i) \quad (2.34)$$

A soma dos quadrados da diferença entre os valores medidos reais e os calculados é mínima. A função de custo dos mínimos quadrados é apresentada em (2.35) e (2.36), onde N é o número de amostras.

$$J = \frac{1}{N} \sum_{i=1}^N \varepsilon(i)^2 = \frac{1}{N} \sum_{i=1}^N \varepsilon^T \varepsilon \quad (2.35)$$

$$J = \frac{1}{N} (y - \phi \hat{\theta})^T (y - \phi \hat{\theta}) = y^T y - y^T \phi \hat{\theta} - \hat{\theta}^T \phi^T y + \hat{\theta}^T \phi^T \phi \hat{\theta} \quad (2.36)$$

Derivando (2.36) e igualando a zero a fim de minimizar a função de custo J , tem-se (2.37). Portanto, a Equação (2.38) também é conhecida como estimador de mínimos quadrados, onde $\hat{\phi}^T \hat{\phi}$ é a matriz de informação dos regressores e $\hat{\theta}$ é a variável com os parâmetros desejados.

$$\frac{\partial J}{\partial \hat{\theta}} = -2\hat{\phi}^T y + 2\hat{\phi}^T \hat{\phi} \hat{\theta} = 0 \quad (2.37)$$

$$\hat{\theta} = [\hat{\phi}^T \hat{\phi}]^{-1} \hat{\phi}^T y \quad (2.38)$$

2.4 Controle *Fuzzy*

A Teoria dos Conjuntos Difusos proposta por Lofti A. Zadeh em seu artigo (ZADEH, 1965) constitui uma generalização da teoria dos conjuntos clássicos ou *crisp*.

A definição algébrica na lógica convencional, ou teoria clássica dos conjuntos assume o conceito de pertinência a um conjunto $\{0, 1\}$, ou seja, para um conjunto A em um universo X , os elementos deste universo pertencem ou não pertencem a um dado conjunto, conforme a afirmação expressa em (2.39).

$$f_A = \begin{cases} 1 & \text{se e somente se } x \in A \\ 0 & \text{se e somente se } x \notin A \end{cases} \quad (2.39)$$

A proposição de Zadeh é mais abrangente, em um conjunto difuso a função característica associada pode assumir infinitos valores no intervalo $[0, 1]$. Considera-se então que um conjunto *fuzzy* F em um universo de discurso X é definido por uma função de pertinência de grau $\mu_F(x)$, ou seja, $\mu_F(x): X \rightarrow [0, 1]$, e pode ser representado como um conjunto de pares

ordenados com valores de uma variável x e os respectivos graus de pertinência definido por (2.40). De modo que $\mu_F(x)$ revela o quanto x é compatível com o conjunto F (ALMEIDA, 2014).

$$F = \{(x, \mu_F(x) | x \in X)\} \quad (2.40)$$

Uma das vantagens dos conjuntos difusos refere-se as escolhas ou atribuições de valores dos parâmetros das funções de pertinência que possibilitam a modelagem de informações vagas ou imprecisas. A outra vantagem é a respeito das variáveis linguísticas que auxiliam de uma forma qualitativa para o controle do sistema, pois permitem representar um dado problema através de expressões linguísticas como Negativo, Zero, Positivo, por exemplo (LIMA, SANTOS e PINHEIRO, 2014). Nesse contexto, as características incorporam tanto o conhecimento objetivo dos valores envolvidos quanto o conhecimento subjetivo das variáveis linguísticas.

Os sistemas *fuzzy* são paradigmas computacionais para o processamento de informação e possibilita executar tarefas que exijam algum tipo de inteligência semelhante as ponderações nas tomadas de decisões de um ser humano. Na área de sistemas de controle, esta técnica é uma alternativa eficiente quando as plantas são de difícil modelagem, tempo morto elevado e/ou zona morta alta (BEZERRA, 2009).

2.4.1 Modelo de Mamdani

O primeiro controlador baseado em conjuntos *fuzzy* foi proposto por Mamdani (1974) para o controle de uma máquina a vapor. O controlador *fuzzy* possui três etapas: Fuzificação, Sistema de Inferência e Defuzificação. 1) A interface de Fuzificação utiliza as funções de pertinência contidas na base de conhecimento e realiza um mapeamento de valores reais de entrada *crisp* (convencional) para conjuntos *fuzzy*. 2) O Sistema de Inferência é incorporado à base de regras e a partir dos valores de entrada avalia quais regras estão ativas em determinado período de tempo para gerar as ações de controle. 3) A Defuzificação converte um conjunto *fuzzy* em um sistema de valor *crisp*. Esta etapa gera o sinal de saída para o controle de um determinado processo (DURSUN e DURDU, 2016). A Figura 2.6 apresenta a estrutura do modelo de Mamdani.

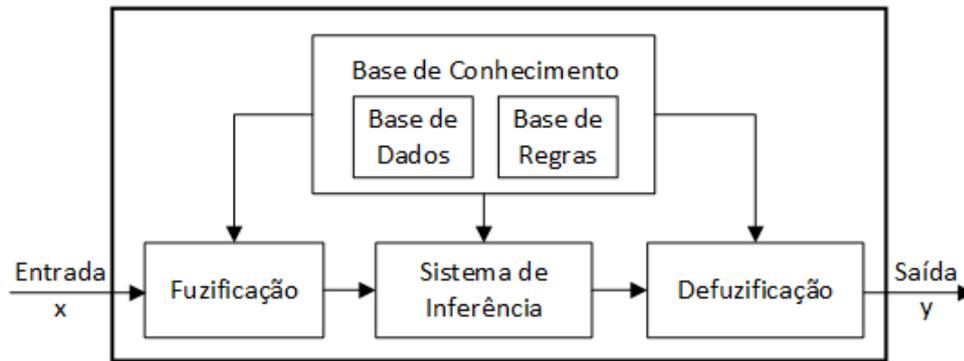


Figura 2.6 – Diagrama do modelo de inferência de Mamdani.
Fonte: Adaptado de Vermaas (2010).

Os blocos de Fuzificação, Sistema de Inferência e Defuzificação, compõem a máquina de inferência *fuzzy*. O sistema de Inferência é o componente que a depender do conjunto de entradas, gera as saídas *fuzzy* de acordo com a relação definida na Base de Conhecimento (VERMAAS, 2010). A base de conhecimento representa o modelo do sistema a ser controlado. É constituída por uma base de dados (funções de pertinência), onde é determinado o domínio das variáveis e por uma base de regras difusas, onde está o conhecimento do controle na forma de regras (sistema de decisão).

As funções de pertinência também são chamadas de termos e podem ter diferentes formas, a depender do contexto envolvido e do modo que se deseja representar. As mais utilizadas na literatura são a triangular, a trapezoidal e a gaussiana.

O conjunto de regras descreve o comportamento dinâmico do sistema e podem ser obtidas de várias formas, tais como: conhecimento de especialistas, análise qualitativa de dados, técnicas de agrupamento, técnicas de regressão, métodos de treinamento e associações com técnicas de inteligência artificial (LIMA, SANTOS e PINHEIRO, 2014). As regras expressam o conhecimento na forma *IF* (conjunto de condições satisfeitas) *THEN* (conjunto de consequências). Basicamente, a regra de controle difusa é uma declaração condicional em que o antecedente é uma condição em seu domínio de aplicação e o consequente é a ação de controle para o sistema.

O modelo tipo Mamdani possui a característica de tanto os seus antecedentes quanto os seus consequentes serem mapeados com conjuntos difusos. A estrutura de duas regras linguísticas são representadas na forma (2.41).

$$\begin{aligned} R_1 : & \text{if } X \text{ is } A_1 \text{ and } Y \text{ is } B_1 \text{ then } Z \text{ is } C_1 \\ R_2 : & \text{if } X \text{ is } A_2 \text{ and } Y \text{ is } B_2 \text{ then } Z \text{ is } C_2 \end{aligned} \quad (2.41)$$

As regras possuem duas entradas e uma saída. Onde X e Y são variáveis linguísticas relacionadas ao processo, e Z é a variável linguística de controle. Na primeira regra, por exemplo, estas variáveis são modeladas pelos conjuntos *fuzzy* A_1 , B_1 e C_1 .

Uma técnica comum na composição de vários conjuntos *fuzzy* de entrada para cada regra é o método de inferência MAX-MIN. O “MIN” implica em um conectivo “E” e resulta na intersecção dos termos de entrada. O “MAX” implica em um conectivo “OU”, responsável pela união dos termos de saída (BEZERRA, 2009). Um exemplo da representação gráfica do algoritmo de inferência MAX-MIN pode ser visualizada na Figura 2.7.

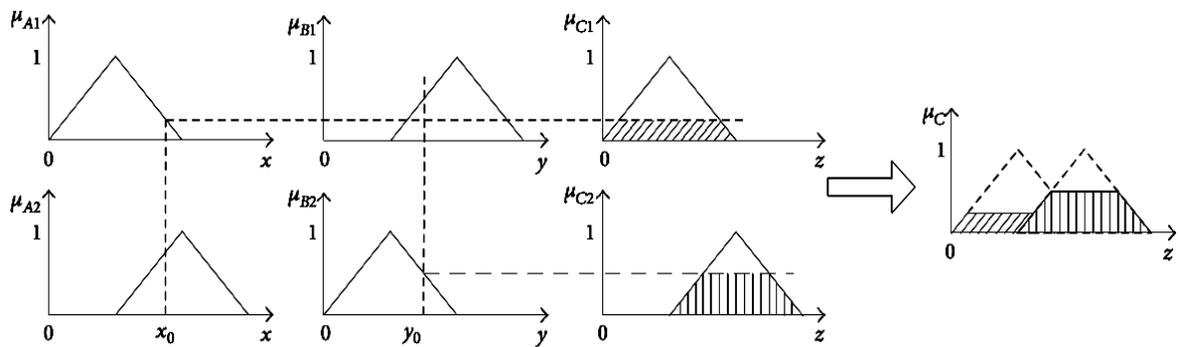


Figura 2.7 – Representação gráfica do método de inferência MAX-MIN.

Fonte: Chen *et. al.* (2015).

A Figura 2.7 mostra um exemplo de cálculo da saída *crisp* após a composição dos conjuntos *fuzzy*, onde x_0 e y_0 correspondem aos valores *crisp* de entrada. Para cada regra r_i é efetuada a combinação com os valores de pertinência μ_{ri} dos antecedentes (variáveis de entrada do controlador). As regras são ativadas individualmente gerando um conjunto de saídas *fuzzy*, estas saídas são agregadas no processo de defuzificação para compor um valor de saída *crisp*.

A capacidade de converter as informações qualitativas em informações quantitativas dada pela defuzificação é realizada por diferentes métodos. Os principais métodos utilizados são Centro de Área e Média dos Máximos. O método que calcula o Centro de Área da função também é conhecido como Centro de Gravidade ou Centróide e determina a união das funções de pertinência da saída por meio da expressão (2.42). De acordo com Bezerra (2009), este método quando aplicado em sistemas de malha fechada mantém o sinal de controle contínuo, o que aumenta a estabilidade e evita as oscilações.

$$y = \frac{\sum_{i=1}^N y_i \mu_C(y_i)}{\sum_{i=1}^N \mu_C(y_i)} \quad (2.42)$$

Onde N é o número de níveis de discretização, ou seja, está relacionado ao número de regras cujo o valor da função de pertinência é maior do que zero, y_i é a posição do centróide da função de pertinência individual e $\mu_C(y_i)$ representa o valor da função de pertinência em C (saída *fuzzy*).

O método média dos máximos, como o nome sugere, calcula a média entre os elementos do universo de discurso que possuem os maiores valores da função de pertinência, conforme a Equação (2.43).

$$y = \sum_{i=1}^n \frac{y_i}{n} \quad (2.43)$$

Onde i representa o i -ésimo elemento do universo de discurso, e n define o universo total desses elementos, cuja função de pertinência é máxima.

2.4.2 Modelo de Takagi-Sugeno

Na década de 80 foi proposta uma estrutura de inferência também baseada em conjuntos *fuzzy* (TAKAGY e SUGENO, 1985). Este modelo demonstra ser um bom aproximador de funções e possui uma estrutura estabelecida com regras funcionais ou de Takagi-Sugeno, expressa por (2.44).

$$R_1 : \text{if } X \text{ is } A_i \text{ and } Y \text{ is } B_i \text{ then } Z = f_i(x, y) \quad (2.44)$$

Conforme se observa nas regras (2.41) e (2.44) a fuzificação das entradas dos operadores antecedentes é semelhante ao modelo de Mamdani. A principal diferença está no consequente das regras do modelo Takagi-Sugeno que apresenta a saída na forma de funções associadas. Como os consequentes de cada regra deste modelo não são conjuntos *fuzzy*, mas funções *crisp*, a saída de todas as regras é determinada pela média ponderada das saídas geradas por cada uma das regras.

Nesse contexto, cabe ressaltar que a estrutura de regras do tipo Mamdani geralmente é utilizada quando tarefas de extração de conhecimento e análise de dados são importantes. Enquanto a estrutura Takagi-Sugeno é aplicada em abordagens que buscam a exatidão na aproximação da função (VERMAAS, 2010).

2.4.3 Sintonia do Controlador *Fuzzy*

Os controladores *fuzzy* possuem ganhos de controle variáveis que contribuem para a melhoria no desempenho de sistemas não lineares, com incertezas e ruídos no processo (PILTAN, 2016).

A sintonia de uma malha de controle *fuzzy* é realizada através de modificações em fatores de escala, ajuste nas funções de pertinência (larguras e posições), ou por meio de alterações nas regras. Geralmente são utilizados analogias com outros sistemas e abordagens mais usuais como heurísticas e o método da tentativa e erro (PINHEIRO, 2000). A definição das funções de pertinência e das regras são importantes para alcançar as especificações de projeto, pois se relacionam com a estabilidade, exatidão e o tempo da resposta.

A estrutura do controlador *Fuzzy* e o seu projeto consiste nas seguintes etapas (SUDHA e ANITA, 2012; VERMAAS, 2010):

1. Definir os universos de discurso das variáveis linguísticas de entrada e saída do controlador;
2. Determinar o número de funções de pertinência de conjuntos difusos que representam cada termo;
3. Estabelecer as regras de controle;
4. Descrever o sistema em termos de conjuntos *fuzzy*, ou seja, determinar o método de inferência;
5. Definir a forma de defuzificação e a atuação do controlador.

Pinheiro e Gomide (1999) mostraram que para as funções de pertinência não simétricas e não igualmente espaçadas o controlador *fuzzy* apresenta características não lineares, pois os seus ganhos são alterados em função da magnitude do sinal de entrada. Esse comportamento torna-se adequado para o controle de processos complexos (não lineares, com variações de parâmetros no tempo, etc.). Outro aspecto importante é que quanto maior o número de conjuntos, até determinado limite, tem-se uma maior exatidão e conseqüentemente a demanda computacional aumenta.

De uma forma geral, as principais características da Lógica *Fuzzy* quando aplicada como técnica de controle (BEZERRA, 2009; ZADEH e YAZDIAN, 2006):

- Não necessita de modelos matemáticos de sistemas de controle;
- O controlador é não linear e flexível;

- É confiável e robusto, pois não é sensível a variação de parâmetros do sistema e permite utilizar proposições linguísticas (regras);
- Não requer entradas precisas, são resistentes a perturbações externas e alterações na planta;
- Solução rápida por ser facilmente embarcado em microprocessadores.

Processos que possuem um alto nível de complexidade, como ordem mais altas, atrasos, características oscilatórias, modelos não lineares e sistemas sem o modelo matemático exato, são características que dificultam o desempenho dos controladores PID convencionais devido a sua estrutura de controle que não garante os requisitos necessários em muitas situações práticas (EL-NAGAR e EL-BARDINI, 2014). O emprego do controle *fuzzy* é uma alternativa nesses casos, pois proporciona uma metodologia para construir controladores não lineares através de um conhecimento heurístico sobre o controle do sistema mesmo sem o modelo matemático do processo.

2.4.4 Controlador *Fuzzy* Proporcional-Derivativo

Os controladores desenvolvidos através da Lógica *Fuzzy* podem ser projetados em diferentes estruturas classificadas de forma semelhante ao controlador PID e suas variações, sendo representadas por *Fuzzy* P, *Fuzzy* PD, *Fuzzy* PID, *Fuzzy* Incremental e *Fuzzy* PD+I. Nesta dissertação, é utilizada a estrutura do controlador *Fuzzy* PD mostrada na Figura 2.8.

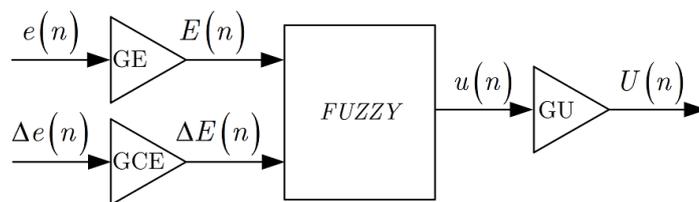


Figura 2.8 – Controlador *Fuzzy* PD.
Fonte: Almeida (2014).

Na descrição do controlador *Fuzzy* PD é considerada a Equação (2.15) que caracteriza a saída de um controlador PID discreto. Anulando-se o termo integrativo tem-se uma ação de controle PD convencional, conforme expressa a Equação (2.45).

$$u(n) = K_p \left(e(n) + T_d \frac{e(n) - e(n-1)}{T_a} \right) \quad (2.45)$$

Para se comparar com o controlador convencional é necessário considerar as entradas com o erro e a derivada do erro. Uma aproximação prática da derivada do erro é definida na Equação (2.46).

$$\dot{e}(n) \approx \Delta e(n) = e(n) - e(n-1) \quad (2.46)$$

Conforme se observa na Figura 2.8 a estrutura do controlador possui algumas terminologias, tais como: o sinal do erro $e(n)$, o ganho que multiplica o erro GE , o erro ponderado pelo ganho $E(n) = GE.e(n)$, a variação do erro $\Delta e(n)$, o ganho GCE , a variação do erro ponderada $\Delta E(n) = GCE.\Delta e(n)$ e o ganho absoluto GU (ALMEIDA, 2014).

O controlador *Fuzzy* PD proporciona a planta um sinal de controle $U(n)$, com a multiplicação do ganho GU pela função não linear do erro e variação do erro, conforme (2.47).

$$U(n) = f(GE.e(n), GCE.\Delta e(n)).GU \quad (2.47)$$

Com a linearização, a função definida pode ser aproximada. Assim, é possível obter a Equação (2.48).

$$f(GE.e(n), GCE.\Delta e(n)) \approx GE.e(n) + GCE.\Delta e(n) \quad (2.48)$$

Substituindo a Equação (2.48) em (2.47), tem-se o seguinte desenvolvimento matemático:

$$\begin{aligned} U(n) &\approx (GE.e(n) + GCE.\Delta e(n)).GU \\ U(n) &\approx GE.GU \left(e(n) + \frac{GCE}{GE}.\Delta e(n) \right) \end{aligned} \quad (2.49)$$

Comparando as Equações (2.15) e (2.49), obtém-se as relações entre os ganhos conforme descrito em (2.50) e (2.51).

$$GU = \frac{K_p}{GE} \quad (2.50)$$

$$GCE = GE.T_d \quad (2.51)$$

O desenvolvimento estabelecido para definir as relações entre os ganhos do controlador *Fuzzy* PD e PD convencional sugerem as mesmas características de respostas (ALMEIDA, 2014). A estrutura de controle PD tem um bom desempenho no regime transitório, tendendo ao menor tempo de acomodação e a característica de redução dos sobressinais. No entanto, esta ação de controle possui algumas desvantagens como o maior tempo de subida e não atua na

correção do erro em regime permanente. Uma possível alternativa para reduzir o erro em regime permanente seria adicionar a ação integral (OBAID *et al.*, 2010). Há de se considerar a medida que são adicionadas entradas ao sistema *fuzzy*, ocorre um aumento exponencial no número de regras e na requisição de processamento, pois a quantidade de regras é definida em função das variáveis de entrada envolvidas no sistema. Isto contribui para que o controlador *Fuzzy PD* possa ter uma menor quantidade de regras em relação a um controlador *Fuzzy PID*.

2.5 A Teoria dos Conjuntos Aproximados

A Teoria dos Conjuntos Aproximados (TCA) foi proposta por Pawlak (1982) com base na teoria clássica de conjuntos, trata-se de uma ferramenta matemática capaz de lidar com incertezas e imprecisões. Este conceito é relevante em várias aplicações que empregam técnicas de Inteligência Artificial, especialmente em áreas de pesquisa, como aprendizado de máquinas, sistemas especialistas, sistemas de decisão e mineração de dados (PAWLAK e SKOWRON, 2007).

A principal vantagem da TCA na análise de dados é que não existe a necessidade de conhecimentos prévios ou adicionais sobre os dados processados, tais como distribuições de probabilidade, intervalos de confiança ou valores de possibilidades (PAWLAK, 1991).

A inteligência de um sistema desenvolvido com a TCA está em sua capacidade de classificar determinadas situações a que é submetido para tomada de decisões adequadas. A relação de indiscernibilidade existente nessa teoria permite considerar eventuais relações nos dados de um sistema de informação, agrupando valores que são semelhantes ou equivalentes (TAJIRI, 2009; LIMA, SANTOS e PINHEIRO, 2014). Esse conceito é importante para definir os atributos redundantes, tornando possível a redução da quantidade de informações que não são essenciais no sistema de informação e a extração do conhecimento sob a forma de regras concisas.

2.5.1 Conceitos básicos

Um espaço de aproximado é definido por um par ordenado $S = (U, A)$, onde U é um conjunto finito não vazio de objetos ou exemplos chamados de universo, e A é um conjunto não vazio de atributos de condições. Seja S um sistema de informação representado por uma tabela de atributos-valores, para cada atributo $a \in A$ existe um conjunto de valores de atributos V_a

determinado por uma função de classificação $f:U \rightarrow V_a$. A Tabela 2.4 ilustra um sistema de informação genérico que possui linhas rotuladas por objetos de interesse (o_i), colunas rotuladas por atributos de condição (a_j) e os atributos de decisão são definidos na coluna d da tabela com a classificação correspondente $f(o_i, d_i)$, caracterizando as representações bidimensionais dos objetos (PINHEIRO, CAMATTA e REZEK, 2012).

Tabela 2.4 – Representação da tabela genérica de um sistema de informação.

	a_1	...	a_j	...	a_n	d
o_1	$f(o_1, a_1)$		$f(o_1, a_j)$		$f(o_1, a_n)$	$f(o_1, d_1)$
\vdots	\vdots		\vdots		\vdots	\vdots
o_i	$f(o_i, a_1)$...	$f(o_i, a_j)$...	$f(o_i, a_n)$	$f(o_i, d_i)$
\vdots	\vdots		\vdots		\vdots	\vdots
o_m	$f(o_m, a_1)$...	$f(o_m, a_j)$...	$f(o_m, a_n)$	$f(o_m, d_m)$

Fonte: Pinheiro *et al.* (2010).

A teoria dos conjuntos aproximados assume cada objeto do universo de discurso associado a alguma informação. Por exemplo, se os objetos são pacientes que possuem um determinado tipo de doença em um hospital, os atributos ou sintomas da doença descrevem as informações sobre os pacientes. A representação de um sistema de informação (SI) ainda pode caracterizar os pacientes por alguns atributos, tais como: gênero, sexo, pressão sanguínea, temperatura corporal, etc. (PAWLAK e SKOWRON, 2007).

Um SI pode conter redundâncias quando os seus objetos são representados pelas mesmas informações. Para tratar esse problema existe o conceito da relação de não discernimento ou indiscernibilidade. Em cada subconjunto de atributos $B \subseteq A$ está associada uma relação de equivalência denominada relação de indiscernibilidade $IND(B)$ conforme a Expressão (2.52). Se essa relação existe, os objetos o_i e o_j não se distinguem entre si pelos atributos de B (LIMA, SANTOS e PINHEIRO, 2014).

$$IND(B) = \left\{ (o_i, o_j) \in U^2 \mid \forall a_k \in B, f(o_i, a_k) = f(o_j, a_k) \right\} \quad (2.52)$$

O conjunto de todas as classes de equivalência de $IND(B)$ é representado pela notação $U/IND(B)$. Dado um conjunto $O \subseteq U$, pode se definir a aproximação inferior em (2.53), e a aproximação superior, expressa por (2.54).

$$B_*(O) = \{o \in U \mid IND(B) \subseteq O\} \quad (2.53)$$

$$B^*(O) = \{o \in U \mid IND(B) \cap O \neq \emptyset\} \quad (2.54)$$

A aproximação inferior $B_*(O)$ consiste nos objetos que podem ser classificados com certeza como membros de O utilizando informações de B . Por outro lado, a aproximação superior $B^*(O)$ contém os objetos que podem ser classificados apenas como possíveis membros de O com base nas informações de B . Esses dois conjuntos exatos fazem parte da representação de um conjunto aproximado (PAWLAK e SKOWRON, 2007).

A região negativa representa todos os elementos do universo U que não fazem parte de O , isto é, os elementos que não pertencem à aproximação superior de O , sendo expressa por (2.55). A partir da diferença entre (2.54) e (2.53) constitui-se a região de fronteira de O em (2.56). A região de fronteira é caracterizada pela incerteza quanto a pertinência dos elementos, onde não é possível afirmar se os mesmos pertencem a O ou a $\neg O$. Esses conceitos podem ser melhor compreendidos com a Figura 2.9.

$$NEG(O) = U - B^*(O) \quad (2.55)$$

$$RF(O) = B^*(O) - B_*(O) \quad (2.56)$$

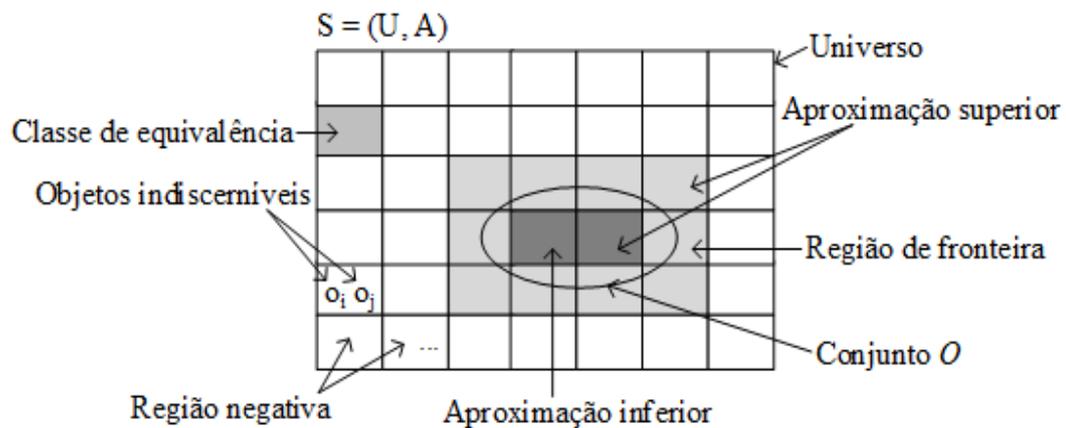


Figura 2.9 – Representação das regiões do conjunto O .

Fonte: Adaptado de Pawlak e Skowron (2007).

Se a região de fronteira for um conjunto vazio, ocorre o mesmo que $B_*(O) = B^*(O)$, isso define um conjunto preciso (*crisp*), caso contrário, existe um conjunto impreciso ou aproximado (*rough*). Em outras palavras, o conjunto é aproximado quando não é possível definir com exatidão os seus elementos. Dessa forma, o conjunto O é representado pelas aproximações $B_*(O)$ e $B^*(O)$, onde todos os elementos do conjunto da aproximação inferior fazem parte do conjunto da aproximação superior.

A qualidade dessas aproximações podem ser medidas em termos dos próprios elementos que a definem. O coeficiente de imprecisão $\alpha_B(O)$ dado por (2.57) expressa a qualidade da aproximação em relação aos elementos do conjunto O (OLIVEIRA, 2013).

$$\alpha_B(O) = \frac{|B_*(O)|}{|B^*(O)|} \quad (2.57)$$

Onde $|B_*(O)|$ e $|B^*(O)|$ denotam a cardinalidade das aproximações inferior e superior para conjuntos não vazios. O valor de $\alpha_B(O)$ deve ficar na faixa $0 \leq \alpha_B(O) \leq 1$. Quanto mais próximo de um, mais preciso é o conjunto O em relação ao conjunto de atributos B . Por outro lado, quanto mais próximo de zero, mais impreciso é O em relação ao conjunto de atributos B .

A matriz de discernibilidade é simétrica $n \times n$, seus elementos são dados por (2.58), onde $i, j = 1, \dots, n$ e $n = |U/IND(B)|$. Os elementos da matriz de discernibilidade estabelecem o conjunto de atributos de B que diferenciam os objetos das classes de equivalência i e j .

$$m_D(i, j) = \{a_k \in B \mid f(o_i, a_k) \neq f(o_j, a_k)\} \quad (2.58)$$

A função de discernibilidade $F(B)$ é uma função booleana dada por (2.59) e (2.60). Dentro de cada célula da matriz de discernibilidade utiliza-se o operador lógico “ \vee ” ou “OR”, e entre as células dessa matriz aplica-se o operador “ \wedge ” ou “AND”. O conjunto formado pelo termo mínimo de $F(B)$ determina os redutos de B . Um reduto representa a parte essencial de uma base de conhecimento, pois define o conjunto mínimo de atributos necessários para fornecer as mesmas propriedades de um sistema de informação original (LIMA, SANTOS e PINHEIRO, 2014).

$$F(B) = \wedge \{ \vee \bar{m}_D(i, j) \} \quad (2.59)$$

$$\bar{m}_D(i, j) = \{ \bar{a}_k \mid a_k \in m_D(i, j) \} \quad (2.60)$$

Uma relação de equivalência pode possuir mais de um reduto, sendo possível obter conjuntos de regras distintos que resultem nas mesmas decisões (TAJIRI, 2009). A determinação do conjunto mínimo de atributos corresponde a menor cardinalidade encontrada entre os redutos. O processamento de redutos da base de conhecimento pode ser tratado com recursos computacionais e a utilização de técnicas de inteligência artificial como os algoritmos genéticos.

A transformação de um reduto em regras de decisão é realizada ao agregar os atributos condicionais com seus respectivos valores e completar a regra com os atributos de decisão. Para um determinado reduto, um exemplo de regra de decisão é expresso em (2.61). Assim, a teoria dos conjuntos aproximados possibilita gerar um conjunto de regras a partir de um sistema de informação, tratando eventuais redundâncias, incertezas ou imprecisões existentes nos dados (PINHEIRO; CAMATTA e REZEK, 2012).

$$\begin{aligned} & \text{IF } a_1 = f(o_1, a_1) \text{ AND } \dots \text{AND } a_k = f(o_m, a_k) \text{ THEN} \\ & d_1 = f(o_1, d_1) \text{ OR } \dots \text{OR } d_i = f(o_i, d_i) \end{aligned} \quad (2.61)$$

É importante destacar que o processamento de conjuntos aproximados lidam com valores nominais. A TCA não aceita atributos numéricos fracionários, pois o conjunto de valores nominais para esses atributos seriam infinitos. Dessa forma, é necessário um processo de discretização que constitui-se no pré-processamento de dados transformando atributos numéricos em valores nominais (OLIVEIRA, 2013).

Exemplo de aplicação da teoria dos conjuntos aproximados:

Para demonstrar os conceitos dos conjuntos aproximados, a Tabela 2.5 ilustra um exemplo apresentado em Oliveira (2013) que tem como objetivo determinar a menor combinação de exames clínicos necessários para classificar os diagnósticos em um experimento. Os exemplos de exames clínicos (Exame_x) podem classificar três tipos de diagnósticos (D1, D2 e D3). Os resultados dos exames são classificados como positivo (P), negativo (N) e pelas faixas de valores 1, 2, 3 e 4, ou < 12, 12-23, 24-32 e > 32.

Tabela 2.5 – Sistema de informação associado ao exemplo.

	Exame ₁	Exame ₂	Exame ₃	Exame ₄	Diagnóstico
o₁	1	P	2	> 32	D1
o₂	2	P	3	12-23	D2
o₃	4	N	3	24-32	D3
o₄	1	N	2	< 12	D1
o₅	3	N	1	24-32	D2
o₆	1	N	4	< 12	D1

O trabalho com os conjuntos aproximados deve ser realizado a partir de valores discretos e não contínuos. Os valores do “Exame₄” podem ser representados como: F1 para valores < 12, F2 para a faixa 12-23, F3 para 24-32 e F4 para > 32. Algumas informações podem ser obtidas através do sistema de informação:

Conjunto de observações: $U = \{o_1, o_2, o_3, o_4, o_5, o_6\}$.

Conjuntos de atributos de condição: $A = \{\text{Exame}_1, \text{Exame}_2, \text{Exame}_3, \text{Exame}_4\}$.

$V_{\text{Exame}_1} = V_{\text{Exame}_3} = \{1, 2, 3, 4\}$; $V_{\text{Exame}_4} = \{F1, F2, F3, F4\}$; $V_{\text{diag}} = \{D1, D2, D3\}$.

Da Equação (2.52), tem-se:

$IND(\{\text{Exame}_2\}) = \{\{o_1, o_2\}, \{o_3, o_4\}, \{o_3, o_5\}, \{o_3, o_6\}, \{o_4, o_5\}, \{o_4, o_6\}, \{o_5, o_6\}\}$.

As classes de equivalência: $U/IND(\{\text{Exame}_2\}) = \{\{o_1, o_2\}, \{o_3, o_4, o_5, o_6\}\}$.

Seja $B = \{\text{Exame}_1, \text{Exame}_3\}$, pode-se definir:

$$IND(B) = \{\{o_1, o_4\}\};$$

$$U/IND(B) = \{\{o_1, o_4\}, \{o_2\}, \{o_3\}, \{o_5\}, \{o_6\}\}.$$

Das Equações (2.53), (2.54), (2.56) e (2.57), tem-se:

Para $O = \{o_1, o_2, o_3\}$, vem:

$$B^*(O) = \{\{o_2, o_3\}\}; B^*(O) = \{o_1, o_2, o_3, o_4\}; RF(O) = \{o_1, o_4\};$$

$$\alpha_B(O) = \frac{|\{o_2, o_3\}|}{|\{o_1, o_2, o_3, o_4\}|} = \frac{2}{4} = 0,5$$

A Tabela 2.6 ilustra a matriz de discernibilidade definida de acordo a Equação (2.58) para o sistema de informação apresentado.

Tabela 2.6 – Matriz de discernibilidade do exemplo.

	o1	o2	o3	o4	o5	o6
o1	-					
o2	Ex1, Ex3, Ex4	-				
o3	Ex1, Ex2, Ex3, Ex4	Ex1, Ex2, Ex4	-			
o4	Ex2, Ex4	Ex1, Ex2, Ex3, Ex4	Ex1, Ex3, Ex4	-		
o5	Ex1, Ex2, Ex3, Ex4	Ex1, Ex2, Ex3, Ex4	Ex1, Ex3	Ex1, Ex3, Ex4	-	
o6	Ex2, Ex3, Ex4	Ex1, Ex2, Ex3, Ex4	Ex1, Ex3, Ex4	Ex3	Ex1, Ex3, Ex4	-

A função de discernibilidade associada às informações da matriz correspondente é definida por:

$$F(B) = (Ex1 \vee Ex3 \vee Ex4) \wedge (Ex1 \vee Ex2 \vee Ex3 \vee Ex4) \wedge (Ex2 \vee Ex4) \\ \wedge (Ex1 \vee Ex2 \vee Ex3 \vee Ex4) \wedge (Ex2 \vee Ex3 \vee Ex4) \wedge \dots \wedge (Ex1 \vee Ex3 \vee Ex4)$$

Simplificando e colocando a função na forma da “soma do produto”, tem-se:

$$F(B) = (Ex1) \vee (Ex2 \wedge Ex3) \vee (Ex3 \wedge Ex4)$$

Dessa forma, são obtidos os redutos $\{\{\text{Exame}_1\}, \{\text{Exame}_2, \text{Exame}_3\}, \{\text{Exame}_3, \text{Exame}_4\}\}$. É conveniente escolher o reduto com o menor número de atributos (menor cardinalidade), ou seja, o reduto $\{\text{Exame}_1\}$. O exame do tipo 1 é suficiente para classificar o diagnóstico, resultando nas regras de decisão:

IF $\text{Exame}_1 = 1$ THEN Diagnóstico = D1;

IF $\text{Exame}_1 = 2$ THEN Diagnóstico = D2;

IF $\text{Exame}_1 = 4$ THEN Diagnóstico = D3;

IF $\text{Exame}_1 = 3$ THEN Diagnóstico = D2.

2.5.2 Modelos baseados em regras

Os modelos baseados em regras são uma estratégia importante na modelagem e controle de sistemas complexos que utilizam técnicas de inteligência artificial. Os controladores aproximados são aproximadores de funções de controle sob a forma de regras e podem ser aplicados em processos com características lineares e não lineares (MACHADO e PINHEIRO, 2013). Essas aplicações envolvem atributos numéricos que devem ser representados na forma adequada conforme o sistema de informação da Tabela 2.7.

Tabela 2.7 – Representação tabular numérica de um SI.

x_1	x_2	x_3	...	x_N	y
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$...	$x_N^{(1)}$	$y^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$...	$x_N^{(2)}$	$y^{(2)}$
...
$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$...	$x_N^{(k)}$	$y^{(k)}$
...
$x_1^{(m)}$	$x_2^{(m)}$	$x_3^{(m)}$...	$x_N^{(m)}$	$y^{(m)}$
...
$x_1^{(v)}$	$x_2^{(v)}$	$x_3^{(v)}$...	$x_N^{(v)}$	$y^{(v)}$

Fonte: Pinheiro *et al.* (2010).

Os dados ($x_i^{(k)}$) dos atributos de condições (x_i) estão relacionados a funções com valores ($y^{(k)}$) do atributo de decisão (y). Através deste SI são geradas as sentenças definidas em (2.62).

Considerando os valores intermediários nos dados estabelecidos na Tabela 2.7, ou seja, $x_1^{(k)} \leq x_1 \leq x_1^{(m)}$, $x_2^{(k)} \leq x_2 \leq x_2^{(m)}$, ..., $x_N^{(k)} \leq x_N \leq x_N^{(m)}$ (OLIVEIRA, 2013). As regras podem ser redefinidas na forma genérica em (2.63) ou pela forma simplificada (2.64), onde são utilizados nesses modelos apenas conjuntos convencionais (*crisp*) definido em intervalos expressos por $\alpha^{(g)} = [x_1^{(k)}, x_1^{(m)}]$, $\beta^{(g)} = [x_2^{(k)}, x_2^{(m)}]$, $\gamma^{(g)} = [x_N^{(k)}, x_N^{(m)}]$, $\sigma^{(g)} = [y^{(k)}, y^{(m)}]$, sendo $y^{(k)} < y^{(m)}$.

$$\begin{aligned}
 r_1 &: IF x_1 = x_1^{(1)} AND x_2 = x_2^{(1)} AND \dots AND x_n = x_n^{(1)} THEN y = y^{(1)} \\
 r_2 &: IF x_1 = x_1^{(2)} AND x_2 = x_2^{(2)} AND \dots AND x_n = x_n^{(2)} THEN y = y^{(2)} \\
 &\dots \\
 r_k &: IF x_1 = x_1^{(k)} AND x_2 = x_2^{(k)} AND \dots AND x_n = x_n^{(k)} THEN y = y^{(k)} \\
 &\dots \\
 r_m &: IF x_1 = x_1^{(m)} AND x_2 = x_2^{(m)} AND \dots AND x_n = x_n^{(m)} THEN y = y^{(m)}
 \end{aligned} \tag{2.62}$$

$$\begin{aligned}
r_i : & \text{IF } x_1^{(k)} \leq x_1 \leq x_1^{(m)} \text{ AND } x_2^{(k)} \leq x_2 \leq x_2^{(m)} \text{ AND } \dots \\
& \text{AND } x_N^{(k)} \leq x_N \leq x_N^{(m)} \\
& \text{THEN } \min\{y^{(k)}, \dots, y^{(m)}\} \leq y \leq \max\{y^{(k)}, \dots, y^{(m)}\}
\end{aligned} \tag{2.63}$$

$$\begin{aligned}
r_i : & \text{IF } x_1 = \alpha^{(g)} \text{ AND } x_2 = \beta^{(g)} \text{ AND } \dots \text{ AND } x_N = \gamma^{(g)} \\
& \text{THEN } y = \sigma^{(g)}
\end{aligned} \tag{2.64}$$

Para estimar o valor de saída y emprega-se uma expressão de interpolação linear (2.65), capaz de interpolar valores intermediários entre as aproximações inferiores e superiores definidas como $y^{(j)}$, onde $j = k, m$. O índice i refere-se ao número de regras, e a quantidade de atributos condicionais é representada por $n = 1, 2, \dots, N$.

$$y_i(x_n, x_n^j, y_i^j) = y_i^{(k)} + \frac{y_i^{(m)} - y_i^{(k)}}{N} \left(\sum_{n=1}^N \frac{x_n - x_{i,n}^{(k)}}{x_{i,n}^{(m)} - x_{i,n}^{(k)}} \right) \tag{2.65}$$

2.5.3 Ferramenta Computacional

A utilização dos conjuntos aproximados na análise de dados e estudos de relações de equivalência pode se tornar um procedimento extenuante quando realizado manualmente, principalmente para uma grande quantidade de dados. Por se tratar de um método baseado em fundamentos matemáticos bem definidos é possível a criação de algoritmos computacionais para essa aplicação (TAJIRI, 2009). Existem ferramentas computacionais de acesso gratuito desenvolvidas especificamente para o processamento de conjuntos aproximados. Neste projeto é utilizado o Rosetta (OHRN e KOMOROWSKI, 1997). Esta ferramenta auxilia no processamento de dados em sistemas genéricos de informação, permite obter as classes e os redutos, sendo por fim gerado um conjunto de regras de decisão.

2.5.4 Conversão de modelos *Fuzzy* em modelos Aproximados

A metodologia de conversão de modelos *fuzzy* em modelos aproximados é proposta por Guaracy, Pinheiro e Ferreira (2014). Inicialmente, para um modelo *fuzzy* com “ n ” variáveis de entrada e cada variável com “ s ” conjuntos difusos, identifica-se as regiões $P_n^{(m)}$ ($m = 1, \dots, s$) correspondentes aos intervalos em que a função de pertinência do conjunto difuso “ m ” assume

valores maiores que zero. Essas regiões são representadas por conjuntos poliedrais dados por (2.66), em que $H_n^{(m)} \in \mathfrak{R}^{2 \times 1}$ e $S_n^{(m)} \in \mathfrak{R}^2$. A Figura 2.10 apresenta três regiões $P_n^{(m)}$ para uma variável (x_1) com três conjuntos *fuzzy*.

$$P_n^{(m)} = \{x_n \in \mathfrak{R} \mid H_n^{(m)} x_n \leq S_n^{(m)}\} \quad (2.66)$$

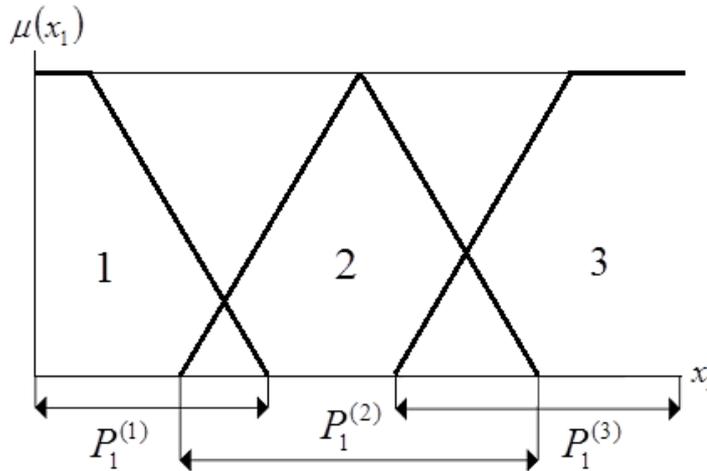


Figura 2.10 – Identificação das regiões $P_n^{(m)}$

Fonte: Herceg, Kvasnica e Fikar (2007).

Novas regiões $D_n^{(i)}$ ($i = 1, \dots, p$) são definidas nas intersecções dos conjuntos $P_n^{(m)}$, conforme expresso em (2.67). Desse modo, as regiões redundantes são desconsideradas, como aquelas em que duas regiões se sobrepõem no mesmo intervalo.

$$D_n^{(i)} = P_n^{(m)} \cap P_n^{(m+1)} \quad (2.67)$$

Os conjuntos restantes $D_n^{(i)}$ ($i = p+1, \dots, q$) são determinados por uma operação de diferença de conjuntos com a equação (2.68). Os conjuntos *crisp* associados a uma determinada variável de entrada x_i são indicados por D_n , conforme em (2.69).

$$\bigcup_{i=p+1}^q D_n^{(i)} = \left\{ x_n \in \mathfrak{R} \mid x_n \in \bigcup_{m=1}^k P_n^{(m)}, x_n \notin \bigcup_{i=1}^p D_n^{(i)} \right\} \quad (2.68)$$

$$D_n = \{D_n^{(1)}, D_n^{(2)}, \dots, D_n^{(q)}\} \quad (2.69)$$

A Figura 2.11 representa o mapeamento de três conjuntos *fuzzy* em cinco conjuntos *crisp*, sendo inclusas as regiões $D_n^{(1)}$ e $D_n^{(2)}$, respectivamente. São utilizadas funções de pertinência do tipo trapezoidal e triangular. No entanto, este mapeamento também é válido para

funções de pertinência do tipo gaussiana conforme pode ser observado em Guaracy, Pinheiro e Ferreira (2014).

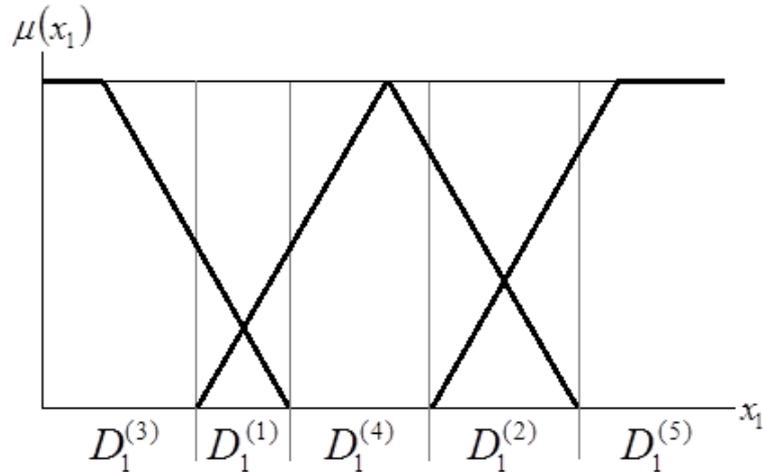


Figura 2.11 – Mapeamento de conjuntos *fuzzy* em conjuntos *crisp*.
Fonte: Herceg, Kvasnica e Fikar (2007).

Os conjuntos $D_n^{(i)}$ são redefinidos de tal forma para que não exista sobreposição dos limites comuns entre duas regiões. A notação em forma de intervalos pode ser expressa pelas Equações (2.70) e (2.71).

$$D_n^{(i)} = \{x_n \in \mathfrak{R} \mid \underline{x}_n^{(i)} \leq x_n < \bar{x}_n^{(i)}\} = [\underline{x}_n^{(i)}, \bar{x}_n^{(i)}) \quad (2.70)$$

$$i = 1, \dots, q-1$$

$$D_n^{(q)} = \{x_n \in \mathfrak{R} \mid \underline{x}_n^{(q)} \leq x_n \leq \bar{x}_n^{(q)}\} = [\underline{x}_n^{(q)}, \bar{x}_n^{(q)}] \quad (2.71)$$

Realizada a decomposição de conjuntos *fuzzy* em conjuntos *crisp*, um sistema de informação $I = (U, A)$ em que o conjunto de atributos dados por $A = \{x_1, x_2, \dots, x_N\}$, e os objetos do universo U são definidos a partir dos produtos cartesianos entre os conjuntos D_n conforme em (2.72). Assim é possível gerar uma estrutura de regras que representem o controlador proposto.

$$U = \{(((D_1 \times D_2) \times D_3) \times \dots) \times D_N\} \quad (2.72)$$

As funções de pertinência do modelo *fuzzy* podem influenciar na quantidade de regras do controlador aproximado. Quanto maior a quantidade de conjuntos, maior será o número de intervalos representando cada termo e melhor poderá ser a exatidão na modelagem. Para o modelo aproximado descrever o comportamento desse sistema, o mapeamento de conjuntos *fuzzy* em conjuntos *crisp* deve possuir mais regiões de decomposição, resultando numa maior quantidade de regras.

Um exemplo desse método de conversão pode ser observado na Tabela 2.8, obtida através do mapeamento das funções de pertinência da Figura 2.12 que possui $x_1 \in [0, 1]$. Esse processo consiste na delimitação das regiões de conjuntos *fuzzy* em conjuntos *crisp*, tornando possível a determinação de um sistema de informação associado com o modelo estabelecido. Esse procedimento de conversão é aplicado no sistema de controle do capítulo 3.

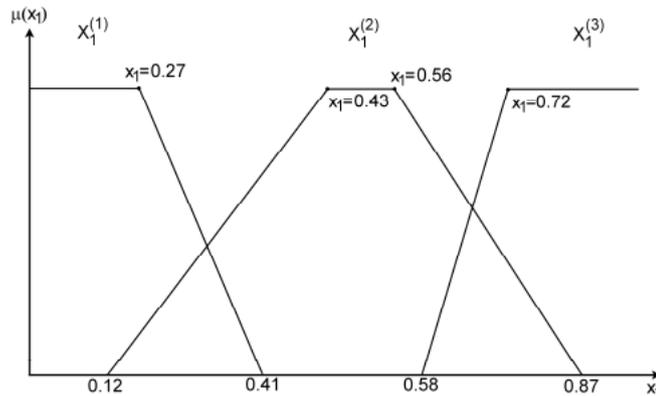


Figura 2.12 – Exemplo do mapeamento de conjuntos *fuzzy* em conjuntos *crisp*.
Fonte: Guaracy, Pinheiro e Ferreira (2014).

Tabela 2.8 – Sistema de Informação associado com o modelo estabelecido.

o_i	x_1
o_1	[0; 0,12)
o_2	[0,12; 0,41)
o_3	[0,41; 0,58)
o_4	[0,58; 0,87)
o_5	[0,87; 1]

Fonte: Guaracy, Pinheiro e Ferreira (2014).

2.5.5 Comparação com Sistemas Fuzzy

Os conjuntos aproximados possuem algumas semelhanças em relação aos conjuntos difusos. Ambos os conceitos tem o conhecimento associado a uma base de regras e são capazes de lidar com as incertezas. É importante evidenciar algumas particularidades existentes em cada teoria, conforme estudos realizados em (DUBOIS e PRADE, 1990; PAWLAK, 1991).

A teoria dos conjuntos difusos e aproximados representam duas abordagens diferentes na forma como tratam as incertezas do conhecimento. Na lógica *fuzzy* a incerteza é modelada de forma gradual com a indefinição dos limites e transições, ou seja, não existe uma fronteira ou linha de transição bem definida entre dois conjuntos adjacentes. Enquanto a teoria dos conjuntos aproximados se baseiam na granulação ou classificação aproximada do

conhecimento, expressa pelas aproximações superiores e inferiores dos conjuntos (TAJIRI, 2009).

Em relação ao desenvolvimento dos modelos considerados é oportuno destacar:

Conjuntos difusos: os modelos são baseados na experiência do especialista e suas relações são estabelecidas através de conectivos lógicos. O controlador *fuzzy* é desenvolvido com base no conhecimento sobre o processo sem a necessidade de um procedimento matemático para a definição das regras e parâmetros de controle. Também podem ser utilizados métodos quantitativos que resultam em uma menor intervenção do especialista. Nas duas situações, os sistemas *fuzzy* são utilizados para definir novos modelos de controle, sendo então analisados durante o seu desenvolvimento.

Conjuntos aproximados: os modelos podem ser desenvolvidos através de algoritmos de análise matemática baseados na teoria dos conjuntos para o tratamento de dados e a obtenção das regras aproximadas. Isso permite afirmar que os modelos construídos com base na TCA necessitam apenas dos dados referentes ao sistema em questão e não dependem de informações adicionais como o grau de pertinência utilizado nos modelos *fuzzy*.

De uma forma geral, os fatores que devem ser considerados no desempenho dos controladores *fuzzy* são a quantidade de entradas e saídas, o número e as formas das funções de pertinência, a técnica de inferência empregada e o método de defuzificação. Os controladores aproximados podem ter fatores mais restritos para impactar no desempenho, tais como a quantidade de variáveis de entradas e saídas, e o número de regras.

A interface de fuzificação utiliza um fator de escala para condicionar os valores de entrada em um universo de discurso normalizado e fuzifica os valores, convertendo números em conjuntos *fuzzy*. Após o procedimento de inferência e obtenção dos valores *crisp* com a defuzificação, novamente realiza-se o escalamento, para converter os valores normalizados em valores dos universos de discurso reais das variáveis (GOMIDE e GUDWIN, 1994). Todas essas etapas proporcionam ao controlador *fuzzy* uma melhor estimativa na modelagem de funções. É necessário ressaltar que a principal vantagem do modelo aproximado é o menor tempo de computação, devido à ausência dos procedimentos de fuzificação e defuzificação citados. Estas características podem ser melhor observadas no capítulo 4, onde se encontram os resultados experimentais.

Pinheiro (2000) descreveu uma forma de justificar as regras usadas em sistemas de controle *fuzzy* através da análise da trajetória do sistema no tempo. A partir das informações do

erro “E” e da variação do erro do sistema “CE”, as regras são inferidas para adequar as respostas do controlador *fuzzy* “U” de acordo a especificação do projeto. As regras linguísticas são denominadas como N (negativo), Z (zero) e P (positivo), conforme definido em (2.73).

$$\begin{aligned}
 &IF E = P \text{ AND } CE = Z \text{ THEN } U = P \text{ (regras : } r_1 \text{ e } r_5) \\
 &IF E = Z \text{ AND } CE = N \text{ THEN } U = N \text{ (regra : } r_2) \\
 &IF E = N \text{ AND } CE = Z \text{ THEN } U = N \text{ (regra : } r_3) \\
 &IF E = Z \text{ AND } CE = P \text{ THEN } U = P \text{ (regra : } r_4)
 \end{aligned}
 \tag{2.73}$$

Os conjuntos aproximados também são modelos baseados em regras aplicados na modelagem e sistemas de controle. Tanto os controladores *fuzzy* quanto os aproximados ativam as regras de forma semelhante a Figura 2.13.

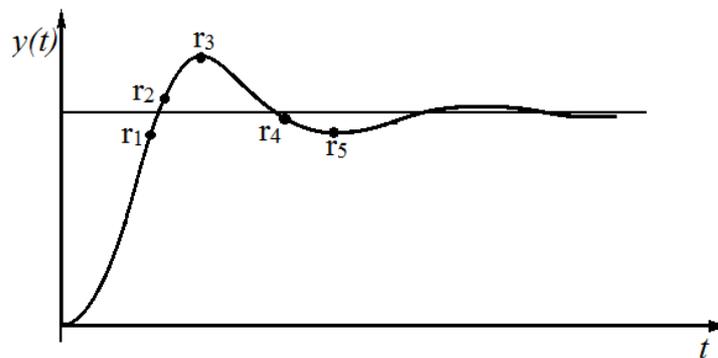


Figura 2.13 – Ilustração das regras ativadas na resposta do sistema de controle.
Fonte: adaptado de Pinheiro (2000).

A regra r_5 representa a situação na qual a saída do sistema está abaixo da referência e a saída do controlador deve ser aumentada. A regra r_3 caracteriza uma situação oposta, onde a saída do sistema é maior que a referência. Neste caso, a saída do controlador deve ser reduzida. Assim, o conjunto de regras atuam em determinadas regiões de interesse a fim de que a saída do sistema siga a referência desejada.

É pertinente acentuar que as duas teorias não se competem, já que são capazes de desenvolver funções específicas distintas. Nesta dissertação, por exemplo, o modelo de controle definido pela lógica *fuzzy* foi fundamental para fornecer o conhecimento necessário no desenvolvimento do controlador aproximado.

3 Conversão entre os Modelos Baseados em Regras

Esse capítulo apresenta a aplicação da abordagem de conversão do modelo de um controlador *fuzzy* em um controlador aproximado (*rough*). As simulações computacionais são realizadas no controle de posição de um motor CC com o objetivo de avaliar o desempenho de ambos os compensadores.

3.1 Caracterização do problema

Conforme discutido anteriormente no capítulo 2, o controlador *fuzzy* possui um comportamento adequado no controle de processos não lineares ou com variações de parâmetros no tempo. No entanto, um dos principais problemas desse tipo de controle é o tempo de computação que é maior em relação aos controladores convencionais, devido às operações complexas como fuzificação e defuzificação. Com o objetivo de reduzir o tempo de processamento é proposto o controlador aproximado que possui uma estrutura computacional mais simples.

O procedimento envolve a concepção de três tipos de controladores: o primeiro é um controlador PID com o método de Ziegler-Nichols no ajuste dos ganhos, os quais são usados no projeto do controlador *fuzzy*, a fim de obter um melhor desempenho. Este controlador serve de modelo para o compensador baseado em regras, denominado nesta dissertação como controlador aproximado (*rough*). A estratégia utilizada no terceiro controlador defini as regiões de decomposição de conjuntos *fuzzy* em conjuntos *crisp*, ou seja, trata-se de uma metodologia de conversão de modelos *fuzzy* em modelos aproximados.

O projeto de controladores devem satisfazer a determinadas especificações que podem ser avaliadas em termos da resposta transitória à entrada em degrau. Os critérios de desempenho para o sistema com o controlador foram especificados como:

- Tempo de subida (t_r) ≤ 1 s;
- Tempo de acomodação (t_s) ≤ 2 s;
- Máximo *overshoot* (M_P) ≤ 10 %;
- Erro em regime permanente (e) ≤ 1 %.

Para os estudos realizados neste capítulo utilizou-se o modelo de um motor de corrente contínua com os parâmetros obtidos em Rathore *et al.* (2015), conforme a Tabela 3.1.

Tabela 3.1 – Parâmetros do motor CC.

Parâmetros	Valor	Unidade
Resistência de Armadura R_a	1	Ω
Indutância de Armadura L_a	0,5	H
Constante Força da Eletromotriz K_b	0,01	Vs/rad
Constante de Torque K_T	0,01	Nm/A
Coefficiente de Amortecimento B_m	0,1	Nms/rad
Momento de Inércia J_m	0,01	Kg-m ² /rad

O controle dos motores CC são feitos principalmente pela tensão aplicada na armadura ou no campo (ZADEH, YAZDIAN e MOHAMADIAN, 2006). Neste projeto, o controle é realizado pela armadura. A modelagem matemática destes sistemas podem ser realizadas através de funções de transferência. A substituição dos parâmetros da Tabela 3.1 na Função de Transferência dada em (2.12) define a planta resultante (3.1). Dessa forma, simulações computacionais foram realizadas no Matlab/Simulink[®] para obter as respostas do controle de posição do motor, considerando entradas do tipo degrau e os controladores desenvolvidos.

$$\frac{\theta(s)}{V_a(s)} = \frac{0,01}{0,005s^3 + 0,06s^2 + 0,1001s} \quad (3.1)$$

3.2 Projeto do compensador PID

A sintonia do compensador pode ser realizada através de várias técnicas que permitem determinar os seus parâmetros, estabelecendo as especificações do regime transitório e do regime permanente em malha fechada para a obtenção do desempenho desejado (OGATA, 2003). Nesse capítulo o método de sintonia de Ziegler-Nichols é utilizado no projeto do controlador PID, os parâmetros obtidos auxiliam na determinação dos ganhos de escala do controlador *Fuzzy PD*.

O primeiro método de Ziegler-Nichols também conhecido por curva de reação do processo é utilizado se a planta não possuir integradores nem pólos complexos (OGATA, 2003). Como a planta envolvida possui um integrador, este método não é adequado conforme ilustra a Figura 3.1, em que a resposta não apresenta um aspecto de “S” quando aplicada uma entrada degrau. Esta é a principal desvantagem do método que envolve a curva de reação, pois grande parte dos sistemas são mais complexos comparado aos sistemas de primeira ordem com atraso.

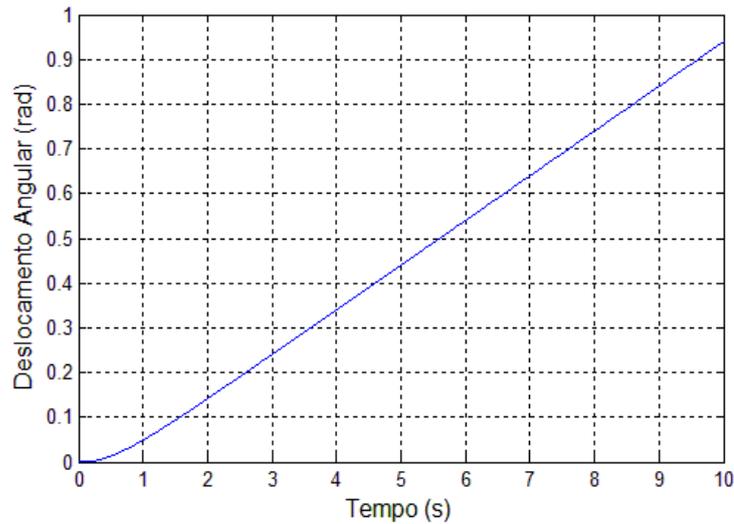


Figura 3.1 – Resposta do sistema a entrada tipo degrau sem o controlador.

De acordo a Figura 3.1, a resposta do sistema apresenta um aumento linear na posição e não acompanha a entrada degrau, sendo então necessário um compensador na malha de controle.

A outra forma de ajuste é o método da sensibilidade crítica que permite adotar $T_I = \infty$, $T_D = 0$ e em seguida alterar o ganho do controlador proporcional para colocar o sistema no modo de oscilação e realizar o procedimento de ajuste, mas nem sempre é possível que a planta entre em oscilação. Nesse contexto, a partir da planta é possível definir qual dos dois métodos de Ziegler-Nichols deve ser utilizado, nesta dissertação utilizou-se o método da sensibilidade crítica.

O modelo matemático da planta é conhecido e o valor do ganho que torna o sistema marginalmente estável, também chamado de ganho crítico K_{cr} , pode ser obtido pelo critério de Routh-Hurwitz que é um método algébrico e informa sobre a estabilidade absoluta do sistema. Conforme o sistema apresentado na Figura 2.5, a função de transferência de malha fechada é dada por (3.2). E em (3.3) é definida a Equação característica em malha fechada.

$$\frac{\theta(s)}{V(s)} = \frac{0,01K_p}{0,005s^3 + 0,06s^2 + 0,1001s + 0,01K_p} \quad (3.2)$$

$$0,005s^3 + 0,06s^2 + 0,1001s + 0,01K_p = 0 \quad (3.3)$$

Após definir a Equação característica aplica-se o critério de Routh de acordo a Tabela 3.2.

Tabela 3.2 – Critério de Estabilidade de Routh.

s^3	0,005	0,1001
s^2	0,06	$0,01K_p$
s^1	$\frac{6,006 \cdot 10^3 - 5 \cdot 10^{-5} K_p}{0,06}$	-
s^0	$0,01K_p$	-

A partir da 3ª linha e 2ª coluna da Tabela 3.2, é determinado o ganho $K_P = K_{cr} = 120,12$. A Figura 3.2 mostra a resposta da malha de controle para o ganho crítico calculado e apresenta o valor da oscilação sustentada de período P_{cr} igual a 1,4 segundos.

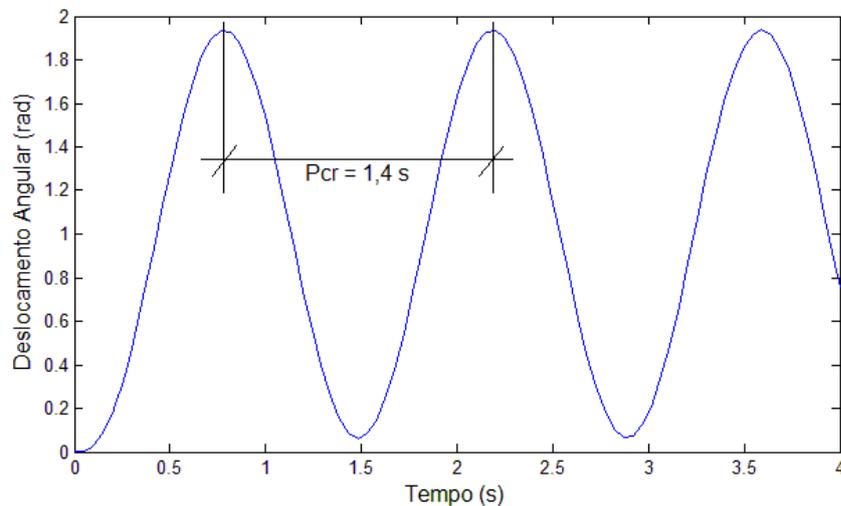


Figura 3.2 – Resposta da malha de controle no modo de oscilação.

A partir do ganho crítico e do período de oscilação da malha de controle é possível determinar os valores dos ganhos e das constantes de tempo do controlador com base na Tabela 2.2, referente as regras do segundo método de Ziegler-Nichols. Assim, foram determinados os valores $K_P = 72,072$, $T_I = 0,7$ s e $T_D = 1,175$ s. Uma vez que $K_I = K_P/T_I$ e $K_D = K_P \cdot T_D$, são calculados os valores dos ganhos resultantes $K_P = 72,072$, $K_I = 102,960$ e $K_D = 12,613$. Com a abordagem de Ziegler-Nichols não há como estimar previamente o *overshoot* e o tempo de acomodação. Esses parâmetros podem ser obtidos na Figura 3.3 a partir da resposta ao degrau do sistema com o controlador PID clássico.

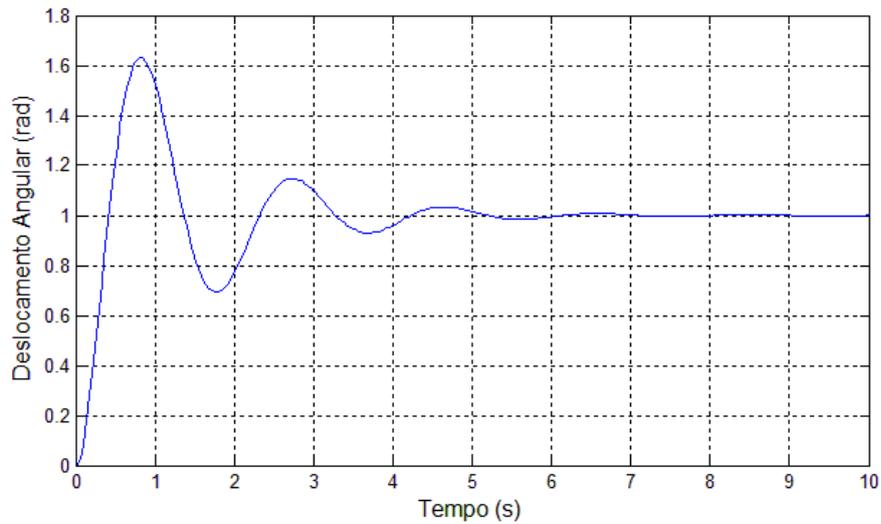


Figura 3.3 – Resposta ao degrau com o segundo método de Ziegler-Nichols.

Geralmente o método de Ziegler-Nichols fornece uma pequena margem de fase. Portanto, produz um grande *overshoot* e tempo de acomodação (SUDHA e ANITA, 2012). Nesse caso, a Figura 3.3 mostra um *overshoot* de 63% e o tempo de acomodação de aproximadamente 5 segundos.

Os critérios de desempenho não foram atendidos com o método de Ziegler-Nichols. Além disso, os distúrbios externos e as não linearidades podem comprometer a estabilidade do sistema (SUDHA e ANITA, 2012). Os controladores convencionais não possuem um bom desempenho em sistemas não lineares, complexos e que não tem modelos exatos (AMLASHI, 2012). Uma alternativa encontra-se na teoria dos conjuntos difusos proposta por Zadeh (1965), que proporciona o método de controle *fuzzy* para lidar com as incertezas do sistema.

3.3 Projeto do compensador *Fuzzy* PD

Dentre as variantes do controlador PID desenvolvidos através da Lógica *Fuzzy*, tem-se a estrutura do controlador *Fuzzy* PD utilizada no desenvolvimento de um sistema de controle difuso junto ao modelo de um motor de corrente contínua, como mostra a Figura 3.4. O sistema de controle *fuzzy* admite como entrada o erro e a variação do erro. O sinal de saída do processo realiza a retroalimentação do sistema.

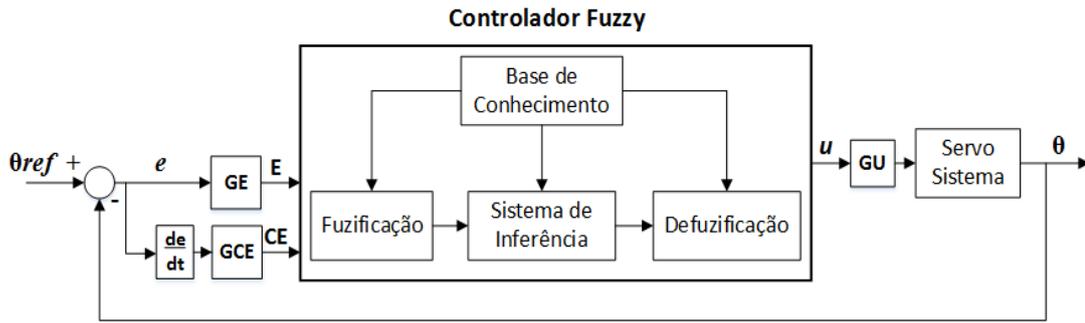


Figura 3.4 – Estrutura do controlador *Fuzzy*.
Fonte: adaptado de Dursun e Durdu, 2016.

Para o controle de posição convém observar que a diferença entre a posição de referência $\theta_{ref}(t)$ e a posição atual $\theta(t)$, determina o erro do sistema $e(t)$ expresso em (3.4).

$$e(t) = \theta_{ref}(t) - \theta(t) \quad (3.4)$$

A partir das expressões $GU = K_p/GE$ e $GCE = GE.T_D$, definidas no capítulo 2 em (2.50) e (2.51) é possível obter os ganhos do controlador *Fuzzy* PD. Estes ganhos se baseiam em valores conhecidos do ganho proporcional e tempo derivativo provenientes do método de Ziegler-Nichols aplicado no controlador PID convencional. Assumindo inicialmente que o valor de GE é igual a 0,5. Os ganhos resultantes são $GE = 0,5$, $GCE = 0,0875$ e $GU = 144,144$.

As funções de pertinência simétricas e igualmente espaçadas são aplicadas em muitos trabalhos. Em Lima, Santos e Pinheiro (2014), esses tipos de funções foram utilizadas na estratégia de controle de um sistema de suspensão veicular, tornando possível suavizar eventuais discontinuidades na força de atuação, reduzindo os problemas relacionados a solavancos e ruídos indesejáveis. Nesta dissertação, as funções de pertinência triangulares e trapezoidais foram definidas por facilitarem o procedimento de ajuste, sendo a maior parte simétricas e igualmente espaçadas, conforme as Figuras 3.5, 3.6 e 3.7, respectivamente.

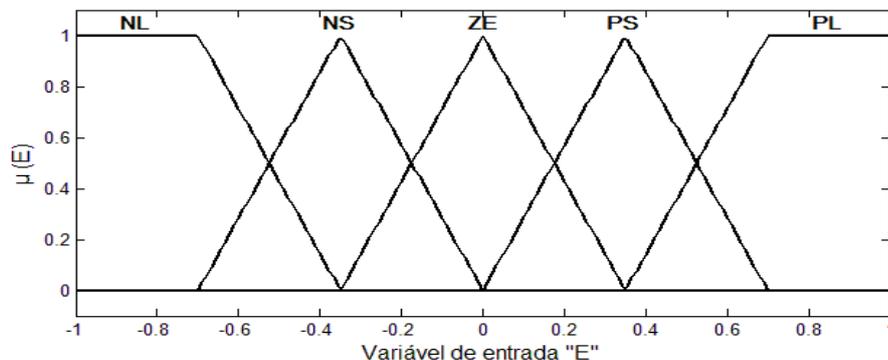


Figura 3.5 – Funções de pertinência para a entrada E.

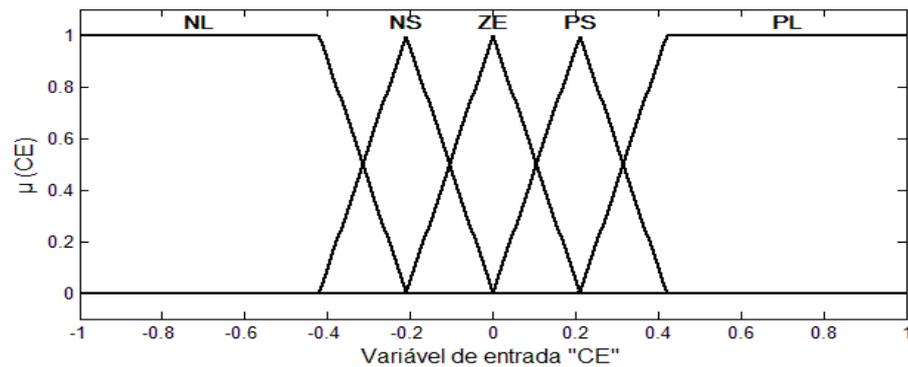


Figura 3.6 – Funções de pertinência para a entrada CE.

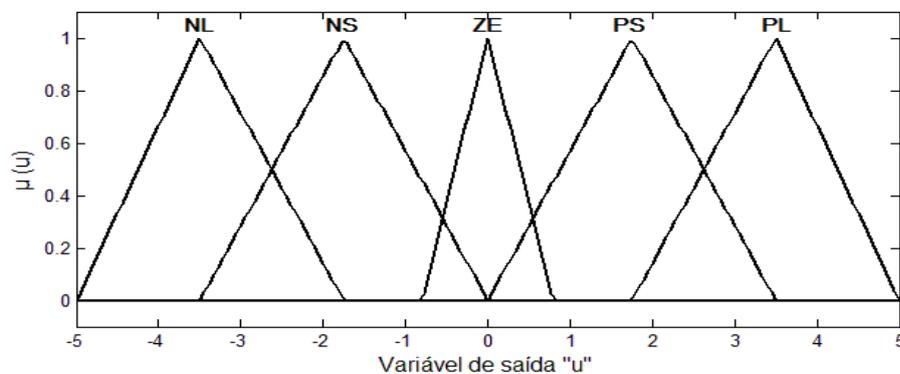


Figura 3.7 – Funções de pertinência para a saída u.

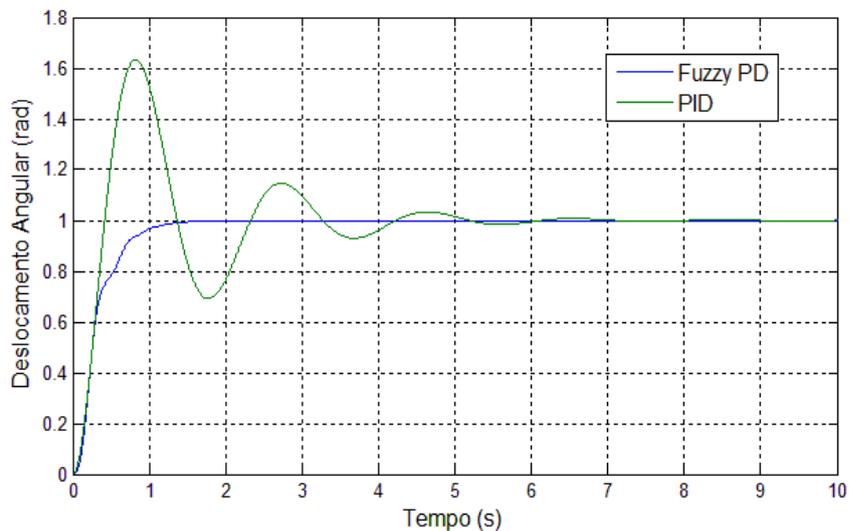
O sistema possui duas variáveis de entrada: erro “E” e variação do erro “CE”, estas variáveis tem o universo de discurso de intervalo [-1; 1]. Ao comparar as funções triangulares das Figuras 3.5 e 3.6, percebe-se que os limites da variação do erro são mais reduzidos em relação aos valores absolutos do erro, esse ajuste foi realizado por tentativa e erro. A variável de saída representa a tensão de controle com funções de pertinências no intervalo de [-5; 5] e corresponde a uma faixa de tensão de -5 V a 5 V, onde os valores negativos são usados para o eixo do motor girar em um sentido e os valores positivos fazem o eixo girar no sentido contrário. Este procedimento é semelhante ao aplicado na parte experimental no capítulo 4. As funções de pertinência constituí-se de cinco termos linguísticos estabelecidos como NL (negativo grande), NS (negativo pequeno), ZE (zero), PS (positivo pequeno) e PL (positivo grande). É usado o método centróide para a defuzificação, e o tipo de inferência Mamdani.

O projeto do controlador requer um conjunto de regras de controle baseadas em suas entradas. O número de regras para o controlador é obtido através da multiplicação do número de termos difusos por eles mesmo. Como o sistema possui duas variáveis de entrada, cada uma com cinco termos linguísticos, existem então 25 combinações possíveis. E o sistema é composto por 25 regras de controle que estão resumidas na Tabela 3.3.

Tabela 3.3 – Base de regras do controlador *Fuzzy* PD usado na simulação.

u		CE				
		NL	NS	ZE	PS	PL
E	NL	NL	NL	NS	NS	ZE
	NS	NL	NS	NS	ZE	PS
	ZE	NS	NS	ZE	PS	PS
	PS	NS	ZE	PS	PS	PL
	PL	ZE	PS	PS	PL	PL

De acordo o resultado obtido com controlador clássico, buscou-se definir um controlador inteligente para reduzir o *overshoot* e o tempo de acomodação. A robustez a incertezas paramétricas e a boa capacidade de mapeamento de processos não lineares contribuem para o melhor desempenho do controlador *fuzzy* em relação ao controlador PID convencional, como mostra a Figura 3.8. Observa-se que o *overshoot* foi eliminado e o tempo de acomodação reduziu de 5 s para 1,16 s. O compensador *fuzzy* apresentou respostas adequadas para o sistema. Desse modo, este é o modelo utilizado no processo de conversão para o modelo aproximado.

Figura 3.8 – Comparação da resposta ao degrau com o controlador PID e *Fuzzy* PD.

3.4 Obtenção do compensador Aproximado

A partir do controlador *Fuzzy* PD aplicado no controle de posição do motor CC construiu-se um sistema de informação que é utilizado para definir o comportamento do controlador aproximado. Com base nas funções de pertinência foram delimitadas as regiões de decomposição de conjuntos *fuzzy* em conjuntos *crisp*.

A Figura 3.9 representa o sistema de controle proposto. Trata-se da substituição do bloco do controlador *fuzzy* por um bloco composto por um conjunto de regras que define o controlador em termos de conjuntos aproximados. Dessa forma, os dois sistemas de controle possuem condições de operação semelhantes o que possibilita comparar os resultados.

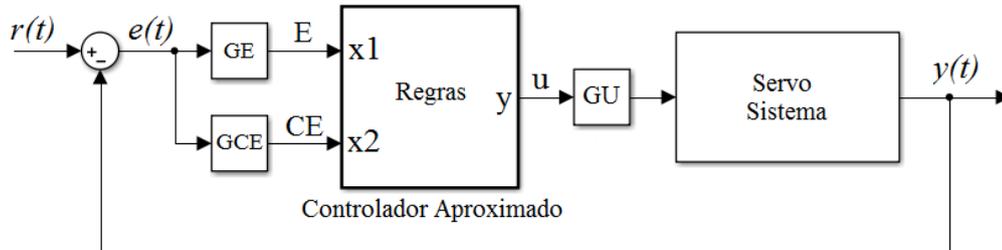


Figura 3.9 – Malha de controle com o controlador Aproximado.

A variável “E” (erro) está associada à informação x_1 , a variável “CE” (variação do erro) está associada com x_2 , e “u” com a informação de saída $F(x)$. Os intervalos dos conjuntos *crisp* são definidos em $\alpha_1 = [-1; -0,7)$, $\alpha_2 = [-0,7; -0,35)$, $\alpha_3 = [-0,35; 0)$, $\alpha_4 = [0; 0,35)$, $\alpha_5 = [0,35; 0,7)$, $\alpha_6 = [0,7; 1]$, $\beta_1 = [-1; -0,42)$, $\beta_2 = [-0,42; 0)$, $\beta_3 = [0; 0,42)$, $\beta_4 = [0,42; 1]$, conforme a Tabela 3.4.

Tabela 3.4 – Sistema de Informação obtido com o controle *Fuzzy*.

o_i	$x_1=E$	$x_2=CE$	$F(x)=u$
o1	[-1,00; -0,70)	[-1,00; -0,42)	[-3,42; -3,42]
o2	[-0,70; -0,35)	[-1,00; -0,42)	[-3,42; -3,42]
o3	[-0,35; 0,00)	[-1,00; -0,42)	[-3,42; -1,75]
o4	[0,00; 0,35)	[-1,00; -0,42)	[-1,75; -1,75]
o5	[0,35; 0,70)	[-1,00; -0,42)	[-1,75; 7,63e-17]
o6	[0,70; 1,00]	[-1,00; -0,42)	[7,6e-17; 7,6e-17]
o7	[-1,00; -0,70)	[-0,42; 0,00)	[-3,42; -1,75]
o8	[-0,70; -0,35)	[-0,42; 0,00)	[-3,42; -1,75]
o9	[-0,35; 0,00)	[-0,42; 0,00)	[-3,42; 7,63e-17]
o10	[0,00; 0,35)	[-0,42; 0,00)	[-1,75; 1,75]
o11	[0,35; 0,70)	[-0,42; 0,00)	[-1,75; 1,75]
o12	[0,70; 1,00]	[-0,42; 0,00)	[7,6e-17; 1,75]
o13	[-1,00; -0,70)	[0,00; 0,42)	[-1,75; 7,63e-17]
o14	[-0,70; -0,35)	[0,00; 0,42)	[-1,75; 1,75]
o15	[-0,35; 0,00)	[0,00; 0,42)	[-1,75; 1,75]
o16	[0,00; 0,35)	[0,00; 0,42)	[7,6e-17; 3,42]
o17	[0,35; 0,70)	[0,00; 0,42)	[1,75; 3,42]
o18	[0,70; 1,00]	[0,00; 0,42)	[1,75; 3,42]
o19	[-1,00; -0,70)	[0,42; 1,00]	[7,6e-17; 7,6e-17]
o20	[-0,70; -0,35)	[0,42; 1,00]	[7,6e-17; 1,75]
o21	[-0,35; 0,00)	[0,42; 1,00]	[1,75; 1,75]
o22	[0,00; 0,35)	[0,42; 1,00]	[1,75; 3,42]
o23	[0,35; 0,70)	[0,42; 1,00]	[3,42; 3,42]
o24	[0,70; 1,00]	[0,42; 1,00]	[3,42; 3,42]

O sistema de informação apresentado na Tabela 3.4 foi desenvolvido através dos intervalos estabelecidos para as variáveis de entrada e das saídas obtidas pelo editor *ANFIS* do Matlab®, ilustrado na Figura 3.10.

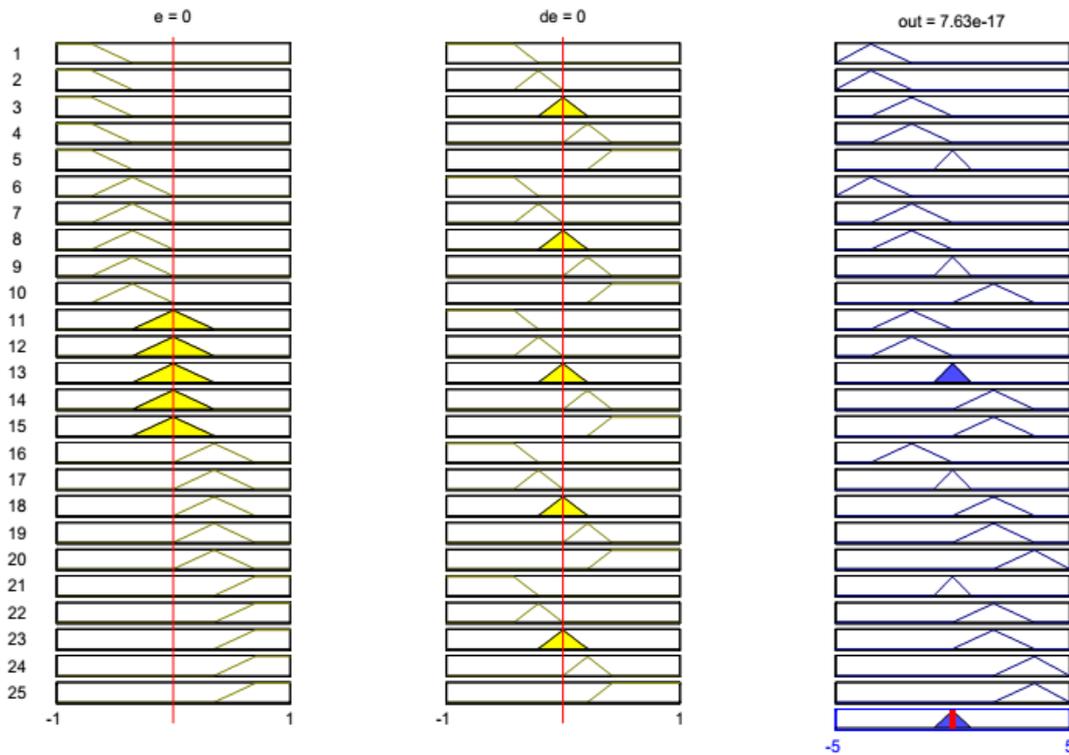


Figura 3.10 – Funções de Pertinência no *ANFIS*.

As regras de decisão são dadas por (3.5), o formato do conjunto de regras utilizam instruções do tipo *IF-THEN* que modelam o sistema de controle aproximado conforme os atributos de condição e decisão do sistema de informação.

$$\begin{aligned}
 r_1 &: \text{IF } x_1 = \alpha_1 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{1,0} + c_{1,1}x_1 + c_{1,2}x_2; \\
 r_2 &: \text{IF } x_1 = \alpha_2 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{2,0} + c_{2,1}x_1 + c_{2,2}x_2; \\
 r_3 &: \text{IF } x_1 = \alpha_3 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{3,0} + c_{3,1}x_1 + c_{3,2}x_2; \\
 r_4 &: \text{IF } x_1 = \alpha_4 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{4,0} + c_{4,1}x_1 + c_{4,2}x_2; \\
 r_5 &: \text{IF } x_1 = \alpha_5 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{5,0} + c_{5,1}x_1 + c_{5,2}x_2; \\
 r_6 &: \text{IF } x_1 = \alpha_6 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{6,0} + c_{6,1}x_1 + c_{6,2}x_2; \\
 r_7 &: \text{IF } x_1 = \alpha_1 \text{ AND } x_2 = \beta_2 \text{ THEN } y = c_{7,0} + c_{7,1}x_1 + c_{7,2}x_2; \\
 r_8 &: \text{IF } x_1 = \alpha_2 \text{ AND } x_2 = \beta_2 \text{ THEN } y = c_{8,0} + c_{8,1}x_1 + c_{8,2}x_2; \\
 &\dots \\
 r_{24} &: \text{IF } x_1 = \alpha_6 \text{ AND } x_2 = \beta_4 \text{ THEN } y = c_{24,0} + c_{24,1}x_1 + c_{24,2}x_2.
 \end{aligned} \tag{3.5}$$

Uma alternativa para representar um modelo aproximado com regras expressas na forma (3.5) é determinando os coeficientes numéricos com base na fórmula de interpolação linear (2.65), resultando nas Equações (3.6) e (3.7) (MACHADO e PINHEIRO, 2013). Convém observar que o método de conversão de modelos *fuzzy* em modelos aproximados pode ser aplicado tanto em sistemas *fuzzy* do tipo Mamdani, quanto o tipo Takagi-Sugeno.

$$c_{i,n} = \frac{1}{N} \left(\frac{y_i^{(m)} - y_i^{(k)}}{x_{i,n}^{(m)} - x_{i,n}^{(k)}} \right) \quad (3.6)$$

$$c_{i,0} = y_i^{(k)} + \frac{y_i^{(m)} - y_i^{(k)}}{N} \left(\sum_{n=1}^N \frac{-x_{i,n}^{(k)}}{x_{i,n}^{(m)} - x_{i,n}^{(k)}} \right) \quad (3.7)$$

Com os dados da Tabela 3.4 e as Equações (3.6) e (3.7) são definidos os coeficientes das regras aproximadas na Tabela 3.5. Conforme pode-se observar o controlador aproximado é constituído por 24 regras e a sua resposta ao degrau é apresentada em vermelho na Figura 3.11. O código utilizado na simulação está disponível no Apêndice B.1.

Tabela 3.5 – Coeficientes do conjunto de regras.

<i>i</i>	<i>c_{i,0}</i>	<i>c_{i,1}</i>	<i>c_{i,2}</i>
1	-3,4200	0,0000	0,0000
2	-3,4200	0,0000	0,0000
3	-1,1453	2,3857	1,4397
4	-1,7500	0,0000	0,0000
5	-1,1164	2,5000	1,5086
6	0,0000	0,0000	0,0000
7	0,1983	2,7833	1,9881
8	-0,9150	2,3857	1,9881
9	0,0000	4,8857	4,0714
10	0,0000	5,0000	4,1667
11	-1,7500	5,0000	4,1667
12	-1,1667	2,9167	2,0833
13	1,1667	2,9167	2,0833
14	1,7500	5,0000	4,1667
15	0,0000	5,0000	4,1667
16	0,0000	4,8857	4,0714
17	0,9150	2,3857	1,9881
18	-0,1983	2,7833	1,9881
19	0,0000	0,0000	0,0000
20	1,1164	2,5000	1,5086
21	1,7500	0,0000	0,0000
22	1,1453	2,3857	1,4397
23	3,4200	0,0000	0,0000
24	3,4200	0,0000	0,0000

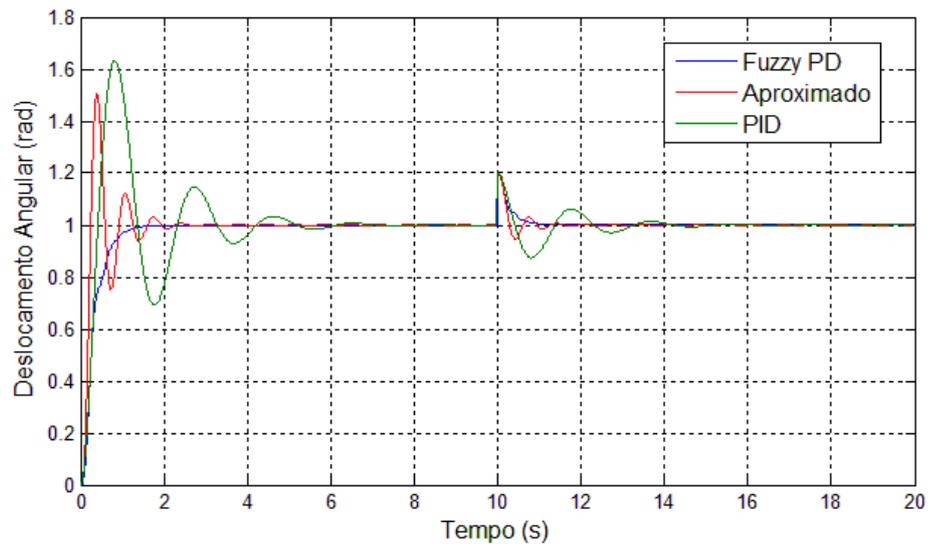


Figura 3.11 – Resposta ao degrau do controlador aproximado com o distúrbio externo.

A simulação da entrada tipo degrau mostra que o controlador aproximado possui um *overshoot* de 50%, e o tempo de acomodação de 1,85 s. O controlador PID possui um *overshoot* de aproximadamente 63% e o tempo de acomodação em torno de 5 s. Os resultados apresentados na Tabela 3.6 indicam o melhor desempenho do controlador aproximado em comparação ao controlador PID. Isso ocorre devido o controlador convencional possuir ganhos constantes, enquanto o controlador aproximado realiza a mudança de ganhos com a ativação de determinadas regras de acordo a necessidade do sistema.

Uma das características dos controladores é a capacidade de rejeitar perturbações externas. Uma máquina elétrica geralmente tem a posição do seu eixo afetada por esses distúrbios. Diante dessas considerações, um distúrbio externo com o tipo de deslocamento em degrau de 0,2 radianos é aplicado no instante $t = 10$ s. Percebe-se na Figura 3.11 que os controladores *fuzzy* e aproximado eliminam o distúrbio abrupto em torno de 1 s. Enquanto o controlador PID consegue atenuar o distúrbio em 5 s. Esse resultado era esperado devido os controladores inteligentes serem mais robustos para lidar com os distúrbios.

Para verificar a resposta dos controladores PID, *fuzzy* e aproximado, realizou-se simulações baseadas no mesmo sinal de entrada para o sistema com o modelo do motor CC considerado. Os resultados ilustrados na Tabela 3.6 estão relacionados com a Figura 3.11 que representa a dinâmica original dos métodos de controle de posição abordados. Percebe-se que o controlador *Fuzzy PD* atende as especificações com a determinação da ausência de *overshoot* (M_p), o menor tempo de acomodação (T_s) e o tempo de subida (T_r). Contudo, este controlador apresentou o maior tempo de subida. A outra desvantagem é o seu maior tempo de execução

do algoritmo de controle (T_e), aproximadamente 18 vezes maior em relação ao controlador PID e 2,5 vezes maior em relação ao controlador aproximado. E isso era esperado, devido a maior complexidade existente na estrutura do controlador *fuzzy*.

Tabela 3.6 – Comparação do desempenho dos controladores.

Critério	PID	Fuzzy PD	Aproximado
<i>Overshoot: M_p (%)</i>	63,14	0	50,6
Tempo de subida: T_r (s)	0,280	0,560	0,140
Tempo de acomodação: T_s (s)	4,940	1,160	1,850
Tempo de execução: T_e (s)	0,219	3,880	1,541
Quantidade de regras	-	25	24
ITAE	1,304	0,097	0,171
IAE	0,979	0,348	0,378
ISE	0,434	0,216	0,193

Os critérios de desempenho ITAE, IAE e ISE apresentados na Tabela 3.6 são usados para avaliar e comparar o desempenho das malhas de controle. Os menores valores destes critérios indicam o melhor desempenho. Dessa forma, percebe-se o menor desempenho do controlador PID por possuir índices mais altos. Com base nesses critérios também fica demonstrado que existe uma semelhança entre o desempenho dos controladores *fuzzy* e o aproximado.

É possível encontrar uma melhor aproximação no regime transitório ao aumentar a quantidade de regiões de decomposição de conjuntos *fuzzy* em conjuntos *crisp*. Entretanto, isso aumenta a quantidade de regras e conseqüentemente o tempo de processamento do sistema. Neste caso, a alternativa encontrada para melhorar a semelhança dos controladores *fuzzy* e aproximado foi manter a mesma quantidade de regras e realizar o ajuste dos ganhos de acordo a Tabela 2.1, devido a capacidade do ajuste desses parâmetros melhorar a resposta do sistema. Assim, com as mesmas 24 regras aproximadas e o ajuste do ganho proporcional, ou seja, a sua redução de $GE = 0,5$ para $GE = 0,23$, tornou-se possível eliminar o *overshoot* e aumentar a semelhança entre os controladores inteligentes, tanto no regime transitório quanto na rejeição ao distúrbio, conforme mostra a Figura 3.12.

O resultado da simulação da Figura 3.12 mostra que a estrutura de controle proposta possui um comportamento semelhante ao controlador *fuzzy* e também é possível realizar o ajuste de parâmetros, demonstrando a flexibilidade no desenvolvimento do controle. Nesse contexto, o controlador resultante é mais rápido e mantém as características de desempenho, atendendo às especificações do projeto.

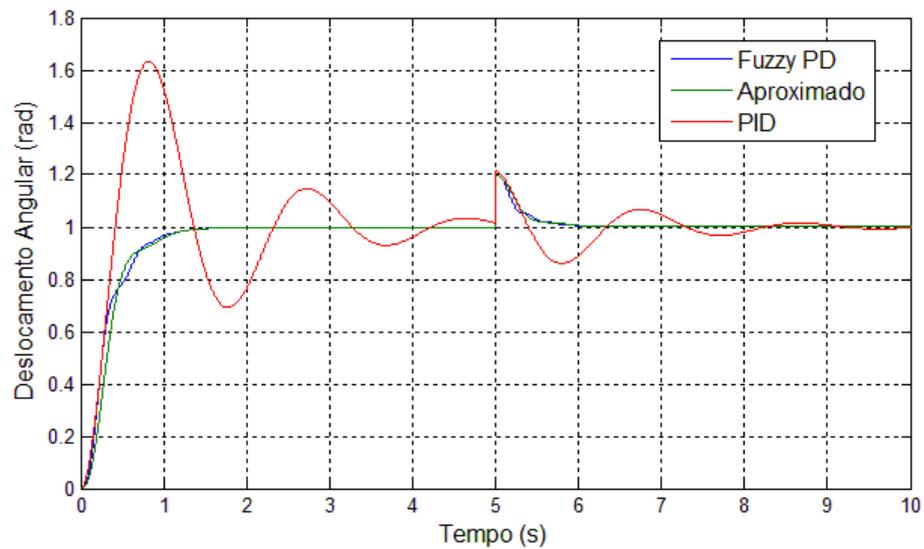


Figura 3.12 – Resposta ao degrau do controlador aproximado com o ajuste de ganho.

3.5 Conclusão

A abordagem deste capítulo mostrou as vantagens de cada método de controle inteligente. O controlador *fuzzy* apresentou um melhor desempenho no regime transitório enquanto o controlador aproximado mostrou um menor tempo de computação. Contudo, os índices de desempenho baseados no erro e a resposta resultante de ambos os controladores foram semelhantes.

O capítulo 4 apresenta o desenvolvimento experimental de uma estrutura de controle *fuzzy* conforme a Figura 3.4. Neste caso, são realizados testes práticos em um motor CC e um novo sistema de inferência *fuzzy* é usado como modelo operacional para o desenvolvimento do controlador aproximado.

4 Desenvolvimento de um controlador Aproximado

As informações de sistemas reais geralmente são incertas, imprecisas ou incompletas (LIMA, SANTOS e PINHEIRO, 2014). Neste capítulo os conceitos dos conjuntos aproximados são usados para processar essas informações e gerar o conhecimento sob a forma de regras para serem aplicadas em malhas de controle.

4.1 Caracterização do problema

Em sistemas reais os motores CC tendem a apresentar não linearidades e variações de parâmetros. Os controladores robustos que utilizam estratégias de inteligência artificial são uma alternativa para o controle de posição exato. Este capítulo apresenta o desenvolvimento de um controlador aproximado a partir de um modelo de controle *fuzzy* com o objetivo de obter uma estrutura de controle simplificada e com menor tempo de processamento. Ao invés da conversão baseada em funções de pertinência como foi visto no capítulo 3, o controlador *fuzzy* é utilizado como modelo operacional para determinar a base de conhecimento que define o controlador proposto.

O controle de posição do eixo do motor CC conta com a tensão aplicada em seus terminais e a posição angular resultante na saída. O processo é realizado em malha fechada, onde as informações do sinal de saída são utilizadas para correção do erro entre o sinal de referência e a posição real.

O desenvolvimento do sistema de controle proposto envolve a identificação de sistemas, o projeto de um controlador *Fuzzy* PD e o processamento de dados de ensaios utilizando os conceitos dos conjuntos aproximados.

4.2 Construção do protótipo do sistema de posicionamento

Um sistema de aquisição de dados ligado a um computador via USB é responsável por obter os dados de entrada do sistema pelo conversor analógico-digital (A/D) e permite executar as ações de comando no atuador através do conversor digital-analógico (D/A), considerando um período de amostragem $T_a = 0,01$ s. A Figura 4.1 ilustra o sistema aplicado no ensaio prático. O bloco denominado módulo representa o *driver* de acionamento.

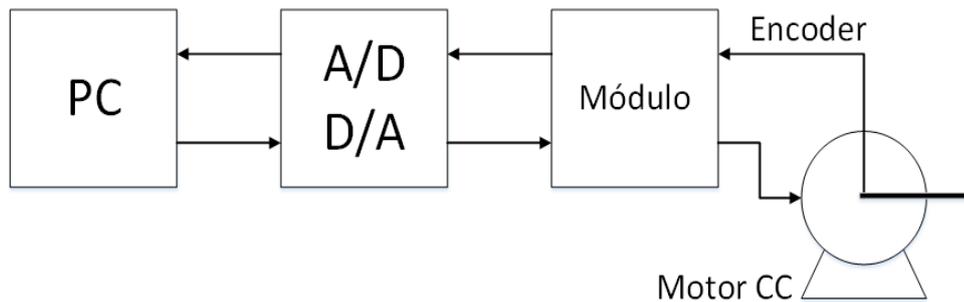


Figura 4.1 – Sistema de controle de posição considerado no ensaio.

O computador realiza o processamento das informações e a execução dos algoritmos de controle no Matlab/Simulink[®]. Este *software* possui um conjunto de diagramas de blocos de suporte a placa *Arduino Mega 2560* usada no processo de aquisição de dados a fim de obter uma simplicidade e baixo custo do projeto. A placa de aquisição de dados tem como principais características um microcontrolador *ATmega 2560* de 8 *bits*, conversor A/D com uma resolução de 10 *bits*, memória *Flash* e RAM.

Os resultados experimentais foram obtidos através de medidas reais do protótipo construído. É interessante observar que a placa de aquisição de dados, o computador e o *software* Simulink[®] constituem o equipamento de medida que possibilitou a comparação de desempenho dos algoritmos e a geração dos gráficos resultantes que representam as respostas dinâmicas do sistema.

O controle de sistemas requer a utilização de atuadores e sensores. O atuador do sistema é um motor de ímã permanente com uma tensão nominal de 6 V, corrente de 70 mA, e velocidade de 530 RPM. Este motor possui o circuito equivalente semelhante as máquinas CC de excitação externa. A principal diferença é que as máquinas de ímãs permanentes não possuem o enrolamento de campo, apenas o enrolamento de armadura, pois o campo é gerado pelos ímãs. Isso resulta em máquinas de tamanho reduzido e com um projeto mais simples, onde o seu acionamento é realizado pelo circuito *driver ponte H*.

O sensor utilizado é um encoder quadrático acoplado ao eixo do motor. Este transdutor converte o movimento angular em uma série de pulsos digitais que por sua vez são fundamentais para determinar a velocidade, posição ou direção. Nesse projeto o encoder é utilizado para a medição da posição angular do eixo do motor e também serve para informar o seu sentido de rotação (horário ou anti-horário). Assim, essas informações indicam a posição da haste acoplada ao eixo motor e o seu sentido de deslocamento. A Figura 4.2 mostra o diagrama de blocos da malha de controle que está relacionada com a estrutura apresentada na Figura 4.1.

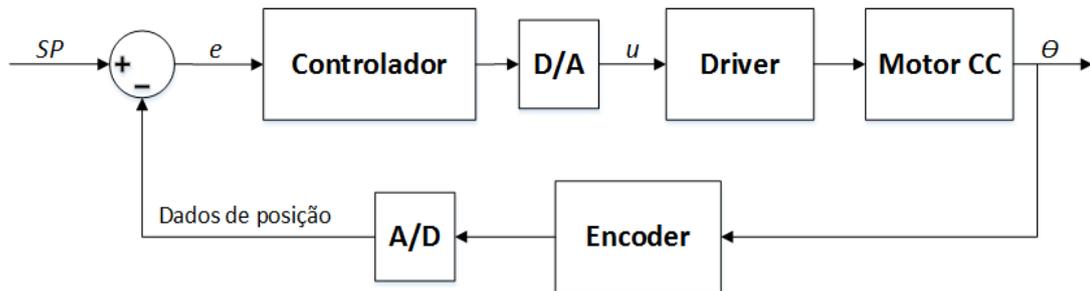


Figura 4.2 – Diagrama de blocos da malha de controle para o ensaio.

Neste capítulo os sistemas foram desenvolvidos experimentalmente com base em um protótipo e nas malhas de controle definidas no ambiente Simulink[®] (conforme está disponível no apêndice A). Dessa forma, os gráficos dos ensaios práticos foram obtidos com a mesma ferramenta computacional usada nas simulações. A Figura 4.3 mostra a foto do protótipo construído com o objetivo de avaliar a resposta dos controladores em um sistema real. Uma placa de circuito impresso foi projetada a fim de conectar os componentes (*driver*, motor, encoder e bateria) a placa de aquisição de dados.

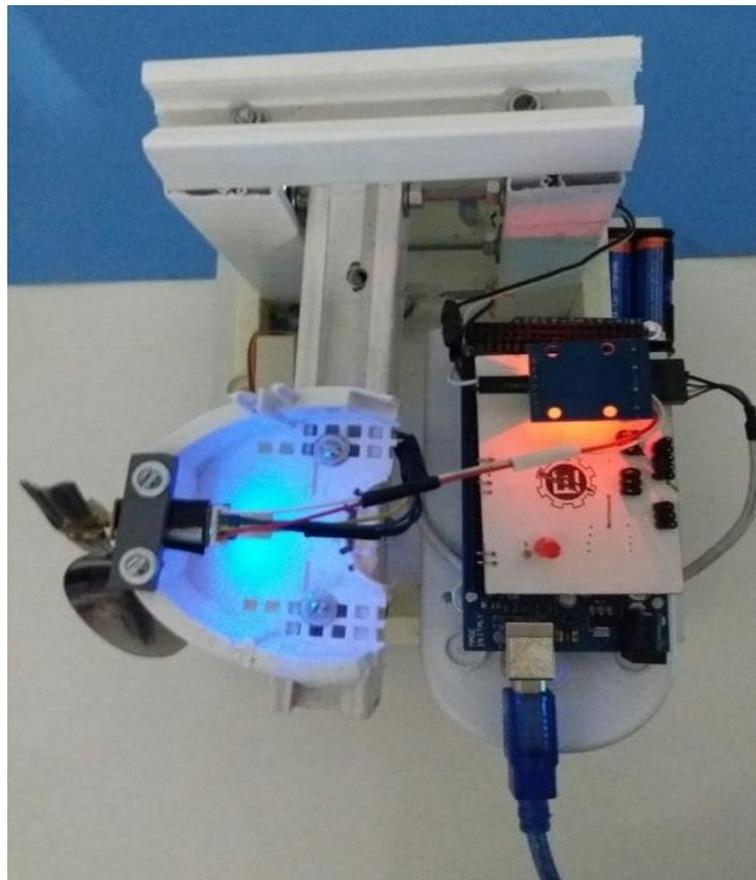


Figura 4.3 – Protótipo do sistema utilizado nos experimentos práticos.

4.3 Identificação de um modelo ARX

A identificação do sistema é realizada antes da sintonia do controlador. Para isto é desejável um sinal de entrada que excite a dinâmica da planta. Do ponto de vista numérico quanto mais aleatório for o sinal de entrada melhor condicionado será o problema. Diante deste fato, o sinal de sequência binária pseudo-aleatória (PRBS) é usado para excitar o sistema. Um exemplo da aplicação desses sinais foi realizado em um conjunto de microrrobôs com motores CC, onde a identificação contou com os sinais de entrada PRBS e os sinais de saída referente a posição dos robôs (AGUIRRE, 2004). Os sinais PRBS assumem apenas dois valores, o nível alto (+V) e o valor correspondente ao nível baixo (-V). A variação dos níveis desse sinal ocorre em instantes bem definidos $t = 0, T_b, 2T_b$. Os períodos são iguais a $T = (2^n - 1)T_b$, onde n é o número de *bits* do registro de deslocamento e T_b é o intervalo entre *bits*.

O intervalo entre *bits*, deve ser menor que a constante de tempo do processo. Convém observar que se T_b possuir um valor alto, o sistema interpretará o sinal PRBS como um degrau. Por outro lado, se o valor de T_b for muito pequeno, o sistema não irá responder a tempo uma transição antes de chegar a outra. Diante dessas considerações existe um método heurístico apresentado em Aguirre (2004) que sugere o valor de T_b conforme a relação (4.1).

$$\frac{\tau_{\min}}{10} \leq T_b \leq \frac{\tau_{\min}}{3} \quad (4.1)$$

Onde τ_{\min} é a menor constante de tempo de interesse, sendo obtido um valor em torno de 0,3 s através da resposta ao degrau. O sinal PRBS de sequência m é gerado a partir de operações OU-Exclusivo entre alguns *bits* do registro de deslocamento conforme o circuito mostrado no Apêndice A.1. A periodicidade do sinal é definida com o número de *bits* do registrador de deslocamento $n = 11$ e o intervalo entre *bits* $T_b = 0,1$ s, determinado a partir da relação 4.1.

A Figura 4.4 mostra parte dos dados usados no processo de identificação. O sinal PRBS (uI) gerado em um computador é aplicado ao motor. A saída é a posição angular dada em graus (yI). O sistema foi excitado pelo sinal PRBS por um período de 40 segundos. Aplicou-se a maior faixa dessa resposta na modelagem e aproximadamente 20% das amostras foram utilizadas na validação do modelo.

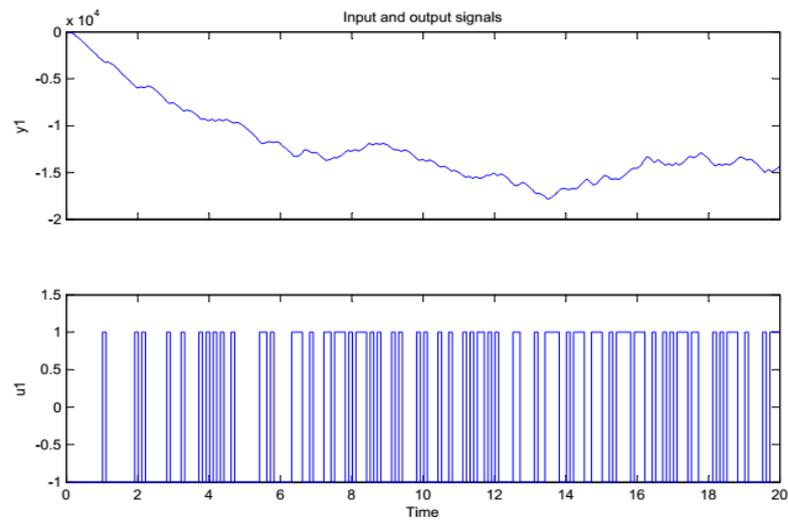


Figura 4.4 – Sinais de entrada e saída usados na identificação do sistema.

Em muitos casos o modelo linear é suficiente para o desenvolvimento do sistema de controle. A partir dos dados de entrada e saída foi estabelecido o modelo ARX que utiliza os mínimos quadrados para estimar os parâmetros. A identificação realizou-se com o auxílio da *toolbox System Identification Tool* do Matlab[®]. O processo de validação consiste na aplicação do sinal PRBS na entrada do modelo para comparar a posição simulada com a posição real medida, conforme a Figura 4.5. Através de medições experimentais das entradas e saídas do sistema foi obtido o modelo ARX dado por (4.2), no qual apresentou uma correlação de 90,98% em relação aos dados de ensaio.

Modelo ARX : $A(q)y(t) = B(q)u(t) + e(t)$

$$A(q) = 1 - 1,361q^{-1} + 0,3255q^{-2} - 0,05411q^{-3} - 0,01995q^{-4} + 0,1098q^{-5} \quad (4.2)$$

$$B(q) = 16,98q^{-3} + 12,07q^{-4}$$

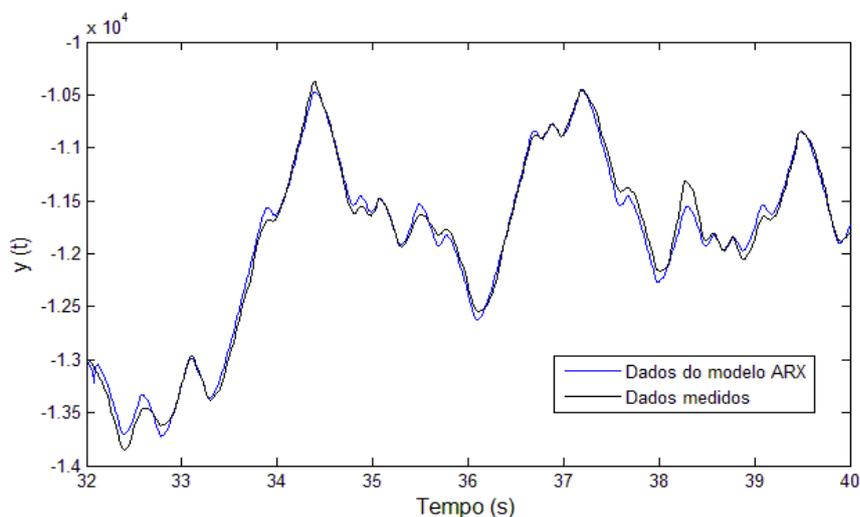


Figura 4.5 – Resposta do modelo ARX e os dados do ensaio.

4.4 Projeto do controlador *Fuzzy* PD

O modelo identificado é usado para gerar os ganhos do controlador PD convencional em malha fechada através do bloco de ajuste automático do Matlab/Simulink[®]. Esta malha de controle em diagrama de blocos se encontra no Apêndice A.2. A resposta ao degrau apresentou um tempo de acomodação de 1,5 s como mostra a Figura 4.6, sendo obtido $K_P = 38,85 \cdot 10^{-5}$, $K_D = 1,51 \cdot 10^{-5}$ e $T_D = 0,039$ s. A partir desses valores foram empregadas as relações (2.50) e (2.51) que determinam os ganhos resultantes do controlador *Fuzzy* PD: $GE = 0,0001$, $GCE = 3,9 \cdot 10^{-6}$ e $GU = 3,885$.

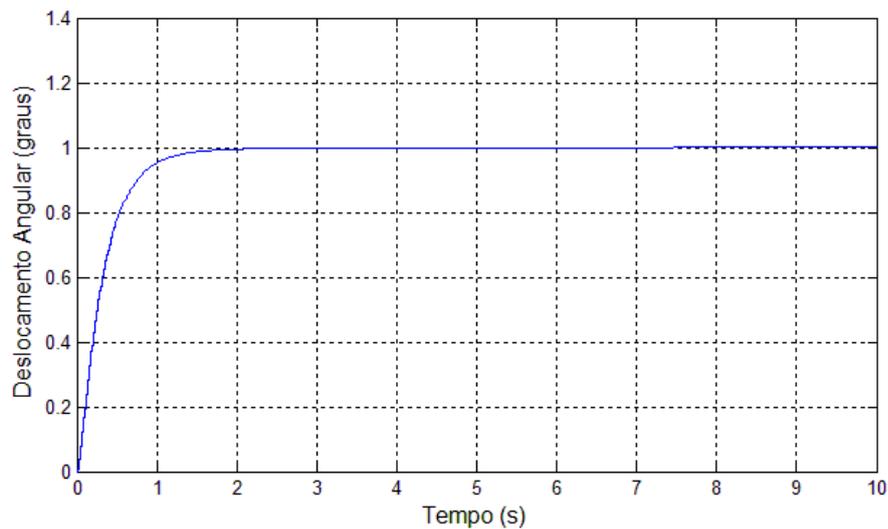


Figura 4.6 – Simulação da malha de controle com o controlador convencional PD.

A estrutura do controlador *Fuzzy* PD utilizada neste capítulo é semelhante a da Figura 3.4, sendo adicionado os ganhos de conversão de graus para radianos e vice-versa, já que o controlador trabalha com valores em radianos e a saída é dada em graus. O sistema com o controlador *fuzzy* usado no ensaio está disponível no Apêndice A.3.

Geralmente o projeto de controladores envolvem incertezas em processos reais devido a variação de parâmetros, não linearidades ou mesmo a utilização de modelos lineares a fim de diminuir a complexidade matemática. Esses fatores justificam a aplicação do controlador *fuzzy* por ser capaz de lidar com essas incertezas sem a necessidade de um modelo exato.

A metodologia de projeto do controlador *fuzzy* realizou-se com base nas abordagens clássicas para a sintonia experimental por tentativa e erro (GOMIDE e GUDWIN, 1994). Esse processo se inicia com um controlador mais simples. Neste caso, determinou-se um menor número de variáveis linguísticas (5 variáveis) e funções de pertinências simétricas, cuja primeira resposta do ensaio prático é apresentada na Figura 4.7 para o ângulo de referência em

90 graus. Percebe-se um elevado erro em regime permanente devido o modelo ARX simulado e as não linearidades do sistema real. Dessa forma, o projeto prossegue com a sintonia do controlador *fuzzy* que é uma das etapas mais custosas, dado que existem muitos parâmetros no procedimento de sintonia.

De acordo com os testes realizados e a experiência adquirida sobre o processo, novas variáveis linguísticas foram adicionadas, além das modificações em algumas funções de pertinência para torná-las assimétricas e condizentes com a necessidade da aplicação.

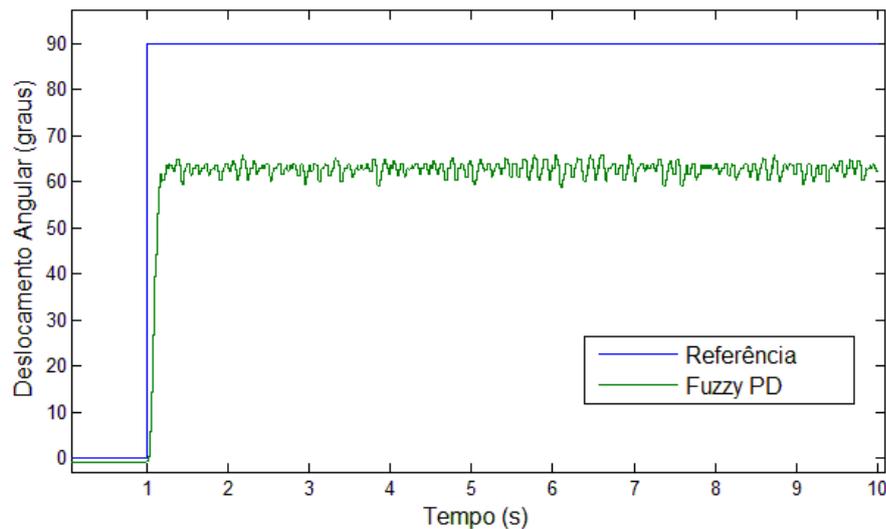


Figura 4.7 – Resposta do controlador *Fuzzy PD* com 5 variáveis linguísticas.

Para o procedimento de sintonia é necessário uma interface que permita a interação com o projetista e uma rápida modificação dos parâmetros envolvidos (GOMIDE e GUDWIN, 1994). O apêndice A mostra os sistemas de controle modelados com os diagramas de blocos do Matlab/Simulink®. Esta ferramenta computacional permite a realização de ajustes no controlador, alterações de parâmetros da malha de controle e a observação do desempenho do sistema. Nesse contexto, o sistema de inferência do controlador *fuzzy* foi desenvolvido através do *Fuzzy Logic Toolbox* do Matlab®.

A conversão dos valores das variáveis de posição em valores de pertinência dos conjuntos difusos corresponde a fuzificação. Nesta etapa foram definidas duas variáveis de entrada: o erro “E”, determinado pela diferença entre o valor de referência e o valor da posição atual, e a derivada deste erro “CE”. A Figura 4.8 mostra as funções de pertinência utilizadas para a variável do erro.

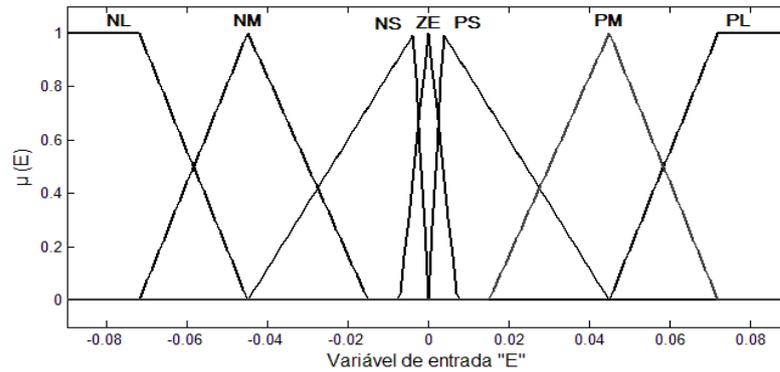


Figura 4.8 – Funções de pertinência da variável E do controlador *fuzzy*.

Percebe-se que o intervalo no qual o erro é considerado zero é o mais reduzido. Este limite tem um papel fundamental no desempenho do controlador, pois contribui para o menor erro em regime permanente. A Figura 4.9 mostra as funções de pertinência utilizadas para fuzzificar a variável da derivada do erro.

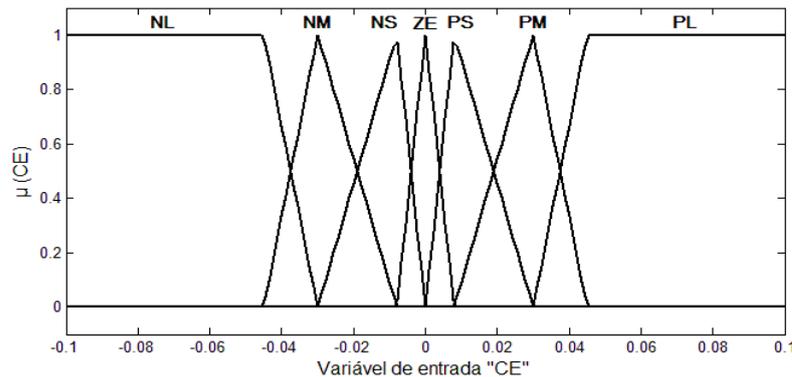


Figura 4.9 – Funções de pertinência da variável CE do controlador *fuzzy*.

O universo de discurso da variável “E” abrange o intervalo $[-0,09; 0,09]$ e a variável “CE” possui o intervalo $[-0,1; 0,1]$. Esses parâmetros foram obtidos com base nos ensaios experimentais para o ajuste da referência, mostrando que o erro e a variação do erro possuem valores baixos. A partir dos ajustes realizados foram definidas sete funções de pertinências denominadas: NL (negativo grande), NM (negativo médio), NS (negativo pequeno), ZE (zero), PS (positivo pequeno), PM (positivo médio) e PL (positivo grande). Utilizou-se o tipo de inferência Mamdani por ser uma técnica eficiente no controle de motores de corrente contínua. As funções de pertinência definidas para a saída “u” são mostradas na Figura 4.10.

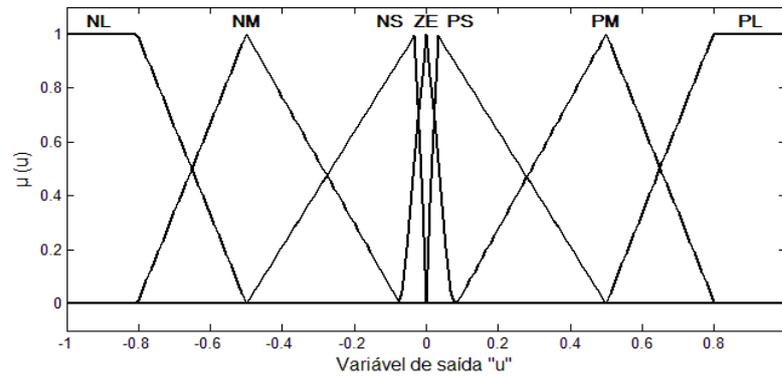


Figura 4.10 – Funções de pertinência da variável u do controlador *fuzzy*.

O sistema em questão possui um circuito de acionamento que funciona com níveis de tensão entre 0 e 5 V em duas portas de entrada A e B, onde cada uma determina um sentido de rotação para o motor. Os valores de saída foram limitados entre -1 a 1, ou seja, em um universo de discurso de saída com o intervalo $[-1; 1]$. Este limite está relacionado com a saída digital do sistema de controle implementado no Simulink®, o nível mínimo deste sinal corresponde a tensão de 5 V na entrada A que faz o motor girar em um sentido e o nível máximo equivale a 5 V aplicado na entrada B que faz o motor girar no sentido contrário. O código usado para esse procedimento encontra-se no Apêndice B.2.

As funções triangulares e trapezoidais foram utilizadas com o propósito de facilitar o desenvolvimento do sistema de inferência. Isso contribuiu para o ajuste de funções de pertinência, através das alterações nos universos de discurso ou mudanças na forma das funções. O método de inferência MAX-MIN é utilizado para a composição de cada regra de controle e estabelece a relação entre elas. O processo de defuzificação é calculado pelo método Centro de Área. Com base nas entradas e funções de pertinências definidas, foram determinadas 49 regras que estão resumidas na Tabela 4.1. A Figura 4.11 mostra a resposta do controlador *fuzzy* para o ensaio realizado com o ângulo de referência em 90 graus.

Tabela 4.1 – Base de regras do controlador *Fuzzy* PD usado no ensaio.

u		CE						
		NL	NM	NS	ZE	PS	PM	PL
E	NL	NL	NL	NL	NL	NM	NS	ZE
	NM	NL	NL	NL	NM	NS	ZE	PS
	NS	NL	NL	NM	NS	ZE	PS	PM
	ZE	NL	NM	NS	ZE	PS	PM	PL
	PS	NM	NS	ZE	PS	PM	PL	PL
	PM	NS	ZE	PS	PM	PL	PL	PL
	PL	ZE	PS	PM	PL	PL	PL	PL

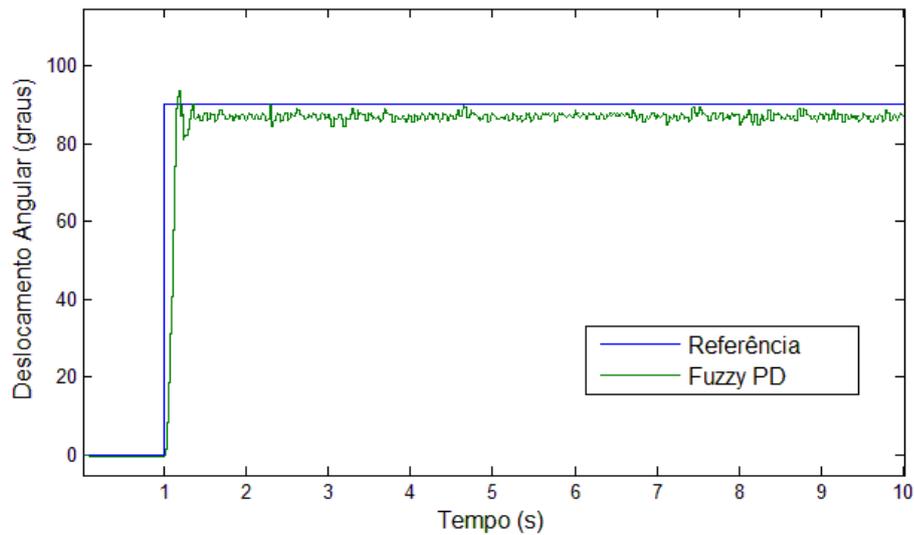


Figura 4.11 – Resposta do controlador *Fuzzy PD* com 7 variáveis linguísticas.

Uma característica importante é que as funções de pertinência denominadas NS e PS são assimétricas devido a intenção de melhorar as condições do sistema diante as não linearidades. A Figura 4.11 mostra que o tempo de acomodação foi de 0,35 s, o *overshoot* de aproximadamente 4% e o erro em regime permanente é inferior a 4,5%, conforme o especificado. Neste caso, o aumento na quantidade de variáveis linguísticas também contribuiu para a redução considerável do erro em regime permanente.

4.5 Projeto do controlador Aproximado

O modelo baseado em regras aproximadas pode ser obtido com base nos dados de um sistema físico. Desse modo, o projeto do controlador aproximado é realizado a partir de um sistema de informação gerado com a aquisição de dados do sistema de controle de posição real que utiliza o controlador *fuzzy*. A Figura 4.12 apresenta a malha de controle e os pontos usados para determinar os atributos.

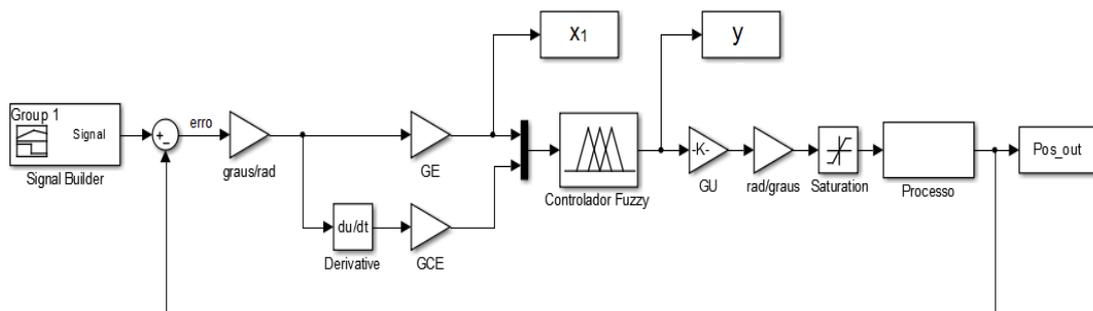


Figura 4.12 – Exemplo do controlador *Fuzzy PD* em uma malha de controle.

A partir dos resultados obtidos com o controlador *fuzzy* foram definidas as especificações de projeto do controlador aproximado: tempo de acomodação igual ou menor que 2 segundos, o *overshoot* igual ou menor que 30% e o erro em regime permanente menor que 5%. Devido a estrutura simplificada do controlador aproximado, espera-se um menor desempenho no regime transitório. Desse modo, foi considerado um *overshoot* maior em relação ao sistema *fuzzy*.

A Figura 4.13 mostra a resposta da malha de controle com o controlador *fuzzy* usado como modelo operacional. O sistema foi executado a partir do sinal de referência que abrange os ângulos de interesse até o limite de 130 graus, a fim de determinar os dados necessários do sistema de informação da Tabela 4.2. Foram definidos dois atributos de condição: x_1 indica a ação proporcional e x_2 refere-se a amostra anterior da saída do controlador *fuzzy* $y(k-1)$. O atributo de decisão y indica a informação da atuação do controlador $y(k)$.

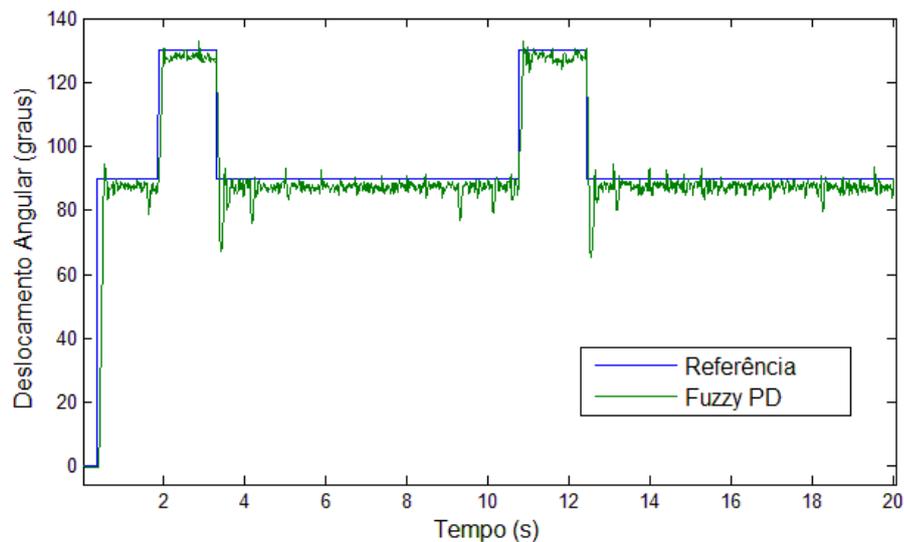


Figura 4.13 – Resposta da malha de controle que define os dados do SI.

Tabela 4.2 – Sistema de Informação do modelo de controle.

<i>Amostras</i>	x_1	x_2	y
⋮	⋮	⋮	⋮
14	0	0	0
15	-0,00764	0	-0,001228
16	-0,00764	-0,001228	-0,000349
⋮	⋮	⋮	⋮
423	-0,248	-0,0224	-0,013
424	-0,248	-0,013	-0,0109
425	-0,241	-0,0109	-0,0099
⋮	⋮	⋮	⋮
1999	-0,0573	-0,0027	-0,000161
2000	-0,0191	-0,000161	0,002168

Os dados do SI da Tabela 4.2 são inseridos no *software* Rosetta para gerar um conjunto de regras que representam o controlador proposto. O método de discretização foi o *Equal Frequency Binning* sendo atribuído 3 divisões (classes de discretização). Assim, os dados são divididos em intervalos com aproximadamente o mesmo número de observações. Os redutos foram calculados com base em um algoritmo genético, resultando em 9 regras para o modelo aproximado. O resumo dos procedimentos realizados no Rosetta foram:

1. *Import IS* \rightarrow *plain format* (insere os dados do SI no Rosetta);
2. *Discretization* \rightarrow *Equal frequency binning* \rightarrow *Number of Intervals=3* (Divisão de cada atributo de condição em 3 intervalos);
3. *Reduction* \rightarrow *Genetic Algorithm* \rightarrow *Full* (Cálculo do reduto);
4. *Generate rules* (Geração de regras).

O exemplo do conjunto de regras geradas pelo *Rosetta* é apresentado em (4.2). Neste caso, existem dois antecedentes e uma consequência em cada regra. Há de se considerar que a utilização desse *software* permite definir regras consistentes, pois as redundâncias do SI não são incluídas no processo. Essas regras são do tipo condição-ação e podem ser usadas como base de conhecimento para representar o comportamento do novo controlador em questão. O código do controlador aproximado com todas as regras envolvidas está disponível no Apêndice B.3.

R1: $x_1([-0.031849, *]) \text{ AND } x_2([-0.001219, *]) \Rightarrow y(0.000000) \text{ OR } y(-0.001228) \dots \text{ OR } y(0.014500) \text{ OR } y(0.002670) \text{ OR } y(-0.000605) \text{ OR } y(0.009880) \text{ OR } y(0.006137) \text{ OR } y(-0.005728)$;

R2: $x_1([-0.031849, *]) \text{ AND } x_2([-0.002940, -0.001219]) \Rightarrow y(-0.000349) \text{ OR } y(-0.001660) \text{ OR } y(-0.004570) \dots \text{ OR } y(-0.000871) \text{ OR } y(-0.001951) \text{ OR } y(0.002865) \text{ OR } y(0.001821)$;

R3: $x_1([*, -0.050949]) \text{ AND } x_2([-0.001219, *]) \Rightarrow y(-0.150999) \text{ OR } y(0.001030) \text{ OR } y(-0.001190) \text{ OR } y(-0.007580) \text{ OR } y(-0.009950) \dots \text{ OR } y(-0.011668) \text{ OR } y(-0.005800) \text{ OR } y(-0.004009) \text{ OR } y(-0.008585)$;

...

R9: $x_1([-0.031849, *]) \text{ AND } x_2([*, -0.002940]) \Rightarrow y(0.000498) \dots \text{ OR } y(0.003846) \text{ OR } y(0.003251) \text{ OR } y(-0.000110) \text{ OR } y(0.002262) \text{ OR } y(0.000960)$. (4.2)

O símbolo “*” indica os valores dos limites inferiores e superiores dos atributos de condição que devem ser representados por $(x^{(k)}, x^{(m)})$, nesse caso os valores são -1,580 e 0,692 para x_1 , e -0,151 e 0,087 para x_2 , respectivamente. Para compor cada regra foram selecionados os valores mínimos e máximos do atributo de decisão que possui n valores, e são representados por $(y^{(k)}, y^{(m)})$. Desse modo, o exemplo de como a regra 9 pode ser reescrita é indicado no formato (4.3), por meio de intervalos que podem ativar a regra a depender das condições estabelecidas nos valores de entrada.

$$\begin{aligned} r_9 : IF -0,031849 \leq x_1 \leq 0,692 \text{ AND } -0,151 \leq x_2 \leq -0,00294 \\ THEN -0,00011 \leq y \leq 0,003846 \end{aligned} \quad (4.3)$$

A Figura 4.14 mostra a estrutura do modelo baseado em regras que representa o controlador aproximado (*rough*). Essa é uma estrutura simplificada e não necessita dos fatores de escala *GE* e *GCE* existentes no controlador *fuzzy*. O modelo aproximado foi definido através de atributos relevantes para o controle, expressos pela notação: $x_1 = u(k)$, $x_2 = y(k-1)$ e $y = y(k)$, onde o operador Z^{-1} determina as amostras anteriores da saída. A malha de controle completa com esse controlador é apresentada em diagramas de blocos no Apêndice A.4.

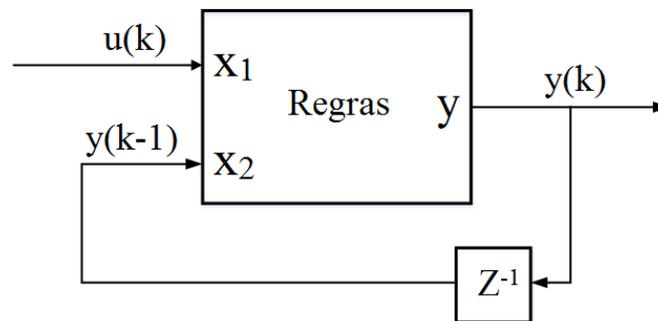


Figura 4.14 – Estrutura do controlador Aproximado.

A estimação de valores do modelo aproximado é realizada através da expressão de interpolação (2.65) e torna possível estimar valores mesmo em situações com diversas variáveis de entrada. Em (4.4) é apresentado um exemplo da aplicação dessa equação para o cálculo do valor de saída da regra 9, sendo considerado os dois atributos condicionais $N = 2$ e as variáveis de entrada x_1 e x_2 .

$$\begin{aligned} y_9 = -0,00011 + \left(\frac{0,003846 + 0,00011}{2} \right) \\ \left[\left(\frac{x_1 + 0,031849}{0,692 + 0,031849} \right) + \left(\frac{x_2 + 0,151}{-0,00294 + 0,151} \right) \right] \end{aligned} \quad (4.4)$$

O Simulink® permite a simulação e o controle de processos reais. Esse ambiente foi utilizado no desenvolvimento da malha de controle através dos diagramas de blocos. A placa de aquisição de dados possui um microcontrolador Atmega 2560 e está conectada ao computador. Realizou-se então os ensaios práticos com o sistema de controle proposto a fim de posicionar a haste em um ângulo desejado. A Figura 4.15 mostra a resposta do controlador aproximado para o sinal de referência em 90 graus.

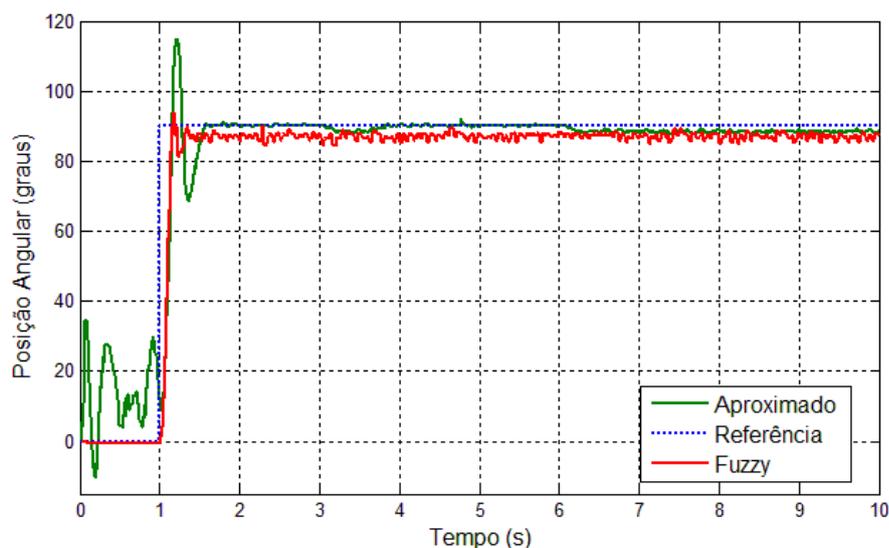


Figura 4.15 – Resposta do controle de posição para a referência em 90 graus.

O sistema de controle de posição baseado nos conceitos dos conjuntos aproximados mostrou-se estável. Com a utilização do controlador aproximado a posição do eixo do motor se estabiliza em 0,53 s, o erro em regime permanente é menor que 2,3% e o *overshoot* é de 27,5%. Pela Figura 4.15, observa-se um menor tempo de acomodação e o menor *overshoot* na resposta do controlador *fuzzy* devido as ações de controle proporcional e derivativa que atuam sobre o regime transitório.

O conjunto de informações é dividido quando se particiona o conjunto universo em determinada quantidade de classes de discretização (TAJIRI, 2009). É oportuno ressaltar que o aumento dessas classes promove o aumento da quantidade de regras, isso pode contribuir para a semelhança de comportamento com o modelo original no regime transitório e a consequente redução do *overshoot*. Contudo, dentre as especificações desejadas, buscou-se pela menor quantidade de regras e pelo menor erro em regime permanente. Dessa forma, o controlador definido atende as especificações de desempenho em um menor tempo de processamento.

O controlador aproximado determina o novo comportamento das ações de controle que são baseadas no conhecimento do modelo *fuzzy*. Este modelo geralmente possui maior exatidão

na modelagem de funções de controle em virtude da sobreposição dos conjuntos. No entanto, neste caso é possível perceber uma menor oscilação do controlador aproximado em regime permanente. Isso demonstra que o número de discretizações foi suficiente para manter a continuidade do sinal. A Figura 4.16 apresenta a resposta dos sistemas mediante variações no sinal de referência.

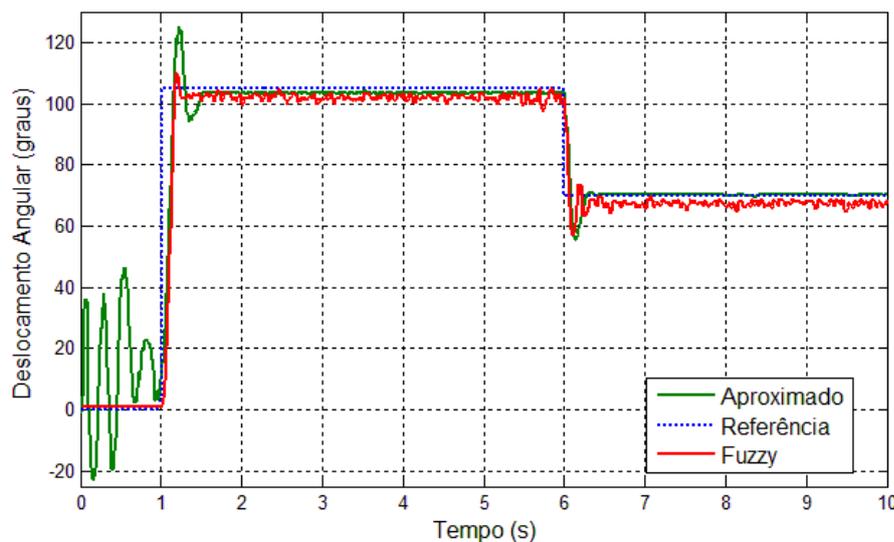


Figura 4.16 – Resposta do controle de posição com variação no sinal de entrada.

Os sinais de saída da Figura 4.16 mostram a capacidade de rastreamento do sistema tanto com o controlador *fuzzy* quanto o aproximado. Inicialmente, o sinal de entrada é definido em 105 graus e o tempo de acomodação é semelhante ao da Figura 4.15. Em seguida ocorre a mudança do sinal de referência para 70 graus e após 0,3 s o sistema proposto novamente se encontra em regime permanente. O *overshoot* do controlador aproximado nesse ensaio foi reduzido para 19% o que caracteriza um melhor desempenho na resposta transitória para um ângulo mais elevado.

Nas Figuras 4.15 e 4.16, percebe-se uma oscilação na resposta do controlador aproximado quando o sinal de referência ainda é nulo. Isso pode ser atribuído ao mapeamento da função de controle em questão e ao sistema de informação da Tabela 4.2 que possui poucas amostras para essa posição, já que na prática o posicionamento da haste foi realizado para os ângulos maiores que zero. Nesse contexto, é pertinente afirmar que o maior número de amostras em uma base de conhecimento pode contribuir para a melhor representação de determinado comportamento.

A quantidade de funções de pertinência e o número de entradas necessárias para o controle do processo resultaram em 49 regras para o controlador *fuzzy*. Enquanto o controlador

aproximado foi definido com 9 regras. Diferente dos sistemas *fuzzy* o modelo aproximado da Figura 4.14 não utiliza conjuntos difusos, mas apenas conjuntos convencionais (*crisp*). Assim, os valores de entrada são compatíveis com o modelo especificado e inseridos diretamente no conjunto de regras sem a necessidade dos procedimentos de escalonamento, fuzificação e defuzificação, resultando em uma estrutura de controle mais simples. Essas características do controlador aproximado contribuem para uma redução significativa do tempo de execução T_e do algoritmo de controle, sendo aproximadamente 72 vezes menor em relação ao controlador *fuzzy*. Isso demonstra a viabilidade do modelo aproximado para aplicações de sistemas de controle em tempo real.

A Tabela 4.3 resume os resultados obtidos com base no ensaio em regime permanente apresentado na Figura 4.15 para o sistema com os controladores em estudo.

Tabela 4.3 – Comparação dos resultados dos controladores.

Critério	Fuzzy PD	Aproximado
<i>Overshoot: M_p (%)</i>	4	27,5
Tempo de acomodação: T_s (s)	0,35	0,53
Tempo de execução: T_e (s)	19,42	0,27
Erro em regime permanente: e (%)	4,4	2,2
Quantidade de regras	49	9
ITAE	278,35	59,67
IAE	66,70	36,37
ISE	1332	852,30

Os índices de desempenho baseados no erro ITAE, IAE e ISE apresentados na Tabela 4.3, foram obtidos através do ensaio prático. Convém ressaltar que os menores valores retratam o melhor desempenho da malha de controle. Neste caso, os três critérios indicaram a malha com o controlador aproximado como a de melhor desempenho comparado ao controlador *fuzzy*. Dentre os critérios analisados, o ITAE apresenta a maior diferença entre o desempenho dos controladores, pois o erro inicial elevado no regime transitório é pouco ponderado, enquanto o erro que ocorre no regime permanente possui uma maior ponderação. Esses fatores contribuíram para indicar o melhor desempenho da malha de controle que possui o controlador aproximado.

A fim de se verificar a robustez dos controladores, realizou-se o ensaio experimental com a adição de carga na haste. O resultado do teste experimental é ilustrado na Figura 4.17. O acréscimo de carga reduziu a oscilação do sinal em regime permanente do sistema com o controlador *fuzzy*. As mesmas condições de carga foram estabelecidas para o controlador aproximado, conforme esperado, o *overshoot* aumentou para 32,3% e o tempo de acomodação

aumentou para 0,8 s, caracterizando o menor desempenho no regime transitório. Uma mudança de posição ocorre no instante $t = 6$ s, e a carga provocou um maior afundamento do sinal de saída em ambos os controladores, sendo então corrigida a posição em menos de 0,5 s. Enfim, a carga também proporcionou uma redução na oscilação inicial quando o sinal de referência é nulo.

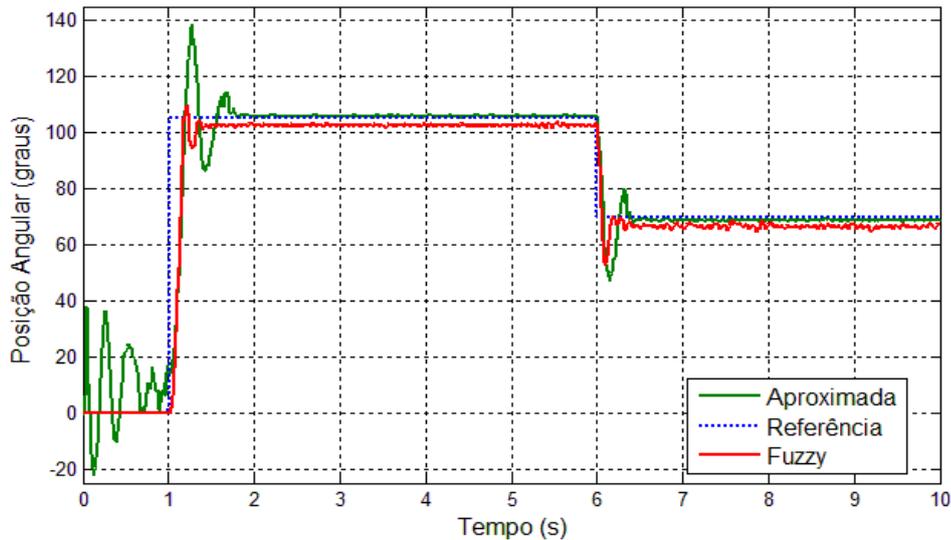


Figura 4.17 – Resposta do controle de posição com carga e variação no sinal de referência.

As respostas do controlador aproximado mostram condições estáveis e características dinâmicas adequadas para o sistema em malha fechada. Isso se deve ao fato das regras representarem determinadas condições operacionais do sistema, e por serem obtidas a partir de dados de um ensaio real, resultando numa abordagem interessante para ser aplicada em processos não lineares.

4.6 Conclusão

Esse capítulo mostrou o desenvolvimento do controlador aproximado a partir do conhecimento de um modelo de controle previamente estabelecido. Construiu-se um protótipo e foram realizados ensaios para a aquisição de dados do sistema com o controlador *fuzzy*. Com o conhecimento adquirido, utilizou-se recursos computacionais que contribuíram para o desenvolvimento e análise dos resultados do sistema de controle.

A estrutura proposta apresentou uma redução considerável na quantidade de regras, no tempo de processamento e na complexidade do modelo original. Os resultados obtidos mostraram a viabilidade da aplicação dessa abordagem no controle de posição do motor CC.

5 Conclusão Geral

A utilização de técnicas de inteligência artificial tem sido relevante para o controle de processos. Tanto os sistemas *fuzzy* quanto os conjuntos aproximados podem desempenhar um papel fundamental como aproximadores de funções de controle através de modelos baseados em regras. Para o exemplo prático da aplicação desses conceitos, utilizou-se o motor de corrente contínua e o controle de posição do seu eixo.

Esse projeto destacou duas abordagens para o desenvolvimento do controlador aproximado. A primeira tratou sobre um processo de conversão de modelos *fuzzy* em modelos aproximados. Esse procedimento foi realizado com a determinação dos conjuntos *crisp* baseados nas funções de pertinência. Através da compilação manual determinou-se o sistema de informação e as regras associadas ao modelo *fuzzy*. Neste caso, a estrutura do controlador aproximado permaneceu com os ganhos de escala que foram usados em pequenos ajustes para uma melhor aproximação ao comportamento do controlador *fuzzy*. Esse fato mostrou a flexibilidade do modelo obtido.

O controlador aproximado realiza o mapeamento de funções de controle através das regras aproximadas que são ativadas de acordo a necessidade do processo. Isso possibilitou obter uma característica de compensação mais adequada em relação ao controlador PID convencional que possui os ganhos constantes.

A segunda abordagem apresentada no capítulo 4 exigiu uma menor intervenção humana em seu desenvolvimento. Com os dados do ensaio do sistema de controle *fuzzy* foi possível construir uma base de conhecimento para representar o comportamento de um novo controlador em termos de conjuntos aproximados. Deste modo, utilizou-se recursos computacionais desenvolvidos para o processamento dos conjuntos aproximados, permitindo o trabalho com uma maior quantidade de amostras no SI e a realização prática do objetivo proposto.

Nos dois casos foram definidas estruturas de controle baseadas em conjuntos *crisp*, ou seja, sem a necessidade dos procedimentos de fuzificação e defuzificação existentes no controlador *fuzzy*. Na segunda abordagem essa característica proporcionou ao algoritmo do controlador aproximado uma redução no tempo de processamento de aproximadamente 72 vezes comparado ao controlador *fuzzy*. Uma característica importante em sistemas de tempo real. Essa redução significativa é atribuída a simplicidade computacional estabelecida na estrutura de controle e mostra a viabilidade da aplicação desses sistemas em processadores de baixo custo.

Os resultados experimentais e das simulações computacionais demonstram que a teoria dos conjuntos aproximados é uma alternativa para o projeto de controladores com características estáveis e o comportamento dinâmico propício para o sistema.

No intuito de dar continuidade aos estudos envolvidos nesta dissertação, sugere-se para os trabalhos posteriores:

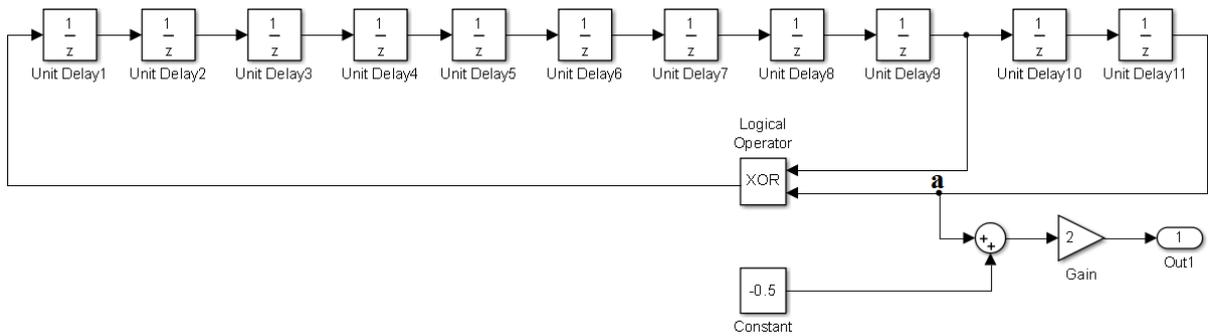
A aplicação das abordagens apresentadas em processos mais complexos, tais como sistemas com múltiplas entradas e saídas ou em sistemas com múltiplos sensores. Neste caso, pode ser avaliado a contribuição dos redutos na eliminação das entradas e saídas ou dos sensores que tiverem redundâncias em suas informações.

A utilização dos conjuntos aproximados para o projeto de controladores *fuzzy*, através da geração de regras difusas ou sintonia de parâmetros.

É pertinente uma abordagem com o estudo e desenvolvimento direto de um controlador aproximado a partir do processo especificado, sem a utilização de informações estabelecidas por outro controlador. Nesse contexto, pode-se estudar métodos de aprendizado de máquina que considere atributos relevantes para extrair o conhecimento em forma de regras.

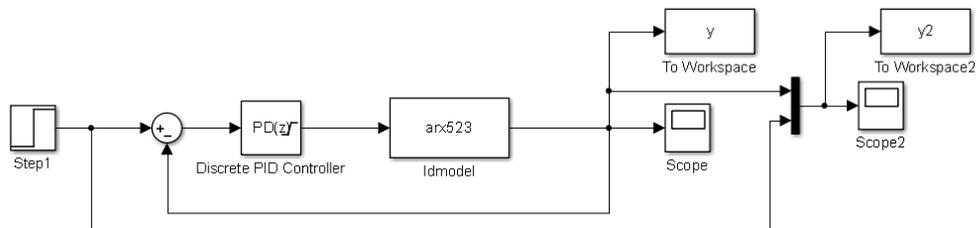
Apêndice A – Diagramas de Blocos usados no Sistema de Controle

A.1. Circuito para gerar o sinal PRBS (capítulo 4).

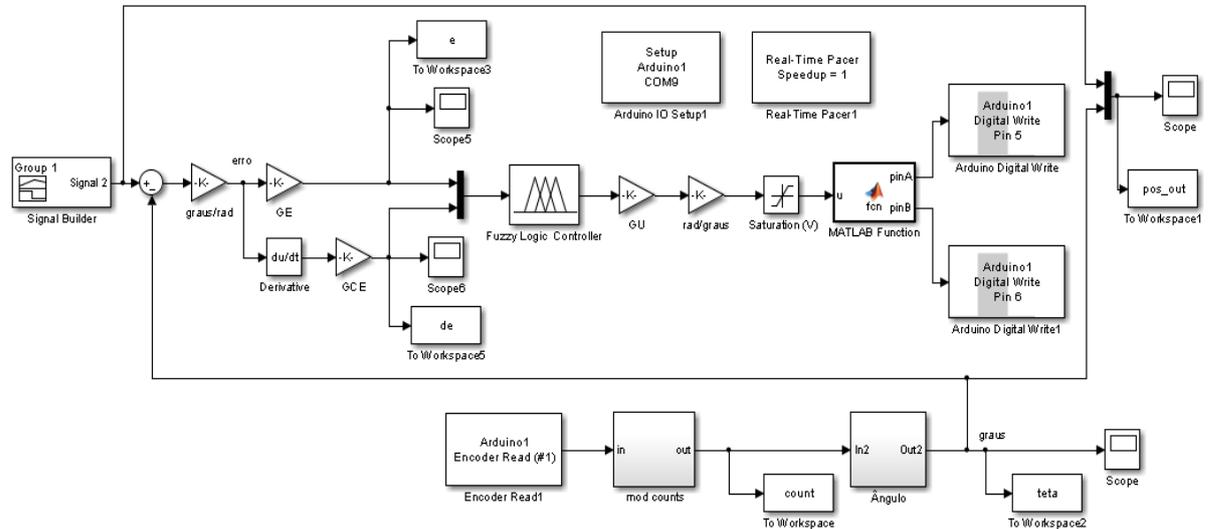


O sinal no ponto “a” varia entre 0 e 1. Após a polarização e amplificação os níveis do sinal de saída variam entre -1 e 1.

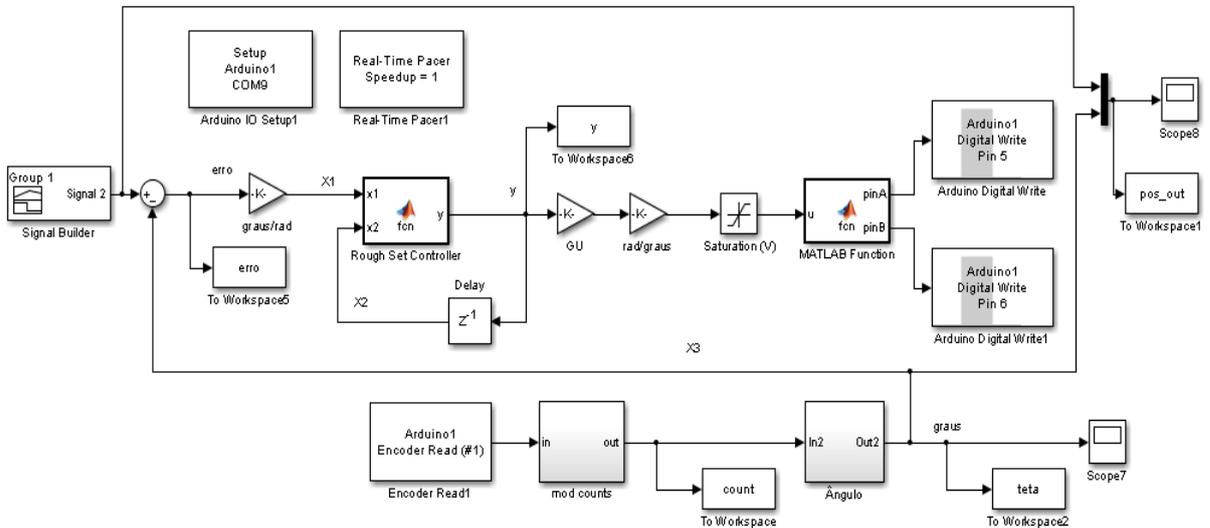
A.2. Modelo do controlador PD convencional discreto usado no ajuste dos ganhos (capítulo 4).



A.3. Modelo do sistema de controle *fuzzy* utilizado no controle de posição do eixo do motor CC (capítulo 4).



A.4. Modelo do sistema de controle aproximado (*rough*) utilizado no controle de posição do eixo do motor CC (capítulo 4).



O código fonte do controlador aproximado com todas as regras envolvidas foi inserido no bloco *Embedded Matlab[®] Function*.

Apêndice B – Códigos Fonte do Sistema de Controle

B.1. Código fonte do controlador Aproximado (capítulo 3)

```

function y = fcn(x1, x2)
% Valores extremos do conjunto crisp do modelo Rough
A1k=-1;      A1m=-0.7;      B1k=-1;      B1m=-0.42;
A2k=-0.7;    A2m=-0.35;    B2k=-0.42;  B2m=0;
A3k=-0.35;   A3m=0;          B3k=0;      B3m=0.42;
A4k=0;       A4m=0.35;      B4k=0.42;   B4m=1;
A5k=0.35;    A5m=0.7;
A6k=0.7;     A6m=1;

c10=-3.42;   c11=0;          c12=0;
c20=-3.42;   c21=0;          c22=0;
c30=-1.1453; c31=2.3857;    c32=1.4397;
c40=-1.75;   c41=0;          c42=0;
c50=-1.1164; c51=2.5;        c52=1.5086;
c60=0;       c61=0;          c62=0;
c70=0.1983;  c71=2.7833;    c72=1.9881;
c80=-0.9150; c81=2.3857;    c82=1.9881;
c90=0;       c91=4.8857;    c92=4.0714;
c100=0;      c101=5;         c102=4.1667;
c110=-1.75;  c111=5;         c112=4.1667;
c120=-1.1667; c121=2.9167;   c122=2.0833;
c130=1.1667; c131=2.9167;   c132=2.0833;
c140=1.75;   c141=5;         c142=4.1667;
c150=0;      c151=5;         c152=4.1667;
c160=0;      c161=4.8857;   c162=4.0714;
c170=0.9150; c171=2.3857;   c172=1.9881;
c180=-0.1983; c181=2.7833;   c182=1.9881;
c190=0;      c191=0;         c192=0;
c200=1.1164; c201=2.5;       c202=1.5086;
c210=1.75;   c211=0;         c212=0;
c220=1.1453; c221=2.3857;   c222=1.4397;
c230=3.42;   c231=0;         c232=0;
c240=3.42;   c241=0;         c242=0;
y = 0;

if ((x1 >= A1k) & (x1 < A1m)) & ((x2 >= B1k) & (x2 < B1m))
    y = c10 + c11*x1 + c12*x2;
end

if ((x1 >= A2k) & (x1 < A2m)) & ((x2 >= B1k) & (x2 < B1m))
    y = c20 + c21*x1 + c22*x2;
end

if ((x1 >= A3k) & (x1 < A3m)) & ((x2 >= B1k) & (x2 < B1m))
    y = c30 + c31*x1 + c32*x2;
end

if ((x1 >= A4k) & (x1 < A4m)) & ((x2 >= B1k) & (x2 < B1m))
    y = c40 + c41*x1 + c42*x2;
end

```

```

if (((x1 >= A5k) & (x1 < A5m)) & ((x2 >= B1k) & (x2 < B1m)))
    y = c50 + c51*x1 + c52*x2;
end

if (((x1 >= A6k) & (x1 <= A6m)) & ((x2 >= B1k) & (x2 < B1m)))
    y = c60 + c61*x1 + c62*x2;
end

%-----
if (((x1 >= A1k) & (x1 < A1m)) & ((x2 >= B2k) & (x2 < B2m)))
    y = c70 + c71*x1 + c72*x2;
end

if (((x1 >= A2k) & (x1 < A2m)) & ((x2 >= B2k) & (x2 < B2m)))
    y = c80 + c81*x1 + c82*x2;
end

if (((x1 >= A3k) & (x1 < A3m)) & ((x1 >= B2k) & (x1 < B2m)))
    y = c90 + c91*x1 + c92*x2;
end

if (((x1 >= A4k) & (x1 < A4m)) & ((x1 >= B2k) & (x1 < B2m)))
    y = c100 + c101*x1 + c102*x2;
end

if (((x1 >= A5k) & (x1 < A5m)) & ((x1 >= B2k) & (x1 < B2m)))
    y = c110 + c111*x1 + c112*x2;
end

if (((x1 >= A6k) & (x1 <= A6m)) & ((x1 >= B2k) & (x1 < B2m)))
    y = c120 + c121*x1 + c122*x2;
end

%-----
if (((x1 >= A1k) & (x1 < A1m)) & ((x1 >= B3k) & (x1 < B3m)))
    y = c130 + c131*x1 + c132*x2;
end

if (((x1 >= A2k) & (x1 < A2m)) & ((x1 >= B3k) & (x1 < B3m)))
    y = c140 + c141*x1 + c142*x2;
end

if (((x1 >= A3k) & (x1 < A3m)) & ((x1 >= B3k) & (x1 < B3m)))
    y = c150 + c151*x1 + c152*x2;
end

if (((x1 >= A4k) & (x1 < A4m)) & ((x1 >= B3k) & (x1 < B3m)))
    y = c160 + c161*x1 + c162*x2;
end

if (((x1 >= A5k) & (x1 < A5m)) & ((x1 >= B3k) & (x1 < B3m)))
    y = c170 + c171*x1 + c172*x2;
end

if (((x1 >= A6k) & (x1 <= A6m)) & ((x1 >= B3k) & (x1 < B3m)))
    y = c180 + c181*x1 + c182*x2;
end
%-----

```

```

if (((x1 >= A1k) & (x1 < A1m)) & ((x1 >= B4k) & (x1 <= B4m)))
    y = c190 + c191*x1 + c192*x2;
end

if (((x1 >= A2k) & (x1 < A2m)) & ((x1 >= B4k) & (x1 <= B4m)))
    y = c200 + c201*x1 + c202*x2;
end

if (((x1 >= A3k) & (x1 < A3m)) & ((x1 >= B4k) & (x1 <= B4m)))
    y = c210 + c211*x1 + c212*x2;
end

if (((x1 >= A4k) & (x1 < A4m)) & ((x1 >= B4k) & (x1 <= B4m)))
    y = c220 + c221*x1 + c222*x2;
end

if (((x1 >= A5k) & (x1 < A5m)) & ((x1 >= B4k) & (x1 <= B4m)))
    y = c230 + c231*x1 + c232*x2;
end

if (((x1 >= A6k) & (x1 <= A6m)) & ((x1 >= B4k) & (x1 <= B4m)))
    y = c240 + c241*x1 + c242*x2;
end
end
end

```

B.2. Código usado para estabelecer o sinal de saída (capítulo 4)

```

function [pinA, pinB]=fcu(u)
%#codegen
%#eml
if u>0
    pinA = uint8(u);
    pinB = uint8(0);
else
    pinA = uint8(0);
    pinB = uint8(-u);
end
end

```

B.3. Código fonte do controlador Aproximado (capítulo 4)

```

function y = fcn(x1, x2)
%Variaveis discretizadas x1=erro, x2=derivada do erro, y=saída
x1k_1=-0.031849; x1m_1=0.692; x2k_1=-0.001219; x2m_1=0.0871;
x1k_2=-0.031849; x1m_2=0.692; x2k_2=-0.002940; x2m_2=-0.001219;
x1k_3=-1.58; x1m_3=-0.050949; x2k_3=-0.001219; x2m_3=0.0871;
x1k_4=-1.58; x1m_4=-0.050949; x2k_4=-0.151; x2m_4=-0.002940;
x1k_5=-1.58; x1m_5=-0.050949; x2k_5=-0.002940; x2m_5=-0.001219;
x1k_6=-0.050949; x1m_6=-0.031849; x2k_6=-0.001219; x2m_6=0.0871;
x1k_7=-0.050949; x1m_7=-0.031849; x2k_7=-0.002940; x2m_7=-0.001219;
x1k_8=-0.050949; x1m_8=-0.031849; x2k_8=-0.151; x2m_8=-0.001219;
x1k_9=-0.031849; x1m_9=0.692; x2k_9=-0.151; x2m_9=-0.002940;

%Saídas
y1k = -0.005728; y1m = 0.014500;
y2k = -0.004570; y2m = 0.008710;
y3k = -0.150999; y3m = 0.001030;

```

```

y4k = -0.026799; y4m = 0.001372;
y5k = -0.084200; y5m = 0.002900;
y6k = -0.011046; y6m = 0.005750;
y7k = -0.006383; y7m = 0.002330;
y8k = -0.006361; y8m = 0.001130;
y9k = -0.000110; y9m = 0.003846;

%Condições iniciais
y=0;
cont_regra1=0;
cont_regra2=0;
cont_regra3=0;
cont_regra4=0;
cont_regra5=0;
cont_regra6=0;
cont_regra7=0;
cont_regra8=0;
cont_regra9=0;

%Regra 1
if (((x1 >= x1k_1) & (x1 <= x1m_1)) & ((x2 >= x2k_1) & (x2 <= x2m_1)))
    % interpolacao
    % Para a saída: THEN(-0.005728299≤y≤0.014500000)
    y = y1k + ((y1m - y1k)/2)*((x1-x1k_1)/(x1m_1 - x1k_1) +
        (x2-x2k_1)/(x2m_1 - x2k_1));
    cont_regra1= cont_regra1 +1
end

%Regra 2
if (((x1 >= x1k_2) & (x1 <= x1m_2)) & ((x2 >= x2k_2) & (x2 <= x2m_2)))
    % interpolacao
    % Para a saída: THEN(-0.004570000≤y≤0.008710000)
    y = y2k + ((y2m - y2k)/2)*((x1-x1k_2)/(x1m_2 - x1k_2) +
        (x2-x2k_2)/(x2m_2 - x2k_2));
    cont_regra2= cont_regra2 +1
end

%Regra 3
if (((x1 >= x1k_3) & (x1 <= x1m_3)) & ((x2 >= x2k_3) & (x2 <= x2m_3)))
    % interpolacao
    % Para a saída: THEN(-0.150999993≤y≤0.001030000)
    y = y3k + ((y3m - y3k)/2)*((x1-x1k_3)/(x1m_3 - x1k_3) +
        (x2-x2k_3)/(x2m_3 - x2k_3));
    cont_regra3= cont_regra3 +1
end

%Regra 4
if (((x1 >= x1k_4) & (x1 <= x1m_4)) & ((x2 >= x2k_4) & (x2 <= x2m_4)))
    % interpolacao
    % Para a saída: THEN(-0.026799999≤y≤0.001372916)
    y = y4k + ((y4m - y4k)/2)*((x1-x1k_4)/(x1m_4 - x1k_4) +
        (x2-x2k_4)/(x2m_4 - x2k_4));
    cont_regra4= cont_regra4 +1
end

%Regra 5
if (((x1 >= x1k_5) & (x1 <= x1m_5)) & ((x2 >= x2k_5) & (x2 <= x2m_5)))
    % interpolacao
    % Para a saída: THEN(-0.084200002≤y≤0.002900000)
    y = y5k + ((y5m - y5k)/2)*((x1-x1k_5)/(x1m_5 - x1k_5) +

```

```

        (x2-x2k_5)/(x2m_5 - x2k_5));
    cont_regra5= cont_regra5 +1
end

%Regra 6
if (((x1 >= x1k_6) & (x1 <= x1m_6)) & ((x2 >= x2k_6) & (x2 <= x2m_6)))
    % interpolacao
    % Para a saída: THEN(-0.011046429≤y≤0.005750000)
    y = y6k + ((y6m - y6k)/2)*((x1-x1k_6)/(x1m_6 - x1k_6) +
        (x2-x2k_6)/(x2m_6 - x2k_6));
    cont_regra6= cont_regra6 +1
end

%Regra 7
if (((x1 >= x1k_7) & (x1 <= x1m_7)) & ((x2 >= x2k_7) & (x2 <= x2m_7)))
    % interpolacao
    % Para a saída: THEN(-0.006383702≤y≤0.002330000)
    y = y7k + ((y7m - y7k)/2)*((x1-x1k_7)/(x1m_7 - x1k_7) +
        (x2-x2k_7)/(x2m_7 - x2k_7));
    cont_regra7= cont_regra7 +1
end

%Regra 8
if (((x1 >= x1k_8) & (x1 <= x1m_8)) & ((x2 >= x2k_8) & (x2 <= x2m_8)))
    % interpolacao
    % Para a saída: THEN(-0.006361744≤y≤0.001130000)
    y = y8k + ((y8m - y8k)/2)*((x1-x1k_8)/(x1m_8 - x1k_8) +
        (x2-x2k_8)/(x2m_8 - x2k_8));
    cont_regra8= cont_regra8 +1
end

%Regra 9
if (((x1 >= x1k_9) & (x1 <= x1m_9)) & ((x2 >= x2k_9) & (x2 <= x2m_9)))
    % interpolacao
    % Para a saída: THEN(-0.000110248≤y≤0.003846828)
    y = y9k + ((y9m - y9k)/2)*((x1-x1k_9)/(x1m_9 - x1k_9) +
        (x2-x2k_9)/(x2m_9 - x2k_9));
    cont_regra9= cont_regra9 +1
end
end
end

```

Referências

- AGUIRRE, L. A. **Introdução à Identificação de Sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais**. 2^a. ed. Belo Horizonte: Editora UFMG, 2004.
- AHMED, H.; SINGH G.; BHARDWAJ V.; SAURAV S.; ARGAWAL S. **Controlling of DC Motor using Fuzzy logic controller**. Conference on advances in communication and control systems, pp.666-670, 2013.
- AKBARI-HASANJANI, R.; JAVADI, S.; SABBAGHI-NADOOSHAN, R. **DC motor speed control by self-tuning fuzzy PID algorithm**. Transactions of the Institute of Measurement and Control, vol. 37, no. 2, pp. 164-176, 2015.
- ALMEIDA, J. P. L. S. **Técnicas Fuzzy Aplicadas ao Controle Descentralizado**. 2014. 82 p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2014.
- AMLASHI, N. J. S. **Design and Implementation of Fuzzy Position Control System for Tracking Applications and Performance Comparison with Conventional PID**. International Journal of Artificial Intelligence, vol.1, no.1, pp.31-44, 2012.
- ANTUNES, F. **Uma nova abordagem para representações e identificações de classes de sistemas dinâmicos não lineares**. 2007. 67 p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá, 2007.
- BEZERRA, S. T. M. **Sistema Fuzzy para Controle Piezométrico de Sistemas de Distribuição de Água Visando à Economia de Água e Energia**. 2009. 142 p. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal da Paraíba, João Pessoa-PB, 2009.
- CHEN, Z; ZHU Y.; DI Y.; FENG S. **A dynamic resource scheduling method based on fuzzy control theory in cloud environment**. Hindawi Publishing Corporation Journal of Control Science and Engineering, no. 34, pp.1-10, 2015.
- DECARLO, R. A.; ZAK, S. H.; MATTHEWS, G. P. **Variable structure control of nonlinear multivariable systems: a tutorial**. Proceedings of the IEEE, vol. 76, no. 3, pp. 212-232, 1988.
- DUBOIS, D.; PRADE, H. **Rough fuzzy sets and fuzzy rough sets**. International Journal of General System, vol. 17, pp. 191–209, 1990.
- DURSUN, E. H.; DURDU, A. **Position control by using PD type fuzzy logic: Experimental study on rotary servo system**. IEEE International Conference on Electronics Computers and Artificial Intelligence, Ploiesti, România, pp. 1-6, 2016.
- EL-NAGAR, A. M.; EL-BARDINI, M. **Practical implementation for the interval type-2 fuzzy PID controller using a low cost microcontroller**. Ain Shams Engineering Journal, vol. 5, no. 2, pp. 475-487, 2014.

FITZGERALD, A. E.; KINGSLEY, C. J.; UMAS, S. D. **Máquinas Elétricas: Com introdução à eletrônica de potência**. 6ª. ed. Porto Alegre: Bookman, 2006.

FRANKLIN, J; POWEL, D.; EMAMI-NAENI, A. **Sistemas de Controle para Engenharia**. 6ª ed. Porto Alegre: Bookman, 2013.

GOMIDE, F. A. C.; GUDWIN, R. R. **Modelagem, controle, sistemas e lógica fuzzy**. SBA Controle e Automação, vol. 4, no. 3, pp. 97-115, 1994.

GUARACY, F.; PINHEIRO, C.; FERREIRA, L. **Converting fuzzy models to rough models**. IEEE International Conference on Industrial Informatics, pp. 37-42, 2014.

GUIMARÃES A. O. **Algoritmo genético aplicado no controle de posição do rotor de um motor de corrente contínua com rejeição a distúrbios por ação feedforward**. 2013. 72p. Dissertação (Mestrado em Sistemas de Comunicação e Informação) – Universidade Federal Rural do Semi-Árido. Mossoró – RN, 2013.

HAT, M. K; IBRAHIM B. S. K. K.; MOHD T. A. T.; HASSAN, M.K. **Model based design of PID controller for BLDC motor with implementation of embedded arduino mega controller**. International Conference on Electrical and Electronic Engineering. Malaysia, vol. 10, no. 19, pp. 8588-8594, October 2015.

HERCEG, M.; KVASNICA, M.; FIKAR, M. **Transformation of Fuzzy Takagi-Sugeno Models into Piecewise Affine Models**. Proceedings of the International Conference on Rough Sets and Intelligent System Paradigms, Warsaw, vol. 4585, pp. 211-220, 2007.

JIANG, Y.; ZHAO, H.; FU, H. **A control method to avoid obstacles for an intelligent car based on rough sets and neighborhood systems**. International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE, pp. 66-70, 2015.

LIMA, I. L.; SANTOS, F. A. O.; PINHEIRO, C. A. M. **Inteligência Artificial**. 1ª ed. Rio de Janeiro: Elsevier, 2014.

LORDELO, A.; FAZZOLARI, H. **Controle robusto de posição do motor DC com estrutura variável e modos deslizantes de ordem completa**. Congresso Brasileiro de Automática. pp. 4624-4630, 2012.

MACHADO, K; PINHEIRO, C. **Rough controllers with state feedback**. Engineering Applications of Artificial Intelligence, vol. 26, no. 10, pp. 2467-2479, 2013.

MAMDANI, E. H. **Applications of fuzzy algorithms for control of simple dynamic plant**. Proceedings of the institution of electrical engineers, vol. 121, no. 12, pp. 1585-1588, 1974.

NAMAZOV, M; BASTURK, O. **DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods**. Turkish Journal of Fuzzy Systems, vol. 1, no. 1, pp. 36-54, 2010.

OBAID, Z. A.; SULAIMAN, N.; MARHABAN, M. H.; HAMIDON, M. N. **Analysis and performance evaluation of PD-like fuzzy logic controller design based on MATLAB and FPGA**. International Journal of Computer Science, vol. 10, pp. 11, 2010.

OGATA, K. **Engenharia de Controle Moderno**. 4^a ed. São Paulo: Prentice Hall, 2003.

OHRN, A.; KOMOROWSKI, J. **Rosetta - a rough sets toolkit for analyses of data**. Third International Joint Conference on Information Sciences, pp. 403–407, 1997.

OLIVEIRA, R. H. **Modelagem de Sistemas MIMO baseada em Conjuntos Aproximados**. 2013. 140 p. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá-MG, 2013.

PAWLAK, Z. **Rough sets**. International Journal of Computer and Information Sciences, vol. 11, no. 5, pp. 341-356, 1982.

PAWLAK, Z. **Rough Sets: Theoretical Aspects of Reasoning about Data**. London: Netherlands, Kluwer Academic Publishers, 1991.

PAWLAK, Z.; SKOWRON, A. **Rudiments of rough sets**. Information sciences, vol. 177, no.1, pp. 3-27, 2007.

PILTAN, F.; RAHMANI M.; ESMAEILI M.; CHERAGHI H. **Design FPGA-Based Fuzzification Algorithm for Model-free Control Techniques**. International Journal of Hybrid Information Technology, vol. 9, no. 8, pp. 243-264, 2016.

PINHEIRO, C. A. M. **Análise e projeto de sistemas de controle fuzzy: uma abordagem no domínio da frequência**. 2000. 133 p. Tese (Doutorado em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas-SP, 2000.

PINHEIRO, C.; CAMATTA, U.; REZEK, A. **Rough controller synthesis**. Fuzzy Logic: Controls, Concepts, Theories and Applications. INTECH, pp. 281–300, 2012.

PINHEIRO, C.; GOMIDE, F. **Equivalência entre controladores fuzzy e compensadores clássicos**. Simpósio Brasileiro de Automação Inteligente, São Paulo, pp. 08-10, 1999.

PINHEIRO, C.; GOMIDE F.; CARPINTEIRO O.; LIMA I. **Granular Synthesis of Rule-Based Models and Function Approximation using Rough Sets**. Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, JingTao Yao, IGI Global, Hershey, pp. 408-425, 2010.

PINTO, J. E. M. G. **Aplicação prática do método de sintonia de controladores PID utilizando o método do relé com histerese**. 2014. 116 p. Dissertação (Mestrado em Ciências) – Universidade Federal do Rio Grande do Norte, Natal-RN, 2014.

RATHORE, N. S.; SINGH, V. P.; CHAUHAN, D. P. S. **ISE based PID Controller Tuning for Position Control of DC Servo-Motor using LJ**. IEEE International Conference on Signal Processing, Computing and Control, pp. 125-128, 2015.

SINHA, A.; PRASOON P.; BHARADWAJ P. K.; RANASINGHE A. **Nonlinear autonomous control of a two-wheeled inverted pendulum mobile robot based on sliding mode**. International Conference on Computational Intelligence and Networks (CINE). IEEE, pp. 52-57, 2015.

STELLA, G. N. D. **Aplicando a metodologia de desenvolvimento baseado em modelos para funções de software automotivo**. 2015. 123 p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2015.

SUDHA, G.; ANITA R. **Performance Based Comparison between Various Z-N Tuning PID and Fuzzy Logic PID Controller in Position Control System of DC Motor**. International Journal on Soft Computing, vol. 3, no. 3, pp. 55-67, 2012.

TAJIRI, L. L. **Proposição de um controlador digital para conversores buck e boost usando a teoria de conjuntos aproximados**. 2009. 111p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá, 2009.

TAKAGI, T; SUGENO, M. **Fuzzy identification of systems and its applications to modeling and control**. IEEE transactions on systems, man, and cybernetics, vol. 15, no. 1, pp. 116-132, 1985.

VERMAAS, L. L. G. **Aprendizado Supervisionado de Sistemas de Inferência Fuzzy Aplicados em Veículos Inteligentes**. 2010. 117 p. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá-MG, 2010.

ZADEH, L. A., **Fuzzy Sets**. Information and Control, vol. 8, no. 3, pp. 338-353, 1965.

ZADEH, M. H.; YAZDIAN, A.; MOHAMADIAN, M. **Robust Position Control in DC Motor by Fuzzy Sliding Mode Control**. International Symposium on Power Electronics, Electrical Drives, Automation and Motion. IEEE, pp. 1413-1418, 2006.